

NASA-CR-167949
19820025515

NASA CR-167949
PWA-5774-21



STRUCTURAL TAILORING OF ENGINE BLADES (STAEBL)

By:
C. E. Platt, T. K. Pratt and K. W. Brown

UNITED TECHNOLOGIES CORPORATION
POWER SECTOR
PRATT & WHITNEY AIRCRAFT GROUP
COMMERCIAL PRODUCTS DIVISION

Prepared for:
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

NASA Lewis Research Center
Contract NAS 3-22525



NF02707

LIBRARY COPY

SEP 2 1982

LANCLEY RESEARCH CENTER
LIBRARY, NASA
HAMPTON, VIRGINIA

1. Report No. NASA CR-167949		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle STRUCTURAL TAILORING OF ENGINE BLADES (STAEBL)				5. Report Date June 1982	
				6. Performing Organization Code	
7. Author(s) C. E. Platt, T. K. Pratt and K. W. Brown				8. Performing Organization Report No. PWA-5774-21	
9. Performing Organization Name and Address United Technologies Corporation Pratt & Whitney Aircraft Group Commercial Products Division East Hartford, CT				10. Work Unit No.	
				11. Contract or Grant No. NAS 3-22525	
				13. Type of Report and Period Covered Interim Report	
12. Sponsoring Agency Name and Address National Aeronautics & Space Administration Washington, DC 20546				14. Sponsoring Agency Code	
15. Supplementary Notes Project Manager, C. C. Chamis NASA Lewis Research Center 21000 Brookpark Road, M. S. 49-6 Cleveland, OH 44135					
16. Abstract A mathematical optimization procedure was developed for the Structural Tailoring of Engine Blades and was used to structurally tailor two engine fan blades constructed of composite materials without midspan shrouds. The first was a solid blade made from superhybrid composites, and the second was a hollow blade with metal-matrix composite inlays. Three major computerized functions were needed to complete the procedure: approximate analysis with the established input variables, optimization of an objective function, and refined analysis for design verification.					
17. Key Words (Suggested by Author(s)) Approximate Analysis Mathematical Optimization Objective Function Refined Analysis			18. Distribution Statement Unclassified, Unlimited		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages	22. Price*

* For sale by the National Technical Information Service, Springfield, Virginia 22161

TABLE OF CONTENTS

Section	Title	Page
1.0	SUMMARY	1
2.0	INTRODUCTION	3
3.0	STAEBL PROCEDURE	5
3.1	Overview	5
3.2	General Approach to STAEBL Procedure Design	5
3.3	Input to Fan Blade Structural Tailoring Procedure	8
3.3.1	Aerodynamic Stage	8
3.3.2	Support Structure	8
3.3.3	Operating Conditions	8
3.3.4	Materials	9
3.3.5	Objective Function	9
3.3.6	Constraints	9
3.3.7	Design Variables	10
3.4	Approximate Analysis	10
3.4.1	Stress and Vibration Analysis	10
3.4.1.1	Approach	10
3.4.1.2	Solution Procedure	17
3.4.1.3	Verification	23
3.4.2	Foreign Object Damage Analysis	33
3.4.3	Local Foreign Object Damage Analysis	34
3.4.4	Approximate Analysis System	35
3.5	Optimization Method	35
3.5.1	General Optimization Theory and Background	36
3.5.2	COPEs/CONMIN Exact Analysis	39
3.5.2.1	Choice of Search Parameters for COPEs/CONMIN	40
3.5.2.2	Scaling of Design Variables in COPEs/CONMIN	44
3.5.2.3	Number of Function Calls for COPEs/CONMIN	45
3.5.3	COPEs/CONMIN Approximate Analysis	46
3.5.4	COPEs/CONMIN Interfaces to Vibration, Flutter, and Stress Programs	50
3.5.5	Test Cases for STAEBL Optimization Procedure	53
3.6	Refined Analysis	57
4.0	RESULTS	61
4.1	Introduction	61
4.2	Demonstration Considerations	61
4.2.1	Aerodynamic Design	61
4.2.2	Objective Function	61
4.2.2.1	Engine Weight	63
4.2.2.2	Engine Cost	65
4.2.2.3	Maintenance Cost	65
4.2.3	Constraints on Behavior Variables	65
4.2.3.1	High Frequency Fatigue	65
4.2.3.2	Flight Cycle Fatigue	66
4.2.3.3	Bird Ingestion	67
4.3	STAEBL Procedure Demonstration	67
4.3.1	Reference Blade	67

TABLE OF CONTENTS (Continued)

<u>Section</u>	<u>Title</u>	<u>Page</u>
4.3.2	Tailored Blades	69
4.3.2.1	Hollow Blade with Composite Inlay	69
4.3.2.2	Superhybrid Composite Blade	72
5.0	CONCLUSIONS AND RECOMMENDATIONS	77
APPENDIX A	BLADE OPTIMIZATION PROGRAM ORGANIZATION	79
APPENDIX B	FORTRAN LISTING FOR HOLLOW BLADE TAILORING	85
APPENDIX C	FORTRAN LISTING REVISIONS FOR TAILORING SUPERHYBRID BLADE	329
REFERENCES		339
DISTRIBUTION LIST		341

LIST OF ILLUSTRATIONS

<u>Figure</u>	<u>Title</u>	<u>Page</u>
3.2-1	The Structural Tailoring of Engine Blades Procedure	6
3.2-2	The Objective Function Relates Airline Economics to Blade Design Variables	6
3.2-3	Blade Chord Optimization Appears to be a Simple Design Problem	7
3.2-4	Design Problem Complexity is Introduced by Structural Constraints	7
3.4-1	Model Used in Approximate Analyses of a Spanwise Increment of Blade	12
3.4-2	Beam Model Verification with NASTRAN Vibration Analysis of Assembly of Rectangular, Isotropic, Cantilevered Beams	12
3.4-3	Torsional Frequencies Obtained with Planes Remain Plane Section Constraint	13
3.4-4	Low Torsional Frequencies Produced by the Free Warping Condition	14
3.4-5	Constrained Warping Condition and Thin Section Torsional Stiffness Correlated Well Using an Approximate Section Warping Function	15
3.4-6	Results Obtained for a Twisted Flat Plate	16
3.4-7	Radially Stacked Beam Element as Represented by the Laminated Composite Beam Theory	17
3.4-8	Shape Functions for the Element Displacement Response	18
3.4-9	Rectangular Beam Ply Layup	20
3.4-10	Tilted Flat Plate Used in Approximate Analysis Test Case	25
3.4-11	Twisted Flat Plate Used in Approximate Analysis Test Case	26
3.4-12	Airfoil Warping Defined by Spanwise Twist Gradient	29
3.4-13	Approximate Root Centrifugal Stress in the Energy Efficient Engine Fan Blade Compared with NASTRAN	30
3.4-14	NASTRAN Hollow Blade Model	31

LIST OF ILLUSTRATIONS (Continued)

Figure	Title	Page
3.4-15	Hollow Airfoil Beam	31
3.4-16	Laminated Airfoil Beam	32
3.5-1	Feasible Region Is Union of All Points that Satisfy All Constraints	37
3.5-2	Line Search Terminates Either at Minimum of Objective Function or at a Constant Boundary. Sequence of line searches converge to x_{opt} .	40
3.5-3	New Search Direction, s_j , Lies in the Usable Feasible Sector. The value of the push-off factor, θ_j , determines the orientation of the new search direction.	41
3.5-4	Constraint Thickness Parameter, CT, Determines when a Constraint is Satisfied, Violated, or Active	43
3.5-5	For Proper Choice of CT, Two Constraints Become Simultaneously Active so that Search Proceeds down the "Valley" Formed by the Constraints	43
3.5-6	Sequence of Approximate Analyses Using the Method of Feasible Directions Becomes Increasingly More Accurate as More Terms in the Taylor Series Are Determined.	48
3.5-7	COPES/CONMIN is Linked via Subroutine ANALIZ to Approximate Analyses	51
3.5-8	STAEBL Optimization Procedure May Not Converge to Global Optimum if Feasible Region is Disjoint	53
3.5-9	STAEBL Optimization Procedure with Three Flutter Constraints Converges to Global Optimum	54
3.5-10	Exact and Approximate Analyses for STAEBL Procedure Converge to Same Local Optimum	55
3.6-1	Refined Analysis of the Energy Efficient Engine Hollow Fan Blade	58
3.6-2	Energy Efficient Engine Hollow Fan Blade Airfoil Root Stress Predicted by Refined Analysis	59
3.6-3	Energy Efficient Engine Hollow Fan Blade Internal Surface Stress at Solid-to-Hollow Transition Predicted by Refined Analysis	59

LIST OF ILLUSTRATIONS (Continued)

<u>Figure</u>	<u>Title</u>	<u>Page</u>
3.6-4	Refined Analysis of the Energy Efficient Engine Hollow Fan Blade Supports the Conclusion that It Would Not Flutter	60
4.2-1	Fan System Weight Is a Simple Function of Individual Airfoil Weight	64
4.2-2	Full Span Hollow Fan Blade Life Is Derived from Solid Blade Experience Limits	66
4.3-1	Reference Blade Resonance Diagram	68

LIST OF TABLES

<u>Table</u>	<u>Title</u>	<u>Page</u>
3.4-I	NASTRAN TEST CASE FOR PANEL WITH 30 DEGREE PRETWIST	16
3.4-II	COMPARISON OF STATIC ANALYSIS PROCEDURE WITH NASTRAN TEST CASES	24
3.4-III	TILTED FLAT PLATE TIP DEFLECTIONS	25
3.4-IV	TWISTED (30 Degrees) FLAT PLATE TIP DEFLECTIONS	26
3.4-V	APPROXIMATE ANALYSIS NATURAL FREQUENCIES COMPARED WITH NASTRAN	27
3.4-VI	APPROXIMATE ANALYSIS NATURAL FREQUENCIES COMPARED WITH NASTRAN	27
3.4-VII	APPROXIMATE ANALYSIS NATURAL FREQUENCIES COMPARED WITH NASTRAN	27
3.4-VIII	APPROXIMATE PROCEDURE ANALYSIS OF FAN BLADE NATURAL FREQUENCY COMPARED WITH NASTRAN	28
3.4-IX	HOLLOW BLADE FREQUENCY COMPARISON	32
3.4-X	COMPOSITE REINFORCED BLADE FREQUENCY COMPARISON	32
3.4-XI	COMPOSITE REINFORCED BLADE FREQUENCY COMPARISON	33
3.4-XII	APPROXIMATE ANALYSIS SYSTEM MODULE OUTPUTS	36
3.5-I	UNSCALED AND SCALED VALUES OF GRADIENT OF OBJECTIVE FUNCTION RELATIVE TO THE DESIGN VARIABLES FOR HOLLOW FAN BLADE	45
3.5-II	FORTRAN NAMES AND GLOBAL LOCATIONS IN BLOBCM FOR DESIGN VARIABLES, OBJECTIVE FUNCTION, AND CONSTRAINTS FOR STAEBL BLADE APPLICATIONS	52
3.5-III	HOLLOW FAN BLADE HAS 3 ACTIVE CONSTRAINTS FOR FINAL DESIGN	56
4.2-I	ENERGY EFFICIENT ENGINE STUDY AIRPLANE CHARACTERISTICS	62
4.2-II	ENERGY EFFICIENT ENGINE AIRLINE ECONOMIC MODEL	63
4.3-I	HOLLOW BLADE WITH COMPOSITE INLAY	69
4.3-II	MATERIALS PROPERTIES	71
4.3-III	HOLLOW COMPOSITE BLADE WITH COMPOSITE INLAY NATURAL FREQUENCY	72

LIST OF TABLES (Continued)

<u>Table</u>	<u>Title</u>	<u>Page</u>
4.3-IV	SUPERHYBRID COMPOSITE BLADE	73
4.3-V	SUPERHYBRID COMPOSITE BLADE NATURAL FREQUENCIES	75
A-I	APPROXIMATE ANALYSIS SUBROUTINE LIST	80
A-II	PRATT & WHITNEY AIRCRAFT PROPRIETARY ROUTINES	84

1.0 SUMMARY

A program was conducted to develop an optimization procedure for the Structural Tailoring of Engine Blades (STAEBL) and to demonstrate the procedure by using it to design two fan blades of composite materials.

The fan stage of the Energy Efficient Engine which was designed under NASA Contract NAS3-20646, was selected as the aerodynamic configuration upon which to base the tailoring demonstration.

Three major computerized capabilities were needed to complete the procedure: approximate analysis with the established input variables and trial values of design variables, optimization of an objective function, and refined analysis for design verification. To perform approximate analyses, an existing procedure was modified to evaluate low cycle fatigue, vibratory fatigue and foreign object damage for composite blade designs. To incorporate the effects of Direct Operating Cost plus Interest (the objective function) into the design, the COPES/CONMIN optimization program was used. NASTRAN was used to verify that the approximate analysis and optimization procedure had designed a blade that met all design criteria.

To demonstrate the STAEBL procedure in real design situations, it was used to structurally tailor two engine fan blades constructed of composite materials without midspan shrouds. The first was a solid blade made from superhybrid composites, and the second was a hollow blade with metal-matrix composite inlays.

Mathematical optimization applied to shroudless fan blade structural tailoring has been demonstrated to be a very powerful automated design procedure. It provides the capacity to simultaneously evaluate many design variables to optimize a comprehensive objective function while satisfying numerous design constraints.

This Page Intentionally Left Blank

2.0 INTRODUCTION

Fan and compressor blades are designed to provide aerodynamic performance and structural durability at minimum cost to the aircraft/engine operator through alternating aerodynamic and structural design iterations. The current structural design procedure requires that specified criteria be satisfied. These criteria have been derived by correlating particular analyses with extensive empirical experience. The designer is provided with an interactive computer system that conducts vibration, steady state stress, and ingestion analyses of proposed designs, modifies designs for reanalysis, compares results of analysis with criteria and assembles input for non-interactive flight cycle life analysis. The structural designer uses his personal experience to establish the path to follow to improve the design and decide when to terminate the search for the best design within the limits of the proposed aerodynamic configuration.

Thus, the current design procedures for turbine engine blades are partly engineering and partly art. The quality of the design is often the result of the judgment and experience of the engineer or engineering team that performed the task. The penalties for less than optimum designs are weight and cost. The cost penalty may appear as low efficiency or the wasted time of a long development cycle, fixing failures and improving performance. This usually results in less than optimum designs because the constraints of correcting a problem are always more severe than an original design, i.e., the space for the stage is fixed, the performance of other components interacts or is dependent on the part under development, etc. Once the design has been corrected, it is usually at the expense of cost or weight and degradation of the overall engine performance must be accepted.

It is apparent that current blade design procedures are limited by the need for the design engineer to incorporate his experience in trading design variables against each other. This problem is not peculiar to structural blade design; it arises to substantially the same degree in the design of other components. For these reasons, it is appropriate to initiate development of automated procedures to permit the optimized trade-off of variables against each other to improve the blade design and establish a foundation for application to other components. Such formalized optimum design procedures have been developed and used with considerable success for optimum structural design of linear static structures, and are now being developed and used with some success for the aeroelastic tailoring of fixed aircraft wings. The objective of the Structural Tailoring of Engine Blades Program, hereinafter referred to as STAEBL, was to develop a formalized optimum design procedure for engine blades which will meet all the aerothermomechanical design requirements in an aircraft engine environment. The STAEBL procedure will reduce human error in the blade design process by automating with mathematical precision what was formerly user judgement on an interactive system.

To meet the objective of the STAEBL program, six technical tasks were established as part of NASA Contract NAS3-22525:

- Task I: STAEBL Procedure - Design of the general STAEBL procedure.
- Task II: Input - Definition of STAEBL procedure input parameters including initial blade geometry, material properties, loads, weight and cost models, and design constraints.
- Task III: Approximate Analyses - Modification of existing beam analyses to perform vibration, stress and foreign object damage evaluations of composite blades.
- Task IV: Optimization Procedure - Identify a procedure which optimizes the objective function, direct operating cost plus interest, within limits of specified constraints.
- Task V: Refined Analyses - Establish a procedure for using NASTRAN to validate optimized blade designs.
- Task VI: Demonstration and Documentation - Demonstrate and document the STAEBL procedure by using it to tailor two alternate designs of the shroudless Energy Efficient Engine fan blade: one a solid blade made from superhybrid composites; and the second, a hollow blade with metal-matrix composite inlays.

The facility used for the STAEBL program was an IBM System 370 computing system. Most engineering problems were currently programmed for solution on three existing IBM 370 computers. Using IBM's latest virtual storage technology, these computers could accommodate fully computerized interactive design systems, general time-sharing, teleprocessing, real time management/information systems, and management and scientific batch processing.

Section 3.0 of this report presents a description of the STAEBL program design. Section 4.0 describes the results of the demonstration of the procedure and Section 5.0 presents Conclusions and Recommendations emanating from this program.

Appendix A presents the STAEBL procedure organization, identifying the various subroutines used in the overall system. Appendix B presents a complete FORTRAN listing of the STAEBL procedure for the hollow blade. The revisions to this FORTRAN listing for the superhybrid blade are provided in Appendix C.

3.0 STAEBL PROCEDURE

3.1 OVERVIEW

Airfoil structural design is a critical part of the aircraft turbine engine development process. The limitations imposed by durability requirements for the airfoils have a direct bearing on the aerodynamic performance that can be achieved. In addition, a significant portion of engine weight and engine cost is a simple multiple of airfoil weight. The airfoil design problem is complex. Chord, thicknesses at several locations, and internal constructions are selected to simultaneously satisfy vibration, ingestion and flight cycle durability requirements. Mathematical optimization techniques have been developed to expedite solution of this kind of tailoring problem which involves many design variables and many requirements. The airfoil application is particularly appropriate because the complex shapes defined by optimization do not increase manufacturing cost. The basic airfoil aerodynamic shapes are fabricated in accordance with three-dimensional numerical definitions which are readily modified to accept the results of structural tailoring.

Problems associated with structural tailoring of engine blades include: 1) engine blades are designed to operate in a dynamic environment by application of constraints which differ substantially from those applied to linear static structures; 2) analysts and/or designers have hesitated to develop optimization procedures for blades made from homogeneous materials because acceptable designs can be derived from past experience; and 3) finite element analyses, which are too time consuming to be used effectively in an optimization procedure, have been used in designing blades having advanced constructions such as those to be designed in this program.

The approach taken to assemble a procedure which solves these problems is described in Section 3.2. Inputs to the procedure are defined in 3.3. The approximate analyses and controlling method used in automated optimization are described in Sections 3.4 and 3.5. The substantiative refined analyses are described in Section 3.6.

3.2 GENERAL APPROACH TO STAEBL PROCEDURE DESIGN

Figure 3.2-1 summarizes the a procedure for the Structural Tailoring of Engine Blades. Design variables are initialized by input to the procedure and varied during optimization. Approximate analyses for low cycle fatigue, flutter, resonance, and foreign object damage are applied to evaluate position relative to constraints.

The objective function optimized in the STAEBL procedure is derived from the relationships illustrated in Figure 3.2-2. The complexity encountered in finding the design which optimizes this function can be illustrated by examining its relationship to blade chord (Figure 3.2-3). It appears to be simple, but becomes complicated when structural constraints are introduced (Figure 3.2-4). The design that the procedure selects must optimize user economics without violating the imposed constraints.

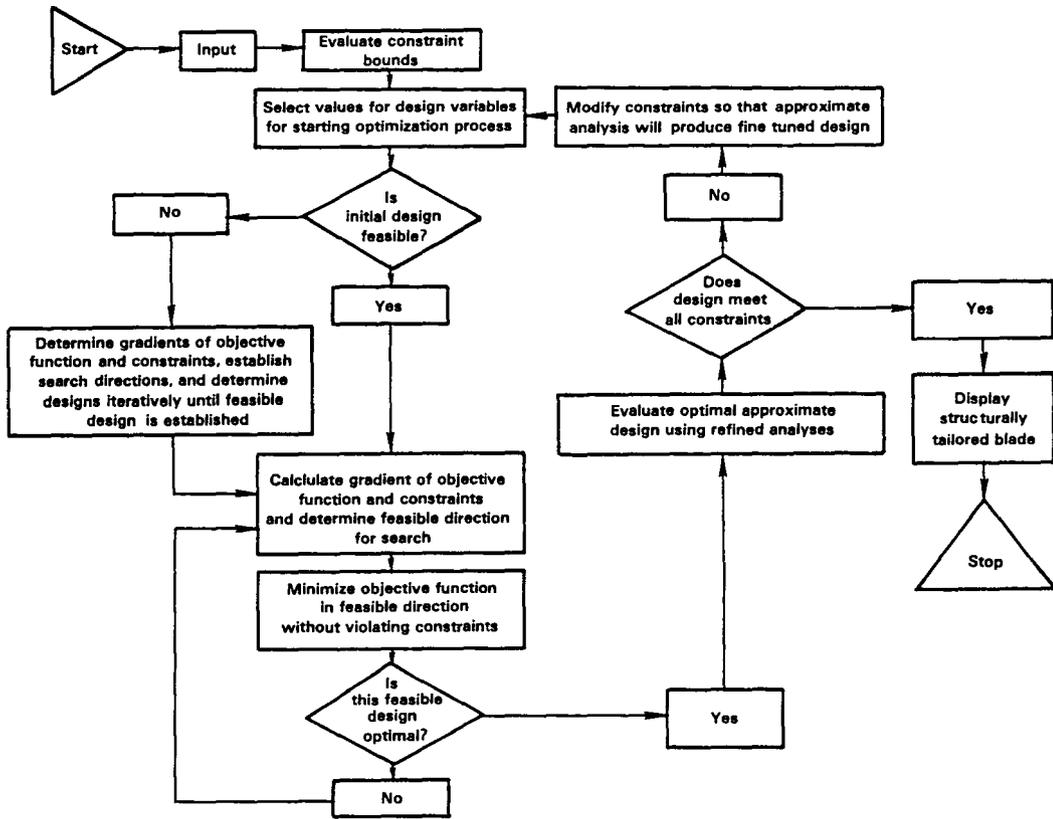


Figure 3.2-1 The Structural Tailoring of Engine Blades Procedure

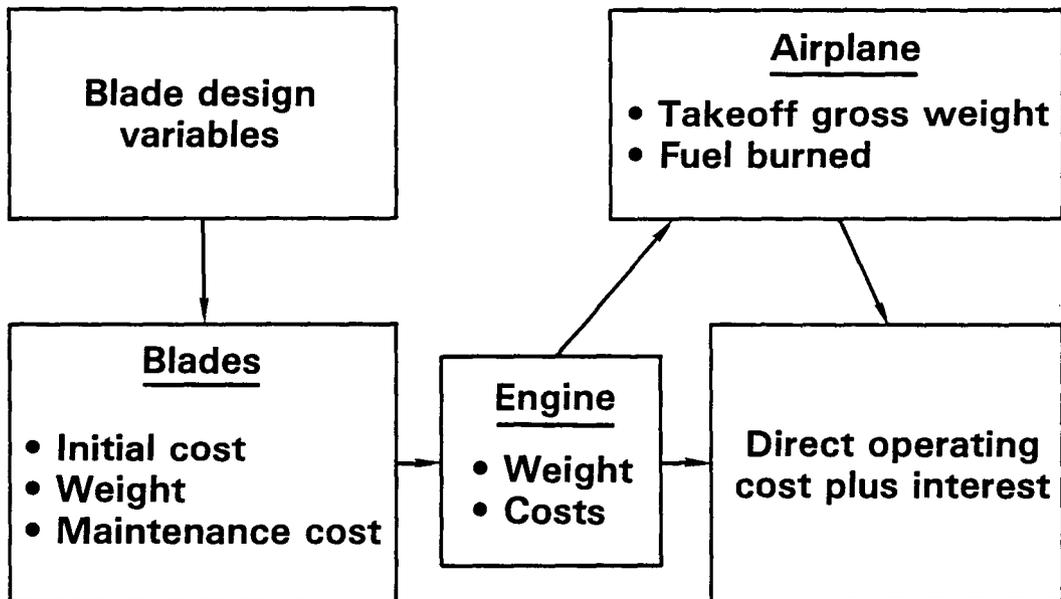


Figure 3.2-2 The Objective Function Relates Airline Economics to Blade Design Variables

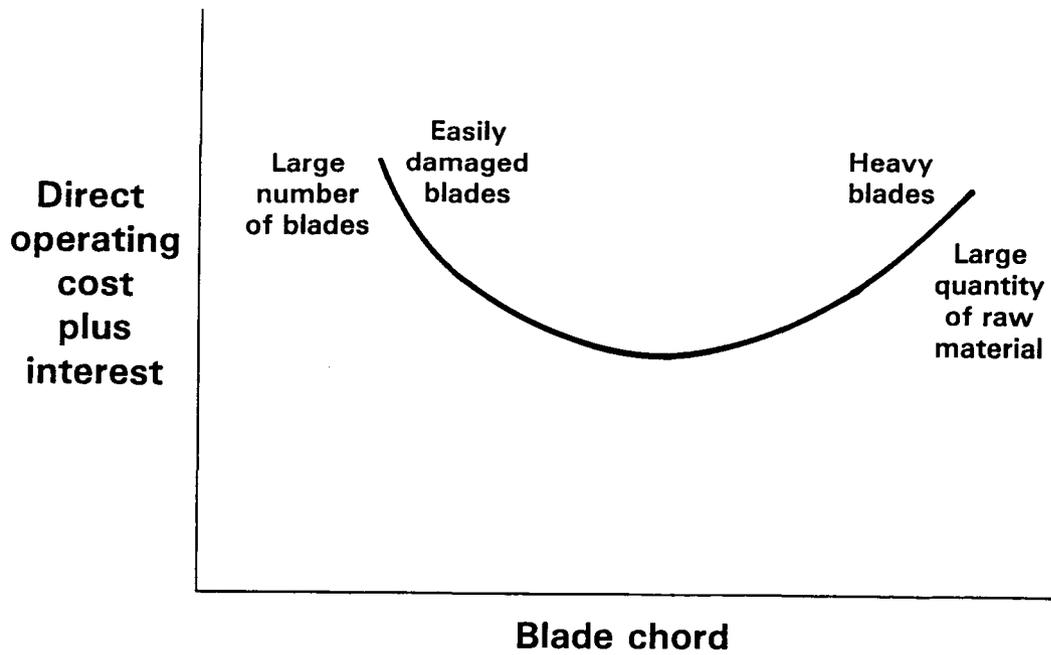


Figure 3.2-3 Blade Chord Optimization Appears to be a Simple Design Problem

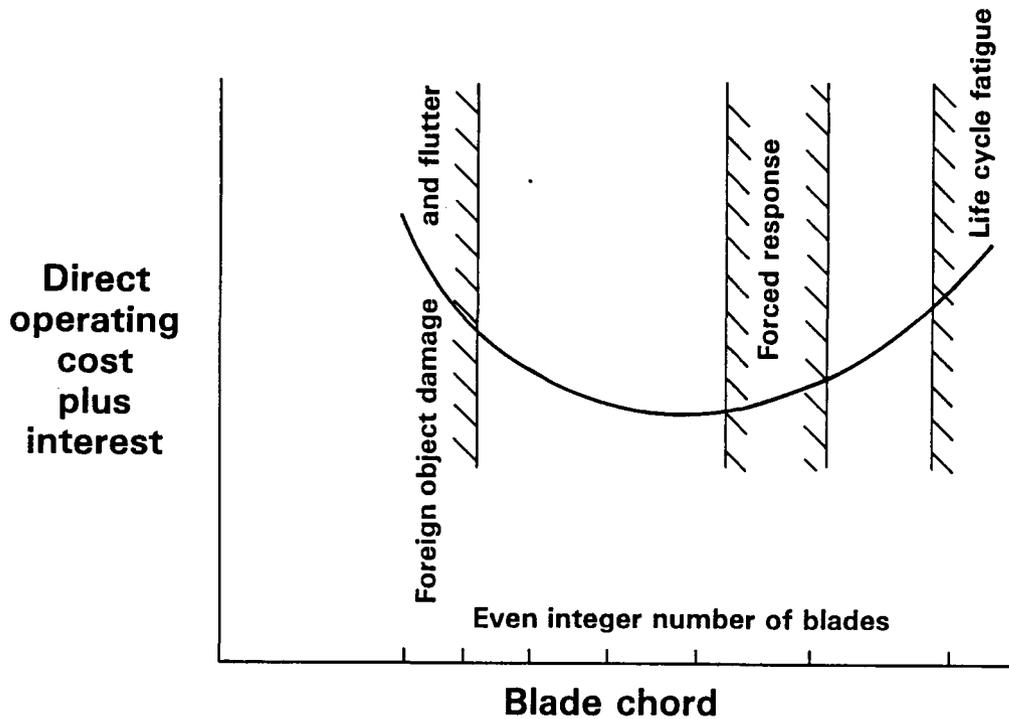


Figure 3.2-4 Design Problem Complexity is Introduced by Structural Constraints

The most effective technique available for solving nonlinear optimization problems was selected from those available. The COPES/CONMIN (Control Program for Engineering Synthesis/CONstrained MINimization) optimization program, a general purpose routine based on the method of feasible directions and developed by G. N. Vanderplaats of the Naval Postgraduate School, was chosen for the optimization procedure.

The above efforts in the STAEBL procedure identify a fine tuned optimum blade design that is validated by NASTRAN refined analysis. The procedure was demonstrated by the design of two composite material shroudless fan blades.

This procedure will reduce human error in the blade design process by automating with mathematical precision what was formerly user judgement in an interactive system.

3.3 INPUT TO FAN BLADE STRUCTURAL TAILORING PROCEDURE

3.3.1 Aerodynamic Stage

The starting point for structural tailoring of an engine blade is a candidate aerodynamic stage design which will deliver the required airflow and pressure ratio. The geometry of this candidate design is input to the structural tailoring procedure in the following form:

- o coordinate definitions of a series of airfoil sections (define stagger, camber, edge radii, chord and thickness, all functions of radius);
- o flowpath boundaries (root and tip radii and convergence angles);
- o number of blades.

3.3.2 Support Structure

The dominant variables which control structural tailoring are frequency dependent and sensitive to blade attachment flexibility. Since the space available for the attachment varies with the airfoil design parameters, attachment flexibility is recognized by increasing the effective length of the candidate aerodynamic blade design. The additional input is:

- o effective inner radius,
- o dimensions of a rectangular section in the extended region.

3.3.3 Operating Conditions

Airfoil peak steady stress is calculated at maximum normal speed to determine life. Fatigue is prevented by tuning to avoid critical resonances at any speed above minimum cruise. Flutter stability and response to injection of a standard bird are calculated at maximum takeoff rotor speed. The inputs required to make these calculations are:

- o rotor speeds,
- o relative flow velocity, Mach number, incidence and density.

3.3.4 Materials

Blade centrifugal stresses and vibratory characteristics result from body loads and are, therefore, fully dependent upon the properties of the blade materials. Blade life is dependent on the strength of the material subjected to a particular stress condition. Composites materials, such as those to be used in the blades tailored in this program, are composed of a fixed proportion of fiber and matrix elements and can be considered to be homogeneous materials with directional properties. Similarly, adhesively bonded plies of metal matrix composite can be considered to be a single material. The net criticality of a local stress state is determined by evaluating a parameter which is a function of the relative criticality of each individual stress component. The inputs which define the required properties for each material are:

- o density,
- o directional moduli and Poisson's ratios,
- o directional cyclic strengths.

3.3.5 Objective Function

The STAEBL procedure optimizes a single benefit which can be related to the final design. The benefit may be as simple as airfoil weight or it may be total value to the engine operator which considers trades between weight, initial cost, maintenance cost and even aerodynamic performance. The benefit expression is kept in generalized form by introducing a FORTRAN definition of:

- o an objective function of design variables or quantities which are defined by the design variables (constant terms are not required).

3.3.6 Constraints

The durability objectives of a blade design are accomplished by imposing limits on the quantities that are calculated in the structural analyses. Margins are established relative to idealized limits to recognize the effects of geometric, material, and operational tolerances and to compensate for approximations in the analyses or underlying assumptions. Inputs to the STAEBL procedure are:

- o minimum allowable predicted aerodynamic damping,
- o minimum allowable difference between predicted frequencies and critical multiples of rotor speed,

- o maximum allowable local and root bird injection stress parameters,
- o limits on design variables (for consistency with various assumptions).

3.3.7 Design Variables

Scaling techniques are provided within the STAEBL procedure to vary the coordinates that define any airfoil section in proportion with changes in chord or maximum thickness (fairing to constant edge radii). Logic is also included to identify the particular material at any point in a composite blade by references to quantities which define the relative position of the limits of that material. A fiber orientation angle is associated with each composite material. Relevant inputs are coded identification of design variables and initial values for starting the iteration and include:

- o root chord (constant scale for all stations),
- o thickness/chord (independent stations),
- o composite material location limits (including the cavity as a zero properties composite),
- o composite material fiber orientation angles.

3.4 APPROXIMATE ANALYSIS

The Pratt & Whitney Aircraft computerized system for designing conventional blade structures consists of a set of analysis programs which are linked to a common data library. Computational efficiency is an important consideration in the selection of these programs because the system is used so frequently. Approximate analyses are used wherever they have been successfully correlated with extensive durability experience. This system provides a proven base for selection of approximate analyses for the STAEBL system.

3.4.1 Stress and Vibration Analysis

3.4.1.1 Approach

The bladed disk vibration analysis program in the existing Pratt & Whitney Aircraft system evaluates a blade with coupled bending and torsional degrees of freedom mounted on a flexible disk and restrained by offset shrouds. The program can operate on blades with two or less shrouds and blade alone analyses can be performed. Resonance diagrams, mode shape plots and input data to the supersonic flutter prediction program are generated.

This beam-type analysis was formulated for application to airfoils with cross section shapes which are highly variable in the spanwise direction. It simulates increments of blade between independently defined airfoil sections. The analysis recognizes the effects of unequal deviation-from-radial of the directions of principal stiffness of the airfoil's leading and trailing edges.

Careful attention to edge stiffness inclination is given because the edge material is located at the greatest distance from the section shear center.

In this system, dynamic influence equations for a section of blade are derived by writing the six degree of freedom compatibility equations and six dynamic equilibrium equations of a rotating vibrating beam blade increment in finite difference form. These increments are then joined by a stacking matrix which rotates the variable vector into the local coordinate system of each successive blade increment, as defined by the line passing through the shear centers of the sections describing the ends of the increment. The shear centers are found by integration of the strength of materials thin section beam formula.

This analysis system was adapted to advanced blade constructions, including composite materials and hollow regions, by introducing new beam blade increment equilibrium and compatibility relationships. With these modifications, a spanwise blade increment is visualized as consisting of a stack of slices, Figure 3.4-1, where the stiffness properties of each slice are derived from lamination theory. Slice load resultants are determined by displacing one end of the beam relative to the other, evaluating slice strain levels, and multiplying by the local slice stiffness. Area integration of the load resultants provides the overall beam stiffness relationship.

The stack of slices analysis concept was verified with a NASTRAN analysis of an assembly of isotropic cantilevered beams of rectangular cross-section as shown in Figure 3.4-2. Figures 3.4-3 through 3.4-5 show the results of several comparisons with a NASTRAN baseline using a plate breakup. When the beam assembly was first analyzed, nodes at each cross-section were constrained using rigid body elements, forcing each plane to deform rigidly as shown in Figure 3.4-3. The results of this analysis were:

1. flapwise bending modes gave good agreement,
2. torsion mode frequencies were too high due to excessive section warping constraints,
3. stiffwise bending modes were too low due to shear flexibilities introduced by the guided cantilevered effect of the parallel beam components.

As shown in Figures 3.4-4 and 3.4-5, the torsion mode problem was alleviated by relaxing the warping rotation constraints along the cross-section. Stiffwise frequencies were improved by refining the spanwise sectional breakup. Figure 3.4-4 shows a no warping constraint condition and individual beam torsional stiffnesses which resulted in a torsional frequency that is too low.

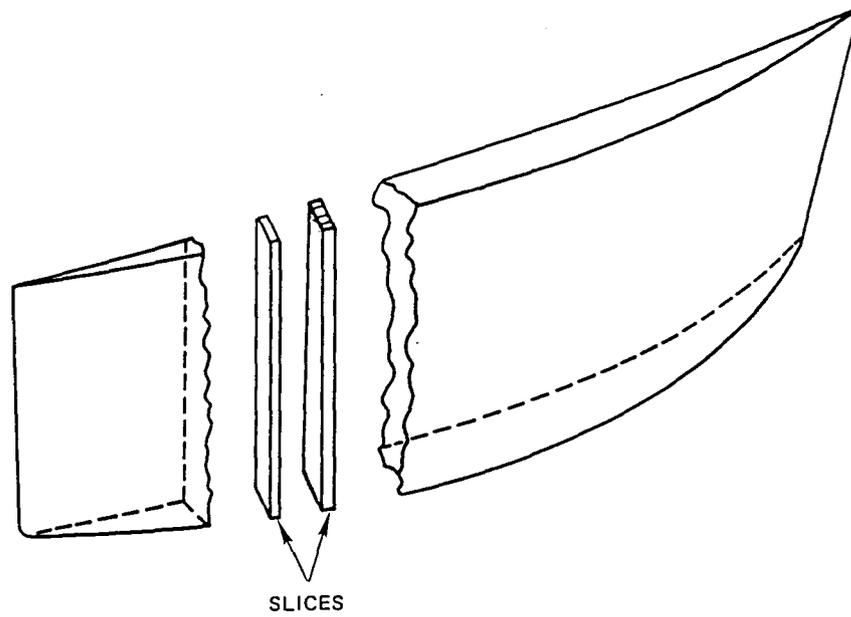


Figure 3.4-1 Model Used in Approximate Analyses of a Spanwise Increment of Blade

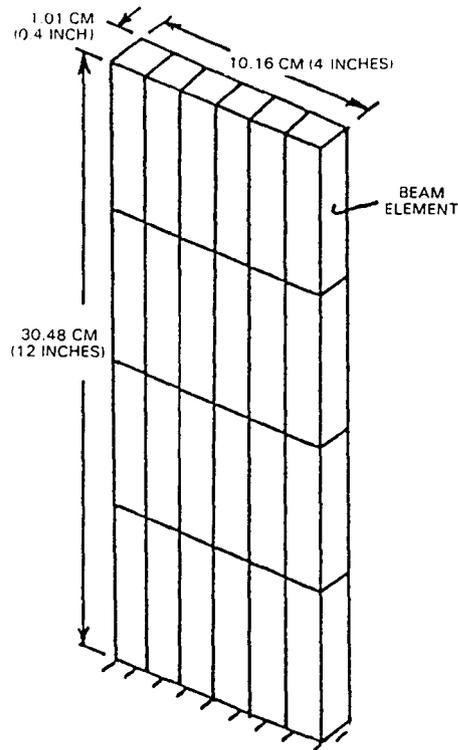


Figure 3.4-2 Beam Model Verification with NASTRAN Vibration Analysis of Assembly of Rectangular, Isotropic, Cantilevered Beams

CYCLES PER SECOND

	PLATE MODEL	BEAM MODEL PLANES REMAIN PLANE
1ST F	90.1	87.6
1ST T	524.1	938.0
2ND F	558.3	535.1
1ST S	821.7	749.4

- WARPING CONSTRAINT MUST BE RELEASED

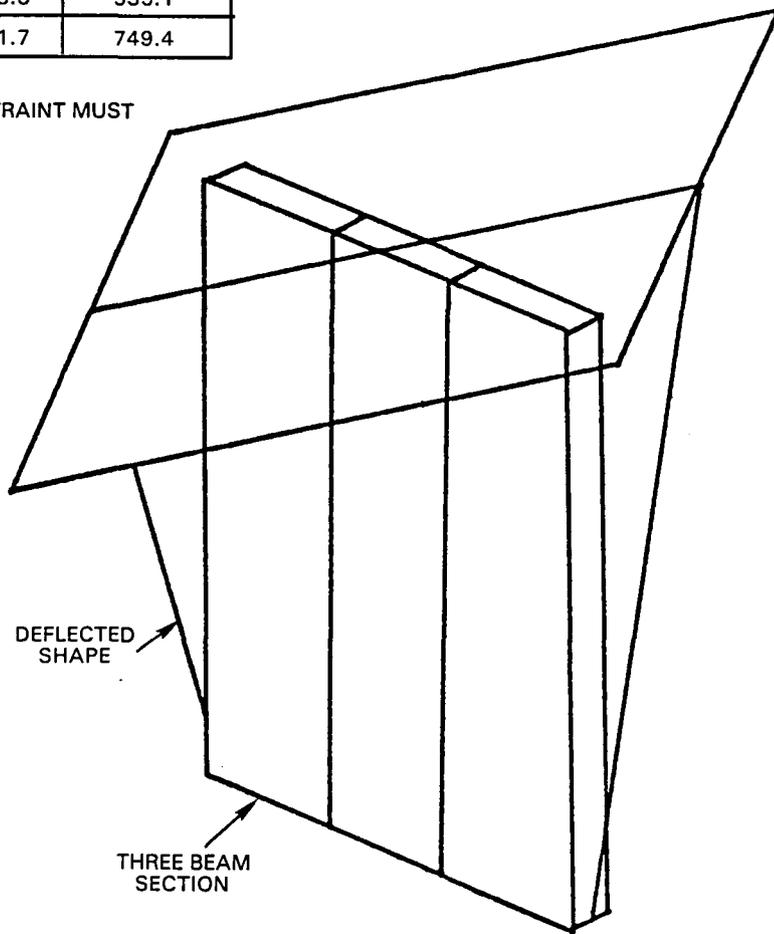


Figure 3.4-3 Torsional Frequencies Obtained with Planes Remain Plane Section Constraint

CYCLES PER SECOND

	PLATE	FREE WARPING
1ST F	90.1	87.6
1ST T	524.1	354.5
2ND F	558.3	535.1
1ST S	821.7	749.4

- THE SUM OF BEAM TORSIONAL STIFFNESS DOES NOT EQUAL SECTION TORSIONAL STIFFNESS

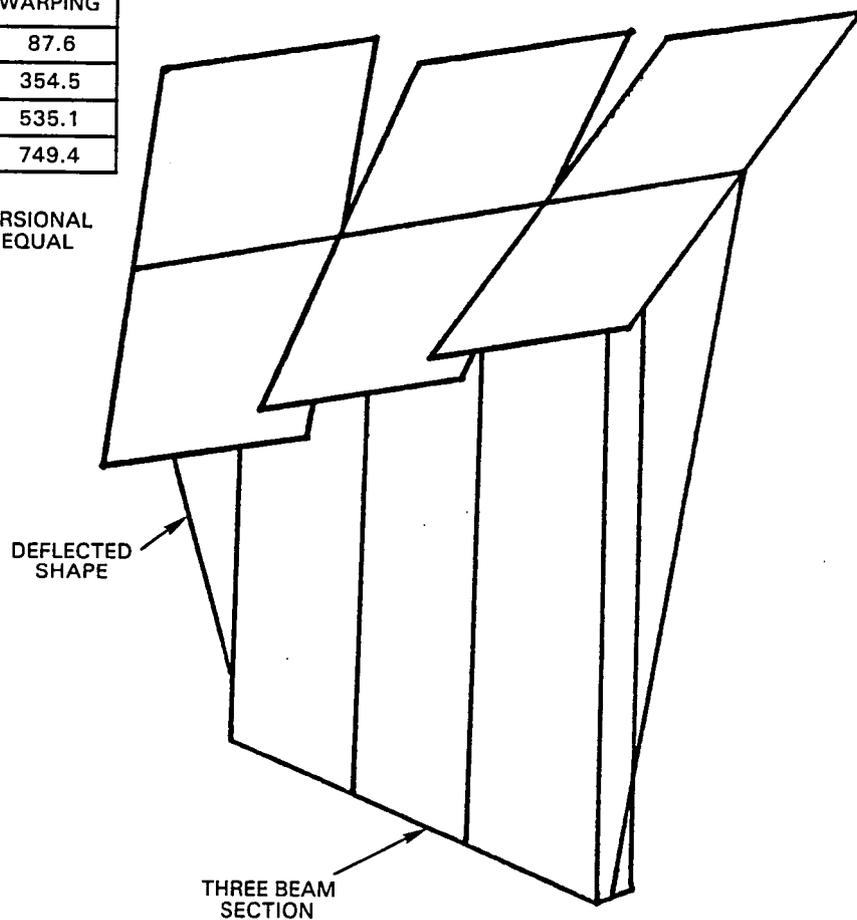


Figure 3.4-4 Low Torsional Frequencies Produced by the Free Warping Condition

- THIN SECTION TORSIONAL STIFFNESS COEFFICIENT PER BEAM, $1/3 BT^3$
- THIN RECTANGULAR SECTION APPROXIMATE WARPING FUNCTION
- THIN RECANGULAR SECTION TORSIONAL STIFFNESS COEFFICIENT

CYCLES PER SECOND		
	PLATE	BEAM
1ST F	90.1	87.6
1ST T	524.1	558.2
2ND F	558.3	535.1
1ST S	821.7	749.4

Figure 3.4-5 Constrained Warping Condition and Thin Section Torsional Stiffness Correlated Well Using an Approximate Section Warping Function

The low torsional stiffnesses were resolved by introducing thin section torsional stiffness coefficients and an approximate warping function. The warping function was evaluated using:

$$u = -y z \left(\frac{d\theta}{dx} \right) \quad (1)$$

This warping function corresponds to the first term of the elasticity solution for the torsion of a rectangular section. The effect on natural frequencies of imposing the warping function on the NASTRAN solution is shown on Figure 3.4-5. As a result of the warping function prescription, each cross-section is now reduced to six degrees of freedom, which is consistent with the existing beam analysis procedure. Since the warping function that was prescribed corresponds rather well with the elasticity solution, little system stiffness was added, and the torsion mode frequency was increased by only 2.5 percent.

The beam assembly procedure was then tested on a plate with a 30 degree pretwist between root and tip. Results of this test case are shown on Table 3.4-I and Figure 3.4-6. All frequencies from the combined beam analysis procedure were found to be within 10 percent of the NASTRAN plate solution.

These test cases verify the concept that a beam section may be assembled from a collection of beam elements with acceptable results for bending and torsion frequencies.

TABLE 3.4-I

NASTRAN TEST CASE FOR PANEL WITH 30 DEGREE PRETWIST
(Cycles per Second)

	NASTRAN Plate Analysis	Connected Beams, Free Warping	Connected Beams, Warping Imposed
1st Flap	90.1	87.6	87.6
2nd Flap	480.1	438.8	438.8
1st Torsion	535.1	567.9	585.5

CYCLES PER SECOND

	PLATE	FREE WARPING	ϕ
1ST F	90.1	87.6	87.6
1 ST T	535.1	567.9	585.5
2ND F	480.1	438.8	438.8
1ST S	941.3	722.8	722.8

- WARPING PRESCRIBED, $\phi = -yz \theta, x$
- TORSIONAL STIFFNESS, $K_T = 1/3 BT^3$
COEFFICIENT

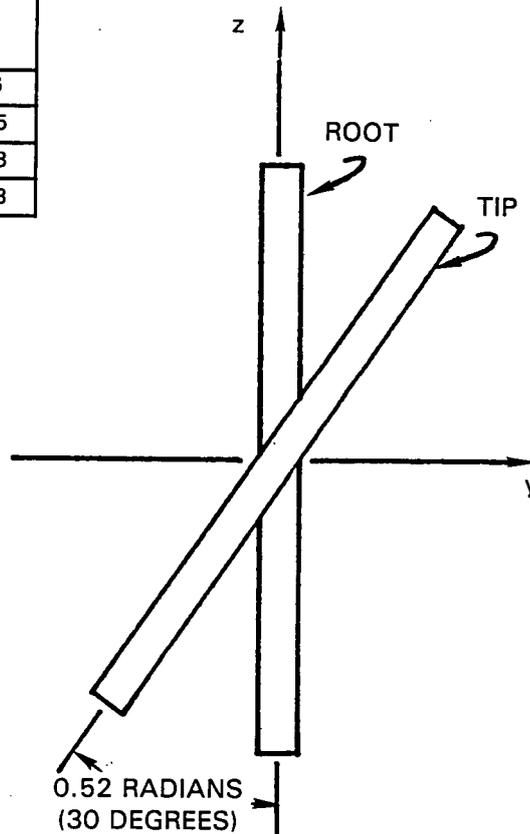


Figure 3.4-6 Results Obtained for a Twisted Flat Plate

3.4.1.2 Solution Procedure

The approximate modelling technique for the design of a composite fan blade uses a laminated composite beam theory. In this theory, the airfoil is divided into radial segments with each radial segment being further divided into a chordwise array of beams as shown in Figure 3.4-7. Each beam has a rectangular cross-section to simplify the composite material characterization. The laminated beam derivation parallels classical laminated plate equations. Differences in stress components and the definition of the stress and moment resultants occur between beam and plate equations.

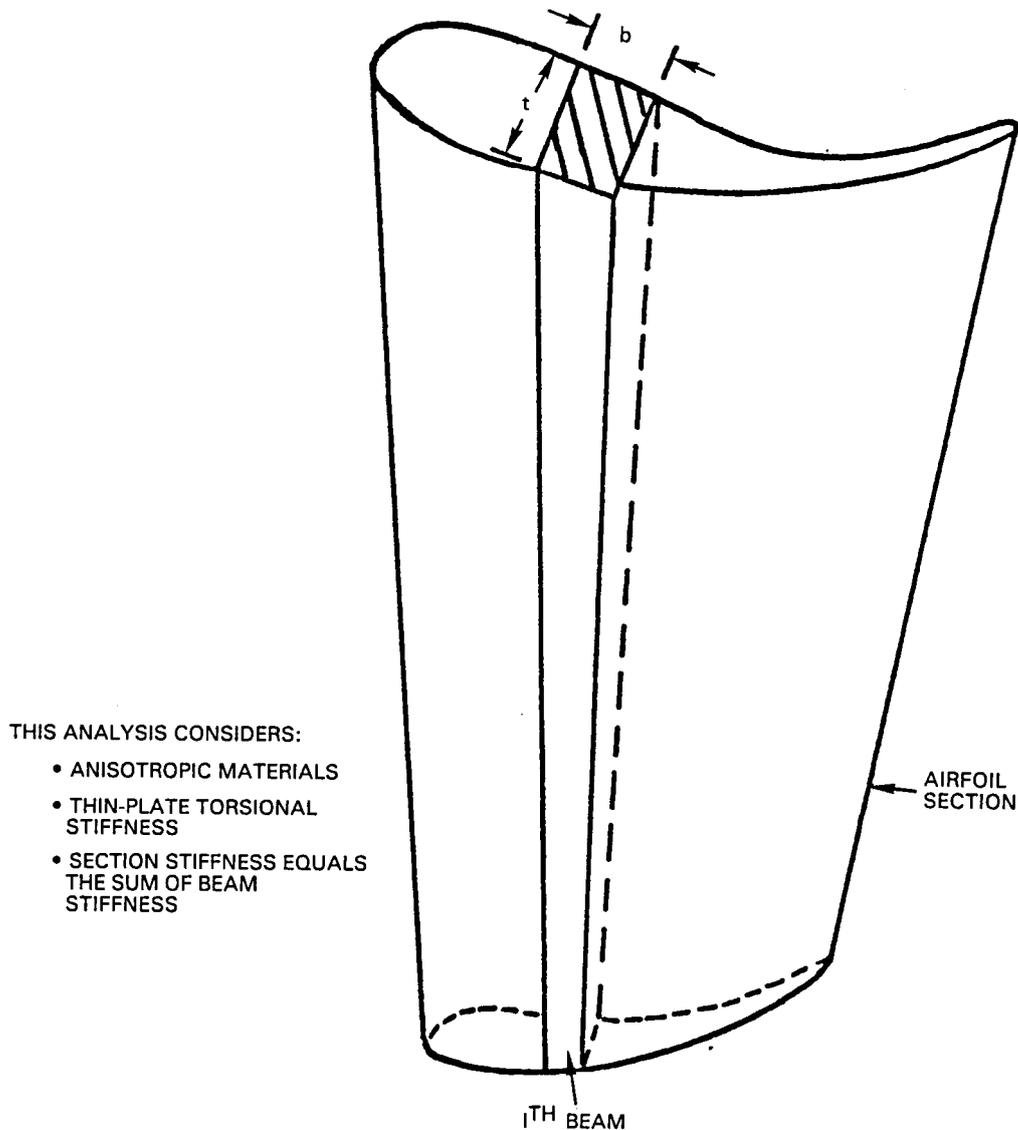


Figure 3.4-7 Radially Stacked Beam Element as Represented by the Laminated Composite Beam Theory

The laminated beam formulation employs a direct stiffness finite element approach. Slice stiffnesses are summed to give the section stiffness. The section stiffness matrix is then transformed into transfer matrix form, so that it may now replace the influence coefficients of the original analysis system.

The laminated beam formulation begins with the assumption of cubic lateral displacement functions and linear membrane and twist displacement functions, as shown in Figure 3.4-8. The 0 subscripts in Figure 3.4-8 refer to neutral axis displacements.

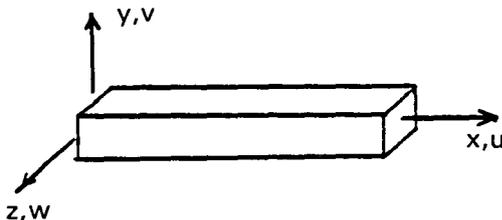
$$\begin{aligned}
 V_0 &= a_1 + a_2X + a_3X^2 + a_4X^3 \\
 W_0 &= a_5 + a_6X + a_7X^2 + a_8X^3 \\
 U_0 &= a_9 + a_{10}X \\
 \theta_0 &= a_{11} + a_{12}X
 \end{aligned}$$


Figure 3.4-8 Shape Functions for the Element Displacement Response

The coefficients of the displacement functions, a , are solved for in terms of nodal point displacements, giving:

$$\{a\} = [\bar{N}] \{\Delta\} \quad (2)$$

The element shape functions, $[N]$, may, therefore, be readily evaluated.

Strains may be evaluated from shape function derivatives: second derivatives in bending, first derivatives in membrane and twist. Thus, the neutral axis strains and curvatures may be related to the shape coefficients, as

$$\{\xi_0\} = [X] \{a\} \quad (3)$$

Away from the neutral axis, with the assumption that planes remain plane in bending, displacements become:

$$\begin{aligned}
 u(x,y,z) &= u_0(x) - y v_{0,x}(x) - z w_{0,x}(x) + C yz \theta_{,x} \\
 v(x,y,z) &= v_0(x) - c_1 y u_{0,x}(x) - xz \theta_{,x} \\
 w(x,y,z) &= w_0(x) - c_2 z u_{0,x}(x) - xy \theta_{,x}
 \end{aligned} \quad (4)$$

where c is a warping function coefficient, and c_1 and c_2 are Poisson ratios. Differentiation of these displacements gives the necessary strain relationships, which may now be related to nodal point displacements.

For the k'th lamina, the rotated stress-strain relation is

$$\begin{Bmatrix} \sigma_x \\ \sigma_{xy} \\ \sigma_{xz} \end{Bmatrix}_K = \begin{bmatrix} Q_{11} & 0 & Q_{13} \\ 0 & Q_{22} & 0 \\ Q_{13} & 0 & Q_{33} \end{bmatrix}_K \begin{Bmatrix} \epsilon_x \\ \gamma_{xy} \\ \gamma_{xz} \end{Bmatrix}_K \quad (5)$$

Integration of the stresses over the laminate gives the beam stress resultants,

$$\begin{Bmatrix} N_x \\ N_{xy} \\ N_{xz} \end{Bmatrix} = \sum_{K=1}^N \int_{A_K} \begin{Bmatrix} \sigma_x \\ \sigma_{xy} \\ \sigma_{xz} \end{Bmatrix}_K dA_K \quad (6)$$

$$\begin{Bmatrix} M_x \\ M_y \\ M_z \end{Bmatrix} = \sum_{K=1}^N \int_{A_K} \begin{Bmatrix} y \sigma_{xz} - z \sigma_{xy} \\ -y \sigma_x \\ -z \sigma_x \end{Bmatrix}_K dA_K$$

where N is the total number of lamina in the laminate.

Substituting the lamina stress-strain law and the strain-displacement relations into the above, and integrating over the rectangular cross-section gives:

$$\begin{Bmatrix} N_x \\ M_x \\ M_y \\ M_z \end{Bmatrix} = \begin{bmatrix} A_{11} & B_{13} & -B_{11} & 0 \\ B_{13} & D_{33} & -D_{13} & 0 \\ -B_{11} & -D_{13} & D_{11} & 0 \\ 0 & 0 & 0 & a^2/12 A_{11} \end{bmatrix} \begin{Bmatrix} U_{0,x} \\ \theta,x \\ K_y \\ K_z \end{Bmatrix} \quad (7)$$

where:

$$\bar{D}_{33} = 2 D_{33} + \frac{b^2}{6} A_{22}$$

$$A_{ij} = a \sum_{K=1}^N Q_{ijK} (Y_K - Y_{K-1})$$

$$B_{ij} = a/2 \sum_{K=1}^N Q_{ijK} (Y_K^2 - Y_{K-1}^2)$$

and

(8)

$$D_{ij} = a/3 \sum_{K=1}^N Q_{ijK} (Y_K^3 - Y_{K-1}^3)$$

with the rectangle dimensions and layup as illustrated in Figure 3.4-9.

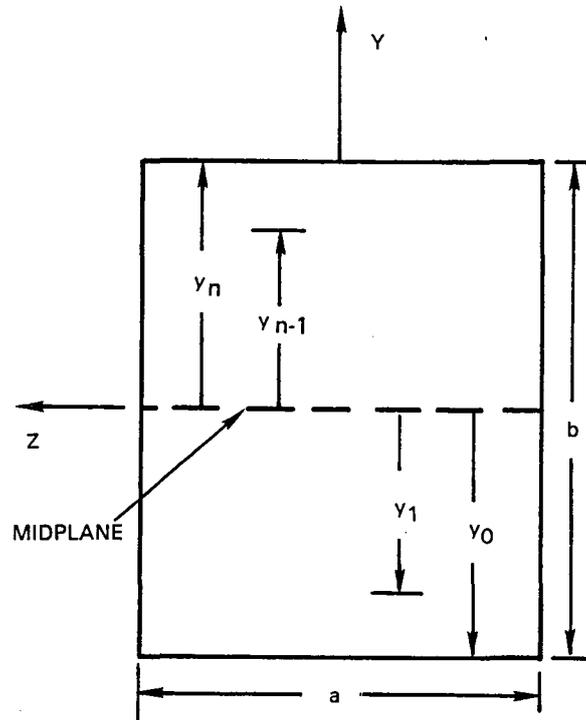


Figure 3.4-9 Rectangular Beam Ply Layup

The element force resultants at the beam ends may now be expressed in terms of the neutral axis strains as:

$$\{f\} = [E] \{\epsilon_0\} . \quad (9)$$

The beam forces may be transferred into the finite element coordinate system, using an equilibrium matrix, giving:

$$\{F\} = [A] \{f\} . \quad (10)$$

The previous relations may now be combined to yield the element stiffness relation:

$$\{F\} = [A] [E] [X] [\bar{N}] \{\Delta\} = [K_B] \{\Delta\} \quad (11)$$

The section stiffness may be generated as the summation of the individual stacked beam stiffnesses, when offsets and cross-section warping are accounted for through a rigid-body linkage:

$$\{F_s\} = \sum_{i=1}^M \left([R_1]_i [K_B]_i [R]_i \right) \{\Delta_s\} = [S_K] \{\Delta_s\} \quad (12)$$

where there are M stacked beams on the section. At the blade root, the warping function is set to zero, to impose warping restraint effects on the blade model.

Once the section stiffness has been generated, to make the new element compatible in form with the existing beam blade analysis, the section stiffness must be transformed to transfer matrix form. Partitioning the element equilibrium equation for ends 1 and 2 gives:

$$\begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} = \begin{bmatrix} S_{K11} & | & S_{K12} \\ \hline S_{K12} & | & S_{K22} \end{bmatrix} \begin{Bmatrix} \Delta_1 \\ \Delta_2 \end{Bmatrix} \quad (13)$$

Reordered to transfer matrix form, the element equation becomes:

$$\begin{Bmatrix} \Delta_2 \\ F_2 \end{Bmatrix} = \begin{bmatrix} -S_{K12}^{-1} S_{K11} & | & S_{K12}^{-1} \\ \hline S_{K21} - S_{K22} S_{K12}^{-1} S_{K11} & | & S_{K22} S_{K12}^{-1} \end{bmatrix} \begin{Bmatrix} \Delta_1 \\ F_1 \end{Bmatrix} = \begin{bmatrix} \bar{S}_K \\ \bar{F}_1 \end{bmatrix} \begin{Bmatrix} \Delta_1 \\ F_1 \end{Bmatrix} \quad (14)$$

The section masses are treated as lumped mass points at the two end node locations. In blade vibration, equilibrium across a concentrated mass point, gives:

$$F_2' = F_2 + m\ddot{U}_2 = F_2 - w^2_m U_2. \quad (15)$$

In matrix form, the mass effect may be expressed as:

$$\begin{Bmatrix} U \\ F \end{Bmatrix}_{2'} = \begin{bmatrix} 1 & 0 \\ -w^2_m & 1 \end{bmatrix} \begin{Bmatrix} U \\ F \end{Bmatrix}_2 = \begin{bmatrix} - \\ [m] \end{bmatrix} \begin{Bmatrix} U \\ F \end{Bmatrix}_2 \quad (16)$$

Hence, in traversing from the beginning of a beam segment to its end, with inertia effects included, it is found:

$$\begin{Bmatrix} \Delta z \\ F_2 \end{Bmatrix}' = \begin{bmatrix} \bar{m} \\ \bar{K} \end{bmatrix} \begin{Bmatrix} \Delta 1 \\ F_1 \end{Bmatrix} \quad (17)$$

The (\bar{m}) (\bar{K}) product above is the final step in making the present theory compatible with the existing beam analysis. Notably, only the (\bar{m}) array has terms which are frequency dependent. Hence, this procedure can be made very efficient for frequency extraction.

For a rotating blade, centrifugal accelerations tend to stiffen the blade, and must be included in the calculation for vibration frequencies. In the present analysis, centrifugal effects are included through a section differential stiffness matrix. The section differential stiffness matrix employed, found in Reference 1, utilizes the element load resultants and geometry to calculate the stiffness increment.

To determine the static load resultants, and also for the calculation of static stresses, a static analysis is performed.

A lumped mass representation is employed in the static analysis as done previously for vibration analysis. At station $i+1$, a static load, P_{i+1} , results from the centrifugal accelerations of the point mass. Thus, with mass effects included:

$$\begin{Bmatrix} U_{i+1} \\ F_{i+1} \end{Bmatrix}' = [K_k] \begin{Bmatrix} U_i \\ F_i \end{Bmatrix} + \begin{Bmatrix} 0 \\ P_{i+1} \end{Bmatrix} \quad (18)$$

For the full blade, from station 1 to station n ,

$$\begin{Bmatrix} U_n \\ F_n \end{Bmatrix}' = \begin{bmatrix} K_{11} & K_{12} \\ K_{12} & K_{22} \end{bmatrix} \begin{Bmatrix} U_1 \\ F_1 \end{Bmatrix} + \begin{Bmatrix} 0 \\ P_T \end{Bmatrix} \quad (19)$$

At station 1, the blade root boundary condition,

$$U_1 = C F_1, \quad (20)$$

allows the root to be spring supported or cantilevered. The final equation system thus becomes:

$$\begin{Bmatrix} U_n \\ F_n \end{Bmatrix}' = \begin{bmatrix} C K_{11} + K_{12} \\ C K_{21} + K_{22} \end{bmatrix} \begin{Bmatrix} F_1 \end{Bmatrix} + \begin{Bmatrix} P_{T1} \\ P_{T2} \end{Bmatrix} \quad (21)$$

At station n , the blade tip boundary condition:

$$\begin{Bmatrix} F_n \end{Bmatrix}' = 0. \quad (22)$$

The root load can then be determined,

$$\{F_1\} = -[C K_{21} + K_{22}]^{-1} \{P_{T2}\} \quad (23)$$

Now, knowing the root loads and deflections, section loads and deflections are calculated according to the recursion relationship of (18).

The components of the centrifugal force acting on a vibrating blade contain terms which are proportional to the displacement. These displacement dependent terms form the "centrifugal mass matrix" of Reference 2. The "centrifugal mass matrix" for the present analysis has been generated at the individual beam level. The application of appropriate planar constraints enables the reduction of the centrifugal mass to the six degree-of-freedom section level. This capability has also been included in the analysis.

3.4.1.3 Verification

Due to intentional similarities between the present stacked beam analysis and the NASTRAN Bar element, analysis verification was greatly simplified.

The approximate static analysis procedure was verified through comparisons with NASTRAN test cases for a flat plate and for a plate with a 30 degree twist. The model consisted of 8 cross-sections, with 11 beams per section. Table 3.4-II summarizes the results of the two test cases, showing nearly exact agreement with NASTRAN for both deflections and reaction loads.

The differential stiffness and centrifugal mass capabilities of the approximate analysis were demonstrated by comparing the results of two test cases with the results of NASTRAN analyses. In the tailoring operation, these restoration effects would only be applied to the vibration analysis but the verification was obtained from iterated static analyses.

The first case, shown in Figure 3.4-10, consisted of a rotating, tilted, flat plate. Table 3.4-III shows almost exact comparisons between the present analysis and a corresponding NASTRAN beam analysis for the local tip deflections.

The second test case, shown on Figure 3.4-11, consisted of a twisted flat plate stacked along a radial line, with a 30 degree twist from root to tip. As shown on Table 3.4-IV, almost exact agreement exists between the present analysis and a corresponding NASTRAN beam analysis.

The vibration analysis capability of the stacked beam solution was verified by comparing predicted natural frequencies with NASTRAN predicted natural frequencies. Table 3.4-V shows excellent agreement between the approximate analysis procedure and a NASTRAN plate model for analysis of a flat plate 30.5cm x 10.2cm x 1.02cm (12in x 4in x 0.4in).

As shown in Table 3.4-VI, when the plate was given a 30 degree twist between root and tip, bending frequency agreement remained good between the approximate analysis and NASTRAN. The first torsion mode frequencies differed by less than 5 percent.

To examine a cross-section more representative of an airfoil, a doubly wedged plate was analyzed. For this model, the edge thicknesses were taken as one quarter of the mid-chord thickness. Good agreement with NASTRAN was obtained, as seen on Table 3.4-VII.

TABLE 3.4-II

COMPARISON OF STATIC ANALYSIS PROCEDURE WITH NASTRAN TEST CASES

	<u>NASTRAN</u>	<u>Static Analysis</u>
<u>Flat Plate (11 Beams/Section-8 Sections)</u>		
<u>Tip Deflections</u>		
Radial, cm (in)	0.4171-3 (0.1642-3)	0.4168-3 (0.1641-3)
Twist, (radians)	0	0
<u>Restraint Forces</u>		
Radial, n (lb)	-2327. (-0.5232+3)	-2327. (-0.5232+3)
Twist, n-m (in-lb)	0	0
<u>Twisted Plate (11x8 Breakup, 30° twist)</u>		
<u>Tip Deflections</u>		
Radial, cm (in)	0.4244-3 (0.1671-3)	0.4244-3 (0.1671-3)
Twist, (radians)	0.3828-4	0.3827-4
<u>Restraint Forces</u>		
Radial, n (lb)	-2330. (-0.5239+3)	-2330. (-0.5238+3)
Twist, n-m (in-lb)	3.11 (0.2754+2)	3.11 (0.2753+2)

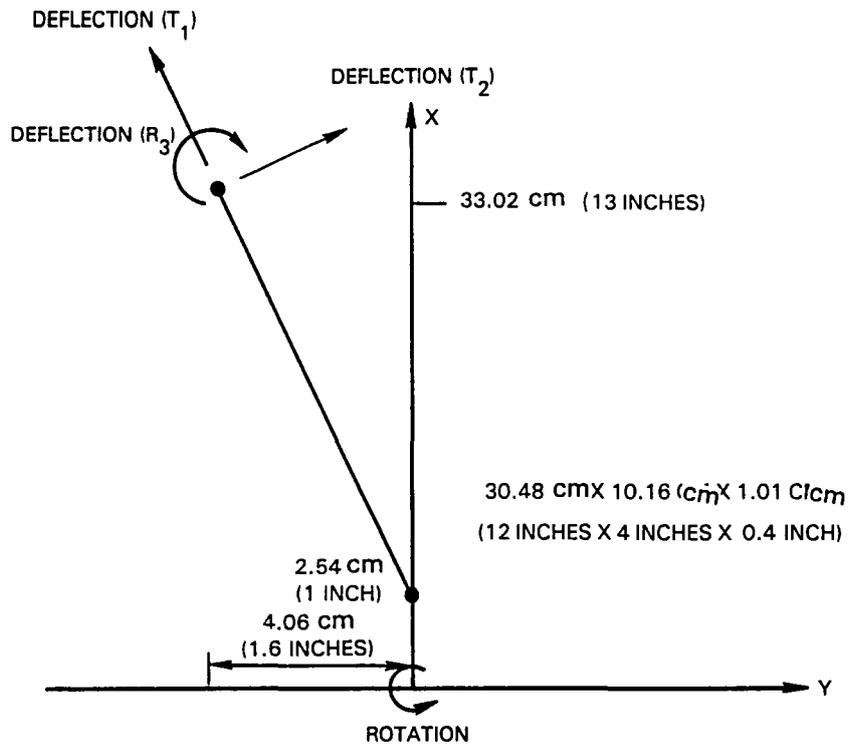


Figure 3.4-10 Tilted Flat Plate Used in Approximate Analysis Test Case

TABLE 3.4-III

TILTED FLAT PLATE TIP DEFLECTIONS
(10,000 rpm, 30.5cm x 10.2cm x 1.02cm)
(12in x 4in x 0.4in)

	Local Tip Deflections		
	T_1 , cm (in)	T_2 , cm (in)	R_3 , radians
<u>Static Analysis</u>			
NASTRAN	0.04881 (0.019217)	1.9461 (0.76617)	0.08524
STAEBL	0.04890 (0.019251)	1.9463 (0.76627)	0.08526
<u>Iterated Static Analysis</u>			
NASTRAN	0.04870 (0.019175)	0.88321 (0.34772)	0.03474
STAEBL	0.04875 (0.019192)	0.88326 (0.34774)	0.34474

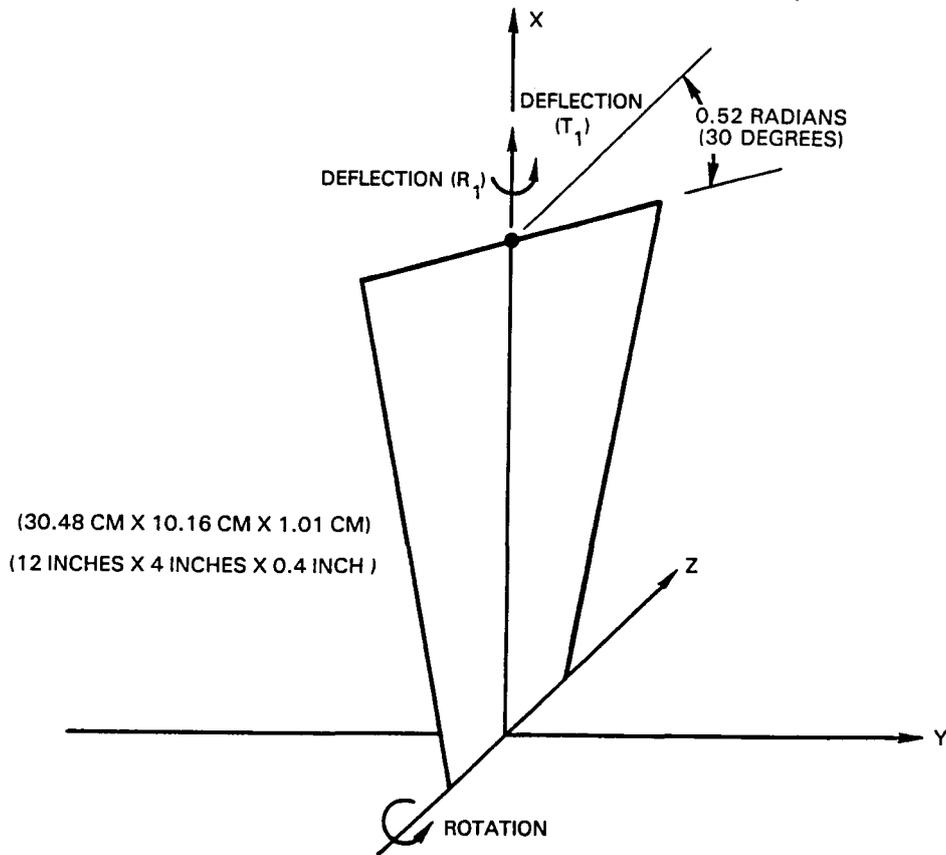


Figure 3.4-11 Twisted Flat Plate Used in Approximate Analysis Test Case

TABLE 3.4-IV

TWISTED (30 Degrees) FLAT PLATE TIP DEFLECTIONS
(1,000 rpm, 30.5cm x 10.2cm x 1.02cm)
(12in x 4in x 0.4in)

	Local Tip Deflections	
	T_1 , cm (in) $\times 10^{-3}$	R_1 (radians) $\times 10^{-4}$
<u>Static Analysis</u>		
NASTRAN	0.4246 (0.16715)	0.38276
STAEBL	0.4245 (0.16711)	0.38272
<u>Iterated Static Analysis</u>		
NASTRAN	0.4246 (0.16715)	0.38261
STAEBL	0.4245 (0.16711)	0.38264

TABLE 3.4-V

APPROXIMATE ANALYSIS NATURAL FREQUENCIES COMPARED WITH NASTRAN
(Flat Plate)

<u>Mode</u>	<u>Cycles per second</u>		<u>Percent Difference</u>
	<u>NASTRAN</u>	<u>STAEBL</u>	
First flap	90	88.5	-1.8
First torsion	524	529.7	+1.0
Second flap	558	552.3	-1.1
First stiff	822	822.5	+0.1

TABLE 3.4-VI

APPROXIMATE ANALYSIS NATURAL FREQUENCIES COMPARED WITH NASTRAN
(30 Degree Twisted Plate)

<u>Mode</u>	<u>Cycles per second</u>		<u>Percent Difference</u>
	<u>NASTRAN</u>	<u>STAEBL</u>	
First flap	90	88.5	-1.8
Second flap	480	474.0	-1.3
First torsion	536	561.5	+4.8
First stiff	941	945.0	+0.4

TABLE 3.4-VII

APPROXIMATE ANALYSIS NATURAL FREQUENCIES COMPARED WITH NASTRAN
(Double-Wedge Plate)

<u>Mode</u>	<u>Cycles per second</u>		<u>Percent Difference</u>
	<u>NASTRAN</u>	<u>STAEBL</u>	
First flap	65.6	65.0	+0.9
Second flap	404.8	405.5	+0.2
First torsion	442.5	460.0	+4.0
First stiff	706.4	705.5	+0.1

Further calibration of the analysis system was obtained by comparing the approximate analysis procedure with NASTRAN for frequency predictions of a rotating fan blade. The comparison, shown in Table 3.4-VIII, gave agreement consistent with the anticipated accuracy of an approximate analysis.

TABLE 3.4-VIII
APPROXIMATE PROCEDURE ANALYSIS OF FAN BLADE NATURAL
FREQUENCY COMPARED WITH NASTRAN

Mode	Cycles per second		Percent Difference
	NASTRAN	STAEBL	
First flap	92.9	93.6	+0.8
First torsion	185.3	163.2	-11.9
Second flap	210.2	205.7	-2.1
First stiff	370.9	349.0	-5.9

When stress distribution predictions were compared between the approximate analysis and previous beam analysis, poor correlations were noted. The problem was identified to be an assumption that warping faired uniformly from zero at the airfoil root to free-beam, as defined by the equations presented in Figure 3.4-12, at the end of the first increment. This assumption failed to recognize the fact that high root stress is a local result of restrained warping. A modification changed the torsional stiffness to recognize that:

$$\frac{\text{twist gradient}}{\text{twist moment}} \propto 1 - e^{-x/A} \quad (24)$$

x = distance from root

A = characteristic length associated with warping restraint

The resulting root centrifugal stress distribution is shown in Figure 3.4-13. Agreement with NASTRAN plate element blade analysis is as good as can be expected of a beam analysis.

An additional processing step was added to the approximate static and vibratory stress calculations in anticipation of the use of composite material in some layers of a tailored blade. A single quantity incorporating all stress components is evaluated. It is $F_1 \sigma_1 + F_2 \sigma_2 + F_6 \sigma_6 + F_{11} \sigma_1^2 + F_{22} \sigma_2^2 + F_{66} \sigma_6^2 + 2 F_{12} \sigma_1 \sigma_2$ where σ_1 and σ_2 are the principal tensile (or compressive) stresses and σ_6 is the shear stress. F_i and F_{ij} are input directional materials properties which are defined to yield an evaluation of the Tsai-Wu failure criteria for a composite material (or octahedral shear stress to strength ratio squared in metal).

The accuracy of the natural frequency prediction in the STAEBL procedure was subjected to an additional check by applying it to the analysis of a hollow blade. The same blade geometry, shown in Figure 3.4-14, was analyzed using NASTRAN plate elements with properties defined by lamination theory. Bending frequencies were in agreement but the error in the torsional frequency prediction was excessive. A similar result was obtained from analysis of a hollow rectangular section beam. It was found that the torsional stiffness equation could be redefined to be in agreement with the formula for a hollow rectangular section beam without changing the successful results that had previously been obtained for a solid airfoil.

The revised stiffness expression, for the thin walled airfoil pictured in Figure 3.4-15 and 3.4-16, is:

$$K = 4G \int_0^b \int_{-t/2}^{-t/2+\delta} y^2 dy dz + 4G \int_0^b \int_{t/2-\delta}^{t/2} y^2 dy dz \quad (25)$$

which agrees with the hollow beam stiffness when thickness, t , approaches a constant. The revised stiffness expression also solved the airfoil problem as is illustrated by the good agreement presented in Table 3.4-IX.

• RADIAL DEFLECTION, u

$$u = (\bar{W}_s - W_s) \theta$$

$$\bar{W}_s = \frac{1}{\text{AREA}} \int_0^S t W_s ds$$

$$W_s = \int_0^S R_T ds$$

$$\theta = \text{TWIST GRADIENT}$$

• ROTATIONS, R_2, R_3

$$R_2 = -R_N \theta$$

$$R_3 = -R_T \theta$$

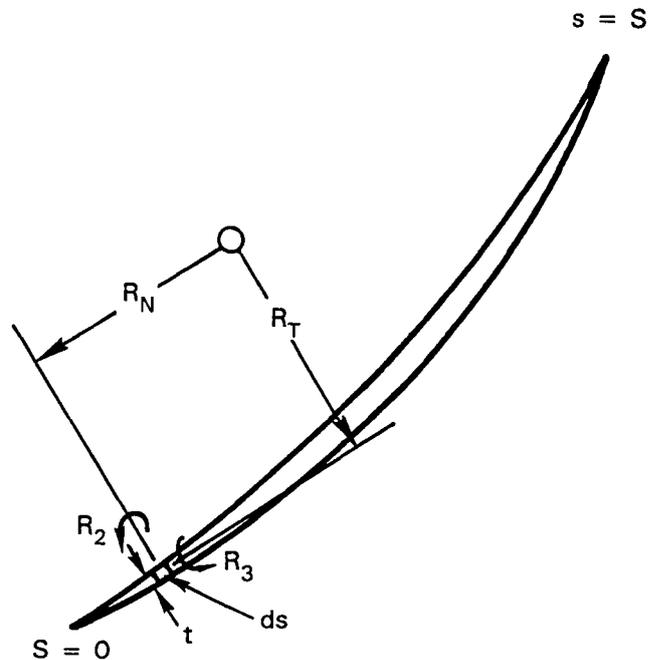


Figure 3.4-12 Airfoil Warping Defined by Spanwise Twist Gradient

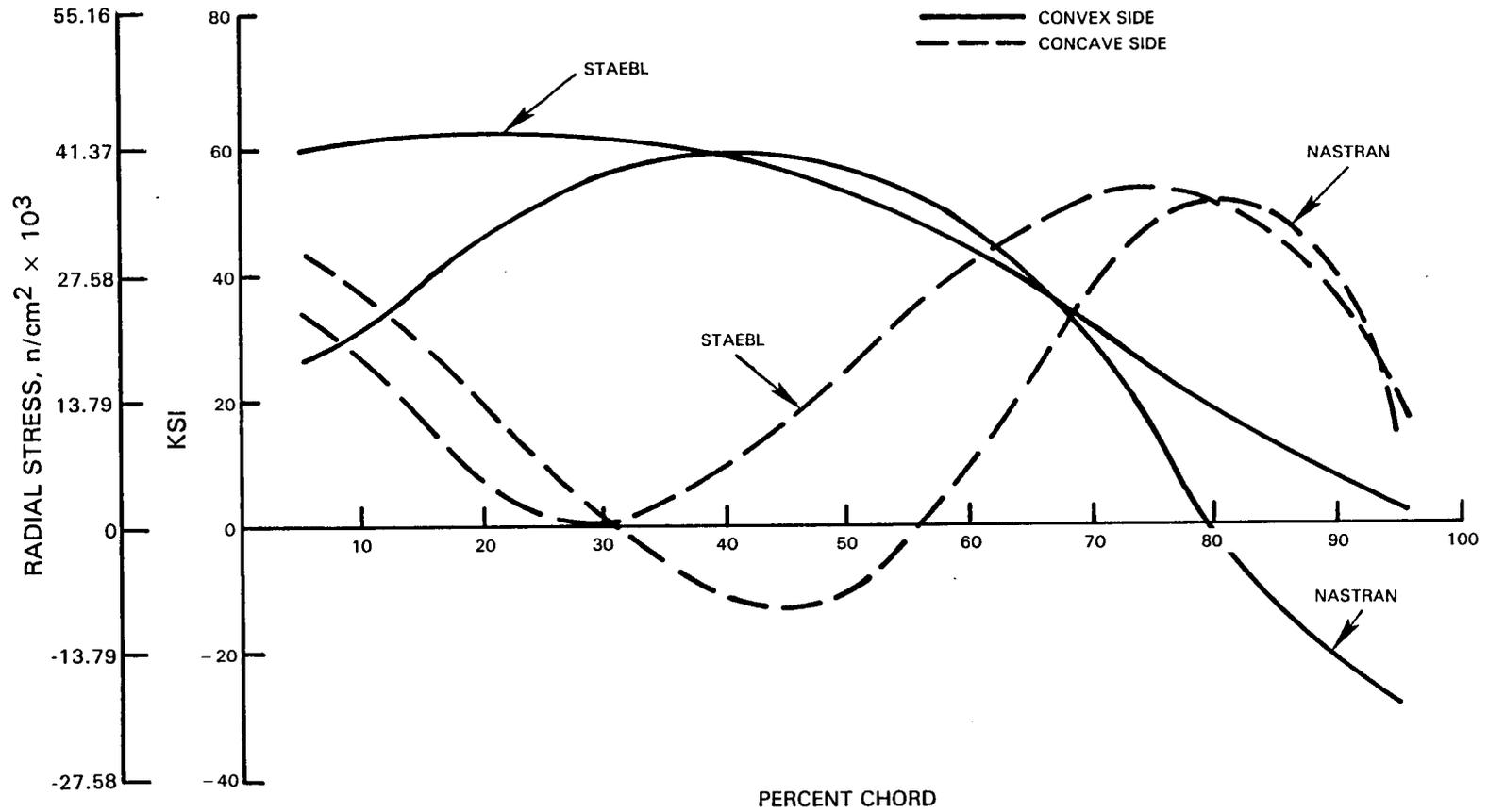


Figure 3.4-13 Approximate Root Centrifugal Stress in the Energy Efficient Engine Fan Blade Compared with NASTRAN

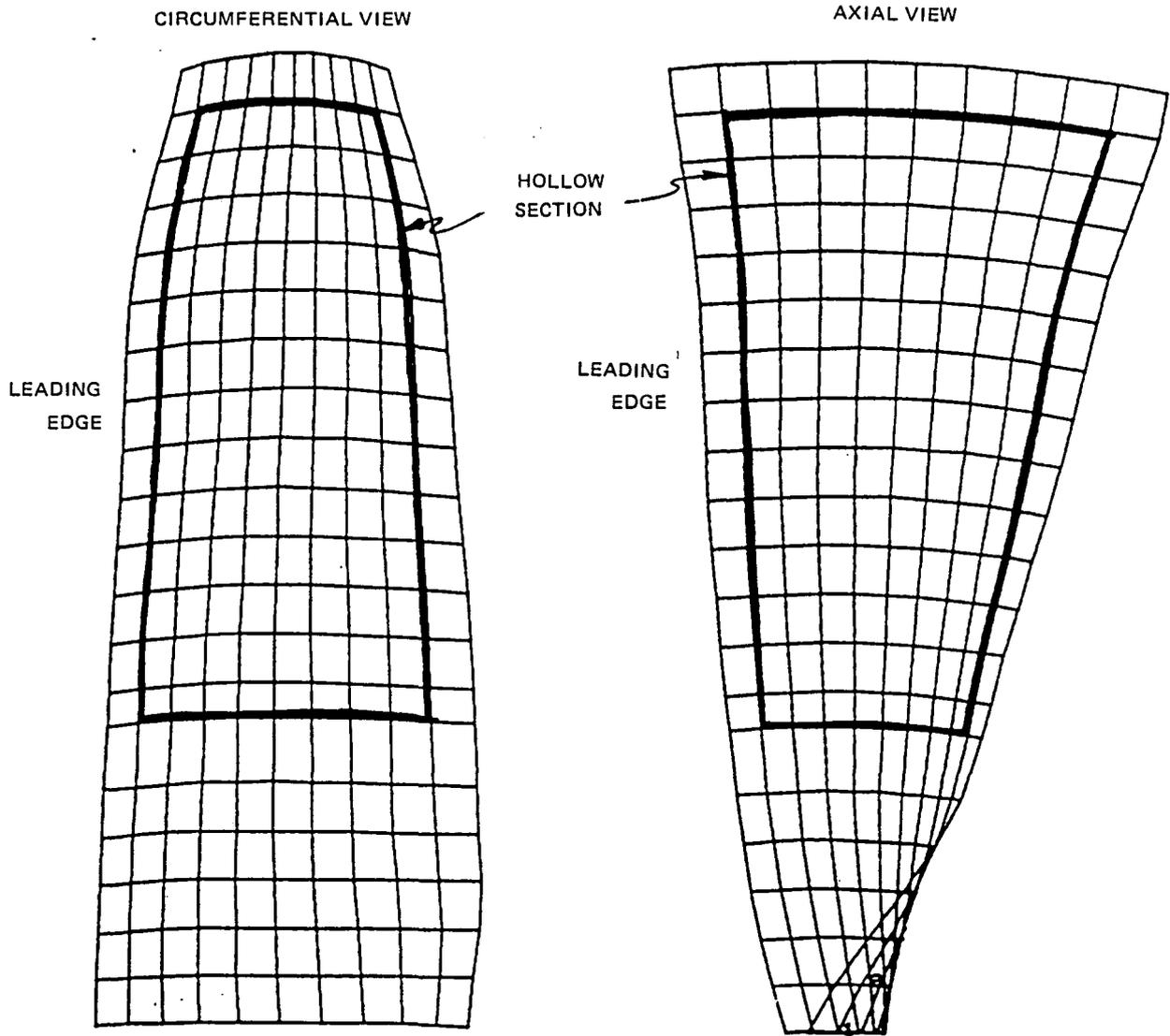


Figure 3.4-14 NASTRAN Hollow Blade Model

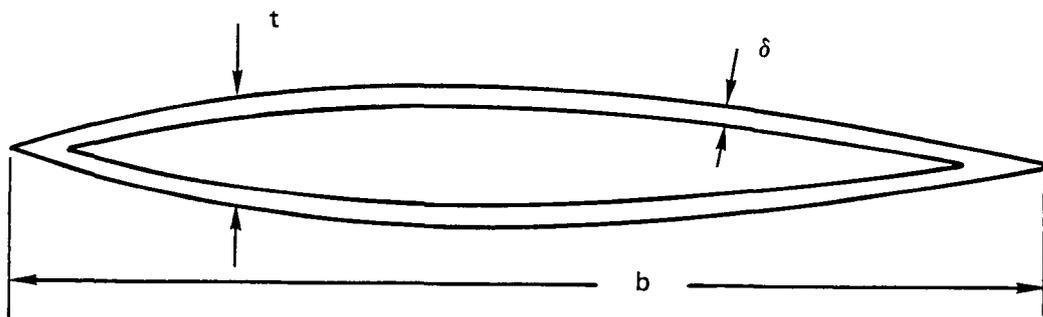


Figure 3.4-15 Hollow Airfoil Beam

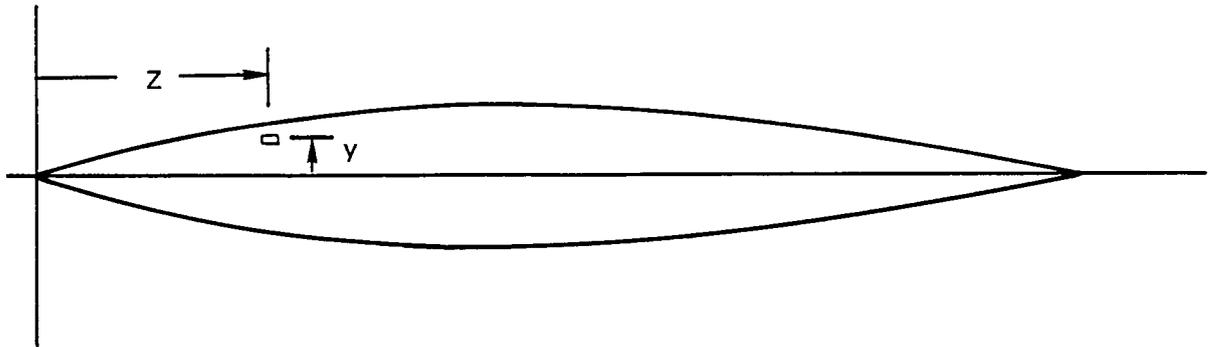


Figure 3.4-16 Laminated Airfoil Beam

TABLE 3.4-IX
HOLLOW BLADE FREQUENCY COMPARISON

	<u>NASTRAN (cps)</u>	<u>Approximate Analysis (cps)</u>	<u>Percent Difference</u>
First Frequency	110.7	113.5	+2.5
Second Frequency	267.4	278.8	+4.3
First Torsion	289.8	295.8	+2.1

A final check of the approximate analyses was accomplished by conducting comparative NASTRAN and approximate airfoil frequency analyses with composite material layers added. The results are presented in Tables 3.4-X and 3.4-XI. The trends are correct and the agreement is within the limits that can be expected of approximate analysis.

TABLE 3.4-X
COMPOSITE REINFORCED BLADE FREQUENCY COMPARISON
(0° Fiber Orientation)

<u>Mode</u>	<u>NASTRAN (cps)</u>	<u>Approximate Analysis (cps)</u>	<u>Percent Difference</u>
First Bending	108.4	111.8	3.1
Second Bending	261.8	276.2	5.5
First Torsion	298.8	312.9	4.7

TABLE 3.4-XI

COMPOSITE REINFORCED BLADE FREQUENCY COMPARISON
(35° Fiber Orientation)

<u>Mode</u>	<u>NASTRAN (cps)</u>	<u>Approximate Analysis (cps)</u>	<u>Percent Difference</u>
First Bending	108.4	111.1	2.5
Second Bending	259.6	270.0	4.0
First Torsion	295.4	298.8	1.2

3.4.2 Foreign Object Damage Analysis

Bird ingestion is a nonlinear transient structural dynamics problem with fluid structure interaction. The blade can fail in the impacted region when local strain exceeds material ductility, or away from the impacted region, when a few cycles of post-impact strain oscillation substantially exceed material strength. Analyses which simulate this behavior have been developed, but high computation time prohibits incorporation in the current design system. Simple chordwise and spanwise bending stress parameters are included in the current Pratt & Whitney Aircraft design system. They represent the effects of moment distributions resulting from a bird slice ingestion at the blade tip. They have been correlated with titanium blade ingestion experience.

3.4.2.1 Spanwise Bending Damage

A more refined linear analysis of spanwise bending response was derived by examination of an impact event where the differential equation of motion on the blade is:

$$[m] \{\ddot{x}\} + [k] \{x\} = \{F(t)\} \quad (26)$$

Considering the response to be a superposition of n natural modes of the blade, the equilibrium equation may be decoupled into n modal equations of the form:

$$\ddot{\xi}_k + \omega_k^2 \xi_k = \frac{1}{m_k} P_k(t) \quad (27)$$

where $\xi_k(t)$ is the modal amplification factor of the k 'th mode, ω_k is the natural frequency, m_k is the modal generalized mass, and $P_k = \lambda_k \{F(t)\}$ is the modal forcing function.

Assuming that the time of load application is short relative to the time when the root stress is most critical, (27) may be solved, giving:

$$\xi_k(t) = \frac{I_k}{m_k \omega_k} \sin(\omega_k t) \quad (28)$$

where I_k is the modal impulse.

Equation (28) provides the means for evaluating the blade root stresses as a function of time. Experience has shown that the highest root stresses occur at the quarter cycle of the first bending time point.

3.4.2.2 Local Damage Analysis

The current design system local impact damage looks at chordwise bending stress at the radius of the center of impact.

The analytical model consists of a blade cross section of incremental span. The loading is distributed over a distance determined by blade to gap and the velocity of the bird relative to the blade. The applied load is assumed to be taken out by the transverse shear and torsional restraint provided by the blade cross sections directly above and below the impacted region. Based on these considerations, the bending moment variation along the chord can be calculated and, since bending stress is inversely proportional to the square of the thickness, the stress parameter is moment/thickness². It peaks at a part chord location.

Scaling to determine an allowable for composite structures has been accomplished based upon the plastic hinge moment capability of the airfoil section.

For a titanium cross-section of thickness, h , the plastic hinge moment is:

$$M_{\max} = 1/4 \sigma_{Y_{TI}} h^2 \quad (29)$$

For a cross-section of a superhybrid blade, the moment carrying capability of the graphite/epoxy has been neglected, because the low ductility of this material would cause it to fracture well before the other blade components reach their maximum loads. In the boron/aluminum, sufficient aluminum ductility exists in the transverse direction for inclusion in the parameter calculation. Hence, on a composite cross-section, the maximum moment becomes:

$$M_{\max} = \sum_{i=1}^n \sigma_{Y_i} \bar{\xi}_i \Delta \xi_i \quad (30)$$

The allowable composite local ingestion parameter then becomes the allowable metallic parameter multiplied by the scale factor:

$$\frac{4 \sum_{i=1}^n \sigma_{Y_i} \bar{\xi}_i \Delta \xi_i}{\sigma_{Y_{TI}} h^2} \quad (31)$$

3.4.3 Flutter Analysis

Flutter is "a self-excited oscillation of an aerodynamic lifting surface". During flutter, the aerodynamic forces couple with the blade elastic and inertia forces and increase the kinetic energy of the blade (negative damping). When this aerodynamic energy exceeds the positive mechanical damping

energy, the blade oscillations grow to destructive amplitudes. Thus, it is imperative that flutter conditions must be avoided to prevent high frequency fatigue failure of blades.

Supersonic flutter of fan stages is evaluated by means of an existing analysis program. Individual modes of vibration and steady state aerodynamic conditions are input to the analysis. Unsteady aerodynamic loads resulting from vibratory motion are calculated by the appropriate aerodynamic analysis. Work done on both the forward and the backward traveling wave implementation of each mode is determined by spanwise integration of the product of resultant unsteady load and input vibratory velocity. Work done is non-dimensionalized by dividing kinetic energy by input vibratory mode and expressed as a logarithmic decrement. The absolute value of the lowest decrement for any mode traveling in either direction represents stage stability limit.

In the analysis of supersonic unstalled flutter, the method of superposition of the basic wave solutions of the linearized flow equation is used to simulate the unsteady aerodynamics of a flat plate cascade oscillating in compressible flow. The method is generalized to cover supersonic relative flow with either subsonic or supersonic axial component ($M_{1x} > 1$), and thus provide a wide range of application.

The basic assumptions of the analysis are the following:

- o Flow is two-dimensional, unsteady, compressible, inviscid, irrotational and isentropic;
- o Cascade is infinite, flat plate, at zero incidence and unstalled;
- o Vibratory motion is small, constant interblade phase angle, 2 degree of freedom (twist and flap) at a blade/disk system natural frequency.

This existing design system analysis was used with evaluation of coefficients at six representative airfoil strips to provide the STAEBL system approximate flutter analysis.

3.4.4 Approximate Analysis System

A comprehensive approximate analysis system has been developed using the approximate stress and vibration analysis, the root and local foreign object damage analysis, and an existing Pratt & Whitney Aircraft aerodynamic damping analysis. The flutter and root damage analyses depend on outputs from the blade vibration analysis; specifically, the natural frequencies and the mode shapes. The intermediate and constraint outputs of the component modules of the approximate analyses are listed on Table 3.4-XII.

TABLE 3.4-XII
APPROXIMATE ANALYSIS SYSTEM MODULE OUTPUTS

<u>Analysis</u>	<u>Intermediate Outputs</u>	<u>Constraints Calculated</u>
Static Analysis	Differential Stiffness	Blade Static Stress
Vibration Analysis	Frequencies, Mode Shapes, Speed Sensitivity	Resonance Margins
Flutter		Aerodynamic Damping Coefficient
Local Foreign Object Damage		Local Foreign Object Damage Parameter
Root Foreign Object Damage		Root Foreign Object Damage Parameter

3.5 OPTIMIZATION METHOD

A common engineering design problem is the determination of values for design variables which minimize design quantity such as weight, drag, or cost, while satisfying a set of auxiliary conditions. In the STAEBL program, the structural design of a composite or hollow fan blade is accomplished by varying airfoil section thicknesses, chord, titanium skin thickness, etc. to minimize a combination of weight and cost subject to constraints on resonance, flutter, stress, and foreign object damage.

3.5.1 General Optimization Theory and Background

The engineering design process can be modeled as a mathematical programming problem in optimization theory. In theoretical terms, this constrained minimization problem can be expressed as follows:

$$\text{minimize } f(\underline{x}), \quad (32)$$

subject to the auxiliary conditions,

$$g_i(\underline{x}) \leq 0, \quad i=1, \dots, m. \quad (33)$$

The quantity $\underline{x} = (x_1, \dots, x_n)$ is the vector of n design variables. The scalar function to be minimized, $f(\underline{x})$, is the objective function; and $g_i(\underline{x}) \leq 0, i=1, \dots, m$, are the m inequality constraints. Upper and lower bounds on the design variables, e.g.,

$$L_i \leq x_i \leq U_i, \quad i=1, \dots, n, \quad (34)$$

are referred to as side constraints. The n -dimensional space spanned by the design variables is design space. If $f(\underline{x})$ and $g_i(\underline{x}), i=1, \dots, m$, are all

linear functions of \underline{x} , then the optimization problem is a linear programming problem (LP) which can be solved by well-known techniques such as Dantzig's simplex method. If $f(\underline{x})$ or any of the $g_i(\underline{x})$'s are nonlinear, then it is a nonlinear programming (NLP) problem for which a number of solution techniques are also available. If the objective function, $f(\underline{x})$, is to be maximized, then the equivalent problem of minimizing $-f(\underline{x})$ is considered.

Any choice of variables, \underline{x} , in design space that satisfies all the constraints, (33) and (34), is a feasible point. As shown in Figure 3.5-1, the union of all feasible points comprises the feasible region. The locus of points which satisfy $g_i(\underline{x}) = 0$, for some i , forms a constraint surface. On one side of the surface, $g_i(\underline{x}) < 0$ and the constraint is satisfied; on the other side, $g_i(\underline{x}) > 0$ and the constraint is violated. Points in the interior of the feasible region are free points; points on the boundary are bound points. If it is composed of two or more distinct sets, the feasible region is disjoint. A design point in the feasible region that minimizes the objective function is an optimal feasible point and is a solution of the problem posed in (32) through (34). As in any nonlinear minimization problem, there can be multiple local minima. In this case, the global minimum is the optimal feasible point. If a design point is on a constraint surface (i.e., $g_i(\underline{x}) = 0$ for some i), then that particular constraint is active. A solution to a structural optimization problem is almost always on the boundary of the feasible region, and is usually at the intersection of two or more constraint surfaces (i.e., there are two or more active constraints).

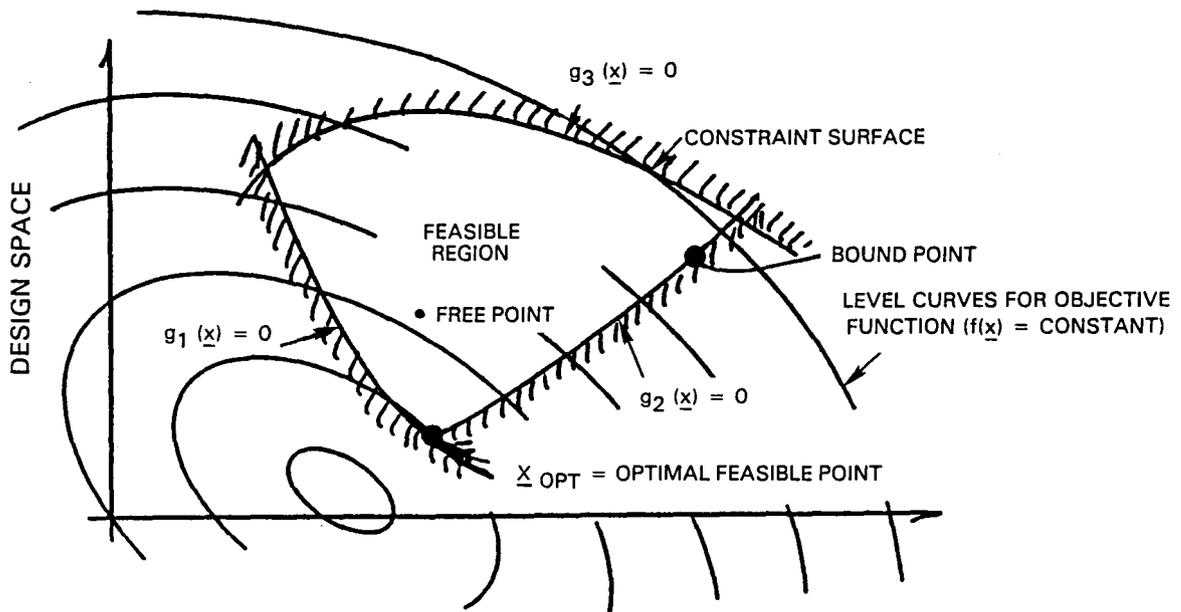


Figure 3.5-1 Feasible Region Is Union of All Points that Satisfy All Constraints

There are two basic approaches to solving the constrained optimization problem posed in (32) through (34): direct methods (e.g., methods of feasible directions) and indirect methods (e.g., penalty function methods).

In a direct method, the objective function and constraints are evaluated independently, and the constraints are treated as limiting surfaces. Zoutendijk's method of feasible directions is an example of a direct method and will be discussed further in Sections 3.5.2 and 3.5.3.

In an indirect method, the problem is reformulated so that (32) through (34) are replaced by a single unconstrained minimization problem. For example, in an exterior penalty function method, violations of the constraints are added onto the objective function to form an augmented objective function,

$$\phi(\underline{x}; R_k) = f(\underline{x}) + R_k \sum_{i=1}^m [g_i(\underline{x})]^2, \quad (35)$$

where:

$$[g_i(\underline{x})]^+ = \begin{cases} 0, & \text{if } g_i(\underline{x}) \leq 0, \\ g_i(\underline{x}), & \text{if } g_i(\underline{x}) > 0. \end{cases} \quad (36)$$

Thus, if a particular constraint is satisfied, then it contributes nothing to the summation in (35). If a constraint is violated, however, then a penalty term is added onto the objective function, which increases as the square of the violation. The design points, \underline{x} , must all satisfy the side constraints (34).

The augmented objective function $\phi(\underline{x}; R_k)$ is minimized for successively increasing values of the penalty parameter R_k , i.e.,

$$R_1 < R_2 < R_3 < \dots \rightarrow \infty. \quad (37)$$

Under rather mild conditions, the sequence of minima $\{x_1, x_2, x_3, \dots\}$ corresponding to (37), converge to a local optimum of the constrained optimization problem identified in (32) through (34). One advantage of this approach is that each of the minimization problems can be solved using a standard unconstrained function minimization technique, such as a conjugate gradient or quasi-Newton method.

Several programs are generally available in software libraries (e.g., IMSL = International Mathematical and Statistical Libraries, Inc., and HARWELL) that can solve the constrained minimization problem using either direct or indirect techniques. Because of its versatility in solving structural optimization problems at Pratt & Whitney Aircraft, NASA/Langley, General Motors, and Ford Motor Co., the COPES/CONMIN computer program was selected for the STAEBL contract. This program was developed by G. N. Vanderplaats of the Naval

Postgraduate School and has the added capability of solving both constrained minimization problems (32) through (34) and unconstrained minimization problems (32). COPEs (COntrol Program for E_ngineering Synthesis) is a user-oriented FORTRAN program that prepares an input data set for the optimization program CONMIN (CONstrained MINimization). Two solution techniques are available for the constrained minimization problem.

1. Exact analysis - utilizes the method of feasible directions applied to the actual objective function and constraints. This approach is discussed in Section 3.5.2.
2. Approximate analysis - utilizes the method of feasible directions applied to Taylor series approximations and to the objective function and constraints. This approach is discussed in Section 3.5.3.

3.5.2 COPEs/CONMIN Exact Analysis: Method of Feasible Directions

In this method, a sequence of designs (x_0, x_1, \dots) is produced which converges to a local optimum design, x_{opt} , provided a feasible region exists. The successive designs are generated iteratively as a sequence of one-dimensional line searches, i.e.,

$$x_{i+1} = x_i + \alpha s_i, \quad (38)$$

for $i = 0, 1, 2, \dots$, where s_i is the search direction and α are chosen so that once the feasible region has been entered, all subsequent iterates remain feasible and the magnitude of the objective function is reduced at each step. If the initial design, x_0 , is infeasible, then gradients of the violated constraints are calculated so that search directions can be established which lead to the feasible region, provided one exists.

Once the feasible region has been entered, a particular direction is pursued until either: a) a local minimum of the objective function, $f(x)$, has been determined or, b) a constraint boundary has been reached. The value of α in (38) at the termination point of this one-dimensional line search in the s_i direction is determined by interpolating polynomial fits of several trial values of the objective function and constraints. A schematic of a typical case is shown in Figure 3.5-2. The initial design, x_0 , is infeasible. The design point, x_i , is a relative minimum of the objective function. The remaining search directions terminate at constant boundaries until x_{opt} is reached.

If a local minimum of the objective function has been reached, then the gradient of the objective function is calculated, and the procedure continues in the direction opposite to this (i.e., the "path of steepest descent"). If a constraint boundary has been reached first, however, then a new search direction can be determined using Zoutendijk's method of feasible directions as follows. A direction, s_i , is usable if the objective function initially does not increase along this path, i.e.,

$$s_i \cdot \nabla f(x_i) < 0. \quad (39)$$

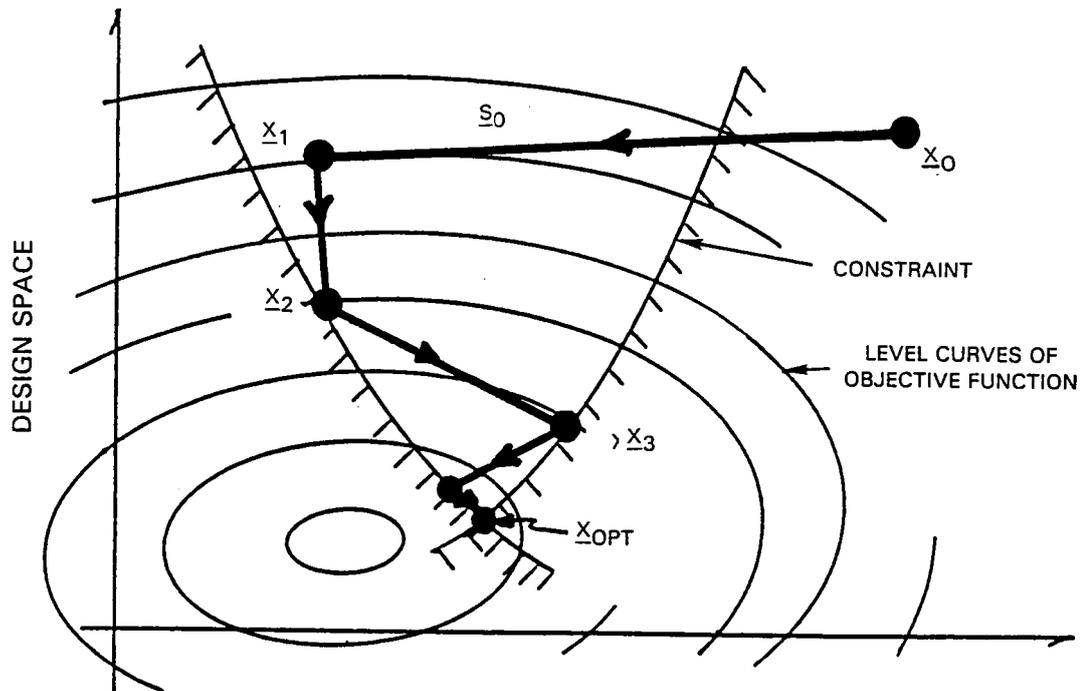


Figure 3.5-2 Line Search Terminates Either at Minimum of Objective Function or at a Constraint Boundary. Sequence of line searches converge to \underline{x}_{opt} .

In addition, \underline{s}_j is feasible if no active constraints are initially violated along this path, i.e.,

$$\underline{s}_j \cdot \bar{\nu} g_j(\underline{x}_j) \leq 0, \quad j = 1, \dots, \text{NAC}, \quad (40)$$

where a subscript, j , is chosen for each of the constraints that are active at \underline{x}_j . As shown schematically in Figure 3.5-2, allowable paths that emanate from \underline{x}_j comprise the usable feasible sector.

3.5.2.1 Choice of Search Parameters for COPES/CONMIN

In Zontendijk's method, the search direction, \underline{s}_j , is determined by solving a sub-optimization problem, i.e.,

$$\begin{aligned} & \text{subject to:} && \text{maximize } \beta, \\ & && \underline{s}_j \cdot \bar{\nu} f(\underline{x}_j) + \beta \leq 0, \\ & && \underline{s}_j \cdot \bar{\nu} g_j(\underline{x}_j) + \theta_j \beta \leq 0, \quad j = 1, \dots, \text{NAC} \\ & && |\underline{s}_j| \text{ bounded.} \end{aligned} \quad (41)$$

The parameter θ_j , the push-off factor, determines the orientation of the new search direction vector, \underline{s}_j , in the usable feasible sector by pushing the

search away from the constraints into the feasible region. As shown in Figure 3.5-3, s_i approaches the constraint surface, $g_j(x)$, tangentially as $\theta_j \rightarrow 0$, and s_i approaches a level curve to the objective function tangentially as $\theta_j \rightarrow \infty$. For a linear constraint, θ_j can be set to zero and the search can proceed along that particular constraint surface. If θ_j is too small, then for nonlinear constraints with convex curvature, the same constraint will be immediately re-encountered. In this case, the search will "skid" along the same constraint boundary with little change in the objective function. If θ_j is too large, then the search will "zigzag" back and forth between two or more constraints, and the objective function will again not be reduced rapidly enough. A compromise value of $\theta_j = 1$ is the default value used by COPES/CONMIN for the initial iteration. Since many of the constraints (e.g., flutter, resonance, etc.) in the STAEBL optimization problems were nearly linear (at least locally), the value $\theta_j = 0.3$ was used for the initial iteration to give more rapid convergence.

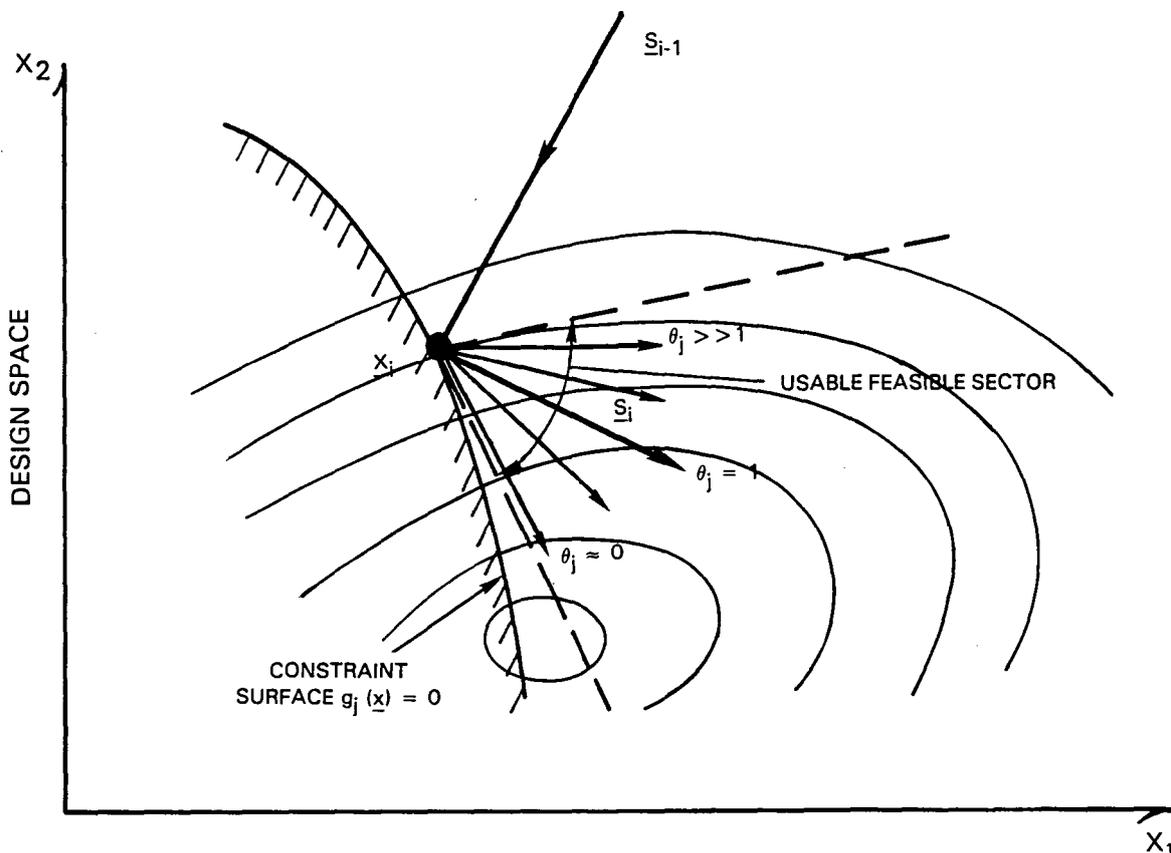


Figure 3.5-3 New Search Direction, s_i , Lies in the Usable Feasible Sector. The value of the push-off factor, θ_j , determines the orientation of the new search direction.

The rate of convergence is also affected by the value of CT, the constraint thickness parameter in COPES/CONMIN. For theoretical purposes, the i th constraint is satisfied if $g_i(x) \leq 0$ and is active if $g_i(x) = 0$. For computational purposes (as shown in Figure 3.5-4), COPES/CONMIN considers the

j th constraint to be satisfied if $g_j(x) \leq CT$ and to be active if $|g_j(x)| \leq -CT$, where CT is a negative number. If $|CT|$ is too small, then one or more constraints can be active on one iteration and inactive on the next, only to become active again on a subsequent iteration - another instance of "zigzagging". A proper choice of CT ensures that two or more constraints will often be simultaneously active when a new search direction is chosen. In this case, as shown in Figure 3.5-5, the search will proceed down the "valley" formed by the constraint surfaces. The default value in COPES/CONMIN is $CT = -0.1$ (i.e., a constraint is considered active if it is within 10 percent of its specified value). For many STAEBL applications, a value $CT = -0.1$ was too large since too many constraints were simultaneously active during the early iterations, and new search directions could not be established. Consequently, the value $CT = -0.05$ was used.

During the COPES/CONMIN optimization procedure, the values of CT and θ_j are updated as follows. After the first few iterations, the value of CT is decreased monotonically so that fewer constraints will be active when new search directions are established. A minimum value of $|CT|$ is given by $CTMIN$; the default value in COPES/CONMIN is $CTMIN = 0.004$. In addition, the value of the push-off factor, θ_j , is also readjusted at each iteration according to the value of the active constraint to which it applies and to the current value of CT . Thus, θ_j is a quadratic function of these parameters, i.e.,

$$\theta_j = \theta_0 \left(\frac{g_j(x_j)}{CT} - 1 \right)^2, \quad (42)$$

where θ_0 is the initial value of θ_j (for STAEBL we have chosen $\theta_0 = 0.3$). A maximum value of $\theta_j = 50$ is also imposed.

The iteration is terminated under three conditions in COPES/CONMIN:

1. If the objective functions for three successive iterates are all within a prescribed error tolerance, then the procedure has converged to a local optimum. COPES/CONMIN uses default values of $DELFUN = 0.0001$ for the relative change in objective function and $(DABFUN = 0.0001) \times$ initial objective value for absolute change in the objective function as its convergence criteria. For STAEBL application, 1 percent differences in the objective function were adequate for convergence so that $DELFUN = DABFUN = 0.01$. These increased values also reduced the number of function calls required for convergence.
2. If convergence has not been obtained after a certain number of iterations inside the feasible region, the procedure is terminated. Either this design can be accepted or else the optimization procedure can be restarted if progress toward an optimum is obviously being made. COPES/CONMIN uses a default value of 20 for the total number of iterations.
3. If the feasible region cannot be located after a certain number of iterations (the COPES/CONMIN default value is 10), then the process is terminated. At this time, either a new starting guess should be chosen, or else the objective function and constraints should be examined to determine whether or not a feasible region exists.

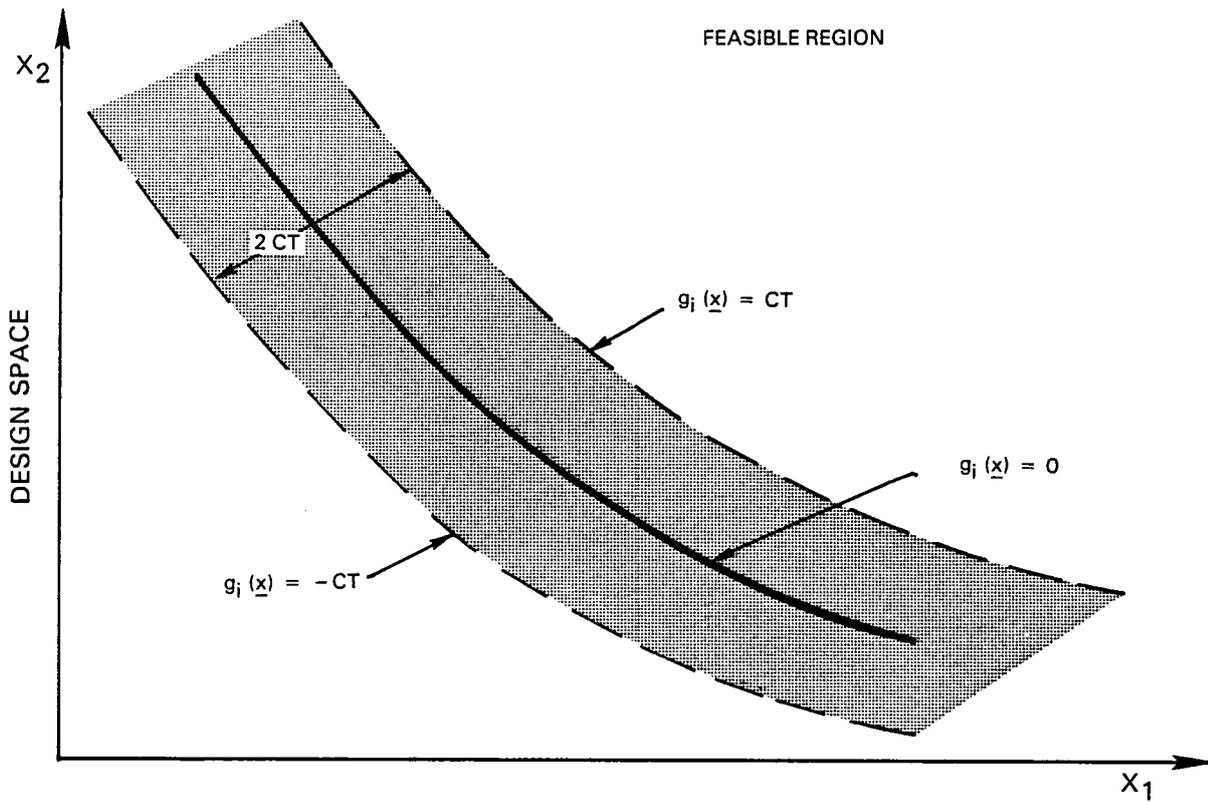


Figure 3.5-4 Constraint Thickness Parameter, CT , Determines when a Constraint is Satisfied, Violated, or Active

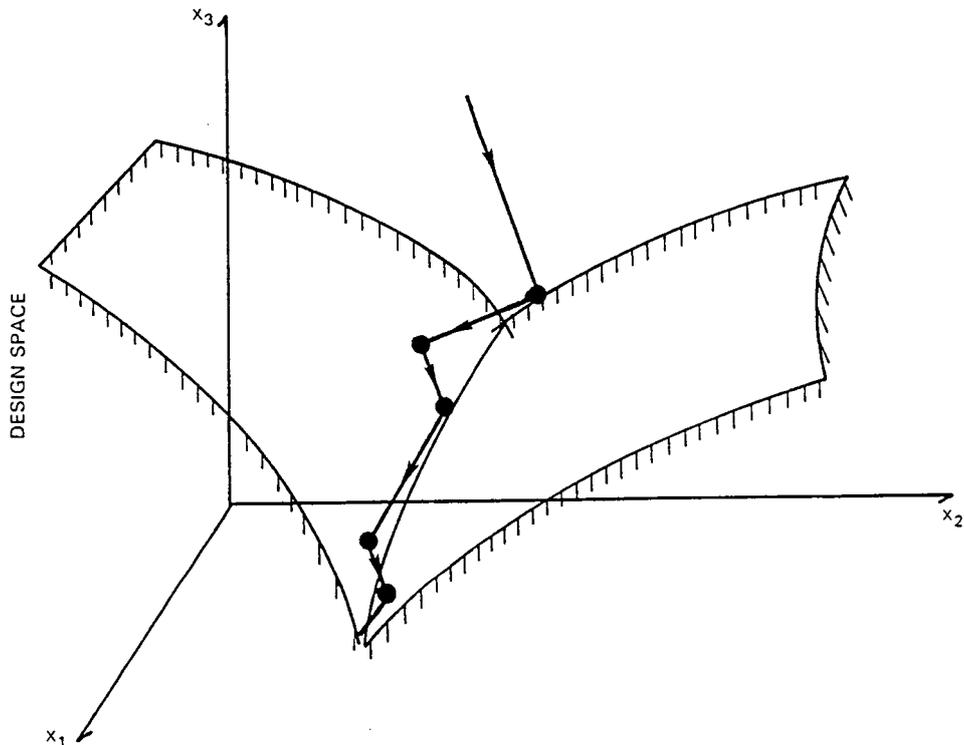


Figure 3.5-5 For Proper Choice of CT , Two Constraints Become Simultaneously Active so that Search Proceeds down the "Valley" Formed by the Constraints

3.5.2.2 Scaling of Design Variables in COPES/CONMIN

Performance of the method of feasible directions can be greatly affected by the scaling of the design variables. At the beginning of each iteration in COPES/CONMIN, a new search direction is established according to Zoutendijk's method (41). This procedure is based upon the gradient of the objective function and each constraint with respect to each of the design variables. The choice of the search direction is very sensitive to the components of these gradients. For example, in a two design variable problem, suppose that a 1 percent change in x_1 leads to a 10 percent change in the objective function, $f(\bar{x})$; whereas a 1 percent change in x_2 leads to only a 0.1 percent change in $f(\bar{x})$. To reduce the objective function most rapidly, the search direction will be primarily in the x_1 direction. The "weak" variable, x_2 , will be virtually unchanged, at least for several iterations. To obtain the optimal design, a relatively large change in x_2 must be made to affect the objective function and constraints.

In a well-formulated problem, the components of the gradient of the objective function with respect to the design variables should all be roughly the same order of magnitude. The scaling option in COPES/CONMIN can be used to equilibrate the gradient components as follows. The i^{th} design variable, x_i , is scaled by dividing it by its initial value x_i^0 , i.e.;

$$\xi_i = x_i / x_i^0 \quad (43)$$

provided x_i^0 is nonzero. Using (43) in the chain rule, the i^{th} component of the scaled gradient with respect to the nondimensional variable, ξ_i , is given by:

$$\frac{\partial f}{\partial \xi_i} = \frac{\partial x_i}{\partial \xi_i} \frac{\partial f}{\partial x_i} = x_i^0 \frac{\partial f}{\partial x_i} \quad (44)$$

Thus, the ratio of the i^{th} components in the scaled gradients is given by:

$$\frac{\partial f}{\partial \xi_i} / \frac{\partial f}{\partial x_i} = x_i^0 \quad (45)$$

The effect of scaling was exhibited by the optimal design of a hollow fan blade with a titanium outer skin. The four design variables were: x_1 and x_2 (blade thicknesses, t_{root} and t_{tip} , at root and tip, respectively); x_3 (distance, d_{root} , from bottom of hollow section to the blade root); and x_4 (titanium skin thickness, t_{Tj}). The objective function, $f(\bar{x})$, was blade weight. As shown in Table 3.5-I, the components of the gradient of the objective function varied by two orders of magnitude since $\partial f / \partial x_3 = 0.027$ and $\partial f / \partial x_4 = 2.7$. Consequently, without scaling, the minimization proceeded by varying the titanium skin thickness, x_4 , and leaving x_3 virtually unchanged - an optimum design was not achieved. When scaling was introduced according to (43), the gradient components in Table 3.5-I were all of the same order of magnitude. In this case, the search direction varied all four design variables (including the hollow section location) simultaneously so that an optimal design was obtained.

TABLE 3.5-I

UNSCALED AND SCALED VALUES OF GRADIENT OF OBJECTIVE FUNCTION RELATIVE TO THE DESIGN VARIABLES FOR HOLLOW FAN BLADE

Design Variable, x_i	Initial value cm (inches)	i^{th} Component of Gradient = $\partial f / \partial x_i$	
		Unscaled	Scaled
$x_1 = t_{\text{root}}$	2.22 (0.875)	0.460	0.403
$x_2 = t_{\text{tip}}$	0.85 (0.334)	0.140	0.047
$x_3 = d_{\text{root}}$	24.51 (9.650)	0.027	0.260
$x_4 = t_{\text{Tj}}$	0.30 (0.120)	2.700	0.324

The scaling options in COPES/CONMIN are controlled by the input parameter NSCAL and are given as follows:

$$\text{NACAL} \begin{cases} > 0 : & \text{Rescale design variables by dividing by current values every NSCAL iteration,} \\ = 0 : & \text{No scaling (default value),} \\ < 0 : & \text{Scale design variables by dividing by user-input scaling variables} \end{cases} \quad (46)$$

For STAEBL demonstration, scaling was always used. The value NSCAL = n+1 (where n = number of design variables) was recommended by G. Vanderplaats since this strategy worked well for unconstrained minimization problems using the conjugate gradient method.

3.5.2.3 Number of Function Calls for COPES/CONMIN

Engineering design problems are considered small or large according to the number of design variables as follows:

$$\begin{array}{ll} \text{Small:} & n \leq 10 , \\ \text{Moderate:} & 10 < n \leq 50 , \\ \text{Large:} & n > 50. \end{array} \quad (47)$$

The number, N, of function calls required for convergence of the method of feasible directions for COPES/CONMIN can be approximated as follows. As indicated in Figure 3.5-2, each iteration consists of a gradient evaluation of the objective function and constraints to determine the search direction, followed by a one-dimensional line search in that direction. When the gradients are not known analytically (as is the case for the STAEBL application), a backward difference gradient approximation is used. For n design variables, n function calls are required for the finite difference

gradient calculation. The one-dimensional line search usually requires 3 additional function evaluations to update the objective function and constraints and to determine where the search should terminate. Thus, for m iterations, with $n+3$ function calls per iteration, we have:

$$N = m (n + 3). \quad (48)$$

Typically, convergence is attained in approximately 10 iterations so that $N \approx 10n + 30$. Note that N increases roughly linearly as a function of the number, n , of design variables.

The limiting feature in these analyses is the computer time required per function call to evaluate the objective function and constraints. For example, calculation of three natural frequencies, mode shapes, resonance function, flutter parameters, stresses, weight, and bird ingestion parameters for a gas turbine engine fan blade requires approximately 25 seconds on an IBM 3033 computer. In this case, 10 iterations for a 12 design-variable optimization problem would require approximately 150 function calls and 1 hour of computer time. Sometimes the number of function calls can be reduced by using the approximate analysis version of COPES/CONMIN, discussed in Section 3.5.3.

3.5.3 COPES/CONMIN Approximate Analysis: Taylor Series Expansions

In the exact analysis version of COPES/CONMIN for the method of feasible directions, the actual vibration, flutter and stress computer programs are implemented for every function call to calculate the objective function and constraints. In the approximate analysis version of COPES/CONMIN, the objective function and constraints are all represented numerically by second order Taylor series expansions about some nominal design point. After the objective function and constraints have been evaluated for several distinct design points, linear and quadratic polynomial surfaces can be curve-fit through these points. Then, instead of implementing the actual vibration, flutter, and stress programs, the objective function and constraints can be closely approximated by evaluating the Taylor series expansions. As higher order terms in the series are determined, the approximate analyses become increasingly accurate.

In general, the Taylor series expansion through quadratic terms for an arbitrary function, f , (representing either the objective function or constraints) is given by:

$$f(\underline{x}) \approx f(\underline{x}_0) + \bar{\nabla} f(\underline{x}_0) \cdot (\underline{x} - \underline{x}_0) + \frac{1}{2} (\underline{x} - \underline{x}_0) \cdot \underline{\underline{H}}(\underline{x}_0) \cdot (\underline{x} - \underline{x}_0), \quad (49)$$

where $\underline{x} = (x_1, x_2, \dots, x_n)^T$ is a vector of n design variables, \underline{x}_0 is the nominal design (center of expansion), and \cdot denotes dot product, i.e., $u \cdot v = u^T v$, where T denotes transpose. The gradient vector, $\bar{\nabla} f(\underline{x}_0)$, and Hessian matrix, $\underline{\underline{H}}(\underline{x}_0)$, are given by:

$$\bar{v}f = \begin{bmatrix} \partial f/\partial x_1 \\ \partial f/\partial x_2 \\ \cdot \\ \cdot \\ \cdot \\ \partial f/\partial x_n \end{bmatrix}, H = \begin{bmatrix} \partial^2 f/\partial x_1^2 & \partial^2 f/\partial x_1 \partial x_2 & \dots & \partial^2 f/\partial x_1 \partial x_n \\ \partial^2 f/\partial x_2 \partial x_1 & \partial^2 f/\partial x_2^2 & \dots & \partial^2 f/\partial x_2 \partial x_n \\ \cdot & \cdot & \cdot & \cdot \\ \partial^2 f/\partial x_n \partial x_1 & \partial^2 f/\partial x_n \partial x_2 & \dots & \partial^2 f/\partial x_n \partial x_n \end{bmatrix} \quad (50)$$

The coefficients in the Taylor series expansion (i.e., $f(x_0)$, and all the terms in the gradient vector and Hessian matrix) in (49) are unknown. Since the Hessian matrix is symmetric, the total number, N_Q , of unknowns in (49) is given by:

$$N_Q = 1 + n + n(n+1)/2. \quad (51)$$

After N_Q linearly independent function evaluations have been made, all unknown coefficients in (49) are determined, and the quadratic Taylor series approximations are available. Thus, the method of feasible directions can then be applied to the quadratic approximations themselves instead of calling the actual vibration and stress programs. By evaluating the weight, natural frequencies, aerodynamic damping parameters, etc. using the Taylor series approximations, COPEs/CONMIN can reduce the number of expensive calls to the vibration programs and thereby reduce the total computer time of the optimization program. In addition, since the objective function and constraints are expressed in terms of quadratic polynomials, the gradient calculations required by the method of feasible directions can be performed analytically rather than by finite differences.

The approximate optimization technique is performed sequentially. The first design, \underline{x}_0 , (the nominal design) is used to determine the constant term, $f(\underline{x}_0)$, in the Taylor series approximation. Another nearby design is selected so that one term in the gradient vector in (49) can be determined. As more designs are evaluated, more terms in the Taylor series can be evaluated. After each new approximation is made, the method of feasible directions is applied to the current approximation to obtain a new local optimum. This new approximate optimum design is then evaluated using the actual vibration and stress programs. This new design point is then appended to the sequence of previous designs so that the next term in the quadratic Taylor series can be determined. After $n + 1$ linearly independent designs have been determined, then all the first order (gradient) terms in the Taylor series (49) are known. The next $n(n+1)/2$ design evaluations are used to determine the Hessian matrix. A sequence of local optimum designs determined by applying the method of feasible directions to a sequence of local Taylor series approximations is

illustrated schematically in Figure 3.5-6. In Figure 3.5-6(A), only linear approximations to the objective function and constraints can be made. In Figure 3.5-6(B), more than $n+1$ design points are known so that some curvature in the objective function and constraint surfaces can be modeled. Eventually, in Figure 3.5-6(C), more than N_Q design points have been evaluated so that a full quadratic approximation can be made.

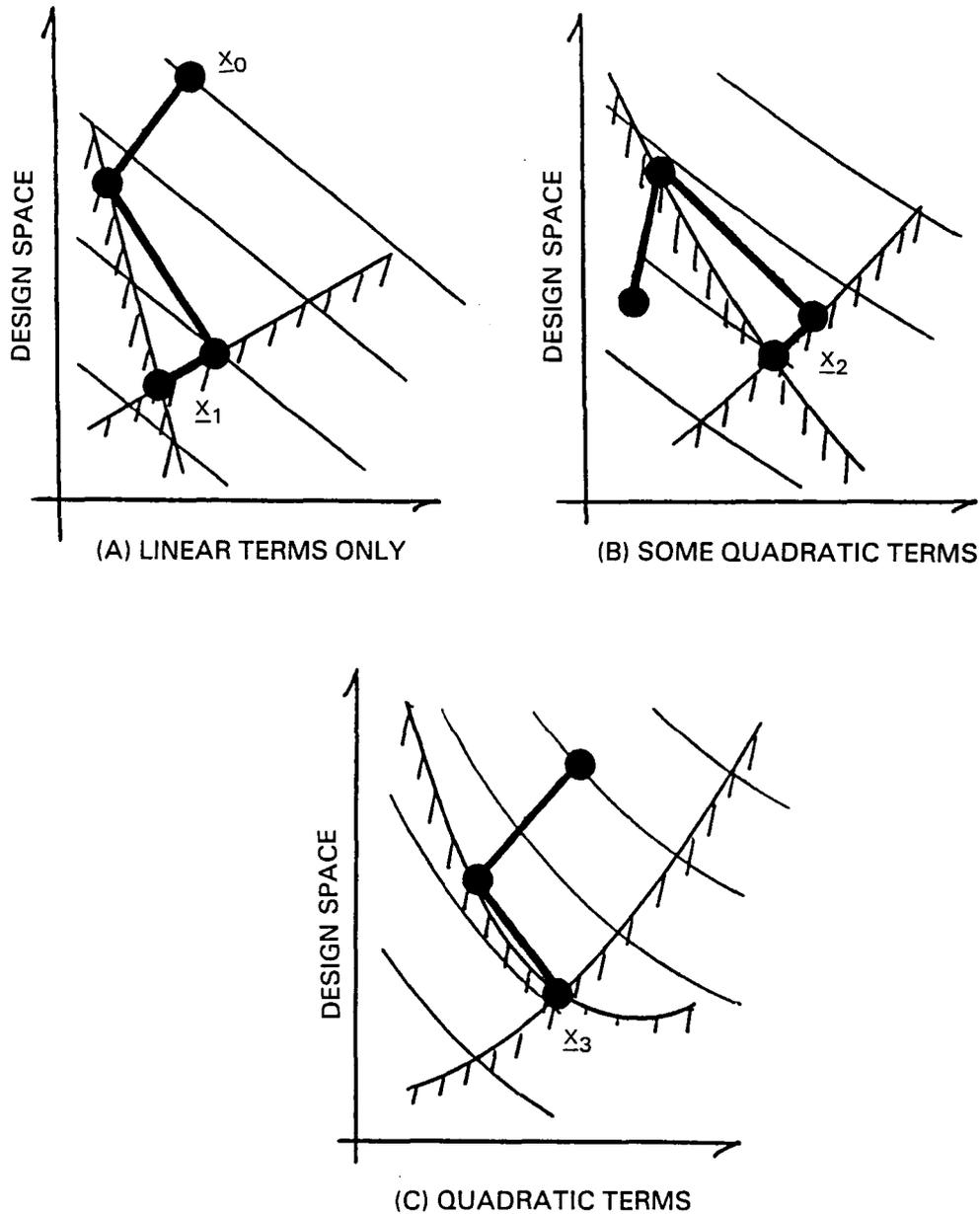


Figure 3.5-6 Sequence of Approximate Analyses Using the Method of Feasible Directions Becomes Increasingly More Accurate as More Terms in the Taylor Series Are Determined.

Since some of the initial approximations to the problem can have unbounded solutions, upper and lower bounds are placed on the design variables to limit the excursion between successive iterates. Once all N_0 terms in (49) have been determined, successive iterates should be close together and converge to a local optimum design. Convergence is obtained when the local optima of two successive approximate problems agree within a certain tolerance. If more than N_0 linearly independent designs are determined, then a weighted least squares fit of the redundant data is used to determine the Taylor series coefficients in (49). A default value for the upper limit in the number of function calls is approximately $3N_0$ in COPES/CONMIN.

Assuming that $m = 10$ iterations are required for convergence for the exact analysis, the number of function calls is given approximately by (48) as $N = 10n + 30$. For the quadratic Taylor series approximation, there are $N_0 = 1 + n + n(n+1)/2$ unknowns. Equating N and N_0 , we find that the break even point is approximately $n = 20$ design variables. Further, since the number of function calls for the quadratic Taylor series approximation and exact analysis are proportional to n^2 and n , respectively, the quadratic Taylor series method rapidly becomes noncompetitive for $n > 2$.

In some cases, certain reduced approximations can be made which can lead to further reductions in the number of function calls required for convergence. Two options are available in COPES/CONMIN in which not all N_0 unknowns in (49) need to be determined. First, if the off-diagonal terms in the Hessian matrix are neglected, then there are only n diagonal terms to be determined. In this model, the total number of unknowns in equation (49) is reduced to:

$$N_D = 1 + n + n, \quad (52)$$

so that N_D is now a linear function of n . Before using the diagonal Hessian matrix option, it is sometimes instructive to analyze the problem using the full Hessian matrix option to determine how large the off-diagonal terms are compared to the diagonal terms. If there are several large off-diagonal terms in critical functions (i.e., objective function or active constraints), then the full quadratic approximation may converge more rapidly than the diagonal Hessian option.

A second reduced approximation is available in COPES/CONMIN by neglecting all the quadratic terms in the Taylor series. The approximating model is now strictly linear and the total number of unknowns in (49) is given by:

$$N_L = 1 + n. \quad (53)$$

Since a sequence of constrained linear optimization problems is solved until convergence is obtained, this approach is known as "sequential linear programming". Each linear subproblem can be solved using either the method of feasible directions or the Simplex method.

In summary, the linear and diagonal Hessian Taylor series approximations require the fewest function evaluations and should be used whenever possible. The success of the approximate techniques depends upon how accurately the Taylor series approximate the objective function and active constraints in a

neighborhood of a local optimum. If the actual functions are relatively smooth (e.g., differentiable with no rapid fluctuations) then the Taylor series fits should be very accurate. On the other hand, if the objective function or constraints are nonsmooth (e.g., discontinuous or rapidly varying), then the approximate techniques may not provide accurate representations of these surfaces near a local optimum. (It should be pointed out that the finite difference gradient calculations for the exact analysis may also experience difficulty in these cases.) Plots of level curves in two-variable function space (i.e., plotting objective function and constraint values as functions of only two design variables while holding the other design variables constant) demonstrate that airfoil weight, resonance margin, and aerodynamic damping exhibit nearly linear variation (at least locally) with blade thickness throughout a large portion of design space. Thus, in many STAEBL applications, quadratic, diagonal Hessian, and sequential linear Taylor series approximations have provided rapid and accurate results.

Convergence difficulties for the approximate optimization methods can occur due to linear dependence among the designs. The unknown coefficients in the Taylor series approximation are determined by solving systems of simultaneous equations. For example, if (49) were evaluated at N_0 designs, then there would be N_0 simultaneous linear equations to be solved for the N_0 unknown coefficients for the objective function and each of the constraints. If two of the N_0 designs were identical or numerically very "close" however, then the coefficient matrices for the linear systems of equations would be singular or numerically singular, respectively (i.e., the designs would be linearly dependent or numerically dependent, respectively). This situation can arise, for example if an initial design either coincides with, or was very close to, a local optima. In this case, it is very possible that this local optimum design point would be generated several times during the first N_0 iterations. Whenever a previously generated design is appended to the sequence of designs, the resulting system of equations would be linearly dependent and could not be solved. By taking several new starting points, COPES/CONMIN attempts to determine a new linearly independent design. There is also an option for the user to supply a priori his own set of linearly independent design vectors. In this case, COPES/CONMIN evaluates the objective function and constraints at these user-supplied points and then determines the surface fits through these points.

The approximate analysis option in COPES/CONMIN is especially useful if only a limited amount of data for the objective function and constraints are available. For example, some finite element programs (such as NASTRAN or MARC) for structural analyses are so time consuming that only a few function calls can be afforded for an optimization program. In other cases, data may be obtainable only through experiments so that only a limited number may be available. In the approximate analysis option, the available data can be entered a priori and the method of feasible directions can be applied to the Taylor series approximations.

3.5.4 COPES/CONMIN Interfaces to Vibration, Flutter, and Stress Programs

The COPES/CONMIN program is limited via subroutine ANALIZ to the approximate vibration, flutter, stress, and foreign object damage programs used for the structural analysis of blades at Pratt & Whitney Aircraft/Commercial Products

Division via subroutine analyse as shown in Figure 3.5-7. Once an optimal feasible design has been obtained by COPES/CONMIN, this blade design must be evaluated by the refined analysis (finite element program) for further tailoring and possible re-optimization.

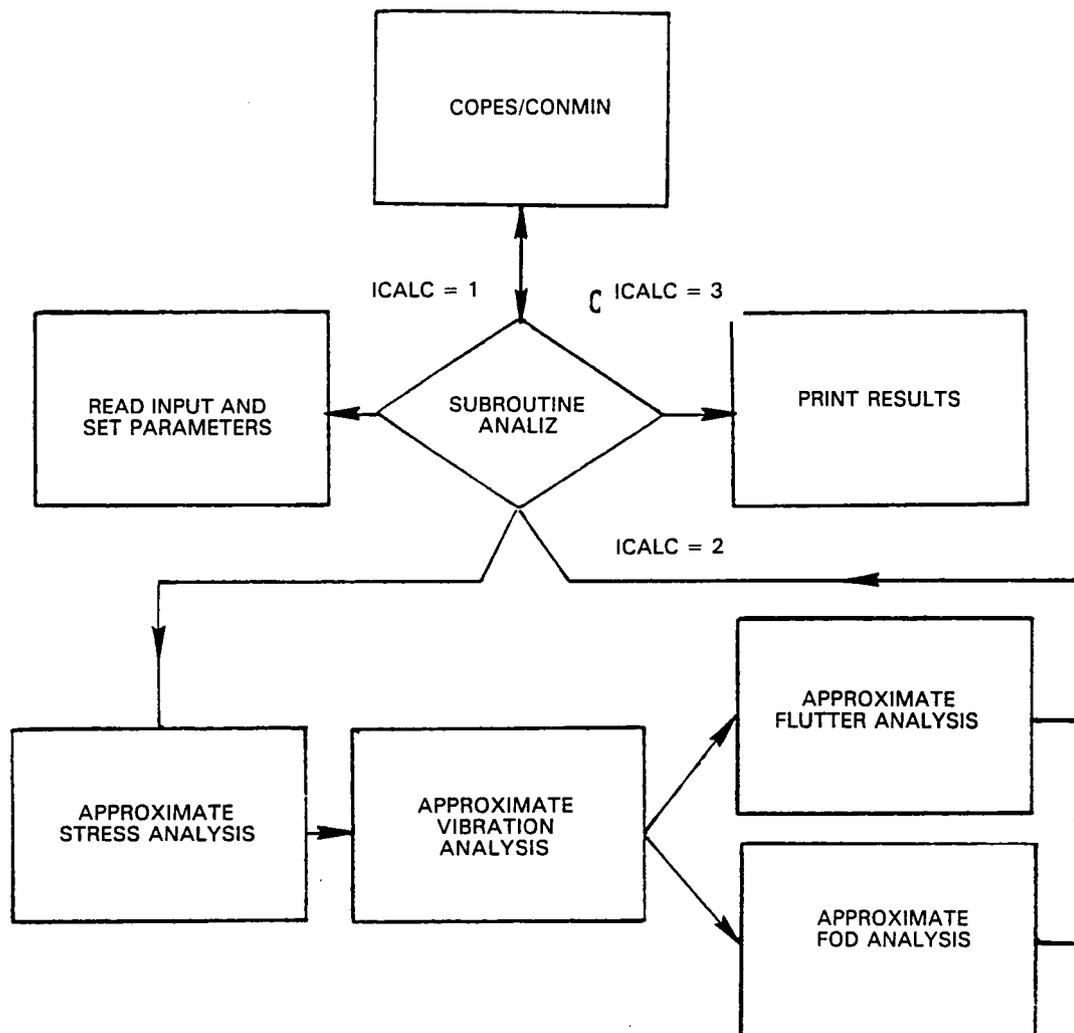


Figure 3.5-7 COPES/CONMIN Is Linked Via Subroutine ANALYZE to Approach analyses

Subroutine ANALIZ is called by COPES/CONMIN in order to evaluate the objective function and constraints. There are 3 options, designated by different values of the parameter ICALC, utilized by COPES/CONMIN when calling subroutine ANALIZ:

- ICALC = 1: Read data, set the parameters that are used throughout the analysis, and analyze the initial design.
- ICALC = 2: Analyze the current design.
- ICALC = 3: Write output data and parameters and results of analysis on final design.

In order to accomplish these tasks, subroutine ANALIZ calls the vibration, flutter, stress, and FOD programs whenever necessary. The transfer of information between COPES/CONMIN and these approximate analyses is accomplished by accessing the data in common block GLOBCM. In Table 3.5-II, the global locations in GLOBCM and the FORTRAN names are given for the design variables, objective function, and constraint information used in the STAEBL procedure included in Appendix B.

TABLE 3.5-II

FORTRAN NAMES AND GLOBAL LOCATIONS IN GLOBCM FOR DESIGN VARIABLES, OBJECTIVE FUNCTION, AND CONSTRAINTS FOR STAEBL BLADE APPLICATIONS

<u>Global Location</u>	<u>FORTRAN Name</u>	<u>Definition</u>
1	OBJF	Airfoil weight
2-6	FN	Frequencies (Root 1 - Root 5)
7-11	DLAR	Aerodynamic damping parameter (Root 1 - Root 5)
12-32	THKVAL	Airfoil thicknesses (max. of 21 stations)
33-37	RF	Resonance margin for 1st order (Root 1 - Root 5)
38-42	RF	Resonance margin for 2nd order (Root 1 - Root 5)
43-47	RF	Resonance margin for 3rd order (Root 1 - Root 5)
48-52	RF	Resonance margin for 4th order (Root 1 - Root 5)
53	BRCC	Root Chord
54-74	FODLSB	LSBIP = Local stress bird ingestion parameter
75	DLE	Distance to hole from leading edge
76	DTE	Distance to hole from trailing edge
77	DROOT	Distance to hole from root edge
78	DTIP	Distance to hole from tip edge
79	TTI	Thickness of titanium
80	TLT	Thickness of borsic titanium
81	OBJFUN	Objective function
82-89	SMAX1S	Root Tsai-Wu (layer 1-8)
90-89	SMAX2S	Hole Tsai-Wu (layer 1-8)
98-105	SMAX3S	Spanwise bending ingestion Tsai-Wu (layer 1-8)
106-113	SMAXLS	Leading edge Tsai-Wu (layer 1-8)
114-121	SMAXTS	Trailing edge Tsai-Wu (layer 1-8)
122-128	Theta	Fiber direction (7 layers)
129	HLRTIO	Hole to length ratio
130	ECRTIO	Edge to chord ratio
131-151	TOVB	t/b's
152	FODMAX	Max LSBIP
153	TIS	Titanium skin thickness
154	TIC	Titanium center thickness
155	PCBA	% boron aluminum
156	VAL	2/TIS + TIC)/THKVAL(1)

3.5.5 Test Cases for STAEBL Optimization Procedure

Several test cases were used to demonstrate certain features of the STAEBL optimization procedure. Since the first three examples involved two design variables, the two-variable function space option of COPES/CONMIN was used. With this feature, the objective function and constraints were evaluated over a rectangular grid of points so that the constraint curves and level curves for the objective function could be presented.

Case 1: Local optimum design for disjoint feasible region

Determine (t_r, t_t) that minimize blade weight:

Subject to:

$$\begin{aligned} 0.25 < t_r < 0.5 \\ 0.1 < t_t < 0.3, \end{aligned}$$

$$\begin{aligned} 130 < \omega_1 < 173 \\ 272 < \omega_2 < 320 \\ 390 < \omega_3 < 504. \end{aligned}$$

For this problem, (t_r, t_t) are the maximum blade thicknesses (in inches) at the root and tip, respectively, and $\omega_i, i = 1, 2, 3$, are the first three blade frequencies (in Hertz). The level curves for the blade weight were equally spaced lines. Due to the strongly nonlinear behavior of ω_3 , the feasible region was composed of two distinct components. As shown in Figure 3.5-8, the initial design was infeasible, and the STAEBL optimization procedure converged to a local optimum design after five iterations.

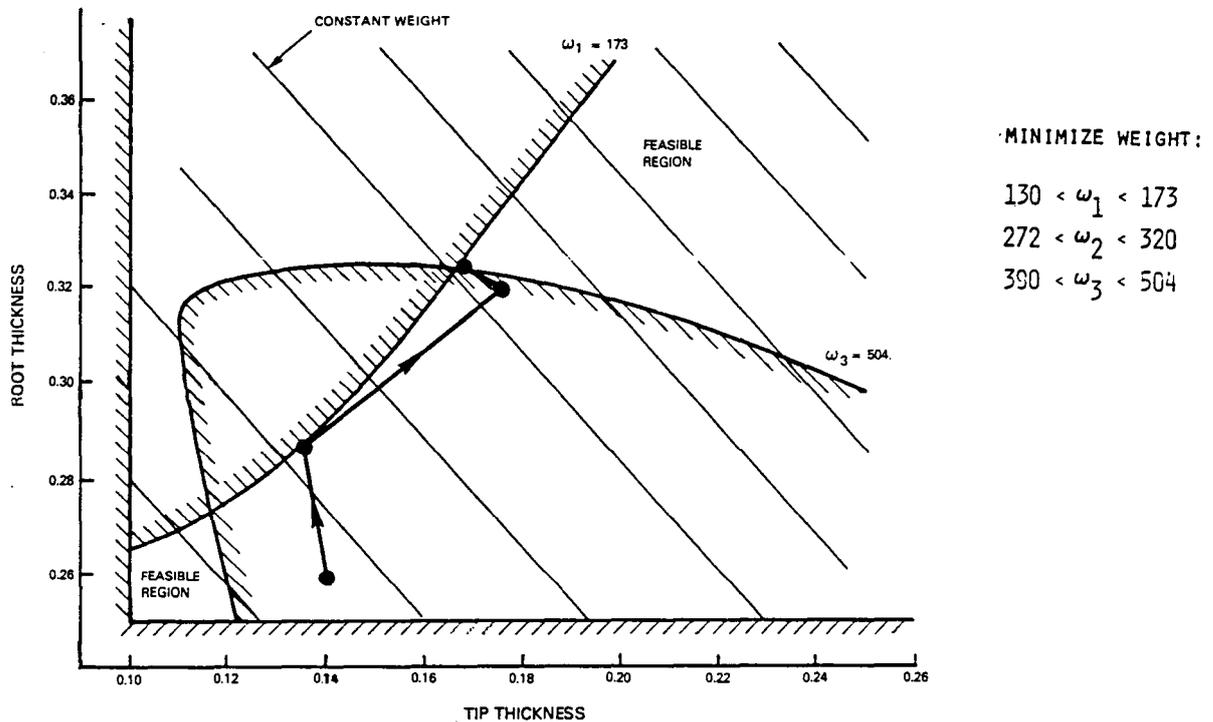


Figure 3.5-8 STAEBL Optimization Procedure May Not Converge to Global Optimum if Feasible Region is Disjoint

Case 2: Global optimum design for disjoint feasible region with three flutter constraints

This example had the same objective function and constraints as Case 1 except that the following constraints on the first three aerodynamic damping parameters were imposed: $\delta_i < 0.015$, $i = 1, 2, 3$. As shown in Figure 3.5-9, these flutter constraints forced the search into a relatively small feasible region containing the global optimum, so that convergence was achieved rather quickly.

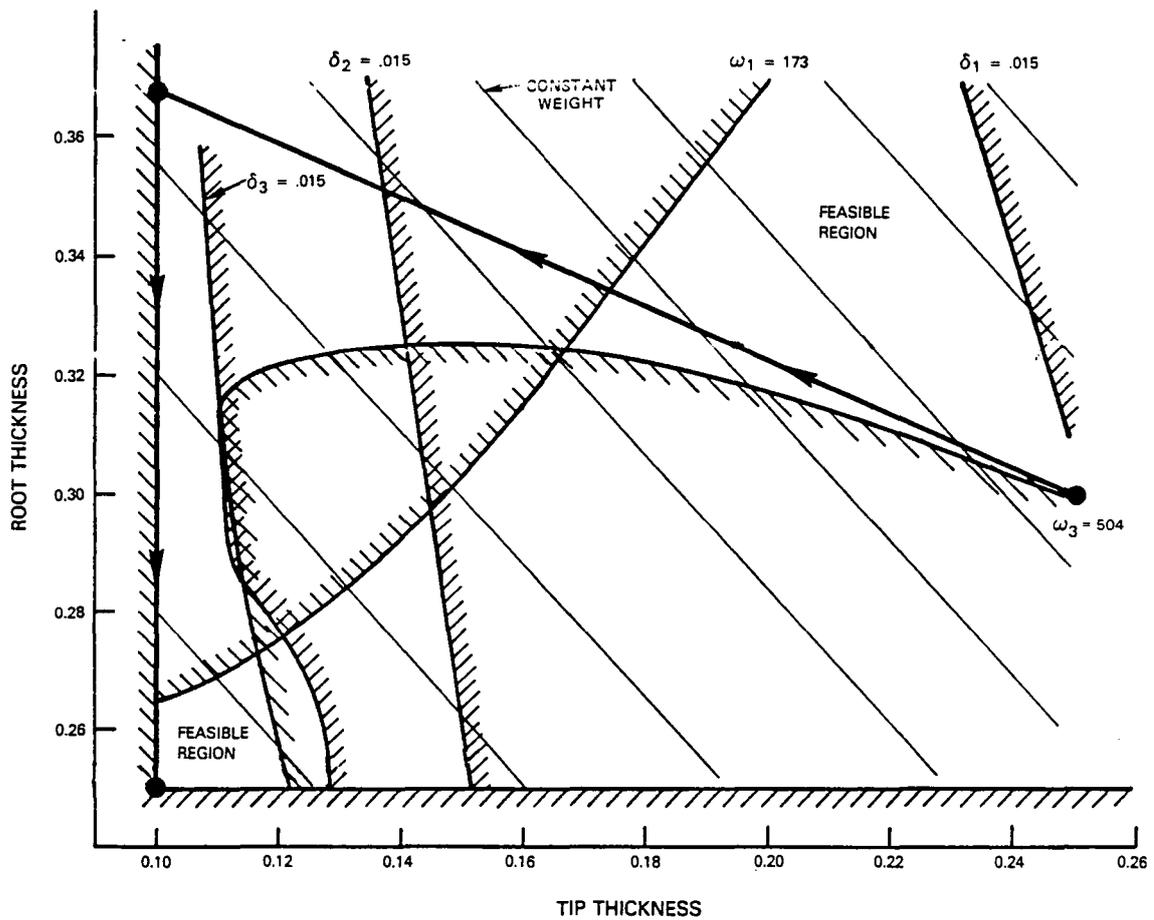


Figure 3.5-9 STAEBL Optimization Procedure with Three Flutter Constraints Converges to Global Optimum

Case 3: Exact and approximate optimization procedures converge to the same local optimum

Again, this example had the same objective function and constraints as Case 1. In this case, a comparison was made between the exact method of feasible directions, and the quadratic Taylor series approximate optimization method. Starting from the same infeasible initial design shown in Figure 3.5-10, both methods converged to the same local optimum design. The exact method required 19 function calls. The quadratic Taylor series had $N_Q = 6$ unknowns and achieved convergence using only six function calls.

	FUNC. CPU CALLS (SEC.)	
EXACT	19	139
APPROX.	6	48

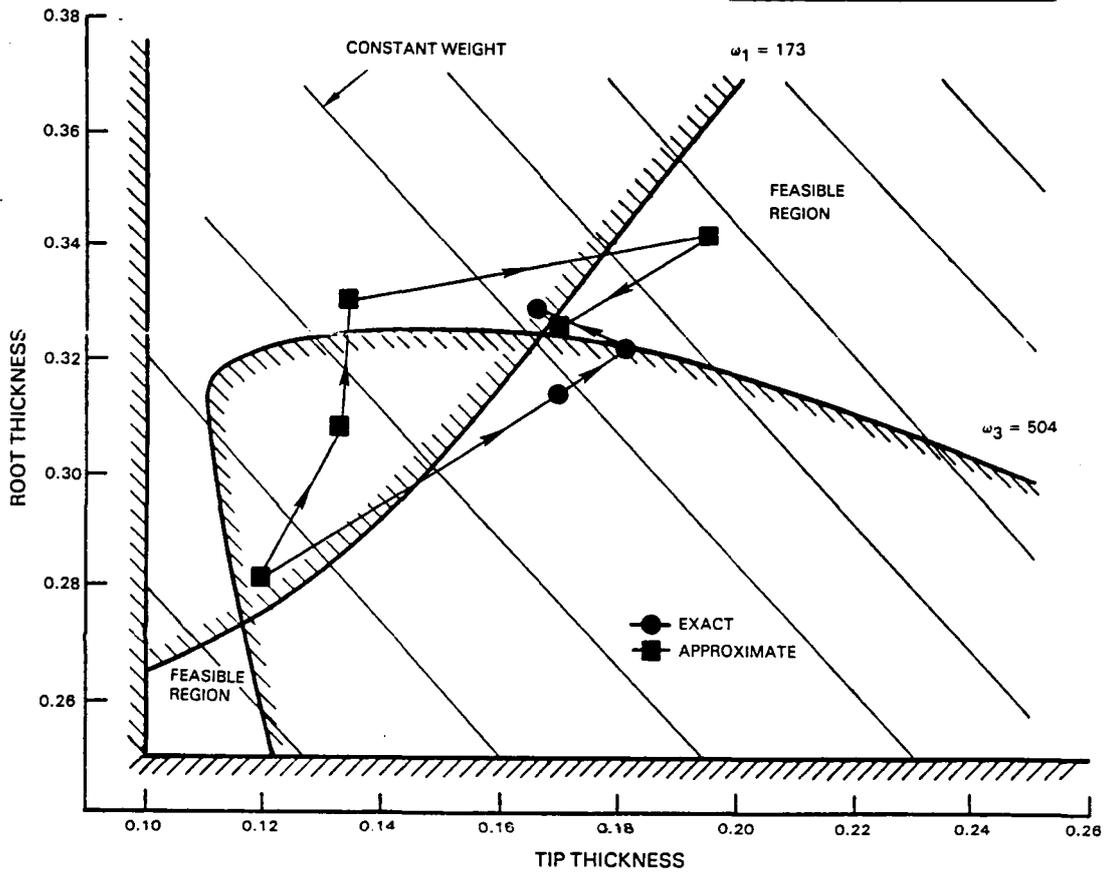


Figure 3.5-10 Exact and Approximate Analyses for STAEBL Procedure Converge to Same Local Optimum

Case 4: Hollow fan blade with titanium skin

Determine (t_r , t_t , d_r , t_{Ti}) that minimizes blade weight + manufacturing cost + maintenance cost:

Subject to:

$$\begin{array}{lll}
 t_r \geq 0 & \delta_j \geq 0 & \text{Root stress} \leq \text{max.} \\
 t_t \geq 0 & 0.025 \leq t_r/b \leq 0.1 & \text{Hole stress} \leq \text{max.} \\
 d_r \geq 2.0 & 0.025 \leq t_t/b \leq 0.1 & \text{Leading edge stress} \leq \text{max.} \\
 t_{Ti} \geq 0.03 & \text{LSBIP} \leq \text{max.} & \text{Trailing edge stress} \leq \text{max.} \\
 & \text{BBIP} \leq \text{max.} &
 \end{array} \quad (55)$$

For this problem, d_r is the distance between blade root and the bottom of the hole, t_{Ti} is the thickness of the titanium skin, b is the blade chord, $\text{LSBIP} = \text{Local Stress Bird Ingestion Parameter}$, and $\text{BBIP} = \text{Bending Bird Ingestion Parameter}$. After beginning with a feasible initial design, up to five constraints were active for some iterates. As shown in Table 3.5-III, convergence was achieved after seven iterations and three constraints were active at the final design.

TABLE 3.5-III

HOLLOW FAN BLADE HAS 3 ACTIVE CONSTRAINTS FOR FINAL DESIGN

Iteration	t_r	t_t	d_r	t_{Ti}	Objective Function	Active Constraints*
0	0.8750	0.3340	9.65	0.1200	1.464	32,33
1	0.8744	0.3388	9.07	0.0974	1.379	1,32,33
2	0.8742	0.3396	9.02	0.0920	1.360	1,7,8,32,33
3	0.8741	0.3397	8.01	0.0922	1.328	1,7,8,32,33
4	0.8988	0.3318	7.04	0.0930	1.304	1,7,8,32,33
5	0.8974	0.3362	5.06	0.0931	1.229	1,7,8,32,33
6	0.9000	0.3358	5.04	0.0920	1.224	1,7,8
7	0.9078	0.3318	4.88	0.0922	1.221	1,8,33

*Active Constraints Notes

- 1: δ_1 (initially violated)
- 7: LSBIP (next to last relevant station)
- 8: LSBIP (last relevant station)
- 32: $t_r/b = 0.1$
- 33: $t_t/b = 0.025$

3.6 REFINED ANALYSIS

NASTRAN was selected for use as the refined analysis method for the STAEBL procedure. It is used regularly to determine steady stresses in solid titanium fan blades for flight cycle life evaluation. A plate element blade model is analyzed in this application. NASTRAN is also used to calculate the vibratory characteristics of composite material fan blades. Equivalent anisotropic material properties are calculated for each finite element using thin laminated plate theory.

Engineering effort in setting up and analyzing solid titanium and composite material blades is minimized through the use of pre-and post-processors. Available processing capabilities include:

- o an airfoil pre-processor which generates a NASTRAN plate model of a blade from the airfoil coordinate descriptions,
- o a laminate pre-processor which calculates the laminate effective stiffness matrices for each finite element and outputs them in a form acceptable to NASTRAN as input data,
- o a NASTRAN module to calculate laminate strains from element stresses,
- o a post-processor to calculate ply stresses from NASTRAN element stresses.

The flight cycle life and vibratory characteristics of the hollow titanium Energy Efficient Engine fan blade were also evaluated using NASTRAN analysis. But separate models of the concave and convex airfoil walls were employed to verify that a sufficient number of ribs were provided. This made the analysis cumbersome and impractical for use in the STAEBL procedure.

It was proposed that hollow blades could be analyzed using a laminated plate model with the central lamina having zero stiffness and density. The Energy Efficient Engine was re-analyzed to substantiate this approach. The airfoil breakup was chosen so that internal ribs are coincident with loci of nodal points. The rib properties are represented by beam elements connecting these nodes. Vibration analysis of the lamination model agrees very well with the more cumbersome original analysis as shown in Figure 3.6-1. The breakup in the region of the airfoil root and solid-to-hollow transition was refined and the centrifugal stresses presented in Figures 3.6-2 and 3.6-3 were obtained. These stresses are consistent with those predicted by the original design analysis.

The procedure for predicting supersonic flutter of fan stages can evaluate the stability of structural modes which are defined by finite plate element analysis. This ability combines the chordwise bending degree of freedom with the flap and twist degrees of freedom included in approximate beam blade analysis. The use of the lamination model for blade modal analysis makes it practical to use the expanded flutter prediction procedure for refined analysis. Results of flutter analyses of the Energy Efficient Engine blade are compared in Figure 3.6-4. The original design flutter analysis reduced the plate element blade mode shapes to equivalent beam blade modes and concluded

that the blade would not flutter. The refined analysis supports this conclusion and provides a more accurate technique to evaluate thinner airfoils which are likely to result from structural tailoring.

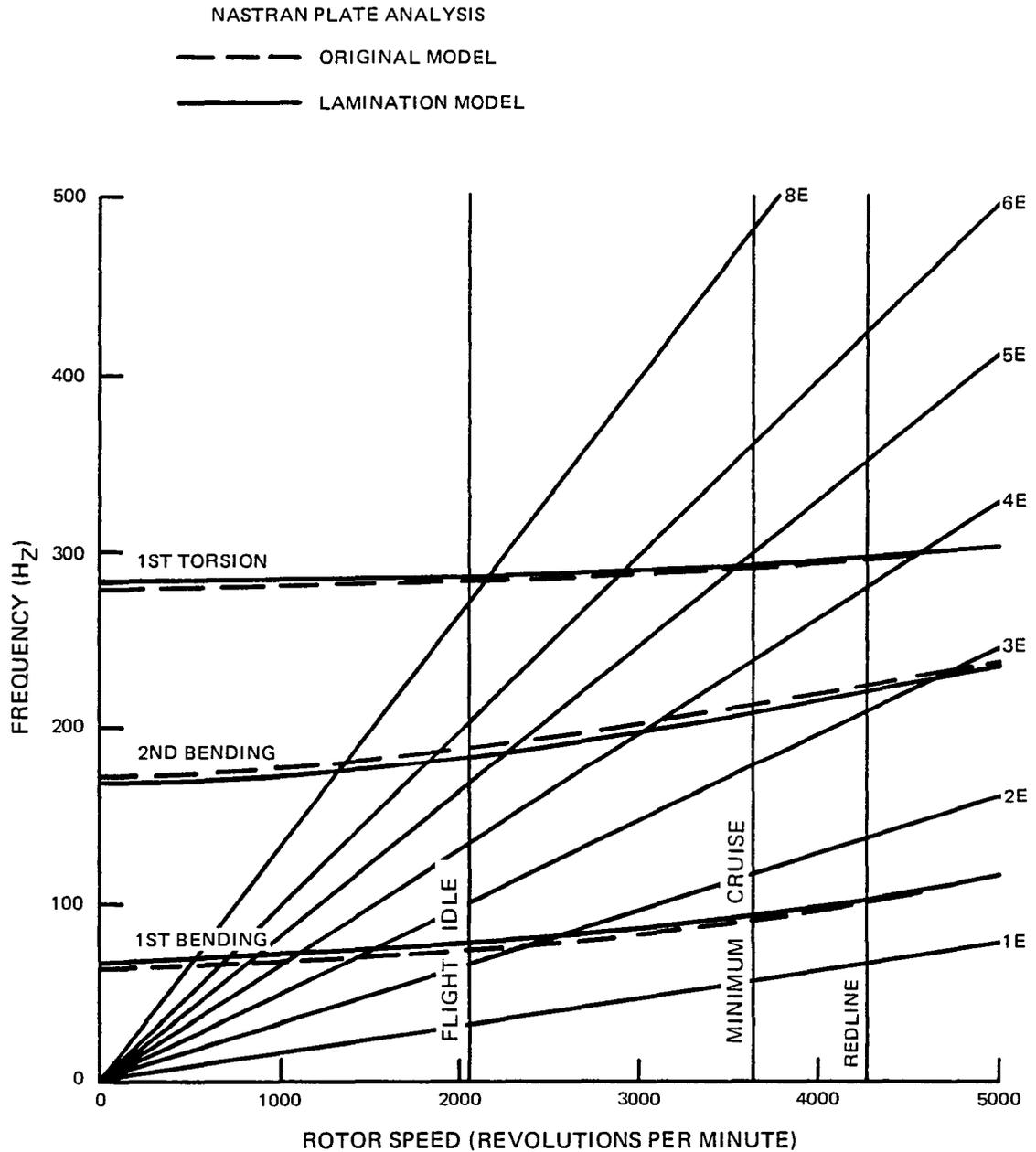


Figure 3.6-1 Refined Analysis of the Energy Efficient Engine Hollow Fan Blade

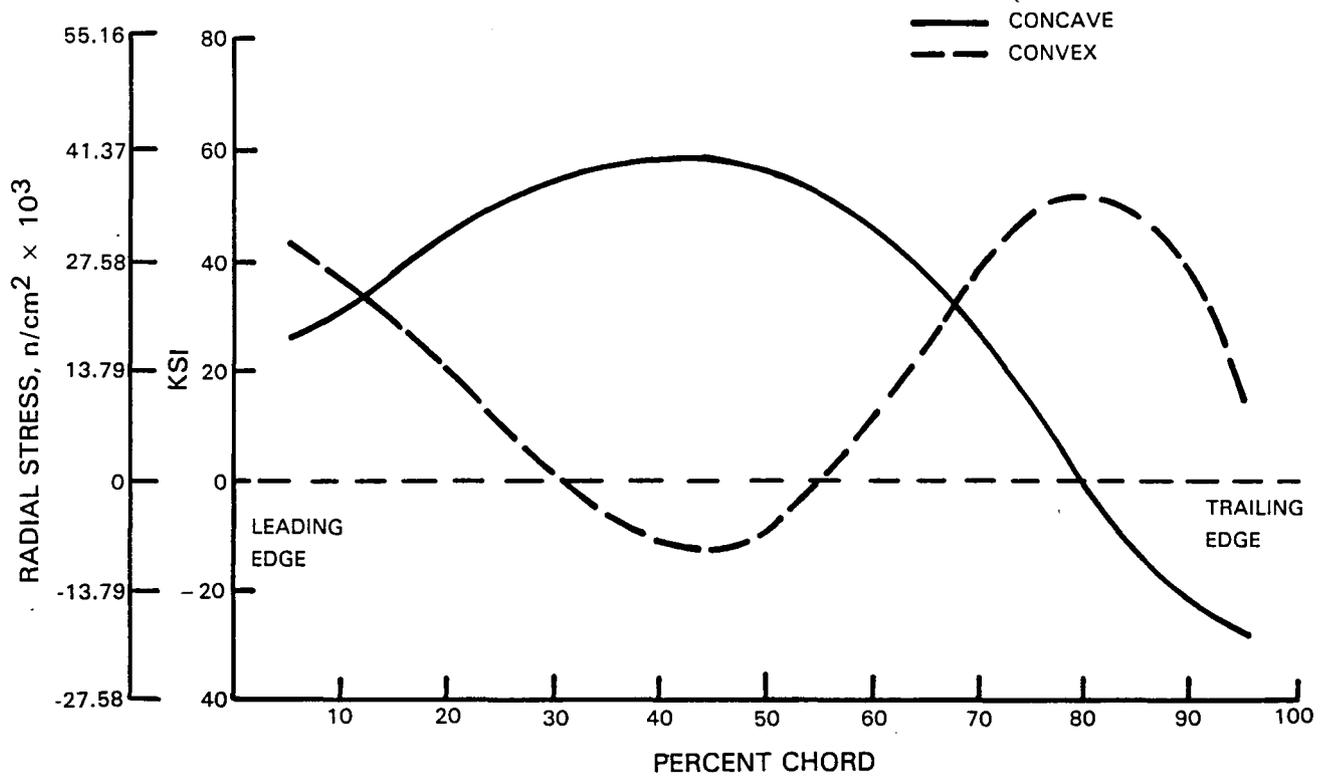


Figure 3.6-2 Energy Efficient Engine Hollow Fan Blade Airfoil Root Stress Predicted by Refined Analysis

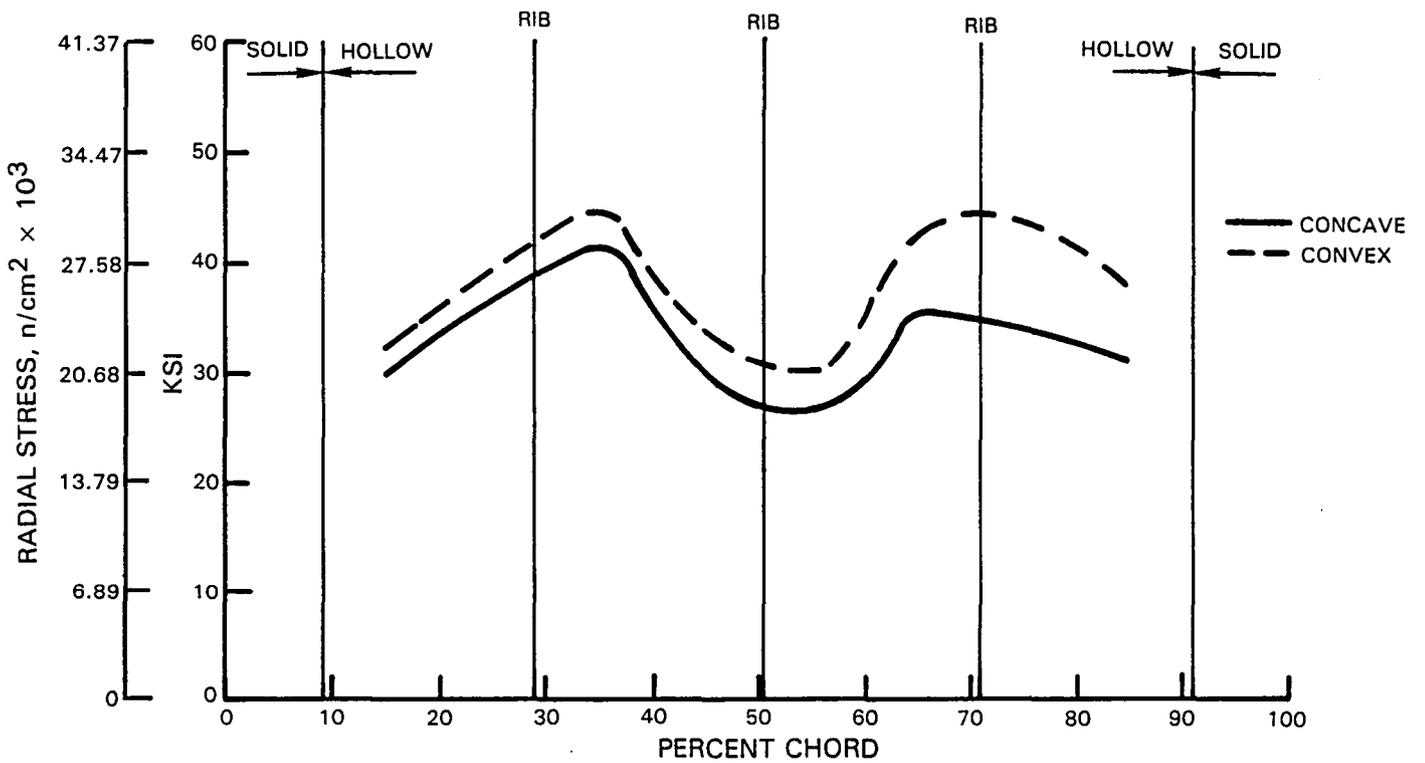


Figure 3.6-3 Energy Efficient Engine Hollow Fan Blade Internal Surface Stress at Solid-to-Hollow Transition Predicted by Refined Analysis

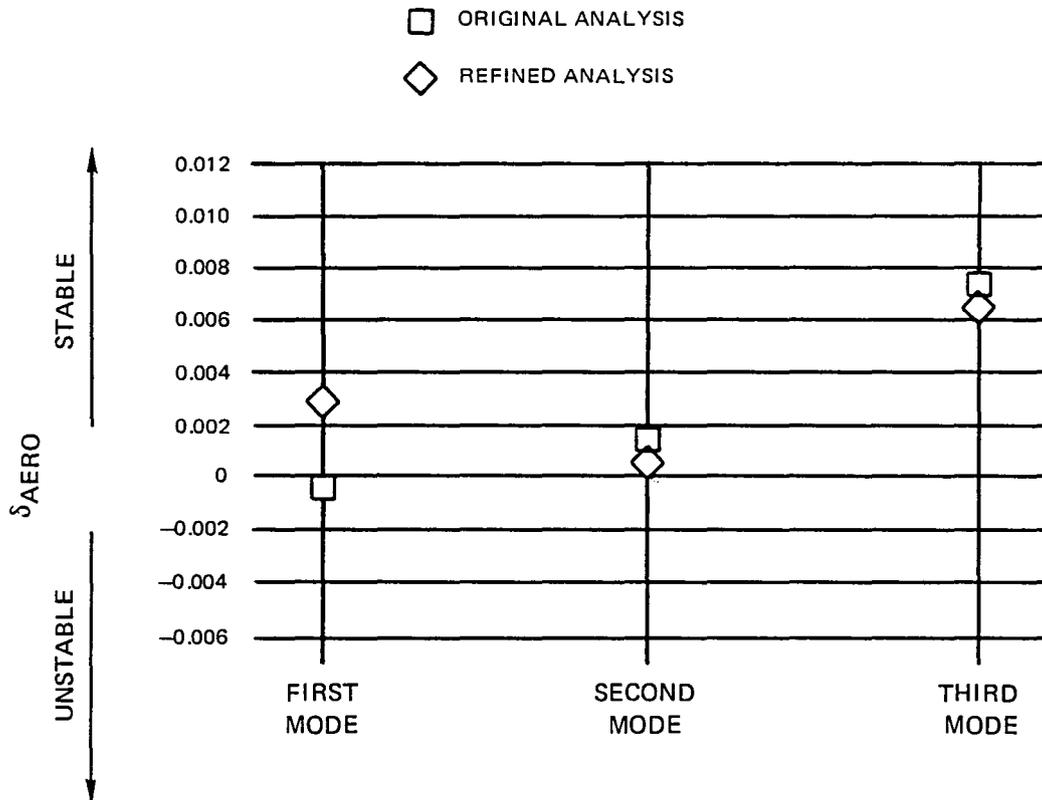


Figure 3.6-4 Refined Analysis of the Energy Efficient Engine Hollow Fan Blade Supports the Conclusion that It Would Not Flutter

4.0 RESULTS

4.1 INTRODUCTION

Structural Tailoring of Engine Blades (STAEBL) was demonstrated by tailoring two shroudless fan blades: 1) a hollow titanium blade with metal matrix composite inlays, and 2) a solid blade made from superhybrid composites. These blades were alternate constructions of the Energy Efficient Engine shroudless fan blade designed by Pratt & Whitney Aircraft as part of NASA Contract NAS3-20646. As a consequence of demonstration blade selection, several features of the Energy Efficient Engine were taken into account in the STAEBL demonstration. Those features and other design considerations will be presented in Section 4.2 while a description of the actual tailored blades will be presented in Section 4.3.

4.2 DEMONSTRATION CONSIDERATIONS

4.2.1 Aerodynamic Design

Engine configuration-dependent parameters of the Energy Efficient Engine fan component design were held constant while others were allowed to vary. Those parameters that were held constant include: the airflow through the fan component of the Energy Efficient Engine, which is 622.7 kg/sec (1372.8 lbm/sec); the tip speed of 456 m/sec (1496 ft/sec); and the average pressure ratio of 1.7. The hub/tip radius ratio of 0.34, the tip/root chordal taper of 1.46, and associated airfoil stagger and camber angles were also held constant during structural tailoring.

The basic blade chord was allowed to vary and the number of blades was changed inversely with chord to maintain constant aerodynamic gap/chord ratio at any radial location. The spanwise distribution of airfoil section maximum thickness was allowed to vary but the ratio of thicknesses at any two fraction of chord positions was held approximately constant. Maximum thickness was never allowed to exceed ten percent of local chord.

4.2.2 Objective Function

The procedure used to establish an objective function for the STAEBL demonstration was modeled after the economic assessment procedure used to guide the design of the Energy Efficient Engine. The Energy Efficient Engine design process was guided by an economic performance assessment of a study airplane defined in the Component Development and Integration Program phase of the contract. The overall airplane characteristics are shown in Table 4.2-I. The aircraft is designed for the full specified payload and range, but the economic analysis is conducted for the typical mission payload and range.

The economic analysis evaluates the effect of changes in engine weight, maintenance cost and first cost against the changes in the aircraft takeoff gross weight and fuel burned to assess the economic effect on the airline. The basis for this analysis is a well developed trade factor technique derived from consideration of airplane aerodynamics, flight mechanisms, propulsion system integration, and weight estimation. The changes in airplane takeoff

gross weight reflect a "rubber" airplane analysis, i.e. improvements to the engine configuration will result in further improvements to the airplane configuration. For example, a concept which reduces engine weight will result in a fuel savings which, in turn, further reduces aircraft weight and aircraft structural component weight, permitting reductions in wing size and engine thrust requirements. Consequently, the initial engine weight benefit "snowballs" in its impact on the aircraft benefit.

TABLE 4.2-I

ENERGY EFFICIENT ENGINE STUDY AIRPLANE CHARACTERISTICS

	<u>Domestic Trijet</u>
Number of Engines	3
Range - kilometers (nautical miles)	
Design mission	5550 (3000)
Typical mission	1295 (700)
Passengers	
Design mission	400
Typical mission	55% load factor
Design takeoff gross weight - kilograms (pounds)	231,000 (510,000)
Cruise Mach number	0.80
Initial Cruise Altitude - meters (feet)	10,700 (35,000)
Takeoff field length - meters (feet)	2440 (8,000)

The life cycle ownership costs determined in this analysis are expressed as direct operating costs plus interest (DOC + I). A trade factor technique derived from considerations of total airline economics provides the basis for this analysis, and includes crew cost, fuel cost, airframe and engine depreciation, airframe and engine maintenance cost, insurance cost, and overhead cost. These trade factors are applied to the specific engine and airplane for which each engine change has been determined. Trade factors for changes in engine weight, maintenance cost, and first cost are applied independently to determine the effect of each engine change on a given economic parameter. Individual effects are then combined to evaluate the total effect of the advanced concept on that parameter. DOC + I is an extension of DOC in that it includes the "cost of money". In other words, it includes an expected return to the airline for their investment in the aircraft/engine system. DOC + I is an appropriate substitution for ROI (return on investment) and includes all of the engine related terms in ROI, and is, therefore, an appropriate parameter for evaluating the effect of engine changes on an airline's economics.

The ground rules for the airline economic model are shown in Table 4.2-II for the Energy Efficient Engine. The 15 percent cost of capital shown is the "interest" in the parameter "DOC + Interest" (DOC + I).

TABLE 4.2-II

ENERGY EFFICIENT ENGINE AIRLINE ECONOMIC MODEL

- o 1977 Dollars
- o \$0.105/Liter (\$0.40/Gallon) Domestic Fuel Cost
- o 0.5% Per Year Insurance
- o Spares - 5% Airframe, 30% Engine
- o Maintenance - Labor Rate = \$9.70/Hr, Burden = 200%
- o Airplane Price - P&WA Equation
- o Depreciation - 15 Year Straight Line to 10% Residual
- o Non-Revenue Flying - 2% Factor on Fuel and Maintenance
- o Ground Time - 15 minutes (Domestic)
- 20 minutes (International)
- o Cost of Capital = 15%

The resulting function is:

$$\begin{aligned} \Delta(\% \text{ DOC} + \text{I}) = & 1.14 \times \left(\frac{\Delta \text{ engine wt (kg)}}{1000} \right) \\ & + 0.54 \times \left(\frac{\Delta \text{ engine cost (\$)}}{\$100,000} \right) \\ & + 0.80 \times \left(\frac{\Delta \text{ engine maint. cost (\$/EFH)}}{\$10} \right) \end{aligned}$$

This function was redefined in terms of blade design variables to provide the objective function for the STAEBL procedure demonstration.

4.2.2.1 Engine Weight

Individual airfoil weight is an output of the subroutine which was assembled to modify existing approximate analyses for application to composite blades and it is a function of all design variables. Total airfoil weight is the product of individual airfoil weight and number of blades, which is inversely dependent upon the blade chord design variable. Disk and containment case design requirements relate to individual and/or total airfoil weight.

A study was conducted to evaluate these relationships using preliminary design procedures which are regularly applied in estimating engine weight. Three different blade constructions were assumed and blade chord was varied over a range of relevant aspect ratios. Individual foil weights and fan system weight were calculated. Cross-plotting the results generated the unexpected conclusion that, within engineering accuracy, fan system weight is a simple function of individual airfoil weight. This function is depicted in Figure 4.2-1.

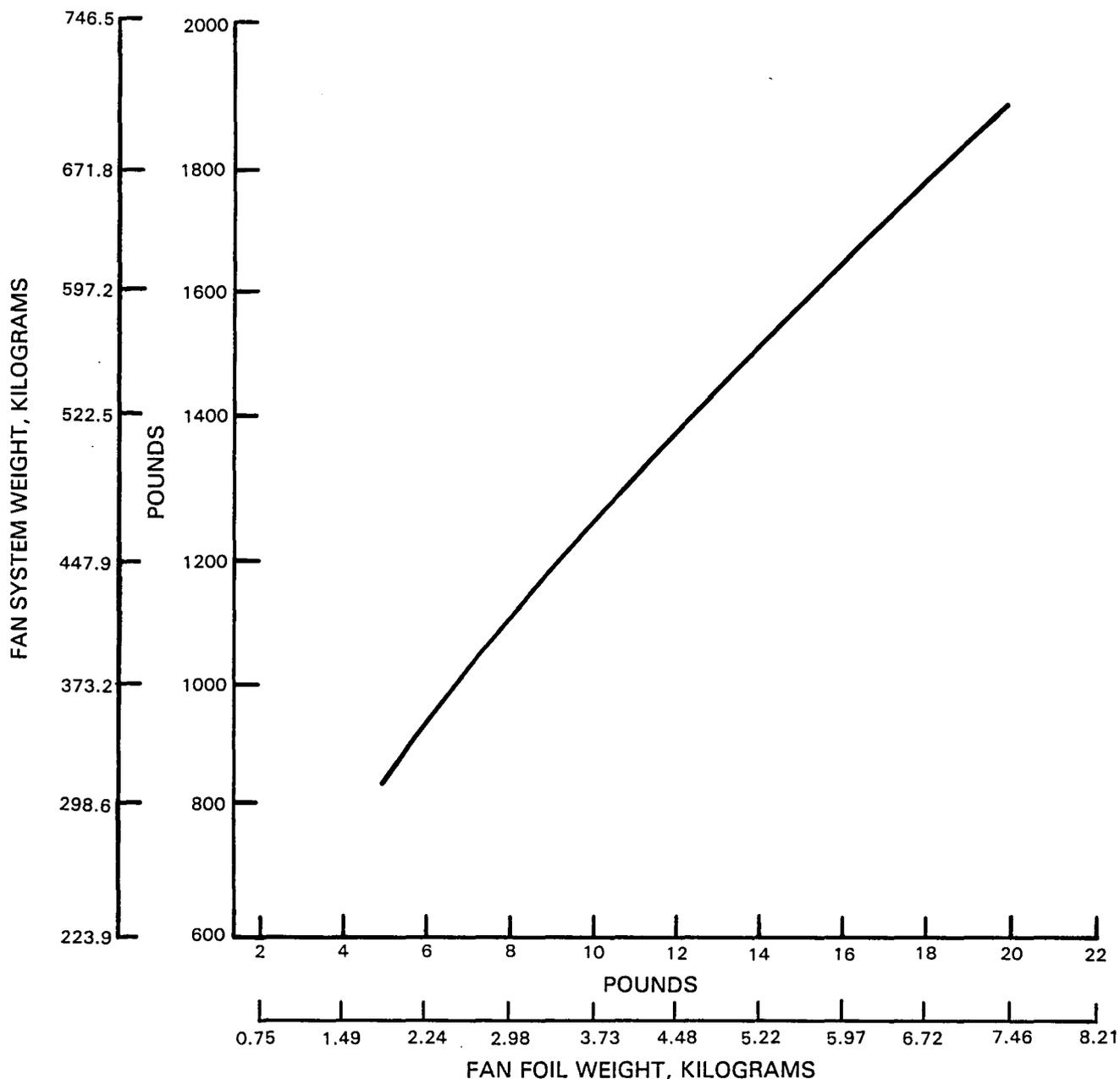


Figure 4.2-1 Fan System Weight Is a Simple Function of Individual Airfoil Weight

4.2.2.2 Engine Cost

Information from reference 3 was used as a basis for estimating costs of individual fan blades. The hollow blade is made from laminations of titanium and borsic-titanium sheet with hollow cavities produced using leachable iron cores. The stacked laminates are canned, hiped and isothermally forged to shape. The superhybrid blade is made from plies of graphite-epoxy, boron-aluminum, and titanium with adhesive called for metal ply bonding. Stacked plies are vacuum debulked and molded. Materials cost depends on amount of each component material which is related to the design variables by the composite blade approximate analysis subroutine. Labor cost depends on blade size as indicated by the design variables root chord and root thickness. Total blade cost is the product of individual blade cost and the number of blades which is inversely proportional to the blade chord design variable. The change in engine cost for changes in design variables is the change in total blade cost plus an experience-based assessment of costs of related structures which reduces to cost per unit engine length multiplied by change in blade chord.

4.2.2.3 Maintenance Cost

Engine maintenance histories show that the dominant factor in fan maintenance cost is the number of blades which must be discarded after an ingestion event because they are damaged beyond repair. Service experience provides a definition of the frequency of major ingestion events, the percentage of blades damaged by an event and the solid titanium blade repair/scrap ratio. A hollow blade is expected to have a lower ratio because damage in or near the cavity is not repairable. The controlling parameter is expected to be the design variable distance from the airfoil leading edge to the forward boundary of the cavity. Experience and judgement have been applied to generate the definition, shown in Figure 4.2-2, of hollow blade scrap life from known end points. Life is increased in proportion with swept flowpath area when the cavity inner bound is outboard of the airfoil root. Superhybrid blades are assumed to be unrepairable. Total maintenance cost per flight hour is equal to blade set fabrication cost divided by scrap life.

4.2.3 Constraints on Behavior Variables

Constraints were imposed during tailoring to ensure that the optimized designs would not be prone to failure from high frequency fatigue, would endure repeated cyclic acceleration to takeoff power, and could ingest seagulls without experiencing damage which would cause engine shutdown. These constraints were consistent with criteria observed during the Energy Efficient Engine fan component design.

4.2.3.1 High Frequency Fatigue

High frequency fatigue in resonant vibration was avoided by tuning natural frequencies so they did not coincide with the frequencies of strong excitations. In the Energy Efficient Engine fan component design, the critical excitation frequencies were recognized to be the first through fourth integer order multiples of rotor speed when the engine is operating at any power between minimum cruise and maximum takeoff. High frequency fatigue in flutter

is avoided by deriving a design which has positive predicted aerodynamic damping in any natural mode of vibration at airflow conditions corresponding to maximum takeoff power. In the Energy Efficient Engine fan component design, minimum stability for any feasible interblade phase angle was evaluated for vibration in each of the first three modes of individual blade vibration.

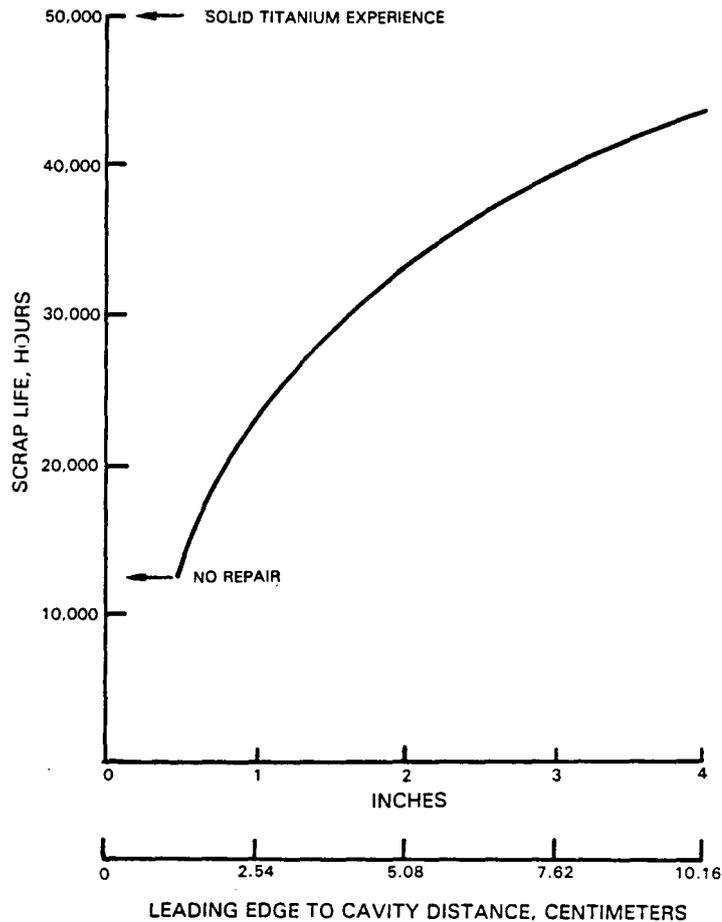


Figure 4.2-2 Full Span Hollow Fan Blade Life Is Derived from Solid Blade Experience Limits

4.2.3.2 Flight Cycle Fatigue

Flight cycle fatigue is evaluated by assuming that operating stresses cycle between zero at shutdown and maximum at standard day takeoff power. In the Energy Efficient Engine fan component design, the operating stress was assumed to be that imposed by centrifugal force on a radially stacked blade because bending caused by tilt is equal and opposite to bending caused by air load at the critical root section.

4.2.3.3 Bird Ingestion

Two types of damage are recognized in designing a fan blade to tolerate bird ingestion: 1) loss of an excessive quantity of local leading edge material, or 2) fracture of the airfoil root. Both of these types of failures of the Energy Efficient Engine fan blade were evaluated using an ingestion stress parameter based on linear elastic analysis calibrated by application to many ingestion experiences.

4.3 STAEBL PROCEDURE DEMONSTRATION

4.3.1 Reference Blade

The Energy Efficient Engine shroudless fan blade design provides a logical baseline for evaluating the blade designs defined by the tailoring demonstration. It is an all titanium blade which is hollow over the outer two-thirds of the airfoil span. The cavity wall thickness is tapered in the spanwise and chordwise directions. A dynamically equivalent constant wall thickness hollow titanium blade with the same external shape and cavity planform location was derived. It provided a reference design consistent with the composite material blade designs which could be defined by the parameters that were varied in the demonstration.

The reference blade dynamic characteristics are equivalent to those of the Energy Efficient Engine shroudless fan blade design. Therefore, the dynamic resonance and flutter constraints observed in designing that blade were directly applicable to tailoring demonstration blades.

The reference blade stress levels are similar to, but not identical with, those in the Energy Efficient Engine blade. The demonstration was made most meaningful by setting limits equals to reference blade levels multiplied by Energy Efficient Engine blade allowable-to-actual ratios. Realistic stress concentrations and local material conditions are automatically covered by this approach.

The local bird ingestion stress parameter was limited to the Energy Efficient Engine blade design allowable level. The root dynamic stress response to bird ingestion was a new uncalibrated analysis. The limit was set to be in proportion with the Energy Efficient Engine blade root stress parameter margin of safety. This limit is much higher than material static strength.

The Tsai-Wu parameter for each material is calculated from ratios of directional stress to directional strength. Therefore, a search for the maximum value found in any material yields a single critical level which is compared with a material independent limit level. The local bird ingestion stress parameter is evaluated in a consistent fashion by ratioing each layer's contribution in proportion with chordwise directional strength.

Preliminary studies had shown that resonance constraints, which create disjoint feasible regions, could lead to tailoring a locally optimum blade design which is not globally optimum. Therefore, the initial optimization of each of the two types of blades was conducted without resonance constraints

and then, starting with the partially tailored design, was finalized with resonance constraints added. Both of the resultant tailored blades had the same basic resonance characteristics as the reference hollow titanium blade (Figure 4.3-1), increasing confidence that global optimums were achieved.

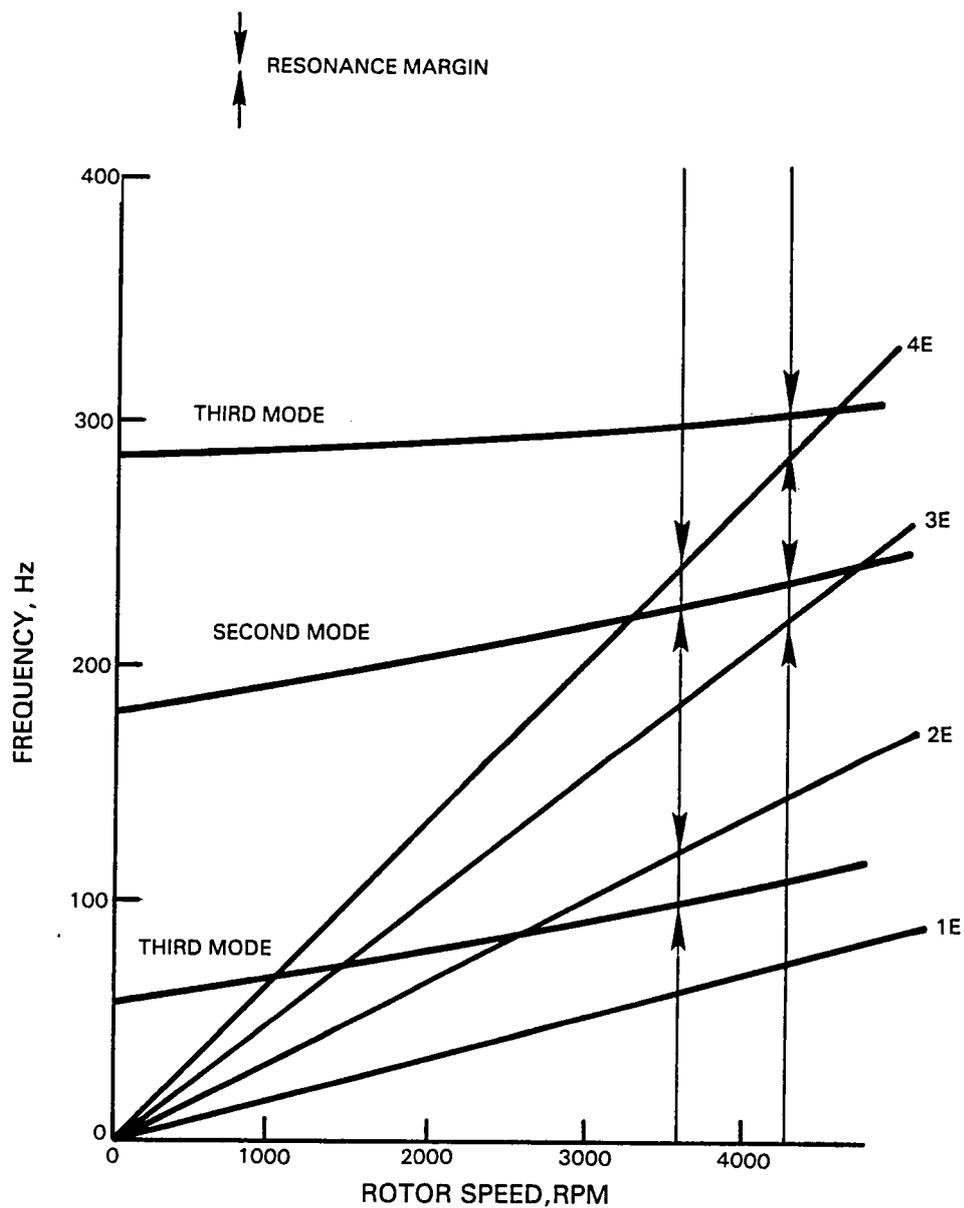


Figure 4.3-1 Reference Blade Resonance Diagram

4.3.2 Tailored Blades

Each tailoring demonstration started with a configuration that was similar to the reference blade (2.5 aspect ratio based on root chord and average length). Each demonstration consisted of two passes through automated exact optimization and subsequent separate refined analyses. During the automated optimization, seventy-four percent of the central processing unit time was spent in approximate structural analysis and twenty-six percent of the time was spent in unsteady aerodynamic (flutter) analysis. The time spent on optimization calculations was negligible. The four optimization efforts successfully identified feasible designs as evaluated by approximate analyses. The first refined analyses generated correction factors which were applied to the approximate analyses for reoptimization. The particular demonstrations are described in the following paragraphs.

4.3.2.1 Hollow Blade with Composite Inlay

In Table 4.3-I, data describing the tailoring of a hollow titanium blade with a borsic titanium inlay in the area of the cavity are compared with the reference blade. Thirteen geometric quantities were varied in this demonstration. Chord at every station was changed in proportion with root chord. The properties which were assumed for borsic-titanium are compared with titanium in Table 4.3-II.

TABLE 4.3-I

HOLLOW BLADE WITH COMPOSITE INLAY

	<u>Reference Blade</u>	<u>Limits</u>	<u>First Tailoring</u>	<u>Second Tailoring</u>
<u>Design Variables</u>				
Root Chord - cm (inches)	23.2 (9.12)	-	16.9 (6.64)	19.0 (7.46)
Thickness/Chord				
Root	0.096	↑	0.077	0.084
25%	0.075	0.025 ≤	0.089	0.092
50%	0.055	t/b ≤ 0.100	0.084	0.045
75%	0.033	↓	0.038	0.025
Tip	0.025		0.033	0.028
<u>Cavity Boundaries - cm (inches)</u>				
From Leading Edge	2.20 (1.00)	≥ 1.27 (0.5)	3.12 (1.23)	1.37 (0.54)
From Trailing Edge	2.20 (1.00)	≥ 1.27 (0.5)	2.48 (0.98)	3.48 (1.37)
From Root	25.2 (9.90)	≥ 5.46 (2.15)	12.97 (5.10)	7.52 (2.96)
From Tip	0.635 (0.25)	≥ 0.635 (0.25)	0.635 (0.25)	0.635 (0.25)

TABLE 4.3-I (Continued)

	<u>Reference Blade</u>	<u>Limits</u>	<u>First Tailoring</u>	<u>Second Tailoring</u>
Cavity Wall - cm (inches)				
Titanium Skin	0.305 (0.120)	≥ 0.076 (0.030)	0.076 (0.030)	0.076 (0.030)
Composite	-	-	0.086 (0.034)	0.160 (0.063)
Borsic Fiber Angle - rad	-	-	-0.012	-0.013
<u>Blade Weight</u> - kilograms (pounds)				
	8.7 (19.2)	-	3.32 (7.3)	4.23 (9.3)
<u>Objective Function</u> - Δ (%DOC+I)				
Engine Weight			-0.42	-0.33
Engine Cost			-0.14	-0.15
Maintenance Cost			-0.09	+0.03
Total			<u>-0.65</u>	<u>-0.45</u>
<u>Constraints</u>				
Resonance Margin - $\Delta f_n/f_n$				
First Mode 2E	0.08	≥ 0.05	0.34	0.28
Second Mode 3E	0.05	≥ 0.05	-0.02	-0.03
Second Mode 4E	0.08	≥ 0.05	0.16	0.19
Third Mode 4E	0.05	≥ 0.05	0.01	0.06
Flutter - Log Decrement				
First Mode	0.006	≥ 0	0.004	0.005
Second Mode	0.001	≥ 0	0.011	0.001
Third Mode	0.007	≥ 0	0.001	0.002
Bird Ingestion				
Local Stress Parameter	52.0	≤ 80.0	66.0	48.0
Root Tsai-Wu	0.9	≤ 10.0	n.g.	2.5
Steady Stress - Tsai-Wu				
Root Edge	0.20	≤ 0.22	0.32	0.08
Root Local	0.25	≤ 0.29	0.32	0.21
Cavity Local	0.26	≤ 0.31	0.32	0.08

TABLE 4.3-II
MATERIALS PROPERTIES
(Divided by Titanium Value)

	<u>Borsic-Titanium</u>	<u>Adhesively Laminated Boron-Aluminum</u>	<u>Graphite Epoxy</u>
Elastic Modulus			
Fiber Direction	2.06	1.73	1.15
Normal Direction	1.80	1.11	0.10
Shear	3.70	1.41	0.14
Density	0.81	0.55	0.35
Strength			
Fiber Tension			
Direction	1.54	1.14	1.45
Fiber Compression			
Direction	1.54	1.64	1.45
Normal Tension			
Direction	0.45	0.12	0.07
Normal Compression			
Direction	0.45	0.26	0.23
Shear	0.47	0.20	0.16

Thickness/chord distribution was defined by straight lines connecting independent levels at five stations. A lower limit was imposed to prevent chordwise bending flutter and an upper limit to avoid aerodynamic performance penalties. The cavity quadrilateral planform boundaries were parallel to adjacent airfoil planform extremities and the cavity walls were laminates of titanium and borsic-titanium. The titanium was of uniform thickness and was located at the airfoil surface. The borsic-titanium was also of uniform thickness and it was located between the titanium surface and the internal cavity. Design variables were the distances from the cavity boundaries to the airfoil extremities and the thickness of the individual materials which comprised the cavity wall. Limits were imposed based on anticipated maintenance penalties in excess of those recognized by the objective function (the root limit ensured that the supporting attachment would be solid titanium). Borsic fiber angle was the final variable.

The first tailoring converged in thirteen iterations. The root Tsai-Wu due to bird ingestion was observed to have varied in an illogical manner which was traced to an error in the attachment section scaling with root chord and thickness. No other parameters were seriously effected. Refined analysis results, which were presented in Table 4.3-I, showed that approximate analysis error had permitted violation of resonance limits, that the second mode frequency be at least five percent higher than three times rotor speed and that the third mode frequency be at least five percent higher than four times rotor speed, both at maximum speed. Root steady stress limits were also violated at the edge and away from the edge.

The attachment scaling logic was corrected and correction factors from the refined analysis of the first tailored blade were applied to the approximate analyses predicted frequencies and steady stresses. A second tailoring, starting with the configuration defined by the first tailoring, converged in ten iterations. Refined analysis showed that the use of correction factors had eliminated all but one limit violation. Comparison of approximate and refined analysis predicted frequencies, Table 4.3-III, showed that the approximate analysis error in the second mode frequency prediction was erratic. Tailoring was terminated since there was no reason to expect that continued iteration would lead to convergence of corrected approximate and refined frequency analyses or to a more realistic design. Central processing unit time averaged 5.9 minutes per iteration.

TABLE 4.3-III

HOLLOW COMPOSITE BLADE WITH COMPOSITE INLAY NATURAL FREQUENCY
(Hertz)

<u>Mode</u>	<u>First Tailoring</u>		<u>Second Tailoring</u>	
	<u>Approximate Analysis</u>	<u>Refined Analysis</u>	<u>Approximate Analysis*</u>	<u>Refined Analysis</u>
1	100	88	106	94
2	187	208	205	208
3	300	287	312	295

*Without correction

The weight of the individual tailored blade is less than half the reference blade weight and the optimized objective function indicates that total weight and cost are both reduced even though more blades are required to hold aerodynamic gap/chord. Maintenance cost penalty is minimal. These benefits were achieved by the use of composite material, chord reduction and extension of the cavity toward the airfoil root. The limiting constraints appear to be resonance and flutter fine tuned by the location of the transition from the thick root to the thin tip and by the chordwise location of the cavity.

4.3.2.2 Superhybrid Composite Blade

Table 4.3-IV presents data describing the tailoring of a solid blade made of superhybrid composite material. Eleven quantities were varied in this demonstration. Chord and thickness were defined as they were in the hollow blade optimization. The blade was sheathed with a uniform thickness titanium skin. The internal construction consisted of a uniform thickness central titanium ply with the balance being composite material, boron-aluminum external to graphite epoxy. The titanium thicknesses and the constant fraction of composite which was boron-aluminum were design variables. A limit was imposed on the titanium skin thickness to avoid a maintenance penalty in

excess of that recognized by the objective function. Boron and graphite fiber angles were also varied. The properties which were assumed for the composite materials were compared with titanium in Table 4.3-II.

TABLE 4.3-IV
SUPERHYBRID COMPOSITE BLADE

<u>Design Variables</u>	<u>Limits</u>	<u>First Tailoring</u>	<u>Second Tailoring</u>
Root Chord - cm (inches)	- -	19.12 (7.53)	22.2 (8.32)
Thickness/Chord			
Root	↑	0.100	0.097
25%	0.025 ≤	0.099	0.100
50%	t/b ≤ 0.100	0.049	0.053
75%		0.026	0.025
Tip	↓	0.026	0.025
Titanium Thickness - cm (inches)			
Skin	≥ 0.076 (0.030)	0.147 (0.058)	0.142 (0.056)
Central	- -	0 0	0.066 (0.026)
B/Al/Composite	-	0.828	0.835
Boron Fiber Angle - radians	-	0.002	0.002
Graphite Fiber Angle - radians	-	0.093	0.093
<u>Blade Weight</u> - kilograms (pounds)	- -	4.13 (9.1)	5.49 (12.1)
<u>Objective Function</u> - Δ(%DOC+I)			
Engine Weight		-0.34	-0.23
Engine Cost		-0.23	-0.18
Maintenance Cost			+0.05
Total		<u>-0.57</u>	<u>-0.36</u>
<u>Constraints</u> - Δfn/fn			
Resonance Margin			
First Mode 2E	0.05	0.19	0.15
Second Mode 3E	0.05	0.05	0.12
Second Mode 4E	0.05	0.15	0.09
Third Mode 4E	0.05	0.03	0.12

TABLE 4.3-IV (Continued)

	<u>Limits</u>	<u>First Tailoring</u>	<u>Second Tailoring</u>
Flutter - Log Decrement			
First Mode	≥ 0	-0.006	0
Second Mode	≥ 0	0.004	0.001
Third Mode	≥ 0	0.003	0.009
Bird Ingestion			
Local Stress Parameter	≤ 80.0	77.0	80.0
Root Tsai-Wu	≤ 10.0	2.6	1.7
Steady Stress - Tasi-Wu			
Root Edge	≤ 0.22	0.16	0.12
Root Local	≤ 0.29	0.22	0.24

The rectangular section attachment used to simulate hollow blade support structure flexibility did not have a logical material composition when applied to the superhybrid blade so the airfoil root was extended inward an equivalent distance. This simulated the transition to a curved attachment which has been used in hybrid construction to avoid composite ply inner ends in highly stressed regions.

A review of tailoring progress after five iterations exposed undue influence of the leading edge stress limit applied to the forward extremity of the composite core material. Since this limit is based on the particular characteristics of titanium material, it was subsequently applied only to the titanium skin. The local limit was applied to the composite material at any location.

The first tailoring converged after ten additional iterations. The refined analysis results, which were presented in Table 4.3-IV, showed that approximate analysis error had permitted violation of the third mode resonance limit and the first mode flutter stability. A second tailoring with correction factors converged in thirteen iterations. The differences between the approximate and refined analysis predicted frequencies, Table 4.3-V, were less than they were in the hollow blade demonstration and the refined analysis showed all limits to be satisfied. Central processing unit time averaged 5.1 minutes per iteration.

The tailored superhybrid composite blade weight and optimized objective function are not as beneficial as those provided by the tailored hollow blade with composite inlays, but they represent considerable improvement over the reference blade. Chord reduction and the boron aluminum portion of the composite material core appear to have provided the benefits achieved. Flutter stability appears to be the dominant constraint. Local bird ingestion capacity established the thickness of the titanium skin.

TABLE 4.3-V
 SUPERHYBRID COMPOSITE BLADE NATURAL FREQUENCIES
 (Hertz)

<u>Mode</u>	<u>First Tailoring</u>		<u>Second Tailoring</u>	
	<u>Approximate Analysis</u>	<u>Refined Analysis</u>	<u>Approximate Analysis*</u>	<u>Refined Analysis</u>
1	106	104	113	108
2	229	218	247	230
3	302	289	338	314

*Uncorrected

This Page Intentionally Left Blank

5.0 CONCLUSIONS AND RECOMMENDATIONS

Mathematical optimization applied to shroudless fan blade structural tailoring has been demonstrated to be a very powerful automated design procedure. It provides the capacity to simultaneously evaluate many design variables to optimize a comprehensive objective function while observing numerous design constraints. The fan blade application of this study is relatively novel because dynamic design constraints preponderate and, since critical excitations can be avoided by making structural frequencies higher or lower, disjoint feasible regions are a practical possibility. This did not introduce limiting problems.

The composite blade tailoring application demonstrated the capacity of the STAEBL procedure to select values for a large number of design variables because the fabricated internal constructions could be changed in many ways without affecting aerodynamic performance. But checkout studies also demonstrated that the procedure is a useful tool for homogeneous material blade tailoring. The various natural modes of vibration are sensitive to spanwise distribution of airfoil thickness to a level of refinement that can only be defined by several variables. Even the most experienced design analyst would only be able to find an approximation of the best distribution and, in some instances, he might not find any acceptable distribution when the feasible region is small.

Unresolved discrepancies between the final corrected approximate analysis of one of the demonstration blades and the subsequent refined analysis exposed an apparent limitation of the STAEBL procedure. It is suspected that the approximations which are inherent in simple beam theory limit accuracy when the procedure is applied to a composite material blade and, to a lesser degree, when it is applied to a shroudless fan blade of any material. But the fact that a finite beam element approach was used to derive a practical composite material blade analysis indicates that a special purpose plate element blade analysis could also be used in a tailoring procedure. It would eliminate the problem by improving consistency between the approximate and refined analyses.

Spanwise thickness distribution has been shown to be an important characteristic of an airfoil dynamic design. Chordwise distribution within aerodynamically allowable limits may also be significant but it can not be pursued effectively because of the inexact nature of the stress parameter used to evaluate local edge damage caused by bird ingestion. An existing nonlinear finite element analysis has the desired technical capability but it is too cumbersome for inclusion in a tailoring procedure. It is feasible that the critical features of the local response simulation could be incorporated in a fast linear analysis. It is recommended that the STAEBL procedure be revised to include a special purpose plate element for airfoil stress and vibration analyses. It is also recommended that the STAEBL procedure be revised to include another special purpose plate element for analysis of linearized large deflection local edge response to foreign object damage impact.

This Page Intentionally Left Blank

APPENDIX A

BLADE OPTIMIZATION PROGRAM ORGANIZATION

The COPES/CONMIN program forms the basis for the blade optimization program. COPES/CONMIN, through subroutine ANALIZ, calls the blade vibration, flutter, and foreign object damage analyses for evaluation of the constraints and the objective function of the design vector.

Table A-I shows a tabulation of the 74 subroutines developed or acquired under this contract, along with a list of other routines called by each routine, and a brief statement of the function of each subroutine. Table A-I, then, serves as an effective flow chart, detailing the precise program organization. On Table A-II are listed the Pratt & Whitney Aircraft proprietary routines and/or analysis systems which were used but not developed in the current program. The curve fit and matrix inversion routines may be replaced by standard routines from mathematical packages. MNW137 does the frequency and mode shape determination for the STAEBL generated beam stiffness, using a determinant search technique. MNT983 is the driving routine for a proprietary supersonic flutter analysis.

TABLE A-I

APPROXIMATE ANALYSIS SUBROUTINE LIST

<u>Routine</u>	<u>External References</u>		<u>Function</u>
Main Program	MYTIME COPE01 ANALIZ COPE02 COPE18 COPE04	COPE05 COPE06 COPE07 COPE09 COPE14	Main Program for COPES/CONMIN Constrained Optimization Program
ANALIZ	MYTIME RDW137 RDDATA TMAX WTW137 CALCTH	MHW137 RSFUNC OBJTV ITT983 WTT983	Constraint Calculation for COPES/CONMIN
BBIP	TSAIWV	MATPRN	Multi-Mode Root Bird Ingestion Stress
BCOORD			Beam Local Coordinate Systems
BEAM	MATPRN	MATMPY	Beam Stiffness Calculation
BIRDF	MATPRN		Bird Impulse Calculation
CALCTH			Input Blade Thickness Calculation
CNMNO1			Gradient Information
CNMNO2			Descent Vector
CNMNO3	CNMO4		One-Dimensional Search
CNMNO4			One-Dimensional Function Minimum
CNMNO5	CNMNO8		Search Direction
CNMNO6	CNMNO7	CNMNO4	One-Dimensional Search
CNMNO7			Zero of I-D Function
CNMNO8			Special Linear Optimization
CONMIN	CNMNO1 CNMNO2 CNMNO5	CNMNO3 CNMNO6	Constrained or Unconstrained Function Minimization

TABLE A-I (Cont'd.)

<u>Routine</u>	<u>External References</u>		<u>Function</u>
COPE01	COPE08		Control Input for COPEs
COPE02	CONMIN	COPE03	Control Optimization
COPE03	ANALIZ		Buffer Between CONMIN and COPEs
COPE04	ANALIZ	COPE02	Sensitivity Information
COPE05			Print Sensitivity Information
COPE06			Two Variable Function Space
COPE07			Print Two Variable Function Space
COPE08			Convert Data Input
COPE09	COPE10 COPE12	COPE15	Approximate Optimization
COPE10	COPE11		Set-Up Taylor Expansion
COPE11			Nominal Design for Approx. Optimization
COPE12	COPE13		Least Squares Fit
COPE13			Least Squares Fit Evaluation
COPE14			Print Approximate Optimization Results
COPE15	COPE16	COPE17	Function Evaluation, Approx. Optimization
COPE16			Approximate Function Evaluation
COPE17			Gradient Evaluation, Approx. Optimization
COPE18			Print Optimization Results
CORD2R	BCOORD		Blade Coordinate Systems
ESTIFF	MATPRN		Beam Material Stiffness Matrix
EXNECK	MATPRN	MATMPY	Rectangular Extended Airfoil Neck
FOD	MATPRN		Local Ingestion Parameter
FRPM	MATMPY	MATPRN	Centrifugal Force Calculation

TABLE A-I (Cont'd.)

<u>Routine</u>	<u>External References</u>		<u>Function</u>
HOLLOW	MATPRN		Hollow Blade Lamination
HOLZER	THICK WARP SECTN MATPRN	PARTN MINV MATMPY	Holzer Transform Generation
INPUT			Material and Geometry Input
ITT983	WTT983 MNT983	NDT983	Flutter Nodal Diameter Iteration
KDGG	MATPRN		Differential Stiffness Calculation
LAMINA			Lamina Stress-Strain Relation
LAMINI			Composite Blade Layup
LOAD2	MATPRN	MATMPY	Loads for Holzer Solution
MATADD			Perform Matrix Addition
MATMPY			Perform Matrix Multiplication
MATPRN			Matrix Print
MODMAS	SHAPE	MATMPY	Modal Mass determination
MODSTR	SHAPE MATPRN ZSTRES	MODMAS BIRDE MATMPY	Modal Stress Driver
NDT983			Flutter Nodal Diameter Selection
OBJTV			Objective Function Calculation
PARTN	MATPRN		Matrix Partition Routine
PPI	MATPRN		Matrix Reordering
RBE2	MATMPY	MATPRN	Rigid Body Coupling Matrices
RDDATA			Optimization Input Information
RDT983			Flutter Input
RDW137			Blade Model Input

TABLE A-I (Cont'd.)

<u>Routine</u>	<u>External References</u>	<u>Function</u>
RSFUNC		Resonance Function
SECTN	BCCORD MATPRN ESTIFF BEAM	MATMPY RBEZ FRPM Section Stiffness Matrix
SHAPE	PPI	MATMPY Eigenvector Transformation
SMULT	MATMPY	MATPRN Holzer String Multiplication
STABEL	INPUT CORDZR LAMINA PPI HOLZER STATIC	FOD ZSTRES TSAIWU MATMPY MATPRN Stacked Beam Analysis Driver (Versions Differ for Hollow and Superhybrid blades)
STAEBL	PBMFIT CUBIC	HOLLOW LAMINI Blade Interpolation for Approximate Analysis (Versions Differ for Hollow and Superhybrid Blades)
STATIC	MATPRN PARTN MATMPY	MATADD MINV Blade Static Solution
THICK		Layer Thickness Evaluation
TMAX		Section Maximum Thickness
TSAIWU	MATPRN	Tsai-Wu Failure Criterion
WARP	MATPRN	Thin Section Warping
WTT983		Flutter Output Routine
WTW137		Blade Analysis Output
ZSTRES	MATPRN	MATMPY Composite Layer Stress

TABLE A-II

PRATT & WHITNEY AIRCRAFT PROPRIETARY ROUTINES

<u>Routine</u>	<u>External References</u>	<u>Function</u>
CUBIC		Function Evaluation for Parametric Beam Fit
MINV		Matrix Inversion by Gaussian Elimination
MNT983		Pratt & Whitney Aircraft Flutter Analysis System
MNW137	STABEL	Pratt & Whitney Aircraft Beam Analysis System, Modified Through STABEL Analysis Package for Hollow and Composite Blades
MYTIME		Clock Times, Used to Bookkeep Analysis Time Consumption
PBMFIT		Parametric Beam Fit Coefficient Evaluation

APPENDIX B
COMPILED FORTRAN FOR HOLLOW BLADE
DEMONSTRATION CASE

+VERSION 1.3.0 (01 MAY 80) SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 82.141/10.47.36 PAGE 1
 REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE
 OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECCOUNT(60) SIZE(MAX) AUTODBL(DBL4)
 SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT COSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```

C          DATA SET U498DSTAEB AT LEVEL 019 AS OF 03/15/82
C          DATA SET U498DSTAEB AT LEVEL 018 AS OF 01/20/82          00001
C          DATA SET U498STAEBL AT LEVEL 015 AS OF 10/06/81         00002
C          DATA SET U498STAEBL AT LEVEL 013 AS OF 08/24/81         00003
C          DATA SET U498STAEBL AT LEVEL 012 AS OF 08/14/81         00004
ISN 0002   SUBROUTINE STAEBL          00005
ISN 0003   COMMON /BLKCH/ XP(3,53,21) , NP(21) , IND(21) , INP      00005
ISN 0004   COMMON /BLK 2/ NOACR , NOACR1 , NSTA , NSTA1 , POISSH,MST2,MST3 00007
ISN 0005   COMMON /BLK 4/ RPM , XNOACR , BR , PI , TERM , NBS       00008
ISN 0006   COMMON /BLK 7/ AREA(21) , CF(21) , AA(21) , AKG(21) , IT(7,21) 00009
ISN 0007   COMMON /BLK12/ XBARX(21) , XIMAX(21) , XIMIN(21) , YBARY(21) , 00010
           1 TLTA(21)          00011
ISN 0008   COMMON /BLKAA/ ALPMIN(21) , XSC(21) , YSC(21) , XX(21) , YY(21) 00012
ISN 0009   COMMON /BLK8/ SMB(21),ITLLE(18),VARI(235),TMAX(21),HALPHA(21) 00013
ISN 0010   COMMON /BLK A/ ALPHA(21) , HA(21) , HMNT(21) , HALMIN(21) , 00014
           1 HINERT(21)       00015
ISN 0011   COMMON /BLK 9/ DELTAC(21) , DELTAZ(21) , P(21) , R(21) , SHNT(21) 00016
ISN 0012   COMMON /INPUTT/ XSAVE(1000) , YSAVE(1000) , ZSAVE(1000) , TSAVE(1000) 00017
ISN 0013   COMMON /SC/ XSCSV(21) , YSCSV(21) , ZSCSV(21) ,          00018
           1 ALSAVE(21) , POLARI(21) , ASAVE(21)          00019
ISN 0014   COMMON /ANAL33/ DLED , DTED , DROOTD , DTIPD , TTID , TLTD , NCD 00020
           1 , NCK , TSKIN , TCENTR , PST , PCE          00021
ISN 0015   DIMENSION YH(53,21) , XM(53,21) , THC(53,21) , T(120) , TML(53,21) 00022
           1 , XMF(50,21) , YMF(50,21) , THLF(50,21) , XCG(50,21) , YCG(50,21) 00023
           2 , XSCCG(21) , YSCCG(21) , XSCG(21) , YSCG(21) , ZSCG(21) , 00024
           3 XG(50,21) , YG(50,21) , ZG(50,21)          00025
           4 , XSCR(21) , YSCR(21) , ZSCR(21)           00026
           5 , X(50,21) , Y(50,21) , Z(50,21)          00027
ISN 0016   REAL*4   A(51),B(51),C(51),D(51),A1(51),B1(51),C1(51),D1(51), 00028
           1         A2(51),B2(51),C2(51),D2(51),A3(51),B3(51),C3(51), 00029
           2         D3(51),S1(51),S2(51)              00030
ISN 0017   REAL*4   XMM(51),YMM(51),THM(51),XXMF(25),YYMF(25),THLMF(25) 00031
ISN 0018   REAL*4   SONE,STWO          00032
C          00033
C          CALCULATE MEAN Y VALUE AND THICKNESS NORMAL TO THE CHORD 00034
C          00035
C          NBR = 1          00036
ISN 0019   DO 20 I = NBR,NSTA          00037
ISN 0020   NPS = NP(I)          00038
ISN 0021   DO 10 N = 1,NPS          00039
ISN 0022   YH(N,I) = (XP(2,N,I) + XP(3,N,I)) / 2.0      00040
ISN 0023   XM(N,I) = XP(1,N,I)          00041
ISN 0024   THC(N,I) = XP(2,N,I) - XP(3,N,I)            00042
ISN 0025   10 CONTINUE          00043
ISN 0026   20 CONTINUE          00044
C          00045
C          CALCULATE THICKNESS NORMAL TO MEANLINE        00046
C          00047
ISN 0028   DO 40 I = NSR,NSTA          00048
ISN 0029   NPS = NP(I)-1          00049
ISN 0030   TML(1,I) = 0.0          00050
ISN 0031   TML(NPS+1,I) = 0.0        00051
ISN 0032   DO 30 N = 2,NPS          00052
ISN 0033   THT1 = ATAN2((YH(N,I) - YH(N-1,I)) , (XM(N,I) - XM(N-1,I))) 00053
ISN 0034   THT2 = ATAN2((YH(N+1,I) - YH(N,I)) , (XM(N+1,I) - XM(N,I))) 00054

```

```

+VERSION 1.3.0 (01 MAY 80)  STAEBL  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 82.141/10.47.36  PAGE 2
  ISN 0035      THT = ( THT1 + THT2 ) / 2.0 00055
  ISN 0036      TNL(N,I) = THC(N,I) * COS(THT) 00056
  ISN 0037      30 CCNTINUE 00057
  ISN 0038      40 CCONTINUE 00058
  ISN 0039      NSTNB = NSTA - NBR 00059
  ISN 0040      DO 45 I = NBR,NSTA,NSTNB 00060
  ISN 0041      NPS = NP(I) 00061
  C      WRITE (6,900) (YM(N,I),N=1,NPS) 00062
  C      WRITE (6,900) (XM(N,I),N=1,NPS) 00063
  C      WRITE (6,900) (THC(N,I),N=1,NPS) 00064
  C      WRITE (6,900) (TNL(N,I),N=1,NPS) 00065
  ISN 0042      45 CCONTINUE 00066
  C      00067
  C      SAVE MEAN Y VALVES AND X VALVES IN AN ARRAY AND INTERPOLATE 00068
  C      FOR EQUAL INCREMENT X VALVES . REPEAT FOR THICKNESSES 00069
  C      00070
  ISN 0043      DO 110 I = NBR,NSTA 00071
  ISN 0044      NPF = 15 00072
  ISN 0045      NPS = NP(I) 00073
  C      00074
  C      FILL IN DUMMY ARRAYS XMM,YMM,TMM FROM XM,YM,TML VECTORS 00075
  C      00076
  ISN 0046      DO 201 K = 1,NPS 00077
  ISN 0047      TMM(K) = TML(K,I) 00078
  ISN 0048      XMM(K) = XM(K,I) 00079
  ISN 0049      201 YMM(K) = YM(K,I) 00080
  C      00081
  C      00082
  C      CALL PBMFIT FOR CURVE FIT 00083
  C      00084
  ISN 0050      CALL PBMFIT(XMM,YMM,1.,1.,NPS,0,A,B,C,D,A1,B1,C1,D1,S1) 00085
  C      00086
  ISN 0051      CALL PBMFIT(XMM,TMM,1.,1.,NPS,0,A2,B2,C2,D2,A3,B3,C3,D3,S2) 00087
  C      00088
  C      EQUAL BREAKUP ARC-LENGTH 00089
  C      00090
  ISN 0052      SARC = S1(NPS) / NPF 00091
  C      00092
  C      NOW SEARCH S1 ARRAY FOR INTERVAL VALUE 00093
  C      00094
  ISN 0053      NODE = 1 00095
  ISN 0054      ARC1 = SARC / 2. 00096
  ISN 0055      205 N = 1 00097
  ISN 0056      210 IF(ARC1 .LE. S1(N)) GO TO 220 00098
  ISN 0058      N = N + 1 00099
  ISN 0059      GO TO 210 00100
  ISN 0060      220 DIST = (ARC1 - S1(N-1)) / (S1(N) - S1(N-1)) 00101
  ISN 0061      SONE = (S1(N) - S1(N-1)) * DIST 00102
  ISN 0062      STNO = (S2(N) - S2(N-1)) * DIST 00103
  C      00104
  ISN 0063      CALL CUBIC(A(N-1),B(N-1),C(N-1),D(N-1),SONE,XXMF(NODE)) 00105
  ISN 0064      CALL CUBIC(A1(N-1),B1(N-1),C1(N-1),D1(N-1),SONE,YYMF(NODE)) 00106
  ISN 0065      CALL CUBIC(A3(N-1),B3(N-1),C3(N-1),D3(N-1),STNO,TTMLF(NODE)) 00107
  C      00108
  ISN 0066      NODE = NODE + 1 00109
  ISN 0067      ARC1 = ARC1 + SARC 00110
  ISN 0068      IF(NODE .LE. NPF) GO TO 205 00111
  C      00112
  C      FILL IN THE XMF,YMF,TMLF VECTORS WITH THE INTERPOLATED VALUES 00113

```

88

```

C      00114
ISH 0070      DO 230 K = 1,NPF      00115
ISH 0071      XMF(K,I) = XXMF(K)    00116
ISH 0072      YMF(K,I) = YMF(K)    00117
C      WRITE(6,902) I,K,XMF(K,I),YMF(K,I) 00118
ISH 0073      902 FORMAT(5X,'I,K,XMF,YMF',2I5,2E12.5) 00119
ISH 0074      230 TTMLF(K,I) = TTMLF(K) 00120
C      00121
ISH 0075      110 CONTINUE      00122
C      00123
C      TRANSLATE COORDINATES TO AN ENGINE AXIS SYSTEM 00124
C      00125
ISH 0076      DO 130 I = NSR,NSTA 00126
C      00127
ISH 0077      XCHORD = SMB(I)      00128
C      00129
ISH 0078      XSCLE = XCHORD - XSC(I) 00130
C      00131
ISH 0079      XSCCG(I) = XSCLE - (XCHORD - XBARX(I)) 00132
ISH 0080      YSCCG(I) = YSC(I) - YBARY(I) 00133
C      00134
ISH 0081      DO 120 N = 1,NPF      00135
ISH 0082      XCG(N,I) = XMF(N,I) - (XCHORD - XBARX(I)) 00136
ISH 0083      YCG(N,I) = YMF(N,I) - YBARY(I) 00137
ISH 0084      120 CONTINUE      00138
ISH 0085      130 CONTINUE      00139
C      00140
C      SHIFT XY PLANE TO ENGINE AXIS YZ 00141
C      00142
C      00143
ISH 0086      DO 150 I = NSR,NSTA 00144
ISH 0087      XSCG(I) = R(I)      00145
ISH 0088      YSCG(I) = YSCCG(I) 00146
ISH 0089      ZSCG(I) = XSCCG(I) 00147
ISH 0090      DO 140 N = 1,NPF      00148
ISH 0091      XG(N,I) = R(I)      00149
ISH 0092      YG(N,I) = YCG(N,I) 00150
ISH 0093      ZG(N,I) = XCG(N,I) 00151
ISH 0094      140 CONTINUE      00152
ISH 0095      150 CONTINUE      00153
C      00154
C      ROTATE THRU ALPHA CHORD - 90.0 00155
C      00156
ISH 0096      DO 170 I = NSR,NSTA 00157
C      ALPHA(I) = 90. * .0174533 00158
ISH 0097      ANG = -ALPHA(I) + 90.0 * .0174533 00159
ISH 0098      EM = COS(ANG)      00160
ISH 0099      EN = SIN(ANG)      00161
ISH 0100      XSCR(I) = XSCG(I) 00162
ISH 0101      YSCR(I) = YSCG(I) * EM + EN * ZSCG(I) 00163
ISH 0102      ZSCR(I) = EM * ZSCG(I) - EN * YSCG(I) 00164
C      WRITE (6,900) ANG, EN , EM , XSCR(I) , YSCR(I) , ZSCR(I) , AREA(I) 00165
ISH 0103      DO 160 N = 1,NPF      00166
ISH 0104      X(N,I) = XG(N,I)      00167
ISH 0105      Y(N,I) = YG(N,I)* EM + EN * ZG(N,I) 00168
ISH 0106      Z(N,I) = EM * ZG(N,I) - EN * YG(N,I) 00169
ISH 0107      160 CONTINUE      00170
ISH 0108      170 CONTINUE      00171
C      DO 175 I = NSR,NSTA,NSTNB 00172
C      WRITE (6,900) (XCG(N,I),N=1,NPF)

```

1

11

```

C      WRITE (6,900) (YCG(N,I),N=1,NPF)      00173
C      WRITE (6,900) (XG(N,I),N=1,NPF)      00174
C      WRITE (6,900) (YG(N,I),N=1,NPF)      00175
C      WRITE (6,900) (ZG(N,I),N=1,NPF)      00176
C      WRITE (6,900) (X(N,I),N=1,NPF)      00177
C      WRITE (6,900) (Y(N,I),N=1,NPF)      00178
C      WRITE (6,900) (Z(N,I),N=1,NPF)      00179
C      WRITE (6,900) (TMLF(N,I),N=1,NPF)     00180
C 175 CONTINUE                               00181
ISN 0109      IJ = 1                          00182
ISN 0110      IF(IJ .EQ. 1) GO TO 811         00183
C                                                    00184
C      T03902  GENERATED NODES AND THICKNESSES 00185
C                                                    00186
ISN 0112      DO 800 I = NBR,NSTA            00187
ISN 0113      DO 801 J = 1,NPF              00188
ISN 0114      801 READ(5,802) Y(J,I),Z(J,I) 00189
ISN 0115      802 FORMAT(/,32X,2F8.0)       00190
C                                                    00191
ISN 0116      DO 805 J = 1,NPF              00192
ISN 0117      READ(5,803) P1,P2             00193
ISN 0118      803 FORMAT(24X,F8.0,/,24X,F8.0) 00194
ISN 0119      805 TMLF(J,I) = (P1 + P2) / 2. 00195
C                                                    00196
C      WRITE(6,810) I                         00197
C310 FORMAT(5X,'T039 NODES AND THICKNESSES FOR SECTION ',I5) 00198
C      WRITE(6,900) (X(N,I),N=1,NPF)        00199
C      WRITE(6,900) (Y(N,I),N=1,NPF)        00200
C      WRITE(6,900) (Z(N,I),N=1,NPF)        00201
C      WRITE(6,900) (TMLF(N,I),N=1,NPF)     00202
C                                                    00203
ISN 0120      800 CONTINUE                   00204
ISN 0121      811 CONTINUE                   00205
C                                                    00206
ISN 0122      J = 0                          00207
ISN 0123      K = 0                          00208
ISN 0124      DO 190 I = NBR,NSTA            00209
ISN 0125      K = K + 1                      00210
ISN 0126      XSCSV(K) = XSCR(I)            00211
ISN 0127      YSCSV(K) = YSCR(I)            00212
ISN 0128      ZSCSV(K) = ZSCR(I)            00213
ISN 0129      ALSAVE(K) = ALPHA(I)          00214
ISN 0130      POLARI(K) = XIMIN(I) + XIMAX(I) 00215
ISN 0131      ASAVE(K) = AREA(I)            00216
ISN 0132      DO 180 N = 1,NPF              00217
ISN 0133      J = J + 1                      00218
ISN 0134      XSAVE(J) = X(N,I)             00219
ISN 0135      YSAVE(J) = Y(N,I)             00220
ISN 0136      ZSAVE(J) = Z(N,I)             00221
ISN 0137      TSAVE(J) = TMLF(N,I)          00222
C      WRITE (7,901) XSAVE(J) , YSAVE(J) , ZSAVE(J) , TSAVE(J) 00223
ISN 0138      180 CONTINUE                   00224
ISN 0139      190 CONTINUE                   00225
ISN 0140      K = 0                          00226
ISN 0141      DO 200 I = NBR,NSTA            00227
ISN 0142      K = K + 1                      00228
C      WRITE (7,901) XSCSV(K) , YSCSV(K) , ZSCSV(K) , ALSAVE(K) 00229
C      1 , POLARI(K) , ASAVE(K)              00230
ISN 0143      200 CONTINUE                   00231

```

```
+VERSION 1.3.0 (01 MAY 80)   STAEBL   SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)   DATE 82.141/10.47.36   PAGE 5
  ISN 0144      NTMNR = NSTA - NBR + 1      00232
  ISN 0145      IF (NCD .EQ. 1)             00233
                ICALL HOLLOW (DLED , DTED , DROOTD , DTIPD ,      00234
                2 TTID , TLTD , NPF , NTMNR)                       00235
  ISN 0147      IF (NCD .EQ. 2)             00236
                1 CALL LAMINI (TSKIN , TCENTR , PBT , PGE , NPF , NTMNR) 00237
  ISN 0149      901 FORMAT (6E12.5)         00238
  ISN 0150      900 FORMAT (1X,9F8.5)       00239
  ISN 0151      RETURN                      00240
  ISN 0152      END                         00241
```

```
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT HCMAP NOFORMAT GOSTIT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS*   SOURCE STATEMENTS = 151, PROGRAM SIZE = 141916, SUBPROGRAM NAME =STAEBL
*STATISTICS*   NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****
```

2972K BYTES OF CORE NOT USED

```

C          DATA SET U498DSTABE AT LEVEL 048 AS OF 03/31/82
C
ISN 0002      SUBROUTINE BBIP ( NC )                00001*35
ISN 0003      IMPLICIT REAL*8(A-H,O-Z)             00002*35
C                                                    00003*35
C                                                    00004*35
C          MULTI-MODE BENDING BIRD INGESTION PARAMETER 00005*35
C                                                    00006*35
ISN 0004      COMMON /FAIL/ VV(42),TSAI(8,25)      00007*35
ISN 0005      COMMON /SMAXX/ SMAX1(8),SMAX2(8),SMAX3(8),SMAXLE(8),SMAXTE(8) 00008*37
ISN 0006      COMMON /MODAL1/ STRESS(3,3,8,25),FACTOR(3),RMASS(3) 00009*35
ISN 0007      COMMON /BLK15/ DIANOD(100),FN1(100),FILL1(300) 00010*35
ISN 0008      DIMENSION SEFF(3,8,25)              00011*35
C                                                    00012*35
C          CALCULATE TIME TO 1/4 OF FIRST BENDING    00013*35
C                                                    00014*35
ISN 0009      TIME = 0.25 / FN1(1)                 00015*35
C                                                    00016*35
ISN 0010      WRITE(6,200) TIME                    00017*35
ISN 0011      200 FORMAT(//,5X,'BBIP CALCULATION , MODAL SUPERPOSITION METHOD',/, 00018*35
                15X,'QUARTER OF FIRST BENDING (SECONDS) =',E12.5,/) 00019*35
C                                                    00020*35
C          EFFECTIVE STRESS CALCULATION - MODAL SUPERPOSITION 00021*35
C          FOR EACH LAYER, EACH BEAM - THREE COMPONENTS 00022*35
C                                                    00023*35
ISN 0012      ARG = FN1(1) * 2. * 3.141593 * TIME 00024*35
ISN 0013      FS1 = FACTOR(1) * DSIN( ARG )        00025*35
ISN 0014      ARG = FN1(2) * 2. * 3.141593 * TIME 00026*35
ISN 0015      FS2 = FACTOR(2) * DSIN( ARG )        00027*35
ISN 0016      ARG = FN1(3) * 2. * 3.141593 * TIME 00028*35
ISN 0017      FS3 = FACTOR(3) * DSIN( ARG )        00029*35
C                                                    00030*35
ISN 0018      DO 100 I = 1,3                        00031*35
ISN 0019      DO 100 J = 1,8                        00032*35
ISN 0020      DO 100 K = 1,NC                       00033*35
ISN 0021      100 SEFF(I,J,K) = STRESS(1,I,J,K) * FS1 + STRESS(2,I,J,K) * FS2 + 00034*35
                1 STRESS(3,I,J,K) * FS3            00035*35
C                                                    00036*35
C          TSAI-WU FAILURE CALCULATION               00037*35
C                                                    00038*35
ISN 0022      CALL TSAIWU ( SEFF,NC )               00039*35
C                                                    00040*35
C          PICK OUT THE MAX TSAI-WU FOR EACH LAYER OVER ALL BEAMS 00041*35
C                                                    00042*35
ISN 0023      DO 110 I = 1,8                        00043*35
ISN 0024      SMAX3(I) = 0.                         00044*36
ISN 0025      DO 110 J = 1,NC                       00045*35
ISN 0026      110 IF(SMAX3(I) .LT. TSAI(I,J)) SMAX3(I) = TSAI(I,J) 00046*36
C                                                    00047*36
ISN 0028      WRITE(6,300)                          00048*36
ISN 0029      300 FORMAT(//,5X,'** MAX TSAI-WU ROOT STRESS ANALYSIS **') 00049*36
ISN 0030      CALL MATPRN(SMAX1,8,1,'MAX1')         00050*36
ISN 0031      WRITE(6,301)                          00051*36
ISN 0032      301 FORMAT(//,5X,'** MAX TSAI-WU HOLLOW SECTION ANALYSIS **') 00052*36
ISN 0033      CALL MATPRN(SMAX2,8,1,'MAX2')         00053*36
ISN 0034      WRITE(6,302)                          00054*36

```

92

```
+VERSION 1.3.0 (01 MAY 80)      BBIP      SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)      DATE 82.141/10.47.43      PAGE 2
  ISN 0035      302  FORMAT(//,5X,'** MAX TSAI-WU BBIP ANALYSIS **')      00055*36
  ISN 0036      CALL MATPRN(SMAX3,8,1,'MAX3')      00056*36
  ISN 0037      WRITE(6,303)      00057*37
  ISN 0038      303  FORMAT(//,5X,'** TSAI-WU AT LE AND TE **')      00058*37
  ISN 0039      CALL MATPRN (SMAXLE,8,1,' LE ')      00059*37
  ISN 0040      CALL MATPRN (SMAXTE,8,1,' TE ')      00060*37
                                     C      00061*35
  ISN 0041      RETURN      00062*35
  ISN 0042      END      00063*35
```

```
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS*      SOURCE STATEMENTS =      41, PROGRAM SIZE =      6244, SUBPROGRAM NAME = BBIP
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****
```

3012K BYTES OF CORE NOT USED

```

C 00064*35
C 00065*35
ISN 0002 SUBROUTINE BIRDF( NR,NC,ISTA,BFORCE,SQUASH ) 00066*35
ISN 0003 IMPLICIT REAL*8 (A-H,O-Z) 00067*35
C 00068*35
C 00069*35
C CALCULATES THE BIRD IMPACT FORCE 00070*35
C ASSUMPTIONS 1. 2 INCHES FROM TIP 00071*35
C 2. 2.5 LB BIRD - 4 INCH DIAMETER 00072*35
C 3. VBIRD = 180 KNOTS 00073*35
C 4. LOADED OVER SQUASH-UP TIME 00074*35
C 5. SLICE FROM CENTER OF SPHERE 00075*35
C 00076*35
ISN 0004 COMMON /INPUTT/ X(1000),Y(1000),Z(1000),T(1000) 00077*35
ISN 0005 COMMON /BLK 3/ FN,BLADES,BETA,THR,THT,MST1,NBB 00078*35
ISN 0006 COMMON /BC/ C(6,6),RPM 00079*35
ISN 0007 COMMON /SC/ XSC(21),YSC(21),ZSC(21),ALPHA(21),SECI(21),SECA(21) 00080*35
ISN 0008 COMMON /BLK 8/ SMB(21),ITL(18),VARI(235),TMAX(21),HALPHA(21) 00081*35
ISN 0009 DIMENSION BFORCE(6,1) 00082*35
C 00083*35
C IMPACT STATION , ISTA 00084*35
C 00085*35
ISN 0010 ISTA = 1 00086*35
ISN 0011 XTOP = X(NR * NC) - 2. 00087*35
ISN 0012 51 NODE = (ISTA-1) * NC + 1 00088*35
ISN 0013 IF(X(NODE) .GT. XTOP) GO TO 50 00089*35
ISN 0015 ISTA = ISTA + 1 00090*35
ISN 0016 GO TO 51 00091*35
ISN 0017 50 CONTINUE 00092*39
C 00093*39
ISN 0018 ISTA = ISTA - 1 00094*39
ISN 0019 NODE = NODE - NC 00095*39
C 00096*35
C WRITE(6,300) NODE,ISTA 00097*35
C300 FORMAT(//,5X,'** IN BIRD **,'NODE,ISTA',2I5) 00098*35
C 00099*35
C CALCULATE THE IMPACT PARAMETERS 00100*35
C 00101*35
ISN 0020 VBLADE = X(NODE) * RPM 00102*35
ISN 0021 VBIRD = 180. * 1.688 * 12. 00103*35
ISN 0022 VREL = DSQRT( VBLADE**2 + VBIRD**2 ) 00104*35
ISN 0023 ARG = VBIRD / VBLADE 00105*35
ISN 0024 PHI = DATAN (ARG) 00106*35
ISN 0025 THETA = ALPHA(ISTA) - PHI 00107*35
ISN 0026 GAP = 2. * 3.141593 * X(NODE) / BLADES 00108*35
ISN 0027 SL = GAP * DTAN (PHI) 00109*35
ISN 0028 RBIRD = 2. 00110*35
ISN 0029 SQUASH = 2. * RBIRD / VREL 00111*35
ISN 0030 RHO = 0.036 / 386.4 00112*35
ISN 0031 BMASS = SL * 3.141593 * RBIRD**2 * RHO 00113*35
C 00114*35
ISN 0032 F = BMASS * VREL * DSIN(THETA) / SQUASH 00115*35
C 00116*35
C CENTER OF IMPACT , COI 00117*35
C 00118*35
ISN 0033 COI = SL / 2. / DSIN(PHI)

```

94

```

C      00119*35
C      00120*35
C      00121*35
ISN 0034 TORQUE = SHB(ISTA) / 2. - COI      00122*35
ISN 0035 TORQUE = TORQUE * F              00123*35
C      00124*35
C      00125*35
C      00126*35
ISN 0036 BFORCE(1,1) = 0.                 00127*35
ISN 0037 BFORCE(2,1) = F                  00128*35
ISN 0038 BFORCE(3,1) = 0.                 00129*35
ISN 0039 BFORCE(4,1) = TORQUE             00130*35
ISN 0040 BFORCE(5,1) = 0.                 00131*35
ISN 0041 BFORCE(6,1) = 0.                 00132*35
C      00133*36
C      00134*35
C      00135*36
C      1 SQUASH,RHO,BMASS,F,COI,TORQUE    00136*36
C200 FORMAT(//,5X,'BBIP ANALYSIS',/,5X,'STATION,NODE,VBLADE,VBIRD', 00137*36
C      1 2I5,2E12.5,/,5X,'VREL,PHI,THETA,SL,RBIRD,SQUASH',6E12.5, 00138*36
C      2 /,5X,'RHO,BMASS,F,COI,TORQUE',5E12.5,/) 00139*36
C      00140*35
ISN 0042 RETURN                            00141*35
ISN 0043 END                                00142*35

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTNT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 42, PROGRAM SIZE = 1068, SUBPROGRAM NAME = BIRDF
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

3012K BYTES OF CORE NOT USED

```

    C          00143
ISN 0002      SUBROUTINE BCOORD(IP,IPP,BLOCAL)          00144
    C          00145
    C          THIS ROUTINE CALCULATES THE BEAM LOCAL COORDINATE SYSTEM,BLOCAL 00146
    C          THE SYSTEM IS LOCATED AT THE BEAM END I 00147
    C          00148
ISN 0003      IMPLICIT REAL*8(A-H,O-Z)                00149
ISN 0004      COMMON /INPUTT/ X(1000),Y(1000),Z(1000),T(1000) 00150
ISN 0005      COMMON /COCRD1/ CLOCAL(3,3,1000)          00151
ISN 0006      DIMENSION BLOCAL(3,3)                    00152
    C          00153
    C          LONGITUDINAL VECTOR , X                  00154
    C          00155
ISN 0007      X1 = X(IPP) - X(IP)                      00156
ISN 0008      X2 = Y(IPP) - Y(IP)                      00157
ISN 0009      X3 = Z(IPP) - Z(IP)                      00158
ISN 0010      XMAG = DSQRT( X1**2 + X2**2 + X3**2 )    00159
ISN 0011      X1 = X1 / XMAG                          00160
ISN 0012      X2 = X2 / XMAG                          00161
ISN 0013      X3 = X3 / XMAG                          00162
    C          00163
    C          LOCAL Z UNIT VECTOR = X CROSS Y          00164
    C          00165
ISN 0014      Z1 = X2 * CLOCAL(2,3,IP) - X3 * CLOCAL(2,2,IP) 00166
ISN 0015      Z2 = X3 * CLOCAL(2,1,IP) - X1 * CLOCAL(2,3,IP) 00167
ISN 0016      Z3 = X1 * CLOCAL(2,2,IP) - X2 * CLOCAL(2,1,IP) 00168
ISN 0017      ZMAG = DSQRT( Z1**2 + Z2**2 + Z3**2 )    00169
ISN 0018      Z1 = Z1 / ZMAG                          00170
ISN 0019      Z2 = Z2 / ZMAG                          00171
ISN 0020      Z3 = Z3 / ZMAG                          00172
    C          00173
    C          THE LOCAL Y UNIT VECTOR = Z CROSS X      00174
    C          00175
ISN 0021      BLOCAL(2,1) = Z2 * X3 - Z3 * X2          00176
ISN 0022      BLOCAL(2,2) = Z3 * X1 - Z1 * X3          00177
ISN 0023      BLOCAL(2,3) = Z1 * X2 - Z2 * X1          00178
    C          00179
ISN 0024      BLOCAL(1,1) = X1                        00180
ISN 0025      BLOCAL(1,2) = X2                        00181
ISN 0026      BLOCAL(1,3) = X3                        00182
ISN 0027      BLOCAL(3,1) = Z1                        00183
ISN 0028      BLOCAL(3,2) = Z2                        00184
ISN 0029      BLOCAL(3,3) = Z3                        00185
    C          00186
ISN 0030      RETURN                                  00187
ISN 0031      END                                    00188

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
 *OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
 STATISTICS SOURCE STATEMENTS = 30, PROGRAM SIZE = 718, SUBPROGRAM NAME =BCCOORD
 STATISTICS NO DIAGNOSTICS GENERATED
 ***** END OF COMPILATION *****

3016K BYTES OF CORE NOT USED

REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

96

```

C
ISN 0002      SUBROUTINE BEAM(NC,IBEAM,BLOCAL,BCMGG,RPM)          00189
C                                                     00190
C                                                     00191
C THIS ROUTINE DEFINES THE FOLLOWING MATRICES;          00192
C CN(12,12) -- COEFFICIENT MATRIX                      00193
C X(6,12) -- STRAIN-DISPLACEMENT MATRIX               00194
C A(12,8) -- EQUILIBRIUM MATRIX                       00195
C BM(12,12)- THE BEAM MASS (NO ROTATIONAL INERTIA)    00196
C BCMGG(12,12) - THE BEAM CENTRIFUGAL RESTORING MATRIX 00197
C THEN FINDS THE BEAM STIFFNESS MATRIX , BK(12,12)    00198
C                                                     00199
ISN 0003      IMPLICIT REAL*8(A-H,O-Z)                  00200
ISN 0004      COMMON /BSTIFF/ BK(12,12),BM(12,12)       00201
ISN 0005      COMMON /ESTIF/ E(8,6)                    00202
ISN 0006      COMMON /LAYER/ TH(7,25),BMASS(25),BSPAN(25),BHIDTH(25) 00203
ISN 0007      COMMON /REOROR/ P(12,12),PT(12,12)       00204
ISN 0008      DIMENSION CN(12,12),X(6,12),A(12,8),BLOCAL(3,3),BCMGG(12,12) 00205
ISN 0009      DIMENSION D1(6,12),D2(8,12),D3(12,12),CHGG(12,12),BL(12,12) 00206
ISN 0010      DIMENSION BLT(12,12)                     00207
C                                                     00208
C FIRST ZERO ALL MATRICES TO BE USED                   00209
C                                                     00210
ISN 0011      DO 100 I = 1,12                           00211
ISN 0012      DO 100 J = 1,12                           00212
ISN 0013      BL(I,J) = 0.                              00213
ISN 0014      CMGG(I,J) = 0.                            00214
ISN 0015      100 CN(I,J) = 0.                          00215
C                                                     00216
ISN 0016      DO 101 I = 1,6                            00217
ISN 0017      DO 101 J = 1,12                           00218
ISN 0018      101 X(I,J) = 0.                           00219
C                                                     00220
ISN 0019      DO 102 I = 1,12                           00221
ISN 0020      DO 102 J = 1,8                            00222
ISN 0021      102 A(I,J) = 0.                           00223
C                                                     00224
C BEGIN LOOP FOR BEAM (J)                              00225
C                                                     00226
C                                                     00227
C THE COEFFICIENT MATRIX , CN(12,12)                  00228
C                                                     00229
ISN 0022      S = BSPAN(IBEAM)                          00230
ISN 0023      S2 = S**2                                  00231
ISN 0024      S3 = S**3                                  00232
ISN 0025      CN(1,1) = 1.                              00233
ISN 0026      CN(2,2) = -1.                             00234
ISN 0027      CN(3,1) = -1. / S                        00235
ISN 0028      CN(3,3) = 1. / S                         00236
ISN 0029      CN(4,2) = -1. / S                        00237
ISN 0030      CN(4,4) = 1. / S                         00238
ISN 0031      CN(5,5) = 1.                              00239
ISN 0032      CN(6,6) = 1.                              00240
ISN 0033      CN(7,5) = -3. / S2                       00241
ISN 0034      CN(7,6) = -2. / S                       00242
ISN 0035      CN(7,7) = 3. / S2                       00243

```



```

+VERSION 1.3.0 (01 MAY 80)      BEAM  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)      DATE 82.141/10.48.06      PAGE  3
  ISN 0081      CALL MATHPY(A,D2,BK,12,8,12)      00303
  C      00304
  C      REORDER THE BEAM STIFFNESS, BK, TO LOOK LIKE A NASTRAN VECTOR      00305
  C      BK = P * BK * P(TRANS)      00306
  C      00307
  ISN 0082      CALL MATMPY(BK,PT,D3,12,12,12)      00308
  ISN 0083      CALL MATMPY(P,D3,BK,12,12,12)      00309
  C      00310
  C      FORM THE BEAM MASS MATRIX, BM      00311
  C      00312
  ISN 0084      DO 15 I = 1,12      00313
  ISN 0085      DO 15 J = 1,12      00314
  ISN 0086      15  BM(I,J) = 0.      00315
  ISN 0087      BMM = BMASS(IBEAM) / 2.      00316
  ISN 0088      BM(1,1) = BMM      00317
  ISN 0089      BM(2,2) = BMM      00318
  ISN 0090      BM(3,3) = BMM      00319
  ISN 0091      BM(7,7) = BMM      00320
  ISN 0092      BM(8,8) = BMM      00321
  ISN 0093      BM(9,9) = BMM      00322
  C      00323
  C      THE BEAM CENTRIFUGAL RESTORING MATRIX , BCMGG      00324
  C      -- FIRST DEFINE THE TRANSFORMATION MATRIX , BL      00325
  C      00326
  ISN 0094      IROW = 1      00327
  ISN 0095      DO 90 I = 1,4      00328
  ISN 0096      DO 90 J = 1,3      00329
  ISN 0097      ICOL = (I-1) * 3 + 1      00330
  ISN 0098      DO 91 K = 1,3      00331
  ISN 0099      BL(IROW,ICOL) = BLOCAL(J,K)      00332
  ISN 0100      91  ICOL = ICOL + 1      00333
  ISN 0101      90  IROW = IROW + 1      00334
  C      00335
  ISN 0102      DO 92 I = 1,12      00336
  ISN 0103      DO 92 J = 1,12      00337
  ISN 0104      92  BLT(I,J) = BL(J,I)      00338
  C      00339
  C      NONZERO VALUES OF THE CENTRIFUGAL RESTORING MATRIX      00340
  C      WHEN DEFINED IN THE GLOBAL COORDINATE SYSTEM      00341
  C      00342
  ISN 0105      CMGG(2,2) = BMASS(IBEAM) / 2. * RPM**2      00343
  ISN 0106      CMGG(8,8) = CMGG(2,2)      00344
  C      00345
  C      TRANSFORM CMGG TO THE LOCAL BEAM SYSTEM TO FIND BCMGG      00346
  C      BCMGG = BL * CMGG * BLT      00347
  C      00348
  ISN 0107      CALL MATHPY(CMGG,BLT,D3,12,12,12)      00349
  ISN 0108      CALL MATHPY(BL,D3,BCMGG,12,12,12)      00350
  C      00351
  C      CALL MATPRN(BL,12,12,' BL ')      00352
  C      CALL MATPRN(CMGG,12,12,'CMGG')      00353
  C      CALL MATPRN(BCMGG,12,12,'BCMGG')      00354
  C      00355
  C      WRITE(6,7)      00356
  C7  FORMAT(/,5X,'BEAM STIFFNESS,BK',/)      00357
  C      CALL MATPRN(BK,12,12,'BK ')      00358
  C      WRITE(6,16)      00359
  C16  FORMAT(/,5X,'BEAM MASS , BM',/)      00360
  C      CALL MATPRN(BM,12,12,'BM ')      00361

```

```
+VERSION 1.3.0 (01 MAY 80)      BEAM      SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)      DATE 82.141/10.48.06      PAGE 4
      C
      ISN 0109      RETURN      00362
      ISN 0110      END      00363
      00364
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS*      SOURCE STATEMENTS =      109, PROGRAM SIZE =      10198, SUBPROGRAM NAME = BEAM
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****
                                     2996K BYTES OF CORE NOT USED
```

REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```

100
C
ISN 0002      SUBROUTINE CORD2R(XP,YP,ZP,ISPAN,ICHORD)      00365
ISN 0003      IMPLICIT REAL*8(A-H,O-Z)                    00366
ISN 0004      COMMON /COORD1/ CLOCAL(3,3,1000)            00367
ISN 0005      DIMENSION XP(1),YP(1),ZP(1),BLOCAL(3,3)    00368
ISN 0006      NLCNPC=ICHORD                                00370
ISN 0007      NLRNPR=ISPAN                                 00371
C
C
C      THIS ROUTINE CALCULATES THE CORD2R FOR THE BLADE  00372
C
C
ISN 0008      IP = 1                                       00375
ISN 0009      DO 100 I = 1,ISPAN                             00376
ISN 0010      IP = (I-1) * NLCNPC + 1                       00377
ISN 0011      DO 101 J = 1,ICHORD                           00378
C
ISN 0012      IF(ICHORD .EQ. 1) GO TO 200                   00379
C
ISN 0014      IF(I.EQ.1.AND.J.EQ.1) GO TO 1                 00380
ISN 0016      IF(I.EQ.1.AND.J.GT.1.AND.J.LT.NLCNPC) GO TO 2 00381
ISN 0018      IF(I.EQ.1.AND.J.EQ.NLCNPC) GO TO 3            00382
ISN 0020      IF(I.GT.1.AND.I.LT.NLRNPR.AND.J.EQ.1) GO TO 4 00383
ISN 0022      IF(I.GT.1.AND.I.LT.NLRNPR.AND.J.GT.1.AND.J.LT.NLCNPC) GO TO 5 00384
ISN 0024      IF(I.GT.1.AND.I.LT.NLRNPR.AND.J.EQ.NLCNPC) GO TO 6 00385
ISN 0026      IF(I.EQ.NLRNPR.AND.J.EQ.1) GO TO 7           00386
ISN 0028      IF(I.EQ.NLRNPR.AND.J.GT.1.AND.J.LT.NLCNPC) GO TO 8 00387
ISN 0030      IF(I.EQ.NLRNPR.AND.J.EQ.NLCNPC) GO TO 9     00388
ISN 0032      GO TO 10                                      00389
ISN 0033      1 X1 = XP(IP+NLCNPC) - XP(1)                  00390
ISN 0034      X2 = YP(IP+NLCNPC) - YP(1)                   00391
ISN 0035      X3 = ZP(IP+NLCNPC) - ZP(1)                   00392
ISN 0036      Z1 = 0.                                       00393
ISN 0037      Z2 = YP(IP+1) - YP(1)                        00394
ISN 0038      Z3 = ZP(IP+1) - ZP(1)                        00395
ISN 0039      GO TO 10                                      00396
ISN 0040      2 X1 = XP(IP+NLCNPC) - XP(IP)                 00397
ISN 0041      X2 = YP(IP+NLCNPC) - YP(IP)                  00398
ISN 0042      X3 = ZP(IP+NLCNPC) - ZP(IP)                   00399
ISN 0043      Z1 = 0.                                       00400
ISN 0044      Z2 = YP(IP+1) - YP(IP-1)                    00401
ISN 0045      Z3 = ZP(IP+1) - ZP(IP-1)                    00402
ISN 0046      GO TO 10                                      00403
ISN 0047      3 X1 = XP(IP+NLCNPC) - XP(IP)                 00404
ISN 0048      X2 = YP(IP+NLCNPC) - YP(IP)                  00405
ISN 0049      X3 = ZP(IP+NLCNPC) - ZP(IP)                   00406
ISN 0050      Z1 = 0.                                       00407
ISN 0051      Z2 = YP(IP) - YP(IP-1)                      00408
ISN 0052      Z3 = ZP(IP) - ZP(IP-1)                      00409
ISN 0053      GO TO 10                                      00410
ISN 0054      4 X1 = XP(IP+NLCNPC) - XP(IP-NLCNPC)         00411
ISN 0055      X2 = YP(IP+NLCNPC) - YP(IP-NLCNPC)         00412
ISN 0056      X3 = ZP(IP+NLCNPC) - ZP(IP-NLCNPC)         00413
ISN 0057      Z1 = 0.                                       00414
ISN 0058      Z2 = YP(IP+1) - YP(IP)                      00415
ISN 0059      Z3 = ZP(IP+1) - ZP(IP)                      00416
ISN 0060      GO TO 10                                      00417

```

```

ISN 0061    5    X1 = XP(IP+NLCNPC) - XP(IP-NLCNPC)    00420
ISN 0062            X2 = YP(IP+HLCNPC) - YP(IP-NLCNPC)    00421
ISN 0063            X3 = ZP(IP+NLCNPC) - ZP(IP-NLCNPC)    00422
ISN 0064            Z1 = 0.    00423
ISN 0065            Z2 = YP(IP+1) - YP(IP-1)    00424
ISN 0066            Z3 = ZP(IP+1) - ZP(IP-1)    00425
ISN 0067            GO TO 10    00426
ISN 0068    6    X1 = XP(IP+NLCNPC) - XP(IP-NLCNPC)    00427
ISN 0069            X2 = YP(IP+NLCNPC) - YP(IP-NLCNPC)    00428
ISN 0070            X3 = ZP(IP+NLCNPC) - ZP(IP-NLCNPC)    00429
ISN 0071            Z1 = 0.    00430
ISN 0072            Z2 = YP(IP) - YP(IP-1)    00431
ISN 0073            Z3 = ZP(IP) - ZP(IP-1)    00432
ISN 0074            GO TO 10    00433
ISN 0075    7    X1 = XP(IP) - XP(IP-NLCNPC)    00434
ISN 0076            X2 = YP(IP) - YP(IP-NLCNPC)    00435
ISN 0077            X3 = ZP(IP) - ZP(IP-NLCNPC)    00436
ISN 0078            Z1 = 0.    00437
ISN 0079            Z2 = YP(IP+1) - YP(IP)    00438
ISN 0080            Z3 = ZP(IP+1) - ZP(IP)    00439
ISN 0081            GO TO 10    00440
ISN 0082    8    X1 = XP(IP) - XP(IP-NLCNPC)    00441
ISN 0083            X2 = YP(IP) - YP(IP-NLCNPC)    00442
ISN 0084            X3 = ZP(IP) - ZP(IP-NLCNPC)    00443
ISN 0085            Z1 = 0.    00444
ISN 0086            Z2 = YP(IP+1) - YP(IP-1)    00445
ISN 0087            Z3 = ZP(IP+1) - ZP(IP-1)    00446
ISN 0088            GO TO 10    00447
ISN 0089    9    X1 = XP(IP) - XP(IP-NLCNPC)    00448
ISN 0090            X2 = YP(IP) - YP(IP-NLCNPC)    00449
ISN 0091            X3 = ZP(IP) - ZP(IP-NLCNPC)    00450
ISN 0092            Z1 = 0.    00451
ISN 0093            Z2 = YP(IP) - YP(IP-1)    00452
ISN 0094            Z3 = ZP(IP) - ZP(IP-1)    00453
ISN 0095    10    CONTINUE    00454
C    NOW FORM THE UNIT VECTORS , FIRST LOCAL Z THEN Y AND FINALLY X    00455
     ZMAG =DSQRT(Z1**2 + Z2**2 + Z3**2)    00456
     Z1 = Z1 / ZMAG    00457
     Z2 = Z2 / ZMAG    00458
     Z3 = Z3 / ZMAG    00459
ISN 0100            XMAG =DSQRT(X1**2 + X2**2 + X3**2)    00460
ISN 0101            X1 = X1 / XMAG    00461
ISN 0102            X2 = X2 / XMAG    00462
ISN 0103            X3 = X3 / XMAG    00463
C    LOCAL Y UNIT VECTOR , Z CROSS X    00464
     Y1 = Z2*X3 - Z3*X2    00465
     Y2 = -Z1*X3 + Z3*X1    00466
     Y3 = Z1*X2 - Z2*X1    00467
     YMAG =DSQRT(Y1**2 + Y2**2 + Y3**2)    00468
     Y1 = Y1 / YMAG    00469
     Y2 = Y2 / YMAG    00470
     Y3 = Y3 / YMAG    00471
C    LOCAL X UNIT VECTOR , Y CROSS Z    00472
     X1 = Y2*Z3 - Y3*Z2    00473
     X2 = -Y1*Z3 + Y3*Z1    00474
     X3 = Y1*Z2 - Y2*Z1    00475
ISN 0114            A1=0    00476
ISN 0115            A2=0    00477
ISN 0116            A3=0    00478

```

```

+VERSION 1.3.0 (01 MAY 80)   CORD2R   SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)   DATE 82.141/10.48.12   PAGE 3
  ISN 0117   200 CONTINUE                               00479
  ISN 0118                               IF(ICHORD .EQ. 1) GO TO 201   00480
  ISN 0120                               GO TO 202                     00481
  ISN 0121   201 CLOCAL(2,1,IP) = 0.                   00482
  ISN 0122                               CLOCAL(2,2,IP) = 1.         00483
  ISN 0123                               CLOCAL(2,3,IP) = 0.         00484
                                     C                               00485
  ISN 0124                               II = IP                       00486
  ISN 0125                               IF(I .EQ. ISPAN) II = IP - 1   00487
  ISN 0127                               III = II + 1                    00488
  ISN 0128                               CALL BCOORD(II,III,BLOCAL)   00489
                                     C                               00490
  ISN 0129                               DO 400 K = 1,3                 00491
  ISN 0130                               DO 400 L = 1,3                 00492
  ISN 0131   400 CLOCAL(K,L,IP) = BLOCAL(K,L)         00493
  ISN 0132                               GO TO 302                     00494
                                     C                               00495
  ISN 0133   202 CONTINUE                               00496
                                     C                               00497
  ISN 0134                               IIP=IP+1000                    00498
  ISN 0135                               CLOCAL(1,1,IP) = X1           00499
  ISN 0136                               CLOCAL(1,2,IP) = X2           00500
  ISN 0137                               CLOCAL(1,3,IP) = X3           00501
  ISN 0138                               CLOCAL(2,1,IP) = Y1           00502
  ISN 0139                               CLOCAL(2,2,IP) = Y2           00503
  ISN 0140                               CLOCAL(2,3,IP) = Y3           00504
  ISN 0141                               CLOCAL(3,1,IP) = Z1           00505
  ISN 0142                               CLOCAL(3,2,IP) = Z2           00506
  ISN 0143                               CLOCAL(3,3,IP) = Z3           00507
  ISN 0144   302 CONTINUE                               00508
                                     C                               00509
                                     C300 WRITE(6,300) IP           00510
                                     C300 FORMAT(5X,'CLOCAL FOR NODE',I5)
                                     C                               00511
                                     C301 WRITE(6,301)((CLOCAL(K,L,IP),L=1,3),K=1,3)
                                     C301 FORMAT(5X,3E12.5)           00512
                                     C                               00513
  ISN 0145   101 IP = IP + 1                           00514
  ISN 0146   100 CONTINUE                               00515
  ISN 0147                               RETURN                       00516
  ISN 0148                               END                           00517

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

STATISTICS SOURCE STATEMENTS = 147, PROGRAM SIZE = 2588, SUBPROGRAM NAME =CORD2R

STATISTICS NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

2992K BYTES OF CORE NOT USED

REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```

C
ISN 0002      SUBROUTINE ESTIFF(ISEC,NC,NR,I)                                00518
C                                                     00519*35
C                                                     00520
C THIS ROUTINE DETERMINES THE MATERIAL STIFFNESS MATRIX , E ,          00521
C FOR EACH BEAM . EVALUATED AT EACH END                                00522
C                                                     00523
ISN 0003      IMPLICIT REAL*8(A-H,O-Z)                                    00524
ISN 0004      COMMON /ESTIF/ E(8,6)                                       00525
ISN 0005      COMMON /QIJ/ Q(3,3,7)                                       00526
ISN 0006      COMMON /LAYER/ TH(7,25),BMASS(25),BSPAN(25),BWIDTH(25)      00527
ISN 0007      COMMON /INPUTT/ X(1000),YY(1000),Z(1000),T(1000)          00528*35
ISN 0008      COMMON /BLK 8/ SNB(21),ITL(18),VARI(235),TMAX(21),HALPHA(21) 00529*35
ISN 0009      DIMENSION Y(8)                                              00530
C                                                     00531
C DETERMINE THE LAYER THICKNESSES                                       00532
ISN 0010      Y(1) = -(TH(1,I) + TH(2,I) + TH(3,I) + TH(4,I)/2.)          00533
ISN 0011      DO 101 J = 2,8                                             00534
ISN 0012      101 Y(J) = Y(J-1) + TH(J-1,I)                               00535
C                                                     00536
C FIND A11,A22                                                           00537
C                                                     00538
ISN 0013      A11 = 0.                                                    00539
ISN 0014      A22 = 0.                                                    00540
ISN 0015      DO 102 J = 1,7                                             00541
ISN 0016      A11 = A11 + Q(1,1,J) * (Y(J+1) - Y(J))                    00542
ISN 0017      102 A22 = A22 + Q(2,2,J) * (Y(J+1) - Y(J))                00543
ISN 0018      A11 = A11 * EWIDTH(I)                                       00544
ISN 0019      A22 = A22 * BWIDTH(I)                                       00545
C                                                     00546
C FIND B11,B13                                                           00547
C                                                     00548
ISN 0020      B11 = 0.                                                    00549
ISN 0021      B13 = 0.                                                    00550
ISN 0022      DO 103 J = 1,7                                             00551
ISN 0023      B11 = B11 + Q(1,1,J) * (Y(J+1)**2 - Y(J)**2)              00552
ISN 0024      103 B13 = B13 + Q(1,3,J) * (Y(J+1)**2 - Y(J)**2)          00553
ISN 0025      B11 = B11 * BWIDTH(I) / 2.                                  00554
ISN 0026      B13 = B13 * BWIDTH(I) / 2.                                  00555
C ROUND OFF BIJ VALUES TO ZERO IF LESS THAN 10.                       00556
ISN 0027      CHECK = 10.                                                 00557
ISN 0028      IF(DABS(B11) .LT. CHECK) B11 = 0.                          00558
ISN 0030      IF(DABS(B13) .LT. CHECK) B13 = 0.                          00559
C                                                     00560
C FIND D11,D33,D13                                                       00561
C                                                     00562
ISN 0032      D11 = 0.                                                    00563
ISN 0033      D13 = 0.                                                    00564
ISN 0034      D33 = 0.                                                    00565
ISN 0035      DO 104 J = 1,7                                             00566
ISN 0036      D11 = D11 + Q(1,1,J) * (Y(J+1)**3 - Y(J)**3)              00567
ISN 0037      D13 = D13 + Q(1,3,J) * (Y(J+1)**3 - Y(J)**3)              00568
ISN 0038      104 D33 = D33 + Q(3,3,J) * (Y(J+1)**3 - Y(J)**3)          00569
ISN 0039      D11 = D11 * BWIDTH(I) / 3.                                  00570
ISN 0040      D13 = D13 * BWIDTH(I) / 3.                                  00571
ISN 0041      D33 = D33 * BWIDTH(I) / 3.                                  00572

```

104

```

ISN 0042      C      ROUND OFF D13 TO ZERO IF LESS THAN CHECK = 10.          00573
              C      IF(DABS(D13) .LT. CHECK) D13 = 0.                      00574
              C      WRITE(6,1) I,A11,A22,B11,B13,D11,D13,D33              00575
              C1     FORMAT(5X,'ESTIFF FOR BEAM',I5,5X,'A11,A22,B11,B13,D11,D13,D33', 00576
              C      1/,5X,7E10.3)                                         00577
              C      FORM THE E MATRIX                                       00578
              C      DO 105 J = 1,8                                          00579
              C      DO 105 JJ = 1,6                                          00580
              C      E(J,JJ) = 0.                                             00581
              C      E(1,1) = A11                                             00582
              C      E(1,2) = 2. * B13                                        00583
              C      E(1,3) = -B11                                           00584
              C      E(2,1) = B13                                             00585
              C      TOTAL = Y(8) - Y(1)                                       00586
              C      E(2,2) = 4. * D33                                        00587
              C      E(2,3) = -D13                                           00588
              C      E(3,1) = A11                                             00589
              C      E(3,2) = 2. * B13                                        00590*37
              C      E(3,4) = -B11                                           00591*35
              C      E(4,1) = B13                                             00592
              C      E(4,2) = E(2,2)                                          00593
              C      E(4,4) = -D13                                           00594
              C      E(5,1) = -B11                                           00595
              C      E(5,2) = -2. * D13                                        00596
              C      E(5,3) = D11                                             00597
              C      E(6,1) = -B11                                           00598*35
              C      E(6,2) = -2. * D13                                        00599
              C      E(6,4) = D11                                             00600
              C      E(7,5) = BWIDTH(I)**2/12. * A11                         00601
              C      E(8,6) = E(7,5)                                          00602
              C      ROOT WARPNG RESTRAINED ( (2,2) TERM NOT CHANGED)       00603
              C      IF(ISEC .GT. 1) RETURN                                    00604
              C      E(1,2) = E(1,2) / 2.                                     00605
              C      E(5,2) = E(5,2) / 2.                                     00606
              C      CALL MATPRN(E,8,6,' E ')                                  00607
              C      RETURN                                                    00608*41
              C      END                                                        00609*41
ISN 0068      C      IF(ISEC .GT. 1) RETURN                                    00610*41
ISN 0070      C      E(1,2) = E(1,2) / 2.                                     00611*41
ISN 0071      C      E(5,2) = E(5,2) / 2.                                     00612*41
              C      CALL MATPRN(E,8,6,' E ')                                  00613*41
              C      RETURN                                                    00614
ISN 0072      C      RETURN                                                    00615
ISN 0073      C      END                                                        00616
              C      RETURN                                                    00617
              C      END                                                        00618
              C      RETURN                                                    00619
              C      END                                                        00619

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
STATISTICS SOURCE STATEMENTS = 72, PROGRAM SIZE = 1290, SUBPROGRAM NAME =ESTIFF
STATISTICS NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

3004K BYTES OF CORE NOT USED

REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFCRMT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

ISN 0002	C	SUBROUTINE EXNECK (B)	00620
	C		00621*39
	C	EXTENDED NECK STIFFNESS - RECTANGULAR - TITANIUM	00622*39
	C	15 % THICK NECK	00623*39
	C		00624*39
ISN 0003	C	COMMON /STRS/ RBEE(12,12,25,21),FOUT(12,21),SL(25,21),SW(25,21)	00625*39
ISN 0004	C	COMMON /STIFF/ SK(12,12),SKK(12,12,21),SNM(6,6,21)	00626*39
ISN 0005	C	COMMON /BLKAA/ AA(21),XSC(21),YSC(21),XXX(42)	00627*39
ISN 0006	C	COMMON /BLK12/ XBARX(21),XIMAX(42),YBARY(21),TLTA(21)	00628*39
ISN 0007	C	DIMENSION RBE(12,12),RBET(12,12),D1(12,12)	00629*39
	C		00630*39
ISN 0008	C	T = 0.15 * B	00631*39
ISN 0009	C	RIMIN = B * T**3 / 12.	00632*39
ISN 0010	C	RIMAX = B**3 * T / 12.	00633*39
ISN 0011	C	TORS = B * T**3 / 3. - 0.21 * T**4 * (1. - T**4/12./B**4)	00634*39
ISN 0012	C	TORS = RIMIN + RIMAX	00635*39
	C		00636*39
ISN 0013	C	TORS = 1. * TORS	00637*41
	C		00638*41
ISN 0014	C	E = 16.1E6	00639*41
ISN 0015	C	G = 6.05E6	00640*39
ISN 0016	C	BL = SL(1,1)	00641*39
ISN 0017	C	AREA = B * T	00642*39
	C		00643*39
	C	WRITE(6,10) B,T,RIMIN,RIMAX,TORS,E,G,BL,AREA	00644*39
	C10	FORMAT(5X,'B,T,RIMIN,RIMAX,TORS,E,G,BL,AREA',/,5X,9E12.5)	00645*40
	C		00646*40
	C	FILL IN STIFFNESS ARRAY , SK	00647*39
	C		00648*39
	C		00649*39
ISN 0018	C	DO 100 I = 1,12	00650*39
ISN 0019	C	DO 100 J = 1,12	00651*39
ISN 0020	100	SK(I,J) = 0.	00652*39
	C		00653*39
ISN 0021	C	SK(1,1) = AREA * E / BL	00654*39
ISN 0022	C	SK(1,7) = -SK(1,1)	00655*39
ISN 0023	C	SK(2,2) = 12. * E * RIMAX / BL**3	00656*39
ISN 0024	C	SK(2,6) = 6. * E * RIMAX / BL**2	00657*39
ISN 0025	C	SK(2,8) = -SK(2,2)	00658*39
ISN 0026	C	SK(2,12) = SK(2,6)	00659*39
ISN 0027	C	SK(3,3) = 12. * E * RIMIN / BL**3	00660*39
ISN 0028	C	SK(3,5) = -6. * E * RIMIN / BL**2	00661*39
ISN 0029	C	SK(3,9) = -SK(3,3)	00662*39
ISN 0030	C	SK(3,11) = SK(3,5)	00663*39
ISN 0031	C	SK(4,4) = TORS * G / BL	00664*39
ISN 0032	C	SK(4,10) = -SK(4,4)	00665*39
ISN 0033	C	SK(5,5) = 4. * E * RIMIN / BL	00666*39
ISN 0034	C	SK(5,9) = -SK(3,11)	00667*39
ISN 0035	C	SK(5,11) = 2. * E * RIMIN / BL	00668*39
ISN 0036	C	SK(6,6) = 4. * E * RIMAX / BL	00669*39
ISN 0037	C	SK(6,8) = -SK(2,12)	00670*39
ISN 0038	C	SK(6,12) = 2. * E * RIMAX / BL	00671*39
ISN 0039	C	SK(7,7) = SK(1,1)	00672*39
ISN 0040	C	SK(8,8) = SK(2,2)	00673*39
ISN 0041	C	SK(8,12) = -SK(2,12)	00674*39

```

100
ISN 0042      SK(9,9) = SK(3,3)          00675*39
ISN 0043      SK(9,11) = -SK(3,11)      00676*39
ISN 0044      SK(10,10) = SK(4,4)       00677*39
ISN 0045      SK(11,11) = SK(5,5)       00678*39
ISN 0046      SK(12,12) = SK(6,6)       00679*39
C
C          SYMMETRY CONDITION           00680*39
C          00681*39
C          00682*39
ISN 0047      DO 110 I = 1,12           00683*39
ISN 0048      DO 110 J = 1,12           00684*39
ISN 0049      IF(J .GE. I) GO TO 110    00685*39
ISN 0051      SK(I,J) = SK(J,I)         00686*39
ISN 0052      110 CONTINUE              00687*39
C          00688*39
C          CALL MATPRN(SK,12,12,'SKRT') 00689*40
C          00690*39
C          TRANSLATE TO CG LOCATION      00691*39
C          00692*39
ISN 0053      DZ = XSC(2) - XBARX(2)    00693*46
ISN 0054      DY = YBARY(2) - YSC(2)    00694*46
C          WRITE(6,11) DZ,DY            00695*40
C11          FCRIAT(5X,'DZ,DY',2E12.5)  00696*40
C          00697*39
ISN 0055      DO 200 I = 1,12           00698*39
ISN 0056      DO 200 J = 1,12           00699*39
ISN 0057      RBE(I,J) = 0.             00700*39
ISN 0058      IF(I .EQ. J) RBE(I,J) = 1. 00701*39
ISN 0060      200 CONTINUE              00702*39
C          00703*39
ISN 0061      RBE(1,5) = DZ             00704*39
ISN 0062      RBE(1,6) = -DY            00705*39
ISN 0063      RBE(2,4) = -DZ            00706*39
ISN 0064      RBE(3,4) = DY             00707*39
ISN 0065      RBE(7,11) = DZ            00708*39
ISN 0066      RBE(7,12) = -DY           00709*39
ISN 0067      RBE(8,10) = -DZ           00710*39
ISN 0068      RBE(9,10) = DY            00711*39
C          00712*39
C          CALL MATPRN(RBE,12,12,'RBE ') 00713*40
C          00714*39
C          RBE TRANSPOSE , RBET         00715*39
C          00716*39
ISN 0069      DO 210 I = 1,12           00717*39
ISN 0070      DO 210 J = 1,12           00718*39
ISN 0071      210 RBET(I,J) = RBE(J,I)  00719*39
C          CALL MATPRN(RBET,12,12,'RBET') 00720*40
C          00721*39
ISN 0072      CALL MATMPY(SK,RBE,D1,12,12,12) 00722*39
ISN 0073      CALL MATMPY(RBET,D1,SK,12,12,12) 00723*39
C          00724*39
C          CALL MATPRN(SK,12,12,'SKRT')  00725*40
C          00726*39
ISN 0074      RETURN                    00727*39
ISN 0075      END                        00728*39

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

STATISTICS SOURCE STATEMENTS = 74, PROGRAM SIZE = 4850, SUBPROGRAM NAME =EXNECK

STATISTICS NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

3004K BYTES OF CORE NOT USED

REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE
 OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
 SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```

C                                00729*39
C                                00730
ISH 0002      C      SUBROUTINE FRPM(J,IP,X,Y,BMASS,RPM,FBEAM,BLOCAL) 00731
C                                00732
C      CALCULATING THE FORCES DUE TO RPM ON NODE IP FOR BEAM J 00733
C                                00734
ISH 0003      C      DOUBLE PRECISION X(1),Y(1),BMASS(1),RPM,FBEAM(6,1),BLOCAL(3,3) 00735
ISH 0004      C      DOUBLE PRECISION F(3,1),FF(3,1),A(6) 00736
C                                00737
ISH 0005      C      DO 100 K = 3,6 00738
ISH 0006      C      A(K) = 0. 00739
ISH 0007      C      100 FBEAM(K,1) = 0. 00740
C                                00741
C      NONZERO ACCELERATION COMPONENTS 00742
C                                00743
ISH 0008      C      A(1) = RPM**2 * X(IP) 00744
ISH 0009      C      A(2) = RPM**2 * Y(IP) 00745
C                                00746
C      FORCES 00747
C                                00748
ISH 0010      C      F(1,1) = BMASS(J)/2. * A(1) 00749
ISH 0011      C      F(2,1) = BMASS(J)/2. * A(2) 00750
ISH 0012      C      F(3,1) = 0. 00751
C                                00752
C      WRITE(6,20) J,IP 00753
C20     C      FORMAT(5X,'** IN FRPM-GLOBAL RPM LOADS AT BEAM,END',2I5) 00754
C      CALL MATPRN(F,3,1,' F ') 00755
C                                00756
C      TRANSFORM TO LOCAL BEAM COOR. SYSTEM 00757
C                                00758
ISH 0013      C      CALL MATMPY(BLOCAL,F,FF,3,3,1) 00759
C                                00760
C      INSERT INTO FBEAM 00761
C                                00762
ISH 0014      C      DO 200 K = 1,3 00763
ISH 0015      C      200 FBEAM(K,1) = FF(K,1) 00764
C                                00765
C      WRITE(6,1) J,IP 00766
C1     C      FORMAT(5X,'** IN FRPM **,BEAM AND NODE',2I5) 00767
C      WRITE(6,2) RPM,BMASS(J),A(1),A(2) 00768
C2     C      FORMAT(5X,'RPM,BMASS,A1,A2',4E10.3) 00769
C      CALL MATPRN(FBEAM,6,1,'FBEM') 00770
C                                00771
ISH 0016      C      RETURN
ISH 0017      C      END
    
```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
 *OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
 STATISTICS SOURCE STATEMENTS = 16, PROGRAM SIZE = 706, SUBPROGRAM NAME = FRPM
 STATISTICS NO DIAGNOSTICS GENERATED
 ***** END OF COMPILATION *****

3016K BYTES OF CORE NOT USED

108

ISN 0002	C	SUBROUTINE HOLZER(LL,NR,NC,RPM)	00772
	C		00773
	C	THIS ROUTINE CALCULATES ALL SECTION STIFFNESSES AND MASSES	00774
	C	AND THEN TRANSFORMS THESE INTO HOLZER MATRICES	00775
	C		00776
	C		00777
ISN 0003		IMPLICIT REAL*8(A-H,O-Z)	00778
ISN 0004		COMMON /STIFF/ SK(12,12),SKK(12,12,21),SMH(6,6,21)	00779
ISN 0005		COMMON /REORDR/ P(12,12),PT(12,12)	00780
ISN 0006		COMMON /HSTIFF/ SKP(12,12),SKPP(12,12,21)	00781
ISN 0007		COMMON /INPUTT/ X(1000),Y(1000),Z(1000),T(1000)	00782
ISN 0008		DIMENSION SK11(6,6),SK12(6,6),SK21(6,6),SK22(6,6)	00783
ISN 0009		DIMENSION RL(6),RM(6),SK1(6,6),D1(12,12),DD(6,6)	00784
	C		00785
	C	BEGIN THE SECTION STIFFNESS SOLUTION ,MATRIX SK	00786
	C		00787
ISN 0010		NRMI = NR - 1	00788
ISN 0011		ISKIP = 1	00789
ISN 0012		DO 200 I = 1,NRMI	00790
ISN 0013		IF(LL .EQ. 1) GO TO 99	00791
ISN 0015		GO TO 98	00792
	C		00793
	C	GENERATE THE SECTION STIFFNESS , SK	00794
	C	SECTION MASS , SMH	00795
	C	THE BEAM THICKNESSES ,MASSES AND LENGTHS ARE CALCULATED	00796
	C	THE WARPING FUNCTION WILL ALSO BE EVALUATED	00797
	C		00798
ISN 0016	99	CALL THICK(I,NR,NC,X,Y,Z,T,ISKIP)	00799
ISN 0017		IF (I .EQ. 1) CALL WARP(I,NR,NC)	00800
ISN 0019		II = I + 1	00801
ISN 0020		CALL WARP(II,NR,NC)	00802
ISN 0021		CALL SECTN(I,NR,NC,RPM)	00803
ISN 0022		GO TO 97	00804
	C		00805
	C	THIS SECTION IS FOR SECOND STATIC ANALYSIS WITH PRESTRESS	00806
	C		00807
ISN 0023	98	CONTINUE	00808
ISN 0024		DO 100 J = 1,12	00809
ISN 0025		DO 100 K = 1,12	00810
ISN 0026	100	SK(J,K) = SKK(J,K,I)	00811
ISN 0027	97	CONTINUE	00812
	C	CALL MATPRN(SK,12,12,' SK '	00813
	C		00814
	C	PARTITION THE SECTION STIFFNESS MATRIX ,SK, INTO:	00815
	C	SK11 , SK12 , SK21 , SK22	00816
	C		00817
ISN 0028		CALL PARTN(SK,12,12,SK11,SK12,SK21,SK22,6,6)	00818
	C		00319
	C	INVERT SK12	00820
	C		00821
ISN 0029		CALL MINV(SK12,6,D,RL,RM)	00822
	C		00823
	C	FORM THE SKP MATRIX FROM PARTITIONED SK COMPONENTS	00824
	C		00825
ISN 0030		CALL MATMPY(SK12,SK11,SK1,6,6,6)	00826

```

C          00827
C          CHANGE SIGN ON UPPER QUARTER OF SKP          00828
C          00829
ISN 0031      DO 115 K = 1,6          00830
ISN 0032      DO 115 L = 1,6          00831
ISN 0033      115 SKP(K,L) = -SK1(K,L)          00832
C          00833
ISN 0034      DO 116 K = 1,6          00834
ISN 0035      DO 116 L = 7,12         00835
ISN 0036      LN = L - 6              00836
ISN 0037      116 SKP(K,L) = SK12(K,LM)          00837
C          00838
C          THE SIGN ON SKP21 AND SKP22 ARE NEGATIVE FOR EQUIL. 00839
C          00840
ISN 0038      CALL MATMPY(SK22,SK1,SK11,6,6,6)          00841
ISN 0039      DO 117 K = 7,12         00842
ISN 0040      KK = K - 6              00843
ISN 0041      DO 117 L = 1,6          00844
ISN 0042      117 SKP(K,L) = SK11(KK,L) - SK21(KK,L) 00845
C          00846
ISN 0043      CALL MATMPY(SK22,SK12,SK1,6,6,6)          00847
C          00848
ISN 0044      DO 125 K = 7,12         00849
ISN 0045      KK = K - 6              00850
ISN 0046      JJ = 1                  00851
ISN 0047      DO 125 L = 7,12         00852
ISN 0048      SKP(K,L) = -SK1(KK,JJ)          00853
ISN 0049      125 JJ = JJ + 1          00854
C          00855
C          FILL THE SKPP ARRAY WITH THE SKP MATRIX, THIS SAVES SKP 00856
C          00857
ISN 0050      DO 165 K = 1,12          00858
ISN 0051      DO 165 L = 1,12          00859
ISN 0052      165 SKPP(K,L,I) = SKP(K,L)          00860
C          00861
C          WRITE(6,101) I              00862
C101      FORMAT(/,5X,'SKP IS THE HOLZER STIFFNESS FOR SECTION',I5) 00863
C          CALL MATPRN(SKP,12,12,'SKP ')          00864
C          00865
ISN 0053      200 CONTINUE              00866
C          00867
ISN 0054      RETURN                    00868
ISN 0055      END                        00869

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 54, PROGRAM SIZE = 2792, SUBPROGRAM NAME =HOLZER
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

3004K BYTES OF CORE NOT USED

REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUIT(60) SIZE(MAX) AUTODBL(DBL4)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

110

	C		00870
ISH 0002		SUBROUTINE INPUT (NR,NC)	00871
	C		00872
	C	MATERIAL PROPERTIES AND GEOMETRY INPUT	00873
	C		00874
ISH 0003		IMPLICIT REAL*8(A-H,O-Z)	00875
ISH 0004		COMMON /ZCOEF/ E11(7),E22(7),E33(7),G12(7),G23(7),G13(7),	00876
	1	V12(7),V13(7),V23(7)	00877
ISH 0005		COMMON /PLY1/ PLY(21,25,7),THETA(7),RHO(7)	00878
ISH 0006		COMMON /FAIL/ X1T(7),X1C(7),X2T(7),X2C(7),S6P(7),S6M(7),TSAI(8,25)	00879*35
	C		00880
	C	WRITE(8,1)	00881
	C1	FORMAT(5X,'IS THIS AN ISOTROPIC BLADE, 1=YES')	00882
	C	READ(8,*) ISO	00883
	C	IF(ISO .NE. 1) GO TO 2	00884
	C		00885
	C	ISOTROPIC BLADE	00886
	C		00887
	C	WRITE(8,3)	00888
	C3	FORMAT(5X,'INPUT-E(Psi),V AND DENSITY(LB/IN3),FREE FORMAT')	00889
	C	READ(8,*) E,V,R	00890
ISH 0007		E = 16100000.	00891
ISH 0008		V = .33	00892
ISH 0009		R = .16	00893
ISH 0010		DPLY = 1. / 7.	00894
ISH 0011		YIELD = 110000.	00895*35
ISH 0012		SHEAR = 0.577 * YIELD	00896*36
ISH 0013		DO 100 I = 1,7	00897
	C		00898*35
ISH 0014		X1T(I) = YIELD	00899*35
ISH 0015		X1C(I) = YIELD	00900*35
ISH 0016		X2T(I) = YIELD	00901*35
ISH 0017		X2C(I) = YIELD	00902*35
ISH 0018		S6P(I) = SHEAR	00903*35
ISH 0019		S6M(I) = SHEAR	00904*35
ISH 0020		E11(I) = E	00905
ISH 0021		E22(I) = E	00906
ISH 0022		E33(I) = E	00907
ISH 0023		G12(I) = E / 2. / (1. + V)	00908
ISH 0024		G13(I) = G12(I)	00909
ISH 0025		G23(I) = G12(I)	00910
ISH 0026		V12(I) = V	00911
ISH 0027		V13(I) = V	00912
ISH 0028		V23(I) = V	00913
ISH 0029	100	RHO(I) = R / 386.4	00914
ISH 0030		RHO(3) = 0.	00915*35
ISH 0031		E11(3) = 0.	00916*35
ISH 0032		RHO(4) = 0.	00917*35
ISH 0033		E11(4) = 0.	00918*35
ISH 0034		RHO(5) = 0.	00919*35
ISH 0035		E11(5) = 0.	00920*35
	C		00921*39
	C	INPUT MATERIAL STUFF FOR TI/BORON LAYER	00922*39
	C		00923*39
ISH 0036		E11(2) = 33.2E6	00924*41

ISN 0037	E11(6) = E11(2)	00925*41
ISN 0038	E22(2) = 29.0E6	00926*41
ISN 0039	E22(6) = E22(2)	00927*41
ISN 0040	E33(2) = E22(2)	00928*41
ISN 0041	E33(6) = E33(2)	00929*41
ISN 0042	G12(2) = 22.5E6	00930*41
ISN 0043	G12(6) = G12(2)	00931*41
ISN 0044	G13(2) = G12(2)	00932*41
ISN 0045	G13(6) = G13(2)	00933*41
ISN 0046	G23(2) = G12(2)	00934*41
ISN 0047	G23(6) = G23(2)	00935*41
ISN 0048	V12(2) = 0.3	00936*41
ISN 0049	V12(6) = 0.3	00937*41
ISN 0050	RHO(2) = .000336	00938*41
ISN 0051	RHO(6) = RHO(2)	00939*42
ISN 0052	V13(2) = V12(2)	00940*41
ISN 0053	V13(6) = V13(2)	00941*41
ISN 0054	V23(2) = V12(2)	00942*41
ISN 0055	V23(6) = V23(2)	00943*41
	C THETA(2) = 35./180. * 3.14159265	00944*39
	C THETA(6) = THETA(2)	00945*39
ISN 0056	YIELD1 = 170000.	00946*41
ISN 0057	YIELD2 = 50000.	00947*41
ISN 0058	YIELDS = 30000.	00948*41
ISN 0059	X1T(2) = YIELD1	00949*41
ISN 0060	X1T(6) = YIELD1	00950*41
ISN 0061	X1C(2) = YIELD1	00951*41
ISN 0062	X1C(6) = YIELD1	00952*41
ISN 0063	X2T(2) = YIELD2	00953*41
ISN 0064	X2T(6) = YIELD2	00954*41
ISN 0065	X2C(2) = YIELD2	00955*41
ISN 0066	X2C(6) = YIELD2	00956*41
ISN 0067	S6P(2) = YIELDS	00957*41
ISN 0068	S6P(6) = YIELDS	00958*41
ISN 0069	S6M(2) = YIELDS	00959*41
ISN 0070	S6M(6) = YIELDS	00960*41
		00961*41
		00962
		00963
ISN 0071	RETURN	00964
ISN 0072	END	00965

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
STATISTICS SOURCE STATEMENTS = 71, PROGRAM SIZE = 936, SUBPROGRAM NAME = INPUT
STATISTICS NO DIAGNOSTICS GENERATED
***** END OF COMPILATION ***** 3004K BYTES OF CORE NOT USED

112

	C		00966
ISN 0002		SUBROUTINE KDGG(ISEC,FF,FOUT1,DSKK)	00967
	C		00968
	C	CALCULATES THE DIFFERENTIAL STIFFNESS MATRIX FOR SECTION	00969
	C	ISEC , DSKK, (SEE THE NASTRAN THEORETICAL MANUAL FOR	00970
	C	EXPLANATION 7.2-8)	00971
	C		00972
ISN 0003		IMPLICIT REAL*8(A-H,O-Z)	00973
ISN 0004		COMMON /SC/XSC(21),YSC(21),ZSC(21),ALPHA(21),SECIP(21),SECA(21)	00974
ISN 0005		DIMENSION FF(12,1),FOUT1(12,1),DSK(12,12),DSKK(12,12,20)	00975
ISN 0006		REAL MYA,MYB,MZA,MZB	00976
	C		00977
	C	CALCULATE SECTION LENGTH ,SL.	00978
	C		00979
ISN 0007		X = XSC(ISEC+1) - XSC(ISEC)	00980
ISN 0008		Y = YSC(ISEC+1) - YSC(ISEC)	00981
ISN 0009		Z = ZSC(ISEC+1) - ZSC(ISEC)	00982
ISN 0010		SL = DSQRT(X**2 + Y**2 + Z**2)	00983
	C		00984
	C	DEFINE TERMS CONSISTANT WITH NASTRAN TERMINOLOGY	00985
	C		00986
ISN 0011		FX = FOUT1(7,1)	00987
ISN 0012		VY = FOUT1(8,1)	00988
ISN 0013		VZ = FOUT1(9,1)	00989
ISN 0014		MYB =-FF(11,1)	00990
ISN 0015		MYA =-FOUT1(11,1)	00991
ISN 0016		MZB =-FF(12,1)	00992
ISN 0017		MZA =-FOUT1(12,1)	00993
	C		00994
	C	THE STIFFNESS TERMS IN THE DSKK MATRIX HAVE BEEN SET TO ZERO	00995
	C	BECAUSE THE MOMENT OF INERTIA TERM IS UNKNOWN FOR A COMPOSITE	00996
	C		00997
ISN 0018		DO 100 I = 1,12	00998
ISN 0019		DO 100 J = 1,12	00999
ISN 0020	100	DSKK(I,J,ISEC) = 0.	01000
	C		01001
ISN 0021		DSKK(2,2,ISEC) = 6./5./SL * FX	01002
ISN 0022		DSKK(2,4,ISEC) = -MYB / SL	01003
ISN 0023		DSKK(2,6,ISEC) = FX / 10.	01004
ISN 0024		DSKK(2,8,ISEC) = -DSKK(2,2,ISEC)	01005
ISN 0025		DSKK(2,10,ISEC) = -MYA / SL	01006
ISN 0026		DSKK(2,12,ISEC) = FX / 10.	01007
ISN 0027		DSKK(3,3,ISEC) = DSKK(2,2,ISEC)	01008
ISN 0028		DSKK(3,4,ISEC) = -MZB / SL	01009
ISN 0029		DSKK(3,5,ISEC) = -FX / 10.	01010
ISN 0030		DSKK(3,9,ISEC) = -DSKK(2,2,ISEC)	01011
ISN 0031		DSKK(3,10,ISEC) = -MZA / SL	01012
ISN 0032		DSKK(3,11,ISEC) = -FX / 10.	01013
ISN 0033		DSKK(4,4,ISEC) = SECIP(ISEC) * FX / SECA(ISEC) / SL	01014
ISN 0034		DSKK(4,5,ISEC) = - SL * VY / 6.	01015
ISN 0035		DSKK(4,6,ISEC) = - SL * VZ / 6.	01016
ISN 0036		DSKK(4,8,ISEC) = MYB / SL	01017
ISN 0037		DSKK(4,9,ISEC) = MZB / SL	01018
ISN 0038		DSKK(4,10,ISEC) = - DSKK(4,4,ISEC)	01019
ISN 0039		DSKK(4,11,ISEC) = SL * VY / 6.	01020

1
11

ISN 0040	DSKK(4,12,ISEC) = SL * VZ / 6.	01021
ISN 0041	DSKK(5,5,ISEC) = SL / 7.5 * FX	01022
ISN 0042	DSKK(5,9,ISEC) = FX / 10.	01023
ISN 0043	DSKK(5,10,ISEC) = SL * VY / 6.	01024
ISN 0044	DSKK(5,11,ISEC) = - SL / 30. * FX	01025
ISN 0045	DSKK(6,6,ISEC) = SL / 7.5 * FX	01026
ISN 0046	DSKK(6,8,ISEC) = -FX / 10.	01027
ISN 0047	DSKK(6,10,ISEC) = SL * VZ / 6.	01028
ISN 0048	DSKK(6,12,ISEC) = - SL / 30. * FX	01029
ISN 0049	DSKK(8,8,ISEC) = DSKK(2,2,ISEC)	01030
ISN 0050	DSKK(8,10,ISEC) = MYA / SL	01031
ISN 0051	DSKK(8,12,ISEC) = -FX / 10.	01032
ISN 0052	DSKK(9,9,ISEC) = DSKK(2,2,ISEC)	01033
ISN 0053	DSKK(9,10,ISEC) = MZA / SL	01034
ISN 0054	DSKK(9,11,ISEC) = FX / 10.	01035
ISN 0055	DSKK(10,10,ISEC) = DSKK(4,4,ISEC)	01036
ISN 0056	DSKK(10,11,ISEC) = - SL / 6. * VY	01037
ISN 0057	DSKK(10,12,ISEC) = - SL / 6. * VZ	01038
ISN 0058	DSKK(11,11,ISEC) = SL / 7.5 * FX	01039
ISN 0059	DSKK(12,12,ISEC) = SL / 7.5 * FX	01040
	C	01041
ISN 0060	DO 110 I = 1,12	01042
ISN 0061	DO 110 J = 1,12	01043
ISN 0062	IF(J .LE. I) GO TO 110	01044
ISN 0064	DSKK(J,I,ISEC) = DSKK(I,J,ISEC)	01045
ISN 0065	110 CONTINUE	01046
	C	01047
ISN 0066	DO 200 I = 1,12	01048
ISN 0067	DO 200 J = 1,12	01049
ISN 0068	200 DSK(I,J) = DSKK(I,J,ISEC)	01050
	C	01051
	CALL MATPRN(DSK,12,12,'DSKK')	01052
	C	01053
ISN 0069	RETURN	01053
ISN 0070	END	01054

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTCDBL(DBL4)
 *OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
 STATISTICS SOURCE STATEMENTS = 69, PROGRAM SIZE = 2544, SUBPROGRAM NAME = KDGG
 STATISTICS NO DIAGNOSTICS GENERATED
 ***** END OF COMPILATION *****

3004K BYTES OF CORE NOT USED

REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

SOURCE EBCDIC NOLIST NODECK OBJECT HOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```

114
C      01055
ISN 0002 C      SUBROUTINE LAMINA                                01056
C      01057
C      THIS ROUTINE GENERATES THE LAMINA STRESS-STRAIN MATRIX , Q 01058
C      01059
ISN 0003 C      IMPLICIT REAL*8(A-H,O-Z)                        01060
ISN 0004 C      COMMON /ZCOEF/ E11(7),E22(7),E33(7),G12(7),G23(7),G13(7), 01061
C      01062
C      1      V12(7),V13(7),V23(7)
ISN 0005 C      COMMON /PLY1/ PLY(21,25,7),THETA(7),RHO(7)      01063
ISN 0006 C      COMMON /QIJ/ Q(3,3,7)                               01064
C      01065
ISN 0007 C      N = 7                                           01066
ISN 0008 C      DO 100 I = 1,N                                     01067
ISN 0009 C      WRITE(6,1) I                                     01068
ISN 0010 C      1      FORMAT(/,5X,'LAMINA, LAYER ',I4,/)      01069
C      01070*34
C      CHECK FOR HOLLOW LAYER                                  01071*34
C      01072*34
C      01073*34
ISN 0011 C      IF(E11(I) .EQ. 0.) GO TO 200                    01074
C      DETERMINE THE POISSON RATIOS                            01075
ISN 0013 C      V21 = V12(I) * E22(I) / E11(I)                   01076
ISN 0014 C      V31 = V13(I) * E33(I) / E11(I)                   01077
ISN 0015 C      V32 = V23(I) * E33(I) / E22(I)                   01078
C      01079
ISN 0016 C      DET = 1. - V12(I)*V21 - V13(I)*V31 - V23(I)*V32 - 01080
C      1      V21*V13(I)*V32 - V31*V12(I)*V23(I)
ISN 0017 C      WRITE(6,2) V21,V31,V32,DET                       01081
ISN 0018 C      2      FORMAT(5X,4E12.5,/)                          01082
C      01083
C      C11 = E11(I) * (1. - V23(I)*V32) / DET                  01084
ISN 0019 C      C12 = E22(I) * (V12(I) + V13(I)*V32) / DET        01085
ISN 0020 C      C13 = E33(I) * (V13(I) + V12(I)*V23(I)) / DET    01086
ISN 0021 C      C22 = E22(I) * (1. - V13(I)*V31) / DET          01087
ISN 0022 C      C23 = E22(I) * (V32 + V31*V12(I)) / DET         01088
ISN 0023 C      C33 = E33(I) * (1. - V12(I)*V21) / DET          01089
ISN 0024 C      C44 = G23(I)                                       01090
ISN 0025 C      C55 = G13(I)                                       01091
ISN 0026 C      C66 = G12(I)                                       01092
ISN 0027 C      WRITE(6,3) C11,C12,C13                             01093
ISN 0028 C      3      FORMAT(5X,3E12.5)                          01094
ISN 0029 C      WRITE(6,4) C22,C23                                 01095
ISN 0030 C      4      FORMAT(17X,2E12.5)                         01096
ISN 0031 C      WRITE(6,5) C33                                     01097
ISN 0032 C      5      FORMAT(29X,E12.5,/)                          01098
ISN 0033 C      01099
C      ROTATE THRU ANGLE THETA (I) ABOUT THE 3 AXIS          01100
C      01101
C      C =DCOS( THETA(I) )                                     01102
ISN 0034 C      S =DSIN( THETA(I) )                               01103
ISN 0035 C      01104
C      CP11 = C**4*C11 + 2.*C**2*S**2*(C12+2.*C66) + S**4*C22 01105
ISN 0036 C      CP12 = C**2*S**2*(C11+C22-4.*C66) + (C**4+S**4)*C12 01106
ISN 0037 C      CP13 = C**2*C13 + S**2*C23                       01107
ISN 0038 C      CP16 = -C*S*(C**2*C11-S**2*C22) + C*S*(C**2-S**2)*(C12+2.*C66) 01108
ISN 0039 C      CP22 = S**4*C11 + 2.*C**2*S**2*(C12+2.*C66) + C**4*C22 01109

```

```

ISN 0041      CP23 = S**2*C13 + C**2*C23      01110
ISN 0042      CP26 = -C*S*(S**2*C11-C**2*C22) - S*C*(C**2-S**2)*(C12+2.*C66) 01111
ISN 0043      CP33 = C33                      01112
ISN 0044      CP36 = C*S*(C23-C13)            01113
ISN 0045      CP44 = C**2*C44 + S**2*C55      01114
ISN 0046      CP45 = C*S*(C44-C55)            01115
ISN 0047      CP55 = C**2*C55 + S**2*C44      01116
ISN 0048      CP66 = C**2*S**2*(C11+C22-2.*C12) + (C**2-S**2)**2*C66 01117

C
ISN 0049      CHECK = 10.                     01118
ISN 0050      IF(DABS(CP16) .LT. CHECK) CP16 = 0. 01119
ISN 0052      IF(DABS(CP26) .LT. CHECK) CP26 = 0. 01120
ISN 0054      IF(DABS(CP36) .LT. CHECK) CP36 = 0. 01121
ISN 0056      IF(DABS(CP45) .LT. CHECK) CP45 = 0. 01122

C
ISN 0058      WRITE(6,6) CP11,CP12,CP13,CP16 01123
ISN 0059      6  FORMAT(1X,3E12.5,24X,E12.5) 01124
ISN 0050      WRITE(6,7) CP22,CP23,CP26      01125
ISN 0061      7  FORMAT(13X,2E12.5,24X,E12.5) 01126
ISN 0062      WRITE(6,8) CP33,CP36          01127
ISN 0063      8  FORMAT(25X,E12.5,24X,E12.5) 01128
ISN 0064      WRITE(6,9) CP44,CP45          01129
ISN 0065      9  FORMAT(37X,2E12.5)         01130
ISN 0066      WRITE(6,10) CP55              01131
ISN 0067      10 FORMAT(49X,E12.5)          01132
ISN 0068      WRITE(6,11) CP66              01133
ISN 0069      11 FORMAT(61X,E12.5,/)       01134

C
ISN 0070      Z1 = (CP23 * CP13 - CP12 * CP33) / (CP22 * CP33 - CP23**2) 01135
ISN 0071      Z2 = (CP26 * CP33 - CP23 * CP36) / (CP22 * CP33 - CP23**2) 01136
ISN 0072      Z3 = (-CP13 - CP23 * Z1) / CP33 01137
ISN 0073      Z4 = (CP36 - CP23 * Z2) / CP33 01138

C
ISN 0074      Q(1,1,I) = CP11 + CP13 * Z3 + CP12 * Z1 01139
ISN 0075      Q(1,2,I) = 0. 01140
ISN 0076      Q(1,3,I) = CP13 * Z4 + CP12 * Z2 - CP16 01141
ISN 0077      Q(2,1,I) = 0. 01142
ISN 0078      Q(2,2,I) = CP55 01143
ISN 0079      Q(2,3,I) = 0. 01144
ISN 0080      Q(3,1,I) = Q(1,3,I) 01145
ISN 0081      Q(3,2,I) = 0. 01146
ISN 0082      Q(3,3,I) = -CP36 * Z4 - CP26 * Z2 + CP66 01147

C
C HOLLOW LAYER CONDITION 01148
C
ISN 0083      GO TO 201 01149
ISN 0084      200 CONTINUE 01150
ISN 0085      DO 210 K = 1,3 01151
ISN 0086      DO 210 L = 1,3 01152
ISN 0087      210 Q(K,L,I) = 0. 01153
ISN 0088      201 CONTINUE 01154

C
C
ISN 0089      WRITE(6,13) 01155
ISN 0090      13  FORMAT(5X,'QIJ') 01156
ISN 0091      WRITE(6,12) ((Q(II,JJ,I),JJ=1,3),II=1,3) 01157
ISN 0092      12  FORMAT(5X,3E12.5) 01158
ISN 0093      100 CONTINUE 01159

C

```

+VERSION 1.3.0 (01 MAY 80) LAMINA SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 82.141/10.48.54 PAGE 3
ISH 0094 RETURN 01169
ISH 0095 END 01170
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OSJECT NCMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
STATISTICS SOURCE STATEMENTS = 94, PROGRAM SIZE = 2554, SUBPROGRAM NAME =LAMINA
STATISTICS NO DIAGNOSTICS GENERATED
***** END OF COMPILATION ***** 2996K BYTES OF CORE NOT USED

116

REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

SOURCE EBCDIC NOLIST NODECK OBJECT NOHAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

	C		01171*41
ISN 0002		SUBROUTINE LAMINI (TSKIN,TCENTR,PBT,PGE,NC,NR)	01172*41
	C		01173*41
	C	PREPROCESSOR FOR THE COMPOSITE BLADE	01174*41
	C	INPUT : TSKIN = TITANIUM SKIN THICKNESS IN INCHES	01175*41
	C	TCENTR = TI CENTER LAYER THICKNESS IN INCHES	01176*41
	C	PBT = PERCENTAGE OF REMAINING THICKNESS OF BORON/TI	01177*41
	C	PGE = PERCENTAGE OF REMAINING THICKNESS OF GRAPHITE/EPOO	01178*41
	C	NOTE ** PBT + PGE = 1.0	01179*41
	C		01180*41
ISN 0003		IMPLICIT REAL*8(A-H,O-Z)	01181*41
ISN 0004		COMMON /PLY1/ PLY(21,25,7),THETA(7),RHO(7)	01182*41
ISN 0005		COMMON /INPUTT/ X(1000),Y(1000),Z(1000),T(1000)	01183*41
	C		01184*41
ISN 0006		NRM1 = NR - 1	01185*41
ISN 0007		DO 100 I = 1,NRM1	01186*41
ISN 0008		NODE = (I-1) * NC + 1	01187*41
ISN 0009		DO 100 J = 1,NC	01188*41
	C		01189*41
ISN 0010		TAVE = (T(NODE) + T(NODE+NC)) / 2.	01190*41
	C		01191*41
	C	CHECKON TOTAL TITANIUM LAYER THICKNESS	01192*41
	C		01193*41
ISN 0011		TTI = 2. * TSKIN + TCENTR	01194*41
ISN 0012		TCHECK = TAVE - TTI	01195*41
	C		01196*41
	C	INITIALIZE THICKNESSES	01197*41
	C		01198*41
ISN 0013		TS = TSKIN	01199*41
ISN 0014		TC = TCENTR	01200*41
ISN 0015		TBT = 0.	01201*41
ISN 0016		TGE = 0.	01202*41
	C		01203*41
ISN 0017		IF(TCHECK) 110,110,120	01204*41
	C		01205*41
	C	NO G/E OR TBT LAYERS DUE TO MINIMUM THICKNESS EXCEEDED	01206*41
	C		01207*41
ISN 0018	110	TCHECK = TAVE - 2. * TSKIN	01208*41
ISN 0019		IF(TCHECK) 111,111,112	01209*41
ISN 0020	111	TS = TAVE / 2.	01211*41
ISN 0021		TC = 0.	01212*41
ISN 0022		GO TO 200	01213*41
ISN 0023	112	TC = TAVE - 2. * TS	01214*41
ISN 0024		GO TO 200	01215*41
	C		01216*41
	C	INSERT G/E AND TBT LAYERS	01217*41
	C		01218*41
ISN 0025	120	TBT = PBT * TCHECK / 2.	01219*42
ISN 0026		TGE = PGE * TCHECK / 2.	01220*42
ISN 0027	200	CONTINUE	01221*41
	C		01222*41
	C	FILL THE PLY ARRAY	01223*41
	C		01224*41
ISN 0028		PLY(I,J,1) = TS	01225*41
ISN 0029		PLY(I,J,2) = TBT	01226*41

```

+VERSION 1.3.0 (01 MAY 80) LAMINI SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 82.141/10.49.01 PAGE 2
  ISN 0030 PLY(I,J,3) = TGE 01227*41
  ISN 0031 PLY(I,J,4) = TC 01228*41
  ISN 0032 PLY(I,J,5) = TGE 01229*41
  ISN 0033 PLY(I,J,6) = TBT 01230*41
  ISN 0034 PLY(I,J,7) = TS 01231*42
          C 01232*41
          C WRITE(6,300) I,J,NODE,TAVE,TS,TBT,TGE,TC 01233*42
          C300 FORMAT(5X,'I,J,NODE7TAVE,TS,TBT,TGE,TC',/,5X, 01234*42
          C 1 3I5,5E12.5) 01235*42
          C 01236*42
          C TOTAL = 2. * (TS + TBT + TGE) + TC 01237*42
          C WRITE(6,301) TOTAL 01238*42
          C301 FORMAT(5X,'TOTAL SUMMED THICKNESS =',E12.5) 01239*42
          C 01240*42
          C 01241*41
  ISN 0035 100 NODE = NODE + 1 01242*41
          C 01243*41
  ISN 0036 RETURN 01244*41
  ISN 0037 END 01245*41
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 36, PROGRAM SIZE = 922, SUBPROGRAM NAME =LAMINI
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

3016K BYTES OF CORE NOT USED

REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE
 OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
 SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```

C                                     01246*41
C                                     01247
ISH 0002      SUBROUTINE LOAD2(NR,FF)    01248
C                                     01249
C      PRODUCE A SINGLE LOAD VECTOR, FF, FROM ALL    01250
C      THE STIFFNESSES-LOAD VECTOR PRODUCTS IN THE HOLZER METHOD 01251
C                                     01252
ISN 0003      IMPLICIT REAL*8(A-H,O-Z)    01253
ISN 0004      COMMON /STATC/ SH(12,12,20),F(12,21) 01254
ISN 0005      DIMENSION S1(12,12)        01255
ISN 0006      DIMENSION FF(12,1),F1(12,1),F2(12,1) 01256
C                                     01257
C      INITIALIZE FF, AND INSERT THE LOAD VECTOR AT NODE NR    01258
C                                     01259
C      WRITE(6,10)                                           01260
ISN 0007      DO 50 I = 1,12                               01261
ISN 0008      50 FF(I,1) = F(I,NR)                         01262
C      CALL MATPRN(FF,12,1,' FF ')                        01263
C                                                         01264
ISN 0009      NRM1 = NR - 1                                01265
C                                                         01266
ISN 0010      IF(NRM1 .EQ. 1) RETURN                       01267
C                                                         01268
ISN 0012      DO 100 I = 2,NRM1                            01269
C                                                         01270
C      FILL IN DUMMY ARRAYS                                01271
C                                                         01272
ISN 0013      DO 110 J = 1,12                               01273
ISN 0014      F1(J,1) = F(J,I)                            01274
ISN 0015      DO 110 K = 1,12                               01275
ISN 0016      110 S1(J,K) = SH(J,K,NR-I)                  01276
C      WRITE(6,10)                                         01277
C10  FORMAT(/,5X,'*** IN LOAD2 ***')                      01278
C      CALL MATPRN(S1,12,12,' S1 ')                       01279
C      CALL MATPRN(F1,12,1,' F1 ')                        01280
C                                                         01281
C      PERFORM MATRIX MULTIPLICATION                      01282
C                                                         01283
ISN 0017      CALL MATMPY(S1,F1,F2,12,12,1)              01284
C      CALL MATPRN(F2,12,1,' F2 ')                        01285
C                                                         01286
C      SUM INTO THE FF ARRAY                              01287
C                                                         01288
ISN 0018      DO 120 J = 1,12                               01289
ISN 0019      120 FF(J,1) = FF(J,1) + F2(J,1)            01290
C                                                         01291
ISN 0020      100 CONTINUE                                 01292
C      CALL MATPRN(FF,12,1,' FF ')                        01293
C                                                         01294
ISN 0021      RETURN                                       01295
ISN 0022      END                                          01296
    
```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
 *OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
 STATISTICS SOURCE STATEMENTS = 21, PROGRAM SIZE = 1918, SUBPROGRAM NAME = LOAD2
 STATISTICS NO DIAGNOSTICS GENERATED

119

+VERSION 1.3.0 (01 MAY 80)
***** END OF COMPILATION *****

SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 82.141/10.49.04
3016K BYTES OF CORE NOT USED

PAGE 2

120

REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

ISN 0002	C		01297
		SUBROUTINE MATADD(A,B,C,I,J,S)	01298
	C		01299
	C	MATRIX ADDITION A(I,J) + S*B(I,J) = C(I,J)	01300
	C	WITH, S = SIGN OF B (ADDITION OR SUBTRACTION)	01301
	C		01302
ISN 0003		DOUBLE PRECISION A(I,J),B(I,J),C(I,J),S	01303
	C		01304
ISN 0004		DO 100 K = 1,I	01305
ISN 0005		DO 100 L = 1,J	01306
ISN 0006	100	C(K,L) = A(K,L) + S * B(K,L)	01307
	C		01308
ISN 0007		RETURN	01309
ISN 0008		END	01310

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

STATISTICS SOURCE STATEMENTS = 7, PROGRAM SIZE = 436, SUBPROGRAM NAME =MATADD

STATISTICS NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

3016K BYTES OF CORE NOT USED

REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

122

ISN 0002	C	SUBROUTINE MATMPY(A,B,C,I,J,K)	01311
	C		01312
	C	THIS ROUTINE MULTIPLIES MATRICES A * B = C	01313
	C	A(I,J) , B(J,K) AND C(I,K)	01314
	C		01315
ISN 0003	C	DOUBLE PRECISION A(I,J),B(J,K),C(I,K)	01316
	C		01317
ISN 0004	C	DO 100 L = 1,I	01318
ISN 0005		DO 100 M = 1,K	01319
ISN 0006		C(L,M) = 0.	01320
ISN 0007		DO 100 N = 1,J	01321
ISN 0008	100	C(L,M) = C(L,M) + A(L,N) * B(N,M)	01322
	C		01323
ISN 0009		RETURN	01324
ISN 0010		END	01325
			01326

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

STATISTICS SOURCE STATEMENTS = 9, PROGRAM SIZE = 562, SUBPROGRAM NAME =MATMPY

STATISTICS NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

3016K BYTES OF CORE NOT USED

REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

ISN 0002	C	SUBROUTINE MATPRN(A,I,J,TITLE)	01327
ISN 0003		DOUBLE PRECISION A(I,J)	01328
	C		01329
	C	MATRIX OUTPUT	01330
	C		01331
ISN 0004		WRITE(6,1) TITLE	01332
ISN 0005	1	FORMAT(/,5X,'MATRIX OUTPUT FOR ',A4,/,)	01333
ISN 0006		WRITE(6,2)	01334
ISN 0007	2	FORMAT(5X,'ROW',5X,'COL',5X,'VALUE',/,)	01335
ISN 0008		DO 100 K = 1,I	01336
ISN 0009		DO 100 L = 1,J	01337
ISN 0010		IF(A(K,L) .EQ. 0.) GO TO 100	01338
ISN 0012		WRITE(6,3) K,L,A(K,L)	01339
ISN 0013	100	CONTINUE	01340
ISN 0014	3	FORMAT(5X,I5,3X,I5,3X,E12.5)	01341
	C		01342
ISN 0015		RETURN	01343
ISN 0016		END	01344
			01345

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

STATISTICS SOURCE STATEMENTS = 15, PROGRAM SIZE = 542, SUBPROGRAM NAME =MATPRN

STATISTICS NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

3016K BYTES OF CORE NOT USED

REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTNT NOXREF NOALC NOANSF TERM IBM FLAG(I)

124

	C		01346
	C	DATA SET T972MINV AT LEVEL 001 AS OF 10/14/77	01347
ISN 0002		SUBROUTINE MINV(A,N,D,L,M)	01348
	C		01349
	C	01350
	C		01351
	C	SUBROUTINE MINV	01352
	C		01353
	C	PURPOSE	01354
	C	INVERT A MATRIX	01355
	C		01356
	C	USAGE	01357
	C	CALL MINV(A,N,D,L,M)	01358
	C		01359
	C	DESCRIPTION OF PARAMETERS	01360
	C	A - INPUT MATRIX, DESTROYED IN COMPUTATION AND REPLACED BY	01361
	C	RESULTANT INVERSE.	01362
	C	N - ORDER OF MATRIX A	01363
	C	D - RESULTANT DETERMINANT	01364
	C	L - WORK VECTOR OF LENGTH N	01365
	C	M - WORK VECTOR OF LENGTH N	01366
	C		01367
	C	REMARKS	01368
	C	MATRIX A MUST BE A GENERAL MATRIX	01369
	C		01370
	C	SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED	01371
	C	NONE	01372
	C		01373
	C	METHOD	01374
	C	THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT	01375
	C	IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT	01376
	C	THE MATRIX IS SINGULAR.	01377
	C		01378
	C	01379
	C		01380
ISN 0003		DIMENSION A(1),L(1),M(1)	01381
	C		01382
	C	01383
	C		01384
	C	IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE	01385
	C	C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION	01386
	C	STATEMENT WHICH FOLLOWS.	01387
	C		01388
ISN 0004		DOUBLE PRECISION A,D,BIGA,HOLD	01389
	C		01390
	C	THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS	01391
	C	APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS	01392
	C	ROUTINE.	01393
	C		01394
	C	THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO	01395
	C	CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS. DABS IN STATEMENT	01396
	C	10 MUST BE CHANGED TO DDABS.	01397
	C		01398
	C	01399
	C		01400

```

C          SEARCH FOR LARGEST ELEMENT
C
ISN 0005      D=1.0
ISN 0006      NK=-N
ISN 0007      DO 80 K=1,N
ISN 0008      NK=NK+N
ISN 0009      L(K)=K
ISN 0010      M(K)=K
ISN 0011      KK=NK+K
ISN 0012      BIGA=A(KK)
ISN 0013      DO 20 J=K,N
ISN 0014      IZ=N*(J-1)
ISN 0015      DO 20 I=K,N
ISN 0016      IJ=IZ+I
ISN 0017      10 IF( DABS(BIGA)- DABS(A(IJ))) 15,20,20
ISN 0018      15 BIGA=A(IJ)
ISN 0019      L(K)=I
ISN 0020      M(K)=J
ISN 0021      20 CONTINUE

C
C          INTERCHANGE ROWS
C
ISN 0022      J=L(K)
ISN 0023      IF(J-K) 35,35,25
ISN 0024      25 KI=K-N
ISN 0025      DO 30 I=1,N
ISN 0026      KI=KI+N
ISN 0027      HOLD=-A(KI)
ISN 0028      JI=KI-K+J
ISN 0029      A(KI)=A(JI)
ISN 0030      30 A(JI) =HOLD

C
C          INTERCHANGE COLUMNS
C
ISN 0031      35 I=M(K)
ISN 0032      IF(I-K) 45,45,38
ISN 0033      38 JP=N*(I-1)
ISN 0034      DO 40 J=1,N
ISN 0035      JK=NK+J
ISN 0036      JI=JP+J
ISN 0037      HOLD=-A(JK)
ISN 0038      A(JK)=A(JI)
ISN 0039      40 A(JI) =HOLD

C
C          DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS
C          CONTAINED IN BIGA)
C
ISN 0040      45 IF(BIGA) 48,46,48
ISN 0041      46 D=0.0
ISN 0042      RETURN
ISN 0043      48 DO 55 I=1,N
ISN 0044      IF(I-K) 50,55,50
ISN 0045      50 IK=NK+I
ISN 0046      A(IK)=A(IK)/(-BIGA)
ISN 0047      55 CONTINUE

C
C          REDUCE MATRIX
C
ISN 0048      DO 65 I=1,N

```

```

01401
01402
01403
01404
01405
01406
01407
01408
01409
01410
01411
01412
01413
01414
01415
01416
01417
01418
01419
01420
01421
01422
01423
01424
01425
01426
01427
01428
01429
01430
01431
01432
01433
01434
01435
01436
01437
01438
01439
01440
01441
01442
01443
01444
01445
01446
01447
01448
01449
01450
01451
01452
01453
01454
01455
01456
01457
01458
01459

```

126

```

ISN 0049      IK=IK+I                01460
ISN 0050      HOLD=A(IK)             01461
ISN 0051      IJ=I-N                 01462
ISN 0052      DO 65 J=1,N            01463
ISN 0053      IJ=IJ+N                01464
ISN 0054      IF(I-K) 60,65,60      01465
ISN 0055      60 IF(J-K) 62,65,62    01466
ISN 0056      62 KJ=IJ-I+K           01467
ISN 0057      A(IJ)=HOLD*A(KJ)+A(IJ) 01468
ISN 0058      65 CONTINUE            01469
C                                                    01470
C      DIVIDE ROW BY PIVOT            01471
C                                                    01472
ISN 0059      KJ=K-N                 01473
ISN 0060      DO 75 J=1,N            01474
ISN 0061      KJ=KJ+N                01475
ISN 0062      IF(J-K) 70,75,70      01476
ISN 0063      70 A(KJ)=A(KJ)/BIGA    01477
ISN 0064      75 CONTINUE            01478
C                                                    01479
C      PRODUCT OF PIVOTS              01480
C                                                    01481
ISN 0065      D=D*BIGA                01482
C                                                    01483
C      REPLACE PIVOT BY RECIPROCAL    01484
C                                                    01485
ISN 0066      A(KK)=1.0/BIGA          01486
ISN 0067      80 CONTINUE            01487
C                                                    01488
C      FINAL ROW AND COLUMN INTERCHANGE 01489
C                                                    01490
ISN 0068      K=N                     01491
ISN 0069      100 K=(K-1)             01492
ISN 0070      IF(K) 150,150,105      01493
ISN 0071      105 I=L(K)              01494
ISN 0072      IF(I-K) 120,120,108    01495
ISN 0073      108 JQ=N*(K-1)         01496
ISN 0074      JR=N*(I-1)             01497
ISN 0075      DO 110 J=1,N           01498
ISN 0076      JK=JQ+J                01499
ISN 0077      HOLD=A(JK)             01500
ISN 0078      JI=JR+J                01501
ISN 0079      A(JK)=-A(JI)           01502
ISN 0080      110 A(JI) =HOLD         01503
ISN 0081      120 J=M(K)              01504
ISN 0082      IF(J-K) 100,100,125    01505
ISN 0083      125 KI=K-N              01506
ISN 0084      DO 130 I=1,N           01507
ISN 0085      KI=KI+N                01508
ISN 0086      HOLD=A(KI)             01509
ISN 0087      JI=KI-K+J              01510
ISN 0088      A(KI)=-A(JI)           01511
ISN 0089      130 A(JI) =HOLD         01512
ISN 0090      GO TO 100               01513
ISN 0091      150 RETURN              01514
ISN 0092      END                     01515

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
STATISTICS SOURCE STATEMENTS = 91, PROGRAM SIZE = 1502, SUBPROGRAM NAME = MINV

STATISTICS NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

3004K BYTES OF CORE NOT USED

REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE
 OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
 SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

128

	C		01516*35
ISN 0002		SUBROUTINE MODMAS(I,EX,NR,RMASS)	01517*35
ISN 0003		IMPLICIT REAL*8(A-H,O-Z)	01518*35
	C		01519*35
	C	MODAL MASS FOR THE I MODE	01520*35
	C		01521*35
ISN 0004		COMMON /STIFF/ SK(12,12),SKK(12,12,21),SMM(6,6,21)	01522*35
ISN 0005		DIMENSION EX(12,21,2),RMASS(3),DISP(6,1),SMMM(6,6),DISPT(1,6)	01523*35
ISN 0006		DIMENSION D1(6,1)	01524*35
	C		01525*35
ISN 0007		RMASS(I) = 0.	01526*35
ISN 0008		DO 100 J = 1,NR	01527*35
	C		01528*35
	C	CONVERT THE W137 EIGENVECTOR, EX, TO STABEL FORMAT	01529*35
	C		01530*35
ISN 0009		CALL SHAPE(J,EX,DISP)	01531*35
	C		01532*35
	C	NOW MULTIPLY - DISP * SMMM * DISP	01533*35
	C		01534*35
ISN 0010		DO 110 K = 1,6	01535*35
ISN 0011		DO 110 L = 1,6	01536*35
ISN 0012	110	SMMM(K,L) = SMM(K,L,J)	01537*35
	C		01538*35
ISN 0013		DO 120 K = 1,6	01539*35
ISN 0014	120	DISPT(1,K) = DISP(K,1)	01540*35
	C		01541*35
ISN 0015		CALL MATMPY(SMMM,DISP,D1,6,6,1)	01542*35
	C		01543*35
ISN 0016		CALL MATMPY(DISP,D1,D2,1,6,1)	01544*35
	C		01545*35
ISN 0017	100	RMASS(I) = RMASS(I) + D2	01546*35
	C		01547*35
ISN 0018		RETURN	01548*35
ISN 0019		END	01549*35

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
 *OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
 STATISTICS SOURCE STATEMENTS = 18, PROGRAM SIZE = 1034, SUBPROGRAM NAME =MODMAS
 STATISTICS NO DIAGNOSTICS GENERATED
 ***** END OF COMPILATION ***** 3016K BYTES OF CORE NOT USED

REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTNT NOXREF NOALC NOANSF TERM IBM FLAG(I)

	C		01550*35
ISN 0002		SUBROUTINE MODSTR (I,NC,NR)	01551*35
ISN 0003		IMPLICIT REAL*8(A-H,O-Z)	01552*35
	C		01553*35
	C	THIS ROUTINE IS THE DRIVER FOR THE MODAL STRESS CALCULATION	01554*35
	C		01555*35
ISN 0004		COMMON /CALCA/ FILL(1468),MS(2),FILL2(588),EX(12,21,2),SS(12,2)	01556*35
ISN 0005		COMMON /STRS/ RBE(12,12,25,21),FOUT(12,21),SL(25,21),SH(25,21)	01557*35
ISN 0006		COMMON /MODAL/ SIGMA(3,8,25)	01558*35
ISN 0007		COMMON /MODAL1/ STRESS(3,3,8,25),FACTOR(3),RMASS(3)	01559*35
ISN 0008		COMMON /BLK15/ DIAHOD(100),FNI(100),FILL1(300)	01560*35
ISN 0009		DIMENSION DISP(6,1),BFORCE(6,1),DISPT(1,6),PLOAD(1,1)	01561*35
ISN 0010		DIMENSION SMAX(8)	01562*36
	C		01563*35
	C	CONVERT W137 EIGENVECTOR TO STABEL FORMAT FOR STATIONS 1 AND 2	01564*35
	C		01565*35
ISN 0011		CALL SHAPE(2,EX,DISP)	01566*39
	C	CALL MATPRN(DISP,6,1,'DISP')	01567*35
ISN 0012		DO 100 J = 1,6	01568*35
ISN 0013	100	FOUT(J,2) = DISP(J,1)	01569*39
	C		01570*35
ISN 0014		CALL SHAPE(3,EX,DISP)	01571*39
	C	CALL MATPRN(DISP,6,1,'DISP')	01572*35
ISN 0015		DO 110 J = 1,6	01573*35
ISN 0016	110	FOUT(J,3) = DISP(J,1)	01574*39
	C		01575*35
	C	CALCULATE ROOT STRESS FOR MODE I	01576*35
	C		01577*35
ISN 0017		CALL ZSTRES(2,NC,SMAX)	01578*39
	C		01579*35
	C	STORE THE STRESS VALUES FOR MODE I IN STRESS(3,3,8,25)	01580*35
	C		01581*35
ISN 0018		DO 120 JJ = 1,NC	01582*35
ISN 0019		DO 120 J = 1,8	01583*35
ISN 0020		DO 120 K = 1,3	01584*35
ISN 0021	120	STRESS(I,K,J,JJ) = SIGMA(K,J,JJ)	01585*35
	C		01586*35
	C	CALCULATE THE MODAL MASS FOR MODE I	01587*35
	C		01588*35
ISN 0022		CALL MODMAS (I,EX,NR,RMASS)	01589*35
	C		01590*35
	C	CALCULATE THE BIRD FORCE AND MOMENT	01591*35
	C		01592*35
ISN 0023		CALL BIRDF(NR,NC,ISTA,BFORCE,SQUASH)	01593*35
	C		01594*35
	C	FIND THE STABEL MODE SHAPE AT STATION ISTA	01595*35
	C		01596*35
ISN 0024		CALL SHAPE (ISTA,EX,DISP)	01597*35
	C	CALL MATPRN(DISP,6,1,'DISP')	01598*43
	C		01599*35
ISN 0025		DO 130 J = 1,6	01600*35
ISN 0026	130	DISPT(1,J) = DISP(J,1)	01601*35
	C		01602*35
	C	MODAL LOAD,PLOAD	01603*35
	C		01604*35

130

```

+VERSION 1.3.0 (01 MAY 80)  MODSTR  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 82.141/10.49.33  PAGE  2
  ISN 0027      CALL MATMPY ( DISPT,BFORCE,PLOAD,1,6,1)  01605*35
                CALL MATPRN ( PLOAD,1,1,'PLOAD')  01606*35
  C
  C
  C      PARTICIPATION FACTOR , FACTOR  01607*35
  C
  ISN 0028      PI = PLOAD(1,1)  01608*35
  ISN 0029      TO = SQUASH  01609*35
  ISN 0030      FREQ = FN1(I) * 2. * 3.141593  01610*35
  C
  ISN 0031      FACTOR(I) = PI * TO / RMASS(I) / FREQ  01611*35
  C      CALL MATPRN(FN1,100,1,'FN1 ')  01612*35
  C      CALL MATPRN(RMASS,3,1,'RMASS')  01613*35
  C      CALL MATPRN(FACTOR,3,1,'FACT')  01614*35
  C
  ISN 0032      RETURN  01615*36
  ISN 0033      END  01616*36
                01617*36
                01618*35
                01619*35
                01620*35
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 32, PROGRAM SIZE = 1218, SUBPROGRAM NAME =MODSTR
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

3012K BYTES OF CORE NOT USED

	C		01621
ISN 0002		SUBROUTINE PARTN(SK,II,JJ,SK11,SK12,SK21,SK22,KK,LL)	01622
	C		01623
	C	THIS ROUTINE PARTITIONS THE SK MATRIX INTO SK11,SK12,	01624
	C	SK21 AND SK22	01625
	C		01626
ISN 0003		IMPLICIT REAL*8(A-H,O-Z)	01627
ISN 0004		DIMENSION SK(II,JJ),SK11(KK,LL),SK12(KK,LL),SK21(KK,LL)	01628
ISN 0005		DIMENSION SK22(KK,LL)	01629
	C		01630
	C	SK11	01631
	C		01632
ISN 0006		DO 100 I = 1,6	01633
ISN 0007		DO 100 J = 1,6	01634
ISN 0008	100	SK11(I,J) = SK(I,J)	01635
	C		01636
	C	SK12	01637
	C		01638
ISN 0009		DO 101 I = 1,6	01639
ISN 0010		L = 1	01640
ISN 0011		DO 101 J = 7,12	01641
ISN 0012		SK12(I,L) = SK(I,J)	01642
ISN 0013	101	L = L + 1	01643
	C		01644
	C	SK21	01645
	C		01646
ISN 0014		DO 102 I = 7,12	01647
ISN 0015		K = I - 6	01648
ISN 0016		DO 102 J = 1,6	01649
ISN 0017		SK21(K,J) = SK(I,J)	01650
ISN 0018	102	CONTINUE	01651
	C		01652
	C	SK22	01653
	C		01654
ISN 0019		DO 103 I = 7,12	01655
ISN 0020		K = I - 6	01656
ISN 0021		L = 1	01657
ISN 0022		DO 104 J = 7,12	01658
ISN 0023		SK22(K,L) = SK(I,J)	01659
ISN 0024	104	L = L + 1	01660
ISN 0025	103	CONTINUE	01661
	C		01662
	C	CALL MATPRN(SK11,6,6,'SK11')	01663
	C	CALL MATPRN(SK12,6,6,'SK12')	01664
	C	CALL MATPRN(SK21,6,6,'SK21')	01665
	C	CALL MATPRN(SK22,6,6,'SK22')	01666
	C		01667
ISN 0026		RETURN	01668
ISN 0027		END	01669

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
 *OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
 STATISTICS SOURCE STATEMENTS = 26, PROGRAM SIZE = 890, SUBPROGRAM NAME = PARTN
 STATISTICS NO DIAGNOSTICS GENERATED
 ***** END OF COMPILATION *****

3016K BYTES OF CORE NOT USED

REQUESTED OPTIONS: AUTO DBL(4),SOURCE

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTO DBL(4)

SCURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

132

ISN 0002	C	SUBROUTINE PPI(II)	01670
			01671
	C		01672
	C	THIS ROUTINE SETS UP THE REORDER MATRICES	01673
	C	P AND PI ARE CONSTANT AND DO NOT CHANGE	01674
	C		01675
ISN 0003		IMPLICIT REAL*8(A-H,O-Z)	01676
ISN 0004		COMMON /REOROR/ P(12,12),PT(12,12)	01677
	C		01678
	C	INITIALIZE	01679
	C		01680
ISN 0005		DO 100 I = 1,12	01681
ISN 0006		DO 100 J = 1,12	01682
ISN 0007	100	P(I,J) = 0.	01683
	C		01684
ISN 0008		IF(II .GT. 1) GO TO 50	01685
	C		01686
	C	SET NON-ZERO VALUES OF P	01687
	C		01688
ISN 0010		P(1,1) = 1.	01689
ISN 0011		P(2,5) = 1.	01690
ISN 0012		P(3,9) = 1.	01691
ISN 0013		P(4,2) = 1.	01692
ISN 0014		P(5,10) = 1.	01693
ISN 0015		P(6,6) = 1.	01694
ISN 0016		P(7,3) = 1.	01695
ISN 0017		P(8,7) = 1.	01696
ISN 0018		P(9,11) = 1.	01697
ISN 0019		P(10,4) = 1.	01698
ISN 0020		P(11,12) = 1.	01699
ISN 0021		P(12,8) = 1.	01700
ISN 0022		GO TO 51	01701
ISN 0023	50	P(1,7) = 1.	01702
ISN 0024		P(2,8) = 1.	01703
ISN 0025		P(3,12) = -1.	01704
ISN 0026		P(4,9) = -1.	01705
ISN 0027		P(5,11) = 1.	01706
ISN 0028		P(6,10) = 1.	01707
ISN 0029		P(7,1) = 1.	01708
ISN 0030		P(8,6) = -1.	01709
ISN 0031		P(9,2) = 1.	01710
ISN 0032		P(10,5) = 1.	01711
ISN 0033		P(11,3) = -1.	01712
ISN 0034		P(12,4) = 1.	01713
ISN 0035	51	CONTINUE	01714
	C		01715
	C	NOW DETERMINE THE TRANSPOSE MATRIX,PT	01716
	C		01717
ISN 0036		DO 200 I = 1,12	01718
ISN 0037		DO 200 J = 1,12	01719
ISN 0038		PT(I,J) = P(J,I)	01720
ISN 0039	200	CONTINUE	01721
	C	CALL MATPRN(P,12,12,' P')	01722
	C	CALL MATPRN(PT,12,12,'PT')	01723
	C		01724

1 1

+VERSION 1.3.0 (01 MAY 80) PP1 SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 82.141/10.49.51 PAGE 2
ISN 0040 RETURN 01725
ISN 0041 END 01726
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOOBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NCMAP NOFORMAT GOSTHT NOXREF NOALC NOANSF TERM IBM FLAG(I)
STATISTICS SOURCE STATEMENTS = 40, PROGRAM SIZE = 534, SUBPROGRAM NAME = PP1
STATISTICS NO DIAGNOSTICS GENERATED
***** END OF COMPILATION ***** 3016K BYTES OF CORE NOT USED

134

	C		01727
ISH 0002		SUBROUTINE RBE2(X,Y,Z,ISEC,IBEAM,NC,BLOCAL)	01728
	C		01729
	C	THIS ROUTINE CALCULATES THE RIGID BODY TRANSFORMATION	01730
	C	MATRICES RBL12 AND RBL1	01731
	C		01732
	C	RBL12 IS WRITTEN IN THE LOCAL BEAM COORD. SYSTEM	01733
	C		01734
ISH 0003		IMPLICIT REAL*8(A-H,O-Z)	01735
ISH 0004		COMMON /RBE/ RBL12(12,12),RBL1(12,12)	01736
ISH 0005		COMMON /WARPP/ HS(26,21),RH(25,21),RT(25,21)	01737
ISH 0006		COMMON /SC/ XSC(21),YSC(21),ZSC(21),ALPHA(21),SECIP(21),SECA(21)	01738
ISH 0007		DIMENSION RBL(6,6),RL(6),RH(6)	01739
ISH 0008		DIMENSION X(1),Y(1),Z(1),XG(3),XGB(3),XA1(3)	01740
ISH 0009		DIMENSION XA2(3),XB1(3),XB2(3),XGB(3),BLOCAL(3,3)	01741
ISH 0010		DIMENSION V1(3),V2(3),VL1(3),VL2(3)	01742*34
	C		01743
	C	FORMING RBL12	01744
	C	TRANSFORM GLOBAL COORD. POINTS TO SECTION SYSTEM	01745
	C		01746
	C	LAYER 1-BEAM COORD IS XA1,SECTION COORD IS XB1	01747
	C		01748
ISH 0011		IP = (ISEC-1) * NC + IBEAM	01749
	C		01750*34
	C	POSITION VECTORS FOR LAYERS 1 AND 2 , V1 AND V2	01751*34
	C		01752*34
ISH 0012		V1(1) = X(IP) - XSC(ISEC)	01753*34
ISH 0013		V1(2) = Y(IP) - YSC(ISEC)	01754*34
ISH 0014		V1(3) = Z(IP) - ZSC(ISEC)	01755*34
	C		01756*34
ISH 0015		V2(1) = X(IP+NC) - XSC(ISEC+1)	01757*34
ISH 0016		V2(2) = Y(IP+NC) - YSC(ISEC+1)	01758*34
ISH 0017		V2(3) = Z(IP+NC) - ZSC(ISEC+1)	01759*34
	C		01760*34
	C	ROTATE FROM GLOBAL SYSTEM TO BEAM LOCAL	01761*34
	C		01762*34
ISH 0018		CALL MATMPY(BLOCAL,V1,VL1,3,3,1)	01763*34
ISH 0019		CALL MATMPY(BLOCAL,V2,VL2,3,3,1)	01764*34
	C		01765*34
ISH 0020		X1 = VL1(1)	01766*34
ISH 0021		Y1 = VL1(2)	01767*34
ISH 0022		Z1 = VL1(3)	01768*34
ISH 0023		X2 = VL2(1)	01769*34
ISH 0024		Y2 = VL2(2)	01770*34
ISH 0025		Z2 = VL2(3)	01771*34
ISH 0026		XL = XSC(ISEC+1) - XSC(ISEC)	01772*34
	C		01773*34
	C		01774
ISH 0027		DO 102 I = 1,12	01775
ISH 0028		DO 102 J = 1,12	01776
ISH 0029		RBL1(I,J) = 0.	01777
ISH 0030	102	RBL12(I,J) = 0.	01778
	C		01779
ISH 0031		DO 103 I = 1,12	01780
ISH 0032		RBL1(I,I) = 1.	01781

```

ISN 0033      103 RBL12(I,I) = 1.                                01782
C                                                     01783
ISN 0034      RBL12(1,4) = -WS(IBEAM,ISEC) / XL              01784
ISN 0035      RBL12(1,5) = Z1                                01785
ISN 0036      RBL12(1,6) = -Y1                               01786
ISN 0037      RBL12(1,10) = -RBL12(1,4)                     01787
ISN 0038      RBL12(2,4) = -Z1                               01788
ISN 0039      RBL12(2,6) = X1                                01789
ISN 0040      RBL12(3,4) = Y1                                01790
ISN 0041      RBL12(3,5) = -X1                               01791
ISN 0042      RBL12(5,4) = RT(IBEAM,ISEC) / XL              01792
ISN 0043      RBL12(5,10) = -RBL12(5,4)                     01793
ISN 0044      RBL12(6,4) = RN(IBEAM,ISEC) / XL              01794
ISN 0045      RBL12(6,10) = -RBL12(6,4)                     01795
ISN 0046      RBL12(7,4) = -WS(IBEAM,ISEC+1) / XL           01796
ISN 0047      RBL12(7,10) = -RBL12(7,4)                     01797
ISN 0048      RBL12(7,11) = Z2                               01798
ISN 0049      RBL12(7,12) = -Y2                             01799
ISN 0050      RBL12(8,10) = -Z2                              01800
ISN 0051      RBL12(8,12) = X2                               01801
ISN 0052      RBL12(9,10) = Y2                               01802
ISN 0053      RBL12(9,11) = -X2                              01803
ISN 0054      RBL12(11,4) = RT(IBEAM,ISEC+1) / XL           01804
ISN 0055      RBL12(11,10) = -RBL12(11,4)                   01805
ISN 0056      RBL12(12,4) = RN(IBEAM,ISEC+1) / XL           01806
ISN 0057      RBL12(12,10) = -RBL12(12,4)                   01807
C CALL MATPRN(RBL12,12,12,'RBL12')                          01808
ISN 0058      RBL1(1,5) = RBL12(1,5)                         01809
ISN 0059      RBL1(1,6) = RBL12(1,6)                         01810
ISN 0060      RBL1(2,4) = RBL12(2,4)                         01811
ISN 0061      RBL1(2,6) = RBL12(2,6)                         01812
ISN 0062      RBL1(3,4) = RBL12(3,4)                         01813
ISN 0063      RBL1(3,5) = RBL12(3,5)                         01814
ISN 0064      RBL1(7,11) = RBL12(7,11)                       01815
ISN 0065      RBL1(7,12) = RBL12(7,12)                       01816
ISN 0066      RBL1(8,10) = RBL12(8,10)                       01817
ISN 0067      RBL1(8,12) = RBL12(8,12)                       01818
ISN 0068      RBL1(9,10) = RBL12(9,10)                       01819
ISN 0069      RBL1(9,11) = RBL12(9,11)                       01820
C CALL MATPRN(RBL1,12,12,'RBL1')                             01821
C                                                         01822
C                                                         01823*39
ISN 0070      IF( ISEC .EQ. 2 ) GO TO 300                     01824*41
ISN 0072      IF( ISEC .GT. 1 ) RETURN                         01825*39
C                                                         01826*39
C RESET RBL12 FOR EXTENED NECK REGION - NO WARPING          01827*39
C                                                         01828*39
ISN 0074      DO 200 I = 1,12                                 01829*39
ISN 0075      DO 200 J = 1,12                                 01830*39
ISN 0076      200 RBL12(I,J) = RBL1(I,J)                     01831*39
ISN 0077      300 CONTINUE                                    01832*41
ISN 0078      DO 305 I = 1,6                                  01833*41
ISN 0079      DO 305 J = 1,12                                 01834*41
ISN 0080      305 RBL12(I,J) = RBL1(I,J)                     01835*41
C                                                         01836*39
ISN 0081      RETURN                                          01837
ISN 0082      END                                             01838

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

+VERSION 1.3.0 (01 MAY 80) RBE2 SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 82.141/10.49.55
STATISTICS SOURCE STATEMENTS = 81, PROGRAM SIZE = 1496, SUBPROGRAM NAME = RBE2
STATISTICS NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

PAGE 3

3004K BYTES OF CORE NOT USED

136

REQUESTED OPTIONS: AUTOABL(DBL4),SOURCE

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOABL(DBL4)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP HOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```

C
ISN 0002      C      SUBROUTINE SECTN(I,NR,NC,RPM)                                01839
C                                                     01840
C                                                     01841
C THIS ROUTINE CALCULATES THE SECTION STIFFNESS MATRIX,SKK AND SK 01842
C THE SECTION STIFFNESSES ARE SUMMED FROM THE LOCAL BEAMS        01843
C THE SECTION MASS IS ALSO DETERMINED FROM THE BEAM MASSES      01844
C ALSO THE SECTION CENTRIFUGAL RESTORING MATRIC ,CMGG, IS FOUND. 01845
C                                                     01846
ISN 0003      C      IMPLICIT REAL*8(A-H,O-Z)                                01847
ISN 0004      COMMON /SC/XSC(21),YSC(21),ZSC(21),ALPHA(21),SECIP(21),SECA(21) 01848
ISN 0005      COMMON /RBE/ RBL12(12,12),RBL1(12,12)                01849
ISN 0006      COMMON /STRS/ RBE(12,12,25,21),FOUT(12,21),SL(25,21),SH(25,21) 01850*34
ISN 0007      COMMON /COORD1/ CLOCAL(3,3,1000)                    01851
ISN 0008      COMMON /BSTIFF/ BK(12,12),EM(12,12)                 01852
ISN 0009      COMMON /STATC/ SH(12,12,20),F(12,21)              01853
ISN 0010      COMMON /LAYER/ TH(7,25),BMASS(25),BSPAN(25),BWIDTH(25) 01854
ISN 0011      DIMENSION TBSS(12,12),R(12,12),R1(12,12)          01855
ISN 0012      COMMON /STIFF/ SK(12,12),SKK(12,12,21),SMM(6,6,21) 01856
ISN 0013      COMMON /DSTIFF/ CMGG(12,12,20),DSKK(12,12,20)     01857
ISN 0014      COMMON /INPUTT/ X(1000),Y(1000),Z(1000),T(1000)   01858
ISN 0015      COMMON /BLK 8/ SMB(21),ITLLE(18),VARI(235),TMAX(21),HALPHA(21) 01859*34
ISN 0016      COMMON /ANAL33/ DUM6(6),NCD,NCK,DUM4(4)            01860*47
ISN 0017      DIMENSION BLOCAL(3,3),SM1(6,6),SM2(6,6),SM2OLD(6,6) 01861
ISN 0018      DIMENSION SM(12,12),FBEAM1(6,1),FBEAM2(6,1),FOLD(6,1) 01862
ISN 0019      COMMON /TRANS/ TS(3,3,21)                          01863
ISN 0020      DIMENSION D1(12,12),D2(12,12),SKOLD(12,12),TS1(3,3),TS2(3,3) 01864
ISN 0021      DIMENSION FF(12,1),FSEC(12,1),FTOT1(6,1),FTOT2(6,1) 01865
ISN 0022      DIMENSION BCMGG(12,12),CMG(12,12),TBS1(3,3),TBS2(3,3) 01866
ISN 0023      DIMENSION RF(12,12)                                01867
C                                                     01868
C BEGIN THE SECTION STIFFNESS SOLUTION FOR SECTION I            01869
C                                                     01870
C FORM THE SECTION COORDINATE SYSTEM ,TS                        01871
C TS1 FOR LAYER ONE                                             01872
C TS2 FOR LAYER TWO                                             01873
C                                                     01874
ISN 0024      C      IP = (I-1) * NC + 1                                01875
ISN 0025      IP = IP + NC / 2                                    01876
ISN 0026      IPP = IP + NC                                     01877
ISN 0027      NRMI = NR - 1                                      01878
C                                                     01879
C LOCAL X UNIT VECTOR                                           01880
C                                                     01881
ISN 0028      C      II = I + 1                                        01882
ISN 0029      IF(I .EQ. 1) II = 1                                01883
ISN 0031      IF(I .EQ. NRMI) II = NRMI                          01884
ISN 0033      99  XSX = XSC(II+1) - XSC(II)                       01885
ISN 0034      XSY = YSC(II+1) - YSC(II)                         01886
ISN 0035      XSZ = ZSC(II+1) - ZSC(II)                         01887
C      WRITE(6,6) XSC(II),YSC(II),ZSC(II),XSC(II+1),YSC(II+1),ZSC(II+1) 01888
C6  FORMAT(5X,6E12.5)                                           01889
ISN 0036      XSMAG = DSQRT( XSX**2 + XSY**2 + XSZ**2 )          01890
C      WRITE(6,4) XSMAG,I,II                                       01891
C4  FORMAT(5X,'MAGNITUDE',E12.5,2I5)                             01892
ISN 0037      XSX = XSX / XSMAG                                   01893

```

```

+VERSION 1.3.0 (01 MAY 80)      SECTN  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 82.141/10.50.02      PAGE  2
  ISN 0038      XSY = XSY / XSMAG      01894
  ISN 0039      XSZ = XSZ / XSMAG      01895
  C
  C      LOCAL Z UNIT VECTOR (INITIALLY)  01896
  C      01897
  ISN 0040      ZSX = 0.                01898
  ISN 0041      ZSY = DCOS (ALPHA(II))  01899
  ISN 0042      ZSZ = DSIN (ALPHA(II))  01900
  C      01901
  C      LOCAL Y UNIT VECTOR YS = ZS CROSS XS  01902
  C      01903
  C      01904
  ISN 0043      YSX = ZSY * XSZ - ZSZ * XSY  01905
  ISN 0044      YSY = ZSZ * XSX - ZSX * XSZ  01906
  ISN 0045      YSZ = ZSX * XSY - ZSY * XSX  01907
  ISN 0046      YSMAG = DSQRT ( YSX**2 + YSY**2 + YSZ**2 )  01908
  C      WRITE(6,4) YSMAG,I,II          01909
  ISN 0047      YSX = YSX / YSMAG      01910
  ISN 0048      YSY = YSY / YSMAG      01911
  ISN 0049      YSZ = YSZ / YSMAG      01912
  C      01913
  C      REDEFINE THE LOCAL ZS VECTOR , ZS = XS CROSS YS  01914
  C      01915
  C      01916
  ISN 0050      IF( I .EQ. NRM1 .AND. NRM1 .NE. 1 ) II = II + 1
  ISN 0052      98  TS(3,1,II) = XSY * YSZ - XSZ * YSY  01917
  ISN 0053      TS(3,2,II) = XSZ * YSX - XSX * YSZ  01918
  ISN 0054      TS(3,3,II) = XSX * YSY - XSY * YSX  01919
  C      01920
  C      FILL THE TS ARRAY  01921
  C      01922
  ISN 0055      TS(1,1,II) = XSX      01923
  ISN 0056      TS(1,2,II) = XSY      01924
  ISN 0057      TS(1,3,II) = XSZ      01925
  ISN 0058      TS(2,1,II) = YSX      01926
  ISN 0059      TS(2,2,II) = YSY      01927
  ISN 0060      TS(2,3,II) = YSZ      01928
  C      01929
  ISN 0061      DO 300 L = 1,3          01930
  ISN 0062      DO 300 M = 1,3          01931
  ISN 0063      300 IF(DABS(TS(L,M,II))).LE. .001) TS(L,M,II) = 0.  01932
  C      01933
  ISN 0065      IF( I .EQ. 1 ) II = II + 1  01934
  ISN 0067      IF(NRM1 .EQ. 1 .AND. II.EQ. 2) GO TO 98  01935
  ISN 0069      IF(I.EQ.1.AND.II.EQ.2) GO TO 99  01936
  C      01937
  C      WRITE(6,2) I  01938
  C      FORMAT(/,5X/'TS MATRIX FOR SECTION',I5,/)  01939
  C      II = I+1  01940
  C      DO 161 L = I,II  01941
  C      WRITE(6,162) L  01942
  C162  FORMAT(5X,'SECTION NODE',I5)  01943
  C161  WRITE(6,1) ((TS(J,K,L),K=1,3),J=1,3)  01944
  C1    FORMAT(1X,3(E12.5,2X),/)  01945
  C      01946
  C      INITIALIZE THE SK AND SM MATRIX AND LOAD VECTORS, FTOT1,FTOT2  01947
  C      01948
  C      01949
  ISN 0071      DO 100 J = 1,12        01949
  ISN 0072      DO 100 K = 1,12        01950
  ISN 0073      CNG(J,K) = 0.          01951
  ISN 0074      SM(J,K) = 0.           01952

```

138

1

11

+VERSION 1.3.0 (01 MAY 80)		SECTN	SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)	DATE 82.141/10.50.02	PAGE 3
ISN 0075	100	SK(J,K) = 0.		01953	
ISN 0076		DO 180 J = 1,6		01954	
ISN 0077		FTOT1(J,1) = 0.		01955	
ISN 0078	180	FTOT2(J,1) = 0.		01956	
	C			01957	
	C	BEGIN LOOP FOR EACH BEAM		01958	
	C			01959	
ISN 0079		DO 200 J = 1,NC		01960	
	C			01961	
	C	CALCULATE THE BEAM COORDINATE SYSTEM ,BLOCAL(3,3)		01962	
	C			01963	
ISN 0080		IP = (I-1) * NC + J		01964	
ISN 0081		IPP = IP + NC		01965	
	C			01966	
ISN 0082		CALL BCOORD(IP,IPP,BLOCAL)		01967	
	C	CALL MATPRN(BLOCAL,3,3,'BLOC')		01968	
	C			01969	
	C	CALCULATE THE BEAM MATERIAL STIFFNESS MATRIX , E		01970	
	C			01971	
ISN 0083		CALL ESTIFF(I,NC,NR,J)		01972*35	
	C			01973	
	C	DETERMINE THE BEAM STIFFNESS MATRIX , BK		01974	
	C	THE BEAM CENTRIFUGAL RESTORING MATRIX, BCNGG		01975	
	C			01976	
ISN 0084		CALL BEAM(NC,J,BLOCAL,BCNGG,RPN)		01977	
	C			01978	
	C	THE BEAM(J) STIFFNESS MATRIX IS NOW BUILT AND MUST		01979	
	C	BE SUMMED INTO THE SECTION (I) STIFFNESS MATRIX ,SK		01980	
	C			01981	
	C	FORM THE TBS TRANSFORMATION MATRIX		01982	
	C	TBS = BLOCAL * TS(TRANSPOSE)		01983	
	C			01984	
ISN 0085		DO 103 K = 1,3		01985	
ISN 0086		DO 103 L = 1,3		01986	
ISN 0087		TS1(K,L) = TS(L,K,I)		01987	
ISN 0088	103	TS2(K,L) = TS(L,K,I+1)		01988	
ISN 0089		CALL MATMPY(BLOCAL,TS1,TBS1,3,3,3)		01989	
ISN 0090		CALL MATMPY(BLOCAL,TS2,TBS2,3,3,3)		01990	
	C	WRITE(6,3) I		01991	
	C3	FORMAT(/,5X,'TBS1,TBS2 MATRIX FOR SECTION',I5,/)		01992	
	C	WRITE(6,1) ((TBS1(K,L),L=1,3),K=1,3)		01993	
	C	WRITE(6,1) ((TBS2(K,L),L=1,3),K=1,3)		01994	
	C			01995	
	C	THE RIGID BODY TRANSFORMATION MATRIX , RBL12		01996	
	C	ALSO CALCULATE THE RIGID BODY MATRIX FOR STATIC LOADS WHICH		01997	
	C	DOES NOT INCLUDE WARPING- ONLY EQUILIBRIUM , RBL1		01998	
	C			01999	
ISN 0091		CALL RBE2(X,Y,Z,I,J,NC,BLOCAL)		02000	
	C			02001	
	C	RESTRAINED WARPING AT THE ROOT		02002	
	C			02003	
ISN 0092		IF(I .EQ. 1) GO TO 190		02004	
ISN 0094		GO TO 191		02005	
ISN 0095	190	IWARP = 1		02006	
ISN 0096		IF(IWARP .GT. 1) GO TO 191		02007	
ISN 0098		WRITE(6,77)		02008	
ISN 0099	77	FORMAT(/,5X,'*** PLANES REMAIN PLANE ROOT CONDITION ***',/)		02009	
ISN 0100		RBL12(1,4) = 0.		02010	
ISN 0101		RBL12(1,10) = 0.		02011	

	+VERSION 1.3.0 (01 MAY 80)	SECTN	SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)	DATE 82.141/10.50.02	PAGE 4
	ISN 0102		RBL12(5,4) = 0.	02012	
	ISN 0103		RBL12(5,10) = 0.	02013	
	ISN 0104		RBL12(6,4) = 0.	02014	
	ISN 0105		RBL12(6,10) = 0.	02015	
	ISN 0106	191	CONTINUE	02016	
		C		02017	
		C	DEFINE THE TBSS MATRIX	02018	
		C		02019	
	ISN 0107		DO 105 K = 1,12	02020	
	ISN 0108		DO 105 L = 1,12	02021	
	ISN 0109	105	TBSS(K,L) = 0.	02022	
	ISN 0110		IROW = 1	02023	
	ISN 0111		DO 106 K = 1,4	02024	
	ISN 0112		DO 106 L = 1,3	02025	
	ISN 0113		ICOL = (K-1) * 3 + 1	02026	
	ISN 0114		DO 108 M = 1,3	02027	
	ISN 0115		TBSS(IROW,ICOL) = TBS1(L,M)	02028	
	ISN 0116		IF(K .GT. 2) TBSS(IROW,ICOL) = TBS2(L,M)	02029	
	ISN 0118	108	ICOL = ICOL + 1	02030	
	ISN 0119	106	IROW = IROW + 1	02031	
		C	WRITE(6,5) I	02032	
		C5	FORMAT(/,5X,'TBSS MATRIX FOR SECTION',I5)	02033	
		C	CALL MATPRN(TBSS,12,12,'TBSS')	02034	
		C		02035	
		C	CALCULATE THE R MATRIX, R= RBL12 * TBSS	02036	
		C	CALCULATE THE RF MATRIX, RF = RBL1 * TBSS	02037	
		C		02038	
	ISN 0120		CALL MATHPY(RBL12,TBSS,R,12,12,12)	02039	
		C	CALL MATPRN(R,12,12,' R '	02040	
	ISN 0121		CALL MATHPY(RBL1,TBSS,RF,12,12,12)	02041	
		C	CALL MATPRN(RF,12,12,' RF '	02042	
		C		02043	
		C	FILL THE RBE ARRAY FOR LATER USE IN THE STATIC ANALYSIS	02044	
		C	ALSO STORE THE INFO FOR BEAM DIMENSIONS	02045*34	
		C		02046	
	ISN 0122		DO 80 K = 1,12	02047	
	ISN 0123		DO 80 L = 1,12	02048	
	ISN 0124	80	RBE(K,L,J,I) = R(K,L)	02049	
	ISN 0125		SL(J,I) = BSPAN(J)	02050*34	
	ISN 0126		SH(J,I) = BWIDTH(J)	02051*34	
		C		02052	
		C	CALCULATE R1 = TSS * RBL1 * TBB	02053	
		C	IT HAS ALSO BEEN SHOWN THAT R1 = R(TRANPOSE)	02054	
		C	- ALSO TRANPOSE RF AND STORE IN RBL1	02055	
		C		02056	
	ISN 0127		DO 201 K = 1,12	02057	
	ISN 0128		DO 201 L = 1,12	02058	
	ISN 0129		RBL1(K,L) = RF(L,K)	02059	
	ISN 0130	201	R1(K,L) = R(L,K)	02060	
		C	CALL MATPRN(R1,12,12,' R1 '	02061	
		C	CALL MATPRN(RBL1,12,12,'RFT '	02062	
		C		02063	
		C	BEAM(J) CONTRIBUTION TO SECTION (I) STIFFNESS IS:	02064	
		C	SK(K,L) = R1(K,M) * BK(M,N) * R(N,L)	02065	
		C		02066	
	ISN 0131		CALL MATHPY(BK,R,D1,12,12,12)	02067	
	ISN 0132		CALL MATHPY(R1,D1,SKOLD,12,12,12)	02068	
		C		02069	
		C	BEAM(J) CONTRIBUTION TO SECTION(I) MASS , SM	02070	

	+VERSION 1.3.0 (01 MAY 80)	SECTN	SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)	DATE 82.141/10.50.02	PAGE 6
	ISN 0154		A = 0.5 * SMB(2)	02130*47	
	ISN 0155		XDIST = XSC(I) - XSC(2)	02131*39	
	ISN 0156		ARG = - XDIST / A	02132*34	
	ISN 0157		WARP1 = 1. - DEXP(ARG)	02133*34	
	ISN 0158		IF(I .LE. 2) WARP1 = 1.	02134*39	
	ISN 0160		XDIST = XSC(I+1) - XSC(2)	02135*39	
	ISN 0161		ARG = - XDIST / A	02136*34	
	ISN 0162		WARP2 = 1. - DEXP(ARG)	02137*34	
	ISN 0163		IF(I .EQ. 1) WARP2 = 1.	02138*39	
		C		02139*34	
	ISN 0165		SK(4,4) = SK(4,4) / WARP1	02140*34	
	ISN 0166		SK(4,10) = SK(4,10) / WARP1	02141*34	
	ISN 0167		SK(10,4) = SK(10,4) / WARP2	02142*34	
	ISN 0168		SK(10,10) = SK(10,10) / WARP2	02143*34	
		C		02144*39	
		C	REPLACE EXTENDED NECK STIFFNESS WITH RECTANGULAR SECTION NECK	02145*39	
		C		02146*39	
	ISN 0169		IF(I .EQ. 1 .AND. NCD .EQ. 1) CALL EXNECK(SMB(2))	02147*47	
		C		02148	
		C	INSERT SK INTO SKK FOR STORAGE	02149	
		C		02150	
	ISN 0171		DO 110 J = 1,12	02151	
	ISN 0172		DO 110 K = 1,12	02152	
	ISN 0173		CMGG(J,K,I) = CMG(J,K)	02153	
	ISN 0174	110	SKK(J,K,I) = SK(J,K)	02154	
		C		02155	
		C	FILL IN THE MASS MATRIX , SMM, AND THE LOAD VECTOR , F.	02156	
		C		02157	
	ISN 0175		DO 115 J = 1,6	02158	
	ISN 0176		FBEAM1(J,1) = FTOT1(J,1)	02159	
	ISN 0177		FBEAM2(J,1) = FTOT2(J,1)	02160	
	ISN 0178		IROW = J + 6	02161	
	ISN 0179		ICOL = 7	02162	
	ISN 0180		DO 115 K = 1,6	02163	
	ISN 0181		SM1(J,K) = SM(J,K)	02164	
	ISN 0182		SM2(J,K) = SM(IROW,ICOL)	02165	
	ISN 0183	115	ICOL = ICOL + 1	02166	
		C	WRITE(6,30)	02167	
		C30	FORMAT(/,5X,'PUTTING MASS INTO SM')	02168	
		C	CALL MATPRN(SM1,6,6,'SM1')	02169	
		C	CALL MATPRN(SM2,6,6,'SM2')	02170	
		C		02171	
	ISN 0184		NRM1 = NR - 1	02172	
	ISN 0185		IF(I .NE. 1) GO TO 125	02173	
		C		02174	
	ISN 0187		DO 116 J = 1,6	02175	
	ISN 0188		F(J,I) = 0.	02176*38	
	ISN 0189		F(J+6,I) = FBEAM1(J,1)	02177	
	ISN 0190		FOLD(J,1) = FBEAM2(J,1)	02178	
	ISN 0191		DO 116 K = 1,6	02179	
	ISN 0192		SMM(J,K,I) = SM1(J,K)	02180	
	ISN 0193	116	SM2OLD(J,K) = SM2(J,K)	02181	
	ISN 0194		GO TO 126	02182	
	ISN 0195	125	CONTINUE	02183	
	ISN 0196		DO 118 J = 1,6	02184	
	ISN 0197		F(J,I) = 0.	02185*38	
	ISN 0198		F(J+6,I) = FOLD(J,1) + FBEAM1(J,1)	02186	
	ISN 0199		FOLD(J,1) = FBEAM2(J,1)	02187	
	ISN 0200		DO 118 K = 1,6	02188	

ISN 0201		SMM(J,K,I) = SM2OLD(J,K) + SM1(J,K)	02189
ISN 0202	118	SM2OLD(J,K) = SM2(J,K)	02190
ISN 0203	126	IF(I .NE. NRM1) RETURN	02191
ISN 0205		DO 119 J = 1,6	02192
ISN 0206		DO 119 K = 1,6	02193
ISN 0207		F(J,I+1) = 0.	02194*38
ISN 0208		F(J+6,I+1) = FBEAM2(J,1)	02195
ISN 0209	119	SMM(J,K,I+1) = SM2(J,K)	02196
	C		02197
	C	ADD A TIP MOMENT	02198
	C		02199
	C	F(10,I+1) = 1000.	02200
	C		02201
ISN 0210		RETURN	02202
ISN 0211		END	02203

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTONBL(DBL4)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

STATISTICS SOURCE STATEMENTS = 210, PROGRAM SIZE = 17296, SUBPROGRAM NAME = SECTN

STATISTICS NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

2964K BYTES OF CORE NOT USED

REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
 SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

144

	C		02204*35
ISN 0002		SUBROUTINE SHAPE (NODE,EX,DISP)	02205*35
ISN 0003		IMPLICIT REAL*8(A-H,O-Z)	02206*35
	C		02207*35
	C	GIVEN THE W137 EIGENVECTOR CALCULATE THE STABEL SYSTEM EIGEN-	02208*35
	C	VECTOR , FOR A GIVEN NODE	02209*35
	C		02210*35
ISN 0004		COMMON /REORDR/ P(12,12),PT(12,12)	02211*35
ISN 0005		DIMENSION EX(12,21,2),DISP(6,1),W137(6,1),PPT(6,6)	02212*35
	C		02213*35
	C	EXTRACT THE W137 EIGENVECTOR COMPONENT FOR THE NODE	02214*35
	C		02215*35
ISN 0006		DO 100 I = 1,6	02216*35
ISN 0007	100	W137(I,1) = EX(I+6,NODE,1)	02217*35
	C	CALL MATPRN(W137,6,1,'W137')	02218*35
	C		02219*35
	C	TRANSFORMATION ARRAY , PPT	02220*35
	C		02221*35
ISN 0008		CALL PP1(2)	02222*35
	C		02223*35
ISN 0009		DO 110 I = 1,6	02224*35
ISN 0010		DO 110 J = 1,6	02225*35
ISN 0011	110	PPT(I,J) = PT(I,J+6)	02226*35
	C	CALL MATPRN(PPT,6,6,' PPT')	02227*35
	C		02228*35
ISN 0012		CALL MATMPY(PPT,W137,DISP,6,6,1)	02229*35
	C		02230*35
ISN 0013		RETURN	02231*35
ISN 0014		END	02232*35

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

STATISTICS SOURCE STATEMENTS = 13, PROGRAM SIZE = 832, SUBPROGRAM NAME = SHAPE

STATISTICS NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

3016K BYTES OF CORE NOT USED

```

C 02233*35
C 02234
ISN 0002 SUBROUTINE SMULT(NR) 02235
C 02236
C A SLAVE ROUTINE TO PERFORM THE HOLZER STIFFNESS 02237
C MATRIX STRING MULTIPLICATION FOR THE STATIC ANALYSIS 02238
C 02239
ISN 0003 IMPLICIT REAL*8(A-H,O-Z) 02240
ISN 0004 COMMON /STATC/ SH(12,12,20),F(12,21) 02241
ISN 0005 COMMON /HSTIFF/ SKP(12,12),SKPP(12,12,21) 02242
ISN 0006 DIMENSION D1(12,12),D2(12,12),D3(12,12) 02243
C 02244
C INITIALIZE SH MATRIX 02245
C 02246
ISN 0007 NRM1 = NR - 1 02247
ISN 0008 DO 100 I = 1,12 02248
ISN 0009 DO 100 J = 1,12 02249
ISN 0010 100 SH(I,J,1) = SKPP(I,J,NRM1) 02250
C 02251
ISN 0011 IF(NRM1 .EQ. 1) RETURN 02252
C 02253
ISN 0013 DO 200 I = 2,NRM1 02254
ISN 0014 IP = NRM1 - I + 1 02255
ISN 0015 DO 110 J = 1,12 02256
ISN 0016 DO 110 K = 1,12 02257
ISN 0017 D1(J,K) = SH(J,K,I-1) 02258
ISN 0018 110 D2(J,K) = SKPP(J,K,IP) 02259
C 02260
ISN 0019 CALL MATMPY(D1,D2,D3,12,12,12) 02261
C 02262
ISN 0020 DO 120 J = 1,12 02263
ISN 0021 DO 120 K = 1,12 02264
ISN 0022 120 SH(J,K,I) = D3(J,K) 02265
C 02266
C10 WRITE(6,10) 02267
C10 FORMAT(/,5X,'IN SMULT FOR STRING OPERATION') 02268
C CALL MATPRN(D3,12,12,' D3 ') 02269
C 02270
ISN 0023 200 CONTINUE 02270
C 02271
ISN 0024 RETURN 02272
ISN 0025 END 02273

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
 *OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
 STATISTICS SOURCE STATEMENTS = 24, PROGRAM SIZE = 4110, SUBPROGRAM NAME = SMULT
 STATISTICS NO DIAGNOSTICS GENERATED
 ***** END OF COMPILATION *****

3012K BYTES OF CORE NOT USED

REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

146

ISN 0002	C	SUBROUTINE STABEL(NR,NC,RPM1,RPM2,RPM3,FREQ,ILOOP,ISERCH)	02274
			02275*34
	C		02276
	C	THIS PROGRAM IS THE DRIVER FOR A STACKED BEAM ANALYSIS	02277
	C	A STATIC ANALYSIS IS PERFORMED WITH THE HOLZER METHOD	02278
	C	WRITTEN BY NICK MARTIN AND KEN BROWN	02279
	C		02280
ISN 0003		IMPLICIT REAL*8(A-H,O-Z)	02281
ISN 0004		COMMON /INPUT/ X(1000),Y(1000),Z(1000),T(1000)	02282
ISN 0005		COMMON /BC/ C(6,6),RPM	02283
ISN 0006		COMMON /STIFF/ SK(12,12),SKK(12,12,21),SMM(6,6,21)	02284
ISN 0007		COMMON /DSTIFF/ CMGG(12,12,20),DSKK(12,12,20)	02285
ISN 0008		COMMON /SC/ XSC(21),YSC(21),ZSC(21),ALPHA(21),SECIP(21),SECA(21)	02286
ISN 0009		COMMON /HSTIFF/ SKP(12,12),SKPP(12,12,21)	02287
ISN 0010		COMMON /BLKAC/ AA(12,12,20),BB(12,12,20),CC(12,12,20),	02288
	1	STAMAT(12,12,20)	02289
ISN 0011		COMMON /REORDR/ P(12,12),PT(12,12)	02290
ISN 0012		COMMON /STATC/ SH(12,12,20),F(12,21)	02291
ISN 0013		COMMON /HOLE/ IHOLE	02292*34
ISN 0014		COMMON /SMAXX/ SMAX1(8),SMAX2(8),SMAX3(8),SMAXLE(8),SMAXTE(8)	02293*37
ISN 0015		COMMON /MODAL/ SIGMA(3,8,25)	02294*37
ISN 0016		COMMON /FAIL/ XXX(42),TSAI(8,25)	02295*37
ISN 0017		DIMENSION SKPP1(12,12,20),SKPP2(12,12,20)	02296*34
ISN 0018		DIMENSION DS(12,12),CM(12,12),SKKOLD(12,12,20),FOLD(12,21)	02297
ISN 0019		DIMENSION DS1(12,12)	02298
	C		02299
ISN 0020		IF(ILOOP .GT. 1) GO TO 999	02300
ISN 0022		LOOP = 0	02301*34
ISN 0023		RPM = RPM1	02302*35
	C		02303
	C	FOR CHECK OUT RUNS IN W137 THIS CALL TO INPUT IS MADE, REMOVE	02304
	C	AFTER SUITABLE INPUT IS AVAILABLE FROM W137	02305
	C		02306
ISN 0024		CALL INPUT(NR,NC)	02307
	C		02308
	C	DEFINE THE LOCAL NODAL COORDINATE SYSTEMS	02309
	C		02310
ISN 0025		CALL CORD2R(X,Y,Z,NR,NC)	02311
	C		02312
	C	CALCULATE THE LAMINA STRESS-STRAIN RELATION ,QIJ	02313
	C		02314
ISN 0026		CALL LAMINA	02315
	C		02316
	C	DEFINE THE REORDER MATRICES P AND P1	02317
	C		02318
ISN 0027		CALL PPI(1)	02319
	C		02320
	C	SET UP THE BOUNDARY CONDITIONS FOR A FIXED ROOT CONDITION	02321
	C		02322
ISN 0028		DO 50 I = 1,6	02323
ISN 0029		DO 50 J = 1,6	02324
ISN 0030	50	C(I,J) = 0.0	02325
	C		02326
	C	FIND EACH SECTION STIFFNESS AND CORRESPONDING HOLZER STIFFNESS	02327
	C	ALSO DETERMINE THE NODAL LUMPED MASS MATRICES	02328

1

11

```

C 02329
ISN 0031 CALL HOLZER(1,NR,NC,RPM1) 02330
C 02331
C PERFORM THE STATIC ANALYSIS 02332
C 02333
ISN 0032 WRITE(6,100) RPM1 02334
ISN 0033 100 FORMAT(//,5X,'STATIC ANALYSIS, RPM =',E12.5,/) 02335
C 02336
ISN 0034 CALL STATIC(NR,NC,1) 02337
C 02338
C SET THE INITIAL VALUE OF SKKOLD - THE ORIGINAL STIFFNESSES 02339
C 02340
C NRM1 = NR - 1 02341
ISN 0035 DO 201 I = 1,NRM1 02342
ISN 0036 DO 201 J = 1,12 02343
ISN 0037 DO 201 K = 1,12 02344
ISN 0038 201 SKKOLD(J,K,I) = SKK(J,K,I) 02345
ISN 0039 DO 203 I = 1,NR 02346
ISN 0040 DO 203 J = 1,12 02347
ISN 0041 203 FOLD(J,I) = F(J,I) 02348
C 02349
C LOCAL BIRD INGESTION PARAMETER CALCULATION 02350
C 02351
ISN 0043 CALL FOD( NR,NC,RPM1,ALPHA ) 02352
C 02353
ISN 0044 999 IF(ISERCH .GT. 1) GO TO 998 02354
C 02355
C UPDATE THE STIFFNESS MATRICES FOR DIFFERENTIAL STIFFNESS 02356
C AND CENTRIFUGAL RESTORATION - ALSO UPDATE THE LOAD VECTOR, F 02357
C 02358
ISN 0046 888 LOOP = LOOP + 1 02359*34
ISN 0047 RATIO = 1. 02360*34
ISN 0048 IF(LOOP .EQ. 2) RATIO = (RPM2 / RPM1)**2 02361*34
C WRITE(6,500) ISERCH,LOOP,RATIO 02362*34
C500 FORMAT(//,5X,'**',2I5,E12.5,'**',/) 02363*34
ISN 0050 DO 200 I = 1,NRM1 02364
ISN 0051 DO 200 J = 1,12 02365
ISN 0052 DO 200 K = 1,12 02366
ISN 0053 200 SKK(J,K,I) = SKKOLD(J,K,I) + RATIO * (DSKK(J,K,I) - CMGG(J,K,I)) 02367
ISN 0054 DO 202 I = 1,NR 02368
ISN 0055 DO 202 J = 1,12 02369
ISN 0056 202 F(I,J) = FOLD(I,J) 02370*34
C 02371
C NOW REFORMULATE THE HOLZER STIFFNESSES FOR EACH SECTION 02372
C 02373
ISN 0057 CALL HOLZER(2,NR,NC,RPM1) 02374
C 02375
C FOR RPM1 - HOLZER STIFFNESS MATRIX IS SKPP1(12,12,20) 02376*34
C RPM2 - HOLZER STIFFNESS MATRIX IS SKPP2(12,12,20) 02377*34
C 02378*34
ISN 0058 DO 400 I = 1,NRM1 02379*34
ISN 0059 DO 400 J = 1,12 02380*34
ISN 0060 DO 400 K = 1,12 02381*34
ISN 0061 IF(LOOP .EQ. 1) SKPP1(J,K,I) = SKPP(J,K,I) 02382*34
ISN 0063 IF(LOOP .EQ. 2) SKPP2(J,K,I) = SKPP(J,K,I) 02383*34
ISN 0065 400 CONTINUE 02384*34
C 02385*34
ISN 0066 IF(LOOP .EQ. 1) GO TO 888 02386*34
C 02387*34

```

148

	C	RESET THE HOLZER MATRIX FOR THE STATIC ANALYSIS	02388*34
	C		02389*34
	C	WRITE(6,501)	02390*34
	C501	FORMAT(/,5X,'** RESET SKPP MATRIX **',/)	02391*34
ISN 0068		DO 410 I = 1,NRM1	02392*34
ISN 0069		DO 410 J = 1,12	02393*34
ISN 0070		DO 410 K = 1,12	02394*34
ISN 0071	410	SKPP(J,K,I) = SKPP1(J,K,I)	02395*34
	C		02396*34
	C		02397
	C	STATIC ANALYSIS WITH PRESTRESS INCLUDED	02398
	C		02399
ISN 0072		WRITE(6,101) RPM1,RPM2	02400
ISN 0073	101	FORMAT(/,5X,'STATIC ANALYSIS WITH PRESTRESS',/,5X, 1'RPM1 = ',E12.5,5X,'RPM2 = ',E12.5,/)	02401
	C		02402
ISN 0074		CALL STATIC(NR,NC,2)	02403
	C		02404
	C	PERFORM THE STRESS ANALYSIS	02405
	C		02406
ISN 0075		CALL ZSTRES (2,NC,SMAX1)	02407
	C		02408*39
	C	CALCULATE TSAI-WU FOR LE AND TE ROOT STRESS - SMAXLE,SMAXTE	02409*37
	C		02410*37
ISN 0076		CALL TSAIWU (SIGMA,NC)	02411*37
	C		02412*37
ISN 0077		DO 330 I = 1,8	02413*37
ISN 0078		SMAXLE(I) = TSAI(I,1)	02414*37
ISN 0079	330	SMAXTE(I) = TSAI(I,NC)	02415*37
	C		02416*37
	C	HOLLOW SECTION STRESS	02417*37
	C		02418*37
ISN 0080		IF(IHOLE .NE. 0) CALL ZSTRES (IHOLE,NC,SMAX2)	02419*37
ISN 0082		IHOLE1 = IHOLE + 1	02420*36
ISN 0083		IF(IHOLE .NE. 0) CALL ZSTRES(IHOLE1,NC,SMAX3)	02421*40
ISN 0085		IF(IHOLE .EQ. 0) GO TO 501	02422*40
ISN 0087		DO 500 I = 1,8	02423*40
ISN 0088		IF(SMAX3(I) .GT. SMAX2(I)) SMAX2(I) = SMAX3(I)	02424*40
ISN 0090	500	CONTINUE	02425*40
ISN 0091	501	CONTINUE	02426*40
	C		02427*40
	C		02428*40
	C	FORM MASS * STIFFNESS PRODUCTS TO BE SENT TO W137 SOLVER	02429
	C	-MUST CHANGE SIGN ON 21 AND 22 QUADRANTS OF SKPP FIRST	02430
	C	SKPP = HOLZER STIFFNESS MATRICES FOR EACH SECTION	02431
	C		02432
ISN 0092	998	FREQ2 = FREQ * FREQ	02433
ISN 0093		IF(FREQ .LT. 0.) RETURN	02434
ISN 0095		WRITE(6,1) RPM1,RPM3,FREQ	02435
ISN 0096	1	FORMAT(/,5X,'FREQUENCY SEARCH',5X,'RPM1=',E12.5,/, 1 21X,'RPM2=',E12.5,/,21X,'STARTING FREQ.=' ,E12.5,/)	02436*34
			02437
ISN 0097		DO 300 ISEC = 1,NRM1	02438
ISN 0098		DO 301 I = 1,12	02439
ISN 0099		DO 301 J = 1,12	02440
ISN 0100		SKP(I,J) = SKPP1(I,J,ISEC)	02441
ISN 0101		IF(RPM3 .NE. RPM1) SKP(I,J) = SKPP2(I,J,ISEC)	02442*34
ISN 0103		IF(I .GT. 6) SKP(I,J) = -SKP(I,J)	02443*34
ISN 0105	301	CONTINUE	02444
	C		02445
	C		02446

```

+VERSION 1.3.0 (01 MAY 80) STABEL SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 82.141/10.50.24 PAGE 4
C FORM MASS MATRIX AND STORE IT IN SK ARRAY 02447
C 02448
ISN 0106 DO 305 I = 1,12 02449
ISN 0107 DO 305 J = 1,12 02450
ISN 0108 305 SK(I,J) = 0. 02451
ISN 0109 DO 306 I = 1,6 02452
ISN 0110 SK(I,I) = 1. 02453
ISN 0111 306 SK(I+6,I+6) = -1. 02454
C 02455
C NOW INSERT SMM INTO THE 21 QUADRANT OF THE MASS MATRIX SK 02456
C ACTUALLY THE PRODUCT FREQ**2 * SMM IS INSERTED 02457
C 02458
ISN 0112 DO 310 I = 1,6 02459
ISN 0113 DO 310 J = 1,6 02460
ISN 0114 310 SK(I+6,J) = SK(I+6,J) + FREQ2 * SMM(I,J,ISEC+1) 02461
C 02462
ISN 0115 CALL MATMPY(SK,SKP,DS1,12,12,12) 02463
C CALL MATPRN(DS1,12,12,' DS1') 02464
C 02465
C REORDER THE MASS STIFFNESS PRODUCT MATRIX TO AGREE WITH 02466
C THE W137 ORDER 02467
C 02468
ISN 0116 CALL PPI(2) 02469
ISN 0117 CALL MATMPY(DS1,PT,CM,12,12,12) 02470
ISN 0118 CALL MATMPY(P,CM,DS,12,12,12) 02471
C CALL MATPRN(DS,12,12,'DSCP') 02472
C 02473
C FILL IN THE SMK MATRIX 02474
C 02475
ISN 0119 DO 320 I = 1,12 02476
ISN 0120 DO 320 J = 1,12 02477
ISN 0121 320 STAMAT(I,J,ISEC) = DS(I,J) 02478
C 02479
ISN 0122 300 CONTINUE 02480
C 02481
ISN 0123 RETURN 02482
ISN 0124 END 02483

```

```

*OPTIGNS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 123, PROGRAM SIZE = 77400, SUBPROGRAM NAME =STABEL
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

2984K BYTES OF CORE NOT USED

150

ISN 0002	C	SUBROUTINE STATIC(NR,NC,ISKIP)	02484
	C		02485
	C	THIS ROUTINE PERFORMS THE STATIC SOLUTION WITH THE	02486
	C	HOLZER METHOD	02487
	C		02488
	C		02489
ISN 0003		IMPLICIT REAL*8(A-H,O-Z)	02490
ISN 0004		COMMON /STIFF/ SK(12,12),SKK(12,12,21),SMM(6,6,21)	02491
ISN 0005		COMMON /DSTIFF/ CNGG(12,12,20),DSKK(12,12,20)	02492
ISN 0006		COMMON /HSTIFF/ SKP(12,12),SKPP(12,12,21)	02493
ISN 0007		COMMON /STATC/ SH(12,12,20),F(12,21)	02494
ISN 0008		COMMON /BC/ C(6,6),RPH	02495
ISN 0009		COMMON /STRS/ RBE(12,12,25,21),FOUT(12,21),SL(25,21),SM(25,21)	02496*34
ISN 0010		DIMENSION SH11(6,6),SH12(6,6),SH21(6,6),SH22(6,6)	02497
ISN 0011		DIMENSION SHH(12,12),D1(6,6),RL(6),RM(6),FOUT1(12,1)	02498
ISN 0012		DIMENSION FF(12,1),F1(6,1),U(6,1),FF1(6,1),FF2(6,1)	02499
ISN 0013		DIMENSION SDISPL(12,1),SFORCE(12,1),BDISPL(12,1),BFORCE(12,1)	02500
ISN 0014		DIMENSION RB(12,12)	02501
	C		02502
	C	FORM THE HOLZER STIFFNESS MATRIX STRING MULTIPLICATION	02503
	C		02504
ISN 0015		CALL SMULT(NR)	02505
	C		02506
	C	COMBINE LOAD VECTORS AND HOLZER STIFFNESS MATRIX	02507
	C	PRODUCTS INTO ONE PSUEDO LOAD VECTOR, FF.	02508
	C		02509
ISN 0016		CALL LOAD2 (NR,FF)	02510
	C		02511
	C	PARTITION THE SH(12,12,NR-1) MATRIX FOR SOLUTION	02512
	C	THE SHH IS A DUMMY MATRIX	02513
	C		02514
ISN 0017		DO 50 I = 1,12	02515
ISN 0018		DO 50 J = 1,12	02516
ISN 0019	50	SHH(I,J) = SH(I,J,NR-1)	02517
	C	CALL MATPRN(SHH,12,12,'SHH ')	02518
	C		02519
ISN 0020		CALL PARTN(SHH,12,12,SH11,SH12,SH21,SH22,6,6)	02520
	C		02521
	C	SOLVE FOR THE FORCE VECTOR AT NODE 1	02522
	C	F1 = (SH21*C + SH22)I * - FF(7-12)	02523
	C	WITH -I- MEANING THE INVERSE	02524
	C		02525
ISN 0021		DO 100 I = 1,6	02526
ISN 0022	100	FF2(I,1) = -FF(I+6,1)	02527
	C		02528
	C	CALL MATPRN(FF2,6,1,'FF2 ')	02529
ISN 0023		CALL MATMPY(SH21,C,D1,6,6,6)	02530
ISN 0024		CALL MATADD(D1,SH22,SH21,6,6,1.)	02531
	C	CALL MATPRN(SH21,6,6,'SH21')	02532
ISN 0025		CALL MINV(SH21,6,6,RL,RM)	02533
	C	CALL MATPRN(SH21,6,6,'SH21')	02534
ISN 0026		CALL MATMPY(SH21,FF2,F1,6,6,1)	02535
	C	CALL MATPRN(F1,6,1,' F1 ')	02536
	C		02537
	C	SOLVE FOR THE DISPLACEMENT AT NODE,NR	02538

```

C
ISN 0027      DO 110 I = 1,6                02539
ISN 0028      110 FF(I,1) = FF(I,1)        02540
C                                                    02541
C                                                    02542
ISN 0029      CALL MATMPY(SH11,C,SH21,6,6,6) 02543
ISN 0030      CALL MATADD(SH21,SH12,SH11,6,6,1.) 02544
ISN 0031      CALL MATMPY(SH11,F1,FF2,6,6,1) 02545
ISN 0032      CALL MATADD(FF2,FF1,U,6,1,1.) 02546
C                                                    02547
C                                                    02548
C                                                    02549
C THE VECTOR OF DISPL. AND FORCES AT EACH NODE WILL BE STORED
C IN , FOUT(12,21) - DISPL. IN 1-6 AND FORCES IN 7-12. 02550
C FIRST SOLVE FOR DEFLECTIONS AND FORCES AT NCDE 2    02551
C                                                    02552
C THIS SECTION ALSO GENERATES THE DIFFERENTIAL STIFFNESS MATRIX
C AND THE CENTRIFUGAL RESTORING STIFFNESS MATRIX     02553
C                                                    02554
C                                                    02555
ISN 0033      DO 120 I = 1,6                02556
ISN 0034      FOUT(I,NR) = U(I,1)          02557
ISN 0035      FOUT(I+6,NR) = F(I+6,NR)     02558
ISN 0036      FOUT(I,1) = 0.              02559
ISN 0037      FOUT(I+6,1) = F1(I,1)       02560
ISN 0038      FF(I,1) = 0.                02561
ISN 0039      120 FF(I+6,1) = F1(I,1)     02562
C                                                    02563
C GET THE HOLZER STIFFNESS FOR SECTION 1             02564
C                                                    02565
ISN 0040      DO 130 I = 1,6                02566
ISN 0041      DO 130 J = 1,6                02567
ISN 0042      SKP(I,J) = SKPP(I,J,1)      02568
ISN 0043      SKP(I,J+6) = SKPP(I,J+6,1)  02569
ISN 0044      SKP(I+6,J) = - SKPP(I+6,J,1) 02570
ISN 0045      130 SKP(I+6,J+6) = - SKPP(I+6,J+6,1) 02571
C                                                    02572
ISN 0046      CALL MATMPY(SKP,FF,FOUT1,12,12,1) 02573
C CALL MATPRN(FF,12,1,' FF ')            02574
C CALL MATFRN(FOUT1,12,1,' FOU1')        02575
C                                                    02576
C THE DIFFERENTIAL STIFFNESS FOR SECTION 1         02577
C                                                    02578
ISN 0047      IF(ISKIP .LE. 1) CALL KDGG(1,FF,FOUT1,DSKK) 02579
C                                                    02580
ISN 0049      DO 135 I = 1,12              02581
ISN 0050      135 FOUT(I,2) = FOUT1(I,1)  02582
C                                                    02583
C BEGIN LOOP TO SOLVE FOR THE REMAINING NODES 3 - (NR-1) 02584
C                                                    02585
ISN 0051      NRM1 = NR - 1                02586
ISN 0052      IF(NRM1 .EQ. 1) GO TO 201    02587
ISN 0054      DO 200 I = 3,NRM1           02588
C                                                    02589
C FILL IN THE DUMMY DISPL. AND FORCE VECTOR AT NODE ( I-1 ) 02590
C THE EQUILIBRIUM EQ WILL BE USED          02591
C                                                    02592
ISN 0055      DO 210 J = 1,6                02593
ISN 0056      FF(J,1) = FOUT(J,I-1)      02594
ISN 0057      210 FF(J+6,1) = -FOUT(J+6,I-1) + F(J+6,I-1) 02595
C                                                    02596
C GET THE HOLZER STIFFNESS MATRIX FOR SECTION ( I-1 ) 02597

```

1

151

	+VERSION 1.3.0 (01 MAY 80)	STATIC	SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)	DATE 82.141/10.50.28	PAGE 3
		C		02598	
ISH 0058			DO 220 J = 1,6	02599	
ISH 0059			DO 220 K = 1,6	02600	
ISH 0060			SKP(J,K) = SKPP(J,K,I-1)	02601	
ISH 0061			SKP(J,K+6) = SKPP(J,K+6,I-1)	02602	
ISH 0062			SKP(J+6,K) = -SKPP(J+6,K,I-1)	02603	
ISH 0063	220		SKP(J+6,K+6) = - SKPP(J+6,K+6,I-1)	02604	
		C		02605	
ISH 0064			CALL MATHPY(SKP,FF,FOUT1,12,12,1)	02606	
		C	CALL MATPRN(FF,12,1,' FF ')	02607	
		C	CALL MATPRN(FOUT1,12,1,'FOU1')	02608	
		C		02609	
		C	THE DIFFERENTIAL STIFFNESS FOR SECTION , I-1	02610	
		C		02611	
ISH 0065			ISEC = I-1	02612	
ISH 0066			IF(ISKIP .LE. 1) CALL KDGG(ISEC,FF,FOUT1,DSKK)	02613	
		C		02614	
ISH 0068			DO 225 J = 1,12	02615	
ISH 0069	225		FOUT(J,I) = FOUT1(J,1)	02616	
		C		02617	
ISH 0070	200		CONTINUE	02618	
		C		02619	
		C	THE DIFFERENTIAL STIFFNESS FOR SECTION NR-1	02620	
		C		02621	
ISH 0071			ISEC = NR - 1	02622	
ISH 0072			DO 140 I = 1,6	02623	
ISH 0073			FOUT1(I,1) = FOUT(I,NR)	02624	
ISH 0074			FOUT1(I+6,1) = FOUT(I+6,NR)	02625	
ISH 0075			FF(I,1) = FOUT(I,ISEC)	02626	
ISH 0076	140		FF(I+6,1) = -FOUT(I+6,ISEC) + F(I+6,ISEC)	02627	
		C		02628	
		C	CALL MATPRN(FF,12,1,' FF ')	02629	
		C	CALL MATPRN(FOUT1,12,1,'FOU1')	02630	
ISH 0077			IF(ISKIP .LE. 1) CALL KDGG(ISEC,FF,FOUT1,DSKK)	02631	
		C		02632	
ISH 0079	201		CONTINUE	02633	
ISH 0080			IF(ISKIP .EQ. 1) RETURN	02634*34	
		C		02635	
		C	FIND THE LOCAL BEAM DEFLECTIONS AND FORCES FOR EACH SECTION	02636	
		C		02637	
ISH 0082			DO 300 I = 1,NRMI	02638	
ISH 0083			IP = (I-1) * NC + 1	02639	
ISH 0084			WRITE(6,305) I	02640	
ISH 0085	305		FORMAT(/,5X,'DEFLECTIONS AND FORCES FOR SECTION',I5,/)	02641	
		C		02642	
		C	DEFINE SECTION ARRAYS FOR DISPL. AND FORCES	02643	
		C		02644	
ISH 0086			DO 301 J = 1,6	02645	
ISH 0087			SDISPL(J,1) = FOUT(J,I)	02646	
ISH 0088			SDISPL(J+6,1) = FOUT(J,I+1)	02647	
ISH 0089			SFORCE(J,1) = FOUT(J+6,I)	02648	
ISH 0090	301		SFORCE(J+6,1) = FOUT(J+6,I+1)	02649	
		C		02650	
		C	DEFLECTIONS AND FORCES FOR EACH BEAM	02651	
		C		02652	
ISH 0091			DO 400 J = 1,NC	02653	
		C		02654	
ISH 0092			DO 401 K = 1,12	02655	
ISH 0093			DO 401 L = 1,12	02656	

```

ISN 0094      401 RB(K,L) = RBE(K,L,J,I)          02657
              C                                  02658
ISN 0095      CALL MATMPY(RB,SDISPL,BDISPL,12,12,1) 02659
ISN 0096      CALL MATMPY(RB,SFORCE,BFORCE,12,12,1) 02660
              C                                  02661
ISN 0097      IPNC = IP + NC                      02662
ISN 0098      WRITE(6,404) J,IP,IPNC             02663
ISN 0099      404 FORMAT(5X,'BEAM',I5,' DEFLECTIONS, END 1=',I5,' END 2=',I5) 02664
ISN 0100      WRITE(6,402) (BDISPL(K,1),K=1,6)    02665
ISN 0101      402 FORMAT(2X,'END 1',6E12.5)      02666
ISN 0102      WRITE(6,403) (BDISPL(K,1),K=7,12)  02667
ISN 0103      403 FORMAT(2X,'END 2',6E12.5)     02668
ISN 0104      WRITE(6,405)                       02669
ISN 0105      405 FORMAT(14X,'FORCES')           02670
ISN 0106      WRITE(6,402) (BFORCE(K,1),K=1,6)   02671
ISN 0107      WRITE(6,403) (BFORCE(K,1),K=7,12)  02672
              C                                  02673
ISN 0108      400 IP = IP + 1                    02674
ISN 0109      300 CONTINUE                       02675
              C                                  02676
ISN 0110      WRITE(6,240)                       02677
ISN 0111      240 FORMAT(/,5X,' THE STATIC SOLUTION',/,5X,'DISPL. AND ROTATIONS',/, 02678
              15X,'U',10X,'V',10X,'W',10X,'RX',10X,'RY',10X,'RZ',/) 02679
ISN 0112      DO 250 I = 1,NR                    02680
ISN 0113      250 WRITE(6,260) (FOUT(J,I),J=1,6) 02681
ISN 0114      260 FORMAT(6E12.5)                02682
ISN 0115      WRITE(6,270)                      02683
ISN 0116      270 FORMAT(/,5X,'FORCES AND MOMENTS',/,5X,'FX',10X,'FY',10X,'FZ', 02684
              110X,'MX',10X,'MY',10X,'MZ',/)    02685
ISN 0117      DO 280 I = 1,NR                    02686
ISN 0118      280 WRITE(6,260) (FOUT(J,I),J=7,12) 02687
              C                                  02688
              C                                  02689
ISN 0119      RETURN                             02690
ISN 0120      END                                02691

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOBBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 119, PROGRAM SIZE = 7468, SUBPROGRAM NAME =STATIC
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

2988K BYTES OF CORE NOT USED

REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

SOURCE EBCDIC NOLIST NODECK OBJECT NOHAP NOFCRMT GOSTHT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```

154
C      02692
ISN 0002      SUBROUTINE ZSTRES (ISEC,NC,SMAX)      02693*36
C      02694
C      THIS ROUTINE PERFORMS THE STRESS ANALYSIS FOR EACH COMPOSITE      02695
C      BEAM IN THE SECTION      02696
C      02697
ISN 0003      IMPLICIT REAL*8(A-H,O-Z)      02698
ISN 0004      COMMON /STRS/ RBE(12,12,25,21),FOUT(12,21),SL(25,21),SW(25,21)      02699*34
ISN 0005      COMMON /PLY1/ PLY(21,25,7),THETA(7),RHO(7)      02700
ISN 0006      COMMON /ZCOEF/ E11(7),E22(7),E33(7),G12(7),G23(7),G13(7),      02701
1          V12(7),V13(7),V23(7)      02702
ISN 0007      COMMON /REORDR/ P(12,12),PT(12,12)      02703
ISN 0008      COMMON /FAIL/ VV(42),TSAI(8,25)      02704*35
ISN 0009      COMMON /MODAL/ SIGMA(3,8,25)      02705*35
ISN 0010      DIMENSION BDISPL(12,1),SDISPL(12,1),STRN1(6,1),STRN2(6,1)      02706
ISN 0011      DIMENSION R(12,12),TRANS(6,6),STR1(3,1),STR2(3,1)      02707*34
ISN 0012      DIMENSION D3(3,1),D4(3,1),Q(3,3),SIGMA1(3,1),SIGMA2(3,1)      02708
ISN 0013      DIMENSION TT(3,3),SMAX(8)      02709*35
ISN 0014      DIMENSION TTR(3,3),TTI(3,3),QB(3,3),QB1(3,3)      02710*43
C      02711
ISN 0015      TRANS(6,5) = -1.      02712
C      02713
ISN 0016      DO 100 I = 1,NC      02714
C      02715
C      TRANSFER THE SECTION NODE DISPLACEMENTS IN THE FOUT ARRAY      02716
C      TO BEAM I      02717
C      02718
ISN 0017      DO 110 J = 1,12      02719
ISN 0018      DO 110 K = 1,12      02720
ISN 0019      110 R(J,K) = RBE(J,K,I,ISEC)      02721*34
C      CALL MATPRN(R,12,12,' R ')      02722
ISN 0020      DO 120 J = 1,6      02723
ISN 0021      SDISPL(J,1) = FOUT(J,ISEC)      02724*41
ISN 0022      120 SDISPL(J+6,1) = FOUT(J,ISEC+1)      02725*41
C      CALL MATPRN(SDISPL,12,1,'SDIS')      02726
C      02727
ISN 0023      CALL MATMPY(R,SDISPL,BDISPL,12,12,1)      02728
C      CALL MATPRN(BDISPL,12,1,'BDIS')      02729
C      02730*34
C      SHAPE FUNCTION COEFFICIENT EVALUATION      02731*34
C      02732*34
ISN 0024      BL = SL(I,ISEC)      02733*34
ISN 0025      U1 = BDISPL(1,1)      02734*34
ISN 0026      V1 = BDISPL(2,1)      02735*34
ISN 0027      W1 = BDISPL(3,1)      02736*34
ISN 0028      RX1 = BDISPL(4,1)      02737*34
ISN 0029      RY1 = BDISPL(5,1)      02738*34
ISN 0030      RZ1 = BDISPL(6,1)      02739*34
ISN 0031      U2 = BDISPL(7,1)      02740*34
ISN 0032      V2 = BDISPL(8,1)      02741*34
ISN 0033      W2 = BDISPL(9,1)      02742*34
ISN 0034      RX2 = BDISPL(10,1)      02743*34
ISN 0035      RY2 = BDISPL(11,1)      02744*34
ISN 0036      RZ2 = BDISPL(12,1)      02745*34
C      02746*34

```

```

ISN 0037      A2 = 0.                                02747*34
ISN 0038      B2 = ( U2 - U1 ) / BL                  02748*34
ISN 0039      B1 = 0.                                02749*34
ISN 0040      C1 = 3. * (W2 - W1) / BL**2 + (2.*RY1 + RY2) / BL 02750*34
ISN 0041      D1 = 2. * (W1 - W2) / BL**3 - (RY1 + RY2) / BL**2 02751*34
ISN 0042      B = 0.                                  02752*34
ISN 0043      C = 3. * (V2 - V1) / BL**2 - (2.*RZ1 + RZ2) / BL 02753*34
ISN 0044      D = 2. * (V1 - V2) / BL**3 + (RZ1 + RZ2) / BL**2 02754*34
C                                                     02755*34
C CALCULATE THE DEFLECTIONS AND SLOPES AT 10% OF BEAM LENGTH 02756*34
C                                                     02757*34
ISN 0045      X = 0.25 * SL(I,ISEC)                  02758*34
C                                                     02759*34
ISN 0046      BDISPL(7,1) = A2 + B2 * X              02760*34
ISN 0047      BDISPL(11,1) = -2. * C1 * X - 3. * D1 * X**2 - B1 02761*34
ISN 0048      BDISPL(12,1) = 2. * C * X + 3. * D * X**2 + B     02762*34
C                                                     02763*34
C                                                     02764
C                                                     02765
C BEGIN LOOP FOR THE SEVEN LAYERS IN EACH BEAM      02766
C                                                     02767
ISN 0049      T = 0.                                  02768
ISN 0050      DO 51 K = 1,7                           02769
ISN 0051      51 T = T + PLY(ISEC,I,K)                02770*34
ISN 0052      Y = - T / 2.                            02771*34
ISN 0053      Z = SW(I,ISEC) / 2.                     02772*34
C                                                     02773*34
C DETERMINE THE RADIAL DEFLECTIONS AT THE BEAM CORNERS 02774*34
C U1,U2,U3,U4                                         02775*34
C                                                     02776*34
ISN 0054      U1 = BDISPL(7,1) + T/2. * BDISPL(12,1) - Z * BDISPL(11,1) 02777*34
ISN 0055      U2 = BDISPL(7,1) - T/2. * BDISPL(12,1) - Z * BDISPL(11,1) 02778*34
ISN 0056      U3 = BDISPL(7,1) - T/2. * BDISPL(12,1) + Z * BDISPL(11,1) 02779*34
ISN 0057      U4 = BDISPL(7,1) + T/2. * BDISPL(12,1) + Z * BDISPL(11,1) 02780*34
C                                                     02781*34
C WRITE(6,500) I,U1,U2,U3,U4                          02782*36
C500 FORMAT(5X,'** BEAM, CORNER DEFLECS.',I5,4E12.5) 02783*36
C                                                     02784*34
C CORRESPONDING FILAMENT STRAINS , SX1,SX2,SX3,SX4 02785*34
C                                                     02786*34
ISN 0058      SX1 = U1 / X                            02787*34
ISN 0059      SX2 = U2 / X                            02788*34
ISN 0060      SX3 = U3 / X                            02789*34
ISN 0061      SX4 = U4 / X                            02790*34
C WRITE(6,501) SL(I,ISEC),SX1,SX2,SX3,SX4            02791*36
C501 FORMAT(5X,'LENGTH,CORNER STRAINS',5E12.5)      02792*36
C                                                     02793*34
C SLOPE OF STRAIN-THICKNESS LINE - SIDE Z = -SW/2 - SLOPE1 02794*34
C Z = +SH/2 - SLOPE2                                 02795*34
C                                                     02796*34
ISN 0062      SLOPE1 = ( SX2 - SX1 ) / T              02797*34
ISN 0063      SLOPE2 = ( SX3 - SX4 ) / T              02798*34
C WRITE(6,502) SLOPE1,SLOPE2                         02799*36
C502 FORMAT(5X,'STRAIN-THICKNESS SLOPES',2E12.5)    02800*36
C                                                     02801*34
C LOOP TO SOLVE FOR LAYER STRESSES                   02802*34
C                                                     02803*34
ISN 0064      DO 200 LAYER = 1,8                     02804*34
ISN 0065      TEFF = T/2. + Y                        02805*34

```

156

```

C
ISN 0066      SIDE1 = SLOPE1 * TEFF + SX1      02806*34
ISN 0067      SIDE2 = SLOPE2 * TEFF + SX4      02807*34
C
C503 WRITE(6,503) I,LAYER,T,Y,TEFF,SIDE1,SIDE2 02808*34
C          FORMAT(5X,'BEAM,LAYER,T,Y,TEFF,S1,S2',/,5X,2I5,5X,5E12.5) 02809*36
C
ISN 0068      YOLD = Y                        02810*36
ISN 0069      IF(LAYER .GT. 7) GO TO 201      02811*34
ISN 0071      Y = Y + PLY(ISEC,I,LAYER)      02812*41
ISN 0072      201 II = LAYER                  02813*34
ISN 0073      IF(II .GE. 5) II = II - 1      02814*34
C
C          02815*34
C          02816*34
C          02817*34
C          02818
C          02819*43
C          02820*43
C          02821*43
C          02822*43
C          02823*43
C          02824*43
C          02825*43
C          02826
C          02827
C          02828
C          02829*43
C          02830*43
C          02831*43
C          02832*43
C          02833*43
C          02834*43
C          02835*43
C          02836*43
C          02837*43
C          02838*43
C          02839*43
C          02840*43
C          02841*43
C          02842*44
C          02843*43
C          02844*43
C          02845*43
C          02846*43
C          02847*43
C          02848*44
C          02849*44
C          02850*44
C          02851*44
C          02852*44
C          02853*44
C          02854*44
C          02855*34
C          02856*45
C          02857*34
C          02858
C          02859*43
C          02860
C          02861
C          02862
C          02863
C          02864

ISN 0075      CC = DCOS( THETA(II) )          02806*34
ISN 0076      SS = DSIN( THETA(II) )          02807*34
ISN 0077      C2 = CC * CC                    02808*34
ISN 0078      S2 = SS * SS                    02809*36
ISN 0079      CS = CC * SS                    02810*36
ISN 0080      TT(1,1) = C2                    02811*34
ISN 0081      TT(1,2) = S2                    02812*41
ISN 0082      TT(1,3) = 2. * CS               02813*34
ISN 0083      TT(2,1) = S2                    02814*34
ISN 0084      TT(2,2) = C2                    02815*34
ISN 0085      TT(2,3) = -2. * CS              02816*34
ISN 0086      TT(3,1) = -CS                   02817*34
ISN 0087      TT(3,2) = CS                    02818
ISN 0088      TT(3,3) = C2 - S2               02819*43
C
ISN 0089      DO 150 KK = 1,3                  02820*43
ISN 0090      DO 150 LL = 1,3                  02821*43
ISN 0091      TTI(LL,KK) = TT(LL,KK)          02822*43
ISN 0092      150 CONTINUE                    02823*43
C
ISN 0093      TTI(1,3) = -TTI(1,3)            02824*43
ISN 0094      TTI(2,3) = -TTI(2,3)            02825*43
ISN 0095      TTI(3,1) = -TTI(3,1)            02826
ISN 0096      TTI(3,2) = -TTI(3,2)            02827
C
ISN 0097      DO 155 KK = 1,3                  02828
ISN 0098      DO 155 LL = 1,3                  02829*43
ISN 0099      155 TTR(KK,LL) = TTI(LL,KK)      02830*43
C          CALL MATPRN(TT,3,3,'TT ')          02831*43
C          CALL MATPRN(TTI,3,3,'TTI ')         02832*43
C          CALL MATPRN(TTR,3,3,'TTR ')         02833*43
C
ISN 0100      IF(E11(II).EQ.0. .OR. PLY(ISEC,I,II).EQ.0.) GO TO 300 02834*43
C
C          02835*44
C          02836*44
C          02837*44
C          02838*44
C          02839*44
C          02840*44
C          02841*44
C          02842*44
C          02843*44
C          02844*44
C          02845*44
C          02846*44
C          02847*44
C          02848*44
C          02849*44
C          02850*44
C          02851*44
C          02852*44
C          02853*44
C          02854*44
C          02855*34
C          02856*45
C          02857*34
C          02858
C          02859*43
C          02860
C          02861
C          02862
C          02863
C          02864

ISN 0102      V21 = V12(II) * E22(II) / E11(II) 02806*34
ISN 0103      FACTOR = 1. - V12(II) * V21      02807*34
ISN 0104      Q(1,1) = E11(II) / FACTOR        02808*34
ISN 0105      Q(1,2) = V21 * Q(1,1)           02809*36

```

```

ISN 0106      Q(1,3) = 0.                                02865
ISN 0107      Q(2,1) = Q(1,2)                          02866
ISN 0108      Q(2,2) = E22(II) / FACTOR                02867
ISN 0109      Q(2,3) = 0.                              02868
ISN 0110      Q(3,1) = 0.                              02869
ISN 0111      Q(3,2) = 0.                              02870
ISN 0112      Q(3,3) = G12(II)                         02871
C                                                     02872*43
C                                                     02873*43
C ROTATED LAMINA STIFFNESS MATRIX , QB(I,J)           02874*43
C                                                     02875*43
ISN 0113      CALL MATMPY(Q,TTR,QB1,3,3,3)            02876*43
ISN 0114      CALL MATMPY(TTI,QB1,QB,3,3,3)           02877*43
C                                                     02878*43
ISN 0115      QF1 = QB(1,2) - QB(2,3) * QB(1,3) / QB(3,3) 02879*43
ISN 0116      QF2 = QB(2,2) - QB(2,3) * QB(2,3) / QB(3,3) 02880*43
C                                                     02881*43
ISN 0117      STR1(1,1) = SIDE1                        02882*43
ISN 0118      STR1(2,1) = -QF1 / QF2 * SIDE1           02883*43
ISN 0119      STR1(3,1) = (QB(2,3)/QB(3,3)*QF1/QF2-QB(1,3)/QB(3,3)) * SIDE1 02884*43
C                                                     02885*43
ISN 0120      STR2(1,1) = SIDE2                        02886*43
ISN 0121      STR2(2,1) = -QF1 / QF2 * SIDE2           02887*43
ISN 0122      STR2(3,1) = (QB(2,3)/QB(3,3)*QF1/QF2-QB(1,3)/QB(3,3)) * SIDE2 02888*34
C                                                     02889*34
C HOLLOW LAYER - ZERO Q(I,J)                         02890*34
C                                                     02891*34
ISN 0123      GO TO 301                                02892*34
ISN 0124      300 CONTINUE                             02893*34
ISN 0125      DO 310 KK = 1,3                          02894*43
ISN 0126      STR1(KK,1) = 0.                          02895*43
ISN 0127      STR2(KK,1) = 0.                          02896*34
ISN 0128      DO 310 LL = 1,3                          02897*43
ISN 0129      QB(KK,LL) = 0.                           02898*34
ISN 0130      310 Q(KK,LL) = 0.                        02899*34
ISN 0131      301 CONTINUE                             02900*44
C                                                     02901*44
C CALL MATPRN(Q,3,3,' Q ')                            02902*44
C CALL MATPRN(QB,3,3,' QB ')                          02903*44
C CALL MATPRN(STR1,3,1,'STR1')                        02904*43
C CALL MATPRN(STR2,3,1,'STR2')                        02905*43
C ROTATE LAMINA GEOMETRIC STRAINS TO FIBER AXIS      02906*43
C                                                     02907*44
ISN 0132      CALL MATMPY(TTR,STR1,D3,3,3,1)          02908*44
ISN 0133      CALL MATMPY(TTR,STR2,D4,3,3,1)          02909*43
C                                                     02910*43
C CALCULATE THE LAMINA STRESSES                       02911*43
C                                                     02912*34
C                                                     02913
C CALL MATPRN(Q,3,3,'QIJ ')                           02914
C                                                     02915
ISN 0134      CALL MATMPY(Q,D3,SIGMA1,3,3,1)          02916
C                                                     02917
ISN 0135      CALL MATMPY(Q,D4,SIGMA2,3,3,1)          02918
C CALL MATPRN(D3,3,1,' D3 ')                          02919*44
C CALL MATPRN(D4,3,1,' D4 ')                          02920*44
C                                                     02921
C                                                     02922*35
C AVERAGE THE SIDE STRESSES - STORE IN SIGMA        02923*43

```

158

```

C
ISN 0136      DO 210 J = 1,3                02924*35
ISN 0137      210 SIGMA(J,LAYER,I) = (SIGMA1(J,1) + SIGMA2(J,1)) / 2. 02925*35
C                                                    02926*35
C                                                    02927*35
C      WRITE(6,212) (SIGMA(JJ,LAYER,I),JJ=1,3) 02928*36
C212  FORMAT(5X,3E12.5)                        02929*36
C                                                    02930
ISN 0138      200 CONTINUE                  02931
ISN 0139      100 CONTINUE                  02932
C                                                    02933
C                                                    02934*35
C                                                    02935*35
C      CALCULATE THE TSAI-WU FAILURE CRITERION FOR EACH LAYER OF EACH
C      BEAM FOR THE SECTION                    02936*35
C                                                    02937*35
ISN 0140      CALL TSAIHU ( SIGMA,NC )      02938*35
C                                                    02939*35
C      PICK OUT THE MAX TSAI-WU VALUE PER LAYER FOR ALL BEAMS 02940*35
C                                                    02941*35
ISN 0141      DO 220 I = 1,8                02942*35
ISN 0142      SMAX(I) = 0.                  02943*35
ISN 0143      DO 220 J = 1,NC              02944*35
ISN 0144      220 IF(SMAX(I) .LT. TSAI(I,J)) SMAX(I) = TSAI(I,J) 02945*35
C                                                    02946*35
C      CALL MATPRN(SMAX,8,1,'SMAX')          02947*36
ISN 0146      RETURN                        02948
ISN 0147      END                           02949

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 146, PROGRAM SIZE = 5184, SUBPROGRAM NAME =ZSTRES
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

2984K BYTES OF CORE NOT USED

REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

	C		02950
ISH 0002		SUBROUTINE THICK(II,NR,NC,X,Y,Z,THK,ISKIP)	02951
	C		02952
	C	DETERMINE THE LAYER THICKNESS ,TH, ASSUMING THICKNESS	02953
	C	SYMMETRY ABOUT THE CENTER AXIS	02954
	C		02955
ISH 0003		IMPLICIT REAL*8(A-H,O-Z)	02956
ISH 0004		COMMON /PLY1/ PPLY(21,25,7),THETA(7),RHO(7)	02957
ISH 0005		COMMON /LAYER/ TH(7,25),BMASS(25),BSPAN(25),BWIDTH(25)	02958
ISH 0006		COMMON /WIEGHT/ WL(7)	02959
ISH 0007		COMMON /ANAL33/ DUM6(6),NCD,NCK,DUM4(4)	02960*47
ISH 0008		COMMON /BLK 8/ SHB(21),ITTL(18),DUMMH(277)	02961*47
ISH 0009		DIMENSION X(1),Y(1),Z(1),THK(1),TP(7),T(1000),DIST(25)	02962
ISH 0010		DIMENSION PLY(7)	02963
	C		02964
ISH 0011		IP = (II-1) * NC	02965
	C		02966
	C	AVERAGED THICKNESS CALCULATION	02967
	C		02968
ISH 0012		DO 100 I = 1,NC	02969
ISH 0013		IP = IP + 1	02970
ISH 0014		T(IP) = (THK(IP) + THK(IP+NC)) / 2.	02971
	C		02972
	C	AVERAGE THE % VALUES FOR BOTH ENDS TO FIND LAYER THICKNESS	02973
	C		02974
ISH 0015		TOTAL = THK(IP)	02975
ISH 0016		DO 101 J = 1,7	02976
ISH 0017		101 TH(J,I) = PPLY(II,I,J)	02977*33
	C		02978
	C		02979
	C	DETERMINE THE BEAM LENGTH	02980
	C		02981
ISH 0018		XX = X(IP) - X(IP+NC)	02982
ISH 0019		YY = Y(IP) - Y(IP+NC)	02983
ISH 0020		ZZ = Z(IP) - Z(IP+NC)	02984
ISH 0021		BSPAN(I) =DSQRT(XX**2 + YY**2 + ZZ**2)	02985
	C		02986
	C	AVERAGED WIDTH	02987
	C		02988
ISH 0022		XX = X(IP) - X(IP+1)	02989
ISH 0023		YY = Y(IP) - Y(IP+1)	02990
ISH 0024		ZZ = Z(IP) - Z(IP+1)	02991
ISH 0025		DIST(I) =DSQRT(XX**2 + YY**2 + ZZ**2)	02992
ISH 0026		XX = X(IP+NC) - X(IP+NC+1)	02993
ISH 0027		YY = Y(IP+NC) - Y(IP+NC+1)	02994
ISH 0028		ZZ = Z(IP+NC) - Z(IP+NC+1)	02995
ISH 0029		DIST(I) = (DIST(I) + DSQRT(XX**2 + YY**2 + ZZ**2)) / 2.	02996
	C		02997
ISH 0030		100 CONTINUE	02998
	C		02999
	C	BEAM WIDTHS	03000
	C		03001
ISH 0031		NCM1 = NC - 1	03002
ISH 0032		IF(NCM1 .EQ. 0) GO TO 201	03003
ISH 0034		BWIDTH(1) = DIST(1)	03004

```

+VERSION 1.3.0 (01 MAY 80) THICK SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 82.141/10.50.49 PAGE 2
  ISN 0035          BWIDTH(NC) = DIST(NC-1)                                03005
  ISN 0036          DO 200 I = 2,NCM1                                    03006
  ISN 0037          200 BWIDTH(I) = (DIST(I-1) + DIST(I)) / 2.          03007
  ISN 0038          201 CONTINUE                                         03008
  C                                                         03009
  C BEAM MASS                                                         03010
  C                                                         03011
  ISN 0039          DO 300 I = 1,NC                                       03012
  ISN 0040          BMASS(I) = 0.                                          03013
  ISN 0041          AREA = BWIDTH(I) * BSPAN(I)                            03014
  ISN 0042          DO 11 J = 1,7                                          03015
  ISN 0043          11 BMASS(I) = BMASS(I) + AREA * TH(J,I) * RHO(J)     03016
  ISN 0044          300 CONTINUE                                           03017
  C                                                         03018
  C LAYER WIEGHT CALCULATION, RESULTS STORED IN WL(7)                 03019
  C                                                         03020
  ISN 0045          IF(ISKIP .GT. 1) RETURN                                03021
  ISN 0047          IF(II .GT. 1) GO TO 400                                03022
  ISN 0049          DO 401 I = 1,7                                          03023
  ISN 0050          401 WL(I) = 0.                                          03024
  ISN 0051          IF(NCD .EQ. 2) GO TO 400                                03025*47
  ISN 0053          WL(1) = RHO(1) * BSPAN(1) * SMB(2)**2 * 0.15          03026*47
  ISN 0054          WL(1) = WL(1) / 2.                                      03027*48
  ISN 0055          WL(7) = WL(1)                                          03028*47
  ISN 0056          RETURN                                                 03029*47
  ISN 0057          400 CONTINUE                                           03030
  ISN 0058          DO 410 J = 1,NC                                       03031
  ISN 0059          DO 410 I = 1,7                                          03032
  ISN 0060          410 WL(I) = WL(I) + RHO(I) * BSPAN(J) * BWIDTH(J) * TH(I,J) 03033
  ISN 0061          NRM1 = NR - 1                                          03034*34
  ISN 0062          IF(II .EQ. NRM1) CALL MATPRN(WL,7,1,' WL ')          03035*34
  C                                                         03036
  ISN 0064          RETURN                                                 03037
  ISN 0065          END                                                    03038
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECCUNT(60) SIZE(MAX) AUTODBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 64, PROGRAM SIZE = 9726, SUBPROGRAM NAME = THICK
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF CONPIlation *****
3004K BYTES OF CORE NOT USED

```

```

C          03039*35
ISN 0002  SUBROUTINE TSAIWU ( SIGMA,NC )          03040*35
ISN 0003  IMPLICIT REAL*8(A-H,O-Z)             03041*35
C          03042*35
C          03043*35
C          03044*35
ISN 0004  COMMON /FAIL/ X1T(7),X1C(7),X2T(7),X2C(7),S6P(7),S6M(7),TSAI(8,25)03045*35
ISN 0005  DIMENSION SIGMA(3,8,25),F1(8),F2(8),F6(8),F11(8),F22(8),F66(8) 03046*35
ISN 0006  DIMENSION F12(8)                     03047*35
C          03048*35
C          03049*35
C          03050*35
ISN 0007  L = 0                                03051*35
ISN 0008  DO 100 I = 1,8                       03052*35
ISN 0009  L = L + 1                            03053*35
ISN 0010  IF(I .EQ. 5) L = L - 1              03054*35
C          03055*35
ISN 0012  F1(I) = 1./X1T(L) - 1./X1C(L)        03056*35
ISN 0013  F2(I) = 1./X2T(L) - 1./X2C(L)        03057*35
ISN 0014  F6(I) = 1./S6P(L) - 1./S6M(L)        03058*35
ISN 0015  F11(I) = 1. / X1T(L) / X1C(L)        03059*35
ISN 0016  F22(I) = 1. / X2T(L) / X2C(L)        03060*35
ISN 0017  F66(I) = 1. / S6P(L) / S6M(L)        03061*35
ISN 0018  100 F12(I) = -DSQRT( F11(I) * F22(I) ) 03062*35
C          03063*36
C          03064*36
C          03065*36
C          03066*36
C          03067*36
C          03068*36
C          03069*36
C          03070*35
C          03071*35
C          03072*35
ISN 0019  DO 110 J = 1,NC                      03073*35
ISN 0020  DO 110 I = 1,8                       03074*35
ISN 0021  110 TSAI(I,J) = F1(I) * SIGMA(1,I,J) + F2(I) * SIGMA(2,I,J) + 03075*35
1          F6(I) * SIGMA(3,I,J) + F11(I) * SIGMA(1,I,J)**2 + 03076*35
2          F22(I) * SIGMA(2,I,J)**2 + F66(I) * SIGMA(3,I,J)**2 03077*35
3          + F12(I) * SIGMA(1,I,J) * SIGMA(2,I,J) 03078*35
C          03079*36
C          03080*35
C          03081*35
C          03082*35
ISN 0022  RETURN
ISN 0023  END

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
 *OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
 STATISTICS SOURCE STATEMENTS = 22, PROGRAM SIZE = 1112, SUBPROGRAM NAME =TSAIWU
 STATISTICS NO DIAGNOSTICS GENERATED
 ***** END OF COMPILATION *****

3016K BYTES OF CORE NOT USED

REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```

162
C      03083
ISN 0002  C      SUBROUTINE WARP(ISEC,NR,NC)      03084
C      03085
C      THIS ROUTINE CALCULATES THE WARPING TERMS REQUIRED FOR THE 03086
C      WARPING FUNCTION OF A THIN SECTION- THE VALUES WILL BE USED 03087
C      IN SUBROUTINE RBE - (SEE THE THEORY OF ELASTIC STABILITY, 03088
C      TIMOSHENKO AND GERE)      03089
C      03090
C      03091
ISN 0003  C      IMPLICIT REAL*8(A-H,O-Z)      03092
ISN 0004  COMMON /SC/XSC(21),YSC(21),ZSC(21),ALPHA(21),SECIP(21),SECA(21) 03093
ISN 0005  COMMON /INPUTT/ X(1000),Y(1000),Z(1000),T(1000)      03094
ISN 0006  COMMON /LAYER/ TH(7,25),BMASS(25),BSPAN(25),BWIDTH(25) 03095
ISN 0007  COMMON /WARPP/ WS(26,21),RN(25,21),RTT(25,21)      03096
ISN 0008  COMMON /COORD1/ CLOCAL(3,3,1000)      03097
ISN 0009  DIMENSION RT(26),THETA(26),SS(25),WSS(25),ROUT(26) 03098
ISN 0010  DIMENSION SIGN(25)      03099
C      03100
C      ARC LENGTH VARIABLE , S , AND SS.      03101
C      SS= CUMULATIVE ARC LENGTH FOR ELEMENT INTERVALS      03102
C      03103
ISN 0011  C      SS(1) = BWIDTH(1)      03104
ISN 0012  DO 60 I = 2,NC      03105
ISN 0013  60  SS(I) = SS(I-1) + BWIDTH(I)      03106
C      CALL MATPRN(SS,25,1,' SS ')      03107
C      03108
C      DETERMINE THE VALUE , WS , BEGINNING AT THE LEADING EDGE 03109
C      - MUST ALSO CALCULATE THE PERPENDICULAR DISTANCES FROM THE 03110
C      TANGENT,RT AND FROM THE NORMAL,RN TO THE SHEAR CENTER      03111
C      WS = EVALUATED AT NODE POINTS      03112
C      WSS = EVALUATED AT INTERVAL ENDS      03113
C      03114
ISN 0014  C      NODE = ( ISEC-1 ) * NC + 1      03115
ISN 0015  DO 100 I = 1,NC      03116
C      BOTTOM LAYER OF SECTION ISEC      03117
ISN 0016  C      RX = XSC(ISEC) - X(NODE)      03118
ISN 0017  C      RY = YSC(ISEC) - Y(NODE)      03119
ISN 0018  C      RZ = ZSC(ISEC) - Z(NODE)      03120
ISN 0019  C      RMAG = DSQRT( RX**2 + RY**2 + RZ**2 )      03121
ISN 0020  C      IF( RMAG .EQ. 0.) GO TO 50      03122
ISN 0022  C      RX = RX / RMAG      03123
ISN 0023  C      RY = RY / RMAG      03124
ISN 0024  C      RZ = RZ / RMAG      03125
C      03126
C      ANGLE BETWEEN VECTORS R AND LOCAL Z , THETA      03127
C      03128
ISN 0025  50  ARG = RX * CLOCAL(3,1,NODE) + RY * CLOCAL(3,2,NODE) + 03129
C      1  RZ * CLOCAL(3,3,NODE)      03130
ISN 0026  C      THETA(I) = DARCOS( ARG )      03131
C      03132
C      DETERMINE THE DIRECTION OF SHEEP, CW OR CCW, FOR THE SIGN ON 03133
C      WSS - NOTE: THIS CHECK USES THE COORDINATE POINTS NOT THE 03134
C      SIDES ON THE BEAM , SO THE CHECK IS APPROXIMATE      03135
C      03136
ISN 0027  C      IF(I .EQ. NC) SIGN(I) = SIGN(I-1)      03137
ISN 0029  C      IF(I .EQ. NC) GO TO 400

```

```

+VERSION 1.3.0 (01 MAY 80)      WARP      SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)      DATE 82.141/10.51.01      PAGE 2
  ISN 0031      V1Y = Y(NODE) - YSC(ISEC)      03138
  ISN 0032      V1Z = Z(NODE) - ZSC(ISEC)      03139
  ISN 0033      V2Y = Y(NODE+1) - YSC(ISEC)      03140
  ISN 0034      V2Z = Z(NODE+1) - ZSC(ISEC)      03141
  C      03142
  C      V1 CROSS V2 - THE RESULTANT X COMPONENT SETS SIGN OF, SIGN.      03143
  C      XCOMP GT. 0. - SIGN = 1      03144
  C      XCOMP LT. 0. - SIGN = -1      03145
  C      03146
  ISN 0035      XCOMP = V1Y * V2Z - V1Z * V2Y      03147
  ISN 0036      SIGN(I) = 1.      03148
  ISN 0037      IF(XCOMP .LT. 0.) SIGN(I) = -1.      03149
  C      WRITE(6,111) I,NC,SIGN(I)      03150
  C111      FORMAT(5X,'I,NC,SIGN(I)',2I5,E12.5)      03151
  ISN 0039      400 CONTINUE      03152
  C      03153
  ISN 0040      RN(I,ISEC) = -RMAG * DCOS( THETA(I) )      03154
  ISN 0041      RT(I      ) = RMAG * DSIN( THETA(I) )      03155
  ISN 0042      RTT(I,ISEC) = RT(I) * SIGN(I)      03156
  ISN 0043      ROUT(I) = RN(I,ISEC)      03157
  C      03158
  ISN 0044      IF( I .EQ. 1 ) GO TO 101      03159
  ISN 0046      WSS(I) = WSS(I-1) + SIGN(I) * RT(I) * (SS(I) - SS(I-1))      03160
  ISN 0047      GO TO 100      03161
  ISN 0048      101 CONTINUE      03162
  ISN 0049      WSS(I) = SIGN(I) * RT(I) * SS(I)      03163
  C      WRITE(6,112) I,SIGN(I),WSS(I)      03164
  C112      FORMAT(5X,'I,SIGN(I),WSS(I)',I5,2E12.5)      03165
  ISN 0050      100 NODE = NODE + 1      03166
  C      03167
  C      WRITE(6,52) (WSS(II),II=1,NC)      03168
  C52      FORMAT(5X,E10.4)      03169
  C      CALL MATPRN(ROUT,26,1,' RN ')      03170
  C      CALL MATPRN(RT,26,1,' RT ')      03171
  C      CALL MATPRN(THETA,26,1,'THET')      03172
  C      CALL MATPRN(SIGN,25,1,'SIGN')      03173
  C      03174
  C      CALCULATE THE AREA      03175
  C      03176
  ISN 0051      AREA = 0.      03177
  ISN 0052      NODE = (ISEC-1) * NC + 1      03178
  ISN 0053      DO 110 I = 1,NC      03179
  ISN 0054      AREA = AREA + T(NODE) * BWIDTH(I)      03180
  ISN 0055      110 NODE = NODE + 1      03181
  C      WRITE(6,53) AREA      03182
  C53      FORMAT(5X,'AREA=',E12.5)      03183
  C      03184
  C      AVERAGE WARPING , WSBAR      03185
  C      03186
  ISN 0056      WSBAR = 0.      03187
  ISN 0057      NODE = (ISEC-1) * NC + 1      03188
  ISN 0058      DO 200 I = 2,NC      03189
  ISN 0059      NODE = NODE + 1      03190
  ISN 0060      SSS = SS(I) - SS(I-1)      03191
  ISN 0061      200 WSBAR = WSBAR + T(NODE) * ( WSS(I) + WSS(I-1)) / 2. * SSS      03192
  C      03193
  C      ADD THE FIRST TERM AND DIVIDE BY THE TOTAL AREA      03194
  C      03195
  ISN 0062      NODE = (ISEC-1) * NC + 1      03196

```

```

+VERSION 1.3.0 (01 MAY 80)      WARP      SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)      DATE 82.141/10.51.01      PAGE 3
  ISN 0063      WSBAR = WSBAR + T(NODE) * WSS(1) / 2. * SS(1)      03197
    C54      FORMAT(5X,'WSBAR',E12.5)      03198
  ISN 0064      WSBAR = WSBAR / AREA      03199
    C      WRITE(6,54) WSBAR      03200
    C      03201
    C      FINALLY CALCULATE THE WARPING VALUE FOR RADIAL DISPLACEMENT,WS      03202
    C      WS= WSBAR- WS , NOTE;THIS IS NOT COMPLETE BECAUSE NO TWIST      03203
    C      GRADIENT HAS BEEN USED      03204
    C      03205
  ISN 0065      WS(1,ISEC) = WSBAR - WSS(1) / 2.      03206
  ISN 0066      DO 300 I = 2,NC      03207
  ISN 0067      300 WS(I,ISEC) = WSBAR - (WSS(I) + WSS(I-1)) / 2.      03208
    C      WRITE(6,52) (WS(II,ISEC),II=1,NC)      03209
    C      03210
    C      03211
  ISN 0068      RETURN      03212
  ISN 0069      END      03213

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS*      SOURCE STATEMENTS =      68, PROGRAM SIZE =      2582, SUBPROGRAM NAME = WARP
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

3004K BYTES OF CORE NOT USED

REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```

C          DATA SET U498DHOLLO AT LEVEL 003 AS OF 12/10/81
C          DATA SET U498HOLLOW AT LEVEL 001 AS OF 10/30/81
ISN 0002   SUBROUTINE HOLLOW(DLE,DTE,DROOT,DTIP,TTI,TBT,NC,NR)
C
C          THIS ROUTINE CALCULATES THE LAYER THICKNESSES FOR EACH BEAM
C          ALSO FOUND ARE THE BEAM LENGTH, BSPAN, AND THE WIDTH, BWIDTH
C
ISN 0003   IMPLICIT REAL*8 (A-H,O-Z)
ISN 0004   COMMON /LAYER/ TH(7,25),BMASS(25),BSPAN(25),BWIDTH(25)
ISN 0005   COMMON /INPUT/ X(1000),Y(1000),Z(1000),T(1000)
ISN 0006   COMMON /PLY1/ PLY(21,25,7),THETA(7),RHO(7)
ISN 0007   COMMON /HOLE/ IHOLE
ISN 0008   DIMENSION DIST(25)
C
C          A = CHORDWISE DISTANCE OF HOLLOW SECTION PENETRATION OF A BEAM
C          B = SAME AS A BUT IN THE SPAN DISRECTION
C          RBOT = RADIAL DISTANCE TO BOTTOM OF CAVITY
C          RTIP = RADIAL DIATANCE TO TOP OF CAVITY
C
C          CHECK FOR THE CAVITY EXISTENCE
C
ISN 0009   IHOLE = 0
ISN 0010   CAVITY = DLE + DTE + DROOT + DTIP
ISN 0011   IF( CAVITY .EQ. 0. ) GO TO 150
ISN 0013   GO TO 151
ISN 0014   150 NRM1 = NR - 1
ISN 0015   DO 160 I = 1,NRM1
ISN 0016   NODE = (I-1) * NC + 1
ISN 0017   DO 160 J = 1,NC
ISN 0018   TAVE = (T(NODE) + T(NODE+NC)) / 2.
ISN 0019   PLY(I,J,1) = TAVE / 2.
ISN 0020   PLY(I,J,2) = 0.
ISN 0021   PLY(I,J,3) = 0.
ISN 0022   PLY(I,J,4) = 0.
ISN 0023   PLY(I,J,5) = 0.
ISN 0024   PLY(I,J,6) = 0.
ISN 0025   PLY(I,J,7) = TAVE / 2.
ISN 0026   160 NODE = NODE + 1
ISN 0027   RETURN
ISN 0028   151 CONTINUE
C
ISN 0029   RBOT = X(NC/2) + DROOT
ISN 0030   RTIP = X(NC*NR-NC/2) - DTIP
C          WRITE(6,130) RBOT,RTIP
C130      FORMAT(5X,'RBOT,RTIP',2E12.5)
C
ISN 0031   NRM1 = NR - 1
ISN 0032   DO 100 I = 1,NRM1
ISN 0033   NODE = (I-1) * NC + 1
ISN 0034   NODE1 = NODE
ISN 0035   CHORD = 0.
C
ISN 0036   DO 110 J = 1,NC

```

```

C      BEAM LENGTH, BSPAN
C
ISN 0037      XX = X(NODE) - X(NODE+NC)
ISN 0038      YY = Y(NODE) - Y(NODE+NC)
ISN 0039      ZZ = Z(NODE) - Z(NODE+NC)
ISN 0040      BSPAN(J) = DSQRT(XX**2 + YY**2 + ZZ**2)
C      CALL MATPRN(BSPAN,25,1,'SPAN')
C
C      AVERAGED WIDTH, BWIDTH
C
ISN 0041      XX = X(NODE) - X(NODE+1)
ISN 0042      YY = Y(NODE) - Y(NODE+1)
ISN 0043      ZZ = Z(NODE) - Z(NODE+1)
ISN 0044      DIST(J) = DSQRT(XX**2 + YY**2 + ZZ**2)
ISN 0045      XX = X(NODE+NC) - X(NODE+NC+1)
ISN 0046      YY = Y(NODE+NC) - Y(NODE+NC+1)
ISN 0047      ZZ = Z(NODE+NC) - Z(NODE+NC+1)
ISN 0048      DIST(J) = (DIST(J) + DSQRT(XX**2 + YY**2 + ZZ**2)) / 2.
C
ISN 0049      110  NODE = NODE + 1
ISN 0050      NODE = NODE1 - 1
C
ISN 0051      NCM1 = NC - 1
ISN 0052      IF(NCM1 .EQ. 0) GO TO 201
ISN 0054      BWIDTH(1) = DIST(1)
ISN 0055      BWIDTH(NC) = DIST(NC-1)
ISN 0056      DO 111 JJ = 2,NCM1
ISN 0057      111  BWIDTH(JJ) = (DIST(JJ-1) + DIST(JJ)) / 2.
C      CALL MATPRN(BWIDTH,25,1,'WIDTH')
C
ISN 0058      201  CONTINUE
ISN 0059      DO 112 JJ = 1,NC
ISN 0060      112  CHORD = CHORD + BWIDTH(JJ)
C      WRITE(6,131) I,CHORD
C131  FORMAT(5X,'STATION, CHORD',I5,5X,E12.5)
C
ISN 0061      DO 100 J = 1,NC
ISN 0062      NODE = NODE + 1
C
ISN 0063      IF(X(NODE+NC) .LE. RBOT) GO TO 10
ISN 0065      IF(X(NODE+NC) .GT. RBOT .AND. X(NODE+NC) .LE. RTIP) GO TO 20
ISN 0067      IF(X(NODE+NC) .GT. RTIP) GO TO 30
C
C      THIS RADIAL SECTION IS SOLID
C
C
ISN 0069      10  B = 0.
ISN 0070      GO TO 50
C
C      THIS RADIAL SECTION IS HOLLOW / SOLID
C
C
ISN 0071      20  B = BSPAN(J)
ISN 0072      IF(X(NODE) .LT. RBOT) B = X(NODE+NC) - RBOT
ISN 0074      IF(IHOLE .EQ. 0) IHOLE = I
ISN 0076      GO TO 50
C
C      THIS RADIAL SECTION IS NEAR THE TIP, MAY INCLUDE SOME HOLLOW
C
C
ISN 0077      30  B = 0.
ISN 0078      IF(X(NODE) .LT. RTIP) B = RTIP - X(NODE)
C

```

ISN 0080	50	CONTINUE	00114
	C		00115
	C	NOW BEGIN CHORDWISE SEARCH FOR - A - DIMENSION	00116
	C		00117
ISN 0081		IF(B .GT. 0.) GO TO 60	00118
ISN 0083		GO TO 61	00119
	C		00120
	C	THIS IS A HOLLOW BEAM	00121
	C		00122
ISN 0084	60	CDIST = 0.	00123
ISN 0085		DO 120 JJ = 1,J	00124
ISN 0086	120	CDIST = CDIST + BWIDTH(JJ)	00125
	C		00126
	C	CDIST = DISTANCE FROM LE TO BEAM (J)	00127
	C	CTE = DISTANCE FROM LE TO CHORDWISE END OF CAVITY	00128
	C		00129
ISN 0087		CTE = CHORD - DTE	00130
	C		00131
ISN 0088		IF(CDIST .LE. DLE) GO TO 61	00132
ISN 0090		IF(CDIST .GT. DLE .AND. CDIST .LE. CTE) GO TO 62	00133
ISN 0092		CDIST1 = CDIST - BWIDTH(J)	00134
ISN 0093		IF(CDIST1 .LT. CTE .AND. CDIST .GT. CTE) GO TO 63	00135
ISN 0095		IF(CDIST1 .GT. CTE) GO TO 64	00136
	C		00137
	C	SOLID LE SECTION	00138
	C		00139
ISN 0097	61	A = 0.	00140
ISN 0098		GO TO 65	00141
	C		00142
	C	HOLLOW / SOLID SECTION - LE	00143
	C		00144
ISN 0099	62	A = CDIST - DLE	00145
ISN 0100		IF(A .GT. BWIDTH(J)) A = BWIDTH(J)	00146
ISN 0102		GO TO 65	00147
	C		00148
	C	HOLLOW / SOLID SECTION - TE	00149
	C		00150
ISN 0103	63	A = CTE - CDIST1	00151
ISN 0104		GO TO 65	00152
	C		00153
	C	SOLID TE SECTION	00154
	C		00155
ISN 0105	64	A = 0.	00156
ISN 0106	65	CONTINUE	00157
	C		00158
	C	DEFINE THICKNESS OF EACH LAYER IN BEAM (J)	00159
	C		00160
ISN 0107		IF(B .EQ. 0. .OR. A .EQ. 0.) GO TO 80	00161
	C		00162
ISN 0109		TAVE = (T(NODE) + T(NODE+NC)) / 2.	00163
ISN 0110		TCHECK = TAVE - 2. * TTI - 2. * TBT	00164
ISN 0111		IF(TCHECK) 90,90,95	00165
	C		00166
	C	NO HOLLOW SECTION DUE TO MINIMUM THICKNESS	00167
	C		00168
ISN 0112	90	TCHEK1 = TAVE - 2. * TTI	00169
ISN 0113		IF(TCHEK1) 91,91,92	00170
ISN 0114	91	TTI1 = TAVE / 2.	00171
ISN 0115		TBT1 = 0.	00172

168

```

ISN 0116      THOL = 0.                00173
ISN 0117      GO TO 200                00174
ISN 0118      92  TBT1 = TCHEK1 / 2.    00175
ISN 0119      TTI1 = TTI              00176
ISN 0120      THOL = 0.                00177
ISN 0121      GO TO 200                00178
C                                                    00179
C  HOLLOW SECTION THICKNESS IS NONZERO 00180
C                                                    00181
ISN 0122      95  THOL = A * B / BWIDTH(J) / BSPAN(J) * TCHECK 00182
ISN 0123      TBT1 = A * B / BWIDTH(J) / BSPAN(J) * TBT      00183
ISN 0124      TTI1 = TAVE - THOL - 2. * TBT1                 00184
ISN 0125      TTI1 = TTI1 / 2.                               00185
ISN 0126      GO TO 200                                       00186
C                                                    00187
C  SOLID SECTION - ALL TITANIUM        00188
C                                                    00189
ISN 0127      80  TTI1 = (T(NODE) + T(NODE+NC)) / 4.          00190
ISN 0128      TBT1 = 0.                                       00191
ISN 0129      THOL = 0.                                       00192
C                                                    00193
ISN 0130      200 CONTINUE                                       00194
C  WRITE(6,132) I,J,A,B                                         00195
C132 FORMAT(5X,'STATION,BEAM,A,B',2I5,2E12.5)                 00196
ISN 0131      PLY(I,J,1) = TTI1                                   00197
ISN 0132      PLY(I,J,2) = TBT1                                   00198
ISN 0133      PLY(I,J,3) = 0.                                    00199
ISN 0134      PLY(I,J,4) = THOL                                  00200
ISN 0135      PLY(I,J,5) = 0.                                    00201
ISN 0136      PLY(I,J,6) = TBT1                                   00202
ISN 0137      PLY(I,J,7) = TTI1                                   00203
C  WRITE(6,300) I,J,(PLY(I,J,K),K=1,7)                          00204**3
C300 FORMAT(5X,'STATION,BEAM,PLY',2I5,/,5X,7E12.5)           00205**3
C                                                    00206
ISN 0138      100 CONTINUE                                       00207
C                                                    00208
ISN 0139      RETURN                                           00209
ISN 0140      END                                              00210

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NONAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 139, PROGRAM SIZE = 2480, SUBPROGRAM NAME =HOLLOW
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****
2992K BYTES OF CORE NOT USED

```

REQUESTED OPTIONS: AUTODBL(DBL4),SOURCE

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)

SOURCE EBCDIC NOLIST NODACK OBJECT NONAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```

C          DATA SET U498DBMFIT AT LEVEL 003 AS OF 11/02/81
C          DATA SET T926BHFIT AT LEVEL 002 AS OF 05/25/79
ISN 0002          SUBROUTINE BMFIT2 ( X, Y, YP1, YPN, N, IP, A, B, C, D )
ISN 0003          DIMENSION X(N), Y(N), A(N), B(N), C(N), D(N)
ISN 0004          IF ( N - 2 ) 10, 30, 40
ISN 0005          10 WRITE (6,20) N
ISN 0006          20 FORMAT (//, 46H THE NUMBER OF DATA POINTS IS LESS THAN 2.  N =,I4)
ISN 0007          GO TO 240
ISN 0008          30 D(1) = 0.0
ISN 0009          C(1) = 0.0
ISN 0010          B(1) = ( Y(2) - Y(1) ) / ( X(2) - X(1) )
ISN 0011          A(1) = Y(1)
ISN 0012          GO TO 240

C          40 C(1) = X(2) - X(1)
ISN 0013          N1 = N - 1
ISN 0014          N2 = N - 2
ISN 0015          DO 50 I = 2, N1
C          H(I) = X(I+1) - X(I)
C          LAMBDA(I) = H(I) / ( H(I-1) + H(I) )
C          SD(I) = 6.* ((Y(I+1)-Y(I))/H(I) - (Y(I)-Y(I-1))/H(I-1))/(H(I-1)+H(I))
C          C(I) = X(I+1) - X(I)
ISN 0017          D(I) = C(I) / ( C(I-1) + C(I) )
ISN 0018          A(I) = 6.* ((Y(I+1)-Y(I))/C(I)-(Y(I)-Y(I-1))/C(I-1))/(C(I-1)+C(I))
ISN 0019          50 CONTINUE
C          DETERMINE END CONDITIONS AND SET LEFT END CONDITION
C          S = D(N)
C          R = C(N)
C          SD(1) = A(1)
C          LAMBDA(1) = D(1)
C          D(N) = 0.0
ISN 0021          C(N) = 0.0
ISN 0022          IF ( IP - 20 ) 110, 60, 70
ISN 0023          60 D(1) = 0.0
ISN 0024          A(1) = YP1 + YP1
ISN 0025          GO TO 160
ISN 0026          70 IF ( IP - 30 ) 90, 80, 100
ISN 0027          80 D(1) = -2. * ( C(1) + C(2) ) / C(2)
ISN 0028          A(1) = 0.0
ISN 0029          C(N) = ( C(1) + C(1) ) / C(2)
ISN 0030          GO TO 160
ISN 0031          90 D(1) = 0.0
ISN 0032          A(1) = YP1 + YP1
ISN 0033          IF ( IP - 22 ) 170, 180, 190
ISN 0034          100 D(1) = -2. * ( C(1) + C(2) ) / C(2)
ISN 0035          A(1) = 0.0
ISN 0036          C(N) = ( C(1) + C(1) ) / C(2)
ISN 0037          IF ( IP - 32 ) 170, 180, 190
ISN 0038          110 IF ( IP - 10 ) 140, 120, 130
ISN 0039          120 D(1) = 1.0
ISN 0040          A(1) = ( ( Y(2)-Y(1) ) / C(1) - YP1 ) * ( 6. / C(1) )
ISN 0041

```

170

```

ISN 0042      GO TO 160                      00055
ISN 0043      130 D(1) = 1.0                  00056
ISN 0044      A(1) = ( (Y(2)-Y(1))/C(1) - YP1 ) * (6. / C(1) ) 00057
ISN 0045      IF ( IP - 12 ) 170, 180, 190  00058
ISN 0046      140 D(1) = -YP1 - YP1         00059
ISN 0047      A(1) = 0.0                    00060
ISN 0048      IF ( IP - 2 ) 150, 180, 190   00061
ISN 0049      150 IF ( IP ) 300, 160, 170   00062
C                                                    00063
C PERIODIC SPLINE                                00064
C                                                    00065
C LAMBDA(N) = H(1) / ( H(1) + H(N-1) )        00066
C                                                    00067
ISN 0050      300 D(N) = C(1) / ( C(N1) + C(1) ) 00068
C                                                    00069
C SD(N) = 6. * (((Y(2)-Y(N))/H(1) - (Y(N)-Y(N-1))/H(N-1))/(H(N-1)+H(1))) 00070
C                                                    00071
ISN 0051      A(N) = 6. * (((Y(2)-Y(N))/C(1) - (Y(N)-Y(N1))/C(N1)) / (C(N1)+C(1))) 00072
C                                                    00073
C                                                    00074
ISN 0052      B(2) = ( D(2) - 1. ) / 2       00075
ISN 0053      A(2) = A(2) / 2                00076
ISN 0054      D(2) = -D(2) / 2              00077
C                                                    00078
C S(I) = S(I-1)*(LAMBDA(I)-1.) / (2.*Q(I-1)*(1.-LAMBDA(I))) 00079
C U(I) = (SD(I)-U(I-1)*(1.-LAMBDA(I))) / (2.*Q(I-1)*(1.-LAMBDA(I))) 00080
C Q(I) = -LAMBDA(I) / (2.*Q(I-1)*(1.-LAMBDA(I))) 00081
C                                                    00082
ISN 0055      DO 310 I = 3, N1                00083
ISN 0056      B(I) = (B(I-1)*(D(I)-1.)) / (2.+D(I-1)*(1.-D(I))) 00084
ISN 0057      A(I) = (A(I)-A(I-1)*(1.-D(I))) / (2.+D(I-1)*(1.-D(I))) 00085
ISN 0058      D(I) = -D(I) / (2.+D(I-1)*(1.-D(I))) 00086
ISN 0059      310 CONTINUE                    00087
C                                                    00088
C T(N-1) = Q(N-1) + S(N-1)                   00089
C V(N-1) = U(N-1)                             00090
C T(K) = Q(K) * T(K+1) + S(K)                00091
C V(K) = Q(K) * V(K+1) + U(K)                00092
C                                                    00093
ISN 0060      B(N1) = D(N1) + B(N1)           00094
ISN 0061      D(N1) = A(N1)                  00095
ISN 0062      DO 320 I = 2, N2                00096
ISN 0063      K = N - I                       00097
ISN 0064      B(K) = D(K) * B(K+1) + B(K)     00098
ISN 0065      D(K) = D(K) * D(K+1) + A(K)     00099
ISN 0066      320 CONTINUE                    00100
C                                                    00101
C M(N) = ( SD(N) - V(N-1) + LAMBDA(N) * ( V(N-1) - V(2) ) ) / ( 2. + 00102
C T(N-1) - LAMBDA(N) * ( T(N-1) - T(2) ) )    00103
C M(1) = M(N)                                  00104
C                                                    00105
ISN 0067      A(N) = (A(N)-D(N1)+D(N)*(D(N1)-D(2)))/(2.+B(N1)-D(N)*(B(N1)-B(2))) 00105
ISN 0068      A(1) = A(N)                     00107
ISN 0069      DO 330 I = 1, N2                00108
ISN 0070      K = N - I                       00109
C                                                    00110
C M(K) = T(K) * M(N) + V(K)                  00111
C                                                    00112
ISN 0071      A(K) = B(K) * A(N) + D(K)       00113

```

```

ISN 0072      330 CONTINUE                                00114
ISN 0073      GO TO 225                                  00115
C                                                     00116
C SET RIGHT END CONDITION                               00117
C TAU = B(N)                                           00118
C SD(N) = A(N)                                         00119
C                                                     00120
ISN 0074      160 B(N) = -YFN - YPN                      00121
ISN 0075      A(N) = 0.0                                00122
ISN 0076      GO TO 200                                  00123
ISN 0077      170 B(N) = 1.0                            00124
ISN 0078      A(N) = ( YFN - (Y(N)-Y(N1))/C(N1) ) * ( 6. / C(N1) ) 00125
ISN 0079      GO TO 200                                  00126
ISN 0080      180 B(N) = 0.0                            00127
ISN 0081      A(N) = YFN + YPN                          00128
ISN 0082      GO TO 200                                  00129
ISN 0083      190 B(N) = -2. * ( ( C(N1) + C(N-2) ) / C (N-2) ) 00130
ISN 0084      A(N) = 0.0                                00131
ISN 0085      D(N) = ( C(N1) + C(N1) ) / C(N-2)        00132
C                                                     00133
C G(1) = (LAMBDA(1)*SD(2)-2.*SD(1)) / (LAMBDA(1)*(1.-LAMBDA(2))-4.) 00134
C W(1) = (LAMBDA(1)*LAMBDA(2)-2.*R) / (LAMBDA(1)*(1.-LAMBDA(2))-4.) 00135
C G(2) = (SD(2)-G(1)*(1.-LAMBDA(2))) / 2.             00136
C W(2) = (LAMBDA(2)-W(1)*(1.-LAMBDA(2))) / 2.         00137
C                                                     00138
ISN 0086      200 A(1) = (D(1)*A(2)-A(1)-A(1)) / (D(1)*(1.-D(2))-4.) 00139
ISN 0087      B(1) = (D(1)*D(2)-C(N)-C(N)) / (D(1)*(1.-D(2))-4.) 00140
ISN 0088      A(2) = (A(2)-A(1)*(1.-D(2))) / 2.       00141
ISN 0089      B(2) = (D(2)-B(1)*(1.-D(2))) / 2.       00142
C                                                     00143
C G(I) = (SD(I)-G(I-1)*(1.-LAMBDA(I))) / (2-W(I-1)*(1.-LAMBDA(I))) 00144
C W(I) = LAMBDA(I) / (2-W(I-1)*(U-LAMBDA(I)))          00145
C                                                     00146
ISN 0090      DO 210 I = 3, N1                          00147
ISN 0091      A(I) = (A(I)-A(I-1)*(1.-D(I))) / (2.-B(I-1)*(1.-D(I))) 00148
ISN 0092      B(I) = D(I) / (2.-B(I-1)*(1.-D(I)))      00149
ISN 0093      210 CONTINUE                              00150
C                                                     00151
C M(N) = ( SD(N)-S*(G(N-2)-W(N-2)*G(N-1))-TAU*G(N-1) ) / ( 2-(TAU-S* 00152
C W(N-2))*W(N-1) )                                     00153
C M(K) = G(K) - W(K) * M(K+1)                          00154
C                                                     00155
ISN 0094      A(N) = ( A(N) - D(N) * ( A(N2) - B(N2) * A(N1) ) - B(N) * A(N1) ) 00156
ISN 0095      1 / ( 2. - ( B(N) - D(N) * B(N2) ) * B(N1) ) 00157
ISN 0096      DO 220 I = 1, N2                          00158
ISN 0097      K = N - I                                  00159
ISN 0098      A(K) = A(K) - B(K) * A(K+1)              00160
ISN 0099      220 CONTINUE                              00161
C                                                     00162
C M(1) = G(1) - W(1) * M(3)                             00163
C                                                     00164
ISN 0099      A(1) = A(1) - B(1)*A(3)                   00165
C                                                     00166
C COMPUTE COEFFICIENTS                                  00167
C                                                     00168
ISN 0100      225 DO 230 I = 1, N1                      00169
C                                                     00170
C D(I) = ( M(I+1) - M(I) ) / ( 6. * H(I) )             00171
C B(I) = ( Y(I+1) - Y(I) ) / H(I) - H(I) * ( ( M(I+1) + 2 * M(I) ) / 6. 00172

```

```

C C(I) = M(I) / 2      00173
C A(I) = Y(I)         00174
C                     00175
ISN 0101              D(I) = ( A(I+1) - A(I) ) / ( 6. * C(I) ) 00176
ISN 0102              B(I) = ( Y(I+1)-Y(I) ) / C(I) - ( A(I+1)+A(I)+A(I) ) *(C(I) / 6. )00177
ISN 0103              C(I) = A(I) / 2                          00178
ISN 0104              A(I) = Y(I)                              00179
ISN 0105              230 CONTINUE                             00180
ISN 0106              240 RETURN                               00181
ISN 0107              END                                     00182

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT COSTHT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 106, PROGRAM SIZE = 2398, SUBPROGRAM NAME =BMFIT2
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

2984K BYTES OF CORE NOT USED

STATISTICS NO DIAGNOSTICS THIS STEP

172

F64-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED MAP,LIST,LET,,,SIZE=(256K,36K)
 VARIABLE OPTIONS USED - SIZE=(262144,36864)

IEW0132 CUBIC
 IEW0132 PEMFIT
 IEW0132 FOD

MODULE MAP

CONTROL SECTION

ENTRY

NAME	ORIGIN	LENGTH
STAEBL	00	22A5C
BBIP	22A60	1864
BIRDF	242C8	42C
BCOORD	246F8	2CE
BEAM	249C8	27D6
CORD2R	271A0	A1C
ESTIFF	276C0	50A
EXNECK	280D0	12F2
FRPH	293C8	2C2
HOLZER	29690	AE8
INPUT	2A178	3A8
KDGG	2A520	9F0
LAMINA	2AF10	9FA
LAMIN1	2B910	39A
LOAD2	2BCB0	77E
MATADD	2C430	1B4
MATHPY	2C5E8	232
MATPRN	2C820	21E
MINV	2CA40	5DE
MODNAS	2D020	40A
MODSTR	2D430	4C2
PARTH	2D8F8	37A
PP1	2DC78	216
RBE2	2DE90	5D8
SECTN	2E468	4390
SHAPE	327F8	340
SMULT	32B38	100E
STABEL	33B48	12E58
STATIC	469A0	1D2C
ZSTRES	486D0	1440
THICK	49B10	25FE
TSAINU	4C110	458
WARP	4C568	A16
HOLLOW	4CF80	9B0
EHFIT2	4D930	95E
IHOLASCN*	4E290	267
IHOLATN2*	4E4F8	278
IHOLCOS *	4E770	280

NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
IH\$DARCS	4E290	DARCOS	4E290	IH\$DARSN	4E2A8	DARSIN	4E2A8
DATAN2	4E4F8	IH\$DATA2	4E4F8	DATAN	4E510	IH\$DATAN	4E510
DCOS	4E770	IH\$DCOS	4E770				

NAME	ORIGIN	LENGTH
BLKA	652C0	348
BLK9	65608	348
INPUTT	65950	7D00
SC	6D650	3F0
ANAL33	6DA40	58
FAIL	6DA98	790
SMAXX	6E228	140
MODAL1	6E368	3870
BLK15	71B08	FA0
COORD1	72B78	11940
BLK3	844B8	30
BC	844E8	128
BSTIFF	84610	900
ESTIF	84F10	180
LAYER	85090	7D0
REORDR	85850	900
QIJ	86160	1F8
STIFF	86358	7AA0
STRS	8DDF8	96330
PLY1	124128	7348
ZCOEF	12B470	1F8
HSTIFF	12E668	6300
STATC	131968	61E0
CALCA	137B48	50C8
RBE	13CC10	900
MODAL	13D510	12C0
HARPP	13E7D0	31E0
BLKAC	1419B0	16800
DSTIFF	1581B0	B400
TRANS	1635B0	5E8
HOLE	163B98	4
HIEGHT	163BA0	38

NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
------	----------	------	----------	------	----------	------	----------

ENTRY ADDRESS 00

TOTAL LENGTH 163BD8
 ****U4982024 NOW REPLACED IN DATA SET
 AUTHORIZATION CODE IS 0.

DIAGNOSTIC MESSAGE DIRECTORY

IEH0132 ERROR - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE.

PANVALET
THE PROGRAM MANAGEMENT AND SECURITY SYSTEM

PROGRAMS AND ALL SUPPORTING MATERIALS COPYRIGHT 1975 BY PANSOPHC SYSTEMS, INCORPORATED

++WRITE WORK,U498SFOD
***** ABOVE ACTION SATISFACTORILY COMPLETED *****

++WRITE WORK,U498SANLIZ
***** ABOVE ACTION SATISFACTORILY COMPLETED *****

++WRITE WORK,U498SRDATA
***** ABOVE ACTION SATISFACTORILY COMPLETED *****

++WRITE WORK,U498SCALCT
***** ABOVE ACTION SATISFACTORILY COMPLETED *****

++WRITE WORK,U498SIT983
***** ABOVE ACTION SATISFACTORILY COMPLETED *****

++WRITE WORK,U498SND983
***** ABOVE ACTION SATISFACTORILY COMPLETED *****

++WRITE WORK,U498SRD983
***** ABOVE ACTION SATISFACTORILY COMPLETED *****

++WRITE WORK,U498SRD137
***** ABOVE ACTION SATISFACTORILY COMPLETED *****

++WRITE WORK,U498SRSFUN
***** ABOVE ACTION SATISFACTORILY COMPLETED *****

++WRITE WORK,U498STMAX
***** ABOVE ACTION SATISFACTORILY COMPLETED *****

++WRITE WORK,U498SHT983
***** ABOVE ACTION SATISFACTORILY COMPLETED *****

++WRITE WORK,U498SWT137
***** ABOVE ACTION SATISFACTORILY COMPLETED *****

++WRITE WORK,U498SOBJTV
***** ABOVE ACTION SATISFACTORILY COMPLETED *****

++WRITE WORK,PREFIX=U477C
***** ABOVE ACTION SATISFACTORILY COMPLETED *****

++WRITE WORK,U477SIMCON
***** ABOVE ACTION SATISFACTORILY COMPLETED *****

++WRITE WORK,U477MAIN
***** ABOVE ACTION SATISFACTORILY COMPLETED *****

FORTRAN H EXTENDED (ENHANCED)

177

SOURCE STATEMENTS =	105,	PROGRAM SIZE =	4294,	SUBPROGRAM NAME =	FOD
SOURCE STATEMENTS =	187,	PROGRAM SIZE =	3568,	SUBPROGRAM NAME =	ANALIZ
SOURCE STATEMENTS =	70,	PROGRAM SIZE =	1244,	SUBPROGRAM NAME =	RDDATA
SOURCE STATEMENTS =	42,	PROGRAM SIZE =	1032,	SUBPROGRAM NAME =	CALCTH
SOURCE STATEMENTS =	77,	PROGRAM SIZE =	1402,	SUBPROGRAM NAME =	ITT983
SOURCE STATEMENTS =	19,	PROGRAM SIZE =	464,	SUBPROGRAM NAME =	NDT983
SOURCE STATEMENTS =	18,	PROGRAM SIZE =	558,	SUBPROGRAM NAME =	RDT983
SOURCE STATEMENTS =	110,	PROGRAM SIZE =	3472,	SUBPROGRAM NAME =	RDW137
SOURCE STATEMENTS =	33,	PROGRAM SIZE =	1048,	SUBPROGRAM NAME =	RSFUNC
SOURCE STATEMENTS =	58,	PROGRAM SIZE =	1098,	SUBPROGRAM NAME =	TMAX
SOURCE STATEMENTS =	49,	PROGRAM SIZE =	1206,	SUBPROGRAM NAME =	ITT983
SOURCE STATEMENTS =	114,	PROGRAM SIZE =	3002,	SUBPROGRAM NAME =	WTW137
SOURCE STATEMENTS =	58,	PROGRAM SIZE =	1994,	SUBPROGRAM NAME =	OBJTV
SOURCE STATEMENTS =	87,	PROGRAM SIZE =	1730,	SUBPROGRAM NAME =	CNMN01
SOURCE STATEMENTS =	40,	PROGRAM SIZE =	654,	SUBPROGRAM NAME =	CNMN02
SOURCE STATEMENTS =	249,	PROGRAM SIZE =	3520,	SUBPROGRAM NAME =	CNMN03
SOURCE STATEMENTS =	105,	PROGRAM SIZE =	1754,	SUBPROGRAM NAME =	CNMN04
SOURCE STATEMENTS =	122,	PROGRAM SIZE =	2548,	SUBPROGRAM NAME =	CNMN05
SOURCE STATEMENTS =	493,	PROGRAM SIZE =	6746,	SUBPROGRAM NAME =	CNMN06
SOURCE STATEMENTS =	55,	PROGRAM SIZE =	936,	SUBPROGRAM NAME =	CNMN07
SOURCE STATEMENTS =	84,	PROGRAM SIZE =	1390,	SUBPROGRAM NAME =	CNMN08
SOURCE STATEMENTS =	830,	PROGRAM SIZE =	15164,	SUBPROGRAM NAME =	CONMIN
SOURCE STATEMENTS =	706,	PROGRAM SIZE =	15790,	SUBPROGRAM NAME =	COPE01
SOURCE STATEMENTS =	41,	PROGRAM SIZE =	1352,	SUBPROGRAM NAME =	COPE02
SOURCE STATEMENTS =	68,	PROGRAM SIZE =	1246,	SUBPROGRAM NAME =	COPE03
SOURCE STATEMENTS =	230,	PROGRAM SIZE =	3454,	SUBPROGRAM NAME =	COPE04
SOURCE STATEMENTS =	71,	PROGRAM SIZE =	2210,	SUBPROGRAM NAME =	COPE05
SOURCE STATEMENTS =	48,	PROGRAM SIZE =	1118,	SUBPROGRAM NAME =	COPE06
SOURCE STATEMENTS =	52,	PROGRAM SIZE =	1470,	SUBPROGRAM NAME =	COPE07

SOURCE STATEMENTS =	63,	PROGRAM SIZE =	846,	SUBPROGRAM NAME =	COPE08
SOURCE STATEMENTS =	544,	PROGRAM SIZE =	9868,	SUBPROGRAM NAME =	COPE09
SOURCE STATEMENTS =	69,	PROGRAM SIZE =	2038,	SUBPROGRAM NAME =	COPE10
SOURCE STATEMENTS =	57,	PROGRAM SIZE =	1134,	SUBPROGRAM NAME =	COPE11
SOURCE STATEMENTS =	125,	PROGRAM SIZE =	2844,	SUBPROGRAM NAME =	COPE12
SOURCE STATEMENTS =	23,	PROGRAM SIZE =	566,	SUBPROGRAM NAME =	COPE13
SOURCE STATEMENTS =	95,	PROGRAM SIZE =	2632,	SUBPROGRAM NAME =	COPE14
SOURCE STATEMENTS =	63,	PROGRAM SIZE =	1728,	SUBPROGRAM NAME =	COPE15
SOURCE STATEMENTS =	29,	PROGRAM SIZE =	878,	SUBPROGRAM NAME =	COPE16
SOURCE STATEMENTS =	26,	PROGRAM SIZE =	680,	SUBPROGRAM NAME =	COPE17
SOURCE STATEMENTS =	49,	PROGRAM SIZE =	1408,	SUBPROGRAM NAME =	COPE18
SOURCE STATEMENTS =	157,	PROGRAM SIZE =	3780,	SUBPROGRAM NAME =	SIMCON
SOURCE STATEMENTS =	248,	PROGRAM SIZE =	3980,	SUBPROGRAM NAME =	MAIN

REQUESTED OPTIONS: SOURCE,NOMAP,NOXREF,NOLIST,NODECK,OPT(3),AUTODBL(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```

C      DATA SET U498SFOD   AT LEVEL 006 AS OF 03/01/82
C      DATA SET U498SFOD   AT LEVEL 004 AS OF 02/08/82
C      DATA SET U498SFOD   AT LEVEL 002 AS OF 11/16/81
ISN 0002      SUBROUTINE FOD(NR,NC,RPM1,ALPHA1)
C
C      LSBIP - LOCAL STRESS BIRD INGESTION PARAMETER
C
ISN 0003      IMPLICIT REAL*8(A-H,O-Z)
ISN 0004      REAL * 4 VALT , BRSU , NB
ISN 0005      COMMON /INPUTT/ X(1000),Y(1000),Z(1000),T(1000)
ISN 0006      COMMON /BLK 3/ FN,BLADES,BETA,THR,THT,MST1,NBB
ISN 0007      COMMON /BLKAA/ ALPMIN(21),XSC(21),YSC(21),XXX(21),YYY(21)
ISN 0008      COMMON /BLK 8/ SMB(21),ITLLE(18),VARI(235),TMAX(21),HALPHA(21)
ISN 0009      COMMON /LAYER/ TH(7,25),EMASS(25),BSPAN(25),BNIDTH(25)
ISN 0010      COMMON /BIRD/  BLSMAX(21)
ISN 0011      COMMON /PLY1/ PLY(21,25,7),THETA(7),RH(7)
ISN 0012      COMMON /STRS/ RBE(12,12,25,21),FOUT(12,21),SL(25,21),SH(25,21)
C
ISN 0013      DIMENSION TB(25),AREA(25),AAREA(25),TCRS(25),TB1(25)
ISN 0014      DIMENSION TB2(25),BIP(25),XX(25),ALPHA1(1)
C
ISN 0015      WRITE(6,270)
ISN 0016      270 FORMAT(/,10X,'** LSBIP ANALYSIS **',/)
C
C      CALCULATE LSBIP AT EACH STATION, LOCATING A MAXIMUM FOR EACH
C      STATION AND STORING IT IN BLSMAX(21)
C
ISN 0017      DO 500 KK = 1,NR
ISN 0018      NSTA = KK
C
C      XF = DISTANCE OF LOAD APPLICATION ALONG CHORD
C
ISN 0019      VBIRD = 180.0
ISN 0020      NB = BLADES
ISN 0021      PI = 3.14159265
ISN 0022      ALPHA = ALPHA1(NSTA)
ISN 0023      RPM = RPM1 * 30. / PI
ISN 0024      V1 = .42 * 12. * VBIRD * DCOS( ALPHA )
ISN 0025      NODE = (NSTA-1) * NC + 1
ISN 0026      V2 = PI * X(NODE) * RPM / 30. * DSIN( ALPHA )
ISN 0027      VBLADE = X(NODE)/12. * RPM * PI / 30.
C
ISN 0028      WRITE(6,400) NSTA,NC,RPM,ALPHA,NB,VBIRD,XSC(NSTA)
ISN 0029      400 FORMAT(/,5X,'FOD INPUT',5X,'STATION = ',I5,/,19X,'NC = ',I5,
1      /,19X,'RPM(REV/MIN) = ',F7.2,/,19X,'ALPHA CHORD(RADIANS) = ',
2      ,E12.5,/,19X,'NUMBER OF BLADES = ',E12.5,/,19X,
3      'BIRD VELOCITY (FT/SEC) = ',E12.5,/,19X,
4      'SHEAR CENTER (FROM TE) = ',E12.5,/)
C
C      ZERO MATRICES
C
ISN 0030      DO 260 I = 1,25
ISN 0031      TB1(I) = 0.
ISN 0032      TB2(I) = 0.
ISN 0033      TB(I) = 0.

```

```

+VERSION 1.3.0 (01 MAY 80)      FOD      SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)      DATE 82.141/10.52.17      PAGE 2
  ISN 0034      AREA(I) = 0.      00055
  ISN 0035      AAREA(I) = 0.      00056
  ISN 0036      TORS(I) = 0.      00057
  ISN 0037      XX(I) = 0.      00058
  ISN 0038      260 BIP(I) = 0.      00059
  C      00060
  C      DISTANCE TO CENTER OF LOAD APPLICATION FROM LE      00061
  C      00062
  ISN 0039      XF = 2. * PI * X(NODE) / NB / (V2/V1-1.) / DCOS( ALPHA )      00063
  C      00064
  C      P = IMPACT LOAD      00065
  C      00066
  ISN 0040      P = X(NODE)**2 / NB * RPM * DSIN( ALPHA ) * VBIRD * .0000001      00067
  C      00068
  ISN 0041      WRITE(6,401) XF,P      00069
  ISN 0042      401 FORMAT(19X,'XF DISTANCE = ',E12.5,/,19X,      00070
  1          'P LOAD = ',E12.5,/)      00071
  C      00072
  C      METAL ONLY THICKNESS      00073
  C      00074
  ISN 0043      II = KK      00075**6
  ISN 0044      IF(KK .EQ. NR) II = KK - 1      00076**6
  ISN 0046      DO 110 I = 1,NC      00077**6
  ISN 0047      110 BWIDTH(I) = SH(I,II)      00078**6
  ISN 0048      DO 100 I = 1,NC      00079
  ISN 0049      TB1(I) = PLY(II,I,1) + PLY(II,I,2)      00080**6
  ISN 0050      TB2(I) = PLY(II,I,6) + PLY(II,I,7)      00081**6
  ISN 0051      100 TB(I) = TB1(I) + TB2(I)      00082
  C      CALL MATPRN(TB,25,1,' TB ')      00083
  C      00084
  C      AREA CALCULATION - METAL AREA ONLY      00085
  C      00086
  ISN 0052      AREA(1) = BWIDTH(1) * TB(1)      00087
  ISN 0053      DO 101 I = 2,NC      00088
  ISN 0054      101 AREA(I) = AREA(I-1) + BWIDTH(I) * TB(I)      00089
  C      CALL MATPRN(AREA,25,1,'AREA')      00090
  C      00091
  C      EVALUATING THE DOUBLE AREA INTEGRAL , AAREA      00092
  C      00093
  ISN 0055      AAREA(1) = AREA(1) * BWIDTH(1)      00094
  ISN 0056      DO 102 I = 2,NC      00095
  ISN 0057      102 AAREA(I) = AAREA(I-1) + AREA(I) * BWIDTH(I)      00096
  C      CALL MATPRN(AAREA,25,1,'AARE')      00097
  C      00098
  C      TORSION TERMS - THIN SECTION ASSUMPTION      00099
  C      00100
  ISN 0058      NODE = (NSTA-1) * NC + 1      00101
  ISN 0059      NRNC = NR * NC      00102
  ISN 0060      NODE1 = NODE + NC      00103
  ISN 0061      IF (NODE1 .GE. NRNC) NODE1 = NODE - NC      00104
  ISN 0063      DO 200 I = 1,NC      00105
  ISN 0064      TAVE = ( T(NODE) + T(NODE1) ) / 2.      00106
  ISN 0065      TAVE = 0.      00107**6
  ISN 0066      DO 201 J = 1,7      00108**6
  ISN 0067      201 TAVE = TAVE + PLY(II,I,J)      00109**6
  C      00110**6
  ISN 0068      NODE1 = NODE1 + 1      00111
  ISN 0069      TOR1 = 1./3. * BWIDTH(I) * TB(I)**3      00112
  ISN 0070      THOL = TAVE - TB(I)      00113**6

```

ISN	Code	Statement	Address
+VERSION 1.3.0 (01 MAY 80)		FOD SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)	DATE 82.141/10.52.17
			PAGE 3
ISN 0071		TOR2 = 1./3. * BWIDTH(I) * (TAVE**3 - THOL**3)	00114**6
ISN 0072		TORS(I) = TOR1	00115
ISN 0073		IF((TB1(I)+TB2(I)+.0001) .LT. TAVE) TORS(I) = TOR2	00116
ISN 0075		IF(I .EQ. 1) GO TO 200	00117
ISN 0077		TORS(I) = TORS(I-1) + TORS(I)	00118
ISN 0078	200	NODE = NODE + 1	00119
	C	CALL MATPRN(TORS,25,1,'TORS')	00120
	C		00121
	C	DISTANCE FROM L.E. , XX	00122
	C		00123
ISN 0079		XX(1) = EWIDTH(1)/2.	00124
ISN 0080		DO 210 I = 2,NC	00125
ISN 0081	210	XX(I) = XX(I-1) + BWIDTH(I-1)/2. + BWIDTH(I)/2.	00126
	C	CALL MATPRN(XX,25,1,' XX '	00127
	C		00128**6
	C	XBAR = DISTANCE FROM LOAD CENTER TO C.G.	00129**6
	C		00130**6
ISN 0082		TDX = 0.	00131**6
ISN 0083		DO 350 I = 1,NC	00132**6
ISN 0084	350	TDX = TDX + XX(I) * TB(I)	00133**6
ISN 0085		XBAR = TDX / AREA(NC) - XF/2.	00134**6
	C		00135**6
ISN 0086		WRITE(6,351) XBAR	00136**6
ISN 0087	351	FORMAT(5X,'DISTANCE TO LOAD FROM CG. , XBAR =',E12.5)	00137**6
	C		00138**6
	C		00139
	C	LSBIP CALCULATION - BIP	00140
	C		00141
ISN 0088		NC2 = NC	00142**6
ISN 0089		DO 300 I = 1,NC2	00143
ISN 0090		IF(XX(I) .LT. XF) GO TO 333	00144
	C		00145
	C	XX .GT. XF	00146
	C		00147
ISN 0092		BIP(I) = P/TB(I)**2 * (XX(I) - XF/2. - AAREA(I)/AREA(NC) -	00148
	1	XBAR / TORS(NC) * TORS(I))	00149
	C		00150
ISN 0093		GO TO 300	00151
	C		00152
	C	XX .LE. XF	00153
	C		00154
ISN 0094	333	BIP(I) = P/TB(I)**2 * (XX(I)**2/2./XF - AAREA(I)/AREA(NC) -	00155
	1	XBAR / TORS(NC) * TORS(I))	00156
	C		00157
ISN 0095	300	CONTINUE	00158
	C	CALL MATPRN(BIP,25,1,' BIP')	00159
	C		00160
	C	SEARCH FOR MAXIMUM BIP FOR SECTION KK	00161
	C		00162
ISN 0096		BLSMAX(KK) = 0.	00163
ISN 0097		DO 405 I = 1,NC2	00164
ISN 0098		IF(BIP(I) .GT. BLSMAX(KK)) BLSMAX(KK) = BIP(I)	00165
ISN 0100	405	CONTINUE	00166
	C		00167
ISN 0101	500	CONTINUE	00168
	C		00169
ISN 0102		DO 505 I = 1,NR	00170
ISN 0103	505	WRITE(6,506) I,BLSMAX(I)	00171
ISN 0104	506	FORMAT(5X,'MAX BIP , STATION = ',I5,5X,E12.5)	00172

ISN 0105 RETURN
ISN 0106 END

00173
00174
00175

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
STATISTICS SOURCE STATEMENTS = 105, PROGRAM SIZE = 4294, SUBPROGRAM NAME = FOD
STATISTICS NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

2988K BYTES OF CORE NOT USED

182

REQUESTED OPTIONS: SOURCE,NOMAP,NOXREF,NOLIST,NODECK,OPT(3),AUTODBL(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTHT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```

C      DATA SET U498SANLIZ AT LEVEL 019 AS OF 03/16/82
C      DATA SET U498SANLIZ AT LEVEL 018 AS OF 03/15/82      00001
C      DATA SET U498SANLIZ AT LEVEL 017 AS OF 03/15/82      00002
C      DATA SET U498SANLIZ AT LEVEL 016 AS OF 03/12/82      00003
ISN 0002      SUBROUTINE ANALIZ (ICALC)                        00004
ISN 0003      COMMON /BKH137/ BLADWT , FREQ(100)                00005
ISN 0004      COMMON /UIOS/ ISI , I6I                          00006
ISN 0005      COMMON /ANAL03/ RPM , FHS , DFN , TOLL ,ROOT,DRPM,RPMMN,RPMMX  00007
ISN 0006      COMMON /ANAL04/ NSTA , MST1 , MST2 , NOACR , ILT , LONG , ISTE ,  00008
              1 DENB , EB , POIB , BR , SERIES , BRS , BTS , IOPP          00009
ISN 0007      COMMON /ANAL05/ R(21) , BOBR(21),TOB(21),THSTH(21),ALPHA(21),  00010
              1 SOB(21) , RLE(21) , RTE(21) , ADWT(21) , O(21)            00011
ISN 0008      COMMON /ANAL17/ XPS(3,53,21) , TMX(21) , TCC(21) , BMX(21)    00012
              1 , BCC(21)                                                  00013
ISN 0009      COMMON /ANAL30/ NTIS , NRF , IST(21) , VALT(21) , BRSV        00014
ISN 0010      COMMON /ANAL33/ DLED , DTED , DROOTD , DTIPD , TTID , TLTD , NCD  00015
              1 , NCK , TISD , TICD , PCBAD , PCGED                    00016
ISN 0011      COMMON /ANAL54/ HD(5) , DLSV(5)                    00017
ISN 0012      COMMON /PLY1/ PLY(21,25,7) , THETAD(7) , RHO(7)          00018
ISN 0013      COMMON /GLOBCH/ OBJF , FN(5) , DLAR(5) , THKVAL(21) , RF(5,4)  00019
              1 , BRCC , FODLSB(21) , DLE , DTE , DROOT , DTIP , TTI , TLT  00020
              2 , OBJFUN , SMAX1S(8) , SMAX2S(8) , SMAX3S(8)            00021
              3 , SMAXLS(8) , SMAXTS(8) , THETA(7) , HLRATIO , ECRATIO , TOVB(21) 00022
              4 , FODMAX , TIS , TIC , PCBA , VAL                        00023
ISN 0014      COMMON /BIRD/ BLSMAX(21)                            00024
ISN 0015      COMMON /SMAXX/ SMAX1(8) , SMAX2(8) , SMAX3(8) , SMAXL(8)      00025
              1 , SMAXT(8)                                              00026
ISN 0016      DOUBLE PRECISION BLADWT , FREQ , BLSMAX , DLED , DTED , DROOTD  00027
              1 , DTIPD , TTID , TLTD , SMAX1 , SMAX2 , SMAX3          00028
              2 , SMAXL , SMAXT , PLY , THETAD , RHO , TISD , TICD , PCBAD  00029
              3 , PCGED                                                00030
ISN 0017      DLED = DLE                                           00031
ISN 0018      DTED = DTE                                           00032
ISN 0019      DROOTD = DROOT                                        00033
ISN 0020      DTIPD = DTIP                                         00034
ISN 0021      TTID = TTI                                           00035
ISN 0022      TLTD = TLT                                           00036
ISN 0023      TISD = TIS                                           00037
ISN 0024      TICD = TIC                                           00038
ISN 0025      PCBAD = PCBA                                          00039
ISN 0026      PCGED = 1.0 - PCBAD                                   00040
ISN 0027      PCGE = PCGED                                          00041
ISN 0028      DO 40 I = 1,3                                         00042
ISN 0029      THETA(8-I) = THETA(I)                                00043
ISN 0030      40 CONTINUE                                          00044
ISN 0031      DO 50 I = 1,7                                         00045
ISN 0032      THETAD(I) = THETA(I)                                 00046
ISN 0033      50 CONTINUE                                          00047
ISN 0034      IF (ICALC .GT. 1) GO TO 200                          00048
ISN 0036      CALL MYTIME (ITIME)                                  00049
ISN 0037      ITIMEC = 2                                           00050
ISN 0038      WRITE (26,3000) ITIMEC , ITIME                       00051
ISN 0039      3000 FORMAT (100X,2I10)                             00052
ISN 0040      CALL RDW137 (ICALC)                                  00053
ISN 0041      CALL RDDATA                                          00054

```

```

ISN 0042      IF (NRF .NE. 0) CALL RDT983 (ICALC)          00055
ISN 0044      CALL TMAX (ICALC)                          00056
ISN 0045      CALL WTW137 (ICALC)                       00057
ISN 0046      DO 150 I = 1,NTIS                          00058
ISN 0047      THKVAL(I) = VALT(I)                       00059
ISN 0048      150 CONTINUE                               00060
ISN 0049      BRCC = BRVS                                00061
ISN 0050      CALL MYTIME (ITIME)                       00062
ISN 0051      ITIMEC = 3                                00063
ISN 0052      WRITE (26,3000) ITIMEC , ITIME            00064
ISN 0053      RETURN                                    00065
ISN 0054      200 IF (ICALC .GT. 2) GO TO 300           00066
ISN 0056      CALL MYTIME (ITIME)                       00067
ISN 0057      ITIMEC = 4                                00068
ISN 0058      WRITE (26,3000) ITIMEC , ITIME            00069
ISN 0059      NBRS = BRS + .05                          00070
ISN 0060      IF (NCD .EQ. 2) GO TO 210                 00071
ISN 0062      HLRTIO = (DTIP + DROOT) / (R(NSTA) - R(1)) 00072
ISN 0063      ECRTIO = (DLE + DTE) / BRCC               00073
ISN 0064      WRITE (I6I,1000)                          00074
ISN 0065      WRITE (I6I,238) DLE , DTE , DROOT , DTIP , TTI , TLT
ISN 0066      1 , HLRTIO , ECRTIO                       00076
ISN 0066      GO TO 220                                  00077
ISN 0067      210 WRITE (I6I,1030)                       00078
ISN 0068      VAL = (2.0 * TIS + TIC) / (THKVAL(1))     00079
ISN 0069      WRITE (I6I,238) TIS , TIC , PCBA , PCGE , VAL
ISN 0070      220 CONTINUE                               00081
ISN 0071      WRITE (I6I,1010)                           00082
ISN 0072      WRITE (I6I,238) (THETA(I) , I=1,7)       00083
ISN 0073      CALL RDW137 (ICALC)                       00084
ISN 0074      CALL CALCTH                               00085
ISN 0075      CALL TMAX (ICALC)                         00086
ISN 0076      CALL WTW137 (ICALC)                       00087
ISN 0077      CALL MW137                                 00088
ISN 0078      CALL RSFUNC                               00089
ISN 0079      OBJF = BLADWT                              00090
ISN 0080      CALL OBJTV                                00091
ISN 0081      WRITE (I6I,930)                            00092
ISN 0082      WRITE (I6I,238) OBJF , OBJFUN             00093
ISN 0083      NROOT = ROOT + .01                        00094
ISN 0084      DO 230 I = 1,NROOT                        00095
ISN 0085      FN(I) = FREQ(I)                          00096
ISN 0086      230 CONTINUE                               00097
ISN 0087      DO 232 I = 1,NTIS                         00098
ISN 0088      N = IST(I)                                00099
ISN 0089      TOVB(I) = TCC(N) / BCC(N)                 00100
ISN 0090      232 CONTINUE                               00101
ISN 0091      WRITE (I6I,1020)                           00102
ISN 0092      WRITE (I6I,238) (TOVB(I) , I=1,NTIS)     00103
ISN 0093      DO 235 I = NBRS,NSTA                      00104
ISN 0094      FODLSB(I) = BLSMAX(I)                    00105
ISN 0095      235 CONTINUE                               00106
ISN 0096      FODMAX = 0.0                              00107
ISN 0097      RCHKR = R(NBRS) + 2.0 - .005             00108
ISN 0098      RCHKT = R(NSTA) - 2.0 + .005            00109
ISN 0099      DO 234 K = NBRS,NSTA                     00110
ISN 0100      IF (R(K) .LT. RCHKR) GO TO 234           00111
ISN 0102      IF (R(K) .GT. RCHKT) GO TO 234           00112
ISN 0104      BMAX = BLSMAX(K)                         00113

```

```

ISHN 0105      FODMAX = AMAX1(FODMAX,BMAX)          00114
ISHN 0106      IF (FODMAX .EQ. BMAX) RMAX = R(K)    00115
ISHN 0108      WRITE (I6I,238) FODMAX , RMAX       00116
ISHN 0109      234 CONTINUE                          00117
ISHN 0110      WRITE (I6I,970)                      00118
ISHN 0111      WRITE (I6I,238) (FODLSB(I) ,I=NBRS,NSTA) 00119
ISHN 0112      236 FORMAT (/8(3X,E12.5))           00120
ISHN 0113      238 FORMAT (8(3X,E12.5))           00121
ISHN 0114      DO 237 I = 1,8                       00122
ISHN 0115      SMAX1S(I) = SMAX1(I)                00123
ISHN 0116      SMAX2S(I) = SMAX2(I)                00124
ISHN 0117      SMAX3S(I) = SMAX3(I)                00125
ISHN 0118      SMAXLS(I) = SMAXL(I)                00126
ISHN 0119      SMAXTS(I) = SMAXT(I)                00127
ISHN 0120      237 CONTINUE                          00128
ISHN 0121      WRITE (I6I,940)                      00129
ISHN 0122      WRITE(I6I,238) (SMAX1S(I) , I=1,8)  00130
ISHN 0123      WRITE (I6I,950)                      00131
ISHN 0124      WRITE(I6I,238) (SMAX2S(I) , I=1,8)  00132
ISHN 0125      WRITE (I6I,960)                      00133
ISHN 0126      WRITE(I6I,238) (SMAX3S(I) , I=1,8)  00134
ISHN 0127      WRITE (I6I,980)                      00135
ISHN 0128      WRITE (I6I,238) (SMAXLS(I) , I=1,8)  00136
ISHN 0129      WRITE (I6I,990)                      00137
ISHN 0130      WRITE (I6I,238) (SMAXTS(I) , I=1,8)  00138
ISHN 0131      CALL MYTIME (ITIME)                  00139
ISHN 0132      ITIMEC = 5                           00140
ISHN 0133      WRITE (26,3000) ITIMEC , ITIME       00141
ISHN 0134      IF (NRF .EQ. 0) GO TO 240            00142
ISHN 0136      CALL ITT983 (ICALC)                  00143
ISHN 0137      WRITE (I6I,920)                      00144
ISHN 0138      DO 240 I = 1,NRF                     00145
ISHN 0139      WRITE (I6I,910) ND(I) , DLSV(I)      00146
ISHN 0140      DLAR(I) = DLSV(I)                   00147
ISHN 0141      240 CONTINUE                          00148
ISHN 0142      CALL MYTIME (ITIME)                  00149
ISHN 0143      ITIMEC = 6                           00150
ISHN 0144      WRITE (26,3000) ITIMEC , ITIME       00151
ISHN 0145      RETURN                                00152
ISHN 0146      300 CONTINUE                          00153
ISHN 0147      CALL MYTIME (ITIME)                  00154
ISHN 0148      ITIMEC = 7                           00155
ISHN 0149      WRITE (26,3000) ITIMEC , ITIME       00156
ISHN 0150      CALL RDW137 (ICALC)                  00157
ISHN 0151      CALL CALCTH                           00158
ISHN 0152      CALL TMAX (ICALC)                   00159
ISHN 0153      CALL WTW137 (ICALC)                 00160
ISHN 0154      ICALC = 4                            00161
C      CALL WTW137 (ICALC)                          00162
ISHN 0155      ICALC = 3                            00163
ISHN 0156      IF (NRF .EQ. 0) GO TO 350           00164
ISHN 0158      INAVE = -1                           00165
ISHN 0159      IND = 4                              00166
ISHN 0160      MODE = 1                             00167
ISHN 0161      CALL WTT983 (ICALC,MODE,IND,IWAVE)  00168
ISHN 0162      MODE = 2                             00169
ISHN 0163      CALL WTT983 (ICALC,MODE,IND,IWAVE)  00170
ISHN 0164      MODE = 3                             00171
ISHN 0165      CALL WTT983 (ICALC,MODE,IND,IWAVE)  00172

```

```

+VERSION 1.3.0 (01 MAY 80) ANALIZ SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 82.141/10.52.24 PAGE 4
  ISN 0166      350 CONTINUE                                00173
  ISN 0167      NBRS = BRS + .05                            00174
  ISN 0168      WRITE (I6I,900) (THX(I),I=NBRS,NSTA)      00175
  ISN 0169      WRITE (I6I,900) (TCC(I),I=NBRS,NSTA)      00176
  ISN 0170      900 FORMAT (//8(2X,F10.6))                 00177
  ISN 0171      910 FORMAT (5X,I5,5X,E12.5)                00178
  ISN 0172      920 FORMAT (/6X,'ND-SAVE DELAERO-SAVE')    00179
  ISN 0173      930 FORMAT (/6X,'HEIGHT OBJECT FUNCTION') 00180
  ISN 0174      940 FORMAT (/5X,'SMAX1(I) I = 1,8')       00181
  ISN 0175      950 FORMAT (/5X,'SMAX2(I) I = 1,8')       00182
  ISN 0176      960 FORMAT (/5X,'SMAX3(I) I = 1,8')       00183
  ISN 0177      970 FORMAT (/5X,'LSBIP(I) I = 1,NTIS')    00184
  ISN 0178      980 FORMAT (/5X,'SMAXLE(I) I = 1,8')       00185
  ISN 0179      990 FORMAT (/5X,'SMAXTE(I) I = 1,8')       00186
  ISN 0180      1000 FORMAT (/8X,'DLE',12X,'DTE',11X,'DROOT',10X,'DTIP',
1 11X,'TTI',11X,'TBT',12X,'HLRATIO',9X,'ECRATIO')        00188
  ISN 0181      1010 FORMAT (/5X,'THETA(I) I = 1,7')       00189
  ISN 0182      1020 FORMAT (/5X,'T/B(I) I = 1,NTIS')     00190
  ISN 0183      1030 FORMAT (/8X,'TIS',12X,'TIC',11X,'PCBA',11X,'PCGE',9X,'T RATIO') 00191
  ISN 0184      CALL MYTIME (ITIME)                        00192
  ISN 0185      ITIMEC = 8                                  00193
  ISN 0186      WRITE (26,3000) ITIMEC , ITIME             00194
  ISN 0187      RETURN                                     00195
  ISN 0188      END                                        00196

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOBBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

STATISTICS SOURCE STATEMENTS = 187, PROGRAM SIZE = 3568, SUBPROGRAM NAME =ANALIZ

STATISTICS NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

2980K BYTES OF CORE NOT USED

REQUESTED OPTIONS: SOURCE, NOMAP, NOXREF, NOLIST, NODECK, OPT(3), AUTODBL(NONE), NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTHT NOXREF NOALC NOANSF TERM IBM FLAG(I)

C	DATA SET U498SRDATA AT LEVEL 005 AS OF 03/15/82	
C	DATA SET U498SRDATA AT LEVEL 004 AS OF 02/24/82	00001
C	DATA SET U498SRDATA AT LEVEL 003 AS OF 02/23/82	00002
C	DATA SET U498SRDATA AT LEVEL 002 AS OF 01/29/82	00003
C	DATA SET U493SRDATA AT LEVEL 001 AS OF 11/12/81	00004
C	DATA SET U500RDDATA AT LEVEL 003 AS OF 07/16/81	00005
C	DATA SET U500RDDATA AT LEVEL 002 AS OF 06/18/81	00006
C	DATA SET U500RDDATA AT LEVEL 001 AS OF 06/08/81	00007
ISN 0002	SUBROUTINE RDDATA	00008
ISN 0003	COMMON /UIOS/ I5I , I6I	00009
ISN 0004	COMMON /ANAL54/ ND(5) , DLSV(5)	00010
ISN 0005	COMMON /ANAL30/ NTIS , NRF , IST(21) , VALT(21) , BRVS	00011
ISN 0006	COMMON /ANAL31/ NSFA , ISTN(21)	00012
ISN 0007	COMMON /ANAL32/ RPMIN , RPMAX , EORD(4) , NORD	00013
ISN 0008	COMMON /ANAL33/ DLED , DTED , DROOTD , DTIPD , TTID , TLTD , NCD	00014
	1 , NCK , TISD , TICD , PCBAD , PCGED	00015
ISN 0009	COMMON /ANAL04/ NSTA , MST1 , MST2 , NOACR , ILT , LONG , ISTE ,	00016
	1 DENB , EB , POIB , BR , SERIES , BRS , BTS , IOPP	00017
ISN 0010	COMMON /PLY1/ PLY(21,25,7) , THETAD(7) , RHO(7)	00018
ISN 0011	COMMON /GLOBCM/ OBJF , FN(5) , DLAR(5) , THKVAL(21) , RF(5,4)	00019
	1 , BRCC , FODLSB(21) , DLE , DTE , DROOT , DTIP , TTI , TLT	00020
	2 , OBJFUN , SMAX1S(8) , SMAX2S(8) , SMAX3S(8)	00021
	3 , SMAXLS(8) , SMAXTS(8) , THETA(7) , HLRTIO , ECRTIO , TOVB(21)	00022
	4 , FODMAX , TIS , TIC , PCBA	00023
ISN 0012	DOUBLE PRECISION DLED , DTED , DROOTD , DTIPD , TTID , TLTD	00024
	1 , THETAD , PLY , RHO , TISD , TICD , PCBAD , PCGED	00025
C		00026
C	READ NTIS - NUMBER OF THICKNESS INPUT STATIONS (MAX OF 21)	00027
C	READ NRF - NUMBER OF ROOTS CALCULATED BY FLUTTER ANALYSIS	00028
C	(MAX OF 5)	00029
C	READ NSFA - NUMBER OF W137 OUTPUT STATIONS USED FOR	00030
C	THE FLUTTER ANALYSIS (MAX OF 21)	00031
C	READ NORD - NUMBER OF ORDERS FOR RESONANCE FUNCTION	00032
C	CALCULATION (MAX OF 4)	00033
C		00034
ISN 0013	READ (I5I,900) NTIS , NRF , NSFA , NORD , BRVS	00035
C		00036
C	READ NTIS VALUES OF STATION NUMBER AND THICKNESS	00037
C		00038
ISN 0014	READ (I5I,901) (IST(I),VALT(I) , I = 1,NTIS)	00039
C		00040
C	READ NRF VALUES OF NODAL DIAMETER (1 PER ROOT) FOR INITIAL	00041
C	STARTING VALUES FOR THE FLUTTER CALCULATION	00042
C		00043
ISN 0015	IF (NRF .EQ. 0) GO TO 70	00044
ISN 0017	READ (I5I,902) (ND(I) , I = 1,NRF)	00045
ISN 0018	IF (ND(1) .EQ. 0) ND(1) = -4	00046
ISN 0020	IF (ND(2) .EQ. 0) ND(2) = 4	00047
ISN 0022	IF (ND(3) .EQ. 0) ND(3) = 4	00048
ISN 0024	IF (ND(4) .EQ. 0) ND(4) = -4	00049
ISN 0026	IF (ND(5) .EQ. 0) ND(5) = 4	00050
C		00051
C	READ W137 OUTPUT STATION NUMBERS TO BE USED	00052
C	IN THE FLUTTER ANALYSIS	00053
C		00054

188

ISN 0028	60 CONTINUE	00055
ISN 0029	READ (ISI,903) (ISTN(I) , I = 1,NSFA)	00056
ISN 0030	NBRS = BRS + .05	00057
ISN 0031	NCK = NSTA - NBRS + 1	00058
ISN 0032	IF (NCK .NE. NSFA) GO TO 70	00059
ISN 0034	DO 65 I = 1,NSFA	00060
ISN 0035	ISTN(I) = I + 3	00061
ISN 0036	65 CONTINUE	00062
ISN 0037	70 CONTINUE	00063
	C	00064
	C	00065
	C	00066
	C	00067
ISN 0038	READ (ISI,904) RPMIN , RPMX , EORD(1) , EORD(2)	00068
	1 , EORD(3) , EORD(4)	00069
	C	00070
	C	00071
	C	00072
	C	00073
	C	00074
ISN 0039	READ (ISI,905) NCD	00075
ISN 0040	IF (NCD .EQ. 2) GO TO 90	00076
	C	00077
	C	00078
	C	00079
	C	00080
	C	00081
	C	00082
	C	00083
	C	00084
	C	00085
	C	00086
ISN 0042	READ (ISI,906) DLE , DTE , DROOT , DTIP , TTI , TLT	00087
ISN 0043	DLED = DLE	00088
ISN 0044	DTED = DTE	00089
ISN 0045	DROOTD = DROOT	00090
ISN 0046	DTIPD = DTIP	00091
ISN 0047	TTID = TTI	00092
ISN 0048	TLTD = TLT	00093
ISN 0049	90 CONTINUE	00094
	C	00095
	C	00096
	C	00097
ISN 0050	READ (ISI,907) (THETA(I),I=1,7)	00098
ISN 0051	DO 100 I = 1,7	00099
ISN 0052	THETAD(I) = THETA(I)	00100
ISN 0053	100 CONTINUE	00101
ISN 0054	IF (NCD .EQ. 1) GO TO 110	00102
	C	00103
	C	00104
	C	00105
	C	00106
	C	00107
	C	00108
	C	00109
ISN 0056	READ (ISI,907) TIS , TIC , PCBA	00110
ISN 0057	TISD = TIS	00111
ISN 0058	TICD = TIC	00112
ISN 0059	PCBAD = PCBA	00113

```

+VERSION 1.3.0 (01 MAY 80)  RDDATA  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 82.141/10.52.30  PAGE 3
ISN 0060          PCGED = 1.0 - PCBAD          00114
ISN 0061          110 CONTINUE                 00115
ISN 0062          900 FORMAT (4I5,10X,F10.5)    00116
ISN 0063          901 FORMAT (8(I2,F8.0))      00117
ISN 0064          902 FORMAT (16I5)           00118
ISN 0065          903 FORMAT (20I4)           00119
ISN 0066          904 FORMAT (2F10.5,4F5.2)    00120
ISN 0067          905 FORMAT (I2)             00121
ISN 0068          906 FORMAT (6F10.5)         00122
ISN 0069          907 FORMAT (7F10.5)         00123
ISN 0070          RETURN                      00124
ISN 0071          END                          00125

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODCK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS*      SOURCE STATEMENTS =      70, PROGRAM SIZE =      1244, SUBPROGRAM NAME =RDDATA
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

3004K BYTES OF CORE NOT USED

REQUESTED OPTIONS: SOURCE,NOMAP,NOXREF,NOLIST,NODECK,OPT(3),AUTODBL(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTHT NOXREF NOALC NOANSF TERM IBM FLAG(I)

190

```

C          DATA SET U498SCALCT AT LEVEL 010 AS OF 01/29/82
C          DATA SET U498SCALCT AT LEVEL 009 AS OF 01/28/82          00001
C          DATA SET U498SCALCT AT LEVEL 008 AS OF 10/19/81          00002
C          DATA SET U500CALCTH AT LEVEL 007 AS OF 07/23/81          00003
C          DATA SET U500CALCTH AT LEVEL 006 AS OF 07/17/81          00004
C          DATA SET U500CALCTH AT LEVEL 005 AS OF 07/16/81          00005
C          DATA SET U500CALCTH AT LEVEL 004 AS OF 06/17/81          00006
ISN 0002   SUBROUTINE CALCTH          00007
ISN 0003   COMMON /UIOS/ I5I , I6I          00008
ISN 0004   COMMON /ANAL04/ NSTA , MST1 , MST2 , NOACR , ILT , LONG , ISTE , 00009
          1 DENB , EB , POIB , BR , SERIES , BRS , BTS , IOPP          00010
ISN 0005   COMMON /ANAL05/ R(21) , BOBR(21),TOB(21),THSTH(21),ALPHA(21), 00011
          1 SOB(21) , RLE(21) , RTE(21) , ADNT(21) , O(21)          00012
ISN 0006   COMMON /ANAL17/ XPS(3,53,21) , TMX(21) , TCC(21) , BMX(21) 00013
          1 , BCC(21)          00014
ISN 0007   COMMON /ANAL30/ NTIS , NRF , IST(21) , VALT(21) , BRSV          00015
ISN 0008   COMMON /GLOBCM/ OBJF , FN(5) , DLAR(5) , THKVAL(21) , RF(5,4) 00016
          1 , BRCC          00017
ISN 0009   DIMENSION RIST(21) , THKVEL(21)          00018
          C          00019
          C          DETERMINE RADIUS AT EACH INPUT THICKNESS STATION 00020
          C          00021
          C          WRITE (I6I,910) NSTA , BRS          00022
ISN 0010   DO 10 I = 1,NTIS          00023
ISN 0011   N = IST(I)          00024
ISN 0012   RIST(I) = R(N)          00025
ISN 0013   THKVEL(I) = THKVAL(I)          00026
          C          WRITE (I6I,910) I , THKVAL(I) , THKVEL(I)          00027
          C          WRITE (I6I,910) I , RIST(I)          00028
ISN 0014   910 FORMAT (I5,2F12.8)          00029
ISN 0015   10 CONTINUE          00030
          C          00031
          C          CALCULATE THICKNESS AT EACH M137 INPUT RADIUS USING 00032
          C          A LINEAR FIT OF THKVAL          00033
          C          00034
ISN 0016   N = 2          00035
ISN 0017   WRITE (I6I,920)          00036
ISN 0018   920 FORMAT (/, ' STA THICKNESS CHORD' )          00037
ISN 0019   NBRS = BRS + .05          00038
ISN 0020   TCC(NBRS) = THKVEL(1)          00039
ISN 0021   BCC(NBRS) = BRCC          00040
ISN 0022   WRITE (I6I,910) NBRS , TCC(NBRS) , BCC(NBRS)          00041
ISN 0023   IF (N .GT. NTIS) GO TO 30          00042
ISN 0025   NN = NBRS + 1          00043
ISN 0026   DO 20 I = NN,NSTA          00044
ISN 0027   IF( BRSV .EQ. 0.0) GO TO 15          00045
ISN 0029   BCC(I) = BMX(I) * BRCC / BMX(NBRS)          00046
ISN 0030   15 IF(R(I) .LE. RIST(N)) GO TO 16          00047
ISN 0032   N = N + 1          00048
ISN 0033   IF (N .GT. NTIS) GO TO 30          00049
ISN 0035   GO TO 15          00050
ISN 0036   16 TCC(I)=THKVEL(N-1) -(THKVEL(N-1)-THKVEL(N))*(R(I)-RIST(N-1) ) 00051
          1 / (RIST(N) - RIST(N-1))          00052
ISN 0037   WRITE (I6I,910) I , TCC(I) , BCC(I)          00053
ISN 0038   20 CONTINUE          00054
    
```

```
+VERSION 1.3.0 (01 MAY 80)  CALCTH  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 82.141/10.52.33  PAGE 2
      C  WRITE (I6I,910) N , THKVEL(1)  00055
      RETURN  00056
      30 WRITE (I6I,900) N , NTIS  00057
      900 FORMAT (' N = ',I5,' NTIS = ',I5,' N EXCEEDS NTIS'//)  00058
      STOP  00059
      END  00060
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTNT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 42, PROGRAM SIZE = 1032, SUBPROGRAM NAME =CALCTH
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION ***** 3012K BYTES OF CORE NOT USED
```

REQUESTED OPTIONS: SOURCE,NOMAP,NOXREF,NOLIST,NODECK,OPT(3),AUTODBL(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC HOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

192

C	DATA SET U498SIT983 AT LEVEL 009 AS OF 03/10/82	
C	DATA SET U498SIT983 AT LEVEL 008 AS OF 11/09/81	00001
C	DATA SET U500ITT983 AT LEVEL 007 AS OF 06/09/81	00002
C	DATA SET U500ITT983 AT LEVEL 006 AS OF 06/03/81	00003
ISN 0002	SUBROUTINE ITT983 (ICALC)	00004
ISN 0003	COMMON /UIOS/ ISI , I6I	00005
ISN 0004	COMMON /ANAL30/ NTIS , NRF , IST(21) , VALT(21)	00006
ISN 0005	COMMON /ANAL54/ ND(5) , DLSV(5)	00007
ISN 0006	COMMON /SECO/ FLAP(21,21), TWIST(21,21), ACOEF(21,4), ZODALD(21),	00008
	1 RED(21), DELAER(21), F2M5D, F2P5D	00009
ISN 0007	COMMON /JMPCHK/ JMP(5) , MODE	00010
ISN 0008	COMMON /ANAL11/ MAY , NAX , ISTR , HBB , DEND , DENA , ED	00011
	1 , DIAND , BLADES , POID	00012
ISN 0009	MNMIN = 2	00013
ISN 0010	MNMAX = -2	00014
ISN 0011	MNMAX = ((BLADES + .05) / 2.) -2.	00015
ISN 0012	MNMAX = -MNMAX	00016
ISN 0013	LIM = NRF	00017
ISN 0014	DO 5 I = 1,5	00018
ISN 0015	JMP(I) = 0	00019
ISN 0016	5 CONTINUE	00020
ISN 0017	WRITE (I6I,110)	00021
ISN 0018	MODE = 1	00022
ISN 0019	10 INC = 1	00023
ISN 0020	IHAVE = 1	00024
ISN 0021	IF (ND(MODE) .NE. MNMIN) GO TO 13	00025
ISN 0023	ND(MODE) = MNMIN	00026
ISN 0024	GO TO 17	00027
ISN 0025	13 IF (ND(MODE) .EQ. MNMAX .OR. ND(MODE) .EQ. MNMIN)	00028
	1 ND(MODE) = ND(MODE) + 1	00029
ISN 0027	IF (ND(MODE) .EQ. NMAX .OR. ND(MODE) .EQ. MNMIN)	00030
	1 ND(MODE) = ND(MODE) - 1	00031
ISN 0029	17 IF (ND(MODE) .LT. 0) IHAVE = -1	00032
ISN 0031	NODE = IABS (ND(MODE))	00033
ISN 0032	CALL WTT983 (ICALC,MODE,NODE,IWAVE)	00034
ISN 0033	CALL MNT983	00035
ISN 0034	DELSAV = DELAER(1)	00036
ISN 0035	NSAV = ND(MODE)	00037
ISN 0036	WRITE (I6I,100) DELSAV , DELAER(1) , NSAV , ND(MODE)	00038
ISN 0037	NOD = ND(MODE)	00039
ISN 0038	CALL NDT983 (NOD,NODE,IWAVE,INC)	00040
ISN 0039	ND(MODE) = NODE * IWAVE	00041
ISN 0040	CALL WTT983 (ICALC,MODE,NODE,IWAVE)	00042
ISN 0041	CALL MNT983	00043
ISN 0042	IF (DELAER(1) .EQ. DELSAV) GO TO 40	00044
ISN 0044	IF (DELAER(1) .LT. DELSAV) GO TO 20	00045
ISN 0046	INC = -1	00046
ISN 0047	WRITE (I6I,100) DELSAV , DELAER(1) , NSAV , ND(MODE)	00047
ISN 0048	ND(MODE) = ND(MODE) + INC	00048
ISN 0049	IF (ND(MODE) .EQ. 1) ND(MODE) = MNMIN	00049
ISN 0051	GO TO 30	00050
ISN 0052	20 DELSAV = DELAER(1)	00051
ISN 0053	NSAV = ND(MODE)	00052
ISN 0054	WRITE (I6I,100) DELSAV , DELAER(1) , NSAV , ND(MODE)	00053
ISN 0055	30 NOD = ND(MODE)	00054

```

ISN 0056 CALL NDT983 (NCD,MODE,IWAVE,INC) 00055
ISN 0057 ND(MODE) = NODE * IWAVE 00056
ISN 0058 CALL WTT983 (ICALC,MODE,NODE,IWAVE) 00057
ISN 0059 CALL MNT983 00058
ISN 0060 IF (DELAER(1) .GE. DELSAV) GO TO 40 00059
ISN 0062 DELSAV = DELAER(1) 00060
ISN 0063 NSAV = ND(MODE) 00061
ISN 0064 WRITE (I6I,100) DELSAV , DELAER(1) , NSAV , ND(MODE) 00062
ISN 0065 GO TO 30 00063
ISN 0066 40 CONTINUE 00064
ISN 0067 WRITE (I6I,100) DELSAV , DELAER(1) , NSAV , ND(MODE) 00065
ISN 0068 ND(MODE) = NSAV 00066
ISN 0069 DLSV(MODE) = DELSAV 00067
ISN 0070 JMP(MODE) = 0 00068
ISN 0071 MODE = MODE + 1 00069
ISN 0072 IF (MODE .GT. LIM) GO TO 50 00070
ISN 0074 GO TO 10 00071
ISN 0075 50 RETURN 00072
ISN 0076 100 FORMAT (5X,E12.5,5X,E12.5,5X,I5,5X,I5) 00073
ISN 0077 110 FORMAT (/5X,'DELAERO-SAVE',8X,'DELAERO',9X,'ND-SAVE',4X,'ND') 00074
ISN 0078 END 00075

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOOBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 77, PROGRAM SIZE = 1402, SUBPROGRAM NAME =ITT983
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

3004K BYTES OF CORE NOT USED

REQUESTED OPTIONS: SOURCE,NOMAP,NOXREF,NOLIST,NODECK,OPT(3),AUTCDBL(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

194

```

C          DATA SET U498SND983 AT LEVEL 002 AS OF 06/01/80
C          DATA SET U500NDT983 AT LEVEL 001 AS OF 05/15/81          00001
ISN 0002  SUBROUTINE  NDT983 (NOD,NODE,IWAVE,INC)                    00002
C                                                              00003
C          THIS ROUTINE SELECTS A NODAL DIAMETER (NOD) BETWEEN -(NB/2-2) AND 00004
C          -2 AND BETWEEN 2 AND (NB/2-2) AND SETS NODE TO THE ABS(NOD) AND 00005
C          THE WAVE DIRECTION EQUAL TO THE SIGN OF NOD. THIS ROUTINE 00006
C          INCREMENT NOD BY 1(INC) UNTIL DELTA AERO IN THE CALLING ROUTINE 00007
C          IS MINIMIZED.                                           00008
C                                                              00009
ISN 0003  COMMON /ANAL11/ MAY , NAX , ISTR , NBB , DEND , DENA , ED 00010
          1 , DIAND , BLADES , POID                                00011
ISN 0004  COMMON /JMPCHK/ JMP(5) , MODE                             00012
ISN 0005  NMIN = 2                                                00013
ISN 0006  NMAX = ((BLADES +.05) / 2. ) - 2.                       00014
ISN 0007  IF ( IABS(NOD) .GT. NMIN .AND. IABS(NOD) .LT. NMAX ) GO TO 10 00015
ISN 0009  IF (JMP(MODE) .NE. 0) GO TO 10                          00016
ISN 0011  JMP(MODE) = 1                                           00017
ISN 0012  NOD = NOD *(-1)                                         00018
ISN 0013  GO TO 20                                                00019
ISN 0014  10 NOD = NOD + INC                                       00020
ISN 0015  20 IWAVE = 1                                             00021
ISN 0016  IF ( NOD .LT. 0 ) IWAVE = -1                            00022
ISN 0018  NODE = IABS (NOD)                                       00023
ISN 0019  RETURN                                                  00024
ISN 0020  END                                                      00025

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTCDBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

STATISTICS SOURCE STATEMENTS = 19, PROGRAM SIZE = 464, SUBPROGRAM NAME =NDT983

STATISTICS NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

3016K BYTES OF CORE NOT USED

REQUESTED OPTIONS: SOURCE,NOMAP,NOXREF,NOLIST,NODECK,OPT(3),AUTODBL(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```

C          DATA SET U498SRD137 AT LEVEL 004 AS OF 02/01/82
C          DATA SET U498SRD137 AT LEVEL 003 AS OF 11/23/81          00001
C          DATA SET U493SRD137 AT LEVEL 002 AS OF 05/28/81          00002
C          DATA SET U500RDW137 AT LEVEL 001 AS OF 04/06/81          00003
ISN 0002   SUBROUTINE RDH137 (ICALC)                                00004
ISN 0003   COMMON /UIOS/ ISI , I6I                                  00005
ISN 0004   COMMON /ANAL01/ NTEST                                    00006
ISN 0005   COMMON /ANAL02/ ITTLE(18), IOP1 , IOP2 , IOP3           00007
ISN 0006   COMMON /ANAL03/ RPM , FN , DFN , TOLL , ROOT , DRPM,RPMMN,RPMMX 00008
ISN 0007   COMMON /ANAL04/ NSTA , MST1 , MST2 , NOACR , ILT , LONG , ISTE , 00009
1 DENB , EB , POIB , BR , SERIES , BRS , BTS , IOPP              00010
ISN 0008   COMMON /ANAL05/ R(21) , BOBR(21),TOB(21),THSTH(21),ALPHA(21), 00011
1 SOB(21) , RLE(21) , RTE(21) , ADWT(21) , O(21)                 00012
ISN 0009   COMMON /ANAL06/ TLTA(21) , TLTT(21) , XADWT(21) , YADWT(21) , 00013
1 AKX(21) , AKY(21) , AKZ(21)                                     00014
ISN 0010   COMMON /ANAL07/ XP(3,53,21) , NP(21)                    00015
ISN 0011   COMMON /ANAL08/ THET , THER , AKMIN , AKMAX , BRANG     00016
ISN 0012   COMMON /ANAL09/ WTS1 , XSHR1 , YSHR1 , ARSH1 , XISH1 , XISH3 , 00017
1 AKSH1 , SHPOI1 , ANGS1 , NSL                                   00018
ISN 0013   COMMON /ANAL10/ WTS2 , XSHR2 , YSHR2 , ARSH2 , XISH2 , XISH4 , 00019
1 AKSH2 , SHPOI2 , ANGS2                                         00020
ISN 0014   COMMON /ANAL11/ MAY , MAX , ISTR , NDB , DEND , DENA , ED 00021
1 , DIAND , BLADES , POID                                         00022
ISN 0015   COMMON /ANAL12/ RAD(32) , THE (32)                       00023
ISN 0016   COMMON /ANAL13/ SIGBOR , RADST(32) , TANST(32)          00024
ISN 0017   COMMON /ANAL14/ ELP , WP , WBP , DBP , WNP , DNP , EL , W , 00025
1 WS , DS , WZ , DT , RZ , THZ , TTH , DF ,                       00026
2 HT , EMU , DLX , RRT , BHP , TNP , WR , RDR                     00027
ISN 0018   COMMON /ANAL15/ SA , SB , SC , SD , SE , SF , SG , SH , SI , SJ , 00028
1 SK , SL , SN , REF , W1 , T11 , R2 , D2 , D3 , D4 , D5 , D6 , 00029
2 DLR , T1 , T2 , T3 , T4 , RA , BETD                             00030
ISN 0019   COMMON /ANAL16/ JXN(30) , VAL(30,12) , III              00031
ISN 0020   DIMENSION X(3,53)                                        00032
C          READ FIRST FOUR CARDS OF THE W137 INPUT                  00033
C          CARD 1 --- NEW CASE CONTROL CARD                          00034
C          CARD 2 --- TITLE CARD AND RING AND RIM PROPERTIES OPTIONS 00035
C          CARD 3 --- SPEED , FREQUENCIES , ROOT , ECT.             00036
C          CARD 4 --- W137 CONTROL CARD                              00037
C          CARD 4 --- W137 CONTROL CARD                              00038
C          CARD 4 --- W137 CONTROL CARD                              00039
ISN 0021   IXI = ISI                                                00040
ISN 0022   IF (ICALC .GT. 1) IXI = 24                               00041
C          IF (ICALC .EQ. 3) IXI = 5                                 00042
ISN 0024   IF (IXI .EQ. 24 .OR. IXI .EQ. 5) REWIND IXI              00043
ISN 0026   READ (IXI,900) NTEST                                      00044
ISN 0027   READ (IXI,901) ITTLE,IOP1,IOP2,IOP3                     00045
ISN 0028   READ (IXI,902) RPM,FN,DFN,TOLL,ROOT,DRPM,RPMMN,RPMMX    00046
ISN 0029   READ (IXI,903) NSTA,MST1,MST2,NOACR,ILT,LONG,ISTE,DENB,EB, 00047
1 POIB,BR,SERIES,BRS,BTS,IOPP                                     00048
ISN 0030   DO 20 I = 1 , NSTA                                        00049
ISN 0031   READ (IXI,904) R(I),BOBR(I),TOB(I),THSTH(I),ALPHA(I),SCB(I), 00050
1 RLE(I),RTE(I),ADWT(I),O(I)                                     00051
C          READ TILT AND SPRING CARD IF:                             00052
C          1--- TILT IS INPUT AT EACH STATION (ILT = 1)             00053
C          1--- TILT IS INPUT AT EACH STATION (ILT = 1)             00054

```

```

+VERSION 1.3.0 (01 MAY 80)   RDW137   SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)   DATE 82.141/10.52.48   PAGE 2
C       2--- TILT IS INPUT AT THE TIP STATION (ILT = 2 ) AND THE   00055
C       TIP STATION IS BEING READ ( I = NO. OF STATIONS )   00056
C       3--- ADDED WEIGHT HAS BEEN INPUT   00057
C   00058
ISN 0032   IF (ILT .EQ. 1 ) GO TO 5   00059
ISN 0034   IF (ILT .EQ. 2 .AND. I .EQ. NSTA ) GO TO 5   00060
ISN 0036   IF ( ADWT(I) ) 5,6,5   00061
ISN 0037   5 READ (IXI,905) TLTA(I) , TLTT(I) , XADWT(I) , YADWT(I) ,   00062
           1 AKX(I) , AKY(I) , AKZ(I)   00063
C   00064
C   IF T/B IS POSITIVE ---- STOP READING AIRFOIL DATA   00055
C   IF T/B IS NEGATIVE ---- READ COORDINATES   00066
C   00067
ISN 0038   6 IF ( TOB(I) .GT. 0.0 ) GO TO 20   00068
ISN 0040   N = O(I) + .05   00069
ISN 0041   DO 10 JI = 1,3   00070
ISN 0042   READ (IXI,905) ( X(JI,IN) , IN = 1,N )   00071
ISN 0043   10 CONTINUE   00072
ISN 0044   DO 15 M = 1,N   00073
ISN 0045   XP(1,M,I) = X(1,M)   00074
ISN 0046   XP(2,M,I) = X(2,M)   00075
ISN 0047   XP(3,M,I) = X(3,M)   00076
ISN 0048   15 CONTINUE   00077
ISN 0049   NP(I) = N   00078
ISN 0050   20 CONTINUE   00079
C   00080
C   READ ROOT AND TIP ANGLE , M/O , AND BROACH ANGLE   00081
C   00082
ISN 0051   READ (IXI,906) THET , THER , AKMIN , AKMAX , BRANG   00083
C   00084
C   READ SHROUD DATA (MST1 = 0,NO SHROUD )   00085
C   00086
ISN 0052   IF ( MST1 .LT. 1 ) GO TO 30   00087
ISN 0054   READ (IXI,907) WTS1 , XSHR1 , YSHR1 , ARSH1 , XISH1 , XISH3 ,   00088
           1 AKSH1 , SHPOI1 , ANG51 , NSL   00089
ISN 0055   IF ( MST2 .LT. 2 ) GO TO 30   00090
ISN 0057   READ (IXI,907) WTS2 , XSHR2 , YSHR2 , ARSH2 , XISH2 , XISH4 ,   00091
           1 AKSH2 , SHPOI2 , ANG52   00092
C   00093
C   READ DISC CONTROL DATA   00094
C   00095
C   00096
ISN 0058   30 CONTINUE   00096
ISN 0059   READ (IXI,908) MAY , NAX , ISTR , NSB , DEND , DENA , ED   00097
           1 , DIAND , BLADES , POID   00098
C   00099
C   READ DISC RADII AND THICKNESSES IF MAY .NE. 0   00100
C   00101
C   00102
ISN 0060   IF ( MAY .EQ. 0 ) GO TO 40   00102
ISN 0062   READ (IXI,909) (RAD(I),THE(I), I = 1 ,MAY)   00103
ISN 0063   40 CONTINUE   00104
C   00105
C   READ BORE STRESS , STRESS AT EACH STATION , OR NO STRESS   00106
C   00107
C   00108
ISN 0064   IF (ISTR) 50,70,60   00108
ISN 0065   50 READ (IXI,910) SIGBOR   00109
ISN 0066   GO TO 70   00110
ISN 0067   60 J = NAX + 1   00111
ISN 0068   READ (IXI,909) (RADST(I) , TANST(I), I= J , MAY )   00112
ISN 0069   70 CONTINUE   00113

```

198

```

C      00114
C      READ DOVETAIL INPUT OR SKIP TO RING INPUT      00115
C      00116
ISN 0070      IF ( IOP1 .EQ. 0 ) GO TO 80      00117
ISN 0072      READ (IXI,902) ELP,WP,WBP,DBP,WNP,DNP,EL,W      00118
ISN 0073      READ (IXI,902) WS,DS,WZ,DT,RZ,THZ,TTH,DF      00119
ISN 0074      READ (IXI,902) WT,ENU,DLX,RRT,ENP,TNP,WR      00120
ISN 0075      IF ( WR .NE. 0. ) READ (IXI,910) RDR      00121
ISN 0077      80 CONTINUE      00122
C      00123
C      READ DISC RIM INPUT IOP2 = 1 OR 2      00124
C      00125
ISN 0078      IF ( IOP2 - 1 ) 110,100,90      00126
ISN 0079      90 READ (IXI,902) SA , SB , SC , SD , SE , SF , SG , SH      00127
ISN 0080      READ (IXI,902) SI , SJ , SK , SL , SN , REF , WI      00128
ISN 0081      IF ( WI ) 95,110,95      00129
ISN 0082      95 READ (IXI,902) T11 , R2      00130
ISN 0083      GO TO 110      00131
ISN 0084      100 READ (IXI,902) D2 , D3 , D4 , D5 , D6 , DLR , W1      00132
ISN 0085      READ (IXI,902) T1 , T2 , T3 , T4 , RA , BETD      00133
ISN 0086      IF ( W1 ) 105,110,105      00134
ISN 0087      105 READ (IXI,902) T11 , R2      00135
ISN 0088      110 CONTINUE      00136
C      00137
C      READ SPRING AND RING PROPERTY INFORMATION      00138
C      00139
ISN 0089      I = 1      00140
ISN 0090      120 READ (IXI,911) JXN(I) ,(VAL(I,L),L=1,12)      00141
ISN 0091      IF ( JXN(I) .EQ. 0 ) GO TO 130      00142
ISN 0093      I = I + 1      00143
ISN 0094      GO TO 120      00144
ISN 0095      130 I = I + 1      00145
ISN 0096      READ (IXI,911) JXN(I)      00146
ISN 0097      III = I - 2      00147
ISN 0098      RETURN      00148
ISN 0099      900 FORMAT(72X,I1)      00149
ISN 0100      901 FORMAT(18A4,1X,3I1)      00150
ISN 0101      902 FORMAT(8F8.0)      00151
ISN 0102      903 FORMAT(7I2,2X,7F8.0,1X,I1)      00152
ISN 0103      904 FORMAT(9F8.0,1X,F4.0)      00153
ISN 0104      905 FORMAT(9F8.0)      00154
ISN 0105      906 FORMAT(5E8.0)      00155
ISN 0106      907 FORMAT(9F8.0,1X,I1)      00156
ISN 0107      908 FORMAT(4I2,8X,6F8.0)      00157
ISN 0108      909 FORMAT(8F8.2)      00158
ISN 0109      910 FORMAT(F8.2)      00159
ISN 0110      911 FORMAT(I2,6(F3.0,F8.0))      00160
ISN 0111      END      00161

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTHT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 110, PROGRAM SIZE = 3472, SUBPROGRAM NAME =RDW137
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

2988K BYTES OF CORE NOT USED

```

C      DATA SET U498SRSFUN AT LEVEL 003 AS OF 01/28/82
C      DATA SET U498SRSFUN AT LEVEL 002 AS OF 06/24/81      00001
C      DATA SET U500RSFUNC AT LEVEL 001 AS OF 06/22/81      00002
ISN 0002      SUBROUTINE RSFUNC                                00003
ISN 0003      COMMON /UIOS/ I5I , I6I                          00004
ISN 0004      COMMON /ANAL03/ RPM , FNS , DFN , TOLL ,ROOT,DRPM,RPMMN,RPMX  00005
ISN 0005      COMMON /ANAL21/ FNRQ(100) , SPEED(100) , BTA(100) 00006
ISN 0006      COMMON /ANAL32/ RFMN , RPMX , ECRD(4) , NORD      00007
ISN 0007      COMMON /GLOBCH/ QBJF , FN(5) , DLAR(5) , THKVAL(21) , RF(5,4) 00008
ISN 0008      DOUBLE PRECISION FNRQ , SPEED , BTA              00009
C
C      SAVE THE LARGER VALUE OF RFMAX AND RFMIN FOR EACH      00010
C      ORDER AND EACH ROOT                                    00011
C      ORDER AND EACH ROOT                                    00012
C      ORDER AND EACH ROOT                                    00013
ISN 0009      NROOT = ROOT + .05                                00014
ISN 0010      WRITE (I6I,910)                                    00015
ISN 0011      910 FORMAT(/,5X,'FREQUENCY',8X,'SPEED',10X,'BETA') 00016
ISN 0012      DO 3 I = 1,NROOT                                    00017
ISN 0013      WRITE (I6I,900) FNRQ(I) , SPEED(I) , BTA(I)      00018
ISN 0014      3 CONTINUE                                         00019
ISN 0015      900 FORMAT (5(3X,E12.5))                            00020
ISN 0016      WRITE (I6I,920)                                    00021
ISN 0017      920 FORMAT( /,6X,'RPM-MIN',8X,'RPMMAX',9X,'ND(1)', 00022
                1 10X,'ND(2)',10X,'ND(3)',10X,'ND(4)')          00023
ISN 0018      WRITE (I6I,900) RFMN , RPMX , ECRD(1) , ECRD(2) , ECRD(3) 00024
ISN 0019      WRITE (I6I,930)                                    00025
ISN 0020      930 FORMAT (/,7X,'DMAX',11X,'DMIN',11X,'RFMAX',10X,'RFMIN',11X,'RF') 00026
ISN 0021      DO 10 I = 1,NROOT                                    00027
ISN 0022      DO 5 J = 1,NORD                                      00028
ISN 0023      DMAX = FNRQ(I)**2 + BTA(I) * (RPMX**2 - SPEED(I)**2) 00029
ISN 0024      DMIN = FNRQ(I)**2 + BTA(I) * (RPMN**2 - SPEED(I)**2) 00030
ISN 0025      RFMAX = 60. * SQRT(DMAX) / ECRD(J) / RPMX - 1.0    00031
ISN 0026      RFMIN = 1.0 - 60. * SQRT(DMIN) / ECRD(J) / RPMN   00032
ISN 0027      RF(I,J) = RFMAX                                    00033
ISN 0028      IF (RFMIN .GT. RFMAX) RF(I,J) = RFMIN             00034
ISN 0030      WRITE (I6I,900) DMAX , DMIN , RFMAX , RFMIN , RF(I,J) 00035
ISN 0031      5 CONTINUE                                         00036
ISN 0032      10 CONTINUE                                        00037
ISN 0033      RETURN                                             00038
ISN 0034      END                                               00039

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
 *OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
 STATISTICS SOURCE STATEMENTS = 33, PROGRAM SIZE = 1048, SUBPROGRAM NAME =RSFUNC
 STATISTICS NO DIAGNOSTICS GENERATED
 END OF COMPILATION **

3012K BYTES OF CORE NOT USED

REQUESTED OPTIONS: SOURCE, NCMAP, NOXREF, NOLIST, NODECK, OPT(3), AUTODBL(NONE), NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NCMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

200

```

C          DATA SET U498STMAX  AT LEVEL 003 AS OF 02/01/82
C          DATA SET U498STMAX  AT LEVEL 002 AS OF 07/16/81          00001
C          DATA SET U500THMAX  AT LEVEL 001 AS OF 04/06/81          00002
ISN 0002  SUBROUTINE TMAX (ICALC)          00003
ISN 0003  COMMON /UIOS/ I5I , I6I          00004
ISN 0004  COMMON /ANAL04/ NSTA , MST1 , MST2 , NOACR , ILT , LONG , ISTE , 00005
          1 DENB , EB , POIB , BR , SERIES , BRS , BTS , IOPP          00006
ISN 0005  COMMON /ANAL07/ XP(3,53,21) , NP(21)          00007
ISN 0006  COMMON /ANAL17/ XPS(3,53,21) , TMX(21) , TCC(21) , BMX(21) 00008
          1 , ECC(21)          00009
ISN 0007  NBRS = BRS + .05          00010
ISN 0008  IF ( ICALC .EQ. 3 ) GO TO 70          00011
ISN 0010  IF ( ICALC .EQ. 1 .OR. ICALC .EQ. 2 ) GO TO 10          00012
ISN 0012  WRITE (I6I,900) ICALC          00013
ISN 0013  STOP          00014
ISN 0014  10 IF ( ICALC .EQ. 2 ) GO TO 70          00015
C          C          00016
C          DETERMINE MAX THICKNESS OF EACH INPUT STATION AND SAVE COORDINATES 00017
C          C          00018
ISN 0016  DO 30 I = NBRS , NSTA          00019
ISN 0017  NPP = NP(I)          00020
ISN 0018  TMAX =XP(2,1,I) -XP(3,1,I)          00021
ISN 0019  BMX(I) = XP(1,NPP,I) - XP(1,1,I)          00022
ISN 0020  DO 20 J = 1 , NPP          00023
ISN 0021  T =XP(2,J,I) - XP(3,J,I)          00024
ISN 0022  TMAX = AMAX1 (TMAX,T)          00025
ISN 0023  20 CONTINUE          00026
ISN 0024  TMX(I) = TMAX          00027
ISN 0025  30 CONTINUE          00028
ISN 0026  DO 60 I = NBRS , NSTA          00029
ISN 0027  NPP = NP(I)          00030
ISN 0028  DO 50 J = 1 ,NPP          00031
ISN 0029  DO 40 K = 1,3          00032
ISN 0030  XPS(K,J,I) = XP(K,J,I)          00033
ISN 0031  40 CONTINUE          00034
ISN 0032  50 CONTINUE          00035
ISN 0033  60 CONTINUE          00036
ISN 0034  GO TO 999          00037
ISN 0035  70 CONTINUE          00038
C          C          00039
C          RATIO THE Y-UPPERS AND Y-LOWERS TO REFLECT THE NEW          00040
C          THICKNESS FROM COPES-CONMIN          00041
C          C          00042
ISN 0036  DO 90 I = NBRS,NSTA          00043
ISN 0037  NPP = NP(I)          00044
ISN 0038  PER = TCC(I) / TMX(I)          00045
ISN 0039  DO 80 J = 1 , NPP          00046
ISN 0040  T = XPS(2,J,I) - XPS(3,J,I)          00047
ISN 0041  XS = T/TMX(I)          00048
ISN 0042  XP(2,J,I) = XPS(2,J,I) + (PER - 1.) * .5 * T * XS          00049
ISN 0043  XP(3,J,I) = XPS(3,J,I) - (PER - 1.) * .5 * T * XS          00050
ISN 0044  80 CONTINUE          00051
ISN 0045  90 CONTINUE          00052
C          C          00053
C          RATIO THE Y-UPPERS AND Y-LOWERS TO REFLECT THE NEW          00054

```

```

+VERSION 1.3.0 (01 MAY 80)      TMAX      SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)      DATE 82.141/10.52.58      PAGE 2
      C      CHORD FROM COPES-CONMIN      00055
      C      00056
ISN 0046      DO 110 I = NBRS,NSTA      00057
ISN 0047      NPP = NP(I)      00058
ISN 0048      IF (BCC(I) .EQ. 0.0 .OR. BMX(I) .EQ. 0.0) GO TO 110      00059
ISN 0050      PER = BCC(I) / BMX(I)      00060
ISN 0051      DO 100 J = 1 , NPP      00061
ISN 0052      XP(1,J,I) = XP(1,J,I) + (PER - 1.0) * XP(1,J,I)      00062
ISN 0053      XP(2,J,I) = XP(2,J,I) + (PER - 1.0) * XP(2,J,I)      00063
ISN 0054      XP(3,J,I) = XP(3,J,I) + (PER - 1.0) * XP(3,J,I)      00064
ISN 0055      100 CONTINUE      00065
ISN 0056      110 CONTINUE      00066
ISN 0057      900 FORMAT ( ' I CALC = ',I10)      00067
ISN 0058      999 RETURN      00068
ISN 0059      END      00069

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS*      SOURCE STATEMENTS =      58, PROGRAM SIZE =      1098, SUBPROGRAM NAME = TMAX
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

3004K BYTES OF CORE NOT USED

REQUESTED OPTIONS: SOURCE,NOHAP,NOXREF,NOLIST,NODECK,OPT(3),AUTODBL(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOHAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

202

C	DATA SET U498SWT983 AT LEVEL 014 AS OF 01/29/82	
C	DATA SET U498SWT983 AT LEVEL 013 AS OF 12/02/81	00001
C	DATA SET U498SWT983 AT LEVEL 012 AS OF 09/14/81	00002
C	DATA SET U500HTT983 AT LEVEL 011 AS OF 09/09/81	00003
C	DATA SET U500HTT983 AT LEVEL 010 AS OF 06/09/81	00004
ISN 0002	SUBROUTINE HTT983 (ICALC,MODE,IND,IWAVE)	00005
ISN 0003	COMMON/UIOS/ ISI , I6I	00006
ISN 0004	COMMON/ANAL04/ NSTA , MST1 , MST2 , NOCAR , ILT , LONG , ISTE ,	00007
	1 DENB , EB , POIB , BR , SERIES , BRS , BTS , IOPP	00008
ISN 0005	COMMON /ANAL11/ MAY , NAX , ISTR , NBB , DEND , DENA , ED	00009
	1 , DIAND , BLADES , POID	00010
ISN 0006	COMMON /ANAL17/ XPS(3,53,21) , TMX(21) , TCC(21) , BMX(21)	00011
	1 , BCC(21)	00012
ISN 0007	COMMON/ANAL20/ SPEED(5) , ANOD(5) , FRFN(5) , TKEB(5)	00013
	1 , RAD(5,21) , CHD(5,21) , SCT(5,21)	00014
	2 , ALPH(5,21) , PFDF(5,21) , PFAG(5,21) , PWDF(5,21) , PWAG(5,21)	00015
	3 , IRNO(5) , NBLD(5) , NI37(5)	00016
ISN 0008	COMMON/ANAL50/ TITL(20)	00017
ISN 0009	COMMON/ANAL51/ MACH , MAERO , NUMA , MH137 , IWD , IBETA , IGUST	00018
	1 , MGUST , NGUST , IPARM , IPRNT	00019
ISN 0010	COMMON/ANAL52/ TIAERO(20)	00020
ISN 0011	COMMON/ANAL53/ RAERO(21) , AMACH1(21) , VEL1(21) , TEMP1(21)	00021
	1 , PRES(21) , PSII(21) , BETA1(21)	00022
ISN 0012	COMMON /ANAL31/ NSFA , ISTD(21)	00023
ISN 0013	DOUBLE PRECISION SPEED , ANOD , FRFN , TKEB	00024
	1 , RAD , CHD , SCT , ALPH , PFDF , PFAG , PWDF , PWAG	00025
ISN 0014	IF (ICALC .EQ. 2) REWIND 5	00026
ISN 0016	NBR = BRS + .05	00027
ISN 0017	IYI = I6I	00028
ISN 0018	IF (ICALC .EQ. 2) IYI = 5	00029
C		00030
C	WRITE TITLE CARD - - CARD 1	00031
C		00032
ISN 0020	WRITE (IYI,900) TITL	00033
C		00034
C	WRITE CONTROL CARD - - CARD 2	00035
C		00036
ISN 0021	NUMA = 1	00037
ISN 0022	MH137 = 1	00038
ISN 0023	NIN = 5	00039
ISN 0024	MN = 5	00040
ISN 0025	IGUST = 0	00041
ISN 0026	MGUST = 0	00042
ISN 0027	NGUST = 0	00043
ISN 0028	IPARM = 0	00044
ISN 0029	IPRNT = 0	00045
ISN 0030	WRITE (IYI,901) MACH , MAERO , NUMA , MH137 , IWAVE , NIN , MN	00046
	1 , IBETA , IGUST , MGUST , NGUST , IPARM , IPRNT	00047
C		00048
C	WRITE AERO CASE TITLE - - CARD 3	00049
C		00050
ISN 0031	WRITE (IYI,900) TIAERO	00051
C		00052
C	WRITE AERO DATA - - CARD 4	00053
C		00054

```

ISN 0032      DO 10 I = 1,MAERO                00055
ISN 0033      WRITE (IYI,902) RAERO(I) , AMACHI(I) , VEL1(I) , TEMPI(I) 00056
              1 , PRES(I) , PSII(I) , BETA1(I) 00057
ISN 0034      10 CONTINUE                      00058
              C                               00059
              C WRITE W137 CONTROL CARD - - CARD 7 00060
              C                               00061
ISN 0035      DIA = IND                        00062
ISN 0036      ANBLD = BLADES * BMX(NBR) / BCC(NBR) 00063
ISN 0037      WRITE (IYI,903) SPEED(MODE) , DIA , IRNO(MODE) , FRFN(MODE) , 00064
              1 TKEB(MODE) , ANBLD , NSFA      00065
              C WRITE (26,903) SPEED(MODE) , DIA , IRNO(MODE) , FRFN(MODE) , 00066
              C 1 TKEB(MODE) , ANBLD , NSFA    00067
              C                               00068
              C WRITE W137 MODE SHAPE - - CARD 8 00069
              C                               00070
ISN 0038      DO 20 J = 1,NSFA                 00071
ISN 0039      I = ISTN(J)                     00072
ISN 0040      WRITE (IYI,904) RAD(MODE,I) , CHD(MODE,I) , SCT(MODE,I) , 00073
              1 ALPH(MODE,I) , PFDF(MODE,I) , PFAG(MODE,I) , PRDF(MODE,I) , 00074
              2 PWAG(MODE,I)                  00075
ISN 0041      20 CONTINUE                      00076
ISN 0042      IF (IYI .EQ. 5) REWIND 5        00077
ISN 0044      RETURN                          00078
ISN 0045      900 FORMAT (20A4)               00079
ISN 0046      901 FORMAT (5I5,I3,I2,6I5)     00080
ISN 0047      902 FORMAT (2F10.5,3F10.3,2F10.4) 00081
ISN 0048      903 FORMAT (2F10.2,I10,F10.4,2E10.4,I10) 00082
ISN 0049      904 FORMAT (4F10.5,E10.4,F10.4,E10.4,F10.4) 00083
ISN 0050      END                            00084

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORNAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 49, PROGRAM SIZE = 1206, SUBPROGRAM NAME =WTT983
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

3008K BYTES OF CORE NOT USED

REQUESTED OPTIONS: SOURCE,NOMAP,NOXREF,NOLIST,NODECK,OPT(3),AUTODBL(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

204

```

C          DATA SET U498SHT137 AT LEVEL 005 AS OF 03/11/82
C          DATA SET U498SHT137 AT LEVEL 004 AS OF 01/28/82          00001
C          DATA SET U498SHT137 AT LEVEL 003 AS OF 12/02/81          00002
C          DATA SET U498SHT137 AT LEVEL 002 AS OF 05/28/81          00003
C          DATA SET U500WTH137 AT LEVEL 001 AS OF 04/06/81          00004
ISN 0002   SUBROUTINE WTH137 (ICALC)                                00005
ISN 0003   COMMON /UIOS/ I5I , I6I                                  00006
ISN 0004   COMMON /ANAL01/ NTEST                                    00007
ISN 0005   COMMON /ANAL02/ ITTLE(18) , IOP1 , IOP2 , IOP3          00008
ISN 0006   COMMON /ANAL03/ RPM , FN , DFN , TOLL , ROOT , DRPM ,RPMMN,RPMMX 00009
ISN 0007   COMMON /ANAL04/ NSTA , MST1 , MST2 , NOACR , ILT , LONG , ISTE , 00010
1 DENB , EB , POIB , BR , SERIES , BRS , BTS , IOPP              00011
ISN 0008   COMMON /ANAL05/ R(21) , BOER(21),TOB(21),THSTH(21),ALPHA(21), 00012
1 SOB(21) , RLE(21) , RTE(21) , ADHT(21) , O(21)                00013
ISN 0009   COMMON /ANAL06/ TLTA(21) , TLTT(21) , XADWT(21) , YADWT(21) , 00014
1 AKX(21) , AKY(21) , AKZ(21)                                    00015
ISN 0010   COMMON /ANAL07/ XP(3,53,21) , NP(21)                    00016
ISN 0011   COMMON /ANAL08/ THET , THER , AKMIN , AKMAX , BRANG      00017
ISN 0012   COMMON /ANAL09/ WTS1 , XSHR1 , YSHR1 , ARSH1 , XISH1 , XISH3 , 00018
1 AKSH1 , SHPOI1 , ANGS1 , NSL                                   00019
ISN 0013   COMMON /ANAL10/ WTS2 , XSHR2 , YSHR2 , ARSH2 , XISH2 , XISH4 , 00020
1 AKSH2 , SHPOI2 , ANGS2                                        00021
ISN 0014   COMMON /ANAL11/ MAY , NAX , ISTR , NBB , DEND , DENA , ED 00022
1 , DIAND , BLADES , POID                                       00023
ISN 0015   COMMON /ANAL12/ RAD(32) , THE (32)                       00024
ISN 0016   COMMON /ANAL13/ SIGBOR , RADST(32) , TANST(32)          00025
ISN 0017   COMMON /ANAL14/ ELP , HP , WBP , DSP , WNP , DNP , EL , W , 00026
1 WS , DS , WZ , DT , RZ , THZ , TTH , DF ,                      00027
2 WT , EMU , DLX , RRT , BNP , TNP , HR , RDR                    00028
ISN 0018   COMMON /ANAL15/ SA , SB , SC , SD , SE , SF , SG , SH , SI , SJ , 00029
1 SK , SL , SN , REF , W1 , T11 , R2 , D2 , D3 , D4 , D5 , D6 , 00030
2 DLR , T1 , T2 , T3 , T4 , RA , BETD                            00031
ISN 0019   COMMON /ANAL16/ JXN(30) , VAL(30,12) , III              00032
ISN 0020   COMMON /ANAL17/ XPS(3,53,21) , TMX(21) , TCC(21) , BMX(21) 00033
1 , BCC(21)                                                      00034
C                                                                    00035
C          WRITE FIRST FOUR CARDS OF THE W137 INPUT                00036
C          CARD 1 --- NEW CASE CONTROL CARD                          00037
C          CARD 2 --- TITLE CARD AND RING AND RIM PROPERTIES OPTIONS 00038
C          CARD 3 --- SPEED , FREQUENCIES , ROOT , ECT.            00039
C          CARD 4 --- W137 CONTROL CARD                              00040
C                                                                    00041
ISN 0021   IYI = I6I                                                00042
ISN 0022   IF (ICALC .EQ. 1) IYI = 24                              00043
ISN 0024   IF (ICALC .EQ. 2) IYI = 5                               00044
ISN 0026   IF (ICALC .EQ. 4) IYI = 7                               00045
ISN 0028   IF (IYI .EQ. 5 .OR. IYI .EQ. 24) REWIND IYI             00046
ISN 0030   WRITE (IYI,900) NTEST                                    00047
ISN 0031   WRITE (IYI,901) ITTLE , IOP1 , IOP2 , IOP3              00048
ISN 0032   WRITE (IYI,902) RPM , FN , DFN , TOLL , ROOT , DRPM , RPMMN,RPMMX 00049
ISN 0033   WRITE (IYI,903) NSTA , MST1 , MST2 , NOACR , ILT , LONG , ISTE , 00050
1 DENB , EB , POIB , BR , SERIES , BRS , BTS , IOPP              00051
ISN 0034   DO 20 I = 1 , NSTA                                       00052
C                                                                    00053
C          WRITE AIRFOIL DATA                                       00054

```

```

C 00055
ISN 0035 WRITE (IYI,904) R(I) , BOBR(I) , TOB(I) , THSTH(I) , ALPHA(I) , 00056
1 SOB(I) , RLE(I) , RTE(I) , ADWT(I) , O(I) 00057
C 00058
C WRITE TILT AND SPRING CARD IF: 00059
C 1 --- TILT IS INPUT AT EACH STATION (ILT = 1) 00060
C 2 --- TILT IS INPUT AT THE TIP STATION (ILT = 2) AND THE 00061
C TIP STATION IS BEING READ (I = NO. OF STATIONS) 00062
C 3 --- ADDED WEIGHT HAS BEEN INPUT 00063
C 00064
ISN 0036 IF (ILT .EQ. 1) GO TO 5 00065
ISN 0038 IF (ILT .EQ. 1 .AND. I .EQ. NSTA) GO TO 5 00066
ISN 0040 IF (ADWT(I)) 5,6,5 00067
ISN 0041 5 WRITE (IYI,905) TLTA(I) , TLTT(I) , XADWT(I) , YADWT(I) , 00068
1 AKX(I) , AKY(I) , AKZ(I) 00069
C 00070
C IF T/B IS POSITIVE --- STOP READING AIRFOIL DATA 00071
C IF T/B IS NEGATIVE --- READ COORDINATES 00072
C 00073
ISN 0042 6 IF (TOB(I) .GT. 0.0) GO TO 20 00074
ISN 0044 N = O(I) + .05 00075
ISN 0045 DO 10 J = 1 , 3 00076
ISN 0046 WRITE (IYI,906) (XP(J,IN,I) , IN = 1 , N) 00077
ISN 0047 10 CONTINUE 00078
ISN 0048 20 CONTINUE 00079
C 00080
C WRITE ROOT AND TIP ANGLE , M/O , AND BROACH ANGLE 00081
C 00082
ISN 0049 WRITE (IYI,907) THET , THER , AKMIN , AKMAX , ERANG 00083
C 00084
C WRITE SHROUD DATA (MST1 = 0 , NO SHROUDS) 00085
C 00086
ISN 0050 IF (MST1 .LT. 1) GO TO 30 00087
ISN 0052 WRITE (IYI,908) WTS1 , XSHR1 , YSHR1 , ARSH1 , XISH1 , XISH3 , 00088
1 AKSH1 , SHPOI1 , ANGS1 , NSL 00089
ISN 0053 IF (MST2 .LT. 2) GO TO 30 00090
ISN 0055 WRITE (IYI,908) WTS2 , XSHR2 , YSHR2 , ARSH2 , XISH2 , XISH4 , 00091
1 AKSH2 , SHPOI2 , ANGS1 00092
ISN 0056 30 CONTINUE 00093
C 00094
C WRITE DISC CONTROL DATA 00095
C 00096
ISN 0057 BLADE = BLADES 00097
ISN 0059 IF (ICALC .EQ. 1) GO TO 35 00098
ISN 0060 NBR = BRS + .05 00099
ISN 0061 BLADE = BLADES * BMX(NBR) / BCC(NBR) 00100
ISN 0062 35 WRITE (IYI,909) MAY , NAX , ISTR , NBB , DEND , DENA , ED , 00101
1 DIAND , BLADE , POID 00102
C 00103
C WRITE DISC RADII AND THICKNESSES IF MAY .NE. 0 00104
C 00105
ISN 0063 IF (MAY .EQ. 0) GO TO 40 00106
ISN 0065 WRITE (IYI,910) (RAD(I),THE(I) , I = 1 , MAY) 00107
ISN 0066 40 CONTINUE 00108
C 00109
C WRITE BORE STRESS , STRESS AT EACH STATION , OR NO STRESS 00110
C 00111
ISN 0067 IF (ISTR) 50,70,60 00112
ISN 0068 50 WRITE (IYI,911) SIGBOR 00113

```

206

```

ISN 0069      GO TO 70                                00114
ISN 0070      60 J = MAX + 1                          00115
ISN 0071      WRITE (IYI,911) (RADST(I) , TANST(I) , I = J , MAY) 00116
ISN 0072      70 CONTINUE                             00117
C                                                     00118
C WRITE DOVETAIL INPUT OR SKIP TO RING INPUT         00119
C                                                     00120
ISN 0073      IF (IOP1 .EQ. 0) GO TO 80               00121
ISN 0075      WRITE (IYI,910) ELP , WP , WBP , DBP , WNP , DNP , EL , W 00122
ISN 0076      WRITE (IYI,910) WS , DS , WZ , DT , RZ , THZ , TTH , DF 00123
ISN 0077      WRITE (IYI,910) WT , EMU , DLX , RRT , BNP , TNP , WR 00124
ISN 0078      IF (WR .NE. 0.) WRITE (IYI,910) RDR     00125
ISN 0080      80 CONTINUE                             00126
C                                                     00127
C WRITE DISC RIM INPUT IF IOP2 = 1 OR 2              00128
C                                                     00129
C                                                     00130
ISN 0081      IF (IOP2 - 1) 110,100,90                00130
ISN 0082      90 WRITE (IYI,910) SA , SB , SC , SD , SE , SF , SG , SH 00131
ISN 0083      WRITE (IYI,910) SI , SJ , SK , SL , SH , REF , WI 00132
ISN 0084      IF (WI) 95,110,95                       00133
ISN 0085      95 WRITE (IYI,910) T11 , R2              00134
ISN 0086      GO TO 110                               00135
ISN 0087      100 WRITE (IYI,910) D2 , D3 , D4 , D5 , D6 , DLR , WI 00136
ISN 0088      WRITE (IYI,910) T1 , T2 , T3 , T4 , RA , BETD 00137
ISN 0089      IF (WI) 105,110,105                     00138
ISN 0090      105 WRITE (IYI,910) T11 , R2            00139
ISN 0091      110 CONTINUE                             00140
C                                                     00141
C WRITE SPRING AND RING PROPERTY INFORMATION         00142
C                                                     00143
C                                                     00144
ISN 0092      DO 120 I = 1 , III                      00144
ISN 0093      WRITE (IYI,912) JXN(I) , (VAL(I,L) , L = 1 , 12) 00145
ISN 0094      120 CONTINUE                             00146
ISN 0095      IXX = 0                                 00147
ISN 0096      WRITE (IYI,913) IXX                     00148
ISN 0097      WRITE (IYI,913) IXX                     00149
ISN 0098      IF (IYI .EQ. 5) REWIND IYI              00150
ISN 0100      RETURN                                  00151
ISN 0101      900 FORMAT (72X,I1)                     00152
ISN 0102      901 FORMAT (18A4,1X,3I1)                 00153
ISN 0103      902 FORMAT (3F8.1,F8.5,4F8.1)           00154
ISN 0104      903 FORMAT (7I2,2X,F8.5,E8.3,2F8.5,3F8.1,1X,I1) 00155
ISN 0105      904 FORMAT (3F8.5,2F8.4,4F8.5,1X,F4.0) 00156
ISN 0106      905 FORMAT (4F8.5,3E8.1)                00157
ISN 0107      906 FORMAT (9F8.5)                     00159
ISN 0108      907 FORMAT (2F8.4,2F8.0,F8.4)           00159
ISN 0109      908 FORMAT (3F8.5,F8.6,4F8.7,F8.4,1X,I1) 00160
ISN 0110      909 FORMAT (4I2,8X,2F8.5,E8.3,3F8.4)   00161
ISN 0111      910 FORMAT (8F8.5)                     00162
ISN 0112      911 FORMAT (8F8.1)                     00163
ISN 0113      912 FORMAT (I2,F3.0,E8.3,5(F3.0,F8.5)) 00164
ISN 0114      913 FORMAT (7X,I1)                     00165
ISN 0115      END                                     00166

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 114, PROGRAM SIZE = 3002, SUBPROGRAM NAME =NTH137
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****
2984K BYTES OF CORE NOT USED

```

```

C          DATA SET U498SOBJTV AT LEVEL 007 AS OF 03/16/82
C          DATA SET U498SOBJTV AT LEVEL 006 AS OF 03/03/82
C          DATA SET U498SOBJTV AT LEVEL 005 AS OF 02/03/82
C          DATA SET U498SOBJTV AT LEVEL 004 AS OF 12/02/81
C          DATA SET U498SOBJTV AT LEVEL 003 AS OF 11/23/81
C          DATA SET U498SOBJTV AT LEVEL 002 AS OF 11/20/81
C          DATA SET U498SOBJTV AT LEVEL 001 AS OF 11/19/81
ISN 0002  SUBROUTINE  OBJTV
C
C          CALCULATE THE OBJECTIVE FUNCTION FOR A HOLLOW FOIL WITH
C          BORSIC TITANIUM INLAY OR A SUPERHYBRID FOIL
C
ISN 0003  COMMON /GLOBCM/ OBJF , FN(5) , DLAR(5) , THKVAL(21) , RF(5,4)
          1 , BRCC , FODLSB(21) , DLE , DTE , DROOT , DTIP , TTI , TLT
          2 , OBJFUN
ISN 0004  COMMON /PLY1/ PLY(21,25,7) , THETA(7) , RHO(7)
ISN 0005  COMMON /WIEGHT/ WL(7)
ISN 0006  COMMON /ANAL33/ DLED , DTED , DROOTD , DTIPD , TTID , TLTD , NCD
ISN 0007  COMMON /ANAL17/ XPS(3,53,21) , THX(21) , TCC(21) , BMX(21)
          1 , BCC(21)
ISN 0008  COMMON /UIOS/ ISI , I6I
ISN 0009  COMMON /ANAL04/ NSTA , MST1 , MST2 , NOACR , ILT , LONG , ISTE ,
          1 DENS , EB , POIB , BR , SERIES , BRS , BTS , IOPP
ISN 0010  COMMON /ANAL05/ R(21) , BOBR(21),TOB(21),THSTH(21),ALPHA(21),
          1 SOB(21) , RLE(21) , RTE(21) , ADNT(21) , O(21)
ISN 0011  DOUBLE PRECISION PLY , THETA , RHO , WL , DLED , DTED , DROOTD
          1 , DTIPD , TTID , TLTD
C
C          NCD = 1 (HOLLOW FOIL)
C          NCD = 2 (SUPERHYBRID FOIL)
C
ISN 0012  RC = BRCC
ISN 0013  IF (BRCC .EQ. 0.0) RC = BMX(1)
ISN 0015  RT = THKVAL(1)
ISN 0016  IF (NCD .EQ. 2) GO TO 10
ISN 0018  TIVOL = (WL(1) + WL(7)) / RHO(1)
ISN 0019  BTVOL = (WL(2) + WL(6)) / RHO(2)
C
C          RC = ROOT CHORD
C          RT = ROOT THICKNESS
C          TIVOL = VOLUME OF TITANIUM
C          BTVOL = VOLUME OF BORSIC TITANIUM
C
ISN 0020  EMC = 2000. * (.00958 * TIVOL + 1.183 * (BTVOL/116.6)**1.08 )
ISN 0021  ELC = 2000.*( .273 * (RC / 9.25)**.35 + .095 * (RT/.925)**.86)
ISN 0022  EDC = 246.42 * (EMC + ELC) / RC +3700. * RC
ISN 0023  WRITE (I6I,900) RC , RT , TIVOL , BTVOL , EMC , ELC , EDC
ISN 0024  900 FORMAT (7(3X,E12.5)/)
C
C          EMC = MATERIAL COST
C          ELC = LABOR COST
C          EDC = DELTA COST RELATIVE TO E3
C
ISN 0025  TR = R(NSTA)
ISN 0026  HBRS = BRS + .05

```

```

+VERSION 1.3.0 (01 MAY 80)   OBJTV   SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)   DATE 82.141/10.53.08   PAGE 2
  ISN 0027      RR = R(NSRS)                                          00055
  ISN 0028      ARATIO = (TR**2 - (DROOT - 2. )**2) / (TR**2 - RR**2)   00056
  ISN 0029      IF ((DROOT - 2. ) .LT. RR) ARATIO = 1.0                00057
  ISN 0031      EDLE = DLE                                             00058
  ISN 0032      IF ( (DLE - .001) .LT. 0. ) EDLE = 20.0              00059
  ISN 0034      EDMNC = (EMC + ELC) * (4.0 + 6.8 * ARATIO/EDLE) * 220.0 00060
                  1 / 250000.0 / RC                                    00061
  C                                                     00062
  C      TR      = TIP RADIUS                                          00063
  C      RR      = ROOT RADIUS                                         00064
  C      EDMNC   = DELTA MAINTENCE AND MATERIAL COST                   00065
  C                                                     00066
  ISN 0035      WRITE (I6I,910) NSTA , NBRS,TR, RR , DROOT , DLE , ARATIO , EDMNC 00067
  ISN 0036      910 FORMAT (2I5,6(3X,E12.5)/)                          00068
  ISN 0037      FW = ( WL(1) + WL(7) + WL(2) + WL(6) ) * 386.4        00069
  ISN 0038      DWT = 99.5 * FW - 1.233 * FW**2                       00070
  ISN 0039      OBJFUN = .54 * EDC/100000. + .80 * EDMNC/10. + .52 * DWT/1000. 00071
  C                                                     00072
  C      FW      = FOIL WEIGHT                                         00073
  C      DWT     = DELTA WEIGHT RELATIVE TO E3                         00074
  C      OBJFUN  = OBJECTIVE FUNCTION                                  00075
  C                                                     00076
  ISN 0040      WRITE (I6I,900) WL(1) ,WL(2) ,WL(6) ,WL(7) ,FW ,DWT ,OBJFUN 00077
  ISN 0041      GO TO 50                                               00078
  ISN 0042      10 CONTINUE                                           00079
  C                                                     00080
  C      TIVOL   = VOLUME OF TITANIUM                                   00081
  C      BAVOL   = VOLUME OF BORON ALUMINUM                           00082
  C      GEVOL   = VOLUME OF GRAPHITE EPOXY                            00083
  C                                                     00084
  ISN 0043      TIVOL = (WL(1) + WL(4) + WL(7)) / RHO(1)              00085
  ISN 0044      BAVOL = (WL(2) + WL(6)) / RHO(2)                      00086
  ISN 0045      GEVOL = (WL(3) + WL(5)) / RHO(3)                      00087
  C                                                     00088
  C      EMC     = MATERIAL COST                                        00089
  C      ELC     = LABOR COST                                          00090
  C      EDC     = DELTA COST RELATIVE TO E3                           00091
  C      EDMNC   = DELTA MAINTENCE AND MATERIAL COST                   00092
  C                                                     00093
  ISN 0046      EMC = 2000.0 * ( .483 * (GEVOL / 116.6)**1.05         00094
                  1          + .637 * ( ( BAVOL + TIVOL ) / 116.6 )**1.08   00095
                  2          + .1445 * ( RC / 9.25 )**1.774 )            00096
  ISN 0047      ELC = 2000.0 * ( .063 * (RC / 9.25)**1.15            00097
                  1          + .121 * (RT / 9.25)**.82)                00098
  ISN 0048      EDC = 246.42 * (EMC + ELC) / RC + 3700.0 * RC        00099
  ISN 0049      EDMNC = 222.0 * (EMC + ELC) / (12500.0 * RC)         00100
  C                                                     00101
  C      FW      = FOIL WEIGHT                                         00102
  C      DWT     = DELTA WEIGHT RELATIVE TO E3                         00103
  C      OBJFUN  = OBJECTIVE FUNCTION                                  00104
  C                                                     00105
  ISN 0050      FW = (WL(1) +WL(2) +WL(3) +WL(4) +WL(5) + WL(6) +WL(7))*386.4 00106
  ISN 0051      DWT = 99.5 * FW - 1.233 * FW**2                       00107
  ISN 0052      OBJFUN = .54 * EDC/100000. + .80 * EDMNC/10. + .52 * DWT/1000. 00108
  ISN 0053      WRITE (I6I,900) RC , RT , TIVOL , BAVOL , GEVOL       00109
  ISN 0054      WRITE (I6I,900) (WL(I) , I=1,7)                       00110
  ISN 0055      WRITE (I6I,900) (RHO(I) , I=1,7)                      00111
  ISN 0056      WRITE (I6I,900) EMC , ELC , EDC , EDMNC , FW , DWT , OBJFUN 00112
  ISN 0057      50 CONTINUE                                           00113

```

+VERSION 1.3.0 (01 MAY 80) OBJTV SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 82.141/10.53.08 PAGE 3
 ISH 0058 RETURN 00114
 ISH 0059 END 00115
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTCOBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
STATISTICS SOURCE STATEMENTS = 58, PROGRAM SIZE = 1994, SUBPROGRAM NAME = OBJTV
STATISTICS NO DIAGNOSTICS GENERATED
***** END OF COMPILATION ***** 3000K BYTES OF CORE NOT USED

REQUESTED OPTIONS: SCURCE,NOMAP,NOXREF,NOLIST,NODECK,OPT(3),AUTODBL(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

210

```

C          DATA SET U477CNMNO1 AT LEVEL 001 AS OF 02/13/81
C          DATA SET 9188CNMNO1 AT LEVEL 001 AS OF 07/10/80          00001
ISN 0002   SUBROUTINE CNMNO1 (JGOTO,X,DF,G,ISC,IC,A,G1,VLB,VUB,SCAL,C,NCAL,DX00002
           1,DX1,FI,XI,III,N1,N2,N3,N4)          00003
ISN 0003   COMMON /CNMNO1/ DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,ALPHAX00004
           1,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,LINOBJ,ITHAX,IT00005
           2RM,ICNDIR,IGOTO,NAC,INFO,INFOG,ITER          00006
ISN 0004   DIMENSION X(N1), DF(N1), G(N2), ISC(N2), IC(N3), A(N1,N3), G1(N2),00007
           1 VLB(N1), VUB(N1), SCAL(N1), NCAL(2), CN(4)    00008
C          ROUTINE TO CALCULATE GRADIENT INFORMATION BY FINITE DIFFERENCE. 00009
C          BY G. N. VANDERPLAATS          JUNE, 1972.      00010
C          NASA-AMES RESEARCH CENTER,  MOFFETT FIELD, CALIF. 00011
ISN 0005   IF (JGOTO.EQ.1) GO TO 10          00012
ISN 0007   IF (JGOTO.EQ.2) GO TO 70          00013
ISN 0009   INFOG=0          00014
ISN 0010   INF=INFO          00015
ISN 0011   NAC=0          00016
ISN 0012   IF (LINOBJ.NE.0.AND.ITER.GT.1) GO TO 10 00017
C          -----00018
C          GRADIENT OF LINEAR OBJECTIVE          00019
C          -----00020
ISN 0014   IF (NFDG.EQ.2) JGOTO=1          00021
ISN 0016   IF (NFDG.EQ.2) RETURN          00022
ISN 0018   10 CONTINUE          00023
ISN 0019   JGOTO=0          00024
ISN 0020   IF (NFDG.EQ.2.AND.NCON.EQ.0) RETURN 00025
ISN 0022   IF (NCON.EQ.0) GO TO 40          00026
C          -----00027
C          * * * DETERMINE WHICH CONSTRAINTS ARE ACTIVE OR VIOLATED * * * 00028
C          -----00029
ISN 0024   DO 20 I=1,NCON          00030
ISN 0025   IF (G(I).LT.CT) GO TO 20          00031
ISN 0027   IF (ISC(I).GT.0.AND.G(I).LT.CTL) GO TO 20 00032
ISN 0029   NAC=NAC+1          00033
ISN 0030   IF (NAC.GE.N3) RETURN          00034
ISN 0032   IC(NAC)=I          00035
ISN 0033   20 CONTINUE          00036
ISN 0034   IF (NFDG.EQ.2.AND.NAC.EQ.0) RETURN 00037
ISN 0036   IF ((LINOBJ.GT.0.AND.ITER.GT.1).AND.NAC.EQ.0) RETURN 00038
C          -----00039
C          STORE VALUES OF CONSTRAINTS IN G1          00040
C          -----00041
ISN 0038   DO 30 I=1,NCON          00042
ISN 0039   30 G1(I)=G(I)          00043
ISN 0040   40 CONTINUE          00044
ISN 0041   JGOTO=0          00045
ISN 0042   IF (NAC.EQ.0.AND.NFDG.EQ.2) RETURN 00046
C          -----00047
C          CALCULATE GRADIENTS          00048
C          -----00049
ISN 0044   INFOG=1          00050
ISN 0045   INFO=1          00051
ISN 0046   FI=OBJ          00052
ISN 0047   III=0          00053
ISN 0048   50 III=III+1          00054

```

1

11

```

ISN 0049      XI=X(III)                                00055
ISN 0050      DX=FDCH*XI                              00056
ISN 0051      DX=ABS(DX)                              00057
ISN 0052      FDCHI=FDCHM                             00058
ISN 0053      IF (NSCAL.NE.0) FDCHI=FDCHM/SCAL(III)   00059
ISN 0055      IF (DX.LT.FDCHI) DX=FDCHI              00060
ISN 0057      X1=XI+DX                                00061
ISN 0058      IF (NSIDE.EQ.0) GO TO 60                00062
ISN 0060      IF (X1.GT.VUB(III)) DX=-DX             00063
ISN 0062      60  DX1=1./DX                            00064
ISN 0063      X(III)=X1+DX                            00065
ISN 0064      NCAL(1)=NCAL(1)+1                      00066
C -----00067
C           FUNCTION EVALUATION                      00068
C -----00069
ISN 0065      JGOTO=2                                  00070
ISN 0066      RETURN                                  00071
ISN 0067      70  CONTINUE                             00072
ISN 0068      X(III)=X1                               00073
ISN 0069      IF (NFDG.EQ.0) DF(III)=DX1*(OBJ-FI)     00074
ISN 0071      IF (NAC.EQ.0) GO TO 90                  00075
C -----00076
C           DETERMINE GRADIENT COMPONENTS OF ACTIVE CONSTRAINTS
C           00077
C -----00078
ISN 0073      DO 80 J=1,NAC                            00079
ISN 0074      I1=IC(J)                                00080
ISN 0075      80  A(III,J)=DX1*(G(I1)-GI(I1))         00081
ISN 0076      90  CONTINUE                             00082
ISN 0077      IF (III.LT.NDV) GO TO 50                00083
ISN 0079      INFOG=0                                 00084
ISN 0080      INFO=INF                                00085
ISN 0081      JGOTO=0                                  00086
ISN 0082      OBJ=FI                                   00087
ISN 0083      IF (NCON.EQ.0) RETURN                   00088
C -----00089
C           STORE CURRENT CONSTRAINT VALUES BACK IN G-VECTOR
C           00090
C -----00091
ISN 0085      DO 100 I=1,NCON                           00092
ISN 0086      100 G(I)=GI(I)                          00093
ISN 0087      RETURN                                  00094
ISN 0088      END                                     00095

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 87, PROGRAM SIZE = 1730, SUBPROGRAM NAME =CNM1N01
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

3004K BYTES OF CORE NOT USED

REQUESTED OPTIONS: SOURCE,NOHAP,NOXREF,NOLIST,NODECK,OPT(3),AUTODBL(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOHAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```

C          DATA SET U477CNMNO2 AT LEVEL 001 AS OF 02/13/81
C          DATA SET 9188CNMNO2 AT LEVEL 001 AS OF 07/10/80          00001
ISN 0002   SUBROUTINE CNMNO2 (NCALC,SLOPE,DFTDF1,DF,S,N1)          00002
ISN 0003   COMMON /CNMNI/ DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,ALPHAX00003
          1,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,LINOBJ,ITHAX,IT00004
          2RM,ICNDIR,IGOTO,MAC,INFO,INFOG,ITER          00005
ISN 0004   DIMENSION DF(N1), S(N1)          00006
          C          ROUTINE TO DETERMINE CONJUGATE DIRECTION VECTOR OR DIRECTION          00007
          C          OF STEEPEST DESCENT FOR UNCONSTRAINED FUNCTION MINIMIZATION.          00008
          C          BY G. N. VANDERPLAATS          APRIL, 1972.          00009
          C          NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.          00010
          C          NCALC = CALCULATION CONTROL.          00011
          C          NCALC = 0,          S = STEEPEST DESCENT.          00012
          C          NCALC = 1,          S = CONJUGATE DIRECTION.          00013
          C          CONJUGATE DIRECTION IS FOUND BY FLETCHER-REEVES ALGORITHM.          00014
          C          -----00015
          C          CALCULATE NORM OF GRADIENT VECTOR          00016
          C          -----00017
ISN 0005   DFTDF=0.          00018
ISN 0006   DO 10 I=1,NDV          00019
ISN 0007   DFI=DF(I)          00020
ISN 0008   DFTDF=DFTDF+DFI*DFI          00021
          C          -----00022
          C          *****          FIND DIRECTION S          *****00023
          C          -----00024
ISN 0009   IF (NCALC.NE.1) GO TO 30          00025
ISN 0011   IF (DFTDF1.LT.1.0E-20) GO TO 30          00026
          C          -----00027
          C          FIND FLETCHER-REEVES CONJUGATE DIRECTION          00028
          C          -----00029
ISN 0013   BETA=DFTDF/DFTDF1          00030
ISN 0014   SLOPE=0.          00031
ISN 0015   DO 20 I=1,NDV          00032
ISN 0016   DFI=DF(I)          00033
ISN 0017   SI=BETA*S(I)-DFI          00034
ISN 0018   SLOPE=SLOPE+SI*DFI          00035
ISN 0019   20 S(I)=SI          00036
ISN 0020   GO TO 50          00037
ISN 0021   30 CONTINUE          00038
ISN 0022   NCALC=0          00039
          C          -----00040
          C          CALCULATE DIRECTION OF STEEPEST DESCENT          00041
          C          -----00042
ISN 0023   DO 40 I=1,NDV          00043
ISN 0024   40 S(I)=-DF(I)          00044
ISN 0025   SLOPE=-DFTDF          00045
ISN 0026   50 CONTINUE          00046
          C          -----00047
          C          NORMALIZE S TO MAX ABS VALUE OF UNITY          00048
          C          -----00049
ISN 0027   S1=0.          00050
ISN 0028   DO 60 I=1,NDV          00051
ISN 0029   S2=ABS(S(I))          00052
ISN 0030   IF (S2.GT.S1) S1=S2          00053
ISN 0032   60 CONTINUE          00054

```

```
+VERSION 1.3.0 (01 MAY 80)  CNMNO2  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 82.141/10.53.16  PAGE 2
  ISN 0033      IF (S1.LT.1.0E-20) S1=1.0E-20      00055
  ISN 0035      S1=1./S1      00056
  ISN 0036      DFTDF1=DFTDF*S1      00057
  ISN 0037      DO 70 I=1,NDV      00058
  ISN 0038      70  S(I)=S1*S(I)      00059
  ISN 0039      SLOPE=S1*SLOPE      00060
  ISN 0040      RETURN      00061
  ISN 0041      END      00062
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NCMAP NOFORMAT GOSTHT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS*  SOURCE STATEMENTS = 40, PROGRAM SIZE = 654, SUBPROGRAM NAME =CNMNO2
*STATISTICS*  NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****                                3012K BYTES OF CORE NOT USED
```

REQUESTED OPTIONS: SOURCE,NOMAP,NOXREF,NOLIST,NODECK,OPT(3),AUTODBL(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

214

```

C          DATA SET U477CNMN03 AT LEVEL 002 AS OF 03/13/81
C          DATA SET U477CNMN03 AT LEVEL 001 AS OF 02/13/81          00001
C          DATA SET 9188CNMN03 AT LEVEL 001 AS OF 07/10/80          00002
ISN 0002   SUBROUTINE CNMN03 (X,S,SLOPE,ALP,FFF,A1,A2,A3,A4,F1,F2,F3,F4,APP,NO0003
          11,NCAL,KOUNT,JGOTO)          00004
ISN 0003   COMMON /CNMN1/ DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,ALPHAX00005
          1,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,LINOBJ,ITMAX,IT00006
          2RH,ICNDIR,IGOTO,NAC,INFO,INFOG,ITER          00007
ISN 0004   COMMON /UIOS/ I5I, I6I          00008
ISN 0005   DIMENSION X(N1), S(N1), NCAL(2)          00009
C          ROUTINE TO SOLVE ONE-DIMENSIONAL SEARCH IN UNCONSTRAINED          00010
C          MINIMIZATION USING 2-POINT QUADRATIC INTERPOLATION, 3-POINT          00011
C          CUBIC INTERPOLATION AND 4-POINT CUBIC INTERPOLATION.          00012
C          BY G. N. VANDERPLAATS          APRIL, 1972.          00013
C          NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.          00014
C          ALP = PROPOSED MOVE PARAMETER.          00015
C          SLOPE = INITIAL FUNCTION SLOPE = S-TRANSPOSE TIMES DF.          00016
C          SLOPE MUST BE NEGATIVE.          00017
C          OBJ = INITIAL FUNCTION VALUE.          00018
ISN 0006   ZRO=0.          00019
ISN 0007   IF (JGOTO.EQ.0) GO TO 10          00020
ISN 0009   GO TO (50,80,110,140,180,220,270),JGOTO          00021
C          -----00022
C          INITIAL INFORMATION (ALPHA=0)          00023
C          -----00024
ISN 0010   10 IF (SLOPE.LT.0.) GO TO 20          00025
ISN 0012   ALP=0.          00026
ISN 0013   RETURN          00027
ISN 0014   20 CONTINUE          00028
ISN 0015   IF (IPRINT.GT.4) WRITE (I6I,360)          00029
ISN 0017   FFF=OBJ          00030
ISN 0018   AP1=0.          00031
ISN 0019   A1=0.          00032
ISN 0020   F1=OBJ          00033
ISN 0021   A2=ALP          00034
ISN 0022   A3=0.          00035
ISN 0023   F3=0.          00036
ISN 0024   AP=A2          00037
ISN 0025   KOUNT=0          00038
C          -----00039
C          MOVE A DISTANCE AP*S AND UPDATE FUNCTION VALUE          00040
C          -----00041
ISN 0026   30 CONTINUE          00042
ISN 0027   KOUNT=KOUNT+1          00043
ISN 0028   DO 40 I=1,NDV          00044
ISN 0029   40 X(I)=X(I)+AP*S(I)          00045
ISN 0030   IF (IPRINT.GT.4) WRITE (I6I,370) AP          00046
ISN 0032   IF (IPRINT.GT.4) WRITE (I6I,380) (X(I),I=1,NDV)          00047
ISN 0034   NCAL(1)=NCAL(1)+1          00048
ISN 0035   JGOTO=1          00049
ISN 0036   RETURN          00050
ISN 0037   50 CONTINUE          00051
ISN 0038   F2=OBJ          00052
ISN 0039   IF (IPRINT.GT.4) WRITE (I6I,390) F2          00053
ISN 0041   IF (F2.LT.F1) GO TO 120          00054
    
```

```

C -----00055
C                CHECK FOR ILL-CONDITIONING                00056
C -----00057
ISN 0043          IF (KOUNT.GT.5) GO TO 60                  00058
ISN 0045          FF=2.*ABS(F1)                             00059
ISN 0046          IF (F2.LT.FF) GO TO 90                   00060
ISN 0048          FF=5.*ABS(F1)                             00061
ISN 0049          IF (F2.LT.FF) GO TO 60                   00062
ISN 0051          A2=.5*A2                                  00063
ISN 0052          AP=-A2                                    00064
ISN 0053          ALP=A2                                    00065
ISN 0054          GO TO 30                                  00066
ISN 0055          60  F3=F2                                 00067
ISN 0056          A3=A2                                    00068
ISN 0057          A2=.5*A2                                  00069
C -----00070
C                UPDATE DESIGN VECTOR AND FUNCTION VALUE    00071
C -----00072
ISN 0058          AP=A2-ALP                                 00073
ISN 0059          ALP=A2                                    00074
ISN 0060          DO 70 I=1,NDV                             00075
ISN 0061          70  X(I)=X(I)+AP*S(I)                    00076
ISN 0062          IF (IPRINT.GT.4) WRITE (I6I,370) A2     00077
ISN 0064          IF (IPRINT.GT.4) WRITE (I6I,380) (X(I),I=1,NDV) 00078
ISN 0066          NCAL(1)=NCAL(1)+1                       00079
ISN 0067          JGOTO=2                                   00080
ISN 0068          RETURN                                    00081
ISN 0069          80  CONTINUE                              00082
ISN 0070          F2=OBJ                                    00083
ISN 0071          IF (IPRINT.GT.4) WRITE (I6I,390) F2     00084
C                PROCEED TO CUBIC INTERPOLATION.            00085
ISN 0073          GO TO 160                                 00086
ISN 0074          90  CONTINUE                              00087
C -----00088
C                ***** 2-POINT QUADRATIC INTERPOLATION *****00089
C -----00090
ISN 0075          JJ=1                                      00091
ISN 0076          II=1                                      00092
ISN 0077          CALL CNMN04 (II,APP,ZRO,A1,F1,SLOPE,A2,F2,ZRO,ZRO,ZRO) 00093
ISN 0078          IF (APP.LT.ZRO.OR.APP.GT.A2) GO TO 120  00094
ISN 0080          F3=F2                                    00095
ISN 0081          A3=A2                                    00096
ISN 0082          A2=APP                                    00097
ISN 0083          JJ=0                                      00098
C -----00099
C                UPDATE DESIGN VECTOR AND FUNCTION VALUE    00100
C -----00101
ISN 0084          AP=A2-ALP                                 00102
ISN 0085          ALP=A2                                    00103
ISN 0086          DO 100 I=1,NDV                            00104
ISN 0087          100 X(I)=X(I)+AP*S(I)                   00105
ISN 0088          IF (IPRINT.GT.4) WRITE (I6I,370) A2     00106
ISN 0090          IF (IPRINT.GT.4) WRITE (I6I,380) (X(I),I=1,NDV) 00107
ISN 0092          NCAL(1)=NCAL(1)+1                       00108
ISN 0093          JGOTO=3                                   00109
ISN 0094          RETURN                                    00110
ISN 0095          110 CONTINUE                              00111
ISN 0096          F2=OBJ                                    00112
ISN 0097          IF (IPRINT.GT.4) WRITE (I6I,390) F2     00113

```

216

```

ISN 0099      GO TO 150                                00114
ISN 0100      120  A3=2.*A2                            00115
C             -----00116
C             UPDATE DESIGN VECTOR AND FUNCTION VALUE  00117
C             -----00118
ISN 0101      AP=A3-ALP                                00119
ISN 0102      ALP=A3                                  00120
ISN 0103      DO 130 I=1,NDV                          00121
ISN 0104      130  X(I)=X(I)+AP*S(I)                  00122
ISN 0105      IF (IPRINT.GT.4) WRITE (I6I,370) A3     00123
ISN 0107      IF (IPRINT.GT.4) WRITE (I6I,380) (X(I),I=1,NDV) 00124
ISN 0109      NCAL(1)=NCAL(1)+1                      00125
ISN 0110      JGOTO=4                                 00126
ISN 0111      RETURN                                  00127
ISN 0112      140  CONTINUE                          00128
ISN 0113      F3=OBJ                                  00129
ISN 0114      IF (IPRINT.GT.4) WRITE (I6I,390) F3     00130
ISN 0116      150  CONTINUE                          00131
ISN 0117      IF (F3.LT.F2) GO TO 190                 00132
ISN 0119      160  CONTINUE                          00133
C             -----00134
C             *****      3-POINT CUBIC INTERPOLATION      ***** 00135
C             -----00136
ISN 0120      II=3                                    00137
ISN 0121      CALL CNM104 (II,APP,ZRO,A1,F1,SLOPE,A2,F2,A3,F3,ZRO,ZRO) 00138
ISN 0122      IF (APP.LT.ZRO.OR.APP.GT.A3) GO TO 190  00139
C             -----00140
C             UPDATE DESIGN VECTOR AND FUNCTION VALUE.  00141
C             -----00142
ISN 0124      AP1=APP                                 00143
ISN 0125      AP=APP-ALP                             00144
ISN 0126      ALP=APP                                 00145
ISN 0127      DO 170 I=1,NDV                          00146
ISN 0128      170  X(I)=X(I)+AP*S(I)                  00147
ISN 0129      IF (IPRINT.GT.4) WRITE (I6I,370) ALP   00148
ISN 0131      IF (IPRINT.GT.4) WRITE (I6I,380) (X(I),I=1,NDV) 00149
ISN 0133      NCAL(1)=NCAL(1)+1                      00150
ISN 0134      JGOTO=5                                 00151
ISN 0135      RETURN                                  00152
ISN 0136      180  CONTINUE                          00153
ISN 0137      IF (IPRINT.GT.4) WRITE (I6I,390) OBJ   00154
C             -----00155
C             CHECK CONVERGENCE                        00156
C             -----00157
ISN 0139      AA=1.-APP/A2                            00158
ISN 0140      AB2=ABS(F2)                              00159
ISN 0141      AB3=ABS(OBJ)                            00160
ISN 0142      AB=AB2                                  00161
ISN 0143      IF (AB3.GT.AB) AB=AB3                   00162
ISN 0145      IF (AB.LT.1.0E-15) AB=1.0E-15         00163
ISN 0147      AB=(AB2-AB3)/AB                        00164
ISN 0148      IF (ABS(AB).LT.1.0E-15.AND.ABS(AA).LT..001) GO TO 330 00165
ISN 0150      A4=A3                                  00166
ISN 0151      F4=F3                                  00167
ISN 0152      A3=APP                                  00168
ISN 0153      F3=OBJ                                  00169
ISN 0154      IF (A3.GT.A2) GO TO 230                 00170
ISN 0156      A3=A2                                  00171
ISN 0157      F3=F2                                  00172

```

```

+VERSION 1.3.0 (01 MAY 80)   CNMN03   SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)   DATE 82.141/10.53.20   PAGE 4
ISN 0158      A2=APP                      00173
ISN 0159      F2=OBJ                      00174
ISN 0160      GO TO 230                    00175
ISN 0161      190 CONTINUE                  00176
C -----00177
C *****          4-POINT CUBIC INTERPOLATION          ***** 00178
C -----00179
ISN 0162      200 CONTINUE                  00180
ISN 0163      A4=2.*A3                      00181
C      UPDATE DESIGN VECTOR AND FUNCTION VALUE.          00182
ISN 0164      AP=A4-ALP                     00183
ISN 0165      ALP=A4                        00184
ISN 0166      DO 210 I=1,NDV                00185
ISN 0167      210 X(I)=X(I)+AP*S(I)         00186
ISN 0168      IF (IPRINT.GT.4) WRITE (I6I,370) ALP      00187
ISN 0170      IF (IPRINT.GT.4) WRITE (I6I,380) (X(I),I=1,NDV) 00188
ISN 0172      NCAL(1)=NCAL(1)+1             00189
ISN 0173      JGOTO=6                       00190
ISN 0174      RETURN                         00191
ISN 0175      220 CONTINUE                  00192
ISN 0176      F4=OBJ                        00193
ISN 0177      IF (IPRINT.GT.4) WRITE (I6I,390) F4       00194
ISN 0179      IF (F4.GT.F3) GO TO 230        00195
ISN 0181      A1=A2                          00196
ISN 0182      F1=F2                          00197
ISN 0183      A2=A3                          00198
ISN 0184      F2=F3                          00199
ISN 0185      A3=A4                          00200
ISN 0186      F3=F4                          00201
ISN 0187      GO TO 200                      00202
ISN 0188      230 CONTINUE                  00203
ISN 0189      II=4                           00204
ISN 0190      CALL CNM04 (II,APP,A1,A1,F1,SLOPE,A2,F2,A3,F3,A4,F4) 00205
ISN 0191      IF (APP.GT.A1) GO TO 250        00206
ISN 0193      AP=A1-ALP                      00207
ISN 0194      ALP=A1                         00208
ISN 0195      OBJ=F1                         00209
ISN 0196      DO 240 I=1,NDV                00210
ISN 0197      240 X(I)=X(I)+AP*S(I)         00211
ISN 0198      GO TO 280                      00212
ISN 0199      250 CONTINUE                  00213
C -----00214
C      UPDATE DESIGN VECTOR AND FUNCTION VALUE          00215
C -----00216
ISN 0200      AP=APP-ALP                     00217
ISN 0201      ALP=APP                        00218
ISN 0202      DO 260 I=1,NDV                00219
ISN 0203      260 X(I)=X(I)+AP*S(I)         00220
ISN 0204      IF (IPRINT.GT.4) WRITE (I6I,370) ALP      00221
ISN 0206      IF (IPRINT.GT.4) WRITE (I6I,380) (X(I),I=1,NDV) 00222
ISN 0208      NCAL(1)=NCAL(1)+1             00223
ISN 0209      JGOTO=7                       00224
ISN 0210      RETURN                         00225
ISN 0211      270 CONTINUE                  00226
ISN 0212      IF (IPRINT.GT.4) WRITE (I6I,390) OBJ      00227
ISN 0214      280 CONTINUE                  00228
C -----00229
C      CHECK FOR ILL-CONDITIONING                    00230
C -----00231

```

```

+VERSION 1.3.0 (01 MAY 80)  CNM1N03  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 82.141/10.53.20  PAGE 5
  ISN 0215      IF (OBJ.GT.F2.OR.OBJ.GT.F3) GO TO 290      00232
  ISN 0217      IF (OBJ.LE.F1) GO TO 330                  00233
  ISN 0219      AP=A1-ALP                                  00234
  ISN 0220      ALP=A1                                     00235
  ISN 0221      OBJ=F1                                     00236
  ISN 0222      GO TO 310                                  00237
  ISN 0223      290 CONTINUE                               00238
  ISN 0224      IF (F2.LT.F3) GO TO 300                   00239
  ISN 0226      OBJ=F3                                     00240
  ISN 0227      AP=A3-ALP                                  00241
  ISN 0228      ALP=A3                                     00242
  ISN 0229      GO TO 310                                  00243
  ISN 0230      300 OBJ=F2                                  00244
  ISN 0231      AP=A2-ALP                                  00245
  ISN 0232      ALP=A2                                     00246
  ISN 0233      310 CONTINUE                               00247
  C -----
  C                                     UPDATE DESIGN VECTOR 00248
  C -----
  C                                     UPDATE DESIGN VECTOR 00249
  C -----
  ISN 0234      DO 320 I=1,NDV                             00251
  ISN 0235      320 X(I)=X(I)+AP*S(I)                     00252
  ISN 0236      330 CONTINUE                               00253
  C -----
  C                                     CHECK FOR MULTIPLE MINIMA 00254
  C -----
  C                                     CHECK FOR MULTIPLE MINIMA 00255
  C -----
  C                                     CHECK FOR MULTIPLE MINIMA 00256
  ISN 0237      IF (OBJ.LE.FFF) GO TO 350                 00257
  C INITIAL FUNCTION IS MINIMUM.                          00258
  ISN 0239      DO 340 I=1,NDV                             00259
  ISN 0240      340 X(I)=X(I)-ALP*S(I)                     00260
  ISN 0241      ALP=0.                                     00261
  ISN 0242      OBJ=FFF                                    00262
  ISN 0243      350 CONTINUE                               00263
  ISN 0244      JGOTO=0                                    00264
  ISN 0245      RETURN                                    00265
  C -----
  C                                     FORMATS                    00266
  C -----
  C                                     FORMATS                    00267
  C -----
  C                                     FORMATS                    00268
  C -----
  C                                     FORMATS                    00269
  C -----
  C                                     FORMATS                    00270
  ISN 0246      360 FORMAT (/////5X,60H* * * UNCONSTRAINED ONE-DIMENSIONAL SEARCH INFO00271
  ISN 0247      370 FORMAT (/5X,7HALPHA =,E14.5/5X,8HX-VECTOR) 00272
  ISN 0248      380 FORMAT (5X,6E13.5)                    00273
  ISN 0249      390 FORMAT (/5X,5HOBJ =,E14.5)            00274
  ISN 0250      END                                        00275

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 249, PROGRAM SIZE = 3520, SUBPROGRAM NAME = CNM1N03
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

2972K BYTES OF CORE NOT USED

REQUESTED OPTIONS: SOURCE,NOMAP,NOXREF,NOLIST,NODECK,OPT(3),AUTODBL(NONE),NOALC

OPTIONS IN EFFECT: NAHE(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```

C          DATA SET U477CNMNO4 AT LEVEL 001 AS OF 02/13/81
C          DATA SET 9188CNMNO4 AT LEVEL 001 AS OF 07/10/80          00001
ISN 0002  SUBROUTINE CNMNO4 (II,XBAR,EPS,X1,Y1,SLOPE,X2,Y2,X3,Y3,X4,Y4) 00002
C          ROUTINE TO FIND FIRST XBAR.GE.EPS CORRESPONDING TO A MINIMUM 00003
C          OF A ONE-DIMENSIONAL REAL FUNCTION BY POLYNOMIAL INTERPOLATION. 00004
C          BY G. N. VANDERPLAATS          APRIL, 1972.          00005
C          NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.          00006
C          00007
C          II = CALCULATION CONTROL.          00008
C          1:  2-POINT QUADRATIC INTERPOLATION, GIVEN X1, Y1, SLOPE, 00009
C              X2 AND Y2.          00010
C          2:  3-POINT QUADRATIC INTERPOLATION, GIVEN X1, Y1, X2, Y2, 00011
C              X3 AND Y3.          00012
C          3:  3-POINT CUBIC INTERPOLATION, GIVEN X1, Y1, SLOPE, X2, Y2,00013
C              X3 AND Y3.          00014
C          4:  4-POINT CUBIC INTERPOLATION, GIVEN X1, Y1, X2, Y2, X3, 00015
C              Y3, X4 AND Y4.          00016
C          EPS MAY BE NEGATIVE.          00017
C          IF REQUIRED MINIMUM ON Y DOES NOT EXIST, OR THE FUNCTION IS 00018
C          ILL-CONDITIONED, XBAR = EPS-1.0 WILL BE RETURNED AS AN ERROR 00019
C          INDICATOR.          00020
C          IF DESIRED INTERPOLATION IS ILL-CONDITIONED, A LOWER ORDER 00021
C          INTERPOLATION, CONSISTANT WITH INPUT DATA, WILL BE ATTEMPTED, 00022
C          AND II WILL BE CHANGED ACCORDINGLY.          00023
ISN 0003  XBAR1=EPS-1.          00024
ISN 0004  XBAR=XBAR1          00025
ISN 0005  X21=X2-X1          00026
ISN 0006  IF (ABS(X21).LT.1.0E-20) RETURN          00027
ISN 0008  NSLOP=MOD(II,2)          00028
ISN 0009  GO TO (10,20,40,50),II          00029
ISN 0010  10 CONTINUE          00030
C          -----00031
C          II=1: 2-POINT QUADRATIC INTERPOLATION          00032
C          -----00033
ISN 0011  II=1          00034
ISN 0012  DX=X1-X2          00035
ISN 0013  IF (ABS(DX).LT.1.0E-20) RETURN          00036
ISN 0015  AA=(SLOPE+(Y2-Y1)/DX)/DX          00037
ISN 0016  IF (AA.LT.1.0E-20) RETURN          00038
ISN 0018  BB=SLOPE-2.*AA*X1          00039
ISN 0019  XBAR=-.5*BB/AA          00040
ISN 0020  IF (XBAR.LT.EPS) XBAR=XBAR1          00041
ISN 0022  RETURN          00042
ISN 0023  20 CONTINUE          00043
C          -----00044
C          II=2: 3-POINT QUADRATIC INTERPOLATION          00045
C          -----00046
ISN 0024  II=2          00047
ISN 0025  X21=X2-X1          00048
ISN 0026  X31=X3-X1          00049
ISN 0027  X32=X3-X2          00050
ISN 0028  QQ=X21*X31*X32          00051
ISN 0029  IF (ABS(QQ).LT.1.0E-20) RETURN          00052
ISN 0031  AA=(Y1*X32-Y2*X31+Y3*X21)/QQ          00053
ISN 0032  IF (AA.LT.1.0E-20) GO TO 30          00054

```

220

```

ISN 0034      BB=(Y2-Y1)/X21-AA*(X1+X2)          00055
ISN 0035      XBAR=-.5*BB/AA                    00056
ISN 0036      IF (XBAR.LT.EPS) XBAR=XBAR1       00057
ISN 0038      RETURN                            00058
ISN 0039      30 CONTINUE                       00059
ISN 0040      IF (NSLOP.EQ.0) RETURN            00060
ISN 0042      GO TO 10                          00061
ISN 0043      40 CONTINUE                       00062
C      -----00063
C      II=3: 3-POINT CUBIC INTERPOLATION        00064
C      -----00065
ISN 0044      II=3                              00066
ISN 0045      X21=X2-X1                        00067
ISN 0046      X31=X3-X1                        00068
ISN 0047      X32=X3-X2                        00069
ISN 0048      QQ=X21*X31*X32                  00070
ISN 0049      IF (ABS(QQ).LT.1.0E-20) RETURN   00071
ISN 0051      X11=X1*X1                        00072
ISN 0052      DNOM=X2*X2*X31-X11*X32-X3*X3*X21 00073
ISN 0053      IF (ABS(DNOM).LT.1.0E-20) GO TO 20 00074
ISN 0055      AA=((X31*X31*(Y2-Y1)-X21*X21*(Y3-Y1))/(X31*X21)-SLOPE*X32)/DNOM 00075
ISN 0056      IF (ABS(AA).LT.1.0E-20) GO TO 20 00076
ISN 0058      BB=((Y2-Y1)/X21-SLOPE-AA*(X2*X2+X1*X2-2.*X11))/X21 00077
ISN 0059      CC=SLOPE-3.*AA*X11-2.*BB*X1     00078
ISN 0060      BAC=BB*BB-3.*AA*CC              00079
ISN 0061      IF (BAC.LT.0.) GO TO 20          00080
ISN 0063      BAC=SQRT(BAC)                   00081
ISN 0064      XBAR=(BAC-BB)/(3.*AA)           00082
ISN 0065      IF (XBAR.LT.EPS) XBAR=EPS       00083
ISN 0067      RETURN                           00084
ISN 0068      50 CONTINUE                     00085
C      -----00086
C      II=4: 4-POINT CUBIC INTERPOLATION        00087
C      -----00088
ISN 0069      X21=X2-X1                        00089
ISN 0070      X31=X3-X1                        00090
ISN 0071      X41=X4-X1                        00091
ISN 0072      X32=X3-X2                        00092
ISN 0073      X42=X4-X2                        00093
ISN 0074      X11=X1*X1                        00094
ISN 0075      X22=X2*X2                        00095
ISN 0076      X33=X3*X3                        00096
ISN 0077      X44=X4*X4                        00097
ISN 0078      X111=X1*X11                     00098
ISN 0079      X222=X2*X22                     00099
ISN 0080      Q2=X31*X21*X32                  00100
ISN 0081      IF (ABS(Q2).LT.1.0E-30) RETURN  00101
ISN 0083      Q1=X111*X32-X222*X31+X3*X33*X21 00102
ISN 0084      Q4=X111*X42-X222*X41+X4*X44*X21 00103
ISN 0085      Q5=X41*X21*X42                  00104
ISN 0086      DNOM=Q2*Q4-Q1*Q5                00105
ISN 0087      IF (ABS(DNOM).LT.1.0E-30) GO TO 60 00106
ISN 0089      Q3=Y3*X21-Y2*X31+Y1*X32        00107
ISN 0090      Q6=Y4*X21-Y2*X41+Y1*X42        00108
ISN 0091      AA=(Q2*Q6-Q3*Q5)/DNOM           00109
ISN 0092      BB=(Q3-Q1*AA)/Q2                00110
ISN 0093      CC=(Y2-Y1-AA*(X222-X111))/X21-BB*(X1+X2) 00111
ISN 0094      BAC=BB*BB-3.*AA*CC              00112
ISN 0095      IF (ABS(AA).LT.1.0E-20.OR.BAC.LT.0.) GO TO 60 00113

```

```
+VERSION 1.3.0 (01 MAY 80)  CNM04  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 82.141/10.53.25  PAGE 3
  ISN 0097      BAC=SQRT(BAC)                                00114
  ISN 0098      XBAR=(BAC-EB)/(3.*AA)                        00115
  ISN 0099      IF (XBAR.LT.EPS) XBAR=XBAR1                  00116
  ISN 0101      RETURN                                       00117
  ISN 0102      60 CONTINUE                                   00118
  ISN 0103      IF (NSLOP.EQ.1) GO TO 40                     00119
  ISN 0105      GO TO 20                                     00120
  ISN 0106      END                                          00121
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS*   SOURCE STATEMENTS =   105, PROGRAM SIZE =   1754, SUSPROGRAM NAME =CNM04
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****                               3004K BYTES OF CORE NOT USED
```

222

```

C      DATA SET U477CNMNO5 AT LEVEL 002 AS OF 03/13/81
C      DATA SET U477CNMNO5 AT LEVEL 001 AS OF 02/13/81          00001
C      DATA SET 9189CNMNO5 AT LEVEL 001 AS OF 07/10/80          00002
ISN 0002  SUBROUTINE CNMNO5 (G,DF,A,S,B,C,SLOPE,PHI,ISC,IC,MSI,NVC,N1,N2,N3,00003
          1N4,N5)          00004
ISN 0003  COMMON /CNMNO1/ DELFUN,DABFUN,FDCH,FDCHM,CT,CTHIN,CTL,CTLMIN,ALPHAX00005
          1,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,LINOBJ,ITMAX,IT00006
          2RM,ICNDIR,IGOTO,NAC,INFO,INFOG,ITER          00007
ISN 0004  COMMON /UIOS/ I5I, I6I          00008
ISN 0005  DIMENSION DF(N1), G(N2), ISC(N2), IC(N3), A(N1,N3), S(N1), C(N4),
          1MSI(N5), B(N3,N3)          00010
C      ROUTINE TO SOLVE DIRECTION FINDING PROBLEM IN MODIFIED METHOD OF 00011
C      FEASIBLE DIRECTIONS.          00012
C      BY G. N. VANDERPLAATS          MAY, 1972.          00013
C      NASA-ANES RESEARCH CENTER, MOFFETT FIELD, CALIF.          00014
C      NORM OF S VECTOR USED HERE IS S-TRANSPOSE TIMES S.LE.1.          00015
C      IF NVC = 0 FIND DIRECTION BY ZOUTENDIJK'S METHOD. OTHERWISE          00016
C      FIND MODIFIED DIRECTION.          00017
C      -----00018
C      *** NORMALIZE GRADIENTS, CALCULATE THETA'S AND DETERMINE NVC ***00019
C      -----00020
ISN 0006  NDV1=NDV+1          00021
ISN 0007  NDV2=NDV+2          00022
ISN 0008  NAC1=NAC+1          00023
ISN 0009  NVC=0          00024
ISN 0010  THMAX=0.          00025
ISN 0011  CTA=ABS(CT)          00026
ISN 0012  CT1=1./CTA          00027
ISN 0013  CTAM=ABS(CTMIN)          00028
ISN 0014  CTB=ABS(CTL)          00029
ISN 0015  CT2=1./CTB          00030
ISN 0016  CTBM=ABS(CTLMIN)          00031
ISN 0017  A1=1.          00032
ISN 0018  DO 40 I=1,NAC          00033
C      CALCULATE THETA          00034
ISN 0019  NCI=IC(I)          00035
ISN 0020  NCJ=1          00036
ISN 0021  IF (NCI.LE.NCON) NCJ=ISC(NCI)          00037
ISN 0023  C1=G(NCI)          00038
ISN 0024  CTD=CT1          00039
ISN 0025  CTC=CTAM          00040
ISN 0026  IF (NCJ.LE.0) GO TO 10          00041
ISN 0028  CTC=CTEM          00042
ISN 0029  CTD=CT2          00043
ISN 0030  10 IF (C1.GT.CTC) NVC=NVC+1          00044
ISN 0032  THT=0.          00045
ISN 0033  GG=1.+CTD*C1          00046
ISN 0034  IF (NCJ.EQ.0.OR.C1.GT.CTC) THT=THETA*GG*GG          00047
ISN 0036  IF (THT.GT.50.) THT=50.          00048
ISN 0038  IF (THT.GT.THMAX) THMAX=THT          00049
ISN 0040  A(NDV1,I)=THT          00050
C      -----00051
C      NORMALIZE GRADIENTS OF CONSTRAINTS          00052
C      -----00053
ISN 0041  A(NDV2,I)=1.          00054

```

```

+VERSION 1.3.0 (01 MAY 80)   CNMN05   SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)   DATE 82.141/10.53.29   PAGE 2
  ISN 0042      IF (NCI.GT.NCON) GO TO 40      00055
  ISN 0044      A1=0.      00056
  ISN 0045      DO 20 J=1,NDV      00057
  ISN 0046      A1=A1+A(J,I)**2      00058
  ISN 0047      20 CONTINUE      00059
  ISN 0048      IF (A1.LT.1.0E-20) A1=1.0E-20      00060
  ISN 0050      A1=SQRT(A1)      00061
  ISN 0051      A(NDV2,I)=A1      00062
  ISN 0052      A1=1./A1      00063
  ISN 0053      DO 30 J=1,NDV      00064
  ISN 0054      A(J,I)=A1*A(J,I)      00065
  ISN 0055      40 CONTINUE      00066
  C -----00067
  C NORMALIZE GRADIENT OF OBJECTIVE FUNCTION AND STORE IN NAC+1      00068
  C COLUMN OF A      00069
  C -----00070
  ISN 0056      A1=0.      00071
  ISN 0057      DO 50 I=1,NDV      00072
  ISN 0058      A1=A1+DF(I)**2      00073
  ISN 0059      50 CONTINUE      00074
  ISN 0060      IF (A1.LT.1.0E-20) A1=1.0E-20      00075
  ISN 0062      A1=SQRT(A1)      00076
  ISN 0063      A1=1./A1      00077
  ISN 0064      DO 60 I=1,NDV      00078
  ISN 0065      A(I,NAC1)=A1*DF(I)      00079
  C BUILD C VECTOR.      00080
  ISN 0066      IF (NVC.GT.0) GO TO 80      00081
  C -----00082
  C BUILD C FOR CLASSICAL METHOD      00083
  C -----00084
  ISN 0068      NDB=NAC1      00085
  ISN 0069      A(NDV1,NDB)=1.      00086
  ISN 0070      DO 70 I=1,NDB      00087
  ISN 0071      70 C(I)=-A(NDV1,I)      00088
  ISN 0072      GO TO 110      00089
  ISN 0073      80 CONTINUE      00090
  C -----00091
  C BUILD C FOR MODIFIED METHOD      00092
  C -----00093
  ISN 0074      NDB=NAC      00094
  ISN 0075      A(NDV1,NAC1)=-PHI      00095
  C -----00096
  C SCALE THETA'S SO THAT MAXIMUM THETA IS UNITY      00097
  C -----00098
  ISN 0076      IF (THMAX.GT.0.00001) THMAX=1./THMAX      00099
  ISN 0078      DO 90 I=1,NDB      00100
  ISN 0079      A(NDV1,I)=A(NDV1,I)*THMAX      00101
  ISN 0080      90 CONTINUE      00102
  ISN 0081      DO 100 I=1,NDB      00103
  ISN 0082      C(I)=0.      00104
  ISN 0083      DO 100 J=1,NDV1      00105
  ISN 0084      C(I)=C(I)+A(J,I)*A(J,NAC1)      00106
  ISN 0085      110 CONTINUE      00107
  C -----00108
  C BUILD B MATRIX      00109
  C -----00110
  ISN 0086      DO 120 I=1,NDB      00111
  ISN 0087      DO 120 J=1,NDB      00112
  ISN 0088      B(I,J)=0.      00113

```

224

```

ISN 0089      DO 120 K=1,NDV1                00114
ISN 0090      120  B(I,J)=B(I,J)-A(K,I)*A(K,J) 00115
              C -----00116
              C          SOLVE SPECIAL L. P. PROBLEM 00117
              C -----00118
ISN 0091      CALL CNMN08 (NDB,NER,C,MS1,B,N3,N4,N5) 00119
ISN 0092      IF (IPRINT.GT.1.AND.NER.GT.0) WRITE (I6I,180) 00120
              C          CALCULATE RESULTING DIRECTION VECTOR, S. 00121
ISN 0094      SLOPE=0.                        00122
              C -----00123
              C          USABLE-FEASIBLE DIRECTION 00124
              C -----00125
ISN 0095      DO 140 I=1,NDV                  00126
ISN 0096      S1=0.                          00127
ISN 0097      IF (NVC.GT.0) S1=-A(I,NAC1)    00128
ISN 0099      DO 130 J=1,NDB                 00129
ISN 0100      130  S1=S1-A(I,J)*C(J)         00130
ISN 0101      SLOPE=SLOPE+S1*DF(I)          00131
ISN 0102      140  S(I)=S1                   00132
ISN 0103      S(NDV1)=1.                    00133
ISN 0104      IF (NVC.GT.0) S(NDV1)=-A(NDV1,NAC1) 00134
ISN 0106      DO 150 J=1,NDB                 00135
ISN 0107      150  S(NDV1)=S(NDV1)-A(NDV1,J)*C(J) 00136
              C -----00137
              C          NORMALIZE S TO MAX ABS OF UNITY 00138
              C -----00139
ISN 0108      S1=0.                          00140
ISN 0109      DO 160 I=1,NDV                 00141
ISN 0110      A1=ABS(S(I))                   00142
ISN 0111      IF (A1.GT.S1) S1=A1            00143
ISN 0113      160  CONTINUE                   00144
ISN 0114      IF (S1.LT.1.0E-10) RETURN      00145
ISN 0116      S1=1./S1                       00146
ISN 0117      DO 170 I=1,NDV                 00147
ISN 0118      170  S(I)=S1*S(I)               00148
ISN 0119      SLOPE=S1*SLOPE                 00149
ISN 0120      S(NDV1)=S1*S(NDV1)            00150
ISN 0121      RETURN                          00151
              C -----00152
              C          FORMATS 00153
              C -----00154
              C 00155
              C 00156
ISN 0122      180  FORMAT (//5X,46H* * DIRECTION FINDING PROCESS DID NOT CONVERGE/5X, 00157
                  129H* * S-VECTOR MAY NOT BE VALID) 00158
ISN 0123      END                            00159
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 122, PROGRAM SIZE = 2548, SUBPROGRAM NAME =CNMN05
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****
2992K BYTES OF CORE NOT USED

```

```

C          DATA SET U477CNMNO6 AT LEVEL 002 AS OF 03/13/81
C          DATA SET U477CNMNO6 AT LEVEL 001 AS OF 02/13/81          00001
C          DATA SET 9188CNMNO6 AT LEVEL 001 AS OF 07/10/80          00002
ISN 0002   SUBROUTINE CNMNO6 (X,VLB,VUB,G,SCAL,DF,S,G1,G2,CTAM,CTBM,SLOPE,ALP00003
           1,A2,A3,A4,F1,F2,F3,CV1,CV2,CV3,CV4,ALPCA,ALPFES,ALPLN,ALPMIN,ALPNC00004
           2,ALPSAV,ALPSID,ALPTOT,ISC,N1,N2,NCAL,NVC,ICOUNT,IGOOD1,IGOOD2,IGOOD00005
           3D3,IGOOD4,IBEST,III,NLNC,JGOTO)          00006
ISN 0003   COMMON /CNMNI/ DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,ALPHAX00007
           1,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,LIHOB,ITMAX,IT00008
           2RM,ICNDIR,IGOTO,NAC,INFO,INFOG,ITER          00009
ISN 0004   COMMON /UIOS/ I5I, I6I          00010
ISN 0005   DIMENSION X(N1), VLB(N1), VUB(N1), G(N2), SCAL(N1), DF(N1), S(N1),00011
           1 G1(N2), G2(N2), ISC(N2), NCAL(2)          00012
C          ROUTINE TO SOLVE ONE-DIMENSIONAL SEARCH PROBLEM FOR CONSTRAINED 00013
C          FUNCTION MINIMIZATION.          00014
C          BY G. N. VANDERPLAATS          AUG., 1974.          00015
C          NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.          00016
C          OBJ = INITIAL AND FINAL FUNCTION VALUE.          00017
C          ALP = MOVE PARAMETER.          00018
C          SLOPE = INITIAL SLOPE.          00019
C          00020
C          ALPSID = MOVE TO SIDE CONSTRAINT.          00021
C          ALPFES = MOVE TO FEASIBLE REGION.          00022
C          ALPNC = MOVE TO NEW NON-LINEAR CONSTRAINT.          00023
C          ALPLN = MOVE TO LINEAR CONSTRAINT.          00024
C          ALPCA = MOVE TO RE-ENCOUNTER CURRENTLY ACTIVE CONSTRAINT.          00025
C          ALPMIN = MOVE TO MINIMIZE FUNCTION.          00026
C          ALPTOT = TOTAL MOVE PARAMETER.          00027
ISN 0006   ZRO=0.          00028
ISN 0007   IF (JGOTO.EQ.0) GO TO 10          00029
ISN 0009   GO TO (140,310,520),JGOTO          00030
ISN 0010   10 IF (IPRINT.GE.5) WRITE (I6I,730)          00031
ISN 0012   ALPSAV=ALP          00032
ISN 0013   ICOUNT=0          00033
ISN 0014   ALPTOT=0.          00034
C          TOLERANCES.          00035
ISN 0015   CTAM=ABS(CTMIN)          00036
ISN 0016   CTBM=ABS(CTLMIN)          00037
C          PROPOSED MOVE.          00038
ISN 0017   20 CONTINUE          00039
C          -----00040
C          ***** BEGIN SEARCH OR IMPOSE SIDE CONSTRAINT MODIFICATION ***** 00041
C          -----00042
ISN 0018   A2=ALPSAV          00043
ISN 0019   ICOUNT=ICOUNT+1          00044
ISN 0020   ALPSID=1.0E+20          00045
C          INITIAL ALPHA AND OBJ.          00046
ISN 0021   ALP=0.          00047
ISN 0022   F1=OBJ          00048
ISN 0023   KSID=0          00049
ISN 0024   IF (NSIDE.EQ.0) GO TO 70          00050
C          -----00051
C          FIND MOVE TO SIDE CONSTRAINT AND INSURE AGAINST VIOLATION OF 00052
C          SIDE CONSTRAINTS          00053
C          -----00054

```

226

```

ISN 0026      DO 60 I=1,NDV                00055
ISN 0027      SI=S(I)                      00056
ISN 0028      IF (ABS(SI).GT.1.0E-20) GO TO 30 00057
C             ITH COMPONENT OF S IS SMALL. SET TO ZERO. 00058
ISN 0030      S(I)=0.                      00059
ISN 0031      SLOPE=SLOPE-SI*DF(I)         00060
ISN 0032      GO TO 60                     00061
ISN 0033      30 CONTINUE                   00062
ISN 0034      XI=X(I)                      00063
ISN 0035      SI=1./SI                     00064
ISN 0036      IF (SI.GT.0.) GO TO 40       00065
C             LOWER BOUND.                 00066
ISN 0038      XI2=VLB(I)                   00067
ISN 0039      XII=ABS(XI2)                 00068
ISN 0040      IF (XII.LT.1.) XII=1.        00069
C             CONSTRAINT VALUE.            00070
ISN 0042      GI=(XI2-XI)/XII              00071
ISN 0043      IF (GI.GT.-1.0E-6) GO TO 50  00072
C             PROPOSED MOVE TO LOWER BOUND. 00073
ISN 0045      ALPA=(XI2-XI)*SI             00074
ISN 0046      IF (ALPA.LT.ALPSID) ALPSID=ALPA 00075
ISN 0048      GO TO 60                     00076
ISN 0049      40 CONTINUE                   00077
C             UPPER BOUND.                 00078
ISN 0050      XI2=VUB(I)                   00079
ISN 0051      XII=ABS(XI2)                 00080
ISN 0052      IF (XII.LT.1.) XII=1.        00081
C             CONSTRAINT VALUE.            00082
ISN 0054      GI=(XI-XI2)/XII              00083
ISN 0055      IF (GI.GT.-1.0E-6) GO TO 50  00084
C             PROPOSED MOVE TO UPPER BOUND. 00085
ISN 0057      ALPA=(XI2-XI)*SI             00086
ISN 0058      IF (ALPA.LT.ALPSID) ALPSID=ALPA 00087
ISN 0060      GO TO 60                     00088
ISN 0061      50 CONTINUE                   00089
C             MOVE WILL VIOLATE SIDE CONSTRAINT. SET S(I)=0. 00090
ISN 0062      SLOPE=SLOPE-S(I)*DF(I)      00091
ISN 0063      S(I)=0.                      00092
ISN 0064      KSID=KSID+1                   00093
ISN 0065      60 CONTINUE                   00094
C             ALPSID IS UPPER BOUND ON ALPHA. 00095
ISN 0066      IF (A2.GT.ALPSID) A2=ALPSID  00096
ISN 0068      70 CONTINUE                   00097
C             -----00098
C             CHECK ILL-CONDITIONING        00099
C             -----00100
ISN 0069      IF (KSID.EQ.NDV.OR.ICOUNT.GT.10) GO TO 710 00101
ISN 0071      IF (NVC.EQ.0.AND.SLOPE.GT.0.) GO TO 710 00102
ISN 0073      ALPFES=-1.                    00103
ISN 0074      ALPMIN=-1.                    00104
ISN 0075      ALPLN=1.1*ALPSID              00105
ISN 0076      ALPNC=ALPSID                  00106
ISN 0077      ALPCA=ALPSID                  00107
ISN 0078      IF (NCON.EQ.0) GO TO 90      00108
C             STORE CONSTRAINT VALUES IN G1. 00109
ISN 0080      DO 80 I=1,NCON                00110
ISN 0081      G1(I)=G(I)                    00111
ISN 0082      80 CONTINUE                   00112
ISN 0083      90 CONTINUE                   00113

```

```

C -----00114
C           MOVE A DISTANCE A2*S           00115
C -----00116
ISN 0084      ALPTOT=ALPTOT+A2           00117
ISN 0085      DO 100 I=1,NDV             00118
ISN 0086      X(I)=X(I)+A2*S(I)         00119
ISN 0087      100 CONTINUE               00120
ISN 0088      IF (IPRINT.LT.5) GO TO 130 00121
ISN 0090      WRITE (I6I,740) A2        00122
ISN 0091      IF (NSCAL.EQ.0) GO TO 120  00123
ISN 0093      DO 110 I=1,NDV             00124
ISN 0094      110 G(I)=SCAL(I)*X(I)     00125
ISN 0095      WRITE (I6I,750) (G(I),I=1,NDV) 00126
ISN 0096      GO TO 130                  00127
ISN 0097      120 WRITE (I6I,750) (X(I),I=1,NDV) 00128
C -----00129
C           UPDATE FUNCTION AND CONSTRAINT VALUES 00130
C -----00131
ISN 0098      130 NCAL(1)=NCAL(1)+1     00132
ISN 0099      JGOTO=1                    00133
ISN 0100      RETURN                      00134
ISN 0101      140 CONTINUE                00135
ISN 0102      F2=OBJ                      00136
ISN 0103      IF (IPRINT.GE.5) WRITE (I6I,760) F2 00137
ISN 0105      IF (IPRINT.LT.5.OR.NCON.EQ.0) GO TO 150 00138
ISN 0107      WRITE (I6I,770)            00139
ISN 0108      WRITE (I6I,750) (G(I),I=1,NCON) 00140
ISN 0109      150 CONTINUE                00141
C -----00142
C           IDENTIFY ACCAPTABILITY OF DESIGNS F1 AND F2 00143
C -----00144
C           IGOOD = 0 IS ACCAPTABLE.         00145
C           CV = MAXIMUM CONSTRAINT VIOLATION. 00146
ISN 0110      IGOOD1=0                    00147
ISN 0111      IGOOD2=0                    00148
ISN 0112      CV1=0.                      00149
ISN 0113      CV2=0.                      00150
ISN 0114      NVC1=0                      00151
ISN 0115      IF (NCON.EQ.0) GO TO 170    00152
ISN 0117      DO 160 I=1,NCON             00153
ISN 0118      CC=CTAM                     00154
ISN 0119      IF (ISC(I).GT.0) CC=CTBM    00155
ISN 0121      C1=G1(I)-CC                 00156
ISN 0122      C2=G(I)-CC                 00157
ISN 0123      IF (C2.GT.0.) NVC1=NVC1+1  00158
ISN 0125      IF (C1.GT.CV1) CV1=C1       00159
ISN 0127      IF (C2.GT.CV2) CV2=C2       00160
ISN 0129      160 CONTINUE                 00161
ISN 0130      IF (CV1.GT.0.) IGOOD1=1    00162
ISN 0132      IF (CV2.GT.0.) IGOOD2=1    00163
ISN 0134      170 CONTINUE                 00164
ISN 0135      ALP=A2                      00165
ISN 0136      OBJ=F2                      00166
C -----00167
C           IF F2 VIOLATES FEWER CONSTRAINTS THAN F1 BUT STILL HAS CONSTRAINT 00168
C           VIOLATIONS RETURN              00169
C -----00170
ISN 0137      IF (NVC1.LT.NVC.AND.NVC1.GT.0) GO TO 710 00171
C -----00172

```

227

228

```

C IDENTIFY BEST OF DESIGNS F1 ANF F2 00173
C -----00174
C IBEST CORRESPONDS TO MINIMUM VALUE DESIGN. 00175
C IF CONSTRAINTS ARE VIOLATED, IBEST CORRESPONDS TO MINIMUM 00176
C CONSTRAINT VIOLATION. 00177
ISN 0139 IF (IGOOD1.EQ.0.AND.IGOOD2.EQ.0) GO TO 180 00178
C VIOLATED CONSTRAINTS. PICK MINIMUM VIOLATION. 00179
ISN 0141 IBEST=1 00180
ISN 0142 IF (CV1.GE.CV2) IBEST=2 00181
ISN 0144 GO TO 190 00182
ISN 0145 180 CONTINUE 00183
C NO CONSTRAINT VIOLATION. PICK MINIMUM F. 00184
ISN 0146 IBEST=1 00185
ISN 0147 IF (F2.LE.F1) IBEST=2 00186
ISN 0149 190 CONTINUE 00187
ISN 0150 II=1 00188
ISN 0151 IF (NCON.EQ.0) GO TO 230 00189
C -----00190
C ***** 2 - POINT INTERPOLATION *****00191
C -----00192
ISN 0153 III=0 00193
ISN 0154 200 III=III+1 00194
ISN 0155 C1=G(III) 00195
ISN 0156 C2=G(III) 00196
ISN 0157 IF (ISC(III).EQ.0) GO TO 210 00197
C -----00198
C LINEAR CONSTRAINT 00199
C -----00200
ISN 0159 IF (C1.GE.1.0E-5.AND.C1.LE.CTBM) GO TO 220 00201
ISN 0161 CALL CNMNO7 (II,ALP,ZRO,ZRO,C1,A2,C2,ZRO,ZRO) 00202
ISN 0162 IF (ALP.LE.0.) GO TO 220 00203
ISN 0164 IF (C1.GT.CTBM.AND.ALP.GT.ALPFES) ALPFES=ALP 00204
ISN 0166 IF (C1.LT.CTL.AND.ALP.LT.ALPLN) ALPLN=ALP 00205
ISN 0168 GO TO 220 00206
ISN 0169 210 CONTINUE 00207
C -----00208
C NON-LINEAR CONSTRAINT 00209
C -----00210
ISN 0170 IF (C1.GE.1.0E-5.AND.C1.LE.CTAM) GO TO 220 00211
ISN 0172 CALL CNMNO7 (II,ALP,ZRO,ZRO,C1,A2,C2,ZRO,ZRO) 00212
ISN 0173 IF (ALP.LE.0.) GO TO 220 00213
ISN 0175 IF (C1.GT.CTAM.AND.ALP.GT.ALPFES) ALPFES=ALP 00214
ISN 0177 IF (C1.LT.CT.AND.ALP.LT.ALPMC) ALPMC=ALP 00215
ISN 0179 220 CONTINUE 00216
ISN 0180 IF (III.LT.NCON) GO TO 200 00217
ISN 0182 230 CONTINUE 00218
ISN 0183 IF (LINOBJ.GT.0.OR.SLOPE.GE.0.) GO TO 240 00219
C CALCULATE ALPHA TO MINIMIZE FUNCTION. 00220
ISN 0185 CALL CNMNO4 (II,ALPMIN,ZRO,ZRO,F1,SLOPE,A2,F2,ZRO,ZRO,ZRO) 00221
ISN 0186 240 CONTINUE 00222
C -----00223
C PROPOSED MOVE 00224
C -----00225
C MOVE AT LEAST FAR ENOUGH TO OVERCOME CONSTRAINT VIOLATIONS. 00226
ISN 0187 A3=ALPFES 00227
C MOVE TO MINIMIZE FUNCTION. 00228
ISN 0188 IF (ALPMIN.GT.A3) A3=ALPMIN 00229
C IF A3.LE.0, SET A3 = ALPSID. 00230
ISN 0190 IF (A3.LE.0.) A3=ALPSID 00231

```

```

+VERSION 1.3.0 (01 MAY 80)  CNMN06  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 82.141/10.53.39  PAGE 5
C      LIMIT MOVE TO NEW CONSTRAINT ENCOUNTER. 00232
ISN 0192      IF (A3.GT.ALPMC) A3=ALPMC 00233
ISN 0194      IF (A3.GT.ALPLN) A3=ALPLN 00234
C      MAKE A3 NON-ZERO. 00235
ISN 0196      IF (A3.LE.1.0E-20) A3=1.0E-20 00236
C      IF A3=A2=ALPSID AND F2 IS BEST, GO INVOKE SIDE CONSTRAINT 00237
C      MODIFICATION. 00238
ISN 0198      ALPB=1.-A2/A3 00239
ISN 0199      ALPA=1.-ALPSID/A3 00240
ISN 0200      JBEST=0 00241
ISN 0201      IF (ABS(ALPB).LT.1.0E-10.AND.ABS(ALPA).LT.1.0E-10) JBEST=1 00242
ISN 0203      IF (JBEST.EQ.1.AND.IBEST.EQ.2) GO TO 20 00243
C      SIDE CONSTRAINT CHECK NOT SATISFIED. 00244
ISN 0205      IF (NCON.EQ.0) GO TO 260 00245
C      STORE CONSTRAINT VALUES IN G2. 00246
ISN 0207      DO 250 I=1,NCON 00247
ISN 0208      G2(I)=G(I) 00248
ISN 0209      250 CONTINUE 00249
ISN 0210      260 CONTINUE 00250
C      IF A3=A2, SET A3=.9*A2. 00251
ISN 0211      IF (ABS(ALPB).LT.1.0E-10) A3=.9*A2 00252
C      MOVE AT LEAST .01*A2. 00253
ISN 0213      IF (A3.LT.(.01*A2)) A3=.01*A2 00254
C      LIMIT MOVE TO 5.*A2. 00255
ISN 0215      IF (A3.GT.(5.*A2)) A3=5.*A2 00256
C      LIMIT MOVE TO ALPSID. 00257
ISN 0217      IF (A3.GT.ALPSID) A3=ALPSID 00258
C      MOVE A DISTANCE A3*S. 00259
ISN 0219      ALP=A3-A2 00260
ISN 0220      ALPTOT=ALPTOT+ALP 00261
ISN 0221      DO 270 I=1,NDV 00262
ISN 0222      X(I)=X(I)+ALP*S(I) 00263
ISN 0223      270 CONTINUE 00264
ISN 0224      IF (IPRINT.LT.5) GO TO 300 00265
ISN 0226      WRITE (I6I,780) 00266
ISN 0227      WRITE (I6I,740) A3 00267
ISN 0228      IF (HSCAL.EQ.0) GO TO 290 00268
ISN 0230      DO 280 I=1,NDV 00269
ISN 0231      280 G(I)=SCAL(I)*X(I) 00270
ISN 0232      WRITE (I6I,750) (G(I),I=1,NDV) 00271
ISN 0233      GO TO 300 00272
ISN 0234      290 WRITE (I6I,750) (X(I),I=1,NDV) 00273
ISN 0235      300 CONTINUE 00274
C      -----00275
C      UPDATE FUNCTION AND CONSTRAINT VALUES 00276
C      -----00277
ISN 0236      NCAL(1)=NCAL(1)+1 00278
ISN 0237      JGOTO=2 00279
ISN 0238      RETURN 00280
ISN 0239      310 CONTINUE 00281
ISN 0240      F3=OBJ 00282
ISN 0241      IF (IPRINT.GE.5) WRITE (I6I,760) F3 00283
ISN 0243      IF (IPRINT.LT.5.OR.NCON.EQ.0) GO TO 320 00284
ISN 0245      WRITE (I6I,770) 00285
ISN 0246      WRITE (I6I,750) (G(I),I=1,NCON) 00286
ISN 0247      320 CONTINUE 00287
C      -----00288
C      CALCULATE MAXIMUM CONSTRAINT VIOLATION AND PICK BEST DESIGN 00289
C      -----00290

```

```

+VERSION 1.3.0 (01 MAY 80)  CNMN06  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 82.141/10.53.39  PAGE 6
  ISN 0248      CV3=0. 00291
  ISN 0249      IGOOD3=0 00292
  ISN 0250      NVC1=0 00293
  ISN 0251      IF (NCON.EQ.0) GO TO 340 00294
  ISN 0253      DO 330 I=1,NCON 00295
  ISN 0254      CC=CTAM 00296
  ISN 0255      IF (ISC(I).GT.0) CC=CTEM 00297
  ISN 0257      C1=G(I)-CC 00298
  ISN 0258      IF (C1.GT.CV3) CV3=C1 00299
  ISN 0260      IF (C1.GT.0.) NVC1=NVC1+1 00300
  ISN 0262      330 CONTINUE 00301
  ISN 0263      IF (CV3.GT.0.) IGOOD3=1 00302
  ISN 0265      340 CONTINUE 00303
  C DETERMINE BEST DESIGN. 00304
  ISN 0266      IF (IBEST.EQ.2) GO TO 360 00305
  C CHOOSE BETWEEN F1 AND F3. 00306
  ISN 0268      IF (IGOOD1.EQ.0.AND.IGOOD3.EQ.0) GO TO 350 00307
  ISN 0270      IF (CV1.GE.CV3) IBEST=3 00308
  ISN 0272      GO TO 380 00309
  ISN 0273      350 IF (F3.LE.F1) IBEST=3 00310
  ISN 0275      GO TO 380 00311
  ISN 0276      360 CONTINUE 00312
  C CHOOSE BETWEEN F2 AND F3. 00313
  ISN 0277      IF (IGOOD2.EQ.0.AND.IGOOD3.EQ.0) GO TO 370 00314
  ISN 0279      IF (CV2.GE.CV3) IBEST=3 00315
  ISN 0281      GO TO 380 00316
  ISN 0282      370 IF (F3.LE.F2) IBEST=3 00317
  ISN 0284      380 CONTINUE 00318
  ISN 0285      ALP=A3 00319
  ISN 0286      OBJ=F3 00320
  C IF F3 VIOLATES FEWER CONSTRAINTS THAN F1 RETURN. 00321
  ISN 0287      IF (NVC1.LT.NVC) GO TO 710 00322
  C IF OBJECTIVE AND ALL CONSTRAINTS ARE LINEAR, RETURN. 00323
  ISN 0289      IF (LINOBJ.NE.0.AND.NLNC.EQ.NCON) GO TO 710 00324
  C IF A3 = ALPLN AND F3 IS BOTH GOOD AND BEST RETURN. 00325
  ISN 0291      ALPB=1.-ALPLN/A3 00326
  ISN 0292      IF ((ABS(ALPB).LT.1.0E-20.AND.IBEST.EQ.3).AND.(IGOOD3.EQ.0)) GO TO 00327
  1 710 00328
  C IF A3 = ALPSID AND F3 IS BEST, GO INVOKE SIDE CONSTRAINT 00329
  C MODIFICATION. 00330
  ISN 0294      ALPA=1.-ALPSID/A3 00331
  ISN 0295      IF (ABS(ALPA).LT.1.0E-20.AND.IBEST.EQ.3) GO TO 20 00332
  C -----00333
  C ***** 3 - POINT INTERPOLATION *****00334
  C -----00335
  ISN 0297      ALPNC=ALPSID 00336
  ISN 0298      ALPCA=ALPSID 00337
  ISN 0299      ALPFES=-1. 00338
  ISN 0300      ALPMIN=-1. 00339
  ISN 0301      IF (NCON.EQ.0) GO TO 440 00340
  ISN 0303      III=0 00341
  ISN 0304      390 III=III+1 00342
  ISN 0305      C1=G1(III) 00343
  ISN 0306      C2=G2(III) 00344
  ISN 0307      C3=G(III) 00345
  ISN 0308      IF (ISC(III).EQ.0) GO TO 400 00346
  C -----00347
  C LINEAR CONSTRAINT. FIND ALPFES ONLY. ALPLN SAME AS BEFORE. 00348
  C -----00349

```

230

I

II

III

```

+VERSION 1.3.0 (01 MAY 80)  CNMN06  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 82.141/10.53.39  PAGE 7
ISN 0310      IF (C1.LE.CTBM) GO TO 430                                00350
ISN 0312      II=1                                                    00351
ISN 0313      CALL CNMN07 (II,ALP,ZRO,ZRO,C1,A3,C3,ZRO,ZRO)          00352
ISN 0314      IF (ALP.GT.ALPFES) ALPFES=ALP                          00353
ISN 0316      GO TO 430                                              00354
ISN 0317      400 CONTINUE                                           00355
C -----00356
C              NON-LINEAR CONSTRAINT                                00357
C -----00358
ISN 0318      II=2                                                    00359
ISN 0319      CALL CNMN07 (II,ALP,ZRO,ZRO,C1,A2,C2,A3,C3)          00360
ISN 0320      IF (ALP.LE.ZRO) GO TO 430                              00361
ISN 0322      IF (C1.GE.CT.AND.C1.LE.0.) GO TO 410                  00362
ISN 0324      IF (C1.GT.CTAM.OR.C1.LT.0.) GO TO 420                 00363
C ALP IS MINIMUM MOVE.  UPDATE FOR NEXT CONSTRAINT ENCOUNTER.      00364
ISN 0326      410 ALPA=ALP                                           00365
ISN 0327      CALL CNMN07 (II,ALP,ALPA,ZRO,C1,A2,C2,A3,C3)          00366
ISN 0328      IF (ALP.LT.ALPCA.AND.ALPA.GE.ALPA) ALPCA=ALP          00367
ISN 0330      GO TO 430                                              00368
ISN 0331      420 CONTINUE                                           00369
ISN 0332      IF (ALP.GT.ALPFES.AND.C1.GT.CTAM) ALPFES=ALP          00370
ISN 0334      IF (ALP.LT.ALPCN.AND.C1.LT.0.) ALPCN=ALP              00371
ISN 0336      430 CONTINUE                                           00372
ISN 0337      IF (III.LT.NCON) GO TO 390                             00373
ISN 0339      440 CONTINUE                                           00374
ISN 0340      IF (LINOBJ.GT.0.OR.SLOPE.GT.0.) GO TO 450             00375
C -----00376
C              CALCULATE ALPHA TO MINIMIZE FUNCTION                  00377
C -----00378
ISN 0342      II=3                                                    00379
ISN 0343      IF (A2.GT.A3.AND.(IGOOD2.EQ.0.AND.IBEST.EQ.2)) II=2    00380
ISN 0345      CALL CNMN04 (II,ALPMIN,ZRO,ZRO,F1,SLOPE,A2,F2,A3,F3,ZRO,ZRO) 00381
ISN 0346      450 CONTINUE                                           00382
C -----00383
C              PROPOSED MOVE                                        00384
C -----00385
C MOVE AT LEAST ENOUGH TO OVERCOME CONSTRAINT VIOLATIONS.          00386
ISN 0347      A4=ALPFES                                              00387
C MOVE TO MINIMIZE FUNCTION.                                         00388
ISN 0348      IF (ALPMIN.GT.A4) A4=ALPMIN                             00389
C IF A4.LE.0, SET A4 = ALPSID.                                       00390
ISN 0350      IF (A4.LE.0.) A4=ALPSID                                  00391
C LIMIT MOVE TO NEW CONSTRAINT ENCOUNTER.                              00392
ISN 0352      IF (A4.GT.ALPLN) A4=ALPLN                               00393
ISN 0354      IF (A4.GT.ALPCN) A4=ALPCN                               00394
C LIMIT MOVE TO RE-ENCOUNTER CURRENTLY ACTIVE CONSTRAINT.          00395
ISN 0356      IF (A4.GT.ALPCA) A4=ALPCA                               00396
C LIMIT A4 TO 5.*A3.                                                  00397
ISN 0358      IF (A4.GT.(5.*A3)) A4=5.*A3                            00398
C UPDATE DESIGN.                                                      00399
ISN 0360      IF (IBEST.NE.3.OR.NCON.EQ.0) GO TO 470                 00400
C STORE CONSTRAINT VALUES IN G2.  F3 IS BEST.  F2 IS NOT.          00401
ISN 0362      DO 460 I=1,NCON                                         00402
ISN 0363      G2(I)=G(I)                                              00403
ISN 0364      460 CONTINUE                                           00404
ISN 0365      470 CONTINUE                                           00405
C IF A4=A3 AND IGOOD1=0 AND IGOOD3=1, SET A4=.9*A3.                 00406
ISN 0366      ALP=A4-A3                                              00407
ISN 0367      IF ((IGOOD1.EQ.0.AND.IGOOD3.EQ.1).AND.ABS(ALP).LT.1.0E-20) A4=.9*A4+00408

```

232

```

13
C -----00409
C MOVE A DISTANCE A4*S -----00410
C -----00411
C -----00412
ISN 0369 ALP=A4-A3 00413
ISN 0370 ALPTOT=ALPTOT+ALP 00414
ISN 0371 DO 480 I=1,NDV 00415
ISN 0372 X(I)=X(I)+ALP*S(I) 00416
ISN 0373 480 CONTINUE 00417
ISN 0374 IF (IPRINT.LT.5) GO TO 510 00418
ISN 0376 WRITE (I6I,720) 00419
ISN 0377 WRITE (I6I,740) A4 00420
ISN 0378 IF (NSCAL.EQ.0) GO TO 500 00421
ISN 0380 DO 490 I=1,NDV 00422
ISN 0381 490 G(I)=SCAL(I)*X(I) 00423
ISN 0382 WRITE (I6I,750) (G(I),I=1,NDV) 00424
ISN 0383 GO TO 510 00425
ISN 0384 500 WRITE (I6I,750) (X(I),I=1,NDV) 00426
ISN 0385 510 CONTINUE 00427
C -----00428
C UPDATE FUNCTION AND CONSTRAINT VALUES 00429
C -----00430
ISN 0386 NCAL(1)=NCAL(1)+1 00431
ISN 0387 JGOTO=3 00432
ISN 0388 RETURN 00433
ISN 0389 520 CONTINUE 00434
ISN 0390 F4=OBJ 00435
ISN 0391 IF (IPRINT.GE.5) WRITE (I6I,760) F4 00436
ISN 0393 IF (IPRINT.LT.5.OR.NCON.EQ.0) GO TO 530 00437
ISN 0395 WRITE (I6I,770) 00438
ISN 0396 WRITE (I6I,750) (G(I),I=1,NCON) 00439
ISN 0397 530 CONTINUE 00440
C DETERMINE ACCAPTABILITY OF F4. 00441
ISN 0398 IGOOD4=0 00442
ISN 0399 CV4=0. 00443
ISN 0400 IF (NCON.EQ.0) GO TO 550 00444
ISN 0402 DO 540 I=1,NCON 00445
ISN 0403 CC=CTAM 00446
ISN 0404 IF (ISC(I).GT.0) CC=CTBM 00447
ISN 0406 C1=G(I)-CC 00448
ISN 0407 IF (C1.GT.CV4) CV4=C1 00449
ISN 0409 540 CONTINUE 00450
ISN 0410 IF (CV4.GT.0.) IGOOD4=1 00451
ISN 0412 550 CONTINUE 00452
ISN 0413 ALP=A4 00453
ISN 0414 OBJ=F4 00454
C -----00455
C DETERMINE BEST DESIGN 00456
C -----00457
ISN 0415 GO TO (560,610,660),IBEST 00458
ISN 0416 560 CONTINUE 00459
C CHOOSE BETWEEN F1 AND F4. 00460
ISN 0417 IF (IGOOD1.EQ.0.AND.IGOOD4.EQ.0) GO TO 570 00461
ISN 0419 IF (CV1.GT.CV4) GO TO 710 00462
ISN 0421 GO TO 580 00463
ISN 0422 570 CONTINUE 00464
ISN 0423 IF (F4.LE.F1) GO TO 710 00465
ISN 0425 580 CONTINUE 00466
C F1 IS BEST. 00467

```

```

+VERSION 1.3.0 (01 MAY 80)  CNMN06  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 82.141/10.53.39  PAGE 9
ISN 0426      ALPTOT=ALPTOT-A4      00468
ISN 0427      OBJ=F1      00469
ISN 0428      DO 590 I=1,NDV      00470
ISN 0429      X(I)=X(I)-A4*S(I)      00471
ISN 0430      590 CONTINUE      00472
ISN 0431      IF (NCON.EQ.0) GO TO 710      00473
ISN 0433      DO 600 I=1,NCON      00474
ISN 0434      G(I)=G1(I)      00475
ISN 0435      600 CONTINUE      00476
ISN 0436      GO TO 710      00477
ISN 0437      610 CONTINUE      00478
C      CHOOSE BETWEEN F2 AND F4.      00479
ISN 0438      IF (IGOOD2.EQ.0.AND.IGOOD4.EQ.0) GO TO 620      00480
ISN 0440      IF (CV2.GT.CV4) GO TO 710      00481
ISN 0442      GO TO 630      00482
ISN 0443      620 CONTINUE      00483
ISN 0444      IF (F4.LE.F2) GO TO 710      00484
ISN 0446      630 CONTINUE      00485
C      F2 IS BEST.      00486
ISN 0447      OBJ=F2      00487
ISN 0448      A2=A4-A2      00488
ISN 0449      ALPTOT=ALPTOT-A2      00489
ISN 0450      DO 640 I=1,NDV      00490
ISN 0451      X(I)=X(I)-A2*S(I)      00491
ISN 0452      640 CONTINUE      00492
ISN 0453      IF (NCON.EQ.0) GO TO 710      00493
ISN 0455      DO 650 I=1,NCON      00494
ISN 0456      G(I)=G2(I)      00495
ISN 0457      650 CONTINUE      00496
ISN 0458      GO TO 710      00497
ISN 0459      660 CONTINUE      00498
C      CHOOSE BETWEEN F3 AND F4.      00499
ISN 0460      IF (IGOOD3.EQ.0.AND.IGOOD4.EQ.0) GO TO 670      00500
ISN 0462      IF (CV3.GT.CV4) GO TO 710      00501
ISN 0464      GO TO 680      00502
ISN 0465      670 CONTINUE      00503
ISN 0466      IF (F4.LE.F3) GO TO 710      00504
ISN 0468      680 CONTINUE      00505
C      F3 IS BEST.      00506
ISN 0469      OBJ=F3      00507
ISN 0470      A3=A4-A3      00508
ISN 0471      ALPTOT=ALPTOT-A3      00509
ISN 0472      DO 690 I=1,NDV      00510
ISN 0473      X(I)=X(I)-A3*S(I)      00511
ISN 0474      690 CONTINUE      00512
ISN 0475      IF (NCON.EQ.0) GO TO 710      00513
ISN 0477      DO 700 I=1,NCON      00514
ISN 0478      G(I)=G2(I)      00515
ISN 0479      700 CONTINUE      00516
ISN 0480      710 CONTINUE      00517
ISN 0481      ALP=ALPTOT      00518
ISN 0482      IF (IPRINT.GE.5) WRITE (I6I,790)      00519
ISN 0484      JGOTO=0      00520
ISN 0485      RETURN      00521
C      -----00522
C      FORMATS      00523
C      -----00524
C      00525
C      00526

```

+VERSION 1.3.0 (01 MAY 80) CNMNO6 SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 82.141/10.53.39 PAGE 10

ISN 0486 720 FORMAT (/5X,25HTHREE-POINT INTERPOLATION) 00527
ISN 0487 730 FORMAT (////50H* * * CONSTRAINED ONE-DIMENSIONAL SEARCH INFORMATION * * *) 00529
ISN 0488 740 FORMAT (//5X,15HPROPOSED DESIGN/5X,7HALPHA =,E12.5/5X,8HX-VECTOR) 00530
ISN 0489 750 FORMAT (1X,8E12.4) 00531
ISN 0490 760 FORMAT (/5X,5HOBJ =,E13.5) 00532
ISN 0491 770 FORMAT (/5X,17HCONSTRAINT VALUES) 00533
ISN 0492 780 FORMAT (/5X,25HTWO-POINT INTERPOLATION) 00534
ISN 0493 790 FORMAT (/5X,35H* * * END OF ONE-DIMENSIONAL SEARCH) 00535
ISN 0494 END 00536

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NCMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
STATISTICS SOURCE STATEMENTS = 493, PROGRAM SIZE = 6746, SUBPROGRAM NAME =CNMNO6
STATISTICS NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

2924K BYTES OF CORE NOT USED

REQUESTED OPTIONS: SOURCE,NOMAP,NOXREF,NOLIST,NODECK,OPT(3),AUTOCBL(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```

C          DATA SET U477CNMN07 AT LEVEL 001 AS OF 02/13/81
C          DATA SET 9188CNMN07 AT LEVEL 001 AS OF 07/10/80          00001
ISN 0002   SUBROUTINE CNMN07 (II,XBAR,EPS,X1,Y1,X2,Y2,X3,Y3)          00002
C          ROUTINE TO FIND FIRST XBAR.GE.EPS CORRESPONDING TO A REAL ZERO 00003
C          OF A ONE-DIMENSIONAL FUNCTION BY POLYNOMIAL INTERPOLATION. 00004
C          BY G. N. VANDERPLAATS          APRIL, 1972.          00005
C          NASA-AMES RESEARCH CENTER,  MOFFETT FIELD, CALIF.          00006
C          II = CALCULATION CONTROL.          00007
C          1:  2-POINT LINEAR INTERPOLATION, GIVEN X1, Y1, X2 AND Y2. 00008
C          2:  3-POINT QUADRATIC INTERPOLATION, GIVEN X1, Y1, X2, Y2, 00009
C             X3 AND Y3.          00010
C          EPS MAY BE NEGATIVE.          00011
C          IF REQUIRED ZERO ON Y DOES NOT EXITS, OR THE FUNCTION IS      00012
C          ILL-CONDITIONED, XBAR = EPS-1.0 WILL BE RETURNED AS AN ERROR 00013
C          INDICATOR.          00014
C          IF DESIRED INTERPOLATION IS ILL-CONDITIONED, A LOWER ORDER 00015
C          INTERPOLATION, CONSISTANT WITH INPUT DATA, WILL BE ATTEMPTED AND 00016
C          II WILL BE CHANGED ACCORDINGLY.          00017
ISN 0003   XBAR1=EPS-1.          00018
ISN 0004   XBAR=XBAR1          00019
ISN 0005   JJ=0          00020
ISN 0006   X21=X2-X1          00021
ISN 0007   IF (ABS(X21).LT.1.0E-20) RETURN          00022
ISN 0009   IF (II.EQ.2) GO TO 30          00023
C          00024
ISN 0011   10  CONTINUE          00025
C          -----00026
C             II=1: 2-POINT LINEAR INTERPOLATION          00027
C          -----00028
ISN 0012   II=1          00029
ISN 0013   YY=Y1*Y2          00030
ISN 0014   IF (JJ.EQ.0.OR.YY.LT.0.) GO TO 20          00031
C          INTERPOLATE BETWEEN X2 AND X3.          00032
ISN 0016   DY=Y3-Y2          00033
ISN 0017   IF (ABS(DY).LT.1.0E-20) GO TO 20          00034
ISN 0019   XBAR=X2+Y2*(X2-X3)/DY          00035
ISN 0020   IF (XBAR.LT.EPS) XBAR=XBAR1          00036
ISN 0022   RETURN          00037
ISN 0023   20  DY=Y2-Y1          00038
C          INTERPOLATE BETWEEN X1 AND X2.          00039
ISN 0024   IF (ABS(DY).LT.1.0E-20) RETURN          00040
ISN 0026   XBAR=X1+Y1*(X1-X2)/DY          00041
ISN 0027   IF (XBAR.LT.EPS) XBAR=XBAR1          00042
ISN 0029   RETURN          00043
ISN 0030   30  CONTINUE          00044
C          -----00045
C             II=2: 3-POINT QUADRATIC INTERPOLATION          00046
C          -----00047
ISN 0031   JJ=1          00048
ISN 0032   X31=X3-X1          00049
ISN 0033   X32=X3-X2          00050
ISN 0034   QQ=X21*X31*X32          00051
ISN 0035   IF (ABS(QQ).LT.1.0E-20) RETURN          00052
ISN 0037   AA=(Y1*X32-Y2*X31+Y3*X21)/QQ          00053
ISN 0038   IF (ABS(AA).LT.1.0E-20) GO TO 10          00054
    
```

```

+VERSION 1.3.0 (01 MAY 80)  CNMNO7  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 82.141/10.53.50  PAGE 2
  ISN 0040      BB=(Y2-Y1)/X21-AA*(X1+X2)                                00055
  ISN 0041      CC=Y1-X1*(AA*X1+BB)                                       00056
  ISN 0042      BAC=BB*BB-4.*AA*CC                                         00057
  ISN 0043      IF (BAC.LT.0.) GO TO 10                                     00058
  ISN 0045      BAC=SQRT(BAC)                                              00059
  ISN 0046      AA=.5/AA                                                    00060
  ISN 0047      XBAR=AA*(BAC-BB)                                           00061
  ISN 0048      XB2=-AA*(BAC+BB)                                           00062
  ISN 0049      IF (XBAR.LT.EPS) XBAR=XB2                                   00063
  ISN 0051      IF (XB2.LT.XBAR.AND.XB2.GT.EPS) XBAR=XB2                 00064
  ISN 0053      IF (XBAR.LT.EPS) XBAR=XBAR1                                 00065
  ISN 0055      RETURN                                                       00066
  ISN 0056      END                                                         00067
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS*    SOURCE STATEMENTS =      55, PROGRAM SIZE =      936, SUBPROGRAM NAME =CNMNO7
*STATISTICS*    NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****
                                     3016K BYTES OF CORE NOT USED

```

236

REQUESTED OPTIONS: SOURCE,NOMAP,NOXREF,NOLIST,NODECK,OPT(3),AUTODBL(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTHT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```

C          DATA SET U477CHMN08 AT LEVEL 001 AS OF 02/13/81
C          DATA SET 9188CNMN08 AT LEVEL 001 AS OF 07/10/80          00001
ISN 0002  SUBROUTINE CNMN08 (NDB,NER,C,MS1,B,N3,N4,N5)                00002
ISN 0003  DIMENSION C(N4), B(N3,N3), MS1(N5)                        00003
C          ROUTINE TO SOLVE SPECIAL LINEAR PROBLEM FOR IMPOSING S-TRANSPOSE 00004
C          TIMES S.LE.1 BOUNDS IN THE MODIFIED METHOD OF FEASIBLE DIRECTIONS.00005
C          BY G. N. VANDERPLAATS          APRIL, 1972.              00006
C          NASA-AMES RESEARCH CENTER,  MOFFETT FIELD, CALIF.        00007
C          REF. 'STRUCTURAL OPTIMIZATION BY METHODS OF FEASIBLE DIRECTIONS',00008
C          G. N. VANDERPLAATS AND F. MOSES, JOURNAL OF COMPUTERS     00009
C          AND STRUCTURES, VOL 3, PP 739-755, 1973.                 00010
C          FORM OF L. P. IS BX=C WHERE 1ST NDB COMPONENTS OF X CONTAIN VECTOR00011
C          U AND LAST NDB COMPONENTS CONTAIN VECTOR V.  CONSTRAINTS ARE 00012
C          U.GE.0, V.GE.0, AND U-TRANSPOSE TIMES V = 0.             00013
C          NER = ERROR FLAG.  IF NER.NE.0 ON RETURN, PROCESS HAS NOT 00014
C          CONVERGED IN 5*NDB ITERATIONS.                            00015
C          VECTOR MS1 IDENTIFIES THE SET OF BASIC VARIABLES.        00016
C          -----                                                00017
C          CHOOSE INITIAL BASIC VARIABLES AS V, AND INITIALIZE VECTOR MS1 00018
C          -----                                                00019
ISN 0004  NER=1                                                    00020
ISN 0005  M2=2*NDB                                                00021
C          CALCULATE CBMIN AND EPS AND INITIALIZE MS1.              00022
ISN 0006  EPS=-1.0E+10                                           00023
ISN 0007  CBMIN=0.                                                00024
ISN 0008  DO 10 I=1,NDB                                           00025
ISN 0009  BI=B(I,I)                                               00026
ISN 0010  CBMAX=0.                                                00027
ISN 0011  IF (BI.LT.-1.0E-6) CBMAX=C(I)/BI                        00028
ISN 0013  IF (BI.GT.EPS) EPS=BI                                    00029
ISN 0015  IF (CBMAX.GT.CBMIN) CBMIN=CBMAX                         00030
ISN 0017  10  MS1(I)=0                                             00031
ISN 0018  EPS=.0001*EPS                                           00032
ISN 0019  IF (EPS.LT.-1.0E-10) EPS=-1.0E-10                     00033
ISN 0021  IF (EPS.GT.-.0001) EPS=-.0001                          00034
ISN 0023  CBMIN=CBMIN*1.0E-6                                      00035
ISN 0024  IF (CBMIN.LT.1.0E-10) CBMIN=1.0E-10                   00036
ISN 0026  ITER1=0                                                 00037
ISN 0027  NMAX=5*NDB                                             00038
C          -----                                                00039
C          *****          BEGIN NEW ITERATION          *****00040
C          -----                                                00041
ISN 0028  20  ITER1=ITER1+1                                        00042
ISN 0029  IF (ITER1.GT.NMAX) RETURN                                00043
C          FIND MAX. C(I)/B(I,I) FOR I=1,NDB.                      00044
ISN 0031  CBMAX=.9*CBMIN                                          00045
ISN 0032  ICHK=0                                                  00046
ISN 0033  DO 30 I=1,NDB                                           00047
ISN 0034  C1=C(I)                                                 00048
ISN 0035  BI=B(I,I)                                               00049
ISN 0036  IF (BI.GT.EPS.OR.C1.GT.0.) GO TO 30                    00050
ISN 0038  CB=C1/BI                                               00051
ISN 0039  IF (CB.LE.CBMAX) GO TO 30                               00052
ISN 0041  ICHK=I                                                  00053
ISN 0042  CBMAX=CB                                               00054

```

238

```

ISN 0043 30 CONTINUE 00055
ISN 0044 IF (CBMAX.LT.CBMIN) GO TO 70 00056
ISN 0046 IF (ICLK.EQ.0) GO TO 70 00057
C UPDATE VECTOR MS1. 00058
ISN 0048 JJ=ICLK 00059
ISN 0049 IF (MS1(JJ).EQ.0) JJ=ICLK+NDB 00060
ISN 0051 KK=JJ+NDB 00061
ISN 0052 IF (KK.GT.M2) KK=JJ-NDB 00062
ISN 0054 MS1(KK)=ICLK 00063
ISN 0055 MS1(JJ)=0 00064
C -----00065
C PIVOT OF B(ICLK,ICLK) 00066
C -----00067
ISN 0056 BB=1./B(ICLK,ICLK) 00068
ISN 0057 DO 40 J=1,NDB 00069
ISN 0058 40 B(ICLK,J)=BB*B(ICLK,J) 00070
ISN 0059 C(ICLK)=CBMAX 00071
ISN 0060 B(ICLK,ICLK)=BB 00072
C ELIMINATE COEFFICIENTS ON VARIABLE ENTERING BASIS AND STORE 00073
C COEFFICIENTS ON VARIABLE LEAVING BASIS IN THEIR PLACE. 00074
ISN 0061 DO 60 I=1,NDB 00075
ISN 0062 IF (I.EQ.ICLK) GO TO 60 00076
ISN 0064 BB1=B(I,ICLK) 00077
ISN 0065 B(I,ICLK)=0. 00078
ISN 0066 DO 50 J=1,NDB 00079
ISN 0067 50 B(I,J)=B(I,J)-BB1*B(ICLK,J) 00080
ISN 0068 C(I)=C(I)-BB1*CBMAX 00081
ISN 0069 60 CONTINUE 00082
ISN 0070 GO TO 20 00083
ISN 0071 70 CONTINUE 00084
ISN 0072 NER=0 00085
C -----00086
C STORE ONLY COMPONENTS OF U-VECTOR IN 'C'. USE B(I,1) FOR 00087
C TEMPORARY STORAGE 00088
C -----00089
ISN 0073 DO 80 I=1,NDB 00090
ISN 0074 B(I,1)=C(I) 00091
ISN 0075 80 CONTINUE 00092
ISN 0076 DO 90 I=1,NDB 00093
ISN 0077 C(I)=0. 00094
ISN 0078 J=MS1(I) 00095
ISN 0079 IF (J.GT.0) C(I)=B(J,1) 00096
ISN 0081 IF (C(I).LT.0.) C(I)=0. 00097
ISN 0083 90 CONTINUE 00098
ISN 0084 RETURN 00099
ISN 0085 END 00100

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
STATISTICS SOURCE STATEMENTS = 84, PROGRAM SIZE = 1390, SUBPROGRAM NAME =CNMNO8
STATISTICS NO DIAGNOSTICS GENERATED
***** END OF CCMPILATION *****

3004K BYTES OF CORE NOT USED

REQUESTED OPTIONS: SOURCE,NOMAP,NOXREF,NOLIST,NODECK,OPT(3),AUTODBL(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```

C      DATA SET U477CONMIN AT LEVEL 003 AS OF 03/17/81
C      DATA SET U477CONMIN AT LEVEL 002 AS OF 03/13/81          00001
C      DATA SET U477CONMIN AT LEVEL 001 AS OF 02/13/81          00002
C      DATA SET 9188CONMIN AT LEVEL 001 AS OF 07/10/80          00003
ISN 0002  SUBROUTINE CONMIN (X,VLB,VUB,G,SCAL,DF,A,S,G1,G2,B,C,ISC,IC,MS1,N100004
          1,N2,N3,N4,N5)                                          00005
ISN 0003  COMMON /CNMNI/ DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,ALPHAX00006
          1,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,LINOBJ,ITMAX,IT00007
          2RM,ICNDR,IGOTO,HAC,INFO,INFOG,ITER                    00008
ISN 0004  COMMON /UIOS/ I5I, I6I                                00009
ISN 0005  DIMENSION X(N1), VLB(N1), VUB(N1), G(N2), SCAL(N1), DF(N1), A(N1,N00010
          13), S(N1), G1(N2), G2(N2), B(N3,N3), C(N4), ISC(N2), IC(N3), MS1(N00011
          25)                                                    00012
ISN 0006  COMMON /CONSAV/ DM1,DM2,DM3,DM4,DM5,DM6,DM7,DM8,DM9,DM10,DM11,DM1200013
          1,DCT,DCTL,PHI,ABOBJ,CTA,CTAM,CTBM,OBJ1,SLOPE,DX,DX1,FI,XI,DFTDF1,A00014
          2LP,FFF,A1,A2,A3,A4,F1,F2,F3,F4,CV1,CV2,CV3,CV4,APP,ALPCA,ALPFES,AL00015
          3PLN,ALPHIN,ALPNC,ALPSAV,ALPSID,ALPTOT,RSPACE,ICM1,IDM2,IDM3,JDIR,I00016
          4OBJ,KOBJ,KCOUNT,NCAL(2),NFEAS,NSCAL,NCOBJ,NVC,KOUNT,ICOUNT,IGOOD1,00017
          5IGOOD2,IGOOD3,IGOOD4,IBEST,III,NLNC,JGOTO,ISPACE(2)  00018
C      ROUTINE TO SOLVE CONSTRAINED OR UNCONSTRAINED FUNCTION  00019
C      MINIMIZATION.                                           00020
C      BY G. N. VANDERPLAATS                                APRIL, 1972.      00021
C      * * * * * JUNE, 1979 VERSION * * * * * * * * * * 00022
C      NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.      00023
C      REFERENCE; CONMIN - A FORTRAN PROGRAM FOR CONSTRAINED FUNCTION 00024
C      MINIMIZATION: USER'S MANUAL, BY G. N. VANDERPLAATS,  00025
C      NASA TM X-62,282, AUGUST, 1973.                       00026
C      STORAGE REQUIREMENTS:                                  00027
C      PROGRAM - 7000 DECIMAL WORDS (CDC COMPUTER)           00028
C      ARRAYS - APPROX. 2*(NDV**2)+26*NDV+4*NCON,             00029
C      WHERE N3 = NDV+2.                                       00030
C      RE-SCALE VARIABLES IF REQUIRED.                          00031
ISN 0007  IF (NSCAL.EQ.0.OR.IGOTO.EQ.0) GO TO 20              00032
ISN 0009  DO 10 I=1,NDV                                        00033
ISN 0010  X(I)=C(I)                                           00034
ISN 0011  20 CONTINUE                                         00035
C      CONSTANTS.                                             00036
ISN 0012  NDV1=NDV+1                                          00037
ISN 0013  NDV2=NDV+2                                          00038
ISN 0014  IF (IGOTO.EQ.0) GO TO 40                            00039
C      -----00040
C      CHECK FOR UNBOUNDED SOLUTION                            00041
C      -----00042
C      STOP IF OBJ IS LESS THAN -1.0E+40                      00043
C      IF (OBJ.GT.-1.0E+40) GO TO 30                          00044
ISN 0016  WRITE (I6I,980)                                     00045
ISN 0018  GO TO 810                                           00046
ISN 0019  CONTINUE                                           00047
ISN 0020  30 CONTINUE                                         00047
ISN 0021  GO TO (160,390,380,670,690),IGOTO                  00048
C      -----00049
C      SAVE INPUT CONTROL PARAMETERS                          00050
C      -----00051
ISN 0022  40 CONTINUE                                         00052
ISN 0023  IF (IPRINT.GT.0) WRITE (I6I,1220)                  00053
ISN 0025  IF (LINOBJ.EQ.0.OR.(NCON.GT.0.OR.NSIDE.GT.0)) GO TO 50 00054
    
```

240

```

C      TOTALLY UNCONSTRAINED FUNCTION WITH LINEAR OBJECTIVE.      00055
C      SOLUTION IS UNBOUNDED.                                       00056
ISN 0027      WRITE (I6I,970) LINOBJ,NCON,NSIDE                       00057
ISN 0028      RETURN                                               00058
ISN 0029      50 CONTINUE                                           00059
ISN 0030      IDM1=ITRM                                             00060
ISN 0031      IDH2=ITMAX                                           00061
ISN 0032      IDM3=ICNDIR                                          00062
ISN 0033      DM1=DELFUN                                           00063
ISN 0034      DM2=DABFUN                                           00064
ISN 0035      DM3=CT                                               00065
ISN 0036      DM4=CTMIN                                           00066
ISN 0037      DM5=CTL                                             00067
ISN 0038      DM6=CTLMIN                                          00068
ISN 0039      DM7=THETA                                           00069
ISN 0040      DM8=PHI                                             00070
ISN 0041      DM9=FDCH                                             00071
ISN 0042      DM10=FDCHM                                          00072
ISN 0043      DM11=ABOBJ1                                          00073
ISN 0044      DM12=ALPHAX                                          00074
-----
C      00075
C      00076
C      00077
C      00078
C      00079
C      00080
C      00081
C      00082
C      00083
C      00084
C      00085
C      00086
C      00087
C      00088
C      00089
C      00090
C      00091
C      00092
C      00093
C      00094
C      00095
C      00096
C      00097
C      00098
C      00099
C      00100
C      00101
C      00102
C      00103
C      00104
C      00105
C      00106
C      00107
C      00108
C      00109
C      00110
C      00111
C      00112
C      00113

```

```

+VERSION 1.3.0 (01 MAY 80)   CONMIN   SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)   DATE 82.141/10.53.56   PAGE 3
  ISN 0091      DCTL=(CTLMIN/ABS(CTL))*CT1                                00114
  ISN 0092      PHI=5.                                                    00115
  ISN 0093      ABOBJ=ABOBJ1                                              00116
  ISN 0094      NCOBJ=0                                                  00117
  ISN 0095      CTAM=ABS(CTMIN)                                           00118
  ISN 0096      CTBM=ABS(CTLMIN)                                          00119
  C             CALCULATE NUMBER OF LINEAR CONSTRAINTS, NLNC.            00120
  NLNC=0                                               00121
  ISN 0098      IF (NCON.EQ.0) GO TO 70                                    00122
  ISN 0100      DO 60 I=1,NCON                                             00123
  ISN 0101      IF (ISC(I).GT.0) NLNC=NLNC+1                              00124
  ISN 0103      60 CONTINUE                                               00125
  ISN 0104      70 CONTINUE                                               00126
  C             -----00127
  C             CHECK TO BE SURE THAT SIDE CONSTRAINTS ARE SATISFIED    00128
  C             -----00129
  ISN 0105      IF (NSIDE.EQ.0) GO TO 110                                  00130
  ISN 0107      DO 100 I=1,NDV                                             00131
  ISN 0108      IF (VLB(I).LE.VUB(I)) GO TO 80                            00132
  ISN 0110      XX=.5*(VLB(I)+VUB(I))                                     00133
  ISN 0111      X(I)=XX                                                  00134
  ISN 0112      VLB(I)=XX                                                00135
  ISN 0113      VUB(I)=XX                                                00136
  ISN 0114      WRITE (I6I,1120) I                                        00137
  ISN 0115      80 CONTINUE                                               00138
  ISN 0116      XX=X(I)-VLB(I)                                           00139
  ISN 0117      IF (XX.GE.0.) GO TO 90                                    00140
  C             LOWER BOUND VIOLATED.                                     00141
  ISN 0119      WRITE (I6I,1130) X(I),VLB(I),I                            00142
  ISN 0120      X(I)=VLB(I)                                               00143
  ISN 0121      GO TO 100                                                 00144
  ISN 0122      90 CONTINUE                                               00145
  ISN 0123      XX=VUB(I)-X(I)                                           00146
  ISN 0124      IF (XX.GE.0.) GO TO 100                                    00147
  ISN 0126      WRITE (I6I,1140) X(I),VUB(I),I                            00148
  ISN 0127      X(I)=VUB(I)                                               00149
  ISN 0128      100 CONTINUE                                              00150
  ISN 0129      110 CONTINUE                                              00151
  C             -----00152
  C             INITIALIZE SCALING VECTOR, SCAL                            00153
  C             -----00154
  ISN 0130      IF (NSCAL.EQ.0) GO TO 150                                  00155
  ISN 0132      IF (NSCAL.LT.0) GO TO 130                                  00156
  ISN 0134      DO 120 I=1,NDV                                             00157
  ISN 0135      120 SCAL(I)=1.                                             00158
  ISN 0136      GO TO 150                                                 00159
  ISN 0137      130 CONTINUE                                              00160
  ISN 0138      DO 140 I=1,NDV                                             00161
  ISN 0139      SI=ABS(SCAL(I))                                           00162
  ISN 0140      IF (SI.LT.1.0E-20) SI=1.0E-5                             00163
  ISN 0142      SCAL(I)=SI                                                00164
  ISN 0143      SI=1./SI                                                  00165
  ISN 0144      X(I)=X(I)*SI                                              00166
  ISN 0145      IF (NSIDE.EQ.0) GO TO 140                                  00167
  ISN 0147      VLB(I)=VLB(I)*SI                                          00168
  ISN 0148      VUB(I)=VUB(I)*SI                                          00169
  ISN 0149      140 CONTINUE                                              00170
  ISN 0150      150 CONTINUE                                              00171
  C             -----00172

```

242

```

C ***** CALCULATE INITIAL FUNCTION AND CONSTRAINT VALUES ***** 00173
C -----00174
ISN 0151 INFO=1 00175
ISN 0152 NCAL(1)=1 00176
ISN 0153 IGOTO=1 00177
ISN 0154 GO TO 950 00178
ISN 0155 160 CONTINUE 00179
ISN 0156 OBJ1=OBJ 00180
ISN 0157 IF (DABFUN.LE.0.) DABFUN=.001*ABS(OBJ) 00181
ISN 0159 IF (DABFUN.LT.1.0E-10) DABFUN=1.0E-10 00182
ISN 0161 IF (IPRINT.LE.0) GO TO 270 00183
C -----00184
C PRINT INITIAL DESIGN INFORMATION 00185
C -----00186
ISN 0163 IF (IPRINT.LE.1) GO TO 230 00187
ISN 0165 IF (NSIDE.EQ.0.AND.NCON.EQ.0) WRITE (I6I,1290) 00188
ISN 0167 IF (NSIDE.NE.0.OR.NCON.GT.0) WRITE (I6I,1230) 00189
ISN 0169 WRITE(I6I,1240)IPRINT,NDV,ITMAX,NCON,NSIDE,ICNDIR,NSCAL,NFDG,LINOB00190
I,J,ITRM,N1,N2,N3,N4,N5 00191
ISN 0170 WRITE (I6I,1260) CT,CTMIN,CTL,CTLMIN,THETA,PHI,DELFUN,DABFUN 00192
ISN 0171 WRITE (I6I,1250) FDCH,FDCHM,ALPHAX,ABOBJ1 00193
ISN 0172 IF (NSIDE.EQ.0) GO TO 190 00194
ISN 0174 WRITE (I6I,1270) 00195
ISN 0175 DO 170 I=1,NDV,6 00196
ISN 0176 M1=MIN0(NDV,I+5) 00197
ISN 0177 170 WRITE (I6I,1010) I,(VLB(J),J=I,M1) 00198
ISN 0178 WRITE (I6I,1280) 00199
ISN 0179 DO 180 I=1,NDV,6 00200
ISN 0180 M1=MIN0(NDV,I+5) 00201
ISN 0181 180 WRITE (I6I,1010) I,(VUB(J),J=I,M1) 00202
ISN 0182 190 CONTINUE 00203
ISN 0183 IF (NSCAL.GE.0) GO TO 200 00204
ISN 0185 WRITE (I6I,1300) 00205
ISN 0186 WRITE (I6I,1460) (SCAL(I),I=1,NDV) 00206
ISN 0187 200 CONTINUE 00207
ISN 0188 IF (NCON.EQ.0) GO TO 230 00208
ISN 0190 IF (NLNC.EQ.0.OR.NLNC.EQ.NCON) GO TO 220 00209
ISN 0192 WRITE (I6I,1020) 00210
ISN 0193 DO 210 I=1,NCON,15 00211
ISN 0194 M1=MIN0(NCON,I+14) 00212
ISN 0195 210 WRITE (I6I,1030) I,(ISC(J),J=I,M1) 00213
ISN 0196 GO TO 230 00214
ISN 0197 220 IF (NLNC.EQ.NCON) WRITE (I6I,1040) 00215
ISN 0199 IF (NLNC.EQ.0) WRITE (I6I,1050) 00216
ISN 0201 230 CONTINUE 00217
ISN 0202 WRITE (I6I,1440) OBJ 00218
ISN 0203 WRITE (I6I,1450) 00219
ISN 0204 DO 240 I=1,NDV 00220
ISN 0205 X1=1. 00221
ISN 0206 IF (NSCAL.NE.0) X1=SCAL(I) 00222
ISN 0208 240 G1(I)=X(I)*X1 00223
ISN 0209 DO 250 I=1,NDV,6 00224
ISN 0210 M1=MIN0(NDV,I+5) 00225
ISN 0211 250 WRITE (I6I,1010) I,(G1(J),J=I,M1) 00226
ISN 0212 IF (NCON.EQ.0) GO TO 270 00227
ISN 0214 WRITE (I6I,1470) 00228
ISN 0215 DO 260 I=1,NCON,6 00229
ISN 0216 M1=MIN0(NCON,I+5) 00230
ISN 0217 260 WRITE (I6I,1010) I,(G(J),J=I,M1) 00231

```

```

ISN 0218 270 CONTINUE 00232
ISN 0219 IF (IPRINT.GT.1) WRITE (I6I,1360) 00233
C -----00234
C ***** BEGIN MINIMIZATION *****00235
C -----00236
ISN 0221 280 CONTINUE 00237
ISN 0222 ITER=ITER+1 00238
ISN 0223 IF (ABOBJ1.LT..0001) ABOBJ1=.0001 00239
ISN 0225 IF (ABOBJ1.GT..2) ABOBJ1=,2 00240
ISN 0227 IF (ALPHAX.GT.1.) ALPHAX=1. 00241
ISN 0229 IF (ALPHAX.LT..001) ALPHAX=.001 00242
ISN 0231 IF (IPRINT.GT.2) WRITE (I6I,1310) ITER 00243
ISN 0233 IF (IPRINT.GT.3.AND.NCON.GT.0) WRITE (I6I,1320) CT,CTL,PHI 00244
ISN 0235 CTA=ABS(CT) 00245
ISN 0236 IF (NCOBJ.EQ.0) GO TO 340 00246
C -----00247
C NO MOVE ON LAST ITERATION. DELETE CONSTRAINTS THAT ARE NO 00248
C LONGER ACTIVE. 00249
C -----00250
ISN 0238 NNAC=NAC 00251
ISN 0239 DO 290 I=1,NNAC 00252
ISN 0240 IF (IC(I).GT.NCON) NAC=NAC-1 00253
ISN 0242 290 CONTINUE 00254
ISN 0243 IF (NAC.LE.0) GO TO 420 00255
ISN 0245 NNAC=NAC 00256
ISN 0246 DO 330 I=1,NNAC 00257
ISN 0247 300 NIC=IC(I) 00258
ISN 0248 CT1=CT 00259
ISN 0249 IF (ISC(NIC).GT.0) CT1=CTL 00260
ISN 0251 IF (G(NIC).GT.CT1) GO TO 330 00261
ISN 0253 NAC=NAC-1 00262
ISN 0254 IF (I.GT.NAC) GO TO 420 00263
ISN 0256 DO 320 K=I,NAC 00264
ISN 0257 II=K+1 00265
ISN 0258 DO 310 J=1,NDV2 00266
ISN 0259 310 A(J,K)=A(J,II) 00267
ISN 0260 320 IC(K)=IC(II) 00268
ISN 0261 GO TO 300 00269
ISN 0262 330 CONTINUE 00270
ISN 0263 GO TO 420 00271
ISN 0264 340 CONTINUE 00272
ISN 0265 IF (MSCAL.LT.NSCAL.OR.NSCAL.EQ.0) GO TO 360 00273
ISN 0267 IF (NSCAL.LT.0.AND.KCOUNT.LT.ICNDIR) GO TO 360 00274
ISN 0269 MSCAL=0 00275
ISN 0270 KCOUNT=0 00276
C -----00277
C SCALE VARIABLES 00278
C -----00279
ISN 0271 DO 350 I=1,NDV 00280
ISN 0272 SI=SCAL(I) 00281
ISN 0273 XI=SI*X(I) 00282
ISN 0274 SIB=SI 00283
ISN 0275 IF (NSCAL.GT.0) SI=ABS(XI) 00284
ISN 0277 IF (SI.LT.1.0E-10) GO TO 350 00285
ISN 0279 SCAL(I)=SI 00286
ISN 0280 SI=1./SI 00287
ISN 0281 X(I)=XI*SI 00288
ISN 0282 IF (NSIDE.EQ.0) GO TO 350 00289
ISN 0284 VLB(I)=SIB*SI*VLB(I) 00290

```

243

244

```

ISN 0285      VUB(I)=SIB*SI*VUB(I)                00291
ISN 0286      350  CONTINUE                        00292
ISN 0287      IF (IPRINT.LT.4.OR.(NSCAL.LT.0.AND.ITER.GT.1)) GO TO 360  00293
ISN 0289      WRITE (I6I,1330)                    00294
ISN 0290      WRITE (I6I,1460) (SCAL(I),I=1,NDV)  00295
ISN 0291      360  CONTINUE                        00296
ISN 0292      MSCAL=MSCAL+1                        00297
ISN 0293      NAC=0                                00298
C -----00299
C          OBTAIN GRADIENTS OF OBJECTIVE AND ACTIVE CONSTRAINTS  00300
C -----00301
ISN 0294      INFO=2                                00302
ISN 0295      NCAL(2)=NCAL(2)+1                    00303
ISN 0296      IF (NFDG.NE.1) GO TO 370            00304
ISN 0298      IGOTO=2                              00305
ISN 0299      GO TO 950                            00306
ISN 0300      370  CONTINUE                        00307
ISN 0301      JGOTO=0                              00308
ISN 0302      380  CONTINUE                        00309
ISN 0303      CALL CNMN01 (JGOTO,X,DF,G,ISC,IC,A,G1,VLB,VUB,SCAL,C,NCAL,DX,DX1,F00310
                I,XI,III,NI,N2,N3,N4)              00311
ISN 0304      IGOTO=3                              00312
ISN 0305      IF (JGOTO.GT.0) GO TO 950          00313
ISN 0307      390  CONTINUE                        00314
ISN 0308      INFO=1                              00315
ISN 0309      IF (NAC.GE.N3) GO TO 810            00316
ISN 0311      IF (NSCAL.EQ.0.OR.NFDG.EQ.0) GO TO 420  00317
C -----00318
C          SCALE GRADIENTS                                00319
C -----00320
C          SCALE GRADIENT OF OBJECTIVE FUNCTION.          00321
ISN 0313      DO 400 I=1,NDV                        00322
ISN 0314      400  DF(I)=DF(I)*SCAL(I)            00323
ISN 0315      IF (NFDG.EQ.2.OR.NAC.EQ.0) GO TO 420  00324
C          SCALE GRADIENTS OF ACTIVE CONSTRAINTS.        00325
ISN 0317      DO 410 J=1,NDV                        00326
ISN 0318      SCJ=SCAL(J)                          00327
ISN 0319      DO 410 I=1,NAC                       00328
ISN 0320      410  A(J,I)=A(J,I)*SCJ              00329
ISN 0321      420  CONTINUE                        00330
ISN 0322      IF (IPRINT.LT.3.OR.NCON.EQ.0) GO TO 470  00331
C -----00332
C          PRINT                                00333
C -----00334
C          PRINT ACTIVE AND VIOLATED CONSTRAINT NUMBERS.  00335
ISN 0324      M1=0                                  00336
ISN 0325      M2=N3                                 00337
ISN 0326      IF (NAC.EQ.0) GO TO 450             00338
ISN 0328      DO 440 I=1,NAC                       00339
ISN 0329      J=IC(I)                              00340
ISN 0330      IF (J.GT.NCON) GO TO 440            00341
ISN 0332      GI=G(J)                              00342
ISN 0333      C1=CTAM                               00343
ISN 0334      IF (ISC(J).GT.0) C1=CTBM            00344
ISN 0336      GI=GI-C1                             00345
ISN 0337      IF (GI.GT.0.) GO TO 430             00346
C          ACTIVE CONSTRAINT.                            00347
ISN 0339      M1=M1+1                              00348
ISN 0340      MS1(M1)=J                           00349

```

```

+VERSION 1.3.0 (01 MAY 80)   CONMIN   SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)   DATE 82.141/10.53.56   PAGE 7
ISN 0341      GO TO 440      00350
ISN 0342      430  M2=M2+1    00351
                C          VIOLATED CONSTRAINT.      00352
ISN 0343      MS1(M2)=J      00353
ISN 0344      440  CONTINUE   00354
ISN 0345      450  M3=M2-N3   00355
ISN 0346      WRITE (I6I,1060) M1 00356
ISN 0347      IF (M1.EQ.0) GO TO 460 00357
ISN 0349      WRITE (I6I,1070)      00358
ISN 0350      WRITE (I6I,1480) (MS1(I),I=1,M1) 00359
ISN 0351      460  WRITE (I6I,1080) M3 00360
ISN 0352      IF (M3.EQ.0) GO TO 470 00361
ISN 0354      WRITE (I6I,1070)      00362
ISN 0355      M3=N3+1          00363
ISN 0356      WRITE (I6I,1480) (MS1(I),I=M3,M2) 00364
ISN 0357      470  CONTINUE   00365
                C          -----00366
                C          CALCULATE GRADIENTS OF ACTIVE SIDE CONSTRAINTS 00367
                C          -----00368
ISN 0358      IF (NSIDE.EQ.0) GO TO 530 00369
ISN 0360      MCN1=NCON        00370
ISN 0361      M1=0            00371
ISN 0362      DO 510 I=1,NDV   00372
                C          LOWER BOUND.              00373
ISN 0363      XI=X(I)         00374
ISN 0364      XID=VLB(I)      00375
ISN 0365      X12=ABS(XID)    00376
ISN 0366      IF (X12.LT.1.) X12=1. 00377
ISN 0368      GI=(XID-XI)/X12 00378
ISN 0369      IF (GI.LT.-1.0E-6) GO TO 490 00379
ISN 0371      M1=M1+1        00380
ISN 0372      MS1(M1)=-I     00381
ISN 0373      NAC=NAC+1      00382
ISN 0374      IF (NAC.GE.N3) GO TO 810 00383
ISN 0376      MCH1=MCN1+1    00384
ISN 0377      DO 480 J=1,NDV  00385
ISN 0378      480  A(J,NAC)=0. 00386
ISN 0379      A(I,NAC)=-1.   00387
ISN 0380      IC(NAC)=MCN1   00388
ISN 0381      G(MCN1)=GI     00389
ISN 0382      ISC(MCN1)=1    00390
                C          UPPER BOUND.              00391
ISN 0383      490  XID=VUB(I) 00392
ISN 0384      X12=ABS(XID)    00393
ISN 0385      IF (X12.LT.1.) X12=1. 00394
ISN 0387      GI=(XI-XID)/X12 00395
ISN 0388      IF (GI.LT.-1.0E-6) GO TO 510 00396
ISN 0390      M1=M1+1        00397
ISN 0391      MS1(M1)=I      00398
ISN 0392      NAC=NAC+1      00399
ISN 0393      IF (NAC.GE.N3) GO TO 810 00400
ISN 0395      MCH1=MCN1+1    00401
ISN 0396      DO 500 J=1,NDV  00402
ISN 0397      500  A(J,NAC)=0. 00403
ISN 0398      A(I,NAC)=1.    00404
ISN 0399      IC(NAC)=MCH1   00405
ISN 0400      G(MCN1)=GI     00406
ISN 0401      ISC(MCH1)=1    00407
ISN 0402      510  CONTINUE   00408

```

246

```

C -----00409
C                                     PRINT                      00410
C -----00411
C PRINT ACTIVE SIDE CONSTRAINT NUMBERS.                      00412
ISN 0403 IF (IPRINT.LT.3) GO TO 530                          00413
ISN 0405 WRITE (I6I,1090) M1                                00414
ISN 0406 IF (M1.EQ.0) GO TO 530                              00415
ISN 0408 WRITE (I6I,1100)                                    00416
ISN 0409 WRITE(I6I,1480) (MS1(J),J=1,M1)                    00417
ISN 0410 530 CONTINUE                                         00418
C PRINT GRADIENTS OF ACTIVE AND VIOLATED CONSTRAINTS.      00419
ISN 0411 IF (IPRINT.LT.4) GO TO 570                          00420
ISN 0413 WRITE (I6I,1340)                                    00421
ISN 0414 DO 540 I=1,NDV,6                                     00422
ISN 0415 M1=MIN0(MDV,I+5)                                    00423
ISN 0416 540 WRITE (I6I,1010) I,(DF(J),J=I,M1)              00424
ISN 0417 IF (NAC.EQ.0) GO TO 570                             00425
ISN 0419 WRITE (I6I,1350)                                    00426
ISN 0420 DO 560 I=1,NAC                                       00427
ISN 0421 M1=IC(I)                                           00428
ISN 0422 M2=M1-NCON                                          00429
ISN 0423 M3=0                                                00430
ISN 0424 IF (M2.GT.0) M3=IABS(MS1(M2))                       00431
ISN 0426 IF (M2.LE.0) WRITE (I6I,990) M1                    00432
ISN 0428 IF (M2.GT.0) WRITE (I6I,1000) M3                   00433
ISN 0430 DO 550 K=1,NDV,6                                     00434
ISN 0431 M1=MIN0(MDV,K+5)                                    00435
ISN 0432 550 WRITE (I6I,1010) K,(A(J,I),J=K,M1)              00436
ISN 0433 560 WRITE (I6I,1360)                                00437
ISN 0434 570 CONTINUE                                         00438
C -----00439
C ***** DETERMINE SEARCH DIRECTION *****00440
C -----00441
ISN 0435 ALP=1.0E+20                                         00442
ISN 0436 IF (NAC.GT.0) GO TO 580                             00443
C -----00444
C                                     UNCONSTRAINED FUNCTION  00445
C -----00446
C FIND DIRECTION OF STEEPEST DESCENT OR CONJUGATE DIRECTION. 00447
ISN 0438 NVC=0                                               00448
ISN 0439 NFEAS=0                                             00449
ISN 0440 KCOUNT=KCOUNT+1                                    00450
C IF KCOUNT.GT.ICNDIR RESTART CONJUGATE DIRECTION ALGORITHM. 00451
ISN 0441 IF (KCOUNT.GT.ICNDIR.OR.IOBJ.EQ.2) KCOUNT=1        00452
ISN 0443 IF (KCOUNT.EQ.1) JDIR=0                             00453
C IF JDIR = 0 FIND DIRECTION OF STEEPEST DESCENT.           00454
ISN 0445 CALL CNMN02 (JDIR,SLOPE,DFDF1,DF,S,N1)              00455
ISN 0446 GO TO 630                                           00456
ISN 0447 580 CONTINUE                                         00457
C -----00458
C                                     CONSTRAINED FUNCTION    00459
C -----00460
C FIND USABLE-FEASIBLE DIRECTION.                            00461
ISN 0448 KCOUNT=0                                           00462
ISN 0449 JDIR=0                                               00463
ISN 0450 PHI=10.*PHI                                         00464
ISN 0451 IF (PHI.GT.1000.) PHI=1000.                         00465
C CALCULATE DIRECTION, S.                                    00466
ISN 0453 CALL CNMN05 (G,DF,A,S,B,C,SLOPE,PHI,ISC,IC,MS1,NVC,N1,N2,N3,N4,N5)00467

```

```

ISN 0454      IF (IPRINT.LT.3) GO TO 600      00468
ISN 0456      WRITE (I6I,1370)                00469
ISN 0457      DO 590 I=1,NAC,6                00470
ISN 0458      M1=MIND(NAC,I+5)                00471
ISN 0459      590 WRITE (I6I,1010) I,(A(NDV1,J),J=I,M1) 00472
ISN 0460      WRITE (I6I,1210) S(NDV1)        00473
ISN 0461      600 CONTINUE                    00474
C                                                     00475
C ***** ONE-DIMENSIONAL SEARCH ***** 00476
C ----- 00477
ISN 0462      IF (S(NDV1).LT.1.0E-6.AND.NVC.EQ.0) GO TO 710 00478
C ----- 00479
C             FIND ALPHA TO OBTAIN A FEASIBLE DESIGN 00480
C ----- 00481
ISN 0464      IF (NVC.EQ.0) GO TO 630          00482
ISN 0466      ALP=-1.                          00483
ISN 0467      DO 620 I=1,NAC                  00484
ISN 0468      NCI=IC(I)                      00485
ISN 0469      C1=G(NCI)                      00486
ISN 0470      CTC=CTAM                       00487
ISN 0471      IF (ISC(NCI).GT.0) CTC=CTBM     00488
ISN 0473      IF (C1.LE.CTC) GO TO 620        00489
ISN 0475      ALP1=0.                         00490
ISN 0476      DO 610 J=1,NDV                  00491
ISN 0477      610 ALP1=ALP1+S(J)*A(J,I)       00492
ISN 0478      ALP1=ALP1*A(NDV2,I)            00493
ISN 0479      IF (ABS(ALP1).LT.1.0E-20) GO TO 620 00494
ISN 0481      ALP1=-C1/ALP1                   00495
ISN 0482      IF (ALP1.GT.ALP) ALP=ALP1       00496
ISN 0484      620 CONTINUE                    00497
ISN 0485      630 CONTINUE                    00498
C ----- 00499
C             LIMIT CHANCE TO ABOBJ1*OBJ      00500
C ----- 00501
ISN 0486      ALP1=1.0E+20                    00502
ISN 0487      SI=ABS(OBJ)                     00503
ISN 0488      IF (SI.LT..01) SI=.01           00504
ISN 0490      IF (ABS(SLOPE).GT.1.0E-20) ALP1=ABOBJ1*SI/SLOPE 00505
ISN 0492      ALP1=ABS(ALP1)                  00506
ISN 0493      IF (NVC.GT.0) ALP1=10.*ALP1     00507
ISN 0495      IF (ALP1.LT.ALP) ALP=ALP1       00508
C ----- 00509
C             LIMIT CHANGE IN VARIABLE TO ALPHAX 00510
C ----- 00511
ISN 0497      ALP11=1.0E+20                   00512
ISN 0498      DO 640 I=1,NDV                  00513
ISN 0499      SI=ABS(S(I))                    00514
ISN 0500      XI=ABS(X(I))                    00515
ISN 0501      IF (SI.LT.1.0E-10.OR.XI.LT.0.1) GO TO 640 00516
ISN 0503      ALP1=ALPHAX*XI/SI               00517
ISN 0504      IF (ALP1.LT.ALPH11) ALPH11=ALP1 00518
ISN 0506      640 CONTINUE                    00519
ISN 0507      IF (NVC.GT.0) ALPH11=10.*ALPH11 00520
ISN 0509      IF (ALPH11.LT.ALP) ALP=ALPH11  00521
ISN 0511      IF (ALP.GT.1.0E+20) ALP=1.0E+20 00522
ISN 0513      IF (ALP.LE.1.0E-20) ALP=1.0E-20 00523
ISN 0515      IF (IPRINT.LT.3) GO TO 660     00524
ISN 0517      WRITE (I6I,1380)                00525
ISN 0518      DO 650 I=1,NDV,6                00526

```

248

```

ISN 0519      M1=MIN0(NDV,I+5)                                00527
ISN 0520      650  WRITE (I6I,1010) I,(S(J),J=I,M1)          00528
ISN 0521      WRITE (I6I,1110) SLOPE,ALP                    00529
ISN 0522      660  CONTINUE                                  00530
ISN 0523      IF (NCON.GT.0.OR.NSIDE.GT.0) GO TO 680        00531
C             -----00532
C             DO ONE-DIMENSIONAL SEARCH FOR UNCONSTRAINED FUNCTION 00533
C             -----00534
ISN 0525      JGOTO=0                                        00535
ISN 0526      670  CONTINUE                                  00536
ISN 0527      CALL CNMN03 (X,S,SLOPE,ALP,FFF,A1,A2,A3,A4,F1,F2,F3,F4,APP,N1,NCAL00537
              1,KOUNT,JGOTO)                                00538
ISN 0528      IGOTO=4                                       00539
ISN 0529      IF (JGOTO.GT.0) GO TO 950                     00540
ISN 0531      JDIR=1                                         00541
C             PROCEED TO CONVERGENCE CHECK.                 00542
ISN 0532      GO TO 700                                       00543
C             -----00544
C             SOLVE ONE-DIMENSIONAL SEARCH PROBLEM FOR CONSTRAINED FUNCTION 00545
C             -----00546
ISN 0533      680  CONTINUE                                  00547
ISN 0534      JGOTO=0                                        00548
ISN 0535      690  CONTINUE                                  00549
ISN 0536      CALL CNMN06 (X,VLB,VUB,G,SCAL,DF,S,G1,G2,CTAM,CTBM,SLOPE,ALP,A2,A300550
              1,A4,F1,F2,F3,CV1,CV2,CV3,CV4,ALPCA,ALPFES,ALPLN,ALPMIN,ALPNC,ALPSA00551
              2V,ALPSID,ALPTOT,ISC,N1,N2,NCAL,NVC,ICOUNT,IGOOD1,IGOOD2,IGOOD3,IGOO00552
              3OD4,IBEST,III,NLNC,JGOTO)                    00553
ISN 0537      IGOTO=5                                       00554
ISN 0538      IF (JGOTO.GT.0) GO TO 950                     00555
ISN 0540      IF (NAC.EQ.0) JDIR=1                          00556
C             -----00557
C             ***** UPDATE ALPHAX *****00558
C             -----00559
ISN 0542      700  CONTINUE                                  00560
ISN 0543      710  CONTINUE                                  00561
ISN 0544      IF (ALP.GT.1.0E+19) ALP=0.                    00562
C             UPDATE ALPHAX TO BE AVERAGE OF MAXIMUM CHANGE IN X(I) 00563
C             AND ALPHAX.                                     00564
ISN 0546      ALP11=0.                                       00565
ISN 0547      DO 720 I=1,NDV                                  00566
ISN 0548      SI=ABS(S(I))                                    00567
ISN 0549      XI=ABS(X(I))                                    00568
ISN 0550      IF (XI.LT.1.0E-10) GO TO 720                 00569
ISN 0552      ALP1=ALP*SI/XI                                  00570
ISN 0553      IF (ALP1.GT.ALPHAX) ALP11=ALP1               00571
ISN 0555      720  CONTINUE                                  00572
ISN 0556      ALP11=.5*(ALP11+ALPHAX)                       00573
ISN 0557      ALP12=5.*ALPHAX                               00574
ISN 0558      IF (ALP11.GT.ALPHAX) ALP11=ALP12             00575
ISN 0560      ALPHAX=ALP11                                   00576
ISN 0561      NCOBJ=NCOBJ+1                                  00577
C             ABSOLUTE CHANGE IN OBJECTIVE.                 00578
ISN 0562      OBJD=OBJ1-OBJ                                   00579
ISN 0563      OBJB=ABS(OBJD)                                  00580
ISN 0564      IF (OBJB.LT.1.0E-10) OBJB=0.                 00581
ISN 0566      IF (NAC.EQ.0.OR.OBJB.GT.0.) NCOBJ=0          00582
ISN 0568      IF (NCOBJ.GT.1) NCOBJ=0                      00583
C             -----00584
C             PRINT                                          00585
    
```

```

C -----00586
C PRINT MOVE PARAMETER, NEW X-VECTOR AND CONSTRAINTS. 00587
ISN 0570 IF (IPRINT.LT.3) GO TO 730 00588
ISN 0572 WRITE (I6I,1390) ALP 00589
ISN 0573 730 IF (IPRINT.LT.2) GO TO 800 00590
ISN 0575 IF (OBJB.GT.0.) GO TO 740 00591
ISN 0577 IF (IPRINT.EQ.2) WRITE (I6I,1400) ITER,OBJ 00592
ISN 0579 IF (IPRINT.GT.2) WRITE (I6I,1410) OBJ 00593
ISN 0581 GO TO 760 00594
ISN 0582 740 IF (IPRINT.EQ.2) GO TO 750 00595
ISN 0584 WRITE (I6I,1420) OBJ 00596
ISN 0585 GO TO 760 00597
ISN 0586 750 WRITE (I6I,1430) ITER,OBJ 00598
ISN 0587 760 WRITE (I6I,1450) 00599
ISN 0588 DO 770 I=1,NDV 00600
ISN 0589 FF1=1. 00601
ISN 0590 IF (NSCAL.NE.0) FF1=SCAL(I) 00602
ISN 0592 770 G1(I)=FF1*X(I) 00603
ISN 0593 DO 780 I=1,NDV,6 00604
ISN 0594 M1=MINO(NDV,I+5) 00605
ISN 0595 780 WRITE (I6I,1010) I,(G1(J),J=I,M1) 00606
ISN 0596 IF (NCON.EQ.0) GO TO 800 00607
ISN 0598 WRITE (I6I,1470) 00608
ISN 0599 DO 790 I=1,NCON,6 00609
ISN 0600 M1=MINO(NCON,I+5) 00610
ISN 0601 790 WRITE (I6I,1010) I,(G(J),J=I,M1) 00611
ISN 0602 800 CONTINUE 00612
C -----00613
C CHECK FEASABILITY 00614
C -----00615
ISN 0603 IF(NCON.LE.0) GO TO 808 00616
ISN 0605 DO 804 I=1,NCON 00617
ISN 0606 C1=CTAM 00618
ISN 0607 IF(ISC(I).GT.0) C1=CTBM 00619
ISN 0609 IF(G(I).LE.C1) GO TO 804 00620
ISN 0611 NFEAS=NFEAS+1 00621
ISN 0612 GO TO 806 00622
ISN 0613 804 CONTINUE 00623
ISN 0614 IF(NFEAS.GT.0) ABOBJ1=.05 00624
ISN 0616 NFEAS=0 00625
ISN 0617 PHI=5. 00626
ISN 0618 806 IF(NFEAS.GE.10) GO TO 810 00627
ISN 0620 808 CONTINUE 00628
C -----00629
C CHECK CONVERGENCE 00630
C -----00631
C STOP IF ITER EQUALS ITMAX. 00632
ISN 0621 IF (ITER.GE.ITMAX) GO TO 810 00633
C -----00634
C ABSOLUTE CHANGE IN OBJECTIVE 00635
C -----00636
ISN 0623 OBJB=ABS(OBJD) 00637
ISN 0624 KOBJ=KOBJ+1 00638
ISN 0625 IF (OBJB.GE.DABFUN.OR.NFEAS.GT.0) KOBJ=0 00639
C -----00640
C RELATIVE CHANGE IN OBJECTIVE 00641
C -----00642
ISN 0627 IF (ABS(OBJ1).GT.1.0E-10) OBJD=OBJD/ABS(OBJ1) 00643
ISN 0629 ABOBJ1=.5*(ABS(ABOBJ)+ABS(OBJD)) 00644
    
```

250

```

ISN 0630      ABOBJ=ABS(OBJD)                                00645
ISN 0631      IOBJ=IOBJ+1                                  00646
ISN 0632      IF(NFEAS.GT.0.OR.OBJD.GE.DELFUN) IOBJ=0      00647
ISN 0634      IF (IOBJ.GE.ITRM.OR.KOBJ.GE.ITRM) GO TO 810  00648
ISN 0636      OBJ1=OBJ                                      00649
C-----00650
C      REDUCE CT IF OBJECTIVE FUNCTION IS CHANGING SLOWLY  00651
C-----00652
ISN 0637      IF (IOBJ.LT.1.OR.NAC.EQ.0) GO TO 280         00653
ISN 0639      CT=DCT*CT                                    00654
ISN 0640      CTL=CTL*DCTL                                 00655
ISN 0641      IF (ABS(CT).LT.CTMIN) CT=-CTMIN             00656
ISN 0643      IF (ABS(CTL).LT.CTLMIN) CTL=-CTLMIN         00657
ISN 0645      GO TO 280                                    00658
ISN 0646      810 CONTINUE                                 00659
ISN 0647      IF (NAC.GE.N3) WRITE (I6I,1490)              00660
C-----00661
C      ***** FINAL FUNCTION INFORMATION *****          00662
C-----00663
ISN 0649      IF (NSCAL.EQ.0) GO TO 830                    00664
C      UN-SCALE THE DESIGN VARIABLES.                    00665
ISN 0651      DO 820 I=1,NDV                               00666
ISN 0652      XI=SCAL(I)                                  00667
ISN 0653      IF (NSIDE.EQ.0) GO TO 820                   00668
ISN 0655      VLB(I)=XI*VLB(I)                            00669
ISN 0656      VUB(I)=XI*VUB(I)                           00670
ISN 0657      820 X(I)=XI*X(I)                            00671
C-----00672
C      PRINT FINAL RESULTS                                00673
C-----00674
ISN 0658      830 IF (IPRINT.EQ.0.OR.NAC.GE.N3) GO TO 940  00675
ISN 0660      WRITE (I6I,1500)                             00676
ISN 0661      WRITE (I6I,1420) OBJ                         00677
ISN 0662      WRITE (I6I,1450)                             00678
ISN 0663      DO 840 I=1,NDV,6                            00679
ISN 0664      M1=MIN0(NDV,I+5)                            00680
ISN 0665      840 WRITE (I6I,1010) I,(X(J),J=I,M1)        00681
ISN 0666      IF (NCON.EQ.0) GO TO 900                    00682
ISN 0668      WRITE (I6I,1470)                             00683
ISN 0669      DO 850 I=1,NCON,6                            00684
ISN 0670      M1=MIN0(NCON,I+5)                           00685
ISN 0671      850 WRITE (I6I,1010) I,(G(J),J=I,M1)        00686
C      DETERMINE WHICH CONSTRAINTS ARE ACTIVE AND PRINT.  00687
ISN 0672      NAC=0                                        00688
ISN 0673      NVC=0                                        00689
ISN 0674      DO 870 I=1,NCON                              00690
ISN 0675      CTA=CTAM                                     00691
ISN 0676      IF (ISC(I).GT.0) CTA=CTBM                   00692
ISN 0678      GI=G(I)                                     00693
ISN 0679      IF (GI.GT.CTA) GO TO 860                    00694
ISN 0681      IF (GI.LT.CT.AND.ISC(I).EQ.0) GO TO 870    00695
ISN 0683      IF (GI.LT.CTL.AND.ISC(I).GT.0) GO TO 870  00696
ISN 0685      NAC=NAC+1                                    00697
ISN 0686      IC(NAC)=I                                    00698
ISN 0687      GO TO 870                                    00699
ISN 0688      860 NVC=NVC+I                               00700
ISN 0689      MSI(NVC)=I                                   00701
ISN 0690      870 CCNTINUE                                 00702
ISN 0691      WRITE (I6I,1060) NAC                        00703
    
```

```

+VERSION 1.3.0 (01 MAY 80)  COMMIN  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 82.141/10.53.56  PAGE 13
  ISN 0692      IF (NAC.EQ.0) GO TO 880 00704
  ISN 0694      WRITE (I6I,1070) 00705
  ISN 0695      WRITE (I6I,1480) (IC(J),J=1,NAC) 00706
  ISN 0696      880 WRITE (I6I,1080) NVC 00707
  ISN 0697      IF (NVC.EQ.0) GO TO 890 00708
  ISN 0699      WRITE (I6I,1070) 00709
  ISN 0700      WRITE (I6I,1480) (MS1(J),J=1,NVC) 00710
  ISN 0701      890 CONTINUE 00711
  ISN 0702      900 CONTINUE 00712
  ISN 0703      IF (NSIDE.EQ.0) GO TO 930 00713
  C DETERMINE WHICH SIDE CONSTRAINTS ARE ACTIVE AND PRINT. 00714
  NAC=0 00715
  ISN 0705      DO 920 I=1,NDV 00716
  ISN 0706      XI=X(I) 00717
  ISN 0707      XID=VLB(I) 00718
  ISN 0708      X12=ABS(XID) 00719
  ISN 0709      IF (X12.LT.1.) X12=1. 00720
  ISN 0710      GI=(XID-XI)/X12 00721
  ISN 0711      IF (GI.LT.-1.0E-6) GO TO 910 00722
  ISN 0712      NAC=NAC+1 00723
  ISN 0713      MS1(NAC)=-I 00724
  ISN 0714      910 XID=VUB(I) 00725
  ISN 0715      X12=ABS(XID) 00726
  ISN 0716      IF (X12.LT.1.) X12=1. 00727
  ISN 0717      GI=(XI-XID)/X12 00728
  ISN 0718      IF (GI.LT.-1.0E-6) GO TO 920 00729
  ISN 0719      NAC=NAC+1 00730
  ISN 0720      MS1(NAC)=I 00731
  ISN 0721      920 CONTINUE 00732
  ISN 0722      WRITE (I6I,1090) NAC 00733
  ISN 0723      IF (NAC.EQ.0) GO TO 930 00734
  ISN 0724      WRITE (I6I,1100) 00735
  ISN 0725      WRITE (I6I,1480) (MS1(J),J=1,NAC) 00736
  ISN 0726      930 CONTINUE 00737
  ISN 0727      WRITE (I6I,1150) 00738
  ISN 0728      IF (ITER.GE.ITMAX) WRITE (I6I,1160) 00739
  ISN 0729      IF (NFEAS.GE.10) WRITE (I6I,1170) 00740
  ISN 0730      IF (IOBJ.GE.ITRM) WRITE (I6I,1180) ITRM 00741
  ISN 0731      IF (KOBJ.GE.ITRM) WRITE (I6I,1190) ITRM 00742
  ISN 0732      WRITE (I6I,1200) ITER 00743
  ISN 0733      WRITE (I6I,1510) NCAL(1) 00744
  ISN 0734      IF (NCON.GT.0) WRITE (I6I,1520) NCAL(1) 00745
  ISN 0735      IF (NFDG.NE.0) WRITE (I6I,1530) NCAL(2) 00746
  ISN 0736      IF (NCON.GT.0.AND.NFDG.EQ.1) WRITE (I6I,1540) NCAL(2) 00747
  C -----00748
  C RE-SET BASIC PARAMETERS TO INPUT VALUES 00749
  C -----00750
  ISN 0750      940 ITRM=IDM1 00751
  ISN 0751      ITMAX=IDM2 00752
  ISN 0752      ICNDIR=IDM3 00753
  ISN 0753      DELFUN=DM1 00754
  ISN 0754      DABFUN=DM2 00755
  ISN 0755      CT=DM3 00756
  ISN 0756      CTHIN=DM4 00757
  ISN 0757      CTL=DM5 00758
  ISN 0758      CTLMIN=DM6 00759
  ISN 0759      THETA=DM7 00760
  ISN 0760      PHI=DM8 00761
  ISN 0761      FDCH=DM9 00762

```

252

```

ISN 0762      FDCHM=DM10                      00763
ISN 0763      ABOBJ1=DM11                     00764
ISN 0764      ALPHAX=DM12                     00765
ISN 0765      IGOTO=0                          00766
ISN 0766      950 CONTINUE                      00767
ISN 0767      IF (NSCAL.EQ.0.OR.IGOTO.EQ.0) RETURN 00768
              C UN-SCALE VARIABLES.           00769
ISN 0769      DO 960 I=1,NDV                    00770
ISN 0770      C(I)=X(I)                         00771
ISN 0771      960 X(I)=X(I)*SCAL(I)             00772
ISN 0772      RETURN                            00773
              C -----00774
              C                      FORMATS    00775
              C -----00776
              C                      00777
              C                      00778
ISN 0773      970 FORMAT (///5X,72HA COMPLETELY UNCONSTRAINED FUNCTION WITH A LINEAR00779
              1 OBJECTIVE IS SPECIFIED//10X,8HLINOBJ =,I5/10X,8HNCON =,I5/10X,800780
              2HNSIDE =,I5//5X,35HCONTROL RETURNED TO CALLING PROGRAM) 00781
ISN 0774      980 FORMAT (///5X,56HCONMIN HAS ACHIEVED A SOLUTION OF OBJ LESS THAN -00782
              11.0E+40/5X,32HSOLUTION APPEARS TOABE UNBOUNDED/5X,26HOPTIMIZATION 00783
              2IS TERMINATED) 00784
ISN 0775      990 FORMAT (5X,17HCONSTRAINT NUMBER,I5) 00785
ISN 0776      1000 FORMAT (5X,27HSIDE CONSTRAINT ON VARIABLE,I5) 00786
ISN 0777      1010 FORMAT (3X,I5,1H),2X,6E13.5) 00787
ISN 0778      1020 FORMAT (/5X,35HLINEAR CONSTRAINT IDENTIFIERS (ISC)/5X,36HNON-ZERO 00788
              1INDICATES LINEAR CONSTRAINT) 00789
ISN 0779      1030 FORMAT (3X,I5,1H),2X,15I5) 00790
ISN 0780      1040 FORMAT (/5X,26HALL CONSTRAINTS ARE LINEAR) 00791
ISN 0781      1050 FORMAT (/5X,30HALL CONSTRAINTS ARE NON-LINEAR) 00792
ISN 0782      1060 FORMAT (/5X,9HTHERE ARE,I5,19H ACTIVE CONSTRAINTS) 00793
ISN 0783      1070 FORMAT (5X,22HCONSTRAINT NUMBERS ARE) 00794
ISN 0784      1080 FORMAT (/5X,9HTHERE ARE,I5,21H VIOLATED CONSTRAINTS) 00795
ISN 0785      1090 FORMAT (/5X,9HTHERE ARE,I5,24H ACTIVE SIDE CONSTRAINTS) 00796
ISN 0786      1100 FORMAT (5X,43HDECISION VARIABLES AT LOWER OR UPPER BOUNDS,30H (MIN00797
              1US INDICATES LOWER BOUND)) 00798
ISN 0787      1110 FORMAT (/5X,22HONE-DIMENSIONAL SEARCH/5X,15HINITIAL SLOPE =,E12.4,00799
              12X,16HPROPOSED ALPHA =,E12.4) 00800
ISN 0788      1120 FORMAT (///5X,35H* * CONMIN DETECTS VLB(I).GT.VUB(I)/5X,57HFIX IS 00801
              1SET X(I)=VLB(I)=VUB(I) = .5*(VLB(I)+VUB(I) FOR I =,I5) 00802
ISN 0789      1130 FORMAT (///5X,41H* * CONMIN DETECTS INITIAL X(I).LT.VLB(I)/5X,6HX(00803
              1I) =,E12.4,2X,8HVUB(I) =,E12.4/5X,35HX(I) IS SET EQUAL TO VLB(I) F00804
              2OR I =,I5) 00805
ISN 0790      1140 FORMAT (///5X,41H* * CONMIN DETECTS INITIAL X(I).GT.VUB(I)/5X,6HX(00806
              1I) =,E12.4,2X,8HVUB(I) =,E12.4/5X,35HX(I) IS SET EQUAL TO VUB(I) F00807
              2OR I =,I5) 00808
ISN 0791      1150 FORMAT (/5X,21HTERMINATION CRITERION) 00809
ISN 0792      1160 FORMAT (10X,17HITER EQUALS ITMAX) 00810
ISN 0793      1170 FORMAT (10X,62HTEN CONSECUTIVE ITERATIONS FAILED TO PRODUCE A FEAS00811
              1IBLE DESIGN) 00812
ISN 0794      1180 FORMAT (10X,43HABS(1-OBJ(I-1)/OBJ(I)) LESS THAN DELFUN FOR,I3,11H 00813
              1ITERATIONS) 00814
ISN 0795      1190 FORMAT (10X,43HABS(OBJ(I)-OBJ(I-1)) LESS THAN DABFUN FOR,I3,11H 00815
              1ITERATIONS) 00816
ISN 0796      1200 FORMAT (/5X,22HNUMBER OF ITERATIONS =,I5) 00817
ISN 0797      1210 FORMAT (/5X,28HCONSTRAINT PARAMETER, BETA =,E14.5) 00818
ISN 0798      1220 FORMAT (1H1,////12X,27(2H* )/12X,1H*,51X,1H*/12X,1H*,20X,11HC O N 00819
              1M I N,20X,1H*/12X,1H*,51X,1H*/12X,1H*,15X,21H FORTRAN PROGRAM FOR 00820
              2,15X,1H*/12X,1H*,51X,1H*/12X,1H*,9X,33HCONSTRAINED FUNCTION MINIMIO0821
    
```

```

3ZATION,9X,1H*/12X,1H*,51X,1H*/12X,27(2H* )) 00322
ISN 0799 1230 FORMAT (///5X,33HCONSTRAINED FUNCTION MINIMIZATION//5X,18HCONTROL00823
1 PARAMETERS) 00824
ISN 0800 1240 FORMAT (/5X,60HIPRINT NDV ITMAX NCON NSIDE ICNDR NSC00825
1AL NFDG/8I8//5X,12HLINEOBJ ITRM,5X,2HN1,6X,2HN2,6X,2HN3,6X,2HN4,00826
26X,2HN5/8I8) 00827
ISN 0801 1250 FORMAT (/9X,4HFDCH,12X,5HFDCHM,11X,6HALPHAX,10X,6HABOBJ1/1X,4(2X,E00828
114.5)) 00829
ISN 0802 1260 FORMAT (/9X,2HCT,14X,5HCTMIN,11X,3HCTL,13X,6HCTLMIN/1X,4(2X,E14.5)00830
1//9X,5HTheta,11X,3HPHI,13X,6HDELFUN,10X,6HDABFUN/1X,4(2X,E14.5)) 00831
ISN 0803 1270 FORMAT (/5X,40HLOWER BOUNDS ON DECISION VARIABLES (VLB)) 00832
ISN 0804 1280 FORMAT (/5X,40HUPPER BOUNDS ON DECISION VARIABLES (VUB)) 00833
ISN 0805 1290 FORMAT (///5X,35HUNCONSTRAINED FUNCTION MINIMIZATION//5X,18HCONTR00834
1OL PARAMETERS) 00835
ISN 0806 1300 FORMAT (/5X,21HSCALING VECTOR (SCAL)) 00836
ISN 0807 1310 FORMAT (///5X,22HBEGIN ITERATION NUMBER,I5) 00837
ISN 0808 1320 FORMAT (/5X,4HCT =,E14.5,5X,5HCTL =,E14.5,5X,5HPHI =,E14.5) 00838
ISN 0809 1330 FORMAT (/5X,25HNEW SCALING VECTOR (SCAL)) 00839
ISN 0810 1340 FORMAT (/5X,15HGRADIENT OF OBJ) 00840
ISN 0811 1350 FORMAT (/5X,44HGRADIENTS OF ACTIVE AND VIOLATED CONSTRAINTS) 00841
ISN 0812 1360 FORMAT (1H ) 00842
ISN 0813 1370 FORMAT (/5X,37HPUSH-OFF FACTORS, (THETA(I), I=1,MAC)) 00843
ISN 0814 1380 FORMAT (/5X,27HSEARCH DIRECTION (S-VECTOR)) 00844
ISN 0815 1390 FORMAT (/5X,18HCALCULATED ALPHA =,E14.5) 00845
ISN 0816 1400 FORMAT (///5X,6HITER =,I5,5X,5HOBJ =,E14.5,5X,16HNO CHANGE IN OBJ00846
1) 00847
ISN 0817 1410 FORMAT (/5X,5HOBJ =,E15.6,5X,16HNO CHANGE ON OBJ) 00848
ISN 0818 1420 FORMAT (/5X,5HOBJ =,E15.6) 00849
ISN 0819 1430 FORMAT (///5X,6HITER =,I5,5X,5HOBJ =,E14.5) 00850
ISN 0820 1440 FORMAT (//5X,28HINITIAL FUNCTION INFORMATION//5X,5HOBJ =,E15.6) 00851
ISN 0821 1450 FORMAT (/5X,29HDECISION VARIABLES (X-VECTOR)) 00852
ISN 0822 1460 FORMAT (3X,7E13.4) 00853
ISN 0823 1470 FORMAT (/5X,28HCONSTRAINT VALUES (G-VECTOR)) 00854
ISN 0824 1480 FORMAT (5X,15I5) 00855
ISN 0825 1490 FORMAT (/5X,59HTHE NUMBER OF ACTIVE AND VIOLATED CONSTRAINTS EXCEE00856
1DS N3-1./5X,66HDIMENSIONED SIZE OF MATRICES A AND B AND VECTOR IC 00857
2IS INSUFFICIENT/5X,61HOPTIMIZATION TERMINATED AND CONTROL RETURNED00858
3 TO MAIN PROGRAM.) 00859
ISN 0826 1500 FORMAT (1H1,///4X,30HFINAL OPTIMIZATION INFORMATION) 00860
ISN 0827 1510 FORMAT (/5X,32HOBJECTIVE FUNCTION WAS EVALUATED,8X,I5,2X,5HTIMES) 00861
ISN 0828 1520 FORMAT (/5X,35HCONSTRAINT FUNCTIONS WERE EVALUATED,I10,2X,5HTIMES)00862
ISN 0829 1530 FORMAT (/5X,36HGRADIENT OF OBJECTIVE WAS CALCULATED,I9,2X,5HTIMES)00863
ISN 0830 1540 FORMAT (/5X,40HGRADIENTS OF CONSTRAINTS WERE CALCULATED,I5,2X,5HTI00864
IMES) 00865
ISN 0831 END 00866

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTHT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 830, PROGRAM SIZE = 15164, SUBPROGRAM NAME =CONMIN
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

2844K BYTES OF CORE NOT USED

254

```

C          DATA SET U477COPE01 AT LEVEL 003 AS OF 03/17/81
C          DATA SET U477COPE01 AT LEVEL 002 AS OF 03/13/81          00001
C          DATA SET U477COPE01 AT LEVEL 001 AS OF 02/13/81          00002
C          DATA SET 9188COPE01 AT LEVEL 001 AS OF 07/10/80          00003
ISN 0002   SUBROUTINE COPE01 (RA,IA,NDRA,NDIA)                          00004
ISN 0003   COMMON /CNMNI/ DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,ALPHAX00005
          1,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,LINOBJ,ITMAX,IT00006
          2RM,ICNDIR,IGOTO,NAC,INFO,INFOG,ITER                          00007
ISN 0004   COMMON /COPE01/ ATITLE(20)                                  00008
ISN 0005   COMMON /COPE03/ SGNOPT,NCALC,I0BJ,NSV,NSOBJ,NCONA,N2VX,M2VX,N2VY,M00009
          12VY,N2VAR,IPSENS,IP2VAR,IPDBG,NACMX1,NDVTOT,LOCR(25),LOCI(25),ISCR00010
          21,ISCR2,NXAPRX,NPS,NPFS,NPA,NF,INOM,IPAPRX,KMIN,KMAX,XFACT1,XFACT200011
          3,NAN2,NAN3,NPMAX,NPTOT,JNOM,MAXTRM                          00012
ISN 0006   COMMON /UIOS/ I5I, I6I                                    00013
ISN 0007   DIMENSION RA(NDRA), IA(NDIA), TITLE(20)                   00014
ISN 0008   DATA END1/IHE/,END2/IHN/,END3/IHD/                       00015
ISN 0009   DATA COM/1H$/ ,BLANK/1H /                                00016
C          *****00017
C          ROUTINE TO READ CONTROL INPUT FOR COPE03.                  00018
C          *****00019
C          BY G. N. VANDERPLAATS MAR., 1973.                          00020
C          NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.          00021
C          -----00022
C          READ CARD IMAGES AND STORE ON UNIT ISCR2. STORE ON UNIT ISCR1 00023
C          WITHOUT COMMENT CARDS                                       00024
C          -----00025
ISN 0010   REWIND ISCR1                                               00026
ISN 0011   REWIND ISCR2                                               00027
ISN 0012   NCARDS=0                                                    00028
ISN 0013   LOCI(25)=0                                                  00029
ISN 0014   NCOM=0                                                      00030
ISN 0015   ICARD=0                                                      00031
ISN 0016   10 READ (I5I,580) (RA(I),I=1,80)                          00032
ISN 0017   ICARD=ICARD+1                                                00033
ISN 0018   NCARDS=NCARDS+1                                              00034
ISN 0019   WRITE (ISCR2,590) NCARDS,(RA(I),I=1,80)                   00035
ISN 0020   IF (RA(1).EQ.COM) GO TO 10                                    00036
ISN 0022   IF (RA(1).EQ.END1.AND.(RA(2).EQ.END2.AND.RA(3).EQ.END3)) GO TO 20 00037
ISN 0024   IF (NCOM.NE.0) GO TO 30                                       00038
C          TITLE OR END CARD.                                          00039
ISN 0026   20 WRITE (ISCR1,580) (RA(I),I=1,80)                        00040
ISN 0027   IF (NCOM.GT.0) GO TO 70                                       00041
C          IT WAS THE TITLE CARD.                                       00042
ISN 0029   NCOM=1                                                       00043
ISN 0030   GO TO 10                                                       00044
ISN 0031   30 CONTINUE                                                  00045
C          FORMAT DATA AS REQUIRED.                                       00046
ISN 0032   NA=1                                                         00047
ISN 0033   NB=81                                                         00048
ISN 0034   CALL COPE08 (RA(NA),RA(NB),IFORM,NFLD)                     00049
C          DETERMINE NUMBER OF CARDS OF DATA AND ADD BLANKS TO FILL. 00050
ISN 0035   NBC=(NFLD-1)/8+1                                             00051
ISN 0036   NB=80*NBC+80                                                 00052
ISN 0037   NA=10*NFLD+81                                               00053
ISN 0038   IF (NA.GE.NB) GO TO 50                                       00054

```

```

+VERSION 1.3.0 (01 MAY 80)   COPE01   SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)   DATE 82.141/10.54.12   PAGE 2
  ISN 0040      DO 40 I=NA,NB                               00055
  ISN 0041      40  RA(I)=BLANK                             00056
  ISN 0042      50  CONTINUE                                00057
  ISN 0043      WRITE (ISCR1,580) (RA(I),I=81,NB)          00058
  ISN 0044      IF (IFORM.GT.0) GO TO 10                   00059
  C             DATA WAS NOT PREVIOUSLY FORMATTED.       00060
  ISN 0046      NI=1                                        00061
  ISN 0047      DO 60 II=1,NBC                              00062
  ISN 0048      NI=NI+80                                   00063
  ISN 0049      N2=NI+79                                   00064
  ISN 0050      WRITE (ISCR2,590) NCARDS,(RA(I),I=N1,N2)  00065
  ISN 0051      60  ICARD=ICARD+1                          00066
  ISN 0052      GO TO 10                                   00067
  ISN 0053      70  CONTINUE                                00068
  ISN 0054      REWIND ISCR1                               00069
  ISN 0055      REWIND ISCR2                              00070
  C             -----00071
  C             GENERAL SYNTHESIS INFORMATION               00072
  C             -----00073
  C             TITLE.                                     00074
  C --- DATA BLOCK A.                                    00075
  ISN 0056      READ (ISCR1,1190) (ATITLE(I),I=1,20)      00076
  C             CONTROL PARAMETERS.                       00077
  C --- DATA BLOCK B.                                    00078
  ISN 0057      READ (ISCR1,1200) NCALC,NDV,NSV,N2VAR,NXAPRX,IPNPUT,IPDBG 00079
  ISN 0058      IF (NCALC.LT.0.OR.NCALC.GT.6) WRITE (I6I,1220) NCALC 00080
  ISN 0060      IF (NCALC.LT.0.OR.NCALC.GT.6) RETURN      00081
  ISN 0062      IF (IPNPUT.GT.1) GO TO 100                00082
  ISN 0064      WRITE (I6I,970)                           00083
  ISN 0065      WRITE (I6I,980)                           00084
  ISN 0066      WRITE (I6I,990) (ATITLE(I),I=1,20)        00085
  C             -----00086
  C             CARD IMAGE PRINT                          00087
  C             -----00088
  ISN 0067      IF (IPNPUT.GT.0) GO TO 90                  00089
  ISN 0069      WRITE (I6I,870)                           00090
  ISN 0070      WRITE (I6I,880)                           00091
  ISN 0071      DO 80 I=1,ICARD                            00092
  ISN 0072      READ (ISCR2,590) NCARDS,(RA(J),J=1,80)    00093
  ISN 0073      80  WRITE (I6I,890) NCARDS,(RA(J),J=1,80) 00094
  ISN 0074      REWIND ISCR2                              00095
  ISN 0075      90  CONTINUE                               00096
  ISN 0076      WRITE (I6I,1000) (ATITLE(I),I=1,20)       00097
  ISN 0077      WRITE (I6I,1010) NCALC,NDV,NSV,N2VAR,NXAPRX,IPNPUT,IPDBG 00098
  ISN 0078      WRITE (I6I,910)                           00099
  ISN 0079      100 NACMX1=0                               00100
  ISN 0080      NDVTOT=0                                   00101
  ISN 0081      NCONA=0                                    00102
  ISN 0082      NCON=0                                     00103
  ISN 0083      IF (NDV.LE.0) GO TO 270                   00104
  C             -----00105
  C             OPTIMIZATION INFORMATION                   00106
  C             -----00107
  C             OPTIMIZATION CONTROL VARIABLES. - COMMON DEPENDENT. 00108
  C --- DATA BLOCK C.                                    00109
  ISN 0085      READ (ISCR1,1200) IPRINT,ITMAX,ICNDIR,NSCAL,ITRN,LINOBJ,NACMX1,NFD00110
  1G                                                     00111
  C --- DATA BLOCK D.                                    00112
  ISN 0086      READ (ISCR1,1210) FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,DELFUN,00113

```

256

```

IDABFUN,ALPHAX,ABOJJI                                00114
C --- DATA BLOCK E.                                  00115
C TOTAL NO. OF D. V., OBJECTIVE GLOBAL NUMBER, SIGN  00116
C ON OPTIMIZATION OBJECTIVE.                         00117
ISN 0087 READ (ISCR1,920) NDVTOT,IOBJ,SGNOPT          00118
ISN 0088 IF (NDVTOT.LT.NDV) NDVTOT=NDV              00119
ISN 0090 IF (NCALC.EQ.6.AND.NACMX1.EQ.0) NACMX1=2*NDV+2 00120
ISN 0092 IF (NACMX1.LE.0) NACMX1=NDV+2              00121
ISN 0094 IF (IPNPUT.GE.2) GO TO 110                  00122
ISN 0096 IF (ABS(SGNOPT).LT.1.0E-10) SGNOPT=-1.      00123
ISN 0098 WRITE (I6I,1070) IOBJ,SGNOPT               00124
ISN 0099 WRITE (I6I,760) IPRINT,ITMAX,ICNDIR,NSCAL,ITRM,LINOBJ,NACMX1,NFDG 00125
ISN 0100 WRITE (I6I,770) FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,DELFUN,DA00126
1UN,ALPHAX,ABOJJI                                    00127
ISN 0101 110 N2=NDV+3                                 00128
ISN 0102 N3=N2+NDV+2                                 00129
ISN 0103 N4=N3+NDV+2                                 00130
C --- DATA BLOCK F.                                  00131
C DESIGN VARIABLE INFORMATION, LB, UB, INITIAL VALUE, SCAL. 00132
ISN 0104 IF (IPNPUT.LT.2) WRITE (I6I,1080)           00133
ISN 0106 N5=N4+NDV+2                                 00134
ISN 0107 IF (N5.LE.NDRA) GO TO 120                   00135
ISN 0109 WRITE (I6I,780)                             00136
ISN 0110 WRITE (I6I,790)                             00137
ISN 0111 LOCR(25)=N5                                 00138
ISN 0112 GO TO 550                                   00139
ISN 0113 120 CONTINUE                                00140
ISN 0114 NSIDE=0                                     00141
ISN 0115 DO 130 I=1,NDV                              00142
ISN 0116 READ (ISCR1,1060) RA(N2),RA(N3),RA(I),RA(N4),(TITLE(J),J=1,5) 00143
ISN 0117 IF (RA(N2).GT.-1.0E+15.OR.RA(N3).LT.1.0E+15) NSIDE=1 00144
ISN 0119 IF (RA(N2).LE.-1.0E+15) RA(N2)=-1.1E+15   00145
ISN 0121 IF (RA(N3).GE.1.0E+15) RA(N3)=1.1E+15     00146
ISN 0123 IF (IPNPUT.LT.2)WRITE (I6I,1090)I,RA(N2),RA(N3),RA(I),RA(N4),(TITL00147
1E(J),J=1,5)                                         00148
ISN 0125 N2=N2+1                                     00149
ISN 0126 N3=N3+1                                     00150
ISN 0127 N4=N4+1                                     00151
ISN 0128 130 CONTINUE                                00152
C --- DATA BLOCK G.                                  00153
C D. V. NO., GLOBAL LOCATION, MULTIPLYING FACTOR.    00154
ISN 0129 IF (IPNPUT.LT.2) WRITE (I6I,930)           00155
ISN 0131 N5=4*NDV+9                                  00156
ISN 0132 M2=NDVTOT+1                                 00157
ISN 0133 N6=N5+NDVTOT                               00158
ISN 0134 M3=N2+NDVTOT                               00159
ISN 0135 IF (N6.LE.NDRA) GO TO 140                   00160
ISN 0137 WRITE (I6I,780)                             00161
ISN 0138 WRITE (I6I,800)                             00162
ISN 0139 LOCR(25)=N5                                 00163
ISN 0140 GO TO 550                                   00164
ISN 0141 140 CONTINUE                                00165
ISN 0142 IF (M3.LE.NDIA) GO TO 150                   00166
ISN 0144 WRITE (I6I,810)                             00167
ISN 0145 WRITE (I6I,800)                             00168
ISN 0146 LOCI(25)=M3                                 00169
ISN 0147 GO TO 550                                   00170
ISN 0148 150 CONTINUE                                00171
ISN 0149 DO 160 I=1,NDVTOT                           00172

```

+VERSION 1.3.0 (01 MAY 80) COPE01 SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)		DATE 82.141/10.54.12	PAGE 4
ISN 0150	READ (ISCR1,920) IA(M2),IA(I),RA(N5)	00173	
ISN 0151	IF (ABS(RA(N5)).LT.1.0E-20) RA(N5)=1.0	00174	
ISN 0153	IF (IPNPUT.LT.2) WRITE (I6I,940) I,IA(M2),IA(I),RA(N5)	00175	
ISN 0155	M2=M2+1	00176	
ISN 0156	N5=N5+1	00177	
ISN 0157	160 CONTINUE	00178	
ISN 0158	NCON=0	00179	
	C --- DATA BLOCK H.	00180	
	C NUMBER OF CONSTRAINT SETS.	00181	
ISN 0159	READ (ISCR1,920) NCONS	00182	
ISN 0160	IF (IPNPUT.LT.2) WRITE (I6I,1110)	00183	
ISN 0162	IF (IPNPUT.LT.2) WRITE (I6I,1120) NCONS	00184	
ISN 0164	IF (NCONS.EQ.0) GO TO 270	00185	
ISN 0166	IF (IPNPUT.LT.2) WRITE (I6I,1130)	00186	
ISN 0168	N6=4*NDV+NDVTOT+9	00187	
ISN 0169	M3=2*NDVTOT+1	00188	
ISN 0170	M4=2*NDVTOT+NCONS	00189	
ISN 0171	M4A=M4+1	00190	
ISN 0172	L=1	00191	
	C --- DATA BLOCK I.	00192	
ISN 0173	DO 240 I=1,NCONS	00193	
ISN 0174	NNN=N6+3	00194	
ISN 0175	IF (NNN.GT.NDRA) GO TO 250	00195	
	C GLOBAL NO. 1, GLOBAL NO. 2, LINEAR CONSTRAINT ID.	00196	
ISN 0177	READ (ISCR1,1200) ICONI,JCONI,LCONI	00197	
	C LB, NORM, UB, NORM.	00198	
ISN 0178	READ (ISCR1,1210) (RA(J),J=N6,NNN)	00199	
ISN 0179	IF (RA(N6).LE.-1.0E+15) RA(N6)=-1.1E+15	00200	
ISN 0181	IF (RA(N6+2).GE.1.0E+15) RA(N6+2)=1.1E+15	00201	
ISN 0183	IF (RA(N6+1).LT.1.0E-20) RA(N6+1)=ABS(RA(N6))	00202	
ISN 0185	IF (RA(N6+1).LT.1.0E-20) RA(N6+1)=0.1	00203	
ISN 0187	IF (RA(N6+3).LT.1.0E-20) RA(N6+3)=ABS(RA(N6+2))	00204	
ISN 0189	IF (RA(N6+3).LT.1.0E-20) RA(N6+3)=0.1	00205	
	C NUMBER OF VARIABLES IN THIS SET.	00206	
ISN 0191	NVAR=JCONI-ICONI+1	00207	
ISN 0192	IF (NVAR.LT.1) NVAR=1	00208	
ISN 0194	NCONA=NCONA+NVAR	00209	
	C HOW MANY CONSTRAINTS?	00210	
ISN 0195	J1=0	00211	
ISN 0196	IF (RA(N6).GE.-1.0E+15) J1=1	00212	
ISN 0193	IF (RA(N6+2).LT.1.0E+15) J1=J1+1	00213	
ISN 0200	NCONI=J1*NVAR	00214	
ISN 0201	NCON=NCON+NCONI	00215	
ISN 0202	IF (J1.EQ.0) GO TO 180	00216	
	C ADD LINEAR CONSTRAINT IDENTIFIERS TO ISC.	00217	
ISN 0204	DO 170 J=1,NCONI	00218	
ISN 0205	M4=M4+1	00219	
ISN 0206	MMM=M4	00220	
ISN 0207	IF (MMM.GT.NDIA) GO TO 260	00221	
ISN 0209	170 IA(M4)=LCONI	00222	
ISN 0210	180 CONTINUE	00223	
	C ADD LB, UB AND SCAL TO BLU IF NVAR.GT.1.	00224	
ISN 0211	IF (NVAR.EQ.1) GO TO 200	00225	
ISN 0213	NVAR1=NVAR-1	00226	
ISN 0214	DO 190 J=1,NVAR1	00227	
ISN 0215	NNN=N6+7	00228	
ISN 0216	IF (NNN.GT.NDRA) GO TO 250	00229	
ISN 0218	RA(N6+4)=RA(N6)	00230	
ISN 0219	RA(N6+5)=RA(N6+1)	00231	

258

```

ISN 0220      RA(N6+6)=RA(N6+2)          00232
ISN 0221      RA(N6+7)=RA(N6+3)          00233
ISN 0222      N6=N6+4                    00234
ISN 0223      190  CONTINUE                00235
ISN 0224      200  CONTINUE                00236
C             ADD CONSTRAINED VARIABLE GLOBAL IDENTIFIERS TO ICON. 00237
ISN 0225      ICON1=ICONI                 00238
ISN 0226      MMM=M4+NVAR-1              00239
ISN 0227      IF (MMM.GT.NDIA) GO TO 260  00240
ISN 0229      DO 230 J=1,NVAR             00241
ISN 0230      IF (J.EQ.1) GO TO 220      00242
C             SHIFT ISC VECTOR.          00243
ISN 0232      L1=M4+1                     00244
ISN 0233      L2=N4                       00245
ISN 0234      DO 210 K=M4A,M4            00246
ISN 0235      IA(L1)=IA(L2)              00247
ISN 0236      L1=L1-1                    00248
ISN 0237      210  L2=L2-1                00249
ISN 0238      M4=M4+1                    00250
ISN 0239      M4A=M4A+1                  00251
ISN 0240      220  IA(M3)=ICONI           00252
ISN 0241      ICON1=ICON1+1              00253
ISN 0242      230  M3=M3+1                00254
ISN 0243      IF (IPNPUT.LT.2)WRITE (I6I,1100)L,ICONI,JCONI,LCONI,RA(N6),RA(N6+1
00255
1),RA(N6+2),RA(N6+3)                    00256
ISN 0245      N6=N6+4                    00257
ISN 0246      L=NCON+1                   00258
ISN 0247      240  CONTINUE                00259
ISN 0248      IF (IPNPUT.LT.2) WRITE (I6I,900) NCONA 00260
ISN 0250      GO TO 270                   00261
ISN 0251      250  WRITE (I6I,780)         00262
ISN 0252      WRITE (I6I,820)            00263
ISN 0253      LOCR(25)=NNN               00264
ISN 0254      GO TO 550                   00265
ISN 0255      260  WRITE (I6I,810)         00266
ISN 0256      WRITE (I6I,820)            00267
ISN 0257      LOCI(25)=MMM               00268
ISN 0258      GO TO 550                   00269
ISN 0259      270  CONTINUE                00270
C             STARTING LOCATIONS FOR APPROXIMATION INFORMATION. 00271
ISN 0260      NAPR=4*NDV+NDVTOT+4*NCONA+9 00272
ISN 0261      NAPI=2*(NDV+NCONA)+2*NDVTOT+NCONA+1 00273
ISN 0262      NF=0                        00274
ISN 0263      KNAX=0                      00275
ISN 0264      NPTOT=0                     00276
ISN 0265      MAXTRM=0                    00277
ISN 0266      IF (NXAPRX.LE.0) GO TO 450  00278
C             ----- 00279
C             APPROXIMATE ANALYSIS/DESIGN 00280
C             ----- 00281
C --- DATA BLOCK J. 00282
C CONTROL PARAMETERS. 00283
ISN 0268      READ (ISCR1,1200) NF,NPS,NPFS,NPA,INOM,ISCRX,ISCRXF,IPAPRX 00284
ISN 0269      IF (NPA.NE.0) NPA=1         00285
ISN 0271      IF (NPS.EQ.0.AND.NPFS.EQ.0) NPA=1 00286
ISN 0273      IF (ISCRX.EQ.0) ISCRX=5     00287
ISN 0275      IF (ISCRXF.EQ.0) ISCRXF=5   00288
ISN 0277      IF (IPNPUT.LT.2)WRITE (I6I,600)NF,NPS,NPFS,NPA,INOM,ISCRX,ISCRXF,I
00289
1PAPRX 00290

```

```

+VERSION 1.3.0 (01 MAY 80)   COPE01   SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)   DATE 82.141/10.54.12   PAGE 6
ISN 0279      NPSFS=NPS+NPFS                               00291
ISN 0280      IF (NPSFS.LT.2) NPA=1                         00292
ISN 0282      NPTOT=NPS+NPFS+NPA                           00293
ISN 0283      IF (NPTOT.LT.2) NPTOT=2                      00294
ISN 0285      READ (ISCR1,1200) KMIN,KMAX,NPMAX,JNOM,INXLOC,INFLOC,MAXTRM 00295
ISN 0286      IF (INXLOC.EQ.0) NXAPRX=NDVOT                00296
ISN 0288      M=NXAPRX+(NXAPRX*(NXAPRX+1))/2               00297
ISN 0289      IF (NPMAX.LE.0) NPMAX=2*M                   00298
ISN 0291      IF (KMAX.EQ.0) KMAX=3*M-NPTOT+1              00299
ISN 0293      IF (KMIN.EQ.0) KMIN=2*NDV-NPTOT+1           00300
ISN 0295      IF (KMIN.LT.0) KMIN=0                        00301
ISN 0297      IF (KMAX.GT.0.AND.KMAX.LT.KMIN) KMAX=KMIN    00302
ISN 0299      IF (JNOM.EQ.0) JNOM=2*M                      00303
ISN 0301      IF (MAXTRM.LT.1) MAXTRM=3                    00304
ISN 0303      IF (IPNPUT.LT.2)WRITE (I6I,610)KMIN,KMAX,NPMAX,JNOM,INXLOC,INFLOC,00305
                IMAXTRM                                    00306
C --- DATA BLOCK K, PART 1.                               00307
C   DELX BOUNDS ON APPROXIMATE OPTIMIZATION.              00308
ISN 0305      IF (NDV.LE.0) GO TO 290                       00309
ISN 0307      N7=NAPR                                         00310
ISN 0308      NN7=N7+NDV-1                                    00311
ISN 0309      IF (NN7.LE.NDRA) GO TO 280                    00312
ISN 0311      WRITE (I6I,780)                                00313
ISN 0312      WRITE (I6I,560)                                00314
ISN 0313      280 CONTINUE                                   00315
ISN 0314      READ (ISCR1,1210) (RA(I),I=N7,NN7)           00316
ISN 0315      IF (IPNPUT.LT.2) WRITE (I6I,570)             00317
ISN 0317      IF (IPNPUT.LT.2) WRITE (I6I,1160) (RA(I),I=N7,NN7) 00318
C --- DATA BLOCK K, PART 2.                               00319
C   MULTIPLIERS ON DELX.                                   00320
ISN 0319      READ (ISCR1,1210) XFACT1,XFACT2               00321
ISN 0320      IF (XFACT1.LT.1.0E-10) XFACT1=1.5           00322
ISN 0322      IF (XFACT2.LT.1.0E-10) XFACT2=2.            00323
ISN 0324      IF (IPNPUT.LT.2) WRITE (I6I,620) XFACT1,XFACT2 00324
ISN 0326      290 CONTINUE                                   00325
C --- DATA BLOCK L.                                       00326
C   GLOBAL LOCATIONS OF X-VARIABLES.                       00327
ISN 0327      M5=NAPI                                         00328
ISN 0328      MM5=M5+NXAPRX-1                                00329
ISN 0329      IF (MM5.LE.NDIA) GO TO 300                    00330
ISN 0331      WRITE (I6I,760)                                00331
ISN 0332      WRITE (I6I,630)                                00332
ISN 0333      LOCI(25)=MM5                                    00333
ISN 0334      GO TO 550                                       00334
ISN 0335      300 CONTINUE                                   00335
ISN 0336      IF (INXLOC.EQ.0) GO TO 310                    00336
ISN 0338      READ (ISCR1,1200) (IA(I),I=M5,MM5)           00337
ISN 0339      GO TO 330                                       00338
ISN 0340      310 CONTINUE                                   00339
C   X-LOCATIONS ARE DEFAULTED TO DESIGN VARIABLE LOCATIONS. 00340
ISN 0341      DO 320 I=1,NXAPRX                               00341
ISN 0342      IA(M5)=IA(I)                                    00342
ISN 0343      320 M5=M5+1                                      00343
ISN 0344      M5=NAPI                                         00344
ISN 0345      330 CONTINUE                                   00345
ISN 0346      IF (IPNPUT.LT.2) WRITE (I6I,640)             00346
ISN 0348      IF (IPNPUT.LT.2) WRITE (I6I,1160) (IA(I),I=M5,MM5) 00347
C --- DATA BLOCK M.                                       00348
C   GLOBAL LOCATIONS OF FUNCTIONS.                          00349

```

260

```

ISN 0350      M6=NAPI+NXAPRX                      00350
ISN 0351      MM6=M6+NF-1                          00351
ISN 0352      IF (MM6.LE.NDIA) GO TO 340            00352
ISN 0354      WRITE (I6I,780)                       00353
ISN 0355      WRITE (I6I,650)                       00354
ISN 0356      LOCI(25)=MM6                          00355
ISN 0357      GO TO 550                              00356
ISN 0358      340 CONTINUE                           00357
ISN 0359      IF (INFLOC.EQ.0) GO TO 350            00358
ISN 0361      READ (ISCR1,1200) (IA(I),I=M6,MM6)    00359
ISN 0362      GO TO 380                              00360
ISN 0363      350 CONTINUE                           00361
C            FUNCTION LOCATIONS ARE DEFAULTED TO OBJECTIVE AND CONSTRAINT
C            LOCATIONS.                             00362
ISN 0364      NF1=1                                  00364
ISN 0365      M3=2*NDVTOT+1                          00365
ISN 0366      IA(M6)=IOBJ                             00366
ISN 0367      IF (NCONA.EQ.0) GO TO 370            00367
ISN 0369      DO 360 I=1,NCONA                       00368
ISN 0370      IF (IA(M3).EQ.IOBJ) GO TO 360         00369
ISN 0372      NF1=NF1+1                              00370
ISN 0373      M6=M6+1                                00371
ISN 0374      IA(M6)=IA(M3)                          00372
ISN 0375      360 M3=M3+1                             00373
ISN 0376      370 NF=NF1                             00374
ISN 0377      380 M6=NAPI+NXAPRX                     00375
ISN 0378      MM6=M6+NF-1                            00376
ISN 0379      IF (IPNPUT.LT.2) WRITE (I6I,660)     00377
ISN 0381      IF (IPNPUT.LT.2) WRITE (I6I,1180) (IA(I),I=M6,MM6) 00378
C            DATA BLOCK N.                         00379
C            READ INPUT X-VECTORS AND STORE ON UNIT ISCR2. 00380
ISN 0383      REWIND ISCR2                           00381
ISN 0384      IF (NPS.EQ.0) GO TO 410               00382
ISN 0386      N7=NAPR+NDV                             00383
ISN 0387      NN7=N7+NXAPRX-1                       00384
ISN 0388      IF (NN7.LE.NDRA) GO TO 390           00385
ISN 0390      WRITE (I6I,780)                       00386
ISN 0391      WRITE (I6I,670)                       00387
ISN 0392      LOCR(25)=NN7                          00388
ISN 0393      GO TO 550                              00389
ISN 0394      390 CONTINUE                           00390
ISN 0395      IF (IPNPUT.LT.2) WRITE (I6I,680) ISCRX 00391
ISN 0397      DO 400 I=1,NPS                         00392
C            BINARY READ IF ISCRX.NE.5.             00393
ISN 0398      IF (ISCRX.NE.5) READ (ISCRX) (RA(J),J=N7,NN7) 00394
C            FORMATTED READ IF ISCRX.EQ.5.           00395
ISN 0400      IF (ISCRX.EQ.5) READ (ISCR1,1210) (RA(J),J=N7,NN7) 00396
ISN 0402      WRITE (ISCR2) (RA(J),J=N7,NN7)        00397
ISN 0403      IF (IPNPUT.LT.2) WRITE (I6I,710) I,I  00398
ISN 0405      IF (IPNPUT.LT.2) WRITE (I6I,1160) (RA(J),J=N7,NN7) 00399
ISN 0407      400 CONTINUE                           00400
ISN 0408      410 CONTINUE                           00401
C            DATA BLOCK O.                         00402
C            READ INPUT X-F PAIRS AND STORE ON UNIT ISCR2. 00403
ISN 0409      IF (NPFS.EQ.0) GO TO 440               00404
ISN 0411      N7=NAPR+NDV                             00405
ISN 0412      NN7=N7+NXAPRX-1                       00406
ISN 0413      NN8=N7+NF-1                            00407
ISN 0414      IF (NN7.GT.NN8) NN8=NN7                00408

```

+VERSION 1.3.0 (01 MAY 80) COPE01 SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)		DATE 82.141/10.54.12	PAGE 8
ISN 0416	IF (NN8.LE.NDRA) GO TO 420	00409	
ISN 0418	WRITE (I6I,780)	00410	
ISN 0419	WRITE (I6I,690)	00411	
ISN 0420	LOCR(25)=NN8	00412	
ISN 0421	GO TO 550	00413	
ISN 0422	420 CONTINUE	00414	
ISN 0423	NN8=N7+NF-1	00415	
ISN 0424	IF (IPNPUT.LT.2) WRITE (I6I,700) ISCRXF	00416	
ISN 0426	DO 430 I=1,NPFS	00417	
	C X-VECTOR.	00418	
	C BINARY READ IF ISCRXF.NE.5.	00419	
ISN 0427	IF (ISCRXF.NE.5) READ (ISCRXF) (RA(J),J=N7,NN7)	00420	
	C FORMATTED READ IF ISCRXF.EQ.5.	00421	
ISN 0429	IF (ISCRXF.EQ.5) READ (ISCR1,1210) (RA(J),J=N7,NN7)	00422	
ISN 0431	II=I+NPS	00423	
ISN 0432	IF (IPNPUT.LT.2) WRITE (I6I,710) I,II	00424	
ISN 0434	IF (IPNPUT.LT.2) WRITE (I6I,720)	00425	
ISN 0436	IF (IPNPUT.LT.2) WRITE (I6I,1160) (RA(J),J=N7,NN7)	00426	
ISN 0438	WRITE (ISCR2) (RA(J),J=N7,NN7)	00427	
	C FUNCTION VALUES.	00428	
	C BINARY READ IF ISCRXF.NE.5.	00429	
ISN 0439	IF (ISCRXF.NE.5) READ (ISCRXF) (RA(J),J=N7,NN8)	00430	
	C FORMATTED READ IF ISCRXF.EQ.5.	00431	
ISN 0441	IF (ISCRXF.EQ.5) READ (ISCR1,1210) (RA(J),J=N7,NN8)	00432	
ISN 0443	IF (IPNPUT.LE.2) WRITE (I6I,730)	00433	
ISN 0445	IF (IPNPUT.LT.2) WRITE (I6I,1160) (RA(J),J=N7,NN8)	00434	
ISN 0447	WRITE (ISCR2) (RA(J),J=N7,NN8)	00435	
ISN 0448	430 CONTINUE	00436	
ISN 0449	440 CONTINUE	00437	
ISN 0450	450 CONTINUE	00438	
ISN 0451	NSOBJ=0	00439	
ISN 0452	NSVTOT=0	00440	
	C STARTING LOCATIONS FOR SENSITIVITY INFORMATION.	00441	
ISN 0453	NSVR=NAPR+NDV	00442	
ISN 0454	NSVI=NAPI+NXAPRX+NF	00443	
ISN 0455	IF (NSV.LE.0) GO TO 500	00444	
	C -----00445		
	C SENSITIVITY INFORMATION	00446	
	C -----00447		
ISN 0457	IF (IPNPUT.LT.2) WRITE (I6I,1020)	00448	
	C --- DATA BLOCK P, PART 1.	00449	
	C NSOBJ, IPSENS	00450	
ISN 0459	READ (ISCR1,1200) NSOBJ,IPSENS	00451	
	C --- DATA BLOCK P, PART 2.	00452	
	C NSENSZ.	00453	
ISN 0460	M15=NSVI	00454	
ISN 0461	MM15=M15+NSOBJ-1	00455	
ISN 0462	IF (MM15.LE.NDIA) GO TO 460	00456	
ISN 0464	WRITE (I6I,810)	00457	
ISN 0465	WRITE (I6I,830)	00458	
ISN 0466	LOCI(25)=MM15	00459	
ISN 0467	GO TO 550	00460	
ISN 0468	460 CONTINUE	00461	
ISN 0469	READ (ISCR1,1200) (IA(I),I=M15,MM15)	00462	
ISN 0470	IF (IPNPUT.LT.2) WRITE (I6I,960) IPSENS,NSOBJ	00463	
ISN 0472	IF (IPNPUT.LT.2) WRITE (I6I,950) (IA(I),I=M15,MM15)	00464	
ISN 0474	IF (IPNPUT.LT.2) WRITE (I6I,1030)	00465	
ISN 0476	M15=NSVR	00466	
ISN 0477	M16=NSVI+NSOBJ	00467	

262

```

ISN 0478      M17=M16+NSV                00468
ISN 0479      DO 490 I=1,NSV            00469
              C --- DATA BLOCK Q, PART 1. 00470
              C      ISENS, NSENS.       00471
ISN 0480      READ (ISCR1,1200) IA(M16),NN1 00472
ISN 0481      NN15=NN1+NN1-1           00473
ISN 0482      IF (NN15.LE.NDRA) GO TO 470 00474
ISN 0484      WRITE (I6I,780)          00475
ISN 0485      WRITE (I6I,840)          00476
ISN 0486      LOCR(25)=NN15            00477
ISN 0487      GO TO 550                 00478
ISN 0488      470 CONTINUE              00479
              C --- DATA BLOCK Q, PART 2. 00480
              C      SENS.                00481
ISN 0489      READ (ISCR1,1210) (RA(J),J=N15,NN15) 00482
ISN 0490      IF (IPNPUT.GE.2) GO TO 480 00483
ISN 0492      JJ=N15+5                  00484
ISN 0493      IF (JJ.GT.NN15) JJ=NN15   00485
ISN 0495      WRITE (I6I,1040) I,IA(M16),(RA(J),J=N15,JJ) 00486
ISN 0496      JJ=JJ+1                  00487
ISN 0497      IF (JJ.LE.NN15) WRITE (I6I,1050) (RA(J),J=JJ,NN15) 00488
ISN 0499      480 CONTINUE              00489
ISN 0500      NSVTOT=NSVTOT+NN1         00490
ISN 0501      IA(M17)=NN1               00491
ISN 0502      N15=NN15+1                00492
ISN 0503      M16=M16+1                 00493
ISN 0504      M17=M17+1                 00494
ISN 0505      490 CONTINUE              00495
ISN 0506      500 CONTINUE              00496
ISN 0507      M2VX=0                    00497
ISN 0508      M2VY=0                    00498
              C      STARTING LOCATIONS FOR TWO-VARIABLE FUNCTION SPACE INFORMATION. 00499
ISN 0509      N2VR=NSVR+NSVTOT          00500
ISN 0510      N2VI=NSVI+NSOBJ+2*NSV     00501
ISN 0511      IF (N2VAR.LE.0) GO TO 540 00502
              C      -----00503
              C      TWO-VARIABLE FUNCTION SPACE INFORMATION 00504
              C      -----00505
              C --- DATA BLOCK R.       00506
              C      VARIABLE NUMBERS AND NUMBER OF VALUES OF X AND Y. 00507
ISN 0513      READ (ISCR1,1200) N2VX,M2VX,N2VY,M2VY,IP2VAR 00508
ISN 0514      N20=N2VR                  00509
ISN 0515      M20=N2VI                  00510
ISN 0516      MM20=M20+N2VAR-1          00511
ISN 0517      IF (MM20.LE.NDIA) GO TO 510 00512
ISN 0519      WRITE (I6I,810)           00513
ISN 0520      WRITE (I6I,850)           00514
ISN 0521      LOCI(25)=MM20            00515
ISN 0522      GO TO 550                 00516
ISN 0523      510 CONTINUE              00517
              C --- DATA BLOCK S.       00518
              C      GLOBAL VARIABLE NUMBERS CORRESPONDING TO FUNCTIONS OF X AND Y. 00519
ISN 0524      READ (ISCR1,1200) (IA(I),I=M20,MM20) 00520
ISN 0525      IF (IPNPUT.LT.2) WRITE (I6I,1170) IP2VAR 00521
ISN 0527      IF (IPNPUT.LT.2) WRITE (I6I,1180) (IA(I),I=M20,MM20) 00522
              C --- DATA BLOCK T.       00523
              C      VALUES OF X COMPONENTS. 00524
ISN 0529      NN20=N20+M2VX-1           00525
ISN 0530      IF (NN20.LE.NDRA) GO TO 520 00526
    
```

+VERSION 1.3.0 (01 MAY 80) COPE01 SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)		DATE 82.141/10.54.12	PAGE 10
ISN 0532	WRITE (I6I,780)	00527	
ISN 0533	WRITE (I6I,740)	00528	
ISN 0534	LOCR(25)=NN20	00529	
ISN 0535	GO TO 550	00530	
ISN 0536	520 READ (ISCR1,1210) (RA(I),I=N20,NN20)	00531	
ISN 0537	IF (IPNPUT.LT.2) WRITE (I6I,1140) N2VX	00532	
ISN 0539	IF (IPNPUT.LT.2) WRITE (I6I,1160) (RA(I),I=N20,NN20)	00533	
	C --- DATA BLOCK U.	00534	
	C VALUES OF Y COMPONENTS.	00535	
ISN 0541	N21=N20+M2VX	00536	
ISN 0542	NN21=N21+M2VY-1	00537	
ISN 0543	IF (NN21.LE.NDRA) GO TO 530	00538	
ISN 0545	WRITE (I6I,780)	00539	
ISN 0546	WRITE (I6I,750)	00540	
ISN 0547	LOCR(25)=NN21	00541	
ISN 0548	GO TO 550	00542	
ISN 0549	530 CONTINUE	00543	
ISN 0550	NN20=NN21	00544	
ISN 0551	READ (ISCR1,1210) (RA(I),I=N21,NN21)	00545	
ISN 0552	IF (IPNPUT.LT.2) WRITE (I6I,1150) N2VY	00546	
ISN 0554	IF (IPNPUT.LT.2) WRITE (I6I,1160) (RA(I),I=N21,NN21)	00547	
ISN 0556	540 CONTINUE	00548	
	C -----	00549	
	C DYNAMIC STORAGE ALLOCATION	00550	
	C -----	00551	
ISN 0557	NDV2=NDV+2	00552	
	C REAL VARIABLES.	00553	
	C X.	00554	
ISN 0558	LOCR(1)=1	00555	
	C VLB.	00556	
ISN 0559	LOCR(2)=NDV+3	00557	
	C VUB.	00558	
ISN 0560	LOCR(3)=LOCR(2)+NDV2	00559	
	C SCAL.	00560	
ISN 0561	LOCR(4)=LOCR(3)+NDV2	00561	
	C AMULT.	00562	
ISN 0562	LOCR(5)=LOCR(4)+NDV2	00563	
	C BLU.	00564	
ISN 0563	LOCR(6)=LOCR(5)+NDVTOT	00565	
	C DELX.	00566	
ISN 0564	LOCR(7)=LOCR(6)+4*NCONA	00567	
ISN 0565	LOCR(8)=LOCR(7)+NDV	00568	
	C SENS.	00569	
ISN 0566	LOCR(15)=LOCR(8)	00570	
	C XM2V.	00571	
ISN 0567	LOCR(20)=LOCR(15)+NSVTOT	00572	
	C YM2V.	00573	
ISN 0568	LOCR(21)=LOCR(20)+M2VX	00574	
ISN 0569	LOCR(22)=LOCR(21)+M2VY	00575	
	C START OF EXECUTION STORAGE.	00576	
ISN 0570	LOCR(23)=LOCR(22)	00577	
	C INTEGER VARIABLES.	00578	
	C IDSGN.	00579	
ISN 0571	LOCI(1)=1	00580	
	C NDSGN.	00581	
ISN 0572	LOCI(2)=NDVTOT+1	00582	
	C ICON.	00583	
ISN 0573	LOCI(3)=LOCI(2)+NDVTOT	00584	
	C ISC.	00585	

264

ISN 0574		LOCI(4)=LOCI(3)+NCONA	00586
	C	LOCX.	00587
ISN 0575		LOCI(5)=LOCI(4)+2*(NDV+NCONA)	00588
	C	LOCF.	00589
ISN 0576		LOCI(6)=LOCI(5)+NXAPRX	00590
ISN 0577		LOCI(7)=LOCI(6)+NF	00591
	C	NSENSZ.	00592
ISN 0578		LOCI(15)=LOCI(7)	00593
	C	ISENS.	00594
ISN 0579		LOCI(16)=LOCI(15)+NSOBJ	00595
	C	NSENS.	00596
ISN 0580		LOCI(17)=LOCI(16)+NSV	00597
ISN 0581		LOCI(18)=LOCI(17)+NSV	00598
	C	N2VZ.	00599
ISN 0582		LOCI(20)=LOCI(18)	00600
ISN 0583		LOCI(21)=LOCI(20)+N2VAR	00601
	C	START OF EXECUTION STORAGE.	00602
ISN 0584		LOCI(23)=LOCI(21)	00603
	C	EXECUTION STORAGE REQUIREMENTS.	00604
ISN 0585		NRI=NDV	00605
ISN 0586		IF (NACMX1.GT.NRI) NRI=NACMX1	00606
ISN 0588		NR2=3*NCON+12*NDV+NACMX1*(NDV+NACMX1)+3*NRI+12	00607
ISN 0589		NI2=NACMX1+2*NRI+2*NDV+NCON	00608
ISN 0590		NR3=NSV	00609
ISN 0591		IF (NSOBJ.GT.NR3) NR3=NSOBJ	00610
ISN 0593		NR4=N2VAR	00611
ISN 0594		NR5=NR2+NR3	00612
ISN 0595		NI5=NI2	00613
ISN 0596		M=NXAPRX+(NXAPRX*(NXAPRX+1))/2	00614
ISN 0597		IF (MAXTRM.LT.3) M=MAXTRM*NXAPRX	00615
ISN 0599		NR6=3*NXAPRX+6+NDV+2*NF+M*NF	00616
ISN 0600		NI6=NCONA+NXAPRX	00617
ISN 0601		NRI=(KMAX+1-NPTOT)*(NXAPRX+NF+1)	00618
ISN 0602		IF (NRI.LT.NR2) NRI=NR2	00619
ISN 0604		IF (KMAX.LT.0) NRI=NPTOT*(NXAPRX+NF+1)	00620
ISN 0606		NR7=NR6+NRI	00621
ISN 0607		NI7=NI6+NI2	00622
	C	START OF TEMPGRARY STORAGE.	00623
ISN 0608		LOCR(24)=LOCR(23)	00624
ISN 0609		LOCI(24)=LOCI(23)	00625
ISN 0610		IF (NCALC.EQ.2) LOCR(24)=LOCR(23)+NR2	00626
ISN 0612		IF (NCALC.EQ.3) LOCR(24)=LOCR(23)+NR3	00627
ISN 0614		IF (NCALC.EQ.4) LOCR(24)=LOCR(23)+NR4	00628
ISN 0616		IF (NCALC.EQ.5) LOCR(24)=LOCR(23)+NR5	00629
ISN 0618		IF (NCALC.EQ.6.AND.KMAX.LT.0) LOCR(24)=LOCR(23)+NR6	00630
ISN 0620		IF (NCALC.EQ.6.AND.KMAX.GT.0) LOCR(24)=LOCR(23)+NR7	00631
ISN 0622		IF (NCALC.EQ.2) LOCI(24)=LOCI(23)+NI2	00632
ISN 0624		IF (NCALC.EQ.5) LOCI(24)=LOCI(23)+NI5	00633
ISN 0626		IF (NCALC.EQ.6.AND.KMAX.LT.0) LOCI(24)=LOCI(23)+NI6	00634
ISN 0628		IF (NCALC.EQ.6.AND.KMAX.GT.0) LOCI(24)=LOCI(23)+NI7	00635
	C	TOTAL STORAGE REQUIREMENTS.	00636
ISN 0630		LOCR(25)=LOCR(24)	00637
ISN 0631		LOCI(25)=LOCI(24)	00638
ISN 0632		IF (NCALC.EQ.5) LOCR(25)=LOCR(25)+4*NDV+8	00639
ISN 0634		IF (NCALC.EQ.5) LOCI(25)=LOCI(25)+2*NDVTOT	00640
ISN 0636		IF (IPNPUT.LT.2)WRITE (I6I,860)LOCR(23),LOCR(25),NDRA,LOCI(23),LOC	00641
		II(25),NDIA	00642
ISN 0638	550	CONTINUE	00643
ISN 0639		RETURN	00644

	C	-----	00645
	C	FORMATS	00646
	C	-----	00647
ISN 0640	560	FORMAT (/5X,27HUNABLE TO READ DATA BLOCK K)	00648
ISN 0641	570	FORMAT (/5X,43HDELTA-X BOUNDS FOR APPROXIMATE OPTIMIZATION)	00649
ISN 0642	580	FORMAT (80A1)	00650
ISN 0643	590	FORMAT (I5/80A1)	00651
ISN 0644	600	FORMAT (///5X,49H* * APPROXIMATE ANALYSIS/OPTIMIZATION INFORMATION	00652
		1//5X,38HNUMBER OF FUNCTIONS APPROXIMATED, NF =,I5/5X,38HNUMBER OF	00653
		2INPUT X-VECTORS, NPS =,I5/5X,38HNUMBER OF INPUT X-F PAIRS,	00654
		3 NPFS =,I5/5X,38HX-VECTOR FROM ANALIZ, NPA =,I5/5X,300655	00655
		48HNOMINAL DESIGN, INOM =,I5/5X,38HREAD UNIT FOR X-	00656
		5VECTORS, ISCRX =,I5/5X,38HREAD UNIT FOR X-F PAIRS, ISCR	00657
		6XF =,I5/5X,38HPRINT CONTROL, IPAPRX =,I5)	00658
ISN 0645	610	FORMAT (/5X,38HMINIMUM APPROXIMATING CYCLES, KMIN =,I5/5X,38HMAX	00659
		1INUM APPROXIMATING CYCLES, KMAX =,I5/5X,38HMAXIMUM DESIGNS USED	00660
		2IN FIT, NPMAX =,I5/5X,38HNOMINAL DESIGN PARAMETER, JNOM =,	00661
		3I5/5X,38HX-LOCATION INPUT PARAMETER, INXLOC =,I5/5X,38HF-LOCATIO	00662
		4N INPUT PARAMETER, INFLOC =,I5/5X,38HTAYLER SERIES I.D. CODE,	00663
		5 MAXTRM =,I5)	00664
ISN 0646	620	FORMAT (/5X,38HMULTIPLIER ON DELX, XFACT1 =,E12.4/5X,38H	00665
		1MULTIPLIER ON DELX, XFACT2 =,E12.4)	00666
ISN 0647	630	FORMAT (/5X,27HUNABLE TO READ DATA BLOCK L)	00667
ISN 0648	640	FORMAT (/5X,31HGLOBAL LOCATIONS OF X-VARIABLES)	00668
ISN 0649	650	FORMAT (/5X,27HUNABLE TO READ DATA BLOCK M)	00669
ISN 0650	660	FORMAT (/5X,29HGLOBAL LOCATIONS OF FUNCTIONS)	00670
ISN 0651	670	FORMAT (/5X,27HUNABLE TO READ DATA BLOCK N)	00671
ISN 0652	680	FORMAT (/5X,25HX-VECTORS INPUT FROM UNIT,I5)	00672
ISN 0653	690	FORMAT (/5X,27HUNABLE TO READ DATA BLOCK O)	00673
ISN 0654	700	FORMAT (///5X,25HX-F PAIRS INPUT FROM UNIT,I5)	00674
ISN 0655	710	FORMAT (/5X,6HNUMBER,I5,5X,6HDESIGN,I5)	00675
ISN 0656	720	FORMAT (/5X,8HX-VECTOR)	00676
ISN 0657	730	FORMAT (/5X,15HFUNCTION VALUES)	00677
ISN 0658	740	FORMAT (/5X,27HUNABLE TO READ DATA BLOCK T)	00678
ISN 0659	750	FORMAT (/5X,27HUNABLE TO READ DATA BLOCK U)	00679
ISN 0660	760	FORMAT (/5X,58HCONMIN PARAMETERS (IF ZERO, CONMIN DEFAULT WILL OVE	00680
		1R-RIDE)/5X,6HIFPRINT,2X,5HITMAX,3X,6HICNDR,3X,5HNSCAL,3X,4HITRM,300681	00681
		2X,6HLINOBJ,2X,6HNCMX1,3X,4HNFDS/8I8)	00682
ISN 0661	770	FORMAT (/6X,4HFDCH,12X,5HFDCHM,11X,2HCT,14X,5HCTMIN/1X,4(2X,E14.5)	00683
		1//6X,3HCTL,13X,6HCTLMIN,10X,5HTHETA,11X,3HPHI/1X,4(2X,E14.5)//6X,600684	00684
		2HDEFUN,10X,6HDABFUN,10X,6HALPHAX,10X,6HABOBJ1/1X,4(2X,E14.5))	00685
ISN 0662	780	FORMAT (/5X,54HREQUIRED STORAGE IN ARRAY RA EXCEEDS AVAILABLE STO	00686
		1RAGE)	00687
ISN 0663	790	FORMAT (/5X,27HUNABLE TO READ DATA BLOCK F)	00688
ISN 0664	800	FORMAT (/5X,27HUNABLE TO READ DATA BLOCK G)	00689
ISN 0665	810	FORMAT (/5X,54HREQUIRED STORAGE IN ARRAY IA EXCEEDS AVAILABLE STO	00690
		1RAGE)	00691
ISN 0666	820	FORMAT (/5X,27HUNABLE TO READ DATA BLOCK I)	00692
ISN 0667	830	FORMAT (/5X,27HUNABLE TO READ DATA BLOCK P)	00693
ISN 0668	840	FORMAT (/5X,27HUNABLE TO READ DATA BLOCK Q)	00694
ISN 0669	850	FORMAT (/5X,27HUNABLE TO READ DATA BLOCK R)	00695
ISN 0670	860	FORMAT (///5X,39H* * ESTIMATED DATA STORAGE REQUIREMENTS//15X,4HR	00696
		1EAL,26X,7HINTEGER/5X,27HINPUT EXECUTION AVAILABLE,5X,27HINPUT	00697
		2EXECUTION AVAILABLE/I9,2I10,2X,3I10)	00698
ISN 0671	870	FORMAT (1H1,4X,27HCARD IMAGES OF CONTROL DATA//5X,4HCARD,20X,5HIM	00699
		1AGE)	00700
ISN 0672	880	FORMAT (1H0)	00701
ISN 0673	890	FORMAT (I8,1H),2X,80A1)	00702
ISN 0674	900	FORMAT (/5X,40HTOTAL NUMBER OF CONSTRAINED PARAMETERS =,I5)	00703

266

```

ISN 0675 910  FORMAT (//5X,26HCALCULATION CONTROL, NCALC/5X,5HVALUE,3X,7HMEANING00704
1/7X,1H1,5X,15HSINGLE ANALYSIS/7X,1H2,5X,12HOPTIMIZATION/7X,1H3,5X,00705
211HSENSITIVITY/7X,1H4,5X,27HTWO-VARIABLE FUNCTION SPACE/7X,1H5,5X,00706
319HOPTIMUM SENSITIVITY/7X,1H6,5X,24HAFFROXIMATE OPTIMIZATION) 00707
ISN 0676 920  FORMAT (2I10,F10.2) 00708
ISN 0677 930  FORMAT (//5X,16HDESIGN VARIABLES/11X,5HD. V.,5X,6HGLOBAL,4X,11HMUL00709
1TIPLYING/5X,2HID,5X,3HNO.,5X,6HVAR. NO.,5X,6HFACTOR) 00710
ISN 0678 940  FORMAT (2I7,5X,I5,6X,E12.5) 00711
ISN 0679 950  FORMAT (5X,16I5) 00712
ISN 0680 960  FORMAT (/5X,34HPRINT CONTROL, IPSENS =,I5/5X,34HNUMBER 00713
1OF SENSITIVITY OBJECTIVES =,I5//5X,53HGLOBAL NUMBERS ASSOCIATED WI00714
2TH SENSITIVITY OBJECTIVES) 00715
ISN 0681 970  FORMAT (1H1,////,5X,47HC CCCCCC 000000 P P P P P P P EEEEEEE S00716
1SSSSSS/5X,47HC O O P P E S /5X,4700717
2HC O O P P E S /5X,47HC 000718
3 O P P P P P P EEEE SSSSSSS/5X,47HC O O P 00719
4 E S/5X,47HC O O P E 00720
5 S/5X,47HC CCCCCC 000000 P EEEEEEE SSSSSSSC0721
6) 00722
ISN 0682 980  FORMAT (////14X,29HC O N T R O L P R O G R A M//26X,5HF O R//8X00723
1,41HE N G I N E E R I N G S Y N T H E S I S) 00724
ISN 0683 990  FORMAT (////24X,9HT I T L E//5X,20A4) 00725
ISN 0684 1000  FORMAT (1H1,4X,6HTITLE:/5X,20A4) 00726
ISN 0685 1010  FORMAT (//5X,19HCONTROL PARAMETERS;/5X,42HCALCULATION CONTROL, 00727
1 NCALC =,I5/5X,42HNUMBER OF GLOBAL DESIGN VARIABLES, 00728
2NDV =,I5/5X,42HNUMBER OF SENSITIVITY VARIABLES, NSV =,I5/5X,4200729
3HNUMBER OF FUNCTIONS IN TWO-SPACE, N2VAR =,I5/5X,42HNUMBER OF APP00730
4ROXIMATING VAR. NXAPRX =,I5/5X,42HINPUT INFORMATION PRINT COD00731
5E, IPNPUT =,I5/5X,42HDEBUG PRINT CODE, IFDEG 00732
6=,I5) 00733
ISN 0686 1020  FORMAT (///5X,27H* * SENSITIVITY INFORMATION) 00734
ISN 0687 1030  FORMAT (/14X,6HGLOBAL,4X,7HNOMINAL/5X,6HNUMBER,2X,8HVARIABLE,4X,5H00735
1VALUE,6X,18HOFF-NOMINAL VALUES) 00736
ISN 0688 1040  FORMAT (5X,I4,I8,5X,E12.5,1X,5E11.4) 00737
ISN 0689 1050  FORMAT (35X,5E11.4) 00738
ISN 0690 1060  FORMAT (4F10.2,10A4) 00739
ISN 0691 1070  FORMAT (///5X,28H* * OPTIMIZATION INFORMATION//5X,35HGLOBAL VARIA00740
1BLE NUMBER OF OBJECTIVE,10X,1H=,I5/5X,46HMULTIPLIER (NEGATIVE INDI00741
2CATES MINIMIZATION) =,E12.4) 00742
ISN 0692 1080  FORMAT (/5X,27HDESIGN VARIABLE INFORMATION/5X,50HNON-ZERO INITIAL 00743
1VALUE WILL OVER-RIDE MODULE INPUT/5X,5HD. V.,5X,5HLOWER,10X,5HUPPE00744
2R,9X,7HINITIAL/5X,3HNO.,7X,5HSOUND,10X,5HBOUND,10X,5HVALUE,10X,5HS00745
3SCALE) 00746
ISN 0693 1090  FORMAT (I8,4X,E12.5,3X,E12.5,3X,E12.5,3X,E12.5,5A4) 00747
ISN 0694 1100  FORMAT (I8,I7,2I8,5X,E12.5,3X,E12.5,3X,E12.5,3X,E12.5) 00748
ISN 0695 1110  FORMAT (//5X,22HCONSTRAINT INFORMATION) 00749
ISN 0696 1120  FORMAT (/5X,9HTHERE ARE,I3,16H CNSTRAINT SETS) 00750
ISN 0697 1130  FORMAT (11X,6HGLOBAL,2X,6HGLOBAL,2X,6HLINEAR,6X,5HLOWER,6X,13HNORM00751
1IALIZATION,6X,5HUPPER,6X,13HNORMALIZATION/6X,2HID,3X,6HVAR. 1,2X,6H00752
2VAR. 2,4X,2HID,8X,5HBOUND,9X,6HFACTOR,10X,5HBOUND,9X,6HFACTOR) 00753
ISN 0698 1140  FORMAT (//5X,49HGLOBAL VARIABLE NUMBER CORRESPONDING TO X, N2VX =,00754
1I5//5X,20HVALUES OF X-VARIABLE) 00755
ISN 0699 1150  FORMAT (//5X,49HGLOBAL VARIABLE NUMBER CORRESPONDING TO Y, N2VY =,00756
1I5//5X,20HVALUES OF Y-VARIABLE) 00757
ISN 0700 1160  FORMAT (3X,5E12.4) 00758
ISN 0701 1170  FORMAT (///5X,51H* * TWO-VARIABLE FUNCTION SPACE MAPPING INFORMAT00759
1ION//5X,23HPRINT CONTROL, IP2VAR =,I5//5X,52HGLOBAL VARIABLE NUMBE00760
2RS ASSOCIATED WITH F(X,Y), M2VZ) 00761
ISN 0702 1180  FORMAT (5X,10I5) 00762
    
```

+VERSION 1.3.0 (01 MAY 80) COPE01 SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 82.141/10.54.12 PAGE 14
ISN 0703 1190 FORMAT (20A4) 00763
ISN 0704 1200 FORMAT (8I10) 00764
ISN 0705 1210 FORMAT (8F10.2) 00765
ISN 0706 1220 FORMAT (///5X,26H* * * INPUT ERROR, NCALC =,I2,2X,4IHIS LT.0 OR GT00766
1.6 PROGRAM TERMINATED * * *) 00767
ISN 0707 END 00768
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NCMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
STATISTICS SOURCE STATEMENTS = 706, PROGRAM SIZE = 15790, SUBPROGRAM NAME =COPE01
STATISTICS NO DIAGNOSTICS GENERATED
***** END OF COMPILATION ***** 2844K BYTES OF CORE NOT USED

REQUESTED OPTIONS: SOURCE,NOMAP,NOXREF,NOLIST,NODECK,OPT(3),AUTOD9L(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

268

```

C          DATA SET U477COPE02 AT LEVEL 001 AS OF 02/13/81
C          DATA SET 9188COPE02 AT LEVEL 001 AS OF 07/10/80          00001
ISN 0002   SUBROUTINE COPE02 (ARRAY,RA,IA,HARRAY,NDRA,NDIA)          00002
ISN 0003   COMMON /CNMNI/ DELFUN,DABFUN,FDCH,FDCHM,CT,CTHIN,CTL,CTLMIN,ALPHAX00003
1,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,LINOBJ,ITMAX,IT00004
2RH,ICNDIR,IGOTO,NAC,INFO,INFO3,ITER          00005
ISN 0004   COMMON /COPE03/ SGNOPT,NCALC,IOBJ,NSV,NSC3J,NCONA,N2VX,M2VX,N2VY,M00006
12VY,N2VAR,IPSENS,IP2VAR,IPDBG,NACMX1,NDVTOT,LOCR(25),LOCI(25),ISCR00007
21,ISCR2,NXAPRX,NPS,NPFS,NPA,NF,INOM,IPAPRX,KMIN,KMAX,XFACT1,XFACT200008
3,NAN2,NAN3,NPMAX,NPTOT,JNOM,MAXTRM          00009
ISN 0005   DIMENSION ARRAY(NARRAY), RA(NDRA), IA(NDIA)          00010
C          *****C0011
C          ROUTINE TO CONTROL OPTIMIZATION.          00012
C          *****C0013
C          BY G. N. VANDERPLAATS          MAR., 1973.          00014
C          NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.          00015
C          -----00016
C          ARRAY DIMENSIONS          00017
C          -----00018
ISN 0006   NN1=NDV+2          00019
ISN 0007   NN2=2*NDV+NCON          00020
ISN 0008   NN3=NACMX1          00021
ISN 0009   NN4=NN3          00022
ISN 0010   IF (NDV.GT.NN4) NN4=NDV          00023
ISN 0012   NN5=2*NN4          00024
ISN 0013   NN6=NDVTOT          00025
ISN 0014   NN7=NCONA          00026
C          -----00027
C          ARRAY STARTING LOCATIONS          00028
C          -----00029
ISN 0015   X, VLB, VUB, DF, A, S, G1, G2, C, B, SCAL, ISC, IC, MS1          00030
ISN 0016   NX=1          00031
ISN 0017   NVLB=LOCR(2)          00032
ISN 0018   NVUB=LOCR(3)          00033
ISN 0019   NNSCAL=LOCR(4)          00034
ISN 0020   NDF=LOCR(23)          00035
ISN 0021   NS=NDF+NN1          00036
ISN 0022   NA=NG+NN2          00037
ISN 0023   NS=NA+NN1*NN3          00038
ISN 0024   NG1=NS+NN1          00039
ISN 0025   NG2=NG1+NN2          00040
ISN 0026   NC=NG2+NN2          00041
ISN 0027   NB=NC+NN4          00042
ISN 0028   NISC=LOCI(4)          00043
ISN 0029   NIC=LOCI(23)          00044
ISN 0030   N1S1=NIC+NN3          00045
C          AMULT, BLU, IDSGN, NSDGN, ICON          00046
ISN 0031   NAMULT=LOCR(5)          00047
ISN 0032   NBLU=LOCR(6)          00048
ISN 0033   NIOSGN=1          00049
ISN 0034   NNDSGN=LOCI(2)          00050
          NICON=LOCI(3)          00051
C          -----00052
C          OPTIMIZATION          00053
C          -----00054

```

```
ISN 0035      C      IGOTO=0                                00055
              C      CALL COMMIN (X,VLB,VUB,G,SCAL,DF,A,S,G1,G2,B,C,ISC,IC,MS1,N1,N2,N300056
              C      *,N4,N5)                                00057
ISN 0036      10     CONTINUE                                00058
ISN 0037      C      CALL COMMIN (RA(NX),RA(NVLB),RA(HVUB),RA(NG),RA(NNSCAL),RA(NDF),RA00059
              1(NA),RA(NS),RA(NG1),RA(NG2),RA(NB),RA(NC),IA(NISC),IA(NIC),IA(NMS100060
              2),NN1,NN2,NN3,NN4,NN5)                        00061
              C      ANALIZE.                                00062
ISN 0038      C      CALL COPE03 (ARRAY,NARRAY,RA(NX),RA(NG),RA(NAMULT),RA(NBLU),IA(NID00063
              1SGH),IA(NNDSSN),IA(NICON),NH1,NN2,NN6,NN7,ITER,OBJ) 00064
ISN 0039      IF (IGOTO.GT.0) GO TO 10                       00065
ISN 0041      RETURN                                         00066
ISN 0042      END                                           00067
```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTHT NOXREF NOALC NOANSF TERM IBM FLAG(I)

STATISTICS SOURCE STATEMENTS = 41, PROGRAM SIZE = 1352, SUBPROGRAM NAME =COPE02

STATISTICS NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

3012K BYTES OF CORE NOT USED

REQUESTED OPTIONS: SOURCE,NOMAP,NOXREF,NOLIST,NODECK,OPT(3),AUTODBL(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

270

```

C      DATA SET U477COPE03 AT LEVEL 002 AS OF 03/16/81
C      DATA SET U477COPE03 AT LEVEL 001 AS OF 02/13/81          00001
C      DATA SET 9188COPE03 AT LEVEL 001 AS OF 07/10/80          00002
ISN 0002  SUBROUTINE COPE03 (ARRAY,NARRAY,X,G,AMULT,BLU,IDSGN,NDSGN,ICON,NN100003
          1,NN2,NN3,NN7,ITER,OBJ)  :          00004
ISN 0003  COMMON /COPE3/ SGNOPT,NCALC,IOBJ,NSV,NSOBJ,NCONA,N2VX,M2VX,N2VY,M00005
          12VY,N2VAR,IPSENS,IP2VAR,IPDBG,NACMX1,NDVTOT,LOCR(25),LOCI(25),ISCR00006
          21,ISCR2,NXAPRX,NPS,NPFS,HPA,NF,INOM,IPAPRX,KMIN,KMAX,XFACT1,XFACT200007
          3,NAH2,NAH3,NPMAH,NPTOT,INOM,MAXTRM          00008
ISN 0004  COMMON /UIOS/ I5I, I6I          00009
ISN 0005  DIMENSION ARRAY(NARRAY), X(NN1), G(NN2), AMULT(NN6), BLU(4,NN7) 00010
ISN 0006  DIMENSION IDSGN(NN6), NDSGN(NN6), ICON(NN7)          00011
          *****00012
C      BUFFER BETWEEN COMMIN AND COPE3 FUNCTION EVALUATION.    00013
C      *****00014
C      BY G. N. VANDERPLAATS          MAR., 1973.          00015
C      NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.        00016
C      INITIAL ANALYSIS HAS BEEN DONE. IF ITER = 0, GO EVALUATE 00017
C      OBJECTIVE AND CONSTRAINTS.          00018
ISN 0007  IF (ITER.EQ.0) ITER1=0          00019
ISN 0009  IF (ITER.LT.1) GO TO 40          00020
          -----00021
C      PRINT OUTPUT IF DEBUG CONTROL IS TURNED ON          00022
C      -----00023
C      DEBUG OUTPUT AS REQUIRED.          00024
ISN 0011  IF (IPDBG.LT.1) GO TO 20          00025
ISN 0013  IF (ITER.EQ.ITER1.OR.ITER.LE.1) GO TO 20          00026
ISN 0015  XSAV2=X(1)          00027
ISN 0016  X(1)=XSAV1          00028
ISN 0017  N5=LOCR(5)          00029
ISN 0018  M2=LOCI(2)          00030
ISN 0019  DO 10 I=1,NDVTOT          00031
ISN 0020  N=NDSGN(I)          00032
ISN 0021  M=IDSGN(I)          00033
ISN 0022  IF (N.GT.0) ARRAY(M)=X(N)*AMULT(I)          00034
ISN 0024  10 CONTINUE          00035
ISN 0025  ICALC=3          00036
ISN 0026  NAH3=NAH3+1          00037
ISN 0027  CALL ANALIZ (ICALC)          00038
ISN 0028  WRITE (I6I,70)          00039
ISN 0029  ITER1=ITER          00040
ISN 0030  X(1)=XSAV2          00041
ISN 0031  20 CONTINUE          00042
          -----00043
C      TRANSFER DESIGN VARIABLES TO USER ARRAY          00044
C      -----00045
ISN 0032  N5=LOCR(5)          00046
ISN 0033  M2=LOCI(2)          00047
ISN 0034  DO 30 I=1,NDVTOT          00048
ISN 0035  N=NDSGN(I)          00049
ISN 0036  M=IDSGN(I)          00050
ISN 0037  IF (N.GT.0) ARRAY(M)=X(N)*AMULT(I)          00051
ISN 0039  30 CONTINUE          00052
          -----00053
C      ANALIZE          00054

```

```

C -----00055
ISN 0040          ICALC=2          00056
ISN 0041          NAN2=NAN2+1      00057
ISN 0042          CALL ANALIZ (ICALC) 00058
C                SAVE X(1).        00059
ISN 0043          XSAV1=X(1)        00060
C -----00061
C                OBJECTIVE          00062
C -----00063
ISN 0044          40  CONTINUE        00064
ISN 0045          OBJ=-SGNOPT*ARRAY(IOBJ) 00065
ISN 0046          IF (NCONA.EQ.0) RETURN 00066
C -----00067
C                CONSTRAINT VALUES  00068
C -----00069
ISN 0048          J=1                00070
ISN 0049          N=0                00071
ISN 0050          DO 60 I=1,NCONA     00072
C                PARAMETER IDENTIFIER. 00073
ISN 0051          NN=ICON(I)         00074
ISN 0052          CC=ARRAY(NN)       00075
C                LOWER BOUND.        00076
ISN 0053          BB=BLU(1,I)        00077
ISN 0054          IF (BB.LT.-1.0E+15) GO TO 50 00078
C                NORMALIZATION FACTOR. 00079
ISN 0056          C1=BLU(2,I)        00080
C                CONSTRAINT VALUE.    00081
ISN 0057          N=N+1              00082
ISN 0058          G(N)=(BB-CC)/C1     00083
C                UPPER BOUND.        00084
ISN 0059          50  BB=BLU(3,I)    00085
C                NORMALIZATION FACTOR. 00086
ISN 0060          C1=BLU(4,I)        00087
ISN 0051          J=J+4              00088
ISN 0062          IF (BB.GT.1.0E+15) GO TO 60 00089
C                CONSTRAINT VALUE.    00090
ISN 0064          N=N+1              00091
ISN 0065          G(N)=(CC-EB)/C1     00092
ISN 0066          60  CONTINUE        00093
ISN 0067          RETURN              00094
C -----00095
C                FORMATS              00096
C -----00097
ISN 0068          70  FORMAT (1H1)    00099
ISN 0069          END                  00100

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NONAP NOFORMAT GOSTHT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 68, PROGRAM SIZE = 1246, SUBPROGRAM NAME =COPE03
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

3004K BYTES OF CORE NOT USED

272

```

C          DATA SET U477COPE04 AT LEVEL 001 AS OF 02/13/81
C          DATA SET 9158COPE04 AT LEVEL 001 AS OF 07/10/80          00001
ISN 0002   SUBROUTINE COPE04 (ARRAY,NARRAY,SENS,NSENSZ,ISENS,NSENS,TEMP,NN8,N00002
          1N9,NN10,RA,IA,NDRA,NDIA)                                00003
ISN 0003   COMMON /COPE04/ TITLE(20)                               00004
ISN 0004   COMMON /COPE03/ SGOPT,NCALC,IOBJ,NSV,NSOBJ,NCONA,N2VX,M2VX,N2VY,M00005
          12VY,N2VAR,IPSENS,IP2VAR,IPDBG,HACMX1,NDVTOT,LOCR(25),LOCI(25),ISCR00006
          21,ISCR2,NXAPRX,NPS,NPFS,NPA,NF,INOM,IPAPRX,KMIN,KMAX,XFACT1,XFACT200007
          3,NAN2,NAN3,NPMAX,NPTOT,JNOM,MAXTRM                        00008
ISN 0005   COMMON /CNM1/ DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,ALPHAX00009
          1,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFOG,NSCAL,LINOBJ,ITMAX,IT00010
          2RM,ICNOIR,IGOTO,NAC,INFO,INFOG,ITER                      00011
ISN 0006   DIMENSION ARRAY(NARRAY), SENS(NN8), NSENSZ(NN9), ISENS(NN10), NSEN00012
          IS(NN10), TEMP(1), RA(NDRA), IA(NDIA)                    00013
C          *****00014
C          ROUTINE TO PROVIDE SENSITIVITY INFORMATION WITH RESPECT TO 00015
C          A PRESCRIBED SET OF DESIGN VARIABLES.                    00016
C          *****00017
C          BY G. N. VANDERPLAATS MAR., 1973.                        00018
C          STORE OUTPUT ON UNIT ISCR1.                               00019
ISN 0007   REWIND ISCR1                                           00020
ISN 0008   IF (IPDBG.LT.1) IPRINT=0                                00021
C          -----00022
C          WRITE BASIC INFORMATION ON UNIT ISCR1                    00023
C          -----00024
C          TITLE.                                                  00025
ISN 0010   WRITE (ISCR1,330) (TITLE(I),I=1,20)                    00026
C          NCALC, NSV, NSOBJ                                        00027
ISN 0011   WRITE (ISCR1,340) NCALC,NSV,NSOBJ                        00028
C          ISENS(I),I=1,NSV.                                       00029
ISN 0012   WRITE (ISCR1,340) (ISENS(I),I=1,NSV)                    00030
C          NSENSZ(I),I=1,NSOBJ.                                       00031
ISN 0013   WRITE (ISCR1,340) (NSENSZ(I),I=1,NSOBJ)                 00032
ISN 0014   JCALC=3                                                 00033
ISN 0015   ICALC=2                                                 00034
ISN 0016   NDVSAV=NDV                                              00035
C          -----00036
C          ***** NOMINAL *****00037
C          -----00038
ISN 0017   IF (NCALC.EQ.5) GO TO 10                                00039
C          STANDARD SENSITIVITY.                                     00040
ISN 0019   NAN2=NAN2+1                                             00041
ISN 0020   CALL ANALIZ (ICALC)                                       00042
ISN 0021   IF (IPSENS.GT.0) NAN3=NAN3+1                             00043
ISN 0023   IF (IPSENS.GT.0) CALL ANALIZ (JCALC)                     00044
ISN 0025   GO TO 130                                               00045
ISN 0026   10 CONTINUE                                             00046
C          OPTIMUM SENSITIVITY.                                     00047
C          SAVE X, VLB, VUB AND SCAL IN TEMPORARY STORAGE.         00048
ISN 0027   N=4*NDV+8                                               00049
ISN 0028   L=LOCR(24)                                              00050
ISN 0029   DO 20 I=1,N                                             00051
ISN 0030   RA(L)=RA(I)                                             00052
ISN 0031   20 L=L+1                                                00053
C          SAVE IDSGN AND NDSGN IN TEMPORARY STORAGE.              00054

```

```

+VERSION 1.3.0 (01 MAY 80)   COPE04   SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)   DATE 82.141/10.54.31   PAGE 2
ISN 0032      N=2*NDVTOT      00055
ISN 0033      L=LOCI(24)      00056
ISN 0034      DO 30 I=1,N      00057
ISN 0035      IA(L)=IA(I)      00058
ISN 0036      L=L+1            00059
30
C SHIFT DESIGN VARIABLE INFORMATION IF ANY SENSITIVITY VARIABLE IS 00060
C ALSO A DESIGN VARIABLE.      00061
ISN 0037      NDV2=NDV+2      00062
ISN 0038      DO 90 I=1,NSV    00063
C GLOBAL SENSITIVITY VARIABLE LOCATION. 00064
ISN 0039      N=ISENS(I)      00065
C IS THIS ALSO A DESIGN VARIABLE. 00066
ISN 0040      M2=LOCI(2)      00067
ISN 0041      DO 40 J=1,NDVTOT 00068
ISN 0042      L=IA(J)         00069
ISN 0043      IF (L.EQ.N) GO TO 50 00070
ISN 0045      40 M2=M2+1      00071
C SENSITIVITY VARIABLE IS NOT A DESIGN VARIABLE. 00072
ISN 0046      GO TO 90        00073
ISN 0047      50 CONTINUE     00074
C SENSITIVITY VARIABLE IS ALSO A DESIGN VARIABLE. 00075
ISN 0048      NDV=NDV-1      00076
ISN 0049      IDV=IA(M2)      00077
C ELIMINATE THIS DESIGN VARIABLE AND REDUCE HIGHER NUMBER DESIGN 00078
C VARIABLES BY ONE.          00079
ISN 0050      M2=LOCI(2)      00080
ISN 0051      DO 70 J=1,NDVTOT 00081
ISN 0052      IF (IA(M2).NE.IDV) GO TO 60 00082
ISN 0054      IA(M2)=0        00083
C SET DESIGN VARIABLE VALUES TO SENSITIVITY VARIABLE VALUE. 00084
ISN 0055      K=IA(J)         00085
ISN 0056      M5=LOCR(5)+J-1  00086
ISN 0057      ARRAY(K)=ARRAY(N)*RA(M5) 00087
ISN 0058      60 CONTINUE     00088
ISN 0059      IF (IA(M2).GT.IDV) IA(M2)=IA(M2)-1 00089
ISN 0061      70 M2=M2+1      00090
ISN 0062      IF (IDV.EQ.NDV) GO TO 90 00091
C SHIFT X, VLB, VUB AND SCAL. 00092
ISN 0064      DO 80 J=IDV,NDV 00093
C X.                          00094
ISN 0065      RA(J)=RA(J+1)    00095
C VLB.                        00096
ISN 0066      K=J+NDV2        00097
ISN 0067      RA(K)=RA(K+1)    00098
C VUB.                        00099
ISN 0068      K=K+NDV2        00100
ISN 0069      RA(K)=RA(K+1)    00101
C SCAL.                       00102
ISN 0070      K=K+NDV2        00103
ISN 0071      80 RA(K)=RA(K+1) 00104
ISN 0072      90 CONTINUE     00105
ISN 0073      NAN2=NAN2+1     00106
ISN 0074      CALL ANALIZ (ICALC) 00107
ISN 0075      IF (NDV.LE.0) GO TO 100 00108
ISN 0077      CALL COPE02 (ARRAY,RA,IA,NARRAY,NDRA,NDIA) 00109
ISN 0078      100 CONTINUE    00110
ISN 0079      IF (IPSENS.GT.0) NAN3=NAN3+1 00111
ISN 0081      IF (IPSENS.GT.0) CALL ANALIZ (JCALC) 00112
C PUT X, VLB, VUB AND SCAL BACK. 00113

```

274

```

ISN 0083      L=LOCR(24)                                00114
ISN 0084      DO 110 I=1,N                               00115
ISN 0085      RA(I)=RA(L)                                00116
ISN 0086      110 L=L+1                                  00117
                C PUT IDSGN AND NDSGN BACK.              00118
ISN 0087      N=2*NDVTOT                                 00119
ISN 0088      L=LOCI(24)                                00120
ISN 0089      DO 120 I=1,N                               00121
ISN 0090      IA(I)=IA(L)                                00122
ISN 0091      120 L=L+1                                  00123
ISN 0092      130 CONTINUE                               00124
                C -----00125
                C WRITE NOMINAL RESULTS ON UNIT ISCR1    00126
                C -----00127
                C SEHS(I,1).                              00128
ISN 0093      M=1                                        00129
ISN 0094      DO 140 I=1,NSV                             00130
ISN 0095      TEMP(I)=SENS(M)                            00131
ISN 0096      140 M=M+NSENS(I)                           00132
ISN 0097      WRITE (ISCR1,350) (TEMP(I),I=1,NSV)       00133
                C SENSITIVITY OBJECTIVES, OBJZ.         00134
ISN 0098      DO 150 I=1,NSOBJ                           00135
ISN 0099      M=NSENSZ(I)                                00136
ISN 0100      150 TEMP(I)=ARRAY(M)                       00137
ISN 0101      WRITE (ISCR1,350) (TEMP(I),I=1,NSOBJ)     00138
                C -----00139
                C ***** SENSITIVITIES *****00140
                C -----00141
ISN 0102      NSVALI=0                                   00142
ISN 0103      DO 320 II=1,NSV                             00143
                C GLOBAL LOCATION OF SENSITIVITY VARIABLE. 00144
ISN 0104      ISENSV=ISENS(II)                           00145
                C NUMBER OF SENSITIVITY VARIABLES, NSENSV. 00146
ISN 0105      NSENSV=NSENS(II)                           00147
                C WRITE ISENSV AND NSENSV-1 ON UNIT ISCR1. 00148
ISN 0106      NSENSI=NSENSV-1                            00149
ISN 0107      WRITE (ISCR1,340) ISENSV,NSENSI            00150
ISN 0108      IF (NSENSV.LE.1) GO TO 320                 00151
ISN 0110      ID1=0                                       00152
ISN 0111      IF (NCALC.NE.5) GO TO 210                  00153
                C IS THIS SENSITIVITY VARIABLE ALSO A DESIGN VARIABLE. 00154
ISN 0113      NDV=NDVSAV                                  00155
ISN 0114      DO 160 I=1,NDVTOT                           00156
ISN 0115      JJ=I                                        00157
ISN 0116      IF (IA(I).EQ.ISENSV) GO TO 170            00158
ISN 0118      160 CONTINUE                               00159
                C ISENSV IS NOT A DESIGN VARIABLE.         00160
ISN 0119      GO TO 210                                   00161
ISN 0120      170 CONTINUE                               00162
                C ISENSV IS A DESIGN VARIABLE. MODIFY OPTIMIZATION INFORMATION. 00163
ISN 0121      NDV2=NDV+2                                  00164
ISN 0122      NDV=NDV-1                                   00165
                C SAVE X, VLB, VUB AND SCAL FOR THIS DESIGN VARIABLE AND SHIFT 00166
                C REMAINING VARIABLES.                    00167
                C SAVE.                                    00168
ISN 0123      M2=LOCI(2)+JJ-1                            00169
ISN 0124      ID1=IA(M2)                                  00170
ISN 0125      SAVX=RA(ID1)                                00171
ISN 0126      K=ID1+NDV2                                  00172

```

+VERSION 1.3.0 (01 MAY 80) COPE04		SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)	DATE 82.141/10.54.31	PAGE 4
ISN 0127		SAVL=RA(K)	00173	
ISN 0128		K=K+NDV2	00174	
ISN 0129		SAVU=RA(K)	00175	
ISN 0130		K=K+NDV2	00176	
ISN 0131		SAVS=RA(K)	00177	
	C	SHIFT	00178	
ISN 0132		IF (ID1.GT.NDV) GO TO 190	00179	
ISN 0134		DO 180 I=ID1,NDV	00180	
ISN 0135		RA(I)=RA(I+1)	00181	
ISN 0136		K=I+NDV2	00182	
ISN 0137		RA(K)=RA(K+1)	00183	
ISN 0138		K=K+NDV2	00184	
ISN 0139		RA(K)=RA(K+1)	00185	
ISN 0140		K=K+NDV2	00186	
ISN 0141	180	RA(K)=RA(K+1)	00187	
ISN 0142	190	CONTINUE	00188	
	C	MODIFY NDSGN.	00189	
ISN 0143		M2=LOCI(2)	00190	
ISN 0144		DO 200 I=1,NDVTOT	00191	
ISN 0145		IF (IA(M2).EQ.ID1) IA(M2)=0	00192	
ISN 0147		IF (IA(M2).GT.ID1) IA(M2)=IA(M2)-1	00193	
ISN 0149	200	M2=M2+1	00194	
ISN 0150	210	CONTINUE	00195	
	C	-----	00196	
	C	VARY THE VALUE OF THE SENSITIVITY PARAMETER	00197	
	C	-----	00198	
ISN 0151		NSVALI=NSVALI+1	00199	
ISN 0152		NSVALN=NSVALI	00200	
ISN 0153		DO 280 JJ=2,NSENSV	00201	
ISN 0154		NSVALI=NSVALI+1	00202	
ISN 0155		ARRAY(ISENSV)=SENS(NSVALI)	00203	
	C	WRITE SENS(I,J) ON UNIT ISCR1.	00204	
ISN 0156		WRITE (ISCR1,350) SENS(NSVALI)	00205	
	C	ANALIZE.	00206	
ISN 0157		IF (NCALC.EQ.5) GO TO 220	00207	
	C	STANDARD SENSITIVITY.	00208	
ISN 0159		NAN2=NAN2+1	00209	
ISN 0160		CALL ANALIZ (ICALC)	00210	
ISN 0161		IF (IPSENS.GT.0) NAN3=NAN3+1	00211	
ISN 0163		IF (IPSENS.GT.0) CALL ANALIZ (JCALC)	00212	
ISN 0165		GO TO 260	00213	
ISN 0166	220	CONTINUE	00214	
	C	OPTIMUM SENSITIVITY.	00215	
ISN 0167		IF (NDV.EQ.NDVSAV) GO TO 240	00216	
	C	SET LINKED DESIGN VARIABLE VALUES TO PRESCRIBED VALUE.	00217	
ISN 0169		M2=LOCI(2)	00218	
ISN 0170		DO 230 I=1,NDVTOT	00219	
ISN 0171		IF (IA(M2).NE.0) GO TO 230	00220	
ISN 0173		L=IA(I)	00221	
ISN 0174		M5=LOCR(5)+I-1	00222	
ISN 0175		ARRAY(L)=ARRAY(ISENSV)*RA(M5)	00223	
ISN 0176	230	M2=M2+1	00224	
ISN 0177	240	CONTINUE	00225	
ISN 0178		NAN2=NAN2+1	00226	
ISN 0179		CALL ANALIZ (ICALC)	00227	
ISN 0180		IF (NDV.LE.0) GO TO 250	00228	
ISN 0182		CALL COPE02 (ARRAY,RA,IA,NARRAY,NDRA,NDIA)	00229	
ISN 0183	250	CONTINUE	00230	
ISN 0184		IF (IPSENS.GT.0) NAN3=NAN3+1	00231	

276

```

ISN 0186      IF (IPSENS.GT.0) CALL ANALIZ (JCALC)          00232
ISN 0188      260 CONTINUE                                00233
C             -----00234
C             WRITE SENSITIVITY RESULTS ON UNIT ISCR1    00235
C             -----00236
C             OBJZ.                                       00237
ISN 0189      DO 270 I=1,NSOBJ                             00238
ISN 0190      M=NSSENSZ(I)                                00239
ISN 0191      270 TEMP(I)=ARRAY(M)                        00240
ISN 0192      WRITE (ISCR1,350) (TEMP(I),I=1,NSOBJ)     00241
ISN 0193      280 CONTINUE                                00242
ISN 0194      ARRAY(ISENSV)=SENS(NSVALN)                 00243
ISN 0195      IF (NCALC.NE.5.OR.ID1.EQ.0) GO TO 320     00244
C             RESTORE X, VLB, VUB AND SCAL.              00245
ISN 0197      NDV=NDVSAV                                  00246
ISN 0198      IF (ID1.EQ.NDV) GO TO 300                 00247
ISN 0200      L=NDV-1                                    00248
ISN 0201      L1=L                                       00249
ISN 0202      DO 290 I=ID1,L1                            00250
ISN 0203      RA(L+1)=RA(L)                              00251
ISN 0204      K=L+NDV2                                   00252
ISN 0205      RA(K+1)=RA(K)                             00253
ISN 0206      K=K+NDV2                                   00254
ISN 0207      RA(K+1)=RA(K)                             00255
ISN 0208      K=K+NDV2                                   00256
ISN 0209      RA(K+1)=RA(K)                             00257
ISN 0210      290 L=L-1                                  00258
ISN 0211      RA(ID1)=SAVX                               00259
ISN 0212      K=ID1+NDV2                                 00260
ISN 0213      RA(K)=SAVL                                00261
ISN 0214      K=K+NDV2                                   00262
ISN 0215      RA(K)=SAVU                                00263
ISN 0216      K=K+NDV2                                   00264
ISN 0217      RA(K)=SAVS                                 00265
ISN 0218      300 CONTINUE                                00266
C             RESTORE NDSGN.                             00267
ISN 0219      M2=LOCI(2)                                 00268
ISN 0220      DO 310 I=1,NDVTOT                          00269
ISN 0221      IF (IA(M2).GE.ID1) IA(M2)=IA(M2)+1      00270
ISN 0223      IF (IA(M2).EQ.0) IA(M2)=ID1              00271
ISN 0225      310 M2=M2+1                                00272
ISN 0226      320 CONTINUE                                00273
ISN 0227      RETURN                                     00274
C             -----00275
C             FORMATS                                     00276
C             -----00277
ISN 0228      330 FORMAT (20A4)                          00278
ISN 0229      340 FORMAT (16I5)                          00279
ISN 0230      350 FORMAT (5E15.8)                       00280
ISN 0231      END                                        00281

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NCDECK OBJECT NCHAP NCFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 230, PROGRAM SIZE = 3454, SUBPROGRAM NAME =COPE04
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

2972K BYTES OF CORE NOT USED

```

C          DATA SET U477COPE05 AT LEVEL 002 AS OF 03/16/81
C          DATA SET U477COPE05 AT LEVEL 001 AS OF 02/13/81          00001
C          DATA SET 918SCOPE05 AT LEVEL 001 AS OF 07/10/80          00002
ISN 0002   SUBROUTINE COPE05 (RA,IA,NDRA,NDIA,ISCR1)                   00003
ISN 0003   COMMON /UIOS/ I5I, I6I                                     00004
ISN 0004   DIMENSION RA(NDRA), IA(NDIA)                               00005
C          *****00006
C          ROUTINE TO PRINT SENSITIVITY INFORMATION STORED ON UNIT ISCR1. 00007
C          *****00008
C          BY G. N. VANDERPLAATS          JULY, 1974.                 00009
C          NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.          00010
ISN 0005   REWIND ISCR1                                              00011
C          -----00012
C          GENERAL INFORMATION                                         00013
C          -----00014
C          TITLE.                                                      00015
ISN 0006   READ (ISCR1,70) (RA(I),I=1,20)                             00016
C          NCALC, NSV, NSOBJ                                           00017
ISN 0007   READ (ISCR1,80) NCALC,NSV,NSOBJ                           00018
ISN 0008   IF (NCALC.NE.3.AND.NCALC.NE.5) RETURN                    00019
ISN 0010   IF (NCALC.EQ.3) WRITE (I6I,90)                            00020
ISN 0012   IF (NCALC.EQ.5) WRITE (I6I,50)                            00021
ISN 0014   WRITE (I6I,60) (RA(I),I=1,20)                             00022
ISN 0015   WRITE (I6I,100) NSV,NSOBJ                                  00023
C          ISENS(I),I=1,NSV.                                           00024
ISN 0016   READ (ISCR1,80) (IA(I),I=1,NSV)                           00025
ISN 0017   WRITE (I6I,110)                                            00026
ISN 0018   WRITE (I6I,120) (IA(I),I=1,NSV)                           00027
C          NSENSZ(I),I=1,NSOBJ.                                         00028
ISN 0019   READ (ISCR1,80) (IA(I),I=1,NSOBJ)                        00029
ISN 0020   WRITE (I6I,130)                                            00030
ISN 0021   WRITE (I6I,120) (IA(I),I=1,NSOBJ)                        00031
C          -----00032
C          NOMINAL INFORMATION                                         00033
C          -----00034
C          SENS(I),I=1,NSV.                                             00035
ISN 0022   READ (ISCR1,140) (RA(I),I=1,NSV)                         00036
ISN 0023   WRITE (I6I,150)                                            00037
ISN 0024   WRITE (I6I,160) (RA(I),I=1,NSV)                          00038
C          OBJZ(I),I=1,NSOBJ.                                           00039
ISN 0025   READ (ISCR1,140) (RA(I),I=1,NSOBJ)                       00040
ISN 0026   WRITE (I6I,170)                                            00041
ISN 0027   WRITE (I6I,160) (RA(I),I=1,NSOBJ)                        00042
C          -----00043
C          ***** SENSITIVITY INFORMATION *****00044
C          -----00045
ISN 0028   WRITE (I6I,180)                                            00046
ISN 0029   DO 40 ISENS=1,NSV                                          00047
C          ISENSI, NSENSI                                              00048
ISN 0030   READ (ISCR1,80) ISENSI,NSENSI                             00049
ISN 0031   WRITE (I6I,190) ISENSI                                    00050
ISN 0032   IF (NSENSI.EQ.0) WRITE (I6I,200)                          00051
ISN 0034   IF (NSENSI.EQ.0) GO TO 40                                  00052
ISN 0036   DO 30 JJ=1,NSENSI                                          00053
C          SENS(I,J).                                                  00054

```

278

```

ISN 0037      C      READ (ISCR1,140) SENSIJ          00055
                C      03JZ(I),I=1,NSOBJ.          00056
ISN 0038      C      READ (ISCR1,140) (RA(I),I=1,NSOBJ) 00057
ISN 0039      C      N=MINO(4,NSOBJ)              00058
ISN 0040      C      WRITE (I6I,210) SENSIJ,(RA(I),I=1,N) 00059
ISN 0041      C      N=(NSOBJ-1)/4                00060
ISN 0042      C      IF (N.LT.1) GO TO 20          00061
ISN 0044      C      L1=5                          00062
ISN 0045      C      DO 10 I=1,N                   00063
ISN 0046      C      L2=L1+3                       00064
ISN 0047      C      L2=MINO(L2,NSOBJ)            00065
ISN 0048      C      WRITE (I6I,220) (RA(J),J=L1,L2) 00066
ISN 0049      C      L1=L1+4                       00067
ISN 0050      C      20 CONTINUE                   00068
ISN 0051      C      30 CONTINUE                   00069
ISN 0052      C      40 CONTINUE                   00070
ISN 0053      C      RETURN                          00071
                C      -----00072
                C      FORMATS                      00073
                C      -----00074
ISN 0054      C      50 FORMAT (1H1,4X,46HOPTIMUM SENSITIVITY ANALYSIS RESULTS (NCALC=5)) 00075
ISN 0055      C      60 FORMAT (//5X,5HTITLE/5X,20A4) 00076
ISN 0056      C      70 FORMAT (20A4)              00077
ISN 0057      C      80 FORMAT (16I5)              00078
ISN 0058      C      90 FORMAT (1H1,4X,47HSTANDARD SENSITIVITY ANALYSIS RESULTS (NCALC=3)) 00079
ISN 0059      C      100 FORMAT (//5X,36HNUMBER OF SENSITIVITY VARIABLES, NSV,9X,1H=,I5/5X,00080
                C      139HNUMBER OF SENSITIVITY OBJECTIVES, NSOBJ,6X,1H=,I5) 00081
ISN 0060      C      110 FORMAT (//5X,52HGLOBAL NUMBERS ASSOCIATED WITH SENSITIVITY VARIABLE 00082
                C      1ES) 00083
ISN 0061      C      120 FORMAT (5X,10I5)          00084
ISN 0062      C      130 FORMAT (//5X,53HGLOBAL NUMBERS ASSOCIATED WITH SENSITIVITY OBJECTIVE 00085
                C      1VES) 00086
ISN 0063      C      140 FORMAT (5E15.8)          00087
ISN 0064      C      150 FORMAT (///5X,26HNOMINAL DESIGN INFORMATION//5X,31HVALUES OF SENS 00088
                C      ITIVITY VARIABLES) 00089
ISN 0065      C      160 FORMAT (5X,5E13.5)       00090
ISN 0066      C      170 FORMAT (//5X,41HVALUES OF SENSITIVITY OBJECTIVE FUNCTIONS) 00091
ISN 0067      C      180 FORMAT (////5X,28HSENSITIVITY ANALYSIS RESULTS) 00092
ISN 0068      C      190 FORMAT (//5X,15HGLOBAL VARIABLE,I5//10X,1HX,20X,4HF(X)) 00093
ISN 0069      C      200 FORMAT (/5X,35HTHE NOMINAL VALUE IS THE ONLY VALUE/5X,27HSPECIFIED 00094
                C      I FOR THIS VARIABLE) 00095
ISN 0070      C      210 FORMAT (/3X,E12.4,3X,4E13.4) 00096
ISN 0071      C      220 FORMAT (18X,4E13.4)      00097
ISN 0072      C      END                            00098

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOBCL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT COSTHT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 71, PROGRAM SIZE = 2210, SUBPROGRAM NAME =COPE05
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****
2996K BYTES OF CORE NOT USED

```

```

C          DATA SET U477COPE06 AT LEVEL 001 AS OF 02/13/81
C          DATA SET 9188COPE06 AT LEVEL 001 AS OF 07/10/80          00001
ISN 0002   SUBROUTINE COPE06 (ARRAY,RA,IA,NARRAY,NDRA,NDIA)          00002
ISN 0003   COMMON /COPE51/ TITLE(20)                                00003
ISN 0004   COMMON /COPE53/ SGNOPT,NCALC,IOBJ,NSV,NSOBJ,NCONA,N2VX,M2VX,N2VY,M00004
          12VY,N2VAR,IPSENS,IP2VAR,IPDBG,NACHX1,NDVTOT,LOCR(25),LOCI(25),ISCR00005
          21,ISCR2,NXAPRX,NPS,NPFS,NPA,NF,INOM,IPAPRX,KMIN,KMAX,XFACT1,XFACT200006
          3,NAN2,NAN3,NPMAX,NPTOT,JROM,MAXTRM                        00007
ISN 0005   DIMENSION ARRAY(NARRAY), RA(NDRA), IA(NDIA)              00008
          *****00009
C          ROUTINE TO CALCULATE FUNCTIONS OF TWO DESIGN VARIABLES FOR ALL 00010
C          COMBINATIONS OF A SET OF PRESCRIBED VALUES OF THESE VARIABLES. 00011
C          *****00012
C          WRITE OUTPUT INFORMATION ON SCRATCH UNIT ISCR1.          00013
C          BY G. N. VANDERPLAATS                                    AUG., 1974.                    00014
ISN 0006   NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.      00015
          RENIND ISCR1                                                00016
C          -----00017
C          UNIT ISCR1 WRITE                                            00018
C          -----00019
ISN 0007   WRITE (ISCR1,40) (TITLE(I),I=1,20)                        00020
ISN 0008   WRITE (ISCR1,50) NCALC,N2VAR,M2VX,M2VX,M2VY,N2VY        00021
          N2VZ.                                                        00022
ISN 0009   M20=LOCI(20)                                                00023
ISN 0010   M21=LOCI(21)-1                                              00024
ISN 0011   WRITE (ISCR1,50) (IA(I),I=M20,M21)                        00025
          -----00026
C          TWO-VARIABLE FUNCTION SPACE                                00027
C          -----00028
ISN 0012   ICALC=2                                                      00029
ISN 0013   KCALC=3                                                      00030
ISN 0014   ISIGN=1                                                      00031
ISN 0015   N20=LOCR(20)                                                00032
ISN 0016   N21=LOCR(21)-1                                              00033
ISN 0017   DO 30 I=1,M2VX                                              00034
ISN 0018   ARRAY(N2VX)=RA(N20)                                        00035
ISN 0019   DO 20 J=1,M2VY                                              00036
ISN 0020   N21=N21+ISIGN                                               00037
ISN 0021   ARRAY(N2VY)=RA(N21)                                        00038
          ANALIZE.                                                      00039
ISN 0022   NAN2=NAN2+1                                                 00040
ISN 0023   CALL ANALIZ (ICALC)                                        00041
ISN 0024   IF (IP2VAR.GT.0) CALL ANALIZ (KCALC)                      00042
ISN 0026   IF (IP2VAR.GT.0) NAN3=NAN3+1                               00043
          -----00044
C          UNIT ISCR1 WRITE                                            00045
C          -----00046
C          WRITE X, Y.                                                   00047
ISN 0028   WRITE (ISCR1,60) RA(N20),RA(N21)                          00048
          F(X,Y) VALUES.                                              00049
C          N23=LOCR(23)                                                 00050
ISN 0029   N24=N23                                                      00051
ISN 0030   M20=LOCI(20)                                                00052
ISN 0031   DO 10 K=1,N2VAR                                             00053
ISN 0033   N=IA(M20)                                                    00054
  
```

280

```

+VERSION 1.3.0 (01 MAY 80)   COPE06   SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)   DATE 82.141/10.54.45   PAGE 2
  ISN 0034           RA(N24)=ARRAY(N)           00055
  ISN 0035           N24=N24+1                 00056
  ISN 0036           M20=M20+1                 00057
  ISN 0037    10    CONTINUE                    00058
  ISN 0038           N24=N23+N2VAR-1           00059
  ISN 0039           WRITE (ISCR1,60) (RA(K),K=N23,N24) 00060
  ISN 0040    20    CONTINUE                    00061
  ISN 0041           N21=N21+ISIGN             00062
  ISN 0042           N20=N20+1                 00063
  ISN 0043           ISIGN=-ISIGN              00064
  ISN 0044    30    CONTINUE                    00065
  ISN 0045           RETURN                    00066
  C -----00067
  C                               FORMATS        00068
  C -----00069
  C                               00070
  C                               00071
  C                               00072
  ISN 0046    40    FORMAT (20A4)              00073
  ISN 0047    50    FCRMAT (16I5)              00074
  ISN 0048    60    FORMAT (5E15.8)           00075
  ISN 0049           END                       00076

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODACK OBJECT NOMAP NOFORMAT GCSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS*   SOURCE STATEMENTS =    48, PROGRAM SIZE =    1118, SUBPROGRAM NAME =COPE06
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

3008K BYTES OF CORE NOT USED

```

C      DATA SET U477COPE07 AT LEVEL 002 AS OF 03/16/81
C      DATA SET U477COPE07 AT LEVEL 001 AS OF 02/13/81          00001
C      DATA SET 9188COPE07 AT LEVEL 001 AS OF 07/10/80          00002
ISN 0002  SUBROUTINE COPE07 (RA,IA,NDRA,NDIA,ISCR1)                00003
ISN 0003  COMMON /UIOS/ I5I, I6I                                  00004
ISN 0004  DIMENSION RA(NDRA), IA(NDIA)                            00005
C      *****00006
C      ROUTINE TO PRINT TWO VARIABLE FUNCTION SPACE INFORMATION STORED ON00007
C      UNIT ISCR1.                                                00008
C      *****00009
C      BY G. N. VANDERPLAATS          AUG., 1974.                00010
C      NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.          00011
ISN 0005  REWIND ISCR1                                           00012
C      -----00013
C      GENERAL INFORMATION                                         00014
C      -----00015
C      TITLE.                                                       00016
ISN 0006  READ (ISCR1,60) (RA(I),I=1,20)                          00017
ISN 0007  READ (ISCR1,70) NCALC,N2VAR,M2VX,N2VX,M2VY,N2VY        00018
ISN 0008  IF (NCALC.NE.4.AND.NCALC.NE.6) RETURN                  00019
ISN 0010  WRITE (I6I,50)                                          00020
ISN 0011  WRITE (I6I,40) (RA(I),I=1,20)                          00021
C      N2VZ(I),I=1,N2VAR.                                         00022
ISN 0012  READ (ISCR1,70) (IA(I),I=1,N2VAR)                      00023
C      N2VX, N2VY.                                                00024
ISN 0013  WRITE (I6I,120) N2VX,N2VY                               00025
C      N2VZ.                                                       00026
ISN 0014  WRITE (I6I,130)                                         00027
ISN 0015  WRITE (I6I,80) (IA(I),I=1,N2VAR)                       00028
C      -----00029
C      TWO-VARIABLE FUNCTION SPACE INFORMATION                      00030
C      -----00031
ISN 0016  DO 30 I=1,M2VX                                          00032
ISN 0017  WRITE (I6I,140)                                         00033
ISN 0018  DO 30 J=1,M2VY                                          00034
C      X, Y.                                                        00035
ISN 0019  READ (ISCR1,150) XX,YY                                 00036
C      F(X,Y).                                                      00037
ISN 0020  READ (ISCR1,150) (RA(K),K=1,N2VAR)                    00038
ISN 0021  N=4                                                     00039
ISN 0022  IF (N2VAR.LT.4) N=N2VAR                                00040
ISN 0024  IF (J.EQ.1) WRITE (I6I,100) XX,YY,(RA(K),K=1,N)       00041
ISN 0026  IF (J.GT.1) WRITE (I6I,90) YY,(RA(K),K=1,N)          00042
ISN 0028  IF (N.GE.N2VAR) GO TO 20                               00043
ISN 0030  N=5                                                     00044
ISN 0031  M=(N2VAR-1)/4                                          00045
ISN 0032  DO 10 K=1,M                                            00046
ISN 0033  L=N+3                                                  00047
ISN 0034  IF (L.GT.N2VAR) L=N2VAR                                00048
ISN 0036  WRITE (I6I,110) (RA(KK),KK=N,L)                       00049
ISN 0037  10 N=L+1                                              00050
ISN 0039  20 CONTINUE                                           00051
ISN 0039  30 CONTINUE                                           00052
ISN 0040  RETURN                                                 00053
C      -----00054

```

282

	C	FORMATS	
	C	-----	00055
ISN 0041	40	FORMAT (//5X,5HTITLE/5X,20A4)	00056
ISN 0042	50	FORMAT (1H1,4X,35HTWO-VARIABLE FUNCTION SPACE RESULTS)	00057
ISN 0043	60	FORMAT (20A4)	00058
ISN 0044	70	FORMAT (16I5)	00059
ISN 0045	80	FORMAT (5X,10I5)	00060
ISN 0046	90	FORMAT (/15X,E12.4,3X,4E13.4)	00061
ISN 0047	100	FORMAT (/3X,2E12.4,3X,4E13.4)	00062
ISN 0048	110	FORMAT (30X,4E13.4)	00063
ISN 0049	120	FORMAT (///5X,48HGLOBAL NUMBER ASSOCIATED WITH X-VARIABLE, N2VX =	00064
		1,I5//5X,48HGLOBAL NUMBER ASSOCIATED WITH Y-VARIABLE, N2VY =,I5)	00065
ISN 0050	130	FORMAT (//5X,37HGLOBAL NUMBERS ASSOCIATED WITH F(X,Y))	00066
ISN 0051	140	FORMAT (//10X,1HX,11X,1HY,20X,6HF(X,Y))	00067
ISN 0052	150	FORMAT (5E15.8)	00068
ISN 0053		END	00069
			00070

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
 *OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
 STATISTICS SOURCE STATEMENTS = 52, PROGRAM SIZE = 1470, SUBPROGRAM NAME =COPE07
 STATISTICS NO DIAGNOSTICS GENERATED
 ***** END OF COMPILATION *****

3004K BYTES OF CORE NOT USED

REQUESTED OPTIONS: SOURCE,NOMAP,NOXREF,NOLIST,NODECK,OPT(3),AUTODBL(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTHT NOXREF NOALC NOANSF TERM IBM FLAG(I)

	C	DATA SET U477COPE08 AT LEVEL 001 AS OF 02/13/81	
	C	DATA SET 9188COPE08 AT LEVEL 001 AS OF 07/10/80	00001
ISN 0002		SUBROUTINE COPE08 (A,B,IFORM,NFLD)	00002
ISN 0003		DIMENSION A(1), B(1), C(10)	00003
ISN 0004		DATA COMMA/1H,/,BLANK/1H /	00004
	C	ROUTINE TO CONVERT UNFORMATTED DATA TO FORMATTED DATA IN FIELDS	00005
	C	OF 10,EACH FIELD RIGHT JUSTIFIED.	00006
	C	BY G. N. VANDERPLAATS AUG., 1978.	00007
	C	NASA AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.	00008
	C	--- INPUT.	00009
	C	A - ARRAY OF DATA SEPERATED BY COMMAS. MINIMUM DIMENSION OF	00010
	C	IS A(80).	00011
	C	--- OUTPUT.	00012
	C	B - ARRAY OD DATA IN FIELDS OF 10 AND RIGHT JUSTIFIED.	00013
	C	MINIMUM DIMENSION OF B IS 10*NFLD.	00014
	C	IFORM - 0 IF A WAS UNFORMATTED.	00015
	C	1 IF A WAS ALLREADY FORMATTED.	00016
	C	NFLD - NUMBER OF FIELDS OF DATA.	00017
	C	--- NOTE.	00018
	C	1) DATA IS ASSUMED TO BE REAL OR INTEGER DATA WITH NO EMBEDDED	00019
	C	BLANKS WITHIN A GIVEN FIELD.	00020
	C	2) DATA IS CONSIDERED UNFORMATTED IF	00021
	C	A) A COMMA IS FOUND	00022
	C	B) LAST NON-BLANK CHARACTER IS IN COLUMN 1-10 AND THERE IS	00023
	C	NO DECIMAL AND IT IS NOT RIGHT JUSTIFIED.	00024
	C		00025
ISN 0005		IFORM=0	00026
	C	SEARCH FOR LAST NON-BLANK CHARACTER AND SEARCH FOR COMMA.	00027
	C	CALCULATE NUMBER OF NON-BLANK SETS.	00028
		INON=0	00029
ISN 0006		KNON=0	00030
ISN 0007		LST=0	00031
ISN 0008			00032
ISN 0009		DO 10 I=1,80	00033
ISN 0010		IF (A(I).EQ.COMMA) GO TO 20	00034
ISN 0012		JNON=INON	00035
ISN 0013		IF (A(I).EQ.BLANK) INON=0	00036
ISN 0015		IF (A(I).NE.BLANK) INON=1	00037
ISN 0017		IF (INON.GT.JNON) KNON=KNON+1	00038
ISN 0019		IF (A(I).NE.BLANK) LST=I	00039
ISN 0021	10	CONTINUE	00040
	C	NO COMMA WAS FOUND. DATA MAY BE FORMATTED.	00041
ISN 0022		IF (LST.GE.10) GO TO 90	00042
	C	IF MORE THAN ONE SETS OF CHARACTERS, DATA IS ASSUMED FORMATTED.	00043
ISN 0024		IF (KNON.GT.1) GO TO 90	00044
ISN 0026	20	CONTINUE	00045
	C	DATA IS UNFORMATTED.	00046
ISN 0027		K2=10	00047
ISN 0028		NFLD=0	00048
ISN 0029		I=0	00049
ISN 0030	30	CONTINUE	00050
ISN 0031		I=I+1	00051
ISN 0032		IF (I.GT.80) GO TO 110	00052
	C	IGNORE LEADING BLANKS.	00053
ISN 0034		IF (A(I).EQ.BLANK) GO TO 30	00054
	C	CALCULATE NUMBER OF NON-BLANK CHARACTERS IN THIS FIELD.	00054

```

+VERSION 1.3.0 (01 MAY 80)   COPE08   SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)   DATE 82.141/10.54.55   PAGE 2
  ISN 0036           JJ=0                               00055
  ISN 0037           DO 40 J=I,80                       00056
  ISN 0038           IF (A(J).EQ.COMMA.OR.A(J).EQ.BLANK) GO TO 50 00057
  ISN 0040           JJ=JJ+1                             00058
  ISN 0041           40 C(JJ)=A(J)                       00059
  ISN 0042           50 NFLD=NFLD+1                     00060
  ISN 0043           I=I+JJ                             00061
                   C   BLANK FIELD NFLD OF B.          00062
  ISN 0044           K1=K2-9                             00063
  ISN 0045           DO 60 K=K1,K2                       00064
  ISN 0046           60 B(K)=BLANK                       00065
                   C   STORE C IN FIELD NFLD OF B, RIGHT JUSTIFIED. 00066
  ISN 0047           IF (JJ.EQ.0) GO TO 80               00067
  ISN 0049           J1=JJ                               00068
  ISN 0050           K=K2                               00069
  ISN 0051           DO 70 L=1,JJ                       00070
  ISN 0052           B(K)=C(J1)                         00071
  ISN 0053           K=K-1                              00072
  ISN 0054           70 J1=J1-1                         00073
  ISN 0055           80 K2=K2+10                       00074
  ISN 0056           GO TO 30                           00075
  ISN 0057           90 CONTINUE                         00076
                   C   FORMATTED INPUT. STORE A DIRECTLY IN B. 00077
  ISN 0058           IFORM=1                             00078
  ISN 0059           NFLD=8                             00079
  ISN 0060           DO 100 I=1,80                      00080
  ISN 0061           100 B(I)=A(I)                     00081
  ISN 0062           110 CONTINUE                       00082
  ISN 0063           RETURN                             00083
  ISN 0064           END                                00084

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODEL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 63, PROGRAM SIZE = 846, SUBPROGRAM NAME =COPE08
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

3012K BYTES OF CORE NOT USED

286

```

C      CONMIN ARRAYS.                                00055
C      DIMENSIONS.                                    00056
ISN 0049      NN1 = NDV + 2                            00057
ISN 0050      NN2 = 2*NDV + NCON                       00058
ISN 0051      NN3 = NACMX1                             00059
ISN 0052      NN4 = NN3                               00060
ISN 0053      IF (NDV .GT. NN4) NN4 = NDV             00061
ISN 0055      NN5 = 2*NN4                              00062
C      SCAL, DF, G, A, S, G1, G2, C, B, ISC, IC, MS1. 00063
ISN 0056      NMSCAL = LCCR(4)                         00064
ISN 0057      NDF = NTMP                               00065
ISN 0058      NG = NDF + NN1                          00066
ISN 0059      NA = NG + NN2                           00067
ISN 0060      NS = NA + NN1*NN3                       00068
ISN 0061      NG1 = NS + NN1                          00069
ISN 0062      NG2 = NG1 + NN2                         00070
ISN 0063      NC = NG2 + NN2                          00071
ISN 0064      NB = NC + NN4                           00072
ISN 0065      NISC = LOCI(4)                          00073
ISN 0066      NIC = NIDV + NXAPRX                     00074
ISN 0067      NMS1 = NIC + NN3                        00075
C      -----00076
C      DETERMINE IOBJA, ARRAYS IGFN AND IDV.          00077
C      -----00078
C      IOBJA.                                         00079
ISN 0068      M6 = LOCI(6)                             00080
ISN 0069      DO 10 I=1,NF                             00081
ISN 0070          J = IA(M6)                          00082
ISN 0071          IOBJA = I                            00083
ISN 0072          IF (J .EQ. IOBJ) GO TO 20            00084
ISN 0074          M6 = M6+1                            00085
ISN 0075      10 CONTINUE                             00086
C      ERROR - IOBJA NOT FOUND.                       00087
ISN 0076      20 CONTINUE                             00088
ISN 0077      IF (NCONA .EQ. 0) GO TO 60              00089
C      IGFN ARRAY.                                    00090
ISN 0079      M3 = LOCI(3)                             00091
ISN 0080      M23 = LOCI(23)                          00092
ISN 0081      DO 50 I=1,NCONA                          00093
COMMENT: ----GLOBAL LOCATIONS OF CONSTRAINED PARAMETERS.-----00094
ISN 0082          J = IA(M3)                           00095
ISN 0083          M3 = M3 + 1                          00096
COMMENT: ----LOCAL VARIABLE, F, LOCATION.-----00097
ISN 0084          M6 = LOCI(6)                         00098
ISN 0085          DO 30 K=1,NF                          00099
ISN 0086              KK = K                           00100
ISN 0087              L = IA(M6)                       00101
ISN 0088              IF (L .EQ. J) GO TO 40           00102
ISN 0090              M6 = M6 + 1                       00103
ISN 0091          30 CONTINUE                          00104
COMMENT: ERRCR - CONSTRAINED VARIABLE IS NOT AN APPROXIMATE FUNCTION. 00105
ISN 0092          40 CONTINUE                          00106
ISN 0093              IA(M23) = KK                      00107
ISN 0094              M23 = M23 + 1                    00108
ISN 0095          50 CONTINUE                          00109
ISN 0096          60 CONTINUE                          00110
COMMENT: ----IDV ARRAY.-----00111
ISN 0097          N3 = NIDV                            00112
ISN 0098          N5 = LOCI(5)                        00113

```

```

+VERSION 1.3.0 (01 MAY 80)   COPE09   SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)   DATE 82.141/10.54.59   PAGE 3
ISN 0099      DO 90 I=1,NXAPRX      00114
ISN 0100      IA(N3) = 0      00115
COMMENT: ----GLOBAL LOCATION.-----00116
ISN 0101      II = IA(N5)      00117
ISN 0102      N5 = N5 + 1      00118
COMMENT: ----FIND CORRESPONDING DESIGN VARIABLE.-----00119
ISN 0103      N1 = LOCI(1)      00120
ISN 0104      N2 = LOCI(2)      00121
ISN 0105      DO 70 J=1,NDVTOT    00122
ISN 0106      IF (IA(N1) .EQ. II) GO TO 80      00123
ISN 0108      N1 = N1 + 1      00124
ISN 0109      N2 = N2 + 1      00125
ISN 0110      70 CONTINUE      00126
COMMENT: ----THIS APPROXIMATING VARIABLE IS NOT A DESIGN VARIABLE.-----00127
ISN 0111      GO TO 90      00128
ISN 0112      80 CONTINUE      00129
COMMENT: ----DESIGN VARIABLE NUMBER.-----00130
ISN 0113      IA(N3) = IA(N2)      00131
ISN 0114      N3 = N3 + 1      00132
ISN 0115      90 CONTINUE      00133
COMMENT: ---CHECK TO BE SURE EACH INDEPENDENT DESIGN VARIABLE IS ALSO AN00134
COMMENT: ---APPROXIMATING VARIABLE.      00135
ISN 0116      DO 120 I=1,NDV      00136
ISN 0117      N1 = LOCI(1)      00137
ISN 0118      N2 = LOCI(2)      00138
ISN 0119      DO 111 J=1,NDVTOT    00139
COMMENT: ----IS THIS DESIGN VARIABLE I.-----00140
ISN 0120      IF (IA(N2) .NE. I) GO TO 110      00141
COMMENT: ----YES.-----00142
COMMENT: ----GLOBAL VARIABLE NUMBER.-----00143
ISN 0122      IGLOBAL = IA(N1)      00144
ISN 0123      N1 = N1+1      00145
ISN 0124      N5 = LOCI(5)      00146
ISN 0125      DO 100 K=1,NXAPRX    00147
COMMENT: ----IS THIS THE SAME AS IGLOBAL.-----00148
ISN 0126      IF (IA(N5) .EQ. IGLOBAL) GO TO 120      00149
COMMENT: ----NO.-----00150
ISN 0128      N5 = N5+1      00151
ISN 0129      100 CONTINUE      00152
ISN 0130      110 N2=N2+1      00153
ISN 0131      111 CONTINUE      00154
COMMENT: ----ERROR - DESIGN VARIABLE IS NOT AN APPROXIMATING VARIABLE.---00155
ISN 0132      120 CONTINUE      00156
C -----00157
C BEGIN SEQUENTIAL APPROXIMATE OPTIMIZATION.      00158
C -----00159
ISN 0133      ICK1 = 0      00160
ISN 0134      ICK2 = 0      00161
ISN 0135      ICK3 = 0      00162
ISN 0136      130 CONTINUE      00163
ISN 0137      KOUNT = KOUNT + 1      00164
ISN 0138      IF (IPAPRX .LT. 1 .OR. IPAPRX .EQ. 3) GO TO 160      00165
ISN 0140      IF (KMAX .LT. 0) GO TO 160      00166
COMMENT:----- PRINT INITIAL INFORMATION.      00167
COMMENT:----- TITLE.      00168
ISN 0142      IF (KOUNT .GT. 1) GO TO 150      00169
ISN 0144      WRITE (I6I,670)      00170
COMMENT:----- OBJECTIVE FUNCTION.      00171
ISN 0145      WRITE (I6I,550) IOBJA      00172

```

288

```

ISN 0146      IF (NCONA .EQ. 0)                GO TO 140                00173
COMMENT:----- CONSTRAINTS.                00174
ISN 0148      WRITE (I6I,560)                  00175
ISN 0149      N1 = NIGFN                        00176
ISN 0150      N2 = N1 + NCONA - 1              00177
ISN 0151      WRITE (I6I,570) (IA(I),I=N1,N2)  00178
ISN 0152      140 CONTINUE                     00179
COMMENT:----- DESIGN VARIABLES.           00180
ISN 0153      WRITE (I6I,580)                  00181
ISN 0154      N1 = NIDV                        00182
ISN 0155      N2 = N1 + NXAPRX - 1            00183
ISN 0156      WRITE (I6I,570) (IA(I),I=N1,N2)  00184
ISN 0157      150 CONTINUE                     00185
COMMENT:----- ITERATION NUMBER.           00186
ISN 0158      WRITE (I6I,680) KOUNT            00187
ISN 0159      160 CONTINUE                     00188
ISN 0160      NP = NPTOT - 1                  00189
C ----- 00190
C          SET UP ARRAYS XNOM, FNOM, XI, Y.    00191
C ----- 00192
ISN 0161      NXI = NTHP                       00193
ISN 0162      NY = NXI + NXAPRX*NPTOT          00194
ISN 0163      NNIGHT = NY + NF*NPTOT           00195
ISN 0164      IF (KMAX .LT. 0 .AND. INOM .LT. 1) INOM = 1  00196
COMMENT:----- 00197
ISN 0166      CALL COPE10 (RA(NXI),RA(NY),RA(NXNOM),RA(NFNOM),NPTOT,KOUNT,RA(NBL00198
IU),IA(NIGFN),IOBJA,IA(NISC),NXAPRX,NF,NCONA,SGNOPT,CTMIN,CTLMIN,IS00199
2CR2,RA(NNIGHT),INOM,NPMAX,JNOM)              00200
COMMENT:----- 00201
ISN 0167      IF (KMAX .LT. 0) GO TO 170        00202
COMMENT:----- CURRENT OBJECTIVE.           00203
ISN 0169      N1 = NFNOM + IOBJA - 1            00204
ISN 0170      CBJSVAV = -RA(N1)*SGNOPT          00205
ISN 0171      IF (KOUNT .LT. 2 .AND. DABFUN .LE. 0.) DABFUN = .001*ABS(OBJSVAV) 00206
ISN 0173      IF (DABFUN .LT. 1.0E-10) DABFUN = 1.0E-10  00207
ISN 0175      IF (IPAPRX .LT. 1 .OR. IPAPRX .EQ. 3) GO TO 170  00208
COMMENT:----- PRINT CURRENT NOMINAL.       00209
ISN 0177      WRITE (I6I,690) INOM             00210
ISN 0178      N2 = NXNOM + NXAPRX - 1          00211
ISN 0179      WRITE (I6I,700) (RA(I),I=NXNOM,N2)  00212
ISN 0180      WRITE (I6I,710)                  00213
ISN 0181      N2 = NFNOM + NF - 1              00214
ISN 0182      WRITE (I6I,700) (RA(I),I=NFNOM,N2)  00215
ISN 0183      170 CONTINUE                     00216
C ----- 00217
C          LEAST SQUARES FIT FOR TAYLER SERIES EXPANSION.  00218
C ----- 00219
ISN 0184      NX1 = NXAPRX                      00220
ISN 0185      M = NX1 + (NX1*(NX1 + 1))/2      00221
ISN 0186      IF (M .GT. NP) M = NP            00222
ISN 0188      IF (NP .LT. NXAPRX) M = NXAPRX   00223
ISN 0190      MMAX = NXAPRX*MAXTRM             00224
ISN 0191      IF (MAXTRM .GT. 2) MMAX = M      00225
ISN 0193      IF (M .GT. MMAX) M = MMAX        00226
ISN 0195      NAA = NNIGHT + NP                00227
ISN 0196      NFF = NAA + (M*(M + 1))/2       00228
ISN 0197      NSG = NFF + M                   00229
COMMENT:----- 00230
ISN 0198      CALL COPE12 (RA(NXI),RA(NY),NX1,NP,NF,M,RA(NBTAY),RA(NAA),RA(NFF),00231

```

```

IRA(NGG),NXAPRX,NF,NBR,RA(NNGHT),NER)                                00232
COMMENT:-----                                                    00233
ISN 0199      IF (NER .GT. 0) WRITE (I6I,590)                       00234
ISN 0201      IF (KMAX .LT. 0) GO TO 530                            00235
ISN 0203      IF (IPAPRX .LT. 2 .AND. IPAPRX .NE. 4) GO TO 190     00236
COMMENT:-----PRINT TAYLER SERIES COEFFICIENTS.-----          00237
ISN 0205      WRITE (I6I,720)                                       00238
ISN 0206      N3 = LOCI(6)                                           00239
ISN 0207      N1 = N3TAY                                             00240
ISN 0208      DO 180 J=1,NF                                          00241
ISN 0209      N2 = N1 + M - 1                                        00242
ISN 0210      WRITE (I6I,600) J,IA(N3)                              00243
ISN 0211      WRITE (I6I,700) (RA(I),I=N1,N2)                      00244
ISN 0212      N1 = N1 + NBR                                         00245
ISN 0213      N3 = N3 + 1                                           00246
ISN 0214      180 CONTINUE                                          00247
ISN 0215      190 CONTINUE                                          00248
ISN 0216      IF (KOUNT .GT. KMAX) GO TO 470                       00249
C-----
C      INITIALIZE XV, DX, VLB, VUB.                                00251
C-----
C      N1 = NXV                                                    00252
ISN 0218      N1 = NXV                                              00253
ISN 0219      DO 200 I=1,NXAPRX                                     00254
ISN 0220      RA(N1) = 0.                                           00255
ISN 0221      N1 = N1 + 1                                           00256
ISN 0222      200 CONTINUE                                          00257
ISN 0223      N2 = NVLB                                             00258
ISN 0224      N3 = NVUB                                             00259
ISN 0225      N4 = NXNOM                                            00260
ISN 0226      N5 = LOCR(2)                                          00261
ISN 0227      N6 = LOCR(3)                                          00262
ISN 0228      N7 = LOCR(7)                                          00263
ISN 0229      N8 = NDX                                              00264
ISN 0230      L1 = M - NDV                                           00265
ISN 0231      L2 = L1 - NDV                                          00266
ISN 0232      ICK = ICK1 + ICK2 + ICK3                             00267
ISN 0233      DO 210 I=1,NDV                                         00268
ISN 0234      RA(N8) = 0.                                           00269
ISN 0235      XFACT = 1.                                            00270
ISN 0236      IF (I .LE. L1) XFACT = XFACT1                         00271
ISN 0238      L2 = L2 - NDV + I                                       00272
ISN 0239      IF (L2 .GE. 0) XFACT = XFACT2                         00273
COMMENT:-----REDUCE BOUNDS IF ANY ICK.GT.0.-----          00274
ISN 0241      IF (ICK .GT. 0) XFACT = .5                             00275
ISN 0243      DX = XFACT*RA(N7)                                       00276
ISN 0244      XX = RA(N4)                                           00277
ISN 0245      DXL = XX - RA(N5)                                       00278
ISN 0246      IF (DXL .GT. DX) DXL = DX                               00279
ISN 0248      DXU = RA(N6) - XX                                       00280
ISN 0249      IF (DXU .GT. DX) DXU = DX                               00281
ISN 0251      RA(N2) = -DXL                                           00282
ISN 0252      RA(N3) = DXU                                           00283
ISN 0253      N2 = N2 + 1                                           00284
ISN 0254      N3 = N3 + 1                                           00285
ISN 0255      N4 = N4 + 1                                           00286
ISN 0256      N5 = N5 + 1                                           00287
ISN 0257      N6 = N6 + 1                                           00288
ISN 0258      N7 = N7 + 1                                           00289
ISN 0259      N8 = N8 + 1                                           00290

```

290

```

ISN 0260      210 CONTINUE                                00291
ISN 0261      IF (IPAPRX .LT. 2 .AND. IPAPRX .NE. 4) GO TO 220 00292
ISN 0263      WRITE (I6I,610)                            00293
ISN 0264      N1 = NVLB + NDV - 1                        00294
ISN 0265      WRITE (I6I,700) (RA(I),I=NVLB,N1)          00295
ISN 0266      WRITE (I6I,620)                            00296
ISN 0267      N1 = NVUB + NDV - 1                        00297
ISN 0268      WRITE (I6I,700) (RA(I),I=NVUB,N1)          00298
ISN 0269      220 CONTINUE                                00299
C -----00300
C              OPTIMIZE APPROXIMATE FUNCTION.            00301
C -----00302
COMMENT:----- OPTIMIZATION.                            00303
ISN 0270      IGOTO = 0                                   00304
C              CALL CONMIN (X,VLB,VUB,G,SCAL,DF,A,S,G1,G2,B,C,ISC,IC,MS1,N1,N2,N300305
C              *,N4,N5)                                   00306
ISN 0271      230 CONTINUE                                00307
COMMENT:-----00308
ISN 0272      CALL CONMIN (RA(NDX),RA(NVLB),RA(NVUB),RA(NG),RA(NNSCAL),RA(NDF),R00309
              1A(NA),RA(NS),RA(NG1),RA(NG2),RA(HB),RA(HC),IA(NISC),IA(NIC),IA(NMS00310
              21),NN1,NN2,NN3,NN4,NN5)                   00311
COMMENT:-----00312
COMMENT:----- TRANSFER VARIABLES FROM DX TO XV.        00313
ISN 0273      N1 = NIDV                                   00314
ISN 0274      N2 = NXV                                    00315
ISN 0275      DO 240 I=1,NXAPRX                           00316
ISN 0276      II = IA(N1)                                  00317
ISN 0277      N4 = NDX + II - 1                           00318
ISN 0278      IF (II .GT. 0 .AND. II .LE. NDV) RA(N2) = RA(N4) 00319
ISN 0280      N1 = N1 + 1                                  00320
ISN 0281      N2 = N2 + 1                                  00321
ISN 0282      N3 = N3 + 1                                  00322
ISN 0283      240 CONTINUE                                00323
COMMENT:----- APPROXIMATE ANALYSIS.                    00324
COMMENT:-----00325
ISN 0284      CALL COPE15 (RA(NXV),RA(NG),RA(NDF),RA(NA),IA(NISC),IA(NIC),NN1,RA00326
              1(NBLU),NX1,IOBJA,M,RA(NFNOM),RA(NFNEH),RA(NBTAY),NBR,IA(NIGFN),CT,00327
              2CTL,INFO,NAC,NCONA,NDV,NF,OBJ,SGNOPT)      00328
COMMENT:-----00329
ISN 0285      IF (IGOTO .GT. 0) GO TO 230                 00330
C              IF DESIGN PRODUCED ZERO DELTA-X TWICE IN A ROW AND KOUNT.GE.KMIN, 00331
COMMENT:----- TERMINATE.                                00332
ISN 0287      ICK1 = ICK1 + 1                              00333
ISN 0288      SUM = 0.                                     00334
ISN 0289      N1 = NXV                                    00335
ISN 0290      DO 250 I=1,NDV                              00336
ISN 0291      SUM = SUM + RA(N1)**2                       00337
ISN 0292      N1 = N1 + 1                                  00338
ISN 0293      250 CONTINUE                                00339
ISN 0294      IF (SUM .GT. 1.0E-10) ICK1=0               00340
ISN 0296      IF (IPAPRX .GT. 0 .AND. IPAPRX .NE. 3) WRITE (I6I,730) 00341
ISN 0298      IF (ICK1 .GE. 2 .AND. KOUNT .GE. KMIN) GO TO 360 00342
C -----00343
C              INSURE NEW X-VECTOR IS INDEPENDENT        00344
C -----00345
ISN 0300      JJJ = 0                                      00346
ISN 0301      260 JJJ = JJJ+1                              00347
COMMENT:----- NOMINAL X-VECTOR.                        00348
ISN 0302      N1 = NTMP                                    00349
    
```

```

+VERSION 1.3.0 (01 MAY 80)   COPE09   SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)   DATE 82.141/10.54.59   PAGE 7
  ISN 0303      N2      = HXNOM      00350
  ISN 0304      N3      = NXV        00351
  ISN 0305      DO 270 I=1,NXAPRX    00352
  ISN 0306      RA(N1) = RA(N2) + RA(N3) 00353
  ISN 0307      N1      = N1 + 1      00354
  ISN 0308      N2      = N2 + 1      00355
  ISN 0309      N3      = N3 + 1      00356
  ISN 0310      270 CONTINUE          00357
  COMMENT:----- READ X-VECTORS ONE AT A TIME AND COMPARE TO XNOM.00358
  ISN 0311      REWIND ISCR2          00359
  ISN 0312      N1      = NTMP + NXAPRX 00360
  ISN 0313      N2      = N1 + NXAPRX - 1 00361
  ISN 0314      N3      = N2 + 1      00362
  ISN 0315      N4      = N3 + NF - 1    00363
  ISN 0316      DO 290 J=1,NPTOT      00364
  ISN 0317      KK      = J            00365
  COMMENT:----- X-VECTOR.          00366
  ISN 0318      READ (ISCR2) (RA(I),I=N1,N2) 00367
  COMMENT:----- Y-VECTOR. NOT USED. READ TO POSITION ISCR2. 00368
  ISN 0319      READ (ISCR2) (RA(I),I=N3,N4) 00369
  COMMENT:----- COMPARE X WITH XNOM. 00370
  ISN 0320      N5      = N1            00371
  ISN 0321      N6      = NTMP          00372
  ISN 0322      SUM     = 0.            00373
  ISN 0323      DO 280 I=1,NXAPRX      00374
  ISN 0324      SUM     = SUM + (RA(N5) - RA(N6))*2 00375
  ISN 0325      N5      = N5+1          00376
  ISN 0326      N6      = N6+1          00377
  ISN 0327      280 CONTINUE           00378
  ISN 0328      IF (SUM.LT.1.0E-10) GO TO 300 00379
  ISN 0330      290 CONTINUE           00380
  ISN 0331      GO TO 360               00381
  ISN 0332      300 CONTINUE           00382
  C THIS DESIGN IS SAME AS A PREVIOUS DESIGN. 00383
  C MODIFY DELTA-X VECTOR.            00384
  ISN 0333      N6=NVUB                 00385
  ISN 0334      N7=NXV                  00386
  ISN 0335      N8=NVLB                 00387
  ISN 0336      N9=NXV+NDV-1            00388
  ISN 0337      IF (IPAPRX.LT.1.OR.IPAPRX.EQ.3) GO TO 310 00389
  ISN 0339      WRITE (I6I,630)         00390
  ISN 0340      WRITE (I6I,740)         00391
  ISN 0341      WRITE (I6I,700) (RA(I),I=N7,N9) 00392
  ISN 0342      WRITE (I6I,750)         00393
  ISN 0343      N9=NTMP+NXAPRX-1       00394
  ISN 0344      WRITE (I6I,700) (RA(I),I=NTMP,N9) 00395
  ISN 0345      WRITE (I6I,640)         00396
  ISN 0346      310 CONTINUE           00397
  ISN 0347      AMULT=.01*FLOAT(JJJ)    00398
  ISN 0348      DO 320 I=1,NDV          00399
  ISN 0349      BU=RA(N6)               00400
  ISN 0350      BL=RA(N8)               00401
  ISN 0351      IF (BL.LT.-1.0E+15) BL=0. 00402
  ISN 0353      IF (BU.GT.1.0E+15) BU=0. 00403
  ISN 0355      DB=ABS(BU-BL)           00404
  ISN 0356      IF (DB.LT.1.0E-6) DB=.1 00405
  ISN 0358      DX=RA(N7)+AMULT*DB     00406
  ISN 0359      IF (DX.GT.RA(N6)) DX=DX-2.*AMULT*DB 00407
  ISN 0361      IF (DX.LT.RA(N8)) DX=DX+1.5*AMULT*DB 00408

```

292

```

ISN 0363      RA(N7)=DX                                00409
ISN 0364      N5=N6+1                                  00410
ISN 0365      N7=N7+1                                  00411
ISN 0366      N9=N9+1                                  00412
ISN 0367      320  CONTINUE                             00413
ISN 0368      CALL COPE15 (RA(NXV),RA(NG),RA(NDF),RA(NA),IA(NISC),IA(NIC),NN1,RA00414
              1(NBLU),NX1,IOBJA,M,RA(NFHOM),RA(NFNEW),RA(NBTAY),NBR,IA(NIGFN),CT,00415
              2CTL,INFO,NAC,NCONA,NDV,NF,OBJ,SGNOPT)    00416
ISN 0369      IF (JJJ.LT.4) GO TO 260                  00417
              C   FOUR TRIES HAVE FAILED TO PRODUCE A USABLE X-VECTOR. 00418
              C   USE LATEST TRY.                          00419
              C   POSITION ISCR2 IF NEEDED.                  00420
ISN 0371      IF (KK.EQ.NPTOT) GO TO 340               00421
ISN 0373      KK=KK+1                                   00422
ISN 0374      DO 330 J=KK,NPTOT                         00423
ISN 0375      READ (ISCR2) (RA(I),I=N1,N2)            00424
ISN 0376      330  READ (ISCR2) (RA(I),I=N3,N4)        00425
ISN 0377      340  CONTINUE                             00426
ISN 0378      IF (IPAPRX.LT.1.OR.IPAPRX.EQ.3) GO TO 350 00427
ISN 0380      IF (JJJ.EQ.4) WRITE (I6I,650)           00428
ISN 0382      350  CONTINUE                             00429
ISN 0383      360  CONTINUE                             00430
              C   -----00431
              C   UPDATE ANALYSIS.                        00432
              C   -----00433
              C   XNOM.                                    00434
ISN 0384      N1=NXNOM                                  00435
ISN 0385      N2=NXV                                    00436
ISN 0386      DO 370 I=1,NXAPRX                        00437
ISN 0387      RA(N1)=RA(N1)+RA(N2)                    00438
ISN 0388      N1=N1+1                                  00439
ISN 0389      370  N2=N2+1                              00440
              C   GLOBAL VARIABLES.                      00441
ISN 0390      N3=NXNOM                                  00442
ISN 0391      N4=NIDV                                   00443
ISN 0392      DO 410 I=1,NXAPRX                        00444
              C   DESIGN VARIABLE NUMBER.                00445
ISN 0393      II=IA(N4)                                00446
ISN 0394      IF (II.EQ.0) GO TO 400                   00447
              C   DESIGN VARIABLE UPDATE.                00448
ISN 0396      N1=LOCI(1)                               00449
ISN 0397      N2=LOCI(2)                               00450
ISN 0398      N5=LOCR(5)                               00451
ISN 0399      DO 390 J=1,NDVTOT                       00452
ISN 0400      IF (IA(N2).NE.II) GO TO 380             00453
              C   UPDATE VARIABLE J.                     00454
ISN 0402      JJ=IA(N1)                                00455
ISN 0403      ARRAY(JJ)=RA(N3)*RA(N5)                 00456
ISN 0404      380  N1=N1+1                              00457
ISN 0405      N2=N2+1                                  00458
ISN 0406      N5=N5+1                                  00459
ISN 0407      390  CONTINUE                             00460
ISN 0408      400  N3=N3+1                              00461
ISN 0409      410  N4=N4+1                              00462
ISN 0410      IF (IPAPRX.LT.1.OR.IPAPRX.EQ.3) GO TO 420 00463
              C   PRINT APPROXIMATE OPTIMIZATION INFORMATION. 00464
ISN 0412      WRITE (I6I,740)                         00465
ISN 0413      N2=NXV+NDV-1                             00466
ISN 0414      WRITE (I6I,700) (RA(I),I=NXV,N2)        00467
    
```

```

+VERSION 1.3.0 (01 MAY 80)   COPE09   SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)   DATE 82.141/10.54.59   PAGE 9
ISN 0415   WRITE (I6I,750)   00468
ISN 0416   N2=NXNOM+NXAPRX-1   00469
ISN 0417   WRITE (I6I,700) (RA(I),I=NXNOM,N2)   00470
ISN 0418   WRITE (I6I,760)   00471
ISN 0419   N2=NFNEW+NF-1   00472
ISN 0420   WRITE (I6I,700) (RA(I),I=NFNEW,N2)   00473
ISN 0421   420 CONTINUE   00474
ISN 0422   IF ((ICK1.GE.2.AND.KOUNT.GE.KMIN).AND.(IPAPRX.GT.0.AND.IPAPRX.NE.300475
1)) WRITE (I6I,540)   00476
ISN 0424   IF (ICK1.GE.2.AND.KOUNT.GE.KMIN) GO TO 460   00477
ISN 0426   ICALC=2   00478
ISN 0427   NAN2=NAN2+1   00479
ISN 0428   CALL ANALIZ (ICALC)   00480
C NEW FUNCTION VALUES.   00481
ISN 0429   NI=NFNOM   00482
ISN 0430   M6=LOCI(6)   00483
ISN 0431   DO 430 I=1,NF   00484
ISN 0432   II=IA(M6)   00485
ISN 0433   M6=M6+1   00486
ISN 0434   RA(NI)=ARRAY(II)   00487
ISN 0435   430 NI=NI+1   00488
ISN 0436   IF (IPDBG.LT.1) GO TO 440   00489
C DEBUG OUTPUT.   00490
ISN 0438   NAN3=NAN3+1   00491
ISN 0439   ICALC=3   00492
ISN 0440   CALL ANALIZ (ICALC)   00493
ISN 0441   440 CONTINUE   00494
ISN 0442   IF (IPAPRX.LT.1.OR.IPAPRX.EQ.3) GO TO 450   00495
C PRINT PRECISE FUNCTION VALUES.   00496
ISN 0444   WRITE (I6I,770)   00497
ISN 0445   N2=NFNOM+NF-1   00498
ISN 0446   WRITE (I6I,700) (RA(I),I=NFNOM,N2)   00499
ISN 0447   450 CONTINUE   00500
C NEW OBJECTIVE.   00501
ISN 0448   NI=NFNOM+ICBJA-1   00502
ISN 0449   OBJ=-RA(NI)*SGNOPT   00503
C -----00504
C WRITE NEW X AND F VALUES ON ISCR2.   00505
C -----00506
C X-VECTOR.   00507
ISN 0450   NI=NXNOM+NXAPRX-1   00508
ISN 0451   WRITE (ISCR2) (RA(I),I=NXNOM,NI)   00509
C FUNCTIONS.   00510
ISN 0452   NI=NFNOM+NF-1   00511
ISN 0453   WRITE (ISCR2) (RA(I),I=NFNOM,NI)   00512
C UPDATE PARAMETERS.   00513
ISN 0454   NPTOT=NPTOT+1   00514
ISN 0455   IF (JJJ.LT.2.OR.KOUNT.LT.KMIN) INOM=NPTOT   00515
C -----00516
C CONVERGENCE CHECK.   00517
C -----00518
ISN 0457   IF (KOUNT.LT.KMIN) GO TO 130   00519
ISN 0459   ICK2=ICK2+1   00520
ISN 0460   ICK3=ICK3+1   00521
ISN 0461   DEL=ABS(OBJ)   00522
ISN 0462   IF (DEL.LT.1.0E-6) DEL=1.0E-6   00523
ISN 0464   DEL=(OBJ-OBJSAB)/DEL   00524
ISN 0465   DEL=ABS(DEL)   00525
ISN 0466   IF (DEL.GT.DELFUN) ICK2=0   00526

```

294

```

ISN 0468      DEL=ABS(OBJ-OBJSAV)                                00527
ISN 0469      IF (DEL.GT.DABFUN) ICK3=0                          00528
ISN 0471      IF (ICK2.GE.2.OR.ICK3.GE.2) GO TO 460              00529
ISN 0473      IF (KOUNT.LT.KMAX) GO TO 130                       00530
ISN 0475      460 CONTINUE                                       00531
C             -----00532
C             FINAL INFORMATION.                                00533
C             -----00534
ISN 0476      IF (IPAPRX.GT.0.AND.IPAPRX.NE.3) WRITE (I6I,660)   00535
C             GO BACK AND PICK BEST DESIGN.                     00536
ISN 0478      INOM=0                                             00537
ISN 0479      KOUNT=KMAX+1                                       00538
ISN 0480      IF (KOUNT.LT.JNOM) KOUNT=JNOM+1                   00539
ISN 0482      CTSAV=CT                                           00540
ISN 0483      CTLSAV=CTL                                         00541
ISN 0484      IF (ABS(CT).LT.1.0E-10) CT=-.004                  00542
ISN 0486      IF (ABS(CTL).LT.1.0E-10) CTL=-.001                00543
ISN 0488      GO TO 160                                           00544
ISN 0489      470 CONTINUE                                       00545
ISN 0490      CT=CTSAV                                           00546
ISN 0491      CTL=CTLSAV                                         00547
C             STORE FINAL VALUES OF XNOM IN GLOBAL ARRAY.     00548
ISN 0492      N3=NXNOM                                           00549
ISN 0493      N4=NIDV                                           00550
ISN 0494      DO 510 I=1,NXAPRX                                  00551
C             DESIGN VARIABLE NUMBER.                           00552
ISN 0495      II=IA(N4)                                          00553
ISN 0496      IF (II.EQ.0) GO TO 500                             00554
C             DESIGN VARIABLE UPDATE.                           00555
ISN 0498      N1=LOCI(1)                                         00556
ISN 0499      N2=LCCI(2)                                         00557
ISN 0500      N5=LOCR(5)                                         00558
ISN 0501      DO 490 J=1,NDVTOT                                  00559
ISN 0502      IF (IA(N2).NE.II) GO TO 480                       00560
C             UPDATE VARIABLE J.                                00561
ISN 0504      JJ=IA(N1)                                          00562
ISN 0505      ARRAY(JJ)=RA(N3)*RA(N5)                            00563
ISN 0506      480 N1=N1+1                                         00564
ISN 0507      N2=N2+1                                         00565
ISN 0508      N5=N5+1                                         00566
ISN 0509      490 CONTINUE                                       00567
ISN 0510      500 N3=N3+1                                         00568
ISN 0511      510 N4=N4+1                                         00569
C             STORE FINAL VALUES OF FNOM IN GLOBAL ARRAY.     00570
ISN 0512      M6=LOCI(6)                                         00571
ISN 0513      N1=NFNOM                                           00572
ISN 0514      DO 520 I=1,NF                                       00573
ISN 0515      II=IA(M6)                                          00574
ISN 0516      M6=M6+1                                           00575
ISN 0517      ARRAY(II)=RA(N1)                                   00576
ISN 0518      520 N1=N1+1                                         00577
ISN 0519      530 CONTINUE                                       00578
ISN 0520      RETURN                                             00579
C             -----00580
C             FORMATS                                           00581
C             -----00582
ISN 0521      540 FORMAT (//5X,71HTWO CONSECUTIVE APPROXIMATE OPTIMIZATIONS HAVE PRO00583
                    IDUCED THE SAME DESIGN//5X,23HOPTIMIZATION TERMINATED) 00584
ISN 0522      550 FORMAT (/5X,22HAPPROXIMATING FUNCTION,I5,17H IS THE OBJECTIVE) 00585

```

```

ISN 0523 560 FORMAT (/5X,51HAPPROXIMATING FUNCTIONS ASSOCIATED WITH CONSTRAINTS00586
1) 00587
ISN 0524 570 FORMAT (5X,10I5) 00588
ISN 0525 580 FORMAT (//5X,63HDESIGN VARIABLE NUMBERS ASSOCIATED WITH APPROXIMAT00589
ING VARIABLES) 00590
ISN 0526 590 FORMAT (/5X,59H* * LEAST SQUARES FIT TO APPROXIMATION DATA IS SIM00591
IGULAR * */5X,24HRESULTS MAY NOT BE VALID) 00592
ISN 0527 600 FORMAT (/5X,15HFUNCTION NUMBER,I5,25H GLOBAL VARIABLE NUMBER,I5/00593
15X,12HCOEFFICIENTS) 00594
ISN 0528 610 FORMAT (/5X,44HSIDE CONSTRAINTS ON APPROXIMATE OPTIMIZATION//5X,1200595
HLOWER BOUNDS) 00596
ISN 0529 620 FORMAT (/5X,12HUPPER BOUNDS) 00597
ISN 0530 630 FORMAT (//5X,76HOPTIMIZATION HAS PRODUCED AN X-VECTOR WHICH IS THE00598
1 SAME AS A PREVIOUS DESIGN) 00599
ISN 0531 640 FORMAT (/5X,51HTHE FOLLOWING DESIGN IS NOT THE APPROXIMATE OPTIMUM00600
1) 00601
ISN 0532 650 FORMAT (/5X,60HFOUR ATTEMPTS HAVE FAILED TO PRODUCE AN INDEPENDENT00602
1 X-VECTOR/5X,52HOPTIMIZATION WILL CONTINUE WITH MOST RECENT X-VECT00603
2OR) 00604
ISN 0533 660 FORMAT (1H1,4X,40HFINAL RESULT OF APPROXIMATE OPTIMIZATION) 00605
ISN 0534 670 FORMAT (1H1,4X,42HAPPROXIMATE OPTIMIZATION ITERATION HISTORY) 00606
ISN 0535 680 FORMAT (///5X,22HBEGIN ITERATION NUMBER,I5) 00607
ISN 0536 690 FORMAT (/5X,23HNOMINAL DESIGN NUMBER =,I5//5X,8HX-VECTOR) 00608
ISN 0537 700 FORMAT (5X,5E13.5) 00609
ISN 0538 710 FORMAT (/5X,15HFUNCTION VALUES) 00610
ISN 0539 720 FORMAT (//5X,26HTAYLOR SERIES COEFFICIENTS) 00611
ISN 0540 730 FORMAT (//5X,35HRESULTS OF APPROXIMATE OPTIMIZATION) 00612
ISN 0541 740 FORMAT (/5X,14HDELTA-X VECTOR) 00613
ISN 0542 750 FORMAT (/5X,8HX-VECTOR) 00614
ISN 0543 760 FORMAT (/5X,27HAPPROXIMATE FUNCTION VALUES) 00615
ISN 0544 770 FORMAT (/5X,23HPRECISE FUNCTION VALUES) 00616
ISN 0545 END 00617

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODOBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 544, PROGRAM SIZE = 9868, SUBPROGRAM NAME =COPE09
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

2904K BYTES OF CORE NOT USED

REQUESTED OPTIONS: SOURCE,NOMAP,NOXREF,NOLIST,NODECK,OPT(3),AUTODBL(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTHT NOXREF NOALC NOANSF TERM IBM FLAG(I)

296

```

C      DATA SET U477COPE10 AT LEVEL 001 AS OF 02/13/81
C      DATA SET 9188COPE10 AT LEVEL 001 AS OF 07/10/80      00001
ISN 0002  SUBROUTINE COPE10 (XI,Y,XNOM,FNOM,NPTOT,KOUNT,BLU,IGFN,IOBJA,ISC,N00002
          1XAPRX,NF,NCONA,SGNOPT,CTMIN,CTLMIN,ISCR2,WGHT,INOM,HPMAX,JNOM) 00003
ISN 0003  DIMENSION XI(NXAPRX,1), Y(NF,1), XNOM(1), FNOM(1), BLU(4,1), IGFN(00004
          11), ISC(1), WGHT(1)      00005
C      *****00006
C      ROUTINE TO SET UP ARRAYS FOR TAYLER SERIES EXPANSION. 00007
C      *****00008
C      00009
C      BY G. N. VANDERPLAATS      JAN., 1979. 00010
C      NASA AMES RESEARCH CENTER, HOFFETT FIELD, CALIF. 00011
C      00012
ISN 0004  REWIND ISCR2      00013
ISN 0005  DO 10 J=1,NPTOT      00014
ISN 0006  READ (ISCR2) (XI(I,J),I=1,NXAPRX)      00015
ISN 0007  READ (ISCR2) (Y(I,J),I=1,NF)      00016
C      -----00017
C      FIND BEST NOMINAL IF REQUIRED.      00018
C      -----00019
ISN 0008  IF (KOUNT.LE.1.AND.INOM.GT.0) GO TO 20      00020
ISN 0010  IF (KOUNT.GT.1.AND.KOUNT.LE.JNOM) GO TO 20      00021
ISN 0012  CALL COPE11 (NPTOT,Y,NF,INOM,BLU,NCONA,IGFN,IOBJA,SGNOPT,CTMIN,CTL00022
          ININ,ISC)      00023
ISN 0013  20 CONTINUE      00024
C      -----00025
C      CREATE XNOM AND FNOM.      00026
C      -----00027
ISN 0014  DO 30 I=1,NXAPRX      00028
ISN 0015  XNOM(I)=XI(I,INOM)      00029
ISN 0016  DO 40 I=1,NF      00030
ISN 0017  40 FNOM(I)=Y(I,INOM)      00031
ISN 0018  NP=NPTOT-1      00032
ISN 0019  IF (INOM.EQ.NPTOT) GO TO 80      00033
C      -----00034
C      SHIFT XI AND Y.      00035
C      -----00036
ISN 0021  DO 70 J=INOM,NP      00037
ISN 0022  DO 50 I=1,NXAPRX      00038
ISN 0023  50 XI(I,J)=XI(I,J+1)      00039
ISN 0024  DO 60 I=1,NF      00040
ISN 0025  60 Y(I,J)=Y(I,J+1)      00041
ISN 0026  70 CONTINUE      00042
ISN 0027  80 CONTINUE      00043
C      -----00044
C      REPLACE XI BY DELTA-XI AND Y BY DELTA-Y.      00045
C      -----00046
ISN 0028  DO 110 J=1,NP      00047
ISN 0029  DO 90 I=1,NXAPRX      00048
ISN 0030  90 XI(I,J)=XI(I,J)-XNOM(I)      00049
ISN 0031  DO 100 I=1,NF      00050
ISN 0032  100 Y(I,J)=Y(I,J)-FNOM(I)      00051
ISN 0033  110 CONTINUE      00052
C      -----00053
C      WEIGHTING FACTORS.      00054

```

```

C -----00055
ISN 0034          SMAX=1.0E-10          00056
ISN 0035          DO 130 J=1,NP          00057
ISN 0036          SUM=0.                 00058
ISN 0037          DO 120 I=1,NXAPRX      00059
ISN 0038    120   SUM=SUM+XI(I,J)**2     00060
ISN 0039          IF (SUM.GT.SMAX) SMAX=SUM 00061
ISN 0041    130   WGHT(J)=SQRT(SUM)      00062
ISN 0042          SMAX=SQRT(SMAX)        00063
ISN 0043          DO 140 I=1,NP          00064
ISN 0044    140   WGHT(I)=2.-WGHT(I)/SMAX 00065
ISN 0045          IF (NP.LE.NPMAX) RETURN 00066
C -----00067
C          REDUCE THE NUMBER OF DESIGNS TO NPMAX. 00068
C -----00069
ISN 0047          NPMX1=NPMAX+1          00070
ISN 0048          NPSAV=NP               00071
ISN 0049          DO 200 II=NPX1,NPSAV    00072
C          FIND DESIGN WITH MINIMUM WEIGHTING FACTOR. 00073
ISN 0050          WMIN=WGHT(1)           00074
ISN 0051          IMN=1                  00075
ISN 0052          DO 150 I=2,NP          00076
ISN 0053          IF (WGHT(I).GE.WMIN) GO TO 150 00077
ISN 0054          WMIN=WGHT(I)           00078
ISN 0055          IMN=I                  00079
ISN 0057    150   CONTINUE               00080
ISN 0058          IF (IMN.EQ.NP) GO TO 190 00081
C          SHIFT XI, Y AND WGHT.         00082
ISN 0060          NPM1=NP-1              00083
ISN 0061          DO 180 J=IMN,NPM1      00084
ISN 0062          DO 160 I=1,NXAPRX      00085
ISN 0063    160   XI(I,J)=XI(I,J+1)     00086
ISN 0064          DO 170 I=1,NF          00087
ISN 0065    170   Y(I,J)=Y(I,J+1)       00088
ISN 0066    180   WGHT(I)=WGHT(I+1)     00089
ISN 0067    190   NP=NP-1                00090
ISN 0068    200   CONTINUE               00091
ISN 0069          RETURN                 00092
ISN 0070          END                     00093

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOCBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 69, PROGRAM SIZE = 2038, SUBPROGRAM NAME =COPE10
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

3000K BYTES OF CORE NOT USED

REQUESTED OPTICNS: SOURCE,NOMAP,NOXREF,NOLIST,NODECK,OPT(3),AUTODBL(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GUSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

298

```

C          DATA SET U477COPE11 AT LEVEL 001 AS OF 02/13/81
C          DATA SET 9188COPE11 AT LEVEL 001 AS OF 07/10/80          00001
ISN 0002   SUBROUTINE COPE11 (NPTOT,Y,NYR,INOM,BLU,NCONA,IGFN,IOBJA,SGNOPT,CT00002
          IMIN,CTLMIN,ISC)          00003
ISN 0003   DIMENSION Y(NYR,1), BLU(4,1), IGFN(1), ISC(1)          00004
          *****00005
C          ROUTINE TO DETERMINE NOMINAL DESIGN FOR APPROXIMATE OPTIMIZATION. 00006
C          *****00007
C          00008
C          BY G. N. VANDERPLAATS          JAN., 1979.          00009
C          NASA AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.          00010
C          NOMINAL DESIGN IS THE ONE WITH LOWEST OBJECTIVE SATISFYING ALL 00011
C          CONSTRAINTS. IF ALL DESIGNS VIOLATE CONSTRAINTS, THE DESIGN WITH 00012
C          THE LEAST VIOLATION IS FOUND.          00013
C          I1 IS DESIGN WITH LOWEST MAXIMUM CONSTRAINT VALUE.          00014
C          I2 IS THE DESIGN WITH THE LOWEST OBJECTIVE SATISFYING ALL 00015
C          CONSTRAINTS.          00016
ISN 0004   CT1=ABS(CTMIN)          00017
ISN 0005   IF (CT1.LT.0.004) CT1=0.004          00018
ISN 0007   CTL1=ABS(CTLMIN)          00019
ISN 0008   IF (CTL1.LT.0.001) CTL1=0.001          00020
          -----00021
C          FIND MAXIMUM OBJECTIVE.          00022
C          -----00023
ISN 0010   OBJMAX=-Y(IOBJA,1)*SGNOPT          00024
ISN 0011   DO 10 J=2,NPTOT          00025
ISN 0012   OBJ=-Y(IOBJA,J)*SGNOPT          00026
ISN 0013   IF (OBJ.GT.OBJMAX) OBJMAX=OBJ          00027
ISN 0015   10 CONTINUE          00028
          -----00029
C          NOW FIND DESIGN VARIABLE WITH LOWEST OBJECTIVE SATISFYING ALL 00030
C          CONSTRAINTS AND DESIGN WITH LEAST CONSTRAINT VIOLATION.          00031
C          -----00032
ISN 0016   GMAX=1.0E+20          00033
ISN 0017   I1=1          00034
ISN 0018   I2=0          00035
ISN 0019   DO 50 J=1,NPTOT          00036
          C          OBJECTIVE.          00037
ISN 0020   OBJ=-Y(IOBJA,J)*SGNOPT          00038
          C          CONSTRAINTS.          00039
ISN 0021   ICON=0          00040
ISN 0022   G1=-1.          00041
ISN 0023   DO 30 I=1,NCONA          00042
ISN 0024   II=IGFN(I)          00043
ISN 0025   GG=Y(II,J)          00044
          C          LOWER BOUND.          00045
ISN 0026   IF (BLU(1,I).LT.-1.0E+15) GO TO 20          00046
ISN 0028   ICON=ICON+1          00047
ISN 0029   CT=CT1          00048
ISN 0030   IF (ISC(ICON).GT.0) CT=CTL1          00049
ISN 0032   G=(BLU(1,I)-GG)/BLU(2,I)-CT          00050
ISN 0033   IF (G.GT.G1) G1=G          00051
          C          UPPER BOUND.          00052
ISN 0035   20 IF (BLU(3,I).GT.1.0E+15) GO TO 30          00053
ISN 0037   ICON=ICON+1          00054

```

```

+VERSION 1.3.0 (01 MAY 80)   COPE11   SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)   DATE 82.141/10.55.11   PAGE 2
  ISN 0038           CT=CT1                               00055
  ISN 0039           IF (ISC(ICON).GT.0) CT=CTL1         00056
  ISN 0041           G=(GS-BLU(3,I))/BLU(4,I)-CT         00057
  ISN 0042           IF (G.GT.G1) G1=G                  00058
  ISN 0044           30  CONTINUE                        00059
  ISN 0045           IF (G1.LT.0..OR.G1.GT.GMAX) GO TO 40 00060
  ISN 0047           I1=J                                00061
  ISN 0048           GMAX=G1                             00062
  ISN 0049           40  IF (OBJ.GT.OBJMAX.OR.G1.GT.0.) GO TO 50 00063
  ISN 0051           I2=J                                00064
  ISN 0052           OBJMAX=OBJ                          00065
  ISN 0053           50  CONTINUE                        00066
  ISN 0054           INOM=I1                             00067
  ISN 0055           IF (I2.GT.0) INOM=I2               00068
  ISN 0057           RETURN                               00069
  ISN 0058           END                                 00070

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOOBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS*   SOURCE STATEMENTS =    57, PROGRAM SIZE =    1134, SUBPROGRAM NAME =COPE11
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

3009K BYTES OF CORE NOT USED

REQUESTED OPTIONS: SOURCE,NOMAP,NOXREF,NOLIST,NODECK,OPT(3),AUTODBL(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```

C          DATA SET U477COPE12 AT LEVEL 001 AS OF 02/13/81
C          DATA SET 9188COPE12 AT LEVEL 001 AS OF 07/10/80          00001
C          SUBROUTINE COPE12          00002
C          *****00003
C          ROUTINE TO PERFORM A LEAST SQUARES FIT OF AN ARBITRARY FUNCTION OF
C          NV VARIABLES.          00005
C          Y = F(X1,X2,...,XNX) = B(1)*F(1) + B(2)*F(2) + ... + B(M)*F(M) 00006
C          *****00007
C          00008
C          BY G. N. VANDERPLAATS          JAN., 1979.          00009
C          NASA AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.          00010
C          ARGUMENTS,          00011
C          X,Y - INPUT ARRAYS OF OBSERVATIONS OF NP POINTS          00012
C          X(NX,NP), Y(NF,NP)          00013
C          NX - NUMBER OF INDEPENDENT VARIABLES OF WHICH Y IS A FUNCTION. 00014
C          NP - NUMBER OF OBSERVATION POINTS.          00015
C          NF - NUMBER OF SEPERATE CURVE FITS BEING DONE SIMULTANEOUSLY. 00016
C          THIS IS THE NUMBER OF SETS OF Y VALUES.          00017
C          M - NUMBER OF COMPONENTS OF THE FUNCTIONS TO BE FITTED.          00018
C          B - ARRAY OF M COEFFICIENTS OF FUNCTIONAL FIT TO DATE.          00019
C          A - M(M+1)/2 WORK VECTOR.          00020
C          F - WORK VECTOR - F(M).          00021
C          G - WORK VECTOR - G(NF).          00022
C          NXR - DIMENSIONED ROWS OF X.          00023
C          NYR - DIMENSIONED ROWS OF Y.          00024
C          NBR - DIMENSIONED ROWS OF B.          00025
C          WGHT - ARRAY OF WEIGHTING FACTORS - WGHT(NP).          00026
C          NER - ERROR FLAG. IF NER.GT.0, DIAGONAL ELEMENT NER OF A IS
C          LESS THAN 1.0E-10.          00027
C          00028
C          USER SUPPLIED SUBROUTINE, COPE13.          00029
C          USAGE          00030
C          CALL COPE13(XI,F,NX,M)          00031
C          ROUTINE TO EVALUATE COMPONENTS F(1),...,F(M) WHICH ARE TO BE
C          FITTED TO DATA.          00032
C          TO DATA. ROUTINE EVALUATES THE FUNCTIONS FOR A SINGLE VECTOR OF
C          XI AND STORES THE RESULTING VALUES IN VECTOR F.          00033
C          ARGUMENTS.          00034
C          XI - VECTOR OF INDEPENDENT VARIABLES AT WHICH FUNCTIONS ARE
C          TO BE EVALUATED.          00035
C          F - VECTOR OF FUNCTION VALUES.          00036
C          NX - NUMBER OF INDEPENDENT VARIABLES OF WHICH Y IS A FUNCTION. 00037
C          M - NUMBER OF FUNCTION COMPONENTS, ALSO REQUIRED DIMENSION OF F 00038
C          00039
C          SUBROUTINE COPE12 (X,Y,NX,NP,NF,M,B,A,F,G,NXR,NYR,NBR,WGHT,NER) 00040
C          DIMENSION X(NXR,1), Y(NYR,1), B(NBR,1), A(1), F(1), G(1) 00041
C          DIMENSION WGHT(1)          00042
C          IF (NX.LE.NP) GO TO 50          00043
C          SPECIAL CASE. FEWER OBSERVATIONS THAN DESIGN VARIABLES. USE 00044
C          AVERAGE FINITE DIFFERENCE FOR FIRST ORDER EXPANSION. 00045
C          NP1=NP+1          00046
C          AP=FLOAT(NP)          00047
C          DO 20 I=1,NX          00048
C          X(I,NP1)=AP          00049
C          DO 10 J=1,NP          00050

```

ISN 0002
ISN 0003
ISN 0004
ISN 0005

ISN 0007
ISN 0008
ISN 0009
ISN 0010
ISN 0011

+VERSION 1.3.0 (01 MAY 80)		COPE12	SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)	DATE 82.141/10.55.15	PAGE 2
ISN 0012		IF (ABS(X(I,J)).GT.1.0E-10) GO TO 10		00055	
ISN 0014		X(I,NP1)=X(I,NP1)-1.		00056	
ISN 0015		X(I,J)=1.0E+20		00057	
ISN 0016	10	CONTINUE		00058	
ISN 0017		IF (X(I,NP1).LT.1.) X(I,NP1)=1.		00059	
ISN 0019	20	CONTINUE		00060	
ISN 0020		DO 40 I=1,NX		00061	
ISN 0021		DO 40 J=1,NF		00062	
ISN 0022		B(I,J)=0.		00063	
ISN 0023		DO 30 K=1,NP		00064	
ISN 0024	30	B(I,J)=B(I,J)+Y(J,K)/X(I,K)		00065	
ISN 0025	40	B(I,J)=B(I,J)/X(I,NP1)		00066	
ISN 0026		NER=0		00067	
ISN 0027		RETURN		00068	
ISN 0028	50	CONTINUE		00069	
	C	GENERAL CASE. DO LEAST SQUARES FIT.		00070	
	C	A=B=0.		00071	
ISN 0029		DO 60 J=1,NF		00072	
ISN 0030		DO 60 I=1,M		00073	
ISN 0031	60	B(I,J)=0.		00074	
ISN 0032		L=(M*(M+1))/2		00075	
ISN 0033		DO 70 J=1,L		00076	
ISN 0034	70	A(J)=0.		00077	
	C	LOWER TRIANGLE OF A IN SYMMETRIC MODE.		00078	
ISN 0035		DO 100 K=1,NP		00079	
ISN 0036		WGHTK=WGHT(K)		00080	
ISN 0037		CALL COPE13 (X(1,K),F,NX,M)		00081	
ISN 0038		L=0		00082	
ISN 0039		DO 80 J=1,M		00083	
ISN 0040		DO 80 I=1,J		00084	
ISN 0041		L=L+1		00085	
ISN 0042	80	A(L)=A(L)+F(I)*F(J)*WGHTK		00086	
	C	Y*F		00087	
ISN 0043		DO 90 L=1,NF		00088	
ISN 0044		YLK=Y(L,K)*HGHTK		00089	
ISN 0045		DO 90 I=1,M		00090	
ISN 0046	90	B(I,L)=B(I,L)+YLK*F(I)		00091	
ISN 0047	100	CONTINUE		00092	
	C	SOLVE FOR B.		00093	
ISN 0048		IF (M.LE.1) GO TO 200		00094	
	C	LDU DECOMPOSITION.		00095	
ISN 0050		MM1=M-1		00096	
ISN 0051		KK=0		00097	
ISN 0052		DO 110 K=1,MM1		00098	
ISN 0053		NER=K		00099	
ISN 0054		KK=KK+K		00100	
ISN 0055		IF (ABS(A(KK)).LT.1.0E-20) GO TO 220		00101	
ISN 0057		FACT=1./A(KK)		00102	
ISN 0058		A(KK)=FACT		00103	
ISN 0059		KP1=K+1		00104	
ISN 0060		KJ=KK		00105	
ISN 0061		DO 110 J=KP1,M		00106	
ISN 0062		KJ=KJ+J-1		00107	
ISN 0063		GG=A(KJ)*FACT		00108	
ISN 0064		KI=KK		00109	
ISN 0065		IJ=KJ		00110	
ISN 0066		DO 110 I=KP1,J		00111	
ISN 0067		IJ=IJ+1		00112	
ISN 0068		KI=KI+I-1		00113	

302

```

ISN 0069      A(IJ)=A(IJ)-A(KI)*GG      00114
ISN 0070      110  CONTINUE                00115
ISN 0071      KK=KK+M                    00116
ISN 0072      NER=M                      00117
ISN 0073      IF (ABS(A(KK)).LT.1.0E-20) GO TO 220 00118
ISN 0075      A(KK)=1./A(KK)              00119
              C  FORWARD SUBSTITUTION     00120
ISN 0076      MPI=M+1                    00121
ISN 0077      KK=0                        00122
ISN 0078      DO 130 K=1,MM1              00123
ISN 0079      KPI=K+1                    00124
ISN 0080      KK=KK+K                     00125
ISN 0081      AKK=A(KK)                   00126
ISN 0082      DO 120 L=1,NF               00127
ISN 0083      120  B(K,L)=B(K,L)*AKK      00128
ISN 0084      KI=KK                       00129
ISN 0085      DO 130 I=KPI,M              00130
ISN 0086      KI=KI+I-1                   00131
ISN 0087      AKI=A(KI)                   00132
ISN 0088      DO 130 J=1,NF               00133
ISN 0089      B(I,J)=B(I,J)-AKI*B(K,J)    00134
ISN 0090      130  CONTINUE                00135
ISN 0091      KK=KK+M                    00136
ISN 0092      AKK=A(KK)                   00137
ISN 0093      DO 140 J=1,NF               00138
ISN 0094      140  B(M,J)=B(M,J)*AKK      00139
              C  BACK SUBSTITUTION.       00140
ISN 0095      DO 190 I=2,M                00141
ISN 0096      J=MPI-I                    00142
ISN 0097      JJ=J*(J+1)/2                00143
ISN 0098      JK=JJ                       00144
ISN 0099      JPI=J+1                     00145
ISN 0100      DO 150 L=1,NF               00146
ISN 0101      150  G(L)=0.                 00147
ISN 0102      DO 170 K=JPI,M              00148
ISN 0103      JK=JK+K-1                    00149
ISN 0104      AJK=A(JK)                   00150
ISN 0105      DO 160 L=1,NF               00151
ISN 0106      160  G(L)=G(L)+AJK*B(K,L)    00152
ISN 0107      170  CONTINUE                00153
ISN 0108      AJJ=A(JJ)                   00154
ISN 0109      DO 180 L=1,NF               00155
ISN 0110      180  B(J,L)=B(J,L)-AJJ*G(L)  00156
ISN 0111      190  CONTINUE                00157
ISN 0112      NER=0                       00158
ISN 0113      RETURN                       00159
ISN 0114      200  CONTINUE                00160
ISN 0115      AKK=A(1)                    00161
ISN 0116      NER=1                       00162
ISN 0117      IF (ABS(AKK).LT.1.0E-20) GO TO 220 00163
ISN 0119      AKK=1./AKK                  00164
ISN 0120      DO 210 J=1,NF               00165
ISN 0121      210  B(1,J)=B(1,J)*AKK      00166
ISN 0122      NER=0                       00167
ISN 0123      RETURN                       00168
ISN 0124      220  CONTINUE                00169
ISN 0125      RETURN                       00170
ISN 0126      END                          00171

```

+VERSION 1.3.0 (01 MAY 80) COPE12 SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 82.141/10.55.15 PAGE 4
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTHT NOXREF NOALC NOANSF TERM IBM FLAG(I)
STATISTICS SOURCE STATEMENTS = 125, PROGRAM SIZE = 2844, SUBPROGRAM NAME =COPE12
STATISTICS NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

2992K BYTES OF CORE NOT USED

304

```

C      DATA SET U477COPE13 AT LEVEL 001 AS OF 02/13/81
C      DATA SET 9188COPE13 AT LEVEL 001 AS OF 07/10/80          00001
ISN 0002  SUBROUTINE COPE13 (X,F,NX,M)                          00002
ISN 0003  DIMENSION X(1), F(1)                                  00003
C      *****00004
C      ROUTINE TO CALCULATE F VALUES FOR LEAST SQUARES FIT TO QUADRATIC 00005
C      Y-Y0 = DY-TRANSPOSE TIMES X + (1/2 X-TRANSPOSE TIMES H TIMES X. 00006
C      *****00007
C      00008
C      BY G. N. VANDERPLAATS          JAN., 1979.              00009
C      X CONTAINS X-X0.          00010
C      M = MAXIMUM NUMBER OF COEFFICIENTS TO BE CALCULATED.    00011
C      M .LE. NX + (NX+1)/2.    00012
C      DY COEF.          00013
ISN 0004  DO 10 I=1,NX          00014
ISN 0005  F(I)=X(I)          10 00015
C      H COEF. = X1*X1, X2*X2, ... XN*XN, X1*X2... X1*XN...    00016
ISN 0006  II=NX          00017
C      -----00018
C      DIAGONAL ELEMENTS.          00019
C      -----00020
ISN 0007  DO 20 I=1,NX          00021
ISN 0008  II=II+1          00022
ISN 0009  IF (II.GT.M) GO TO 40          00023
ISN 0011  20 F(II)=.5*(X(I)**2)          00024
ISN 0012  IF (NX.LT.2) RETURN          00025
C      -----00026
C      OFF-DIAGONAL ELEMENTS.          00027
C      -----00028
ISN 0014  NXM1=NX-1          00029
ISN 0015  DO 30 I=1,NXM1          00030
ISN 0016  IP1=I+1          00031
ISN 0017  DO 30 J=IP1,NX          00032
ISN 0018  II=II+1          00033
ISN 0019  IF (II.GT.M) GO TO 40          00034
ISN 0021  30 F(II)=X(I)*X(J)          00035
ISN 0022  40 CONTINUE          00036
ISN 0023  RETURN          00037
ISN 0024  END          00038
  
```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
 *OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
 STATISTICS SOURCE STATEMENTS = 23, PROGRAM SIZE = 566, SUBPROGRAM NAME =COPE13
 STATISTICS NO DIAGNOSTICS GENERATED
 ***** END OF COMPILATION *****
 3016K BYTES OF CORE NOT USED

```

C          DATA SET U477COPE14 AT LEVEL 002 AS OF 03/16/81
C          DATA SET U477COPE14 AT LEVEL 001 AS OF 02/13/81          00001
C          DATA SET 9188COPE14 AT LEVEL 001 AS OF 07/10/80          00002
ISN 0002   SUBROUTINE COPE14 (NXAPRX,NF,NPTOT,RA,IA,LOCR,LOCI,TITLE,INOM,NDV,00003
           IIPAPRX,ISCR2,MAXTRM)          00004
ISN 0003   COMMON /UIOS/ I5I, I6I          00005
ISN 0004   DIMENSION RA(1), IA(1), LOCR(1), LOCI(1), TITLE(1)          00006
           *****00007
C          ROUTINE TO PRINT RESULTS OF APPROXIMATE ANALYSIS/OPTIMIZATION. 00008
C          *****00009
C          00010
C          BY G. N. VANDERPLAATS          JAN., 1979.          00011
C          NASA AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.          00012
C          00013
C          -----00014
C          TITLE.          00015
C          -----00016
ISN 0005   WRITE (I6I,90) (TITLE(I),I=1,20)          00017
           -----00018
C          GLOBAL LOCATION OF X AND F(X)          00019
C          -----00020
C          GLOBAL LOCATIONS OF X.          00021
ISN 0006   M5=LOCI(5)          00022
ISN 0007   MM5=M5+NXAPRX-1          00023
ISN 0008   WRITE (I6I,100)          00024
ISN 0009   WRITE (I6I,110) (IA(I),I=M5,MM5)          00025
           C          GLOBAL LOCATIONS OF F(X).          00026
ISN 0010   M6=LOCI(6)          00027
ISN 0011   MM6=M6+NF-1          00028
ISN 0012   WRITE (I6I,140)          00029
ISN 0013   WRITE (I6I,110) (IA(I),I=M6,MM6)          00030
           C          -----00031
C          X-VALUES AND FUNCTIONS, F(X)          00032
C          -----00033
C          X-VALUES.          00034
ISN 0014   N1=LOCR(23)+3*NXAPRX+6          00035
ISN 0015   N2=N1+NXAPRX-1          00036
ISN 0016   WRITE (I6I,120) NPTOT,INOM          00037
ISN 0017   WRITE (I6I,130) (RA(I),I=N1,N2)          00038
           C          F(X) VALUES.          00039
ISN 0018   N1=N1+NXAPRX+NDV          00040
ISN 0019   N2=N1+NF-1          00041
ISN 0020   WRITE (I6I,150)          00042
ISN 0021   WRITE (I6I,130) (RA(I),I=N1,N2)          00043
           C          -----00044
C          TAYLER SERIES COEFFICIENTS.          00045
C          -----00046
ISN 0022   WRITE (I6I,170)          00047
ISN 0023   NP=NPTOT-1          00048
ISN 0024   NBTAY=N1+2*NF          00049
ISN 0025   NBR=NXAPRX+(NXAPRX*(NXAPRX+1))/2          00050
ISN 0026   IF (MAXTRM.LT.3) NBR=MAXTRM*NXAPRX          00051
ISN 0028   DO 30 JJ=1,NF          00052
ISN 0029   M6=LOCI(6)+JJ-1          00053
ISN 0030   M6=IA(M6)          00054
  
```

306

```

ISN 0031      WRITE (I6I,80) JJ,M6          00055
              C      LINEAR TERMS.          00056
ISN 0032      N2=NXAPRX                    00057
ISN 0033      IF (NP.LT.N2) N2=NP          00058
ISN 0035      N2=N2+NBTAY-1                00059
ISN 0036      WRITE (I6I,160)              00060
ISN 0037      WRITE (I6I,130) (RA(I),I=NBTAY,N2) 00061
ISN 0038      IF (NP.LE.NXAPRX) GO TO 20   00062
ISN 0040      IF (MAXTRM.LT.2) GO TO 20'    00063
              C      NON-LINEAR TERMS.     00054
              C      N1 = LOCATION OF FIRST DIAGONAL ELEMENT. 00065
ISN 0042      N1=N2+1                       00066
              C      N2 = LOCATION OF LAST DIAGONAL ELEMENT. 00067
ISN 0043      N2=NXAPRX                     00068
ISN 0044      IF (N2.GT.NP) N2=NP          00069
ISN 0046      N2=N2+N1-1                   00070
              C      N3=LOCATION OF FIRST OFF-DIAGONAL ELEMENT. 00071
ISN 0047      N3=N2+1                       00072
              C      N4=LOCATION OF LAST OFF-DIAGONAL ELEMENT. 00073
ISN 0048      N4=NBR                       00074
ISN 0049      IF (N4.GT.NP) N4=NP          00075
ISN 0051      N4=N4-2*NXAPRX+N3-1          00076
              C      LL = LOCATION OF LAST OFF-DIAGONAL ELEMENT - THIS ROW. 00077
ISN 0052      WRITE (I6I,190)              00078
ISN 0053      II=1                          00079
ISN 0054      DO 10 I=N1,N2                00080
ISN 0055      WRITE (I6I,180) II           00081
ISN 0056      II=II+1                      00082
ISN 0057      LL=N3+NXAPRX-II              00083
ISN 0058      IF (LL.GT.N4) LL=N4          00084
ISN 0060      IF (LL.LT.N3) WRITE (I6I,130) RA(I) 00085
ISN 0062      IF (LL.GE.N3) WRITE (I6I,130) RA(I),(RA(J),J=N3,LL) 00086
ISN 0064      10  N3=LL+1                   00087
ISN 0065      20  CONTINUE                  00088
ISN 0066      NBTAY=NBTAY+NBR              00089
ISN 0067      30  CONTINUE                  00090
ISN 0068      IF (IPAPRX.LT.3) RETURN      00091
ISN 0070      REWIND ISCR2                  00092
ISN 0071      WRITE (I6I,50)                00093
ISN 0072      DO 40 I=1,NPTOT              00094
              C      X-VECTOR.              00095
ISN 0073      READ (ISCR2) (RA(J),J=1,NXAPRX) 00096
ISN 0074      WRITE (I6I,60) I              00097
ISN 0075      WRITE (I6I,130) (RA(J),J=1,NXAPRX) 00098
              C      FUNCTION VALUES.      00099
ISN 0076      READ (ISCR2) (RA(J),J=1,NF)   00100
ISN 0077      WRITE (I6I,70)                00101
ISN 0078      WRITE (I6I,130) (RA(J),J=1,NF) 00102
ISN 0079      40  CONTINUE                  00103
ISN 0080      RETURN                        00104
              C      -----              00105
              C      FORMATS                00106
              C      -----              00107
ISN 0081      50  FORMAT (///5X,18HSUMMARY OF DESIGNS) 00108
ISN 0082      60  FORMAT (/5X,13HDESIGN NUMBER,I5/5X,8HX-VECTOR) 00109
ISN 0083      70  FORMAT (/5X,15HFUNCTION VALUES)    00110
ISN 0084      80  FORMAT (//5X,9HPARAMETER,I5,18H = GLOBAL VARIABLE,I5) 00111
ISN 0085      90  FGRMAT (I1I,4X,44HRESULTS OF APPROXIMATE ANALYSIS/OPTIMIZATION///500112
                1X,5HTITLE/5X,20A4)          00113
    
```

```

+VERSION 1.3.0 (01 MAY 80)   COPE14   SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)   DATE 82.141/10.55.24   PAGE 3
ISN 0086   100   FORMAT (///5X,31HGLOBAL LOCATIONS OF X-VARIABLES)   00114
ISN 0087   110   FORMAT (5X,10I5)   00115
ISN 0088   120   FORMAT (//5X,25HAPPROXIMATION IS BASED ON,I5,8H DESIGNS//5X,31HNOMINAL DESIGN IS DESIGN NUMBER,I5//5X,21HVALUES OF X-VARIABLES)   00116
ISN 0089   130   FORMAT (5X,5E13.4)   00118
ISN 0090   140   FORMAT (//5X,35HGLOBAL LOCATIONS OF FUNCTIONS, F(X))   00119
ISN 0091   150   FORMAT (//5X,25HVALUES OF FUNCTIONS, F(X))   00120
ISN 0092   160   FORMAT (/5X,19HLINEAR TERMS, DEL F)   00121
ISN 0093   170   FORMAT (///5X,39HCOEFFICIENTS OF TAYLOR SERIES EXPANSION)   00122
ISN 0094   180   FORMAT (/5X,3HROW,I5)   00123
ISN 0095   190   FORMAT (//5X,51HNON-LINEAR TERMS, H, BEGINING WITH DIAGONAL ELEMENTS)   00124
ISN 0096   END   00126
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM ISM FLAG(I)
*STATISTICS*   SOURCE STATEMENTS =   95, PROGRAM SIZE =   2632, SUBPROGRAM NAME =COPE14
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****
2996K BYTES OF CORE NOT USED

```

REQUESTED OPTIONS: SOURCE,NOMAP,NOXREF,NOLIST,NODECK,OPT(3),AUTODBL(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

308

```

C          DATA SET U477COPE15 AT LEVEL 001 AS OF 02/13/81
C          DATA SET 9188COPE15 AT LEVEL 001 AS OF 07/10/80          00001
ISN 0002   SUBROUTINE COPE15 (XV,G,DF,A,ISC,IC,NN1,BLU,NX1,IOBJA,M,FNOM,FNEW,00002
          IBTAY,NBR,IGFN,CT,CTL,INFO,NAC,NCONA,NDV,NF,OBJ,SGNOPT)      00003
ISN 0003   DIMENSION XV(1), FNOM(1), FNEW(1), A(NN1,1), BTAY(NBR,1), DF(1), I00004
          IGFN(1), ISC(1), IC(1), G(1), BLU(4,1)                      00005
          *****00006
C          FUNCTION EVALUATION FOR APPROXIMATE OPTIMIZATION.        00007
          *****00008
C          BY G. N. VANDERPLAATS          JAN., 1979.00009
C          NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.        00010
C          00011
C          -----00012
C          OBJECTIVE.                                               00013
C          -----00014
ISN 0004   CALL COPE16 (NX1,XV,NF,FNOM,FNEW,BTAY,NBR,M)              00015
ISN 0005   OBJ=-FNEW(IOBJA)*SGNOPT                                    00016
ISN 0006   IF (INFO.EQ.1) GO TO 20                                    00017
C          GRADIENT OF OBJECTIVE.                                     00018
ISN 0008   CALL COPE17 (NX1,XV,IOBJA,BTAY,NBR,M,DF)                  00019
ISN 0009   DO 10 I=1,NDV                                             00020
ISN 0010   10 DF(I)=-DF(I)*SGNOPT                                     00021
ISN 0011   20 CONTINUE                                              00022
ISN 0012   IF (NCONA.LE.0) GO TO 80                                   00023
          -----00024
C          CONSTRAINTS.                                             00025
C          -----00026
ISN 0014   IF (INFO.EQ.2) NAC=0                                       00027
ISN 0016   ICON=0                                                    00028
ISN 0017   DO 70 I=1,NCONA                                           00029
ISN 0018   J=IGFN(I)                                                 00030
ISN 0019   GG=FNEW(J)                                                 00031
C          LOWER BOUND.                                             00032
ISN 0020   IF (BLU(1,I).LT.-1.0E+15) GO TO 40                        00033
ISN 0022   ICON=ICON+1                                               00034
ISN 0023   G(ICON)=(BLU(1,I)-GG)/BLU(2,I)                            00035
ISN 0024   IF (INFO.EQ.1) GO TO 40                                    00036
C          IS THIS CONSTRAINT ACTIVE OR VIOLATED.                  00037
ISN 0026   CTI=CT                                                    00038
ISN 0027   IF (ISC(ICON).GT.0) CTI=CTL                                00039
ISN 0029   IF (G(ICON).LT.CTI) GO TO 40                              00040
C          ACTIVE CONSTRAINT. CALCULATE GRADIENT.                  00041
ISN 0031   NAC=NAC+1                                                 00042
ISN 0032   IC(NAC)=ICON                                              00043
ISN 0033   MM=M                                                       00044
ISN 0034   IF (ISC(ICON).GT.0) MM=NDV                                00045
ISN 0036   CALL COPE17 (NX1,XV,J,BTAY,NBR,MM,A(1,NAC))              00046
ISN 0037   FF=1./BLU(2,I)                                           00047
ISN 0038   DO 30 K=1,NDV                                             00048
ISN 0039   30 A(K,NAC)=-A(K,NAC)*FF                                  00049
ISN 0040   40 IF (BLU(3,I).GT.1.0E+15) GO TO 60                     00050
ISN 0042   ICON=ICON+1                                               00051
ISN 0043   G(ICON)=(GG-BLU(3,I))/BLU(4,I)                            00052
ISN 0044   IF (INFO.EQ.1) GO TO 60                                    00053
C          IS THIS CONSTRAINT ACTIVE OR VIOLATED.                  00054

```


REQUESTED OPTIONS: SOURCE,NOMAP,NOXREF,NOLIST,NODECK,OPT(3),AUTODBL(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTHT NOXREF NOALC NOANSF TERM IBM FLAG(I)

310

```

C          DATA SET U477COPE16 AT LEVEL 001 AS OF 02/13/81
C          DATA SET 9188COPE16 AT LEVEL 001 AS OF 07/10/80          00001
ISN 0002  SUBROUTINE COPE16 (NX,X,NF,FNOM,FNEW,B,NBR,M)              00002
C          *****00003
C          ROUTINE TO EVALUATE FUNCTIONS APPROXIMATED BY TAYLER SERIES 00004
C          EXPANSION UP TO SECOND ORDER.                             00005
C          *****00006
C          BY G. N. VANDERPLAATS          MAR., 1978.              00007
C          NAVAL SHIP R AND D CENTER.                                00008
C          00009
C          F = F0 + DELF TIMES X + X-TRANPOSE TIMES DEL2F TIMES X. 00010
C          ARGUMENTS.                                              00011
C          NX - NUMBER OF INDEPENDENT VARIABLES CONTAINED IN X.   00012
C          X  - VECTOR OF DELTA VARIABLES X-XNOM. DIMENSIONED X(NX) 00013
C          NF - NUMBER OF FUNCTIONS TO BE EVALUATED.              00014
C          FNOM - NOMINAL FUNCTION VALUES ABOUT WHICH TAYLER SERIES EXPANSION 00015
C          WAS DONE.                                               00016
C          FNEW - NEW APPROXIMATED VALUES. - OUTPUT. DIMENSIONED FNEW(NF) 00017
C          B   - MATRIX OF TAYLER SERIES COEFFICIENTS.              00018
C          B(I,J) CONTAINS DEL F, I=1,NX.                          00019
C          B(NX+I,J) CONTAINS DEL2 TERMS, I = 1,NX*(NX+1)/2.      00020
C          MINIMUM DIMENSIONS - B(M,NF).                          00021
C          NBR - DIMENSIONED ROWS OF B.                            00022
C          M   - TOTAL NUMBER OF COEFFICIENTS CURRENTLY USED.     00023
C          00024
ISN 0003  DIMENSION X(1), FNOM(1), FNEW(1), B(NBR,1)              00025
ISN 0004  DO 50 J=1,NF                                             00026
C          -----00027
C          CONSTANT TERM.                                         00028
C          -----00029
ISN 0005  F=FNOM(J)                                              00030
C          -----00031
C          FIRST ORDER TERMS.                                     00032
C          -----00033
ISN 0006  DO 10 I=1,NX                                           00034
ISN 0007  IF (I.GT.M) GO TO 40                                    00035
ISN 0009  F=F+B(I,J)*X(I)                                        00036
C          -----00037
C          SECOND ORDER TERMS.                                    00038
C          -----00039
C          00040
C          -----00041
C          DIAGONAL ELEMENTS.                                    00042
C          -----00043
ISN 0010  II=NX                                                  00044
ISN 0011  DO 20 I=1,NX                                           00045
ISN 0012  II=II+1                                               00046
ISN 0013  IF (II.GT.M) GO TO 40                                    00047
ISN 0015  F=F+.5*B(II,J)*(X(I)**2)                               00048
C          -----00049
C          OFF-DIAGONAL ELEMENTS.                                 00050
C          -----00051
ISN 0016  IF (NX.LT.2) GO TO 40                                    00052
ISN 0018  NXM1=NX-1                                             00053
ISN 0019  DO 30 I=1,NXM1                                         00054

```

```

+VERSION 1.3.0 (01 MAY 80)   COPE16   SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)   DATE 82.141/10.55.31   PAGE 2
  ISN 0020           IP1=I+1           00055
  ISN 0021           XX=X(I)           00056
  ISN 0022           DO 30 K=IP1,NX     00057
  ISN 0023           II=II+1           00058
  ISN 0024           IF (II.GT.M) GO TO 40 00059
  ISN 0026    30     F=F+B(II,J)*XX*X(K) 00060
  ISN 0027    40     CONTINUE             00061
  ISN 0028    50     FNEH(J)=F           00062
  ISN 0029           RETURN             00063
  ISN 0030           END                 00064

```

```

*OPTIONS IN EFFECT*NAME(HAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS*   SOURCE STATEMENTS =      29, PROGRAM SIZE =      878, SUBPROGRAM NAME =COPE16
*STATISTICS*   NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

3016K BYTES OF CORE NOT USED

+VERSION 1.3.0 (01 MAY 80) COPE17 SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 82.141/10.55.33 PAGE 2
ISH 0027 END 00055
*OPTICNS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECCUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
STATISTICS SOURCE STATEMENTS = 26, PROGRAM SIZE = 680, SUBPROGRAM NAME =COPE17
STATISTICS NO DIAGNOSTICS GENERATED
***** END OF COMPILATION ***** 3016K BYTES OF CORE NOT USED

314

```

C      DATA SET U477COPE18 AT LEVEL 002 AS OF 03/16/81
C      DATA SET U477COPE18 AT LEVEL 001 AS OF 02/13/81          00001
C      DATA SET 9183COPE18 AT LEVEL 001 AS OF 07/10/80          00002
ISN 0002  C      SUBROUTINE COPE18 (IOBJ,NDVTOT,NCONA,RA,IA,LOCR,LOCI,ARRAY) 00003
C      *****00004
C      ROUTINE TO PRINT OPTIMIZATION RESULTS                      00005
C      *****00006
C      BY G. N. VANDERPLAATS MAR., 1979 00007
C      NASA AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.          00008
ISN 0003  C      COMMON /UIOS/ I5I, I6I 00009
ISN 0004  C      DIMENSION RA(1), IA(1), LOCR(1), LOCI(1), ARRAY(1) 00010
C      OBJECTIVE FUNCTION AND DESIGN VARIABLES.                  00011
ISN 0005  C      WRITE (I6I,30) IOBJ,ARRAY(IOBJ) 00012
ISN 0006  C      N2=LOCR(2) 00013
ISN 0007  C      N3=LOCR(3) 00014
ISN 0008  C      N5=LOCR(5) 00015
ISN 0009  C      M2=LOCI(2) 00016
ISN 0010  C      DO 10 I=1,NDVTOT 00017
C      DESIGN VARIABLE NUMBER. 00018
ISN 0011  C      IDV=IA(M2) 00019
ISN 0012  C      N2=LOCR(2)+IDV-1 00020
ISN 0013  C      N3=LOCR(3)+IDV-1 00021
ISN 0014  C      M2=M2+1 00022
C      GLOBAL LOCATION. 00023
ISN 0015  C      IG=IA(I) 00024
C      MULTIPLIER. 00025
ISN 0016  C      AMULT=RA(N5) 00026
ISN 0017  C      N5=N5+1 00027
C      LOWER BOUND. 00028
ISN 0018  C      BL=AMULT*RA(N2) 00029
C      VALUE. 00030
ISN 0019  C      XX=ARRAY(IG) 00031
C      UPPER BOUND. 00032
ISN 0020  C      BU=AMULT*RA(N3) 00033
ISN 0021  C      WRITE (I6I,40) I,IDV,IG,BL,XX,BU 00034
ISN 0022  10 CONTINUE 00035
ISN 0023  C      IF (NCONA.EQ.0) RETURN 00036
C      CONSTRAINTS. 00037
ISN 0025  C      WRITE (I6I,50) 00038
ISN 0026  C      M3=LOCI(3) 00039
ISN 0027  C      N6=LOCR(6) 00040
ISN 0028  C      ID=1 00041
ISN 0029  C      IF (RA(N6).LT.-1.0E+15.AND.RA(N6+2).GT.1.0E+15) ID=0 00042
ISN 0031  C      DO 20 I=1,NCONA 00043
C      GLOBAL LOCATION. 00044
ISN 0032  C      IG=IA(M3) 00045
ISN 0033  C      M3=M3+1 00046
C      LOWER BOUND. 00047
ISN 0034  C      BL=RA(N6) 00048
C      VALUE. 00049
ISN 0035  C      XX=ARRAY(IG) 00050
C      UPPER BOUND. 00051
ISN 0036  C      BU=RA(N6+2) 00052
ISN 0037  C      N6=N6+4 00053
C      IDENTIFICATION NUMBER. 00054
  
```

```

ISN 0038      JD=ID                                00055
ISN 0039      IF (BL.GT.-1.0E+15) ID=ID+1          00056
ISN 0041      IF (BU.LT.1.0E+15) ID=ID+1          00057
ISN 0043      WRITE (I6I,60) JD,IG,BL,XX,BU        00053
ISN 0044      20  CONTINUE                          00059
ISN 0045      RETURN                                00060
C -----
C                                     FORMATS          00062
C -----
ISN 0046      30  FORMAT (1H1,4X,20HOPTIMIZATION RESULTS///5X,18HOBJECTIVE FUNCTION/00064
                15X,15HGLOBAL LOCATION,15,5X,14HFUNCTION VALUE,E12.5///5X,16HDESIGN00065
                2 VARIABLES//14X,5HD. V.,5X,6HGLCBAL,7X,5HLOWER,23X,5HUPPER/8X,2HID00066
                3,5X,3HNO.,5X,8HVAR. NO.,6X,5HBOUND,9X,5HVALUE,9X,5HBOUND) 00067
ISN 0047      40  FORMAT (I10,I7,I11,3X,3E14.5)    00068
ISN 0048      50  FORMAT (///5X,18HDESIGN CONSTRAINTS//15X,6HGLOBAL,7X,5HLOWER,23X,500069
                1HUPPER/9X,2HID,4X,8HVAR. NO.,6X,5HBOUND,9X,5HVALUE,9X,5HBOUND) 00070
ISN 0049      60  FORMAT (I10,I9,3X,3E14.5)        00071
ISN 0050      END                                  00072

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 49, PROGRAM SIZE = 1408, SUBPROGRAM NAME =COPE18
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

3012K BYTES OF CORE NOT USED

316

```

C      DATA SET U477SIMCON AT LEVEL 001 AS OF 02/13/81
C      DATA SET 9188SIMCON AT LEVEL 001 AS OF 07/10/80      00001
ISN 0002  SUBROUTINE SIMCON (NDV,X,VLB,VUB,NCON,ITMAX,IPRINT,ISCAL,DELFUN,DA00002
          IBFUN,FUN,WK,HNK,INK,NINK,OBJ,G,IER)                00003
          -----00004
C      SIMCON                                                00005
C      ROUTINE FOR SIMPLIFIED COMMIN USAGE                  00006
          -----00007
C      00008
C      BY G. N. VANDERPLAATS                                APRIL, 1979 00009
C      NASA AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.    00010
C      00011
C      --- PURPOSE                                          00012
C      00013
C      MINIMIZE OBJ AS A FUNCTION OF X(I),I=1,NDV          00014
C      SUBJECT TO                                          00015
C      G(J).LE.0, J=1,NCON                                 00016
C      VLB(I).LE.X(I).LE.VUB(I), I = 1,NDV                00017
C      ... NOTES                                           00018
C      NCON MAY BE ZERO.                                    00019
C      VLB(I) IS IGNORED IF VLB(I).LT.-1.0E+15.           00020
C      VUB(I) IS IGNORED IF VUB(I).GT.1.0E+15.           00021
C      IF NCON = 0 AND IT IS NOT ESSENTIAL TO LIMIT THE VARIABLES, X(I) 00022
C      THEN SET VLB(I)=-1.0E+16 AND VUB(I)=1.0E+16. THIS WILL IMPROVE 00023
C      THE EFFICIENCY OF THE UNCONSTRAINED OPTIMIZATION.  00024
C      00025
C      --- REFERENCE                                        00026
C      G. N. VANDERPLAATS                                    00027
C      COMMIN - A FORTRAN PROGRAM FOR CONSTRAINED FUNCTION MINIMIZATION 00028
C      USERS MANUAL, NASA TM X-62,282, AUG. 1973.         00029
C      00030
C      *** ARGUMENTS                                       00031
C      00032
C      --- INPUT                                           00033
C      NDV - NUMBER OF INDEPENDENT DESIGN VARIABLES.      00034
C      X   - ARRAY OF INITIAL DESIGN VARIABLES.           00035
C      REQUIRED DIMENSION = NDV + 2.                        00036
C      VLB - ARRAY OF LOWER BOUNDS ON X.                   00037
C      IF VLB(I).LT.-1.0E+15 THE ITH VARIABLE IS NOT BOUNDED. 00038
C      REQUIRED DIMENSION = NDV +2.                         00039
C      VUB - ARRAY OF UPPER BOUNDS ON X.                   00040
C      IF VUB(I).GT.1.0E+15 THE ITH VARIABLE IS NOT BOUNDED. 00041
C      REQUIRED DIMENSION = NDV +2.                         00042
C      NCON - NUMBER OF CONSTRAINT VALUES STORED IN ARRAY G. 00043
C      NCON MAY BE ZERO.                                    00044
C      ITMAX - MAXIMUM PERMISSABLE ITERATIONS IN COMMIN.  00045
C      DEFAULT = 20.                                        00046
C      IPRINT - COMMIN PRINT CONTROL. IPRINT = 0, NO PRINT. 00047
C      IPRINT = 5, MAXIMUM PRINT.                          00048
C      SUGGESTED VALUE IS IPRINT = 3.                     00049
C      ISCAL - SCALING PARAMETER. IF ISCAL = 0 NO SCALING IS DONE. 00050
C      IF ISCAL = 1 THE DESIGN VARIABLES ARE SCALED DURING 00051
C      OPTIMIZATION. RECOMENDED ISCAL = 1.                00052
C      DELFUN - CONVERGENCE TOLORANCE ON FRACTIONAL CHANGE IN OBJECTIVE 00053
C      FUNCTION. IF OBJ DOES NOT CHANGE BY MORE THAN DELFUN 00054
  
```

```

C      FOR THREE CONSECUTIVE ITERATIONS, OPTIMIZATION IS      00055
C      TERMINATED.                                           00056
C      IF DELFUN = 0 IS INPUT, DELFUN = 0.0001 IS USED.      00057
C      DABFUN - CONVERGENCE TOLORANCE CN ABSOLUTE CHANGE IN OBJECTIVE 00058
C      FUNCTION. IF OBJ DOES NOT CHANGE BY MORE THAN DABFUN 00059
C      IN MAGNITUDE FOR THREE CONSECUTIVE ITERATIONS,      00060
C      OPTIMIZATION IS TERMINATED.                            00061
C      IF DABFUN = 0 IS INPUT, DABFUN = 0.001*ABS(INITIAL OBJ) 00062
C      IS USED.                                               00063
C      FUN - NAME OF EXTERNAL SUBROUTINE WHICH EVALUATES OBJECTIVE 00064
C      AND CONSTRAINT FUNCTIONS.                               00065
C      WK - REAL WORK ARRAY.                                  00066
C      NWK - DIMENSION OF WK ARRAY.                            00067
C      MINIMUM PERMISSABLE DIMENSION.                         00068
C      NWK = 16*NDV + 2*(NDV**2) + 2*NCON + 16                00069
C      SPECIAL CASE - NCON = 0, VLB(I).LT.-1.0E+15 AND        00070
C      AND VUB(I).GT.1.0E+15 FOR I = 1,NDV.                  00071
C      NWK = 5*NDV + 11                                        00072
C      IWK - INTEGER WORK ARRAY.                               00073
C      NIWK - DIMENSION OF IWK ARRAY.                          00074
C      MINIMUM PERMISSABLE DIMENSION.                         00075
C      NIWK = 5*NDV + NCON + 6                                 00076
C      SPECIAL CASE - NCON = 0, VLB(I).LT.-1.0E+15 AND        00077
C      AND VUB(I).GT.1.0E+15 FOR I = 1,NDV.                  00078
C      NIWK = 3                                                00079
C --- OUTPUT                                                  00080
C      X - ARRAY OF DESIGN VARIABLES DEFINING THE OPTIMUM.    00081
C      OBJ - OPTIMUM OBJECTIVE FUNCTION VALUE.                00082
C      G - ARRAY OF CONSTRAINT VALUES AT THE OPTIMUM.        00083
C      REQUIRED DIMENSION = NCON + 2*NDV.                      00084
C      SPECIAL CASE - NCON = 0, VLB(I).LT.-1.0E+15 AND        00085
C      AND VUB(I).GT.1.0E+15 FOR I = 1,NDV.                  00086
C      DIMENSION = 1.                                         00087
C      IER - ERROR CODE.                                       00088
C      IER = 0, NO STORAGE ERROR.                              00089
C      IER = 1, ARRAY WK OR IWK IS NOT DIMENSIONED LARGE     00090
C      ENOUGH.                                                 00091
C --- EXTERNAL                                                00092
C      USER-SUPPLIED ROUTINE TO EVALUATE OBJECTIVE AND CONSTRAINT 00093
C      FUNCTIONS.                                              00094
C      SUBROUTINE FUN (NDV,X,NCON,OBJ,G)                       00095
C ... INPUT                                                    00096
C      NDV - NUMBER OF INDEPENDENT DESIGN VARIABLES.          00097
C      X - ARRAY CONTAINING CURRENT VALUES OF THE NDV DESIGN 00098
C      VARIABLES.                                              00099
C      DIMENSION = NDV + 2.                                    00100
C      NCON - NUMBER OF CONSTRAINT VALUES STORED IN G.        00101
C      NCON MAY BE ZERO.                                       00102
C ... OUTPUT                                                  00103
C      OBJ - OBJECTIVE FUNCTION ASSOCIATED WITH X.            00104
C      G - ARRAY CONTAINING THE NCON CONSTRAINT VALUES ASSOCIATED 00105
C      WITH X. IF NCON = 0, G IS NOT CALCULATED.              00106
C      DIMENSION = NCON + 2*NDV IF NCON.GT.0.                 00107
C      DIMENSION = 1 IF NCON = 0.                              00108
C      EXTERNAL FUN                                           00109
C      DIMENSION X(1), VLB(1), VUB(1), G(1), WK(1), IWK(1)    00111
C      COMMON /CNMNI/ DLFUN,DBFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,ALPHAX,A00112
C      IBOBJ1,THETA,OBJ1,NOV1,NCON1,NSIDE,IPRNT1,NFDG,NSCAL,LINOBJ,ITHAX1,00113

```

ISN 0003
 ISN 0004
 ISN 0005


```

+VERSION 1.3.0 (01 MAY 80) SIMCON SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 82.141/10.55.40 PAGE 4
  ISN 0057 NREQI=N2+N3+N5 00173
  ISN 0058 NREQR1=NREQR 00174
  ISN 0059 NREQI1=NREQI 00175
  ISN 0060 IF (NREQR.GT.NWK.OR.NREQI.GT.NIWK) GO TO 130 00176
  ISN 0062 IF (NCON.EQ.0.AND.NSIDE.EQ.0) GO TO 80 00177
  C FIND MAXIMUM POSSIBLE N3. 00178
  ISN 0064 II=N3 00179
  ISN 0065 N3SAV=N3 00180
  ISN 0066 N21=N2+1 00181
  ISN 0067 DO 60 I=II,N21 00182
  ISN 0068 N3=I 00183
  ISN 0069 N4=N3 00184
  ISN 0070 IF (N4.LT.NDV) N4=NDV 00185
  ISN 0072 N5=2*N4 00186
  ISN 0073 NREQR=3*N1+2*N2+N3*(N1+N3)+N4 00187
  ISN 0074 NREQI=N2+N3+N5 00188
  ISN 0075 IF (NREQR.GT.NWK.OR.NREQI.GT.NIWK) GO TO 70 00189
  ISN 0077 N3SAV=I 00190
  ISN 0078 60 CONTINUE 00191
  ISN 0079 70 N3=N3SAV 00192
  ISN 0080 N4=N3 00193
  ISN 0081 IF (N4.LT.NDV) N4=NDV 00194
  ISN 0083 N5=2*N4 00195
  ISN 0084 80 CONTINUE 00196
  C ----- 00197
  C STORAGE ALLOCATION 00198
  C ----- 00199
  ISN 0085 NNSCAL=1 00200
  ISN 0086 NDF=NNSCAL+N1 00201
  ISN 0087 NA=NDF+N1 00202
  ISN 0088 NS=NA+N1*N3 00203
  ISN 0089 NG1=NS+N1 00204
  ISN 0090 NG2=NG1+N2 00205
  ISN 0091 NB=NG2+N2 00206
  ISN 0092 NC=NB+N3*N3 00207
  ISN 0093 NISC=1 00208
  ISN 0094 NIC=N2+1 00209
  ISN 0095 NMS1=NIC+N3 00210
  C REQUIRED STORAGE. 00211
  ISN 0096 NREQR=NC+N4-1 00212
  ISN 0097 NREQI=NMS1+N5-1 00213
  ISN 0098 IF (NREQR.GT.NWK.OR.NREQI.GT.NIWK) GO TO 130 00214
  ISN 0100 IF (NCON.LE.0) GO TO 100 00215
  C DEFINE ISC ARRAY SO ALL CONSTRAINTS ARE NONLINEAR. 00216
  ISN 0102 N=NISC 00217
  ISN 0103 DO 90 I=1,NCON 00218
  ISN 0104 IWK(N)=0 00219
  ISN 0105 90 N=N+1 00220
  ISN 0106 100 CONTINUE 00221
  ISN 0107 IF (IPRINT.LT.1) GO TO 110 00222
  ISN 0109 WRITE (6,210) 00223
  ISN 0110 WRITE (6,230) NWK,NREQR1,NREQR,NIWK,NREQI1,NREQI 00224
  ISN 0111 110 CONTINUE 00225
  C ----- 00226
  C OPTIMIZATION 00227
  C ----- 00228
  ISN 0112 IGOTO=0 00229
  ISN 0113 120 CONTINUE 00230
  ISN 0114 CALL COMMIN (X,VLB,VUB,G,WK(NNSCAL),WK(NDF),WK(NA),WK(NS),WK(NG1),00231

```

320

```

1WK(NG2),WK(NB),WK(NC),IWK(NISC),IWK(NIC),IWK(NMS1),N1,N2,N3,N4,N5)00232
C HAS STORAGE BEEN EXCEEDED. 00233
ISN 0115 IF (NAC.GE.N3) GO TO 140 00234
C EVALUATE OBJECTIVE AND CONSTRAINTS. 00235
ISN 0117 CALL FUN (NDV,X,NCON,OBJ,G) 00236
ISN 0118 OBJ1=OBJ 00237
ISN 0119 IF (IGOTO.GT.0) GO TO 120 00238
C OPTIMIZATION COMPLETE. 00239
ISN 0121 IER=0 00240
ISN 0122 RETURN 00241
C ----- 00242
C REQUIRED STORAGE EXCEEDS AVAILABLE STORAGE 00243
C ----- 00244
ISN 0123 130 CONTINUE 00245
C INSUFFICIENT STORAGE TO START OPTIMIZATION. 00246
ISN 0124 WRITE (6,220) 00247
ISN 0125 WRITE (6,230) NWK,NREQR,NREQR,NIWK,NREQI,NREQI 00248
ISN 0126 WRITE (6,240) 00249
ISN 0127 RETURN 00250
ISN 0128 140 CONTINUE 00251
C INSUFFICIENT STORAGE FOR GRADIENTS DURING OPTIMIZATION. 00252
ISN 0129 WRITE (6,220) 00253
ISN 0130 N3SAV=N3 00254
ISN 0131 N3=1 00255
ISN 0132 IF (NCON.EQ.0) GO TO 160 00256
ISN 0134 DO 150 I=1,NCON 00257
ISN 0135 IF (G(I).GT.-.2) N3=N3+1 00258
ISN 0137 150 CONTINUE 00259
ISN 0138 160 IF (NSIDE.GT.0) N3=N3+NDV 00260
ISN 0140 IF (N3.LT.N3SAV) N3=N3SAV+NDV 00261
ISN 0142 N4=N3 00262
ISN 0143 IF (N4.LT.NDV) N4=NDV 00263
ISN 0145 NREQR=3*N1+2*N2+N3*(N1+N3)+N4 00264
ISN 0146 NREQI=N2+N3+2*N4 00265
ISN 0147 WRITE (6,250) NWK,NIWK,NREQR,NREQI 00266
ISN 0148 RETURN 00267
C ----- 00268
C FORMATS 00269
C ----- 00270
C 00271
C 00272
ISN 0149 170 FORMAT (1H1,////////12X,29(2H* )/12X,1H*,55X,1H*/12X,1H*,22X,11HS.100273
1 M C O N,22X,1H*/12X,1H*,55X,1H*/12X,1H*,18X,1SHFORTRAN PROGRAM FOC00274
2R,18X,1H*/12X,1H*,55X,1H*/12X,1H*,16X,23HSIMPLIFIED CONMIN USAGE,100275
36X,1H*/12X,1H*,55X,1H*/12X,29(2H* )) 00276
ISN 0150 180 FORMAT (////17X,16HINPUT PARAMETERS//17X,36HNUMBER OF DESIGN VARI00277
1BLES, NDV =,I5/17X,36HNUMBER OF CONSTRAINTS, NCON =,I5/100278
27X,36HMAXIMUM ITERATIONS, ITHAX =,I5/17X,36HPRINT CONTR00279
3, IPRINT =,I5/17X,36HSCALING PARAMETER, ISCA00280
4L =,I5/17X,36HCONVERGENCE CRITERIA, DELFUN =,E12.5/17X,36HC00281
5ONVERGENCE CRITERIA, DABFUN =,E12.5) 00282
ISN 0151 190 FORMAT (//30X,16HDESIGN VARIABLES/24X,5HLOWER,7X,7HINITIAL,5X,5HUP00283
1PER/24X,5HBOUND,8X,5HVALUE,6X,5HBOUND/19X,1HI,4X,6HVLB(I),8X,4HX(I00284
2),6X,6HVUB(I)) 00285
ISN 0152 200 FORMAT (15X,I5,3E12.5) 00286
ISN 0153 210 FORMAT (//19X,36HSTORAGE REQUIREMENTS FOR WORK ARRAYS) 00287
ISN 0154 220 FORMAT (////12X,47(1H*)//16X,25HSIMCON OPTIMIZATION ERROR//12X,47H00288
1STORAGE IN WORK ARRAY WK OR IWK IS INSUFFICIENT) 00289
ISN 0155 230 FORMAT (/21X,5HARRAY,3X,9HDIMENSION,2X,8HREQUIRED,3X,4HUSED/20X,7H00290

```

1KK(NNK),I8,2I10/19X,9HIKK(NIWK),I7,2I10) 00291
ISN 0156 240 FORMAT (//22X,27HOPTIMIZATION CANNOT PROCEED//22X,28HRETURNING TO 00292
ICALLING PROGRAM//12X,47(1H*)) 00293
ISN 0157 250 FORMAT (/12X,68HREQUIRED STORAGE FOR GRADIENT INFORMATION EXCEEDS 00294
1AVAILABLE STORAGE//12X,5HARRAY,14X,7HIKK(NKK),5X,9HIKK(NIWK)/12X,9H00295
2DIMENSION,15X,I6,8X,I6/12X,19HSUGGESTED DIMENSION,5X,I6,8X,I6) 00296
ISN 0158 END 00297

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOBBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
STATISTICS SOURCE STATEMENTS = 157, PROGRAM SIZE = 3780, SUBPROGRAM NAME =SIMCON
STATISTICS NO DIAGNOSTICS GENERATED
***** END OF COMPILATION ***** 2996K BYTES OF CORE NOT USED

REQUESTED OPTIONS: SOURCE,NOMAP,NOXREF,NOLIST,NODECK,OPT(3),AUTCDBL(NONE),NOALC

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTCDBL(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

322

```

C      DATA SET U477MAIN  AT LEVEL 004 AS OF 09/16/81
C      DATA SET U477MAIN  AT LEVEL 003 AS OF 03/16/81          00001
C      DATA SET U477MAIN  AT LEVEL 002 AS OF 03/13/81          00002
C      DATA SET U477MAIN  AT LEVEL 001 AS OF 02/13/81          00003
C      DATA SET 9188MAIN  AT LEVEL 001 AS OF 09/12/80          00004
C      DATA SET 9188MAIN  AT LEVEL 001 AS OF 07/10/80          00005
C      *****00006
C      COPE5 - CONTROL PROGRAM FOR ENGINEERING SYNTHESIS.      00007
C      *****00008
ISN 0002  COMMON /CHINH1/ DELFUN,DABFUN,FDCH,FDCHH,CT,CTMIN,CTL,CTLMIN,ALPHAX00009
          1,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFOG,NSCAL,LIN03J,ITMAX,IT00010
          2RM,ICNDIR,IGOTO,NAC,INFO,INFOG,ITER                    00011
ISN 0003  COMMON /COPE51/ TITLE(20)                               00012
ISN 0004  COMMON /COPE52/ RA(5000),IA(1000)                       00013
ISN 0005  COMMON /COPE53/ SENGPT,HCALC,IOBJ,NSV,NSOBJ,NCCNA,N2VX,N2VY,M00014
          12VY,N2VAR,IPSENS,IP2VAR,IPDEG,NACMX1,NDVTOT,LOCR(25),LOCI(25),ISCR00015
          21,ISCR2,NXAPRX,NPS,NPFS,NPA,NF,INOM,IPAPRX,KMIN,KMAX,XFACT1,XFACT20016
          3,NAN2,NAN3,NPHAX,NPTOT,JNOM,MAXTRM                    00017
ISN 0006  COMMON /GLOBCH/ ARRAY(1500)                             00018
ISN 0007  COMMON /UIOS/ ISI, I6I                                  00019
C      BY G. N. VANDERPLAATS          OCT., 1974.                00020
C      NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.        00021
C      NCALC OPTIONS:                                           00022
C      0. READ ALL INPUT AND STOP.                                00023
C      1. SINGLE PASS ANALYSIS.                                  00024
C      2. OPTIMIZATION.                                          00025
C      3. SENSITIVITY - Z = F(X).                                00026
C      4. TWO VARIABLE FUNCTION SPACE - Z = F(X,Y).             00027
C      5. OPTIMUM SENSITIVITY.                                   00028
C      6. ANALYSIS/OPTIMIZATION USING APPROXIMATION TECHNIQUES. 00029
C      -----00030
C      *****          INPUT          *****00031
C      -----00032
C      DIMENSIONS OF ARRAYS ARRAY, RA AND IA.                    00033
ISN 0008  CALL MYTIME (ITIME)                                     00034
ISN 0009  ITIMEC = 1                                             00035
ISN 0010  WRITE (26,3000) ITIMEC , ITIME                          00036
ISN 0011  3000 FORMAT (100X,2I10)                                00037
ISN 0012  ISI = 25                                              00038
ISN 0013  I6I = 26                                              00039
ISN 0014  NARRAY=1500                                           00040
ISN 0015  NDRA=5000                                             00041
ISN 0016  NDIA=1000                                             00042
C      READ GENERAL SYNTHESIS CONTROL INPUT.                     00043
C      SCRATCH TAPE NUMBERS.                                     00044
ISN 0017  ISCR1=11                                              00045
ISN 0018  ISCR2=12                                              00046
ISN 0019  CALL COPE01 (RA,IA,NDRA,NDIA)                          00047
ISN 0020  NAN2=0                                                00048
ISN 0021  NAN3=0                                                00049
ISN 0022  IF (NCALC.LT.0.OR.NCALC.GT.6) GO TO 340               00050
C      CHECK TO INSURE STORAGE REQUIREMENTS DO NOT EXCEED      00051
C      DIMENSIONED SIZES OF ARRAYS RA AND IA.                  00052
ISN 0024  NDRA1=LOCR(25)                                         00053
ISN 0025  NDIA1=LOCI(25)                                         00054

```

```

+VERSION 1.3.0 (01 MAY 80)      MAIN  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 82.141/10.55.45  PAGE  2
  ISN 0026      IF (NDRA1.LE.NDRA.AND.NDIA1.LE.NDIA) GO TO 10      00055
  ISN 0028      WRITE (I6I,360) NDRA,NDRA1,NDIA,NDIA1      00056
  ISN 0029      GO TO 340      00057
  ISN 0030      10 CONTINUE      00058
  C      READ USER INPUT.      00059
  ISN 0031      ICALC=1      00060
  ISN 0032      CALL ANALIZ (ICALC)      00061
  ISN 0033      IF (NCALC.LT.1.OR.NCALC.GT.6) GO TO 340      00062
  C      -----00063
  C      ***** EXECUTION *****00064
  C      -----00065
  ISN 0035      ICALC=2      00066
  ISN 0036      JCALC=3      00067
  ISN 0037      IF (NCALC.NE.2.AND.NCALC.LT.5) GO TO 60      00068
  C      -----00069
  C      IF ABS(X(I)).GT.0 OVER-RIDE USER INPUT OF DECISION VARIABLES FOR 00070
  C      OPTIMIZATION.      00071
  C      -----00072
  ISN 0039      DO 40 I=1,NDV      00073
  ISN 0040      XX=ABS(RA(I))      00074
  C      OVER-RIDE ANALIZ INPUT.      00075
  ISN 0041      N5=LOCR(5)      00076
  ISN 0042      M2=LOCI(2)      00077
  ISN 0043      DO 20 J=1,NDVTOT      00078
  ISN 0044      NN1=IA(M2)      00079
  ISN 0045      M2=M2+1      00080
  ISN 0046      IF (NN1.NE.I) GO TO 20      00081
  ISN 0048      NN1=IA(J)      00082
  ISN 0049      IF (XX.LT.1.0E-10) GO TO 30      00083
  ISN 0051      ARRAY(NN1)=RA(I)*RA(N5)      00084
  ISN 0052      20 N5=N5+1      00085
  ISN 0053      GO TO 40      00086
  ISN 0054      30 RA(I)=ARRAY(NN1)/RA(N5)      00087
  ISN 0055      40 CONTINUE      00088
  C      TRANSFER DESIGN VARIABLES TO ARRAY.      00089
  ISN 0056      M2=LOCI(2)      00090
  ISN 0057      N5=LOCR(5)      00091
  ISN 0058      DO 50 I=1,NDVTOT      00092
  ISN 0059      N=IA(M2)      00093
  ISN 0060      M=IA(I)      00094
  ISN 0061      ARRAY(M)=RA(N)*RA(N5)      00095
  ISN 0062      N5=N5+1      00096
  ISN 0063      50 M2=M2+1      00097
  ISN 0064      60 CONTINUE      00098
  ISN 0065      IF (NCALC.NE.3.AND.NCALC.NE.5) GO TO 80      00099
  C      -----00100
  C      TRANSFER NOMINAL VALUES OF SENSITIVITY VARIABLES TO ARRAY. 00101
  C      -----00102
  ISN 0067      M16=LOCI(16)      00103
  ISN 0068      M17=LOCI(17)      00104
  ISN 0069      N15=LOCR(15)      00105
  ISN 0070      DO 70 I=1,NSV      00106
  ISN 0071      NN=IA(M16)      00107
  ISN 0072      M16=M16+1      00108
  ISN 0073      ARRAY(NN)=RA(N15)      00109
  ISN 0074      N15=N15+IA(M17)      00110
  ISN 0075      70 M17=M17+1      00111
  ISN 0076      80 CONTINUE      00112
  ISN 0077      IF (NCALC.LT.6) GO TO 290      00113

```

324

```

C -----00114
C           INITIALIZATION FOR APPROXIMATE ANALYSIS/OPTIMIZATION 00115
C -----00116
ISN 0079      IF (NPA.EQ.0) GO TO 130 00117
C           ANALIZ INPUT DEFINES AN X-VECTOR. 00118
ISN 0081      M5=LOCI(5) 00119
ISN 0082      N23=LOCR(23) 00120
ISN 0083      DO 120 I=1,NXAPRX 00121
ISN 0084      J=IA(M5) 00122
C           IS THIS A DESIGN VARIABLE. 00123
ISN 0085      DO 90 K=1,NDVTOT 00124
ISN 0086      KK=K 00125
ISN 0087      IF (IA(K).EQ.J) GO TO 100 00126
ISN 0089      90 CONTINUE 00127
C           NO. 00128
ISN 0090      AMULT=1. 00129
ISN 0091      GO TO 110 00130
C           YES. 00131
ISN 0092      100 K=LOCR(5)+KK-1 00132
ISN 0093      AMULT=RA(K) 00133
ISN 0094      110 RA(N23)=ARRAY(J)/AMULT 00134
ISN 0095      M5=M5+1 00135
ISN 0096      120 N23=N23+1 00136
ISN 0097      130 CONTINUE 00137
ISN 0098      IF (NPS.GT.0.OR.NPFS.GT.0) GO TO 190 00138
C           ONLY ONE DESIGN VECTOR IS AVAILABLE. CREATE A SECOND X-VECTOR 00139
C           SO OPTIMIZATION CAN PROCEED. 00140
ISN 0100      N23=LOCR(23) 00141
ISN 0101      N24=N23+NXAPRX 00142
ISN 0102      M5=LOCI(5) 00143
ISN 0103      DO 180 I=1,NXAPRX 00144
C           GLOBAL LOCATION. 00145
ISN 0104      IG=IA(M5) 00146
ISN 0105      M5=M5+1 00147
C           PROPOSED X-VALUE. 00148
ISN 0106      XX=1.1*RA(N23) 00149
ISN 0107      IF (ABS(XX).LT.1.0E-10) XX=.1 00150
C           IS THIS A DESIGN VARIABLE. 00151
ISN 0109      N5=LOCR(5) 00152
ISN 0110      DO 140 J=1,NDVTOT 00153
ISN 0111      JJ=J 00154
ISN 0112      AMJ=RA(N5) 00155
ISN 0113      IF (IA(J).EQ.IG) GO TO 150 00156
C           NO. 00157
ISN 0115      N5=N5+1 00158
ISN 0116      140 CONTINUE 00159
ISN 0117      GO TO 170 00160
ISN 0118      150 CONTINUE 00161
C           YES. WHICH DESIGN VARIABLE IS IT. 00162
ISN 0119      ID=LOCI(2)+JJ-1 00163
ISN 0120      ID=IA(ID) 00164
C           INSURE XX IS WITHIN BOUNDS. 00165
ISN 0121      N2=LOCR(2)+ID-1 00166
ISN 0122      N3=LOCR(3)+ID-1 00167
ISN 0123      BL=RA(N2)*ABS(AMJ) 00168
ISN 0124      BU=RA(N3)*ABS(AMJ) 00169
ISN 0125      IF (BL.LE.BU) GO TO 160 00170
ISN 0127      SAV=BL 00171
ISN 0128      BL=BU 00172

```

+VERSION 1.3.0 (01 MAY 80)		MAIN	SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)	DATE 82.141/10.55.45	PAGE 4
ISN 0129			BU=SAV	00173	
ISN 0130	160		IF (XX.LT.BL) XX=BL	00174	
ISN 0132			IF (XX.GT.BU) XX=BU	00175	
ISN 0134			DX=RA(N23)-XX	00176	
ISN 0135			IF (ABS(DX).LT.1.0E-6) XX=1.001*RA(N23)	00177	
ISN 0137			IF (ABS(XX).LT.1.0E-6) XX=.001	00178	
ISN 0139	170		RA(N24)=XX	00179	
ISN 0140			N23=N23+1	00180	
ISN 0141	180		N24=N24+1	00181	
ISN 0142	190		CONTINUE	00182	
ISN 0143			REHIND ISCR2	00183	
ISN 0144			NPSA=NPS+NPA	00184	
ISN 0145			IF (NPSA.EQ.0) GO TO 250	00185	
ISN 0147			IF (NPS.EQ.0) GO TO 210	00186	
		C	READ X-VECTORS.	00187	
ISN 0149			NXI=LOCR(23)+NPA*NXAPRX	00188	
ISN 0150			DO 200 J=1,NPS	00189	
ISN 0151			NXIJ=NXI+NXAPRX-1	00190	
ISN 0152			READ (ISCR2) (RA(I),I=NXI,NXIJ)	00191	
ISN 0153	200		NXI=NXI+NXAPRX	00192	
ISN 0154	210		CONTINUE	00193	
ISN 0155			IF (NPFS.LE.0) NPSA=NPTOT	00194	
ISN 0157			NXI=LOCR(23)	00195	
ISN 0158			NY=NXI+NXAPRX*NPTOT	00196	
ISN 0159			DO 240 J=1,NPSA	00197	
		C	TRANSFER X-VALUES.	00198	
ISN 0160			M5=LOCI(5)	00199	
ISN 0161			I1=NXI	00200	
ISN 0162			DO 220 I=1,NXAPRX	00201	
ISN 0163			II=IA(M5)	00202	
ISN 0164			ARRAY(II)=RA(I1)	00203	
ISN 0165			M5=M5+1	00204	
ISN 0166	220		I1=I1+1	00205	
		C	ANALIZE.	00206	
ISN 0167			NAN2=NAN2+1	00207	
ISN 0168			CALL ANALIZ (ICALC)	00208	
		C	PUT FUNCTION VALUES IN Y-ARRAY.	00209	
ISN 0169			M6=LOCI(6)	00210	
ISN 0170			I1=NY	00211	
ISN 0171			DO 230 I=1,NF	00212	
ISN 0172			II=IA(M6)	00213	
ISN 0173			RA(I1)=ARRAY(II)	00214	
ISN 0174			I1=I1+1	00215	
ISN 0175	230		M6=M6+1	00216	
ISN 0176			NXI=NXI+NXAPRX	00217	
ISN 0177			NY=NY+NF	00218	
ISN 0178	240		CONTINUE	00219	
ISN 0179	250		CONTINUE	00220	
ISN 0180			IF (NPFS.LE.0) GO TO 270	00221	
ISN 0182			NXI=LOCR(23)+NPSA*NXAPRX	00222	
ISN 0183			NY=LOCR(23)+NXAPRX*NPTOT+NF*NPSA	00223	
		C	READ X AND Y VECTORS.	00224	
ISN 0184			DO 260 J=1,NPFS	00225	
ISN 0185			NXIJ=NXI+NXAPRX-1	00226	
ISN 0186			NYJ=NY+NF-1	00227	
		C	X-VECTOR.	00228	
ISN 0187			READ (ISCR2) (RA(I),I=NXI,NXIJ)	00229	
		C	Y-VECTOR.	00230	
ISN 0188			READ (ISCR2) (RA(I),I=NY,NYJ)	00231	

This Page Intentionally Left Blank

APPENDIX C
REVISED FORTRAN FOR SUPERHYBRID
BLADE DEMONSTRATION CASE

330

```

C      DATA SET U493DSTAEB AT LEVEL TMP AS OF 05/21/82
C      DATA SET U498DSTAEB AT LEVEL 018 AS OF 01/20/82      00001
C      DATA SET U498STAEBL AT LEVEL 015 AS OF 10/06/81      00002
C      DATA SET U498STAEBL AT LEVEL 013 AS OF 08/24/81      00003
C      DATA SET U498STAEBL AT LEVEL 012 AS OF 08/14/81      00004
ISN 0002 SUBROUTINE STAEBL      00005
ISN 0003 COMMON /BLKCW/ XP(3,53,21) , NP(21) , IND(21) , INP      00006
ISN 0004 COMMON /BLK 2/ NOACR , NOACR1 , NSTA , NSTA1 , POISSN,MST2,MST3      00007
ISN 0005 COMMON /BLK 4/ RPM , XNOACR , BR , PI , TERM , NBS      00008
ISN 0006 COMMON /BLK 7/ AREA(21) , CF(21) , AA(21) , AKG(21) , IT(7,21)      00009
ISN 0007 COMMON /BLK12/ XBARX(21) , XIMAX(21) , XIMIN(21) , YBARY(21) ,      00010
      1 TLTA(21)      00011
ISN 0008 COMMON /BLKAA/ ALPHIN(21) , XSC(21) , YSC(21) , XX(21) , YY(21)      00012
ISN 0009 COMMON /BLK8/ SMB(21),ITTL(18),VARI(235),TMAX(21),HALPHA(21)      00013
ISN 0010 COMMON /BLK A/ ALPHA(21) , HA(21) , HMHT(21) , HALMIN(21) ,      00014
      1 HINERT(21)      00015
ISN 0011 COMMON /BLK 9/ DELTAC(21) , DELTAZ(21) , P(21) , R(21) , SHWT(21)      00016
ISN 0012 COMMON /INPUTT/ XSAVE(1000) , YSAVE(1000) , ZSAVE(1000) , TSAVE(1000)      00017
ISN 0013 COMMON /SC/ XSCSV(21) , YSCSV(21) , ZSCSV(21) ,      00018
      1 ALSAVE(21) , FOLARI(21) , ASAVE(21)      00019
ISN 0014 COMMON /ANAL33/ DLED , DTED , DROOTD , DTIPD , TTID , TLTD , NCD      00020
      1 , NCK , TSKIN , TCENTR , PBT , PGE      00021
ISN 0015 DIMENSION YH(53,21) , XM(53,21) , TMC(53,21) , T(120) , TML(53,21)      00022
      1 , XMF(50,21) , YMF(50,21) , TMLF(50,21) , XCG(50,21) , YCG(50,21)      00023
      2 , XSCCG(21) , YSCCG(21) , XSCG(21) , YSCG(21) , ZSCG(21) ,      00024
      3 XG(50,21) , YG(50,21) , ZG(50,21)      00025
      4 , XSCR(21) , YSCR(21) , ZSCR(21)      00026
      5 , X(50,21) , Y(50,21) , Z(50,21)      00027
ISN 0016 REAL*4 A(51),B(51),C(51),D(51),A1(51),B1(51),C1(51),D1(51),      00028
      1 A2(51),B2(51),C2(51),D2(51),A3(51),B3(51),C3(51),      00029
      2 D3(51),S1(51),S2(51)      00030
ISN 0017 REAL*4 XMM(51),YMM(51),TMM(51),XXMF(25),YYMF(25),TTMLF(25)      00031
ISN 0018 REAL*4 SONE,STWO      00032
C      CALCULATE MEAN Y VALUE AND THICKNESS NORMAL TO THE CHORD      00033
C      00034
C      00035
C      00036
ISN 0019 NBR = 1      *20
ISN 0020 NPS = NP(1)      *20
ISN 0021 DO 5 N = 1,NPS      *20
ISN 0022 XP(1,N,1) = XP(1,N,2)      *20
ISN 0023 XP(2,N,1) = XP(2,N,2)      *20
ISN 0024 XP(3,N,1) = XP(3,N,2)      *20
ISN 0025 5 CONTINUE      *20
ISN 0026 DO 20 I = NBR,NSTA      00037
ISN 0027 NPS = NP(I)      00038
ISN 0028 DO 10 N = 1,NPS      00039
ISN 0029 YM(N,I) =(XP(2,N,I) + XP(3,N,I)) / 2.0      00040
ISN 0030 XM(N,I) = XP(1,N,I)      00041
ISN 0031 THIC(N,I) = XP(2,N,I) - XP(3,N,I)      00042
ISN 0032 10 CONTINUE      00043
ISN 0033 20 CONTINUE      00044
C      CALCULATE THICKNESS NORMAL TO MEANLINE      00045
C      00046
C      00047
ISN 0034 DO 40 I = NBR,NSTA      00048
  
```

+VERSION 1.3.0 (01 MAY 80) STAEBL SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)		DATE 82.141/10.58.59	PAGE 2
ISN 0035	NPS = NP(I)-1	00049	
ISN 0036	TML(1,I) = 0.0	00050	
ISN 0037	TML(NPS+1,I) = 0.0	00051	
ISN 0038	DO 30 N =2,NPS	00052	
ISN 0039	THT1 =ATAN2((YM(N,I) - YM(N-1,I)) ,(XM(N,I) - XM(N-1,I)))	00053	
ISN 0040	THT2 =ATAN2((YM(N+1,I) - YM(N,I)) ,(XM(N+1,I) - XM(N,I)))	00054	
ISN 0041	THT = (THT1 + THT2) / 2.0	00055	
ISN 0042	TML(N,I) = TMC(N,I) * COS(THT)	00056	
ISN 0043	30 CONTINUE	00057	
ISN 0044	40 CONTINUE	00058	
ISN 0045	NSTNB = NSTA - NBR	00059	
ISN 0046	DO 45 I = NBR,NSTA,NSTNB	00060	
ISN 0047	NPS = NP(I)	00061	
	C WRITE (6,900) (YM(N,I),N=1,NPS)	00062	
	C WRITE (6,900) (XM(N,I),N=1,NPS)	00063	
	C WRITE (6,900) (TMC(N,I),N=1,NPS)	00064	
	C WRITE (6,900) (TML(N,I),N=1,NPS)	00065	
ISN 0048	45 CONTINUE	00066	
	C	00067	
	C SAVE MEAN Y VALVES AND X VALVES IN AN ARRAY AND INTERPOLATE	00068	
	C FOR EQUAL INCREMENT X VALVES . REPEAT FOR THICKNESSES	00069	
	C	00070	
ISN 0049	DO 110I = NBR,NSTA	00071	
ISN 0050	NPF = 15	00072	
ISN 0051	NPS = NP(I)	00073	
	C	00074	
	C FILL IN DUMMY ARRAYS XMM,YMM,TMM FROM X1,YM,TML VECTORS	00075	
	C	00076	
ISN 0052	DO 201 K = 1,NPS	00077	
ISN 0053	TMM(K) = TML(K,I)	00078	
ISN 0054	XMM(K) = XM(K,I)	00079	
ISN 0055	201 YMM(K) = YM(K,I)	00080	
	C	00081	
	C	00082	
	C CALL PBMFIT FOR CURVE FIT	00083	
	C	00084	
ISN 0056	CALL PBMFIT(XMM,YMM,1.,1.,NPS,0,A,B,C,D,A1,B1,C1,D1,S1)	00085	
	C	00086	
ISN 0057	CALL PBMFIT(XMM,TMM,1.,1.,NPS,0,A2,B2,C2,D2,A3,B3,C3,D3,S2)	00087	
	C	00088	
	C EQUAL BREAKUP ARC-LENGTH	00089	
	C	00090	
ISN 0058	SARC = S1(NPS) / NPF	00091	
	C	00092	
	C NOW SEARCH S1 ARRAY FOR INTERVAL VALUE	00093	
	C	00094	
ISN 0059	NODE = 1	00095	
ISN 0060	ARCI = SARC / 2.	00096	
ISN 0061	205 N = 1	00097	
ISN 0062	210 IF(ARCI .LE. S1(N)) GO TO 220	00098	
ISN 0064	N = N + 1	00099	
ISN 0065	GO TO 210	00100	
ISN 0066	220 DIST = (ARCI - S1(N-1)) / (S1(N) - S1(N-1))	00101	
ISN 0067	SONE = (S1(N) - S1(N-1)) * DIST	00102	
ISN 0068	STWO = (S2(N) - S2(N-1)) * DIST	00103	
	C	00104	
ISN 0069	CALL CUBIC(A(N-1),B(N-1),C(N-1),D(N-1),SONE,XXMF(NODE))	00105	
ISN 0070	CALL CUBIC(A1(N-1),B1(N-1),C1(N-1),D1(N-1),SONE,YYMF(NODE))	00106	
ISN 0071	CALL CUBIC(A3(N-1),B3(N-1),C3(N-1),D3(N-1),STWO,TTMLF(NODE))	00107	

332

```

C
ISN 0072      NODE = NODE + 1
ISN 0073      ARCI = ARCI + SARC
ISN 0074      IF(NODE .LE. NPF) GO TO 205
C
C          FILL IN THE XMF,YMF,TMLF VECTORS WITH THE INTERPOLATED VALUES
C
ISN 0076      DO 230 K = 1,NPF
ISN 0077      XMF(K,I) = XXMF(K)
ISN 0078      YMF(K,I) = YYMF(K)
C          WRITE(6,902) I,K,XMF(K,I),YMF(K,I)
ISN 0079      902  FORMAT(5X,'I,K,XMF,YMF',2I5,2E12.5)
ISN 0080      230  TMLF(K,I) = TTMLF(K)
C
ISN 0081      110 CONTINUE
C
C          TRANSLATE COORDINATES TO AN ENGINE AXIS SYSTEM
C
ISN 0082      DO 130 I = NBR,NSTA
C
ISN 0083      XCHORD = SNB(I)
C
ISN 0084      XSCLE = XCHORD - XSC(I)
C
ISN 0085      XSCCG(I) = XSCLE - (XCHCRD - XBARX(I))
ISN 0086      YSCCG(I) = YSC(I) - YBARY(I)
C
ISN 0087      DO 120 N = 1,NPF
ISN 0088      XCG(N,I) = XMF(N,I) - (XCHORD - XBARX(I))
ISN 0089      YCG(N,I) = YMF(N,I) - YBARY(I)
ISN 0090      120 CONTINUE
ISN 0091      130 CONTINUE
C
C          SHIFT XY PLANE TO ENGINE AXIS YZ
C
ISN 0092      DO 150 I = NBR,NSTA
ISN 0093      XSCG(I) = R(I)
ISN 0094      YSCG(I) = YSCCG(I)
ISN 0095      ZSCG(I) = XSCCG(I)
ISN 0096      DO 140 N = 1,NPF
ISN 0097      XG(N,I) = R(I)
ISN 0098      YG(N,I) = YCG(N,I)
ISN 0099      ZG(N,I) = XCG(N,I)
ISN 0100      140 CONTINUE
ISN 0101      150 CONTINUE
C
C          ROTATE THRU ALPHA CHORD - 90.0
C
ISN 0102      DO 170 I = NBR,NSTA
C          ALPHA(I) = 90. * .0174533
ISN 0103      ANG = -ALPHA(I) + 90.0 * .0174533
ISN 0104      EM = COS(ANG)
ISN 0105      EN = SIN(ANG)
ISN 0106      XSCR(I) = XSCG(I)
ISN 0107      YSCR(I) = YSCG(I) * EM + EN * ZSCG(I)
ISN 0108      ZSCR(I) = EN * ZSCG(I) - EM * YSCG(I)
ISN 0109      WRITE (6,900) ANG, EN, EM, XSCR(I), YSCR(I), ZSCR(I), AREA(I)
ISN 0110      DO 160 N = 1,NPF
C          X(N,I) = XG(N,I)

```

03108
00109
00110
00111
00112
00113
00114
00115
00116
00117
00118
00119
00120
00121
00122
00123
00124
00125
00126
00127
00128
00129
00130
00131
00132
00133
00134
00135
00136
00137
00138
00139
00140
00141
00142
00143
00144
00145
00146
00147
00148
00149
00150
00151
00152
00153
00154
00155
00156
00157
00158
00159
00160
00161
00162
00163
00164
00165
00166

+VERSION 1.3.0 (01 MAY 80) STAEBL SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)		DATE 82.141/10.58.59	PAGE 4
ISN 0111	Y(N,I) = YG(N,I)* EM + EN * ZG(N,I)	00167	
ISN 0112	Z(N,I) = EM * ZG(N,I) - EN * YG(N,I)	00168	
ISN 0113	160 CONTINUE	00169	
ISN 0114	170 CONTINUE	00170	
	C DO 175 I = NBR,NSTA,NSTNB	00171	
	C WRITE (6,900) (XCG(N,I),N=1,NPF)	00172	
	C WRITE (6,900) (YCG(N,I),N=1,NPF)	00173	
	C WRITE (6,900) (XG(N,I),N=1,NPF)	00174	
	C WRITE (6,900) (YG(N,I),N=1,NPF)	00175	
	C WRITE (6,900) (ZG(N,I),N=1,NPF)	00176	
	C WRITE (6,900) (X(N,I),N=1,NPF)	00177	
	C WRITE (6,900) (Y(N,I),N=1,NPF)	00178	
	C WRITE (6,900) (Z(N,I),N=1,NPF)	00179	
	C WRITE (6,900) (TMLF(N,I),N=1,NPF)	00180	
	C 175 CONTINUE	00181	
ISN 0115	IJ = 1	00182	
ISN 0116	IF(IJ .EQ. 1) GO TO 811	00183	
	C	00184	
	C T03902 GENERATED NODES AND THICKNESSES	00185	
	C	00185	
ISN 0118	DO 800 I = NBR,NSTA	00187	
ISN 0119	DO 801 J = 1,NPF	00188	
ISN 0120	801 READ(5,802) Y(J,I),Z(J,I)	00189	
ISN 0121	802 FORMAT(/,32X,2F8.0)	00190	
	C	00191	
ISN 0122	DO 805 J = 1,NPF	00192	
ISN 0123	READ(5,803) P1,P2	00193	
ISN 0124	803 FORMAT(24X,F8.0,/,24X,F8.0)	00194	
ISN 0125	805 TMLF(J,I) = (P1 + P2) / 2.	00195	
	C	00196	
	C WRITE(6,810) I	00197	
	C810 FORMAT(5X,'T039 NODES AND THICKNESSES FOR SECTION ',I5)	00198	
	C WRITE(6,900) (X(N,I),N=1,NPF)	00199	
	C WRITE(6,900) (Y(N,I),N=1,NPF)	00200	
	C WRITE(6,900) (Z(N,I),N=1,NPF)	00201	
	C WRITE(6,900) (TMLF(N,I),N=1,NPF)	00202	
	C	00203	
ISN 0126	800 CONTINUE	00204	
ISN 0127	811 CONTINUE	00205	
	C	00205	
ISN 0128	J = 0	00207	
ISN 0129	K = 0	00208	
ISN 0130	DO 190 I = NBR,NSTA	00209	
ISN 0131	K = K + 1	00210	
ISN 0132	XSCSV(K) = XSCR(I)	00211	
ISN 0133	YSCSV(K) = YSCR(I)	00212	
ISN 0134	ZSCSV(K) = ZSCR(I)	00213	
ISN 0135	ALSAVE(K) = ALPHA(I)	00214	
ISN 0136	POLARI(K) = XIMIN(I) + XIMAX(I)	00215	
ISN 0137	ASAVE(K) = AREA(I)	00216	
ISN 0138	DO 180 N = 1,NPF	00217	
ISN 0139	J = J + 1	00218	
ISN 0140	XSAVE(J) = X(N,I)	00219	
ISN 0141	YSAVE(J) = Y(N,I)	00220	
ISN 0142	ZSAVE(J) = Z(N,I)	00221	
ISN 0143	TSAVE(J) = TMLF(N,I)	00222	
	C WRITE (7,901) XSAVE(J) , YSAVE(J) , ZSAVE(J) , TSAVE(J)	00223	
ISN 0144	180 CONTINUE	00224	
ISN 0145	190 CONTINUE	00225	

334

```

+VERSION 1.3.0 (01 MAY 80)   STAEBL   SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)   DATE 82.141/10.58.59   PAGE 5
ISN 0146           K = 0                                           00226
ISN 0147           DO 200 I = NBR,NSTA                               00227
ISN 0148           K = K + 1                                         00228
                   C WRITE (7,901) XSCSV(K) , YSCSV(K) , ZSCSV(K) , ALSAVE(K) 00229
                   C 1 , POLARI(K) , ASAVE(K)                         00230
ISN 0149           200 CONTINUE                                       00231
ISN 0150           NTMNR = NSTA - NBR + 1                             00232
ISN 0151           IF (NCD .EQ. 1)                                    00233
                   1CALL HOLLOW (DLED , DTED , DROOTD , DTIPD ,      00234
                   2 TTID , TLTD , NPF , NTMNR)                       00235
ISN 0153           IF (NCD .EQ. 2)                                    00236
                   1 CALL LAMINI (TSKIN , TCENTR , PBT , PGE , NPF , NTMNR) 00237
ISN 0155           901 FORMAT (6E12.5)                               00238
ISN 0156           900 FORMAT (1X,9F8.5)                             00239
ISN 0157           RETURN                                             00240
ISN 0158           END                                               00241

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOGBL(DBL4)
*OPTICNS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NCMAP NOFCRMT COSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 157, PROGRAM SIZE = 141976, SUBPROGRAM NAME =STAEBL
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****
2972K BYTES OF CORE NOT USED

```

```

C 00370
ISN 0002 SUBROUTINE INPUT (NR,NC) 00871
C 00872
C MATERIAL PROPERTIES AND GEOMETRY INPUT 00873
C 00874
ISN 0003 IMPLICIT REAL*8(A-H,O-Z) 00875
ISN 0004 COMMON /ZCOEF/ E11(7),E22(7),E33(7),G12(7),G23(7),G13(7), 00876
1 V12(7),V13(7),V23(7) 00877
ISN 0005 COMMON /PLY1/ PLY(21,25,7),THETA(7),RHO(7) 00878
ISN 0006 COMMON /FAIL/ X1T(7),X1C(7),X2T(7),X2C(7),S6P(7),S6M(7),TSAI(6,25) 00379*35
C 00880
C WRITE(8,1) 00881
C1 FORMAT(5X,'IS THIS AN ISOTROPIC BLADE, 1=YES') 00882
C READ(8,*) ISO 00883
C IF(ISO .NE. 1) GO TO 2 00884
C 00885
C ISOTROPIC BLADE 00886
C 00887
C WRITE(8,3) 00888
C3 FORMAT(5X,'INPUT-E(PSI),V AND DENSITY(LB/IN3),FREE FORMAT') 00889
C READ(8,*) E,V,R 00890
ISN 0007 E = 16100000. 00891
ISN 0008 V = .33 00892
ISN 0009 R = .16 00893
ISN 0010 DPLY = 1. / 7. 00894
ISN 0011 YIELD = 110000. 00895*35
ISN 0012 SHEAR = 0.577 * YIELD 00896*36
C *49
C TI LAYERS *49
C *49
ISN 0013 I = 1 *49
ISN 0014 DO 100 II = 1,3 *49
C 00898*35
ISN 0015 X1T(II) = YIELD 00899*35
ISN 0016 X1C(II) = YIELD 00900*35
ISN 0017 X2T(II) = YIELD 00901*35
ISN 0018 X2C(II) = YIELD 00902*35
ISN 0019 S6P(II) = SHEAR 00903*35
ISN 0020 S6M(II) = SHEAR 00904*35
ISN 0021 E11(II) = E 00905
ISN 0022 E22(II) = E 00906
ISN 0023 E33(II) = E 00907
ISN 0024 G12(II) = E / 2. / (1. + V) 00908
ISN 0025 G13(II) = G12(II) 00909
ISN 0026 G23(II) = G12(II) 00910
ISN 0027 V12(II) = V 00911
ISN 0028 V13(II) = V 00912
ISN 0029 V23(II) = V 00913
ISN 0030 RHO(II) = R / 386.4 *49
ISN 0031 THETA(II) = 0. *49
ISN 0032 100 I = I + 3 *49
C *49
C B/A AND G/E LAYERS *49
C *49
ISN 0033 I2 = 2 *49

```

335

ISN 0034	I3 = 3	*49
	C	*49
ISN 0035	E1BA = 31.0E6	*49
ISN 0036	E1BA = 0.9 * E1BA	*49
ISN 0037	E2BA = 20.0E6	*49
ISN 0038	E2BA = 0.9 * E2BA	*49
ISN 0039	V12BA = 0.27	*49
ISN 0040	G3A = 8.5E6	*49
ISN 0041	G3A = 0.9 * G3A	*49
ISN 0042	R3A = 0.097	*49
ISN 0043	R3A = 0.9 * R3A	*49
ISN 0044	X1TBA = 140000.0	*49
ISN 0045	X1TBA = 0.9 * X1TBA	*49
ISN 0046	X1CBA = 200000.0	*49
ISN 0047	X1CBA = 0.9 * X1CBA	*49
ISN 0048	X2TBA = 14800.0	*49
ISN 0049	X2TBA = 0.9 * X2TBA	*49
ISN 0050	X2CBA = 32200.0	*49
ISN 0051	X2CBA = 0.9 * X2CBA	*49
ISN 0052	SBA = 14500.0	*49
ISN 0053	SBA = 0.9 * SBA	*49
	C	*49
ISN 0054	E1GE = 18.5E6	*49
ISN 0055	E2GE = 1.54E6	*49
ISN 0056	V12GE = 0.3	*49
ISN 0057	GGE = 0.85E6	*49
ISN 0058	RGE = 0.056	*49
ISN 0059	X1TGE = 160000.0	*49
ISN 0060	X1CGE = 160000.0	*49
ISN 0061	X2TGE = 7500.0	*49
ISN 0062	X2CGE = 25000.0	*49
ISN 0063	SGE = 10000.0	*49
	C	*49
ISN 0064	DD 200 I1 = 1,2	*49
ISN 0065	E11(I2) = E1BA	*49
ISN 0066	E22(I2) = E2BA	*49
ISN 0067	E33(I2) = E22(I2)	*49
ISN 0068	G12(I2) = G3A	*49
ISN 0069	G13(I2) = G3A	*49
ISN 0070	G23(I2) = G3A	*49
ISN 0071	V12(I2) = V12BA	*49
ISN 0072	V13(I2) = V12BA	*49
ISN 0073	V23(I2) = V12BA	*49
ISN 0074	RHO(I2) = R3A / 386.4	*49
ISN 0075	X1T(I2) = X1TBA	*49
ISN 0076	X1C(I2) = X1CBA	*49
ISN 0077	X2T(I2) = X2TBA	*49
ISN 0078	X2C(I2) = X2CBA	*49
ISN 0079	S6P(I2) = SBA	*49
ISN 0080	S6M(I2) = SBA	*49
	C	*49
ISN 0081	E11(I3) = E1GE	*49
ISN 0082	E22(I3) = E2GE	*49
ISN 0083	E33(I3) = E22(I3)	*49
ISN 0084	G12(I3) = GGE	*49
ISN 0085	G13(I3) = GGE	*49
ISN 0086	G23(I3) = GGE	*49
ISN 0087	V12(I3) = V12GE	*49
ISN 0088	V13(I3) = V12GE	*49

```

+VERSION 1.3.0 (01 MAY 80)      INPUT  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 82.141/10.59.56      PAGE  3
ISN 0089      V23(I3) = V12GE      *49
ISN 0090      RHO(I3) = RGE / 386.4      *49
ISN 0091      X1T(I3) = X1TGE      *49
ISN 0092      X1C(I3) = X1CGE      *49
ISN 0093      X2T(I3) = X2TGE      *49
ISN 0094      X2C(I3) = X2CGE      *49
ISN 0095      S6P(I3) = SGE      *49
ISN 0096      S6M(I3) = SGE      *49
      C      *49
ISN 0097      I2 = I2 + 4      *49
ISN 0098      200 I3 = I3 + 2      *49
      C
      C
      C
ISN 0099      RETURN      00961*41
ISN 0100      END      00962
      00963
      00964
      00965

```

```

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOBBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I).
*STATISTICS* SOURCE STATEMENTS = 99, PROGRAM SIZE = 1402, SUBPROGRAM NAME = INPUT
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

```

3004K BYTES OF CORE NOT USED

This Page Intentionally Left Blank

REFERENCES

1. MacNeal, R. H., ed., " The NASTRAN Theoretical Manual", NASA SP-221(01), April, 1972.
2. Dokainish, M. A. and Rawtani, S., "Pseudo-Static Deformation and Frequencies of Rotating Turbomachinery Blades", AIAA Journal, Vol. 10, No. 11 (Nov., 1972), 1397-1398.
3. Stelson, T. S. and Barth, C. F., "Cost Analysis of Composite Fan Blade Manufacturing Processes", NASA Report CR-159876

This Page Intentionally Left Blank

DISTRIBUTION LIST FOR INTERIM REPORT
NASA CR-167949

STRUCTURAL TAILORING OF ENGINE BLADES (STAEBL)
CONTRACT NAS 3-22525

	Mail Stop	Copies
NASA Lewis Research Center		
21000 Brookpark Road		
Cleveland, OH 44135		
Attn: Contracting Officer	500-313	1
Technical Report Control Office	5-5	1
Technology Utilization Office	7-3	1
AFSC Liaison Office	501-3	1
S&MT Division Contract File	49-6	2
Library	60-3	1
L. Berke	49-6	1
R. H. Johns	49-6	1
L. J. Kiraly	23-2	1
C. C. Chamis	49-6	4
M. S. Hirschbein	49-6	1
J. A. Ziemianski	49-6	1
D. H. Buckley	23-2	1
E. V. Zaretsky	6-1	1
D. P. Fleming	6-1	1
R. E. Kielb	49-6	1
G. V. Brown	49-6	1
J. J. Adamczyk	5-9	1
D. L. Nored	301-2	1
National Aeronautics & Space Administration		
Washington, DC 20546		
Attn: NHS-22/Library		1
RTM-6/S. L. Venneri		1
RTM-6/D. J. Weidman		1
NASA Ames Research Center		
Moffett Field, CA 94035		
Attn: Library	202-3	1
NASA Goddard Space Flight Center		
Greenbelt, MD 20771		
Attn: 252/Library		1
NASA John F. Kennedy Space Center		
Kennedy Space Center, FL 32931		
Attn: Library	AD-CSO-1	1

NASA Langley Research Center Hampton, VA 23665 Attn: Library	185	1
M. F. Card	244	1
NASA Lyndon B. Johnson Space Center Houston, TX 77001 Attn: JM6/Library		1
NASA George C. Marshall Space Flight Center Marshall Space Flight Center, AL 35812 Attn: AS61/Library		1
Jet Propulsion Laboratory 4800 Oak Grove Drive Pasadena, CA 91103 Attn: Library		1
B. Wada		1
R. Levi		1
NASA S&T Information Facility P. O. Box 8757 Baltimore-Washington Int. Airport, MD 21240 Attn: Acquisition Department		10
Air Force Aeronautical Propulsion Laboratory Wright-Patterson AFB, OH 45433 Attn: Z. Gershon		1
E. Bailey		1
Air Force Systems Command Aeronautical Systems Division Wright-Patterson AFB, OH 45433 Attn: Library		1
C. W. Cowie		1
J. McBane		1
Aerospace Corporation 2400 E. El Segundo Blvd. Los Angeles, CA 90045 Attn: Library-Documents		1
Air Force Office of Scientific Research Washington, DC 20333 Attn: A. K. Amos		1
Department of the Army U. S. Army Material Command Washington, DC 20315 Attn: AMCRD-RC		1
U. S. Army Ballistics Research Laboratory Aberdeen Proving Ground, MD 21005 Attn: Dr. Donald F. Haskell	DRXBR-BM	1

Mechanics Research Laboratory
Army Materials & Mechanics Research Center
Watertown, MA 02172
Attn: Dr. Donald W. Oplinger 1

U. S. Army Missile Command
Redstone Scientific Information Center
Redstone Arsenal, AL 35808
Attn: Document Section 1

AFFDL/FBE
Wright-Patterson AFB, OH 45433
Attn: D. W. Smith 1

Commanding Officer
U.S. Army Research Office (Durham)
Box CM, Duke Station
Durham, NC 27706
Attn: Library 1

Bureau of Naval Weapons
Department of the Navy
Washington, DC 20360
Attn: RRRE-6 1

Commander
U.S. Naval Ordnance Laboratory
White Oak
Silver Springs, MD 20910
Attn: Library 1

Director, Code 6180
U. S. Naval Research Laboratory
Washington, DC 20390
Attn: Library 1

Denver Federal Center
U.S. Bureau of Reclamation
P. O. Box 25007
Denver, CO 80225
Attn: P. M. Lorenz 1

Naval Air Propulsion Test Center
Aeronautical Engine Department
Trenton, NJ 08628
Attn: Mr. James Salvino 1

Naval Air Propulsion Test Center
Aeronautical Engine Department
Trenton, NJ 08628
Attn: Mr. Robert DeLucia 1

Federal Aviation Administration
Code ANE-214, Propulsion Section
12 New England Executive Park
Burlington, MA 01803
Attn: Mr. Robert Berman 1

Federal Aviation Administration DOT
Office of Aviation Safety, FOB.10A
800 Independence Ave., SW
Washington, DC 20591
Attn: Mr. John H. Enders 1

FAA, ARD-520
2100 Second Street, SW
Washington, DC 20591
Attn: Commander John J. Shea 1

National Transportation Safety Board
800 Independence Ave., SW
Washington, DC 20594
Attn: Mr. Edward P. Wizniak TE-20 1

Arizona State University
Department of Aerospace Engineering
& Engineering Science
Tempe, AZ 85281
Attn: H. D. Nelson 1

Rockwell International Corporation
Los Angeles International Airport
Los Angeles, CA 90009
Attn: Mr. Joseph Gausselin D422/402 AB71 1

Rensselaer Polytechnic Institute
Troy, NY 12181
Attn: R. Loewy 1

Cleveland State University
Department of Civil Engineering
Cleveland, OH 44115
Attn: J. J. Tomko 1

M.I.T.
Cambridge, MA 02139
Attn: K. Bathe 1
T. H. Pian 1
J. Mar 1
E. A. Witmer 1
J. Dugundji 1

University of Illinois at Chicago Circle
Department of Materials Engineering
Box 4348
Chicago, IL 60680
Attn: Dr. Robert L. Spilker 1

Detroit Diesel Allison
General Motors Corporation
Speed Code T3, Box 894
Indianapolis, IN 46206
Attn: Mr. William Springer 1
Mr. J. Byrd 1

General Motors Corporation Warren, MI 48090 Attn: R. J. Trippet		1
AVCO Lycoming Division 550 South Main Street Stratford, CT 06497 Attn: Mr. Herbert Kaehler		1
Beech Aircraft Corporation, Plant 1 Wichita, KA 67201 Attn: Mr. M. K. O'Connor		1
Bell Aerospace P. O. Box 1 Buffalo, NY 14240 Attn: R. A. Gellatly		1
Boeing Aerospace Company Impact Mechanics Lab P. O. Box 3999 Seattle, WA 98124 Attn: Dr. R. J. Bristow		1
Boeing Commercial Airplane Company P. O. Box 3707 Seattle, WA 98124 Attn: Dr. Ralph B. McCormick		1
Boeing Commercial Airplane Company P. O. Box 3707 Seattle, WA 98124 Attn: Mr. David T. Powell	73-01	1
Boeing Commercial Airplane Company P. O. Box 3707 Seattle, WA 98124 Attn: Dr. John H. Gerstle		1
Boeing Company Wichita, KA 67201 Attn: Mr. C. F. Tiffany		1
McDonnell Douglas Aircraft Corporation P. O. Box 516 Lambert Field, MO 63166 Attn: Library		1
Douglas Aircraft Company 3855 Lakewood Blvd. Long Beach, CA 90846 Attn: Mr. M. A. O'Connor, Jr.	36-41	1

Garrett AiResearch Manufacturing Co.
111 S. 34th Street
P. O. Box 5217
Phoenix, AZ 85010
Attn: L. A. Matsch

1

Mr. R. Stockton
Garrett Turbine Engine Company
Rotor Integrity, 503-42
Mechanical Component Design
111 S 34th Street
P. O. Box 5217
Phoenix, AZ 85010

1

General Dynamics
P. O. Box 748
Fort Worth, TX 76101
Attn: Library

1

General Dynamics/Convair Aerospace
P. O. Box 1128
San Diego, CA 92112
Attn: Library

1

General Electric Company
Interstate 75, Bldg. 500
Cincinnati, OH 45215
Attn: Dr. L. Beitch
Dr. M. Roberts
Dr. V. Gallardo

K221

1

K221

1

K221

1

General Electric Company
Aircraft Engine Group
Lynn, MA 01902
Attn: Mr. Herbert Garten

1

Grumman Aircraft Engineering Corp.
Bethpage, Long Island, NY 11714
Attn: Library
H. A. Armen

1

1

IIT Research Institute
Technology Center
Chicago, IL 60616
Attn: Library

1

Lockheed California Company
P. O. Box 551
Dept. 73-31, Bldg. 90, PL. A-1
Burbank, CA 91520
Attn: Mr. D. T. Pland

1

Lockheed California Company
P. O. Box 551
Dept. 75-71; Bldg. 63, PL. A-1
Burbank, CA 91520
Attn: Mr. Jack E. Wignot

1

Northrop Space Laboratories
3401 West Broadway
Hawthorne, CA 90250
Attn: Library

1

North American Rockwell, Inc.
Rocketdyne Division
6633 Canoga Avenue
Canoga Park, CA 91304
Attn: Library, Dept. 596-306

1

North American Rockwell, Inc.
Space & Information Systems Div.
12214 Lakewood Blvd.
Downey, CA 90241
Attn: Library

1

Norton Company
Industrial Ceramics Div.
Armored & Spectramic Products
Worcester, MA 01606
Attn: Mr. George E. Buron

1

Norton Company
1 New Bond Street
Industrial Ceramics Division
Worcester, MA 01606
Attn: Mr. Paul B. Gardner

1

United Aircraft Corporation
Pratt & Whitney Group
Government Products Division
P. O. Box B2691
West Palm Beach, FL 33402
Attn: Library
R. A. Marmol

1

1

United Aircraft Corporation
Pratt & Whitney Aircraft Group
400 Main Street
East Hartford, CT 06108
Attn: Library
R. Liss
D. H. Hibner

1

1

1

United Aircraft Corporation
Hamilton Standard Division
Windsor Locks, CT 06096
Attn: Dr. G. P. Townsend
Dr. R. A. Cornell

1

1

Aeronautical Research Association of
Princeton, Inc.
P. O. Box 2229
Princeton, NJ 08540
Attn: Dr. Thomas McDonough 1

Republic Aviation
Fairchild Hiller Corporation
Farmington, Long Island, NY
Attn: Library 1

Rohr Industries
Foot of H Street
Chula Vista, CA 92010
Attn: Mr. John Meaney 1

TWA, Inc.
Kansas City International Airport
P. O. Box 20126
Kansas City, MO 64195
Attn: Mr. John J. Morelli 1

Stevens Institute of Technology
Castle Point Station
Hoboken, NJ 07030
Attn: F. Sisto 1
A. T. Chang 1

Mechanical Technologies Inc.
Latham, NY
Attn: M. S. Darlow 1

Shaker Research Corporation
Northway 10, Executive Park
Ballston Lake, NY 12019
Attn: L. Lagace 1

Lockheed Palo Alto Research Labs
Palo Alto, CA 94304
Attn: B. O. Almroth 1

Lockheed Missiles and Space Company
Huntsville Research & Engineering Center
P. O. Box 1103
Huntsville, AL 35894
Attn: H. B. Shirley 1

MacNeal-Schwendler Corporation
7442 North Figueroa Street
Los Angeles, CA 90041
Attn: R. H. MacNeal 1

MARC Analysis Research Corporation
260 Sheridan Avenue, Suite 314
Palo Alto, CA 94306
Attn: P. V. Marcel 1

United Technologies Research Center
East Hartford, CT 06108
Attn: Dr. A. Dennis 1

Ohio State University
Columbus, OH 43210
Attn: A. W. Leissa 1

University of California
Mechanics & Structures Department
School of Engineering & Applied Sciences
Los Angeles, CA 90024
Attn: L. A. Schmit, Jr. 1

Columbia University
New York, NY 10027
Attn: R. Vaicaitis 1

Georgia Institute of Technology
School of Civil Engineering
Atlanta, GA 30332
Attn: S. N. Atluri 1

Georgia Institute of Technology
225 North Avenue
Atlanta, GA 30332
Attn: G. J. Simitis 1

Lawrence Livermore Laboratory
P. O. Box 808, L-421
Livermore, CA 94550
Attn: M. L. Wilkins 1

Lehigh University Institute of Fracture
and Solid Mechanics
Bethlehem, PA 18015
Attn: G. T. McAllister 1

Materials Science Corporation
1777 Walton Road
Blue Bell, PA 19422
Attn: W. B. Rosen 1

National Bureau of Standards
Engineering Mechanics Section
Washington, DC 20234
Attn: R. Mitchell 1

Purdue University School of Aeronautics & Astronautics West Lafayette, IN 47907 Attn: C. T. Sun	1
University of Dayton Research Institute Dayton, OH 45409 Attn: F. K. Bogner	1
Texas A&M University Aerospace Engineering Department College Station, TX 77843 Attn: W. E. Haisler J. M. Vance	1 1
V. P. I. and State University Department of Engineering Mechanics Blacksburg, VA 24061 Attn: R. H. Heller	1
University of Arizona College of Engineering Tucson, AZ 87521 Attn: R. H. Gallagher J. C. Heinrich	1 1
University of California Department of Civil Engineering Berkeley, CA 94720 Attn: E. Wilson	1
University of Kansas School of Engineering Lawrence, KS 66045 Attn: R. H. Dodds	1
University of Virginia School of Engineering & Applied Science Charlottesville VA 22901 Attn: E. J. Gunter	1

End of Document