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
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.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>2.0</td>
<td>INTRODUCTION</td>
<td>3</td>
</tr>
<tr>
<td>3.0</td>
<td>STAEBL PROCEDURE</td>
<td>5</td>
</tr>
<tr>
<td>3.1</td>
<td>Overview</td>
<td>5</td>
</tr>
<tr>
<td>3.2</td>
<td>General Approach to STAEBL Procedure Design</td>
<td>5</td>
</tr>
<tr>
<td>3.3</td>
<td>Input to Fan Blade Structural Tailoring Procedure</td>
<td>8</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Aerodynamic Stage</td>
<td>8</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Support Structure</td>
<td>8</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Operating Conditions</td>
<td>8</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Materials</td>
<td>9</td>
</tr>
<tr>
<td>3.3.5</td>
<td>Objective Function</td>
<td>9</td>
</tr>
<tr>
<td>3.3.6</td>
<td>Constraints</td>
<td>9</td>
</tr>
<tr>
<td>3.3.7</td>
<td>Design Variables</td>
<td>10</td>
</tr>
<tr>
<td>3.4</td>
<td>Approximate Analysis</td>
<td>10</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Stress and Vibration Analysis</td>
<td>10</td>
</tr>
<tr>
<td>3.4.1.1</td>
<td>Approach</td>
<td>10</td>
</tr>
<tr>
<td>3.4.1.2</td>
<td>Solution Procedure</td>
<td>17</td>
</tr>
<tr>
<td>3.4.1.3</td>
<td>Verification</td>
<td>23</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Foreign Object Damage Analysis</td>
<td>33</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Local Foreign Object Damage Analysis</td>
<td>34</td>
</tr>
<tr>
<td>3.4.4</td>
<td>Approximate Analysis System</td>
<td>35</td>
</tr>
<tr>
<td>3.5</td>
<td>Optimization Method</td>
<td>36</td>
</tr>
<tr>
<td>3.5.1</td>
<td>General Optimization Theory and Background</td>
<td>39</td>
</tr>
<tr>
<td>3.5.2</td>
<td>COPES/CONMIN Exact Analysis</td>
<td>40</td>
</tr>
<tr>
<td>3.5.2.1</td>
<td>Choice of Search Parameters for COPES/CONMIN</td>
<td>44</td>
</tr>
<tr>
<td>3.5.2.2</td>
<td>Scaling of Design Variables in COPES/CONMIN</td>
<td>45</td>
</tr>
<tr>
<td>3.5.2.3</td>
<td>Number of Function Calls for COPES/CONMIN</td>
<td>50</td>
</tr>
<tr>
<td>3.5.3</td>
<td>COPES/CONMIN Approximate Analysis</td>
<td>50</td>
</tr>
<tr>
<td>3.5.4</td>
<td>COPES/CONMIN Interfaces to Vibration, Flutter, and Stress Programs</td>
<td>53</td>
</tr>
<tr>
<td>3.5.5</td>
<td>Test Cases for STAEBL Optimization Procedure</td>
<td>57</td>
</tr>
<tr>
<td>3.6</td>
<td>Refined Analysis</td>
<td>61</td>
</tr>
<tr>
<td>4.0</td>
<td>RESULTS</td>
<td>61</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>61</td>
</tr>
<tr>
<td>4.2</td>
<td>Demonstration Considerations</td>
<td>61</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Aerodynamic Design</td>
<td>61</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Objective Function</td>
<td>61</td>
</tr>
<tr>
<td>4.2.2.1</td>
<td>Engine Weight</td>
<td>63</td>
</tr>
<tr>
<td>4.2.2.2</td>
<td>Engine Cost</td>
<td>65</td>
</tr>
<tr>
<td>4.2.2.3</td>
<td>Maintenance Cost</td>
<td>65</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Constraints on Behavior Variables</td>
<td>65</td>
</tr>
<tr>
<td>4.2.3.1</td>
<td>High Frequency Fatigue</td>
<td>65</td>
</tr>
<tr>
<td>4.2.3.2</td>
<td>Flight Cycle Fatigue</td>
<td>66</td>
</tr>
<tr>
<td>4.2.3.3</td>
<td>Bird Ingestion</td>
<td>67</td>
</tr>
<tr>
<td>4.3</td>
<td>STAEBL Procedure Demonstration</td>
<td>67</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Reference Blade</td>
<td>67</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Tailored Blades</td>
<td>69</td>
</tr>
<tr>
<td>4.3.2.1</td>
<td>Hollow Blade with Composite Inlay</td>
<td>69</td>
</tr>
<tr>
<td>4.3.2.2</td>
<td>Superhybrid Composite Blade</td>
<td>72</td>
</tr>
<tr>
<td>5.0</td>
<td>CONCLUSIONS AND RECOMMENDATIONS</td>
<td>77</td>
</tr>
<tr>
<td>APPENDIX A</td>
<td>BLADE OPTIMIZATION PROGRAM ORGANIZATION</td>
<td>79</td>
</tr>
<tr>
<td>APPENDIX B</td>
<td>FORTRAN LISTING FOR HOLLOW BLADE TAILORING</td>
<td>85</td>
</tr>
<tr>
<td>APPENDIX C</td>
<td>FORTRAN LISTING REVISIONS FOR TAILORING SUPERHYBRID BLADE</td>
<td>329</td>
</tr>
<tr>
<td>REFERENCES</td>
<td></td>
<td>339</td>
</tr>
<tr>
<td>DISTRIBUTION LIST</td>
<td></td>
<td>341</td>
</tr>
<tr>
<td>Figure</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>3.2-1</td>
<td>The Structural Tailoring of Engine Blades Procedure</td>
<td>6</td>
</tr>
<tr>
<td>3.2-2</td>
<td>The Objective Function Relates Airline Economics to Blade Design Variables</td>
<td>6</td>
</tr>
<tr>
<td>3.2-3</td>
<td>Blade Chord Optimization Appears to be a Simple Design Problem</td>
<td>7</td>
</tr>
<tr>
<td>3.2-4</td>
<td>Design Problem Complexity is Introduced by Structural Constraints</td>
<td>7</td>
</tr>
<tr>
<td>3.4-1</td>
<td>Model Used in Approximate Analyses of a Spanwise Increment of Blade</td>
<td>12</td>
</tr>
<tr>
<td>3.4-2</td>
<td>Beam Model Verification with NASTRAN Vibration Analysis of Assembly of Rectangular, Isotropic, Cantilevered Beams</td>
<td>12</td>
</tr>
<tr>
<td>3.4-3</td>
<td>Torsional Frequencies Obtained with Planes Remain Plane Section Constraint</td>
<td>13</td>
</tr>
<tr>
<td>3.4-4</td>
<td>Low Torsional Frequencies Produced by the Free Warping Condition</td>
<td>14</td>
</tr>
<tr>
<td>3.4-5</td>
<td>Constrained Warping Condition and Thin Section Torsional Stiffness Correlated Well Using an Approximate Section Warping Function</td>
<td>15</td>
</tr>
<tr>
<td>3.4-6</td>
<td>Results Obtained for a Twisted Flat Plate</td>
<td>16</td>
</tr>
<tr>
<td>3.4-7</td>
<td>Radially Stacked Beam Element as Represented by the Laminated Composite Beam Theory</td>
<td>17</td>
</tr>
<tr>
<td>3.4-8</td>
<td>Shape Functions for the Element Displacement Response</td>
<td>18</td>
</tr>
<tr>
<td>3.4-9</td>
<td>Rectangular Beam Ply Layup</td>
<td>20</td>
</tr>
<tr>
<td>3.4-10</td>
<td>Tilted Flat Plate Used in Approximate Analysis Test Case</td>
<td>25</td>
</tr>
<tr>
<td>3.4-11</td>
<td>Twisted Flat Plate Used in Approximate Analysis Test Case</td>
<td>26</td>
</tr>
<tr>
<td>3.4-12</td>
<td>Airfoil Warping Defined by Spanwise Twist Gradient</td>
<td>29</td>
</tr>
<tr>
<td>3.4-13</td>
<td>Approximate Root Centrifugal Stress in the Energy Efficient Engine Fan Blade Compared with NASTRAN</td>
<td>30</td>
</tr>
<tr>
<td>3.4-14</td>
<td>NASTRAN Hollow Blade Model</td>
<td>31</td>
</tr>
</tbody>
</table>
**LIST OF ILLUSTRATIONS (Continued)**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4-15</td>
<td>Hollow Airfoil Beam</td>
<td>31</td>
</tr>
<tr>
<td>3.4-16</td>
<td>Laminated Airfoil Beam</td>
<td>32</td>
</tr>
<tr>
<td>3.5-1</td>
<td>Feasible Region Is Union of All Points that Satisfy All Constraints</td>
<td>37</td>
</tr>
<tr>
<td>3.5-2</td>
<td>Line Search Terminates Either at Minimum of Objective Function or at a Constant Boundary. Sequence of line searches converge to $x_{opt}$.</td>
<td>40</td>
</tr>
<tr>
<td>3.5-3</td>
<td>New Search Direction, $s_i$, Lies in the Usable Feasible Sector. The value of the push-off factor, $\theta_j$, determines the orientation of the new search direction.</td>
<td>41</td>
</tr>
<tr>
<td>3.5-4</td>
<td>Constraint Thickness Parameter, CT, Determines when a Constraint is Satisfied, Violated, or Active</td>
<td>43</td>
</tr>
<tr>
<td>3.5-5</td>
<td>For Proper Choice of CT, Two Constraints Become Simultaneously Active so that Search Proceeds down the &quot;Valley&quot; Formed by the Constraints</td>
<td>43</td>
</tr>
<tr>
<td>3.5-6</td>
<td>Sequence of Approximate Analyses Using the Method of Feasible Directions Becomes Increasingly More Accurate as More Terms in the Taylor Series Are Determined.</td>
<td>48</td>
</tr>
<tr>
<td>3.5-7</td>
<td>COPES/CONMIN is Linked via Subroutine ANALIZ to Approximate Analyses</td>
<td>51</td>
</tr>
<tr>
<td>3.5-8</td>
<td>STAEBL Optimization Procedure May Not Converge to Global Optimum if Feasible Region is Disjoint</td>
<td>53</td>
</tr>
<tr>
<td>3.5-9</td>
<td>STAEBL Optimization Procedure with Three Flutter Constraints Converges to Global Optimum</td>
<td>54</td>
</tr>
<tr>
<td>3.5-10</td>
<td>Exact and Approximate Analyses for STAEBL Procedure Converge to Same Local Optimum</td>
<td>55</td>
</tr>
<tr>
<td>3.6-1</td>
<td>Refined Analysis of the Energy Efficient Engine Hollow Fan Blade</td>
<td>58</td>
</tr>
<tr>
<td>3.6-2</td>
<td>Energy Efficient Engine Hollow Fan Blade Airfoil Root Stress Predicted by Refined Analysis</td>
<td>59</td>
</tr>
<tr>
<td>3.6-3</td>
<td>Energy Efficient Engine Hollow Fan Blade Internal Surface Stress at Solid-to-Hollow Transition Predicted by Refined Analysis</td>
<td>59</td>
</tr>
<tr>
<td>Figure</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>3.6-4</td>
<td>Refined Analysis of the Energy Efficient Engine Hollow Fan Blade Supports the Conclusion that It Would Not Flutter</td>
<td>60</td>
</tr>
<tr>
<td>4.2-1</td>
<td>Fan System Weight Is a Simple Function of Individual Airfoil Weight</td>
<td>64</td>
</tr>
<tr>
<td>4.2-2</td>
<td>Full Span Hollow Fan Blade Life Is Derived from Solid Blade Experience Limits</td>
<td>66</td>
</tr>
<tr>
<td>4.3-1</td>
<td>Reference Blade Resonance Diagram</td>
<td>68</td>
</tr>
</tbody>
</table>
LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4-I</td>
<td>NASTRAN TEST CASE FOR PANEL WITH 30 DEGREE PRETWIST</td>
<td>16</td>
</tr>
<tr>
<td>3.4-II</td>
<td>COMPARISON OF STATIC ANALYSIS PROCEDURE WITH NASTRAN TEST CASES</td>
<td>24</td>
</tr>
<tr>
<td>3.4-III</td>
<td>TILTED FLAT PLATE TIP DEFLECTIONS</td>
<td>25</td>
</tr>
<tr>
<td>3.4-IV</td>
<td>TWISTED (30 Degrees) FLAT PLATE TIP DEFLECTIONS</td>
<td>26</td>
</tr>
<tr>
<td>3.4-V</td>
<td>APPROXIMATE ANALYSIS NATURAL FREQUENCIES COMPARED WITH NASTRAN</td>
<td>27</td>
</tr>
<tr>
<td>3.4-VI</td>
<td>APPROXIMATE ANALYSIS NATURAL FREQUENCIES COMPARED WITH NASTRAN</td>
<td>27</td>
</tr>
<tr>
<td>3.4-VII</td>
<td>APPROXIMATE ANALYSIS NATURAL FREQUENCIES COMPARED WITH NASTRAN</td>
<td>27</td>
</tr>
<tr>
<td>3.4-VIII</td>
<td>APPROXIMATE PROCEDURE ANALYSIS OF FAN BLADE NATURAL FREQUENCY</td>
<td>28</td>
</tr>
<tr>
<td>3.4-IX</td>
<td>HOLLOW BLADE FREQUENCY COMPARISON</td>
<td>32</td>
</tr>
<tr>
<td>3.4-X</td>
<td>COMPOSITE REINFORCED BLADE FREQUENCY COMPARISON</td>
<td>32</td>
</tr>
<tr>
<td>3.4-XI</td>
<td>COMPOSITE REINFORCED BLADE FREQUENCY COMPARISON</td>
<td>33</td>
</tr>
<tr>
<td>3.4-XII</td>
<td>APPROXIMATE ANALYSIS SYSTEM MODULE OUTPUTS</td>
<td>36</td>
</tr>
<tr>
<td>3.5-I</td>
<td>UNSCALED AND SCALED VALUES OF GRADIENT OF OBJECTIVE FUNCTION RELATIVE</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>TO THE DESIGN VARIABLES FOR HOLLOW FAN BLADE</td>
<td></td>
</tr>
<tr>
<td>3.5-II</td>
<td>FORTRAN NAMES AND GLOBAL LOCATIONS IN BLOBCM FOR DESIGN VARIAB</td>
<td>52</td>
</tr>
<tr>
<td>3.5-III</td>
<td>HOLLOW FAN BLADE HAS 3 ACTIVE CONSTRAINTS FOR FINAL DESIGN</td>
<td>56</td>
</tr>
<tr>
<td>4.2-I</td>
<td>ENERGY EFFICIENT ENGINE STUDY AIRPLANE CHARACTERISTICS</td>
<td>62</td>
</tr>
<tr>
<td>4.2-II</td>
<td>ENERGY EFFICIENT ENGINE AIRLINE ECONOMIC MODEL</td>
<td>63</td>
</tr>
<tr>
<td>4.3-I</td>
<td>HOLLOW BLADE WITH COMPOSITE INLAY</td>
<td>69</td>
</tr>
<tr>
<td>4.3-II</td>
<td>MATERIALS PROPERTIES</td>
<td>71</td>
</tr>
<tr>
<td>4.3-III</td>
<td>HOLLOW COMPOSITE BLADE WITH COMPOSITE INLAY NATURAL FREQUENCY</td>
<td>72</td>
</tr>
<tr>
<td>Table</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.3-IV</td>
<td>SUPERHYBRID COMPOSITE BLADE</td>
<td>73</td>
</tr>
<tr>
<td>4.3-V</td>
<td>SUPERHYBRID COMPOSITE BLADE NATURAL FREQUENCIES</td>
<td>75</td>
</tr>
<tr>
<td>A-I</td>
<td>APPROXIMATE ANALYSIS SUBROUTINE LIST</td>
<td>80</td>
</tr>
<tr>
<td>A-II</td>
<td>PRATT &amp; WHITNEY AIRCRAFT PROPRIETARY Routines</td>
<td>84</td>
</tr>
</tbody>
</table>
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.
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 aerelastic 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 substantive refined analyses are described in Section 3.6.

3.2 GENERAL APPROACH TO STAEBL PROCEDURE DESIGN

Figure 3.2-1 summarizes the 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
Direct operating cost plus interest

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

Even integer number of blades

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 (Controll 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:

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

- effective inner radius,
- 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 ingestion 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
Figure 3.4-3 Torsional Frequencies Obtained with Planes Remain Plane Section Constraint
The sum of beam torsional stiffness does not equal section torsional stiffness.

<table>
<thead>
<tr>
<th></th>
<th>Plate</th>
<th>Free Warping</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st F</td>
<td>90.1</td>
<td>87.6</td>
</tr>
<tr>
<td>1st T</td>
<td>524.1</td>
<td>354.5</td>
</tr>
<tr>
<td>2nd F</td>
<td>568.3</td>
<td>535.1</td>
</tr>
<tr>
<td>1st S</td>
<td>821.7</td>
<td>749.4</td>
</tr>
</tbody>
</table>

Figure 3.4-4 Low Torsional Frequencies Produced by the Free Warping Condition
The low torsional stiffnesses were resolved by introducing thin section torsional stiffness coefficients and an approximate warping function. The warping function was evaluated using:

\[ u = -y z \left( \frac{d \theta}{dx} \right) \]  

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 3.4-1 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)

<table>
<thead>
<tr>
<th></th>
<th>NASTRAN Plate Analysis</th>
<th>Connected Beams, Free Warping</th>
<th>Connected beams, Warping Imposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st Flap</td>
<td>90.1</td>
<td>87.6</td>
<td>87.6</td>
</tr>
<tr>
<td>2nd Flap</td>
<td>480.1</td>
<td>438.8</td>
<td>438.8</td>
</tr>
<tr>
<td>1st Torsion</td>
<td>535.1</td>
<td>567.9</td>
<td>585.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CYCLES PER SECOND</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLATE</td>
</tr>
<tr>
<td>1ST F</td>
</tr>
<tr>
<td>1ST T</td>
</tr>
<tr>
<td>2ND F</td>
</tr>
<tr>
<td>1ST S</td>
</tr>
</tbody>
</table>

- 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 subscripts in Figure 3.4-8 refer to neutral axis displacements.

\[
\begin{align*}
V_0 &= a_1 + a_2 x + a_3 x^2 + a_4 x^3 \\
W_0 &= a_5 + a_6 x + a_7 x^2 + a_8 x^3 \\
U_0 &= a_9 + a_{10} x \\
\Theta_0 &= a_{11} + a_{12} x
\end{align*}
\]

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:

\[
\{a\} = [\tilde{N}] \{\Delta\}.
\]

(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

\[
\{\varepsilon_0\} = [X] \{a\}.
\]

(3)

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

\[
\begin{align*}
\nu(x,y,z) &= \nu_0(x) - y \nu_{0,x}(x) - z w_{0,x}(x) + C yz \theta, x \\
\nu(x,y,z) &= \nu_0(x) - c_1 y u_{0,x}(x) - xz \theta, x \\
w(x,y,z) &= w_0(x) - c_2 z u_{0,x}(x) - xy \theta, x
\end{align*}
\]

(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

$$
\begin{bmatrix}
\sigma_x \\
\sigma_{xy} \\
\sigma_{xz}
\end{bmatrix}^K =
\begin{bmatrix}
Q_{11} & 0 & Q_{13} \\
0 & Q_{22} & 0 \\
Q_{13} & 0 & Q_{33}
\end{bmatrix}^K
\begin{bmatrix}
e_x \\
\gamma_{xy} \\
\gamma_{xz}
\end{bmatrix}^K
$$

(5)

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

$$
\begin{bmatrix}
N_x \\
N_{xy} \\
N_{xz}
\end{bmatrix} =
\sum_{k=1}^{N} \int_{A_k} \begin{bmatrix}
\sigma_x \\
\sigma_{xy} \\
\sigma_{xz}
\end{bmatrix}^k dA_k
$$

(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:

$$
\begin{bmatrix}
N_x \\
M_x \\
M_y \\
M_z
\end{bmatrix} =
\begin{bmatrix}
A_{11} & B_{13} & -B_{11} & 0 \\
B_{13} & D_{33} & -D_{13} & 0 \\
-B_{11} & -D_{13} & D_{11} & 0 \\
0 & 0 & 0 & a^2/12 A_{11}
\end{bmatrix}
\begin{bmatrix}
U_{0,x} \\
U_{0,y} \\
U_{0,z}
\end{bmatrix}
$$

(7)

where:

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

$$
A_{ij} = a \sum_{k=1}^{N} Q_{ij} K_k (Y_k - Y_{k-1})
$$

(8)

$$
B_{ij} = a/2 \sum_{k=1}^{N} Q_{ij} K_k (Y_k^2 - Y_{k-1}^2)
$$

and
\[ D_{ij} = \frac{a}{3} \sum_{K=1}^{N} Q_{ijK} (Y_{K}^3 - Y_{K-1}^3) \]

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:

\[ \{f\} = [E] \{\varepsilon_0\}. \]  \hspace{1cm} (9)

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

\[ \{F\} = [A] \{f\}. \]  \hspace{1cm} (10)

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

\[ \{F\} = [A] [E] [X] [\tilde{N}] \{\Delta\} = [K_B] \{\Delta\} \]  \hspace{1cm} (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:

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

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:

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

Reordered to transfer matrix form, the element equation becomes:

$$\begin{pmatrix} \Delta_2 \\ \Delta_2 \end{pmatrix} = \begin{bmatrix} -S_{K12}^{-1} S_{K11} & S_{K12}^{-1} \\ S_{K21} & -S_{K22} \end{bmatrix} \begin{pmatrix} \Delta_1 \\ F_1 \end{pmatrix} = [S_K] \begin{pmatrix} \Delta_1 \\ F_1 \end{pmatrix} \quad (14)$$

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:

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

Hence, in traversing from the beginning of a beam segment to its end, with
inertia effects included, it is found:
The \((\mathbf{m})\) (\(\mathbf{K}\)) product above is the final step in making the present theory compatible with the existing beam analysis. Notably, only the \((\mathbf{m})\) 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:

\[
\begin{bmatrix}
\Delta z \\
\mathbf{F}_2
\end{bmatrix} = [-\mathbf{m}] [-\mathbf{K}] \begin{bmatrix}
\Delta \mathbf{1} \\
\mathbf{F}_1
\end{bmatrix}
\tag{17}
\]

For the full blade, from station 1 to station \(n\),

\[
\begin{bmatrix}
\mathbf{U}_{i+1} \\
\mathbf{F}_{i+1}
\end{bmatrix}' = \mathbf{K}_k \begin{bmatrix}
\mathbf{U}_i \\
\mathbf{F}_i
\end{bmatrix} + \begin{bmatrix}
0 \\
\mathbf{P}_{i+1}
\end{bmatrix}
\tag{18}
\]

For the full blade, from station 1 to station \(n\),

\[
\begin{bmatrix}
\mathbf{U}_n \\
\mathbf{F}_n
\end{bmatrix}' = \begin{bmatrix}
\mathbf{K}_{11} & \mathbf{K}_{12} \\
\mathbf{K}_{12} & \mathbf{K}_{22}
\end{bmatrix} \begin{bmatrix}
\mathbf{U}_1 \\
\mathbf{F}_1
\end{bmatrix} + \begin{bmatrix}
0 \\
\mathbf{P}_T
\end{bmatrix}
\tag{19}
\]

At station 1, the blade root boundary condition,

\[
\mathbf{U}_1 = \mathbf{C} \mathbf{F}_1,
\tag{20}
\]

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

\[
\begin{bmatrix}
\mathbf{U}_n \\
\mathbf{F}_n
\end{bmatrix}' = \begin{bmatrix}
\mathbf{C} \mathbf{K}_{11} + \mathbf{K}_{12} \\
\mathbf{C} \mathbf{K}_{21} + \mathbf{K}_{22}
\end{bmatrix} \begin{bmatrix}
\mathbf{F}_1 \\
\mathbf{P}_{T1}
\end{bmatrix} + \begin{bmatrix}
\mathbf{P}_{T1} \\
\mathbf{P}_{T2}
\end{bmatrix}
\tag{21}
\]

At station \(n\), the blade tip boundary condition:

\[
\{\mathbf{F}_n\}' = 0.
\tag{22}
\]
The root load can then be determined,

\[
\{F_1\} = -[C K_{21} + K_{22}]^{-1}\{P_T\}
\]  

(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**

<table>
<thead>
<tr>
<th>Flat Plate (11 Beams/Section-8 Sections)</th>
<th>NASTRAN</th>
<th>Static Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tip Deflections</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radial, cm</td>
<td>0.4171-3</td>
<td>0.4168-3</td>
</tr>
<tr>
<td>(in)</td>
<td>(0.1642-3)</td>
<td>(0.1641-3)</td>
</tr>
<tr>
<td>Twist, (radians)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Restraint Forces</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radial, n</td>
<td>-2327.</td>
<td>-2327.</td>
</tr>
<tr>
<td>(lb)</td>
<td>(-0.5232+3)</td>
<td>(-0.5232+3)</td>
</tr>
<tr>
<td>Twist, n-m</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(in-lb)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Twisted Plate (11x8 Breakup, 30° twist)**

| Tip Deflections                          |         |                |
| Radial, cm                              | 0.4244-3| 0.4244-3       |
| (in)                                    | (0.1671-3)  | (0.1671-3)    |
| Twist, (radians)                        | 0.3828-4| 0.3827-4       |
| Restraint Forces                        |         |                |
| Radial, n                               | -2330.  | -2330.         |
| (lb)                                    | (-0.5239+3)  | (-0.5238+3)   |
| Twist, n-m                              | 3.11    | 3.11           |
| (in-lb)                                 | (0.2754+2)  | (0.2753+2)    |

24
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)

<table>
<thead>
<tr>
<th>Local Tip Deflections</th>
<th>T₁, cm (in)</th>
<th>T₂, cm (in)</th>
<th>R₃, radians</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Static Analysis</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NASTRAN</td>
<td>0.04881</td>
<td>1.9461</td>
<td>0.08524</td>
</tr>
<tr>
<td></td>
<td>(0.019217)</td>
<td>(0.76617)</td>
<td></td>
</tr>
<tr>
<td>STAEBL</td>
<td>0.04890</td>
<td>1.9463</td>
<td>0.08526</td>
</tr>
<tr>
<td></td>
<td>(0.019251)</td>
<td>(0.76627)</td>
<td></td>
</tr>
<tr>
<td><strong>Iterated Static Analysis</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NASTRAN</td>
<td>0.04870</td>
<td>0.88321</td>
<td>0.03474</td>
</tr>
<tr>
<td></td>
<td>(0.019175)</td>
<td>(0.34772)</td>
<td></td>
</tr>
<tr>
<td>STAEBL</td>
<td>0.04875</td>
<td>0.88326</td>
<td>0.34474</td>
</tr>
<tr>
<td></td>
<td>(0.019192)</td>
<td>(0.34774)</td>
<td></td>
</tr>
</tbody>
</table>
**Figure 3.4-11** Twisted Flat Plate Used in Approximate Analysis Test Case

**TABLE 3.4-IV**

<table>
<thead>
<tr>
<th>Local Tip Deflections</th>
<th>T₁, cm (in) X10⁻³</th>
<th>R₁ (radians) X10⁻⁴</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Static Analysis</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NASTRAN</td>
<td>0.4246 (0.16715)</td>
<td>0.38276</td>
</tr>
<tr>
<td>STAEBL</td>
<td>0.4245 (0.16711)</td>
<td>0.38272</td>
</tr>
<tr>
<td><strong>Iterated Static Analysis</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NASTRAN</td>
<td>0.4246 (0.16715)</td>
<td>0.38261</td>
</tr>
<tr>
<td>STAEBL</td>
<td>0.4245 (0.16711)</td>
<td>0.38264</td>
</tr>
</tbody>
</table>
### TABLE 3.4-V

**APPROXIMATE ANALYSIS NATURAL FREQUENCIES COMPARED WITH NASTRAN**  
(Flat Plate)

<table>
<thead>
<tr>
<th>Mode</th>
<th>Cycles per second</th>
<th>Percent Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NASTRAN</td>
<td>STAEBL</td>
</tr>
<tr>
<td>First flap</td>
<td>90</td>
<td>88.5</td>
</tr>
<tr>
<td>First torsion</td>
<td>524</td>
<td>529.7</td>
</tr>
<tr>
<td>Second flap</td>
<td>558</td>
<td>552.3</td>
</tr>
<tr>
<td>First stiff</td>
<td>822</td>
<td>822.5</td>
</tr>
</tbody>
</table>

### TABLE 3.4-VI

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

<table>
<thead>
<tr>
<th>Mode</th>
<th>Cycles per second</th>
<th>Percent Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NASTRAN</td>
<td>STAEBL</td>
</tr>
<tr>
<td>First flap</td>
<td>90</td>
<td>88.5</td>
</tr>
<tr>
<td>Second flap</td>
<td>480</td>
<td>474.0</td>
</tr>
<tr>
<td>First torsion</td>
<td>536</td>
<td>561.5</td>
</tr>
<tr>
<td>First stiff</td>
<td>941</td>
<td>945.0</td>
</tr>
</tbody>
</table>

### TABLE 3.4-VII

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

<table>
<thead>
<tr>
<th>Mode</th>
<th>Cycles per second</th>
<th>Percent Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NASTRAN</td>
<td>STAEBL</td>
</tr>
<tr>
<td>First flap</td>
<td>65.6</td>
<td>65.0</td>
</tr>
<tr>
<td>Second flap</td>
<td>404.8</td>
<td>405.5</td>
</tr>
<tr>
<td>First torsion</td>
<td>442.5</td>
<td>460.0</td>
</tr>
<tr>
<td>First stiff</td>
<td>706.4</td>
<td>705.5</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Mode</th>
<th>Cycles per second</th>
<th>Percent Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>First flap</td>
<td>92.9</td>
<td>93.6</td>
</tr>
<tr>
<td>First torsion</td>
<td>185.3</td>
<td>163.2</td>
</tr>
<tr>
<td>Second flap</td>
<td>210.2</td>
<td>205.7</td>
</tr>
<tr>
<td>First stiff</td>
<td>370.9</td>
<td>349.0</td>
</tr>
</tbody>
</table>

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:

\[
\frac{\text{twist gradient}}{\text{twist moment}} = \alpha = 1 - e^{-x/A}
\]  

\[x = \text{distance from root}\]

\[A = \text{characteristic length associated with warping restraint}\]

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 \(F_1\sigma_1 + F_2\sigma_2 + F_6\sigma_6 + F_{11}\sigma_{11}^2 + F_{22}\sigma_{22}^2 + F_{66}\sigma_{66}^2 + 2F_{12}\sigma_1\sigma_2\) where \(\sigma_1\) and \(\sigma_2\) are the principal tensile (or compressive) stresses and \(\sigma_6\) 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:

\[
K = 4G \int \int_{-t/2}^{t/2+\delta} y^2 dy dz + 4G \int \int_{-t/2}^{t/2-\delta} y^2 dy dz
\]  
(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.

\[u = (\bar{W}_s - W_s) \Theta\]

\[\bar{W}_s = \frac{1}{\text{AREA}} \int_o^S t W_s ds\]

\[W_s = \int_o^S R_T ds\]

\[\Theta = \text{TWIST GRADIENT}\]

\[R_2 = -R_N \Theta\]

\[R_3 = -R_T \Theta\]

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**

<table>
<thead>
<tr>
<th>Mode</th>
<th>NASTRAN (cps)</th>
<th>Approximate Analysis (cps)</th>
<th>Percent Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Frequency</td>
<td>110.7</td>
<td>113.5</td>
<td>+2.5</td>
</tr>
<tr>
<td>Second Frequency</td>
<td>267.4</td>
<td>278.8</td>
<td>+4.3</td>
</tr>
<tr>
<td>First Torsion</td>
<td>289.8</td>
<td>295.8</td>
<td>+2.1</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Mode</th>
<th>NASTRAN (cps)</th>
<th>Approximate Analysis (cps)</th>
<th>Percent Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Bending</td>
<td>108.4</td>
<td>111.8</td>
<td>3.1</td>
</tr>
<tr>
<td>Second Bending</td>
<td>261.8</td>
<td>276.2</td>
<td>5.5</td>
</tr>
<tr>
<td>First Torsion</td>
<td>298.8</td>
<td>312.9</td>
<td>4.7</td>
</tr>
</tbody>
</table>
TABLE 3.4-XI
COMPOSITE REINFORCED BLADE FREQUENCY COMPARISON
(35° Fiber Orientation)

<table>
<thead>
<tr>
<th>Mode</th>
<th>NASTRAN (cps)</th>
<th>Approximate Analysis (cps)</th>
<th>Percent Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Bending</td>
<td>108.4</td>
<td>111.1</td>
<td>2.5</td>
</tr>
<tr>
<td>Second Bending</td>
<td>259.6</td>
<td>270.0</td>
<td>4.0</td>
</tr>
<tr>
<td>First Torsion</td>
<td>295.4</td>
<td>298.8</td>
<td>1.2</td>
</tr>
</tbody>
</table>

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:

\[ [m] \{ \ddot{x} \} + [k] \{ x \} = \{ F(t) \} \] (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:

\[ \dddot{\xi}_k + \omega_k^2 \ddot{\xi}_k = \frac{1}{m_k} P_k(t) \] (27)

where \( \ddot{\xi}_k(t) \) is the modal amplification factor of the \( k \)'th mode, \( \omega_k \) 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:

\[ \ddot{\dot{\xi}}_k(t) = \frac{I_k}{m_k \omega_k^2} \sin(\omega_k t) \] (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:

\[ M_{\text{max}} = \frac{1}{4} \sigma Y_{\text{TI}} h^2 \]  

(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:

\[ M_{\text{max}} = \sum_{i=1}^{n} \sigma Y_i \bar{\xi}_i \Delta \xi_i \]  

(30)

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

\[ \frac{4}{\sigma Y_{\text{TI}} h^2} \sum_{i=1}^{n} \sigma Y_i \bar{\xi}_i \Delta \xi_i \]  

(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 ($M_1 > 1$), and thus provide a wide range of application.

The basic assumptions of the analysis are the following:

- Flow is two-dimensional, unsteady, compressible, inviscid, irrotational and isentropic;
- Cascade is infinite, flat plate, at zero incidence and unstalled;
- 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

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

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:

\[
\text{minimize } f(x),
\]

subject to the auxiliary conditions,

\[
g_i(x) \leq 0, \ i=1, \ldots, m.
\]

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

\[
L_i \leq x_i \leq U_i, \ i=1, \ldots, n,
\]

are referred to as side constraints. The \( n \)-dimensional space spanned by the design variables is design space. If \( f(x) \) and \( g_i(x) \), \( i=1, \ldots, m \), are all
linear functions of \( x \), 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 \( f(x) \) or any of the \( g_i(x) \)'s are nonlinear, then it is a nonlinear programming (NP) problem for which a number of solution techniques are also available. If the objective function, \( f(x) \), is to be maximized, then the equivalent problem of minimizing \(-f(x)\) is considered.

Any choice of variables, \( x \), 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 \( g_i(x) = 0 \), for some \( i \), forms a constraint surface. On one side of the surface, \( g_i(x) < 0 \) and the constraint is satisfied; on the other side, \( g_i(x) > 0 \) 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., \( g_i(x) = 0 \) 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](image)
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,

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

where:

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

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, \( x \), must all satisfy the side constraints (34).

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

\[ R_1 < R_2 < R_3 < \ldots \rightarrow \infty. \]  

Under rather mild conditions, the sequence of minima \( \{x_1, x_2, x_3, \ldots\} \) 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 Engineering 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 \((x_0, x_1, \ldots)\) 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.,

\[
x_{i+1} = x_i + \alpha s_i,
\]

for \(i = 0, 1, 2, \ldots\), where \(s_i\) is the search direction and \(\alpha\) 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 \(\alpha\) 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.,

\[
s_i \cdot \nabla f(x_i) < 0. \tag{39}
\]
Figure 3.5-2 Line Search Terminates Either at Minimum of Objective Function or at a Constant Boundary. Sequence of line searches converge to $X_{opt}$.

In addition, $s_j$ is feasible if no active constraints are initially violated along this path, i.e.,

$$s_j \cdot \bar{v}g_j(x_j) \leq 0, \ j = 1, \ldots \ NAC,$$

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

3.5.2.1 Choice of Search Parameters for COPES/CONMIN

In Zontendijk's method, the search direction, $s_j$, is determined by solving a sub-optimization problem, i.e.,

$$\text{maximize } \beta,$$

subject to:

$$s_j \cdot \bar{v}f(x_j) + \beta \leq 0,$$

$$s_j \cdot \bar{v}g_j(x_j) + \theta_j \beta \leq 0, \ j = 1, \ldots, NAC$$

where $|s_j|$ is bounded. The parameter $\theta_j$, the push-off factor, determines the orientation of the new search direction vector, $s_j$, 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 $\theta_j \to 0$, and $s_i$ approaches a level curve to the objective function tangentially as $\theta_j \to \infty$. For a linear constraint, $\theta_j$ can be set to zero and the search can proceed along that particular constraint surface. If $\theta_j$ 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 $\theta_j$ 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.

![Design Space Diagram](image)

**Figure 3.5-3** New Search Direction, $s_i$, Lies in the Usable Feasible Sector. The value of the push-off factor, $\theta_j$, 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
The $i$th constraint to be satisfied if $g_i(x) \leq CT$ and to be active if $|g_i(x)| \leq -CT$, 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 $\theta_j$ 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, $\theta_j$, 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, $\theta_j$ is a quadratic function of these parameters, i.e.,

$$\theta_j = \theta_0 \left( \frac{g_i(x_i)}{CT} - 1 \right)^2,$$

where $\theta_0$ is the initial value of $\theta_j$ (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) x 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, \( f(x) \); whereas a 1 percent change in \( x_2 \) leads to only a 0.1 percent change in \( f(x) \). 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.;

\[
\xi_i = \frac{x_i}{x_i^0}
\]  

(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, \( \xi_i \), is given by:

\[
\frac{\partial f}{\partial \xi_i} = \frac{\partial x_i}{\partial \xi_i} \frac{\partial f}{\partial x_i} = x_i^0 \frac{\partial f}{\partial x_i}
\]  

(44)

Thus, the ratio of the \( i \)th components in the scaled gradients is given by:

\[
\frac{\partial f}{\partial \xi_i} / \frac{\partial f}{\partial x_i} = x_i^0
\]  

(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, \( t_{\text{root}} \) and \( t_{\text{tip}} \), at root and tip, respectively); \( x_3 \) (distance, \( d_{\text{root}} \), from bottom of hollow section to the blade root); and \( x_4 \) (titanium skin thickness, \( t_{\text{Ti}} \)). The objective function, \( f(x) \), was blade weight. As shown in Table 3.5-1, 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

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

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

- $\{ > 0 : \text{Rescale design variables by dividing by current values every NSCALE iteration,} \}$
- $\{ NSCALE = 0 : \text{No scaling (default value),} \}$
- $\{ < 0 : \text{Scale design variables by dividing by user-input scaling variables} \}$

For STAEBL demonstration, scaling was always used. The value $NSCALE = n+1$ (where $n = \text{number of design variables}$) 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:

- Small: $n \leq 10$
- Moderate: $10 < n \leq 50$
- Large: $n > 50$.  

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 \( N=10n + 30 \). 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:

\[
f(x) \approx f(x_0) + \nabla f(x_0) \cdot (x-x_0) + \frac{1}{2} (x-x_0) \cdot H(x_0) \cdot (x-x_0), \quad (49)
\]

where \( x = (x_1, x_2, \ldots, x_n)^T \) is a vector of \( n \) design variables, \( x_0 \) 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, \( \nabla f(x_0) \), and Hessian matrix, \( H(x_0) \), are given by:
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:

\[
N_Q = 1 + n + n(n+1)/2.
\]

(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, \( x_0 \), (the nominal design) is used to determine the constant term, \( f(x_0) \), 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 \( N_0 \) 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 $N_Q$ 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 $N_Q$ 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 $3N_Q$ 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 $N_Q = 1 + n + n(n+1)/2$ unknowns. Equating $N$ and $N_Q$, 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 $N_Q$ 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:

$$N_Q = 1 + n + n,$$  \hspace{1cm} (52)

so that $N_Q$ 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:

$$N_L = 1 + n,$$  \hspace{1cm} (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.

COPES/CONMIN

ICALC = 1

READ INPUT AND SET PARAMETERS

SUBROUTINE ANALIZ

ICALC = 3

PRINT RESULTS

ICALC = 2

APPROXIMATE STRESS ANALYSIS

APPROXIMATE VIBRATION ANALYSIS

APPROXIMATE FLUTTER ANALYSIS

APPROXIMATE FOD ANALYSIS

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

<table>
<thead>
<tr>
<th>Global Location</th>
<th>FORTRAN Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>OBJF</td>
<td>Airfoil weight</td>
</tr>
<tr>
<td>2-6</td>
<td>FN</td>
<td>Frequencies (Root 1 - Root 5)</td>
</tr>
<tr>
<td>7-11</td>
<td>DLAR</td>
<td>Aerodynamic damping parameter (Root 1 - Root 5)</td>
</tr>
<tr>
<td>12-32</td>
<td>THKVAL</td>
<td>Airfoil thicknesses (max. of 21 stations)</td>
</tr>
<tr>
<td>33-37</td>
<td>RF</td>
<td>Resonance margin for 1st order (Root 1 - Root 5)</td>
</tr>
<tr>
<td>38-42</td>
<td>RF</td>
<td>Resonance margin for 2nd order (Root 1 - Root 5)</td>
</tr>
<tr>
<td>43-47</td>
<td>RF</td>
<td>Resonance margin for 3rd order (Root 1 - Root 5)</td>
</tr>
<tr>
<td>48-52</td>
<td>RF</td>
<td>Resonance margin for 4th order (Root 1 - Root 5)</td>
</tr>
<tr>
<td>53</td>
<td>BRCC</td>
<td>Root Chord</td>
</tr>
<tr>
<td>54-74</td>
<td>FODLSB</td>
<td>LSBIP = Local stress bird ingestion parameter</td>
</tr>
<tr>
<td>75</td>
<td>DLE</td>
<td>Distance to hole from leading edge</td>
</tr>
<tr>
<td>76</td>
<td>DTE</td>
<td>Distance to hole from trailing edge</td>
</tr>
<tr>
<td>77</td>
<td>DROOT</td>
<td>Distance to hole from root edge</td>
</tr>
<tr>
<td>78</td>
<td>DTIP</td>
<td>Distance to hole from tip edge</td>
</tr>
<tr>
<td>79</td>
<td>TTI</td>
<td>Thickness of titanium</td>
</tr>
<tr>
<td>80</td>
<td>TLT</td>
<td>Thickness of borsic titanium</td>
</tr>
<tr>
<td>81</td>
<td>OBJFUN</td>
<td>Objective function</td>
</tr>
<tr>
<td>82-89</td>
<td>SMAXTS</td>
<td>Root Tsai-Wu (layer 1-8)</td>
</tr>
<tr>
<td>90-89</td>
<td>SMAX2S</td>
<td>Hole Tsai-Wu (layer 1-8)</td>
</tr>
<tr>
<td>98-105</td>
<td>SMAx3S</td>
<td>Spanwise bending ingestion Tsai-Wu (layer 1-8)</td>
</tr>
<tr>
<td>106-113</td>
<td>SMAXLS</td>
<td>Leading edge Tsai-Wu (layer 1-8)</td>
</tr>
<tr>
<td>114-121</td>
<td>SMAXTS</td>
<td>Trailing edge Tsai-Wu (layer 1-8)</td>
</tr>
<tr>
<td>122-128</td>
<td>Theta</td>
<td>Fiber direction (7 layers)</td>
</tr>
<tr>
<td>129</td>
<td>HLRTIO</td>
<td>Hole to length ratio</td>
</tr>
<tr>
<td>130</td>
<td>ECRTO</td>
<td>Edge to chord ratioi</td>
</tr>
<tr>
<td>131-151</td>
<td>TOVB</td>
<td>t/b's</td>
</tr>
<tr>
<td>152</td>
<td>FODMAX</td>
<td>Max LSBIP</td>
</tr>
<tr>
<td>153</td>
<td>TIS</td>
<td>Titanium skin thickness</td>
</tr>
<tr>
<td>154</td>
<td>TIC</td>
<td>Titanium center thickness</td>
</tr>
<tr>
<td>155</td>
<td>PCBA</td>
<td>% boron aluminum</td>
</tr>
<tr>
<td>156</td>
<td>VAL</td>
<td>2/TIS + TIC)/THKVAL(1)</td>
</tr>
</tbody>
</table>
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:

\[
\begin{align*}
0.25 < t_r &< 0.5 \\
0.1 < t_t &< 0.3,
\end{align*}
\]

\[
\begin{align*}
130 < \omega_1 &< 173 \\
272 < \omega_2 &< 320 \\
390 < \omega_3 &< 504.
\end{align*}
\]

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 \(\omega_3\), 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](image-url)
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](image)
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.

<table>
<thead>
<tr>
<th></th>
<th>FUNC. CALLS (SEC.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXACT</td>
<td>19 139</td>
</tr>
<tr>
<td>APPROX.</td>
<td>6 48</td>
</tr>
</tbody>
</table>

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:

\[
\begin{align*}
  t_r & \geq 0 & \delta_i & \geq 0 & \text{Root stress} & \leq \text{max.} \\
  t_t & \geq 0 & 0.025 \leq t_r/b \leq 0.1 & \text{Hole stress} & \leq \text{max.} \\
  d_r & \geq 2.0 & 0.025 \leq t_r/b \leq 0.1 & \text{Leading edge stress} & \leq \text{max.} \\
  t_{Ti} & \geq 0.03 & \text{LSBIP} & \leq \text{max.} & \text{Trailing edge stress} & \leq \text{max.}
\end{align*}
\]

(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, LSBIP = Local Stress Bird Ingestion Parameter, and BBIP = 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

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

*Active Constraints Notes

1: \(\delta_i\) (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:

- an airfoil pre-processor which generates a NASTRAN plate model of a blade from the airfoil coordinate descriptions,
- 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,
- a NASTRAN module to calculate laminate strains from element stresses,
- 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.

![NASTRAN Plate Analysis](image.png)

**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

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

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**

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

The resulting function is:

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

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, hipped and isothermally forged to shape. The superhybrid blade is made from plies of graphite-epoxy, boron-aluminum, and titanium with adhesive 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 inject 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.

![Graph showing leading edge to cavity distance in centimeters vs. scrap life in hours, with solid titanium experience limits marked]

**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 equal 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.

![Reference Blade Resonance Diagram](image)

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 |

<table>
<thead>
<tr>
<th>Design Variables</th>
<th>Reference Blade Limits</th>
<th>First Tailoring</th>
<th>Second Tailoring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root Chord - cm (inches)</td>
<td>23.2 (9.12)</td>
<td>16.9 (6.64)</td>
<td>19.0 (7.46)</td>
</tr>
<tr>
<td>Thickness/Chord</td>
<td>Root 0.096</td>
<td>0.077 0.084</td>
<td></td>
</tr>
<tr>
<td></td>
<td>25% 0.075 0.025 ≤ 0.089 0.092</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>50% 0.055 t/b ≤ 0.100 0.084 0.045</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>75% 0.033</td>
<td>0.038 0.025</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tip 0.025</td>
<td>0.033 0.028</td>
<td></td>
</tr>
<tr>
<td>Cavity Boundaries - cm (inches)</td>
<td>From Leading Edge 2.20 (1.00) 1.27 (0.5) 3.12 (1.23) 1.37 (0.54)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>From Trailing Edge 2.20 (1.00) 1.27 (0.5) 2.48 (0.98) 3.48 (1.37)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>From Root 25.2 (9.90) 5.46 (2.15) 12.97 (5.10) 7.52 (2.96)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>From Tip 0.635 (0.25) 0.635 (0.25) 0.635 (0.25) 0.635 (0.25)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TABLE 4.3-I (Continued)

<table>
<thead>
<tr>
<th>Reference Blade Limits</th>
<th>First Tailoring</th>
<th>Second Tailoring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cavity Wall - cm (inches)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Titanium Skin</td>
<td>0.305 (0.120)</td>
<td>0.076 (0.030)</td>
</tr>
<tr>
<td></td>
<td>≥ 0.076 (0.030)</td>
<td>0.076 (0.030)</td>
</tr>
<tr>
<td>Composite</td>
<td>-</td>
<td>0.086 (0.034)</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>0.160 (0.063)</td>
</tr>
<tr>
<td>Borsic Fiber Angle - rad</td>
<td>-</td>
<td>-0.012 -0.013</td>
</tr>
<tr>
<td>Blade Weight- kilograms (pounds)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8.7 (19.2)</td>
<td>3.32 (7.3)</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>4.23 (9.3)</td>
</tr>
<tr>
<td>Objective Function - Δ(ΔDOC+I)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engine Weight</td>
<td>-0.42</td>
<td>-0.33</td>
</tr>
<tr>
<td>Engine Cost</td>
<td>-0.14</td>
<td>-0.15</td>
</tr>
<tr>
<td>Maintenance Cost</td>
<td>-0.09</td>
<td>+0.03</td>
</tr>
<tr>
<td>Total</td>
<td>-0.65</td>
<td>-0.45</td>
</tr>
</tbody>
</table>

Constraints

| Resonance Margin - Δfn/fn |                  |                  |
| 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)

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

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)

<table>
<thead>
<tr>
<th>Mode</th>
<th>First Tailoring Approximate Analysis</th>
<th>First Tailoring Refined Analysis</th>
<th>Second Tailoring Approximate Analysis*</th>
<th>Second Tailoring Refined Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>88</td>
<td>106</td>
<td>94</td>
</tr>
<tr>
<td>2</td>
<td>187</td>
<td>208</td>
<td>205</td>
<td>208</td>
</tr>
<tr>
<td>3</td>
<td>300</td>
<td>287</td>
<td>312</td>
<td>295</td>
</tr>
</tbody>
</table>

*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 4.3-11.

**TABLE 4.3-IV**

**SUPERHYBRID COMPOSITE BLADE**

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

73
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>
<thead>
<tr>
<th>Table 4.3-IV (Continued)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>Flutter - Log Decrement</td>
</tr>
<tr>
<td>First Mode</td>
</tr>
<tr>
<td>Second Mode</td>
</tr>
<tr>
<td>Third Mode</td>
</tr>
<tr>
<td>Bird Ingestion</td>
</tr>
<tr>
<td>Local Stress Parameter</td>
</tr>
<tr>
<td>Root Tsai-Wu</td>
</tr>
<tr>
<td>Steady Stress - Tasi-Wu</td>
</tr>
<tr>
<td>Root Edge</td>
</tr>
<tr>
<td>Root Local</td>
</tr>
</tbody>
</table>
TABLE 4.3-V
SUPERHYBRID COMPOSITE BLADE NATURAL FREQUENCIES (Hertz)

<table>
<thead>
<tr>
<th>Mode</th>
<th>First Tailoring</th>
<th>Second Tailoring</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Approximate Analysis</td>
<td>Refined Analysis</td>
</tr>
<tr>
<td>1</td>
<td>106</td>
<td>104</td>
</tr>
<tr>
<td>2</td>
<td>229</td>
<td>218</td>
</tr>
<tr>
<td>3</td>
<td>302</td>
<td>289</td>
</tr>
</tbody>
</table>

*Uncorrected
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.
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>
<thead>
<tr>
<th>Routine</th>
<th>External References</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Program</td>
<td>MYTIME COPE05</td>
<td>Main Program for COPES/CONMIN Constrained Optimization Program</td>
</tr>
<tr>
<td></td>
<td>COPE01 COPE06</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ANALIZ COPE07</td>
<td></td>
</tr>
<tr>
<td></td>
<td>COPE02 COPE09</td>
<td></td>
</tr>
<tr>
<td></td>
<td>COPE18 COPE14</td>
<td></td>
</tr>
<tr>
<td></td>
<td>COPE04</td>
<td></td>
</tr>
<tr>
<td>ANALIZ</td>
<td>MYTIME MNW137</td>
<td>Constraint Calculation for COPES/CONMIN</td>
</tr>
<tr>
<td></td>
<td>RDW137 RSSFUNC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RDDATA OBJTV</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TMAX ITT983</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WTW137 WTT983</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CALCTH</td>
<td></td>
</tr>
<tr>
<td>BBIP</td>
<td>TSAIWV MATPRN</td>
<td>Multi-Mode Root Bird Ingestion Stress</td>
</tr>
<tr>
<td>BCOORD</td>
<td>MATPRN</td>
<td>Beam Local Coordinate Systems</td>
</tr>
<tr>
<td>BEAM</td>
<td>MATPRN MATMPY</td>
<td>Beam Stiffness Calculation</td>
</tr>
<tr>
<td>BIRDF</td>
<td>MATPRN</td>
<td>Bird Impulse Calculation</td>
</tr>
<tr>
<td>CALCTH</td>
<td>MATPRN</td>
<td>Input Blade Thickness Calculation</td>
</tr>
<tr>
<td>CNMN01</td>
<td>CNM04</td>
<td>Gradient Information</td>
</tr>
<tr>
<td>CNMN02</td>
<td>CNMN03</td>
<td>Descent Vector</td>
</tr>
<tr>
<td>CNMN03</td>
<td>CNMN04</td>
<td>One-Dimensional Search</td>
</tr>
<tr>
<td>CNMN04</td>
<td>CNMN05</td>
<td>One-Dimensional Function Minimum</td>
</tr>
<tr>
<td>CNMN05</td>
<td>CNMN07 CNMN04</td>
<td>Search Direction</td>
</tr>
<tr>
<td>CNMN06</td>
<td>CNMN07 CNMN05 CNMN08</td>
<td>One-Dimensional Search</td>
</tr>
<tr>
<td>CNMN07</td>
<td>CNMN08</td>
<td>Zero of I-D Function</td>
</tr>
<tr>
<td>CNMN08</td>
<td>CNMN01 CNMN02 CNMN03</td>
<td>Special Linear Optimization</td>
</tr>
<tr>
<td>CONMIN</td>
<td>CNMN03 CNMN04 CNMN06</td>
<td>Constrained or Unconstrained Function Minimization</td>
</tr>
<tr>
<td>Routine</td>
<td>External References</td>
<td>Function</td>
</tr>
<tr>
<td>---------</td>
<td>---------------------</td>
<td>----------</td>
</tr>
<tr>
<td>COPE01</td>
<td>COPE08</td>
<td>Control Input for COPES</td>
</tr>
<tr>
<td>COPE02</td>
<td>CONMIN COPE03</td>
<td>Control Optimization</td>
</tr>
<tr>
<td>COPE03</td>
<td>ANALIZ</td>
<td>Buffer Between CONMIN and COPES</td>
</tr>
<tr>
<td>COPE04</td>
<td>ANALIZ COPE02</td>
<td>Sensitivity Information</td>
</tr>
<tr>
<td>COPE05</td>
<td></td>
<td>Print Sensitivity Information</td>
</tr>
<tr>
<td>COPE06</td>
<td></td>
<td>Two Variable Function Space</td>
</tr>
<tr>
<td>COPE07</td>
<td></td>
<td>Print Two Variable Function Space</td>
</tr>
<tr>
<td>COPE08</td>
<td></td>
<td>Convert Data Input</td>
</tr>
<tr>
<td>COPE09</td>
<td>COPE10 COPE12 COPE15</td>
<td>Approximate Optimization</td>
</tr>
<tr>
<td>COPE10</td>
<td>COPE11</td>
<td>Set-Up Taylor Expansion</td>
</tr>
<tr>
<td>COPE11</td>
<td></td>
<td>Nominal Design for Approx. Optimization</td>
</tr>
<tr>
<td>COPE12</td>
<td>COPE13</td>
<td>Least Squares Fit</td>
</tr>
<tr>
<td>COPE13</td>
<td></td>
<td>Least Squares Fit Evaluation</td>
</tr>
<tr>
<td>COPE14</td>
<td></td>
<td>Print Approximate Optimization Results</td>
</tr>
<tr>
<td>COPE15</td>
<td>COPE16 COPE17</td>
<td>Function Evaluation, Approx. Optimization</td>
</tr>
<tr>
<td>COPE16</td>
<td></td>
<td>Approximate Function Evaluation</td>
</tr>
<tr>
<td>COPE17</td>
<td></td>
<td>Gradient Evaluation, Approx. Optimization</td>
</tr>
<tr>
<td>COPE18</td>
<td></td>
<td>Print Optimization Results</td>
</tr>
<tr>
<td>CORD2R</td>
<td>BCOORD</td>
<td>Blade Coordinate Systems</td>
</tr>
<tr>
<td>ESTIFF</td>
<td>MATPRN</td>
<td>Beam Material Stiffness Matrix</td>
</tr>
<tr>
<td>EXNECK</td>
<td>MATPRN MATMPY</td>
<td>Rectangular Extended Airfoil Neck</td>
</tr>
<tr>
<td>FOD</td>
<td>MATPRN</td>
<td>Local Ingestion Parameter</td>
</tr>
<tr>
<td>FRPM</td>
<td>MATMPY MATPRN</td>
<td>Centrifugal Force Calculation</td>
</tr>
<tr>
<td>Routine</td>
<td>External References</td>
<td>Function</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>HOLLOW</td>
<td>MATPRN</td>
<td>Hollow Blade Lamination</td>
</tr>
<tr>
<td>HOLZER</td>
<td>THICK PARTN</td>
<td>Holzer Transform Generation</td>
</tr>
<tr>
<td></td>
<td>WARP MINV</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SECTN MATPRN</td>
<td></td>
</tr>
<tr>
<td>INPUT</td>
<td></td>
<td>Material and Geometry Input</td>
</tr>
<tr>
<td>ITT983</td>
<td>WTT983 NDT983</td>
<td>Flutter Nodal Diameter Iteration</td>
</tr>
<tr>
<td>KDGG</td>
<td>MATPRN</td>
<td>Differential Stiffness Calculation</td>
</tr>
<tr>
<td>LAMINA</td>
<td></td>
<td>Lamina Stress-Strain Relation</td>
</tr>
<tr>
<td>LAMINI</td>
<td></td>
<td>Composite Blade Layup</td>
</tr>
<tr>
<td>LOAD2</td>
<td>MATPRN MATMPY</td>
<td>Loads for Holzer Solution</td>
</tr>
<tr>
<td>MATADD</td>
<td></td>
<td>Perform Matrix Addition</td>
</tr>
<tr>
<td>MATMPY</td>
<td></td>
<td>Perform Matrix Multiplication</td>
</tr>
<tr>
<td>MATPRN</td>
<td></td>
<td>Matrix Print</td>
</tr>
<tr>
<td>MODMAS</td>
<td>SHAPE MATMPY</td>
<td>Modal Mass determination</td>
</tr>
<tr>
<td>MODSTR</td>
<td>SHAPE MODMAS BIRDE</td>
<td>Modal Stress Driver</td>
</tr>
<tr>
<td>NDT983</td>
<td></td>
<td>Flutter Nodal Diameter Selection</td>
</tr>
<tr>
<td>OBJTV</td>
<td></td>
<td>Objective Function Calculation</td>
</tr>
<tr>
<td>PARTN</td>
<td>MATPRN</td>
<td>Matrix Partition Routine</td>
</tr>
<tr>
<td>PPI</td>
<td>MATPRN</td>
<td>Matrix Reordering</td>
</tr>
<tr>
<td>RBE2</td>
<td>MATMPY MATPRN</td>
<td>Rigid Body Coupling Matrices</td>
</tr>
<tr>
<td>RDDATA</td>
<td></td>
<td>Optimization Input Information</td>
</tr>
<tr>
<td>RDT983</td>
<td></td>
<td>Flutter Input</td>
</tr>
<tr>
<td>RDW137</td>
<td></td>
<td>Blade Model Input</td>
</tr>
<tr>
<td>Routine</td>
<td>External References</td>
<td>Function</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>RSFUNC</td>
<td></td>
<td>Resonance Function</td>
</tr>
<tr>
<td>SECTN</td>
<td>BCCORD RBEZ</td>
<td>Section Stiffness Matrix</td>
</tr>
<tr>
<td></td>
<td>MATPRN RBEZ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ESTIFF FRPM</td>
<td></td>
</tr>
<tr>
<td>SHAPE</td>
<td>PPI MATMPY</td>
<td>Eigenvector Transformation</td>
</tr>
<tr>
<td>SMULT</td>
<td>MATMPY MATPRN</td>
<td>Holzer String Multiplication</td>
</tr>
<tr>
<td>STABEL</td>
<td>INPUT FOD ZSTRES</td>
<td>Stacked Beam Analysis Driver (Versions</td>
</tr>
<tr>
<td></td>
<td>CORDZR ZSTRES</td>
<td>Differ for Hollow and Superhybrid</td>
</tr>
<tr>
<td></td>
<td>LAMINA TSAIWU</td>
<td>Blades)</td>
</tr>
<tr>
<td></td>
<td>PPI MATMPY</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HOLZER MATPRN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STATIC</td>
<td></td>
</tr>
<tr>
<td>STAEBL</td>
<td>PBMFIT HOLLOW</td>
<td>Blade Interpolation for Approximate</td>
</tr>
<tr>
<td></td>
<td>CUBIC LAMINI</td>
<td>Analysis (Versions Differ for Hollow</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and Superhybrid Blades)</td>
</tr>
<tr>
<td>STATIC</td>
<td>MATPRN MATADD MATADD</td>
<td>Blade Static Solution</td>
</tr>
<tr>
<td></td>
<td>PARTN MINV</td>
<td></td>
</tr>
<tr>
<td>THICK</td>
<td></td>
<td>Layer Thickness Evaluation</td>
</tr>
<tr>
<td>Tmax</td>
<td></td>
<td>Section Maximum Thickness</td>
</tr>
<tr>
<td>TSAIWU</td>
<td>MATPRN</td>
<td>Tsai-Wu Failure Criterion</td>
</tr>
<tr>
<td>WARP</td>
<td>MATPRN</td>
<td>Thin Section Warping</td>
</tr>
<tr>
<td>WTT983</td>
<td></td>
<td>Flutter Output Routine</td>
</tr>
<tr>
<td>WTW137</td>
<td></td>
<td>Blade Analysis Output</td>
</tr>
<tr>
<td>ZSTRES</td>
<td>MATPRN MATMPY</td>
<td>Composite Layer Stress</td>
</tr>
</tbody>
</table>

83
<table>
<thead>
<tr>
<th>Routine</th>
<th>External References</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUBIC</td>
<td></td>
<td>Function Evaluation for Parametric Beam Fit</td>
</tr>
<tr>
<td>MINV</td>
<td></td>
<td>Matrix Inversion by Gaussian Elimination</td>
</tr>
<tr>
<td>MNT983</td>
<td></td>
<td>Pratt &amp; Whitney Aircraft Flutter Analysis System</td>
</tr>
<tr>
<td>MNW137</td>
<td>STABEL</td>
<td>Pratt &amp; Whitney Aircraft Beam Analysis System, Modified Through STABEL Analysis Package for Hollow and Composite Blades</td>
</tr>
<tr>
<td>MYTIME</td>
<td></td>
<td>Clock Times, Used to Bookkeep Analysis Time Consumption</td>
</tr>
<tr>
<td>PBMFIT</td>
<td></td>
<td>Parametric Beam Fit Coefficient Evaluation</td>
</tr>
</tbody>
</table>
APPENDIX B

COMPILED FORTRAN FOR HOLLOW BLADE
DEMONSTRATION CASE
DATA SET U490STEAEB AT LEVEL 019 AS OF 03/15/82
DATA SET U490STAECL AT LEVEL 018 AS OF 01/20/82
DATA SET U490STAEQL AT LEVEL 015 AS OF 10/06/81
DATA SET U490STAEQL AT LEVEL 013 AS OF 06/24/81
DATA SET U490STAEQL AT LEVEL 012 AS OF 04/14/81

ISH 0002
SUBROUTINE STAE81 00005
ISH 0003
COMMON /BLK2/ XP(3,53,21), NP(21), IND(21), INP
ISH 0004
COMMON /BLK 2/ NHOACR, NSTA, NSTA1, POISSN, MST2, MST3
ISH 0005
COMMON /BLK 4/ R7H, XHOACR, BR, PI, TERN, N3S
ISH 0006
COMMON /BLK 7/ AREA(21), CF(21), AA(21), AKG(21), IT(7,21)
ISH 0007
COMMON /BLK12/ XBAR(21), XMAX(21), XMID(21), YBAR(21), 
1 TLTA(21)
ISH 0008
COMMON /BLKAA/ ALPHIN(21), XSCG(21), YSCG(21), YX(21), YY(21)
ISH 0009
COMMON /BLKBA/ SNK(21), TITTLE(10), VARI(235), MAXX(21), MAXF(21)
ISH 0010
COMMON /BLK A/ ALPHI(21), HA(21), HNSH(21), HAM2N(21)
ISH 0011
COMMON /BLK 9/ DELTAC(21), DEFA(21), F(21), R(21), SHMT(21)
ISH 0012
COMMON /INPUT/ XS2AVE(1000), YSAVE(1000), ZSAVE(1000), TSAVE(1000)
ISH 0013
COMMON /S2C/ XSCSG(21), YSCSG(21), ZSCSG(21)
ISH 0014
COMMON /ALGS3/ DLED, DTHD, DTID, TDID, NC0
ISH 0015
DIMENSION YH(53,21), XM(53,21), THTC(53,21), T(120), T(H(53,21))
1, XF(50,21), YF(50,21), THLF(50,21), XCG(50,21), YCG(50,21)
2, XSCSG(21), YSCSG(21), ZSCSG(21)
3, XG(50,21), YG(50,21), ZG(50,21)
4, XSCG(21), YSCG(21), ZSCG(21)
5, XG(50,21), YG(50,21), ZG(50,21)
ISH 0016
REAL*4 A1(51), B1(51), C1(51), D1(51), A1(51), B1(51), C1(51), D1(51),
1, A2(51), B2(51), C2(51), D2(51), A2(51), B2(51), C2(51), D2(51)
2, D3(51), S1(51), S2(51)
ISH 0017
REAL*4 XM(51,1), YM(51,1), XM(51,1), XM(25,1), YM(25,1), THTLF(25)
ISH 0018
REAL*4 SDME, SDTS

C
C CALCULATE MEAN Y VALUE AND THICKNESS NORMAL TO THE CHORD
C
ISH 0019
NBR = 1
ISH 0020
DO 20 I = NBR,NSTA
ISH 0021
NPS = NP(I)
ISH 0022
DO 10 N = 1,NPS
ISH 0023
YM(N,I) = YF(N,I) + XP(3,N,I) / 2.0
ISH 0024
XM(N,I) = XF(N,I)
ISH 0025
THC(N,I) = XP(2,N,I) - XP(3,N,I)
ISH 0026
10 CONTINUE
ISH 0027
20 CONTINUE
C
C CALCULATE THICKNESS NORMAL TO MEANLINE
C
ISH 0028
DO 40 I = NBR,NSTA
ISH 0029
NPS = NP(I)-1
ISH 0030
THL(I,I) = 0.0
ISH 0031
THL(NPS+1,I) = 0.0
ISH 0032
DO 30 N = 2,NPS
ISH 0033
TH1 = ATAN2(XM(1,N,I) - YM(N-1,I)),(YM(N-1,I) - XM(N-1,I))
ISH 0034
TH2 = ATAN2((YM(N+1,I) - YM(N,I)),(XM(N+1,I) - XM(N,I)))
1.3.0 (01 MAY 80) SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)
C

C SAVE MEAN Y VALUES AND X VALUES IN AN ARRAY AND INTERPOLATE
C FOR EQUAL INCREMENT X VALUES . REPEAT FOR THICKNESSES
C

C ISN 0043
DO 110 I = NER,NSTA
C ISN 0044
NPS = 15
C ISN 0045
NPS = NP(I)
C
C FILL IN DUMMY ARRAYS XM,YM,TML FROM XM,YM,TML VECTORS
C

C ISN 0046
DO 201 K = 1,NPS
C ISN 0047
TML(K) = TML(K,I)
C ISN 0048
XM(K) = XM(K,I)
C ISN 0049
201 YM(K) = YM(K,I)
C
C CALL PBWIN FIT CURVE FIT
C
C CALL PBWINFIT(XM,YM,1,1,NPS,0,A,B,C,D,B1,B1,B1,B1,B1)
C
C ISN 0050
CALL PBWINFIT(XM,YM,1,1,NPS,0,A2,B2,C2,D2,A3,B3,C3,D3,S1)
C
C ISN 0051
CALL PBWINFIT(XM,YM,1,1,NPS,0,A2,B2,C2,D2,A3,B3,C3,D3,S2)
C
C EQUAL BREAKUP ARC-LENGTH
C

C ISN 0052
SARC = S1(NPS) / NPF
C
C NON SEARCH S1 ARRAY FOR INTERVAL VALUE
C

C ISN 0053
NODE = 1
C ISN 0054
ARC1 = SARC / 2.
C ISN 0055
205 N = 1
C ISN 0056
210 IF(ARC1 .LE. SI(N)) GO TO 220
C ISN 0057
N = N + 1
C ISN 0058
GO TO 210
C

C ISN 0059
DIST = (ARC1 - SI(N-1)) / (SI(N) - SI(N-1))
C ISN 0060
SAME = (SI(N) - SI(N-1)) / DIST
C ISN 0061
STH = (S2(N) - S2(N-1)) / DIST
C
C CALL CUSIC(A(N-1),B(N-1),C(N-1),D(N-1),S1,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi
C
C ISN 0063
CALL CUSIC(A(N-1),B(N-1),C(N-1),D(N-1),S1,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi
C
C ISN 0064
CALL CUSIC(A(N-1),B(N-1),C(N-1),D(N-1),S1,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi
C
C ISN 0065
CALL CUSIC(A(N-1),B(N-1),C(N-1),D(N-1),S1,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi,ţi
C
C ISN 0066
NODE = NODE + 1
C ISN 0067
ARC1 = ARC1 + SARC
C ISN 0068
IF(NODE .LE. NPF) GO TO 205
C
C FILL IN THE XMF,YMF,THLF VECTORS WITH THE INTERPOLATED VALUES
TRANSLATE COORDINATES TO AN ENGINE AXIS SYSTEM

DO 130 I = NSR,NSTA

XCHORD = SNSK(I)

XSCL = XCHORD - XSCG(I)

XSCGG(I) = XSCL - (XCHORD - XBARX(I))

YSCGG(I) = YSCL - YBARY(I)

DO 120 N = 1,1NPF

XCGW(I) = XCGW(I) - XCHORD - XBARX(I)

YCGW(I) = YCGW(I) - YBARY(I)

120 CONTINUE

130 CONTINUE

SHIFT XY PLANE TO ENGINE AXIS YZ

DO 140 I = 1,1NPF

XSCG(I) = R(I)

YSCG(I) = YSCG(I)

ZSCG(I) = ZSCG(I)

DO 140 N = 1,1NPF

XG(N,I) = R(I)

YG(N,I) = YSCG(I)

ZG(N,I) = ZSCG(I)

140 CONTINUE

150 CONTINUE

ROTATE THRU ALPHA CHORD - 90.0

DO 170 I = NSR,NSTA

ALPHA(I) = 90. * 0.0174533

ANG = -ALPHA(I) + 90.0 + 0.0174533

E1 = COS(ANG)

EN = SIN(ANG)

XSCR(I) = XSCG(I)

YSCR(I) = YSCG(I) * EM + EN * ZSCG(I)

ZSCR(I) = EM * ZSCG(I) - EN * YG(N,I)

170 CONTINUE

WRITE (6,900) ANG, EN, EH, XSCR(I), YSCR(I), ZSCR(I), AREAI

160 CONTINUE

170 CONTINUE

DO 175 I = 1,1NPF

X(N,I) = XG(N,I)

Y(N,I) = YG(N,I) * EM + EN * ZG(N,I)

Z(N,I) = EM * ZG(N,I) - EN * YG(N,I)

175 CONTINUE

WRITE (6,900) (XCG(N,I),N=1,1NPF)
C WRITE (6,900) (YCGH(I,I),H=1,NFF)
C WRITE (6,900) (XG(I,I),H=1,NFF)
C WRITE (6,900) (YG(I,I),H=1,NFF)
C WRITE (6,900) (ZG(I,I),H=1,NFF)
C WRITE (6,900) (XH(I,I),N=1,NFF)
C WRITE (6,900) (YH(I,I),N=1,NFF)
C WRITE (6,900) (ZH(I,I),N=1,NFF)
C WRITE (6,900) (ZH(I,I),H=1,NFF)
C WRITE (6,900) (THF(N,I),H=1,NFF)
C 175 CONTINUE

ISN 0109
IJ = 1
IF(IJ .EQ. 1) GO TO 811
C
C C0902 GENERATED NODES AND THICKNESSES
C
ISN 0112
DO 800 I = NBR,NSTA
ISN 0113
DO 801 J = 1,NFF
ISN 0114
601 READ(5,602) Y(J,I),Z(J,I)
ISN 0115
602 FORMAT(/,12X,2F8.0)
C
ISN 0116
DO 805 J = 1,NFF
ISN 0117
READ(5,603) P1,P2
ISN 0118
603 FORMAT(24X,F8.0,,/,,24X,F8.0)
ISN 0119
605 TMLF(J,I) = (P1 + P2) / 2.
C
C WRITE(6,810) I
C C010 FORMAT(SX,'#0902 NODES AND THICKNESSES FOR SECTION "I5"
C
ISN 0120
800 CONTINUE
ISN 0121
811 CONTINUE
C
ISN 0122
J = 0
ISN 0123
K = 0
ISN 0124
DO 190 I = NBR,NSTA
ISN 0125
K = K + 1
ISN 0126
XSCSV(K) = XSCR(I)
ISN 0127
YSCSV(K) = YSCR(I)
ISN 0128
ZSCSV(K) = ZSCR(I)
ISN 0129
ALSAVE(K) = ALPHA(I)
ISN 0130
POLARI(K) = XIHNI(I) + XIHAX(I)
ISN 0131
ASAVE(K) = AREA(I)
ISN 0132
DO 180 N = 1,NPF
ISN 0133
J = J +1
ISN 0134
XSAVE(J) = XH(I)
ISN 0135
YSAVE(J) = YH(I)
ISN 0136
ZSAVE(J) = ZH(I)
ISN 0137
TSAVE(J) = TMLF(H,I)
C
ISN 0138
180 CONTINUE
ISN 0139
190 CONTINUE
C
ISN 0140
K = 0
ISN 0141
DO 200 I = NBR,NSTA
ISN 0142
K = K + 1
C
ISN 0143
WRITE (7,901) XSAVE(J), YSAVE(J), ZSAVE(J), TSAVE(J)
C
ISN 0144
180 CONTINUE
ISH 0144  NTHUR = NTA - NBR + 1  00232
ISH 0145  IF (NCD .EQ. 1)  00233
1CALL HOLLOW (OLED , DTED , DROOT , DTIPD ,  00234
2 TTID , TLTD , HPF , NTHUR)  00235
ISH 0147  IF (NCD .EQ. 2)  00236
1 CALL LAMINI (TSKIH , TCENTR , PBT , FGE , NPF , NTHUR)  00237
ISH 0149  901 FORMAT (6E12.5)  00238
ISH 0150  900 FORMAT (1X,9F8.5)  00239
ISH 0151  RETURN  00240
ISH 0152  END  00241

*OPTIONS IN EFFECT*NAMEMAIN OPTIMIZE(3) LINESCOUNT(60) SIZE(MAX) AUTOHEDBL(DBL)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT HC MAP NOFORMAT NOALC NOANSF TERM IBM FLAG(1)
*STATISTICS* SOURCE STATEMENTS = 151, PROGRAM SIZE = 141916, SUBPROGRAM NAME =STAEBL
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

2972K BYTES OF CORE NOT USED
DATA SET U498DSTABE AT LEVEL 048 AS OF 03/31/82

SUBROUTINE BBIP ( NC )
IMPLICIT REAL*3(A-H,O-Z)

MULTI-MODE BENDING BIRD INGESTION PARAMETER

COMMON /FAIL/ VV(42), TAIL(8,25)
COMMON /SMAKX/ SMAK1(8), SMAKX(8),SMAKX(8),SMAKX(8),SMAKX(8),SMAKX(8),SMAKX(8)
COMMON /MODAL/ STRESS(3,8,25), FACTOR(3), RMASS(3)
COMMON /DIAM00/ DIAMOD(100), FNI(100), FILL(300)
DIMENSION SEFF(3,8,25)

CALCULATE TIME TO 1/4 OF FIRST BENDING

TIME = 0.25 / FNI(1)

WRITE(6,200) TIME

SUBROUTINE BBIP ( NC )
IMPLICIT REAL*3(A-H,O-Z)

MULTI-MODE BENDING BIRD INGESTION PARAMETER

COMMON /FAIL/ VV(42), TAIL(8,25)
COMMON /SMAKX/ SMAK1(8), SMAKX(8),SMAKX(8),SMAKX(8),SMAKX(8),SMAKX(8),SMAKX(8)
COMMON /MODAL/ STRESS(3,8,25), FACTOR(3), RMASS(3)
COMMON /DIAM00/ DIAMOD(100), FNI(100), FILL(300)
DIMENSION SEFF(3,8,25)

CALCULATE TIME TO 1/4 OF FIRST BENDING

TIME = 0.25 / FNI(1)

WRITE(6,200) TIME

EFFECTIVE STRESS CALCULATION - MODAL SUPERPOSITION

FOR EACH LAYER, EACH BEAM - THREE COMPONENTS

ARG = FNI(1) * 2. * 3.141593 * TIME
FS1 = FACTOR(1) * DSINARG

ARG = FNI(2) * 2. * 3.141593 * TIME
FS2 = FACTOR(2) * DSINARG

ARG = FNI(3) * 2. * 3.141593 * TIME
FS3 = FACTOR(3) * DSINARG

DO 100 I = 1, 3
DO 100 J = 1, 8
DO 100 K = 1, NC
SEFF(I,J,K) = STRESS(I,J,K) * FS1 + STRESS(I,J,K) * FS2 +
STRESS(I,J,K) * FS3

TSAI-WU FAILURE CALCULATION

CALL TSAIWU ( SEFF,NC )

PICK OUT THE MAX TSAI-WU FOR EACH LAYER OVER ALL BEAMS

DO 110 I = 1, 8
DO 110 J = 1, 8
DO 110 K = 1, NC

IF(SMAKX(I) .LT. TSAI(I,J)) SMAKX(I) = TSAI(I,J)

WRITE(6,300)

CALL MATPRN /MAXI/ STA1X1,8,1, 'MAXI')
CALL MATPRM /MAX2/ STA1X2,8,1, 'MAX2')

WRITE(6,300)
**MAX** TSAI-WU BBIP ANALYSIS **')

**TSAI-WU AT LE AND TE **')

CALL MATPRN(SMAXL,8,1,'LE ')

CALL MATPRN(SMAXL,8,1,'TE ')

RETURN

END

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

*OPTIONS IN EFFECT*: SOURCE EBCDIC NOLIST NODECK OBJECT NOAMAP 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
SUBROUTINE BIRD( NR, NC, ISTA, BFORCE, SQUASH )

IMPLICIT REAL*: (A-H,O-Z)

CALCULATES THE BIRD IMPACT FORCE

ASSUMPTIONS
1. 2 INCHES FROM TIP
2. 2.5 LB BIRD - 4 INCH DIAMETER
3. VBIRD = 180 KNOTS
4. LOADED OVER SQUASH-UP TIME
5. SLICE FROM CENTER OF SPHERE

COMMON (INPUTT: X(1000), Y(1000), Z(1000), T(1000))

COMMON /BKL 3/ FN, BLADES, BETA, THR, THT, HST, RBB

COMMON /EC/ C(6,6), RPH

COMMON /SC/ XSC(21), YSC(21), ZSC(21), ALPHA(21), SECIP(21), SECA(21)

COMMON /BLK 6/ SHB(21), ITTLE(18), VARI(235), TMAX(21), HALPHA(21)

DIMENSION BFORCE(6, 1)

IMPACT STATION, ISTA

ISTA = 1

XTOP = X(NR * NC) - 2.

NODE = (ISTA-1) * NC + 1

IF(X(NODE) .GT. XTOP) GO TO 50

ISTA = ISTA + 1

GO TO 51

50 CONTINUE

ISTA = ISTA - 1

NODE = NODE - NC

WRITE(6, 300) NODE, ISTA

C300 FORMAT(/,5X,'** IN BIRD **, NODE, ISTA', 215)

WRITE(6, 300) NODE

CALCULATE THE IMPACT PARAMETERS

VBLADE = X(NODE) * RPH


VREL = DSQR( VBLADE**2 + VBIRD**2 )

ARG = VBIRD / VBLADE

PHI = DATAN (ARG)

THETA = ALPHA(ISTA) - PHI

GAP = 2. * 3.141593 * X(NODE) / BLADES

SL = GAP * DTAN (PHI)

RBIRD = 2.

SQUASH = 2. * RBIRD / VREL

RHO = 0.036 / 366.4

BMASS = SL * 3.141593 * RBIRD**2 * RHO

F = BMASS * VREL * DSIN(THETA) / SQUASH

CENTER OF IMPACT, COI

COI = SL / 2. / DSIN(PHI)
C 00119
C
C 00120
C 00121
C
C 00122
C

ISH 0034
TORQUE = SIG1(ISTA) / 2. - COI

ISH 0035
TORQUE = TORQUE * F

C 00124
C

ISH 0036
BFORCE(1,1) = 0.

ISH 0037
BFORCE(2,1) = F.

ISH 0038
BFORCE(3,1) = 0.

ISH 0039
BFORCE(4,1) = TORQUE.

ISH 0040
BFORCE(5,1) = 0.

ISH 0041
BFORCE(6,1) = 0.

C 00132
C

CALL MATPRN (BFORC,6,1,'BFOR')
C
C
C 00140
C

ISH 0042
RETURN

ISH 0043
END
SUBROUTINE BCOORD(IIP,IPP,BLOCAL)

THIS ROUTINE CALCULATES THE BEAM LOCAL COORDINATE SYSTEM, BLOCAL

THE SYSTEM IS LOCATED AT THE BEAM END I

LONGITUDINAL VECTOR, X

LOCAL Z UNIT VECTOR = X CROSS Y

THE LOCAL Y UNIT VECTOR = Z CROSS X

RETURN

END

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

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

*STATISTICS* SOURCE STATEMENTS = 30, PROGRAM SIZE = 716, SUBPROGRAM NAME = BCOORD

*STATISTICS* NO Diagnostics Generated

END OF Compilation
SUBROUTINE BEAM(NC,IBEAM,BLOCAL,BCHSS,RFM)

THIS ROUTINE DEFINES THE FOLLOWING MATRICES:

\( C(I,J) \) -- COEFFICIENT MATRIX

\( X(I,J) \) -- STRAIN-DISPLACEMENT MATRIX

\( A(I,J) \) -- EQUILIBRIUM MATRIX

\( BCHSS(I,J) \) -- THE BEAM MASS (NO ROTATIONAL INERTIA)

\( BCHSS(12,12) \) -- THE BEAM CENTRIFUGAL RESTORING MATRIX

THEN FINDS THE BEAM STIFFNESS MATRIX, \( BK(I,J) \)

IMPLICIT REAL*8(A-H,O-Z)

COMMON /BSTIFF/ BK(12,12),BM(12,12)

COMMON /ESTIFF/ E(8,6)

COMMON /LAYERS/ TH(7,25),BMASS(25),BSPAN(25),BHEIGHT(25)

COMMON / REGRID/ P(12,12),PT(I,J)

DIMENSION CH(12,12),X(6,12),A(12,6),BLOCAL(3,3),BCHSS(12,12)

DIMENSION D(I,J),D2(6,12),D3(12,12),CHSS(12,12),BL(12,12)

DIMENSION BLT(12,12)

FIRST ZERO ALL MATRICES TO BE USED

DO 100 I = 1,12
  DO 100 J = 1,12
    BL(I,J) = 0.
    CMSS(I,J) = 0.

100

100 I = 1,6
  DO 101 J = 1,12
    X(I,J) = 0.

101

100 I = 1,12
  DO 102 J = 1,6
    A(I,J) = 0.

102

BEGIN LOOP FOR BEAM (J)

S = BSPAN(IBEAM)

S2 = S*S2

S3 = S*S3

CH(1,1) = 1.

CH(2,2) = -1.

CH(3,1) = -1. / S

CH(3,3) = 1. / S

CH(4,2) = -1. / S

CH(4,4) = 1. / S

CH(5,5) = 1.

CH(6,6) = 1.

CH(7,5) = -3. / S2

CH(7,6) = -2. / S

CH(7,7) = 3. / S2
C STRAIN-DISPLACEMENT MATRIX, X(6,12)

X(1,3) = 1.
X(2,4) = 1.
X(3,7) = 2.
X(4,7) = 2.
X(5,11) = 2.
X(6,11) = 2.
X(6,12) = 6.

THE EQUILIBRIUM MATRIX, A(12,8)

A(1,1) = -.5
A(1,3) = -.5
A(2,2) = -.5
A(2,4) = -.5
A(3,1) = .5
A(3,3) = .5
A(4,2) = .5
A(4,4) = .5
A(5,5) = -1. / S
A(5,6) = 1. / S
A(6,5) = -1.
A(7,5) = 1. / S
A(7,6) = -1. / S
A(8,6) = 1.
A(9,7) = -1. / S
A(9,8) = 1. / S
A(10,7) = 1.
A(11,7) = 1. / S
A(11,8) = -1. / S
A(12,8) = -1.

BK = A(12,8)*E(8,6)*X(6,12)*CN(12,12)

INTERMEDIATE MATRICES: D1 = X * CN
D2 = E * D1
REORDER THE BEAM STIFFNESS, BK, TO LOOK LIKE A NASTRAN VECTOR

BK = P * BK * P(TRANS)

 FORM THE BEAM MASS MATRIX, BM

DO 15 I = 1, 12
  DO 15 J = 1, 12
  BM(I, J) = 0.
  BM(1, 1) = MASS(IBEAM)
  BM(2, 2) = BM
  BM(3, 3) = BM
  BM(7, 7) = BM
  BM(8, 8) = BM
  BM(9, 9) = BM

THE BEAM CENTRIFUGAL RESTORING MATRIX, BCHG

-- FIRST DEFINE THE TRANSFORMATION MATRIX, BL

IRON = 1
DO 90 I = 1, 4
  DO 90 J = 1, 3
  ICOL = (I-1) * 3 + 1
  DO 91 K = 1, 3
    BL(IRON, ICOL) = BLOCAL(J, K)
  91
  IRON = IRON + 1
90
DO 92 I = 1, 12
  DO 92 J = 1, 12
  BLT(I, J) = BL(I, J)
92
NONZERO VALUES OF THE CENTRIFUGAL RESTORING MATRIX

WHEN DEFINED IN THE GLOBAL COORDINATE SYSTEM

CHGG(2, 2) = MASS(IBEAM) / 2. * RPM**2
CHGG(8, 8) = CHGG(2, 2)

TRANSFORM CHGG TO THE LOCAL BEAM SYSTEM TO FIND BCHG

BCHG = BL * CHGG * BLT

WRITE(6, 7)
  FORMAT(5X, 'BEAM STIFFNESS, BK')
WRITE(6, 16)
  FORMAT(5X, 'BEAM MASS, BM')
C 00362
ISH 0109 RETURN
C 00363
ISH 0110 END

*OPTIONS IN EFFECT = NAME(HAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOBB(DBL4)
*OPTIONS IN EFFECT = SOURCE EBCDIC NLISIT NODECK OBJECT NOCPAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

*STATISTICS* SOURCE STATEMENTS = 109, PROGRAM SIZE = 16198, SUBPROGRAM NAME = BEAM

*STATISTICS* NO DIAGNOSTICS GENERATED

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

2996K BYTES OF CORE NOT USED
SUBROUTINE COR2R(XP,YP,ZP,ISPAU,ICHORD)

COMMON /COORD1/ LOCAL(3,3,1000)

DIMENSION XP(I),YP(I),ZP(I),BLOCAL(3,3)

NLNPC=ICHORD
NLRNPR=ISPAU

THIS ROUTINE CALCULATES THE COR2R FOR THE BLADE

DO 100 I = 1,ISPAU

X1 = XP(I+NLNPC) - XP(I)
X2 = YP(I+NLNPC) - YP(I)
X3 = ZP(I+NLNPC) - ZP(I)
Z1 = 0.
Z2 = YP(I+1) - YP(I)
Z3 = ZP(I+1) - ZP(I)
GO TO 10

DO 101 J = 1,ICHORD

IF(I.EQ.1.AND.J.EQ.1) GO TO 200
IF(I.EQ.1.AND.J.EQ.1) GO TO 1
IF(I.EQ.1.AND.J.GT.1.AND.J.LT.NLNPC) GO TO 4
IF(I.EQ.1.AND.J.EQ.NLNPC) GO TO 5
IF(I.EQ.1.AND.J.GT.1.AND.J.LT.NLRNPR) GO TO 6
IF(I.EQ.1.AND.J.EQ.NLRNPR) GO TO 7
IF(I.EQ.1.AND.J.GT.1.AND.J.LT.NLRPC) GO TO 8
IF(I.EQ.1.AND.J.EQ.NLRPC) GO TO 9
GO TO 10

X1 = XP(I+NLNPC) - XP(I)
X2 = YP(I+NLNPC) - YP(I)
X3 = ZP(I+NLNPC) - ZP(I)
Z1 = 0.
Z2 = YP(I+1) - YP(I-1)
Z3 = ZP(I+1) - ZP(I-1)
GO TO 10

X1 = XP(I+NLNPC) - XP(I)
X2 = YP(I+NLNPC) - YP(I)
X3 = ZP(I+NLNPC) - ZP(I)
Z1 = 0.
Z2 = YP(I+1) - YP(I)
Z3 = ZP(I+1) - ZP(I)
GO TO 10

X1 = XP(I+NLNPC) - XP(I)
X2 = YP(I+NLNPC) - YP(I)
X3 = ZP(I+NLNPC) - ZP(I)
Z1 = 0.
Z2 = YP(I+1) - YP(I)
Z3 = ZP(I) - ZP(I-1)
GO TO 10

X1 = XP(I+NLNPC) - XP(I)
X2 = YP(I+NLNPC) - YP(I)
X3 = ZP(I+NLNPC) - ZP(I)
Z1 = 0.
Z2 = YP(I+1) - YP(I)
Z3 = ZP(I+1) - ZP(I)
GO TO 10
ISH 0061  5  X1 = XPIP(NLHPC) - XPIP(NLHPC) 00420
ISH 0062  X2 = YPIP(NLHPC) - YPIP(NLHPC) 00421
ISH 0063  X3 = ZPIP(NLHPC) - ZPIP(NLHPC) 00422
ISH 0064  Z1 = 0. 00423
ISH 0065  Z2 = YPIP(IP+1) - YPIP(IP) 00424
ISH 0066  Z3 = ZPIP(IP+1) - ZPIP(IP) 00425
ISH 0067  GO TO 10 00426
ISH 0068  6  X1 = XPIP(IP+NLHPC) - XPIP(IP+NLHPC) 00427
ISH 0069  X2 = YPIP(IP+NLHPC) - YPIP(IP+NLHPC) 00428
ISH 0070  X3 = ZPIP(IP+NLHPC) - ZPIP(IP+NLHPC) 00429
ISH 0071  Z1 = 0. 00430
ISH 0072  Z2 = YPIP(IP) - YPIP(IP+1) 00431
ISH 0073  Z3 = ZPIP(IP) - ZPIP(IP+1) 00432
ISH 0074  GO TO 10 00433
ISH 0075  7  X1 = XPIP(IP) - XPIP(IP+NLHPC) 00434
ISH 0076  X2 = YPIP(IP) - YPIP(IP+NLHPC) 00435
ISH 0077  X3 = ZPIP(IP) - ZPIP(IP+NLHPC) 00436
ISH 0078  Z1 = 0. 00437
ISH 0079  Z2 = YPIP(IP+1) - YPIP(IP) 00438
ISH 0080  Z3 = ZPIP(IP+1) - ZPIP(IP) 00439
ISH 0081  GO TO 10 00440
ISH 0082  8  X1 = XPIP(IP) - XPIP(IP+NLHPC) 00441
ISH 0083  X2 = YPIP(IP) - YPIP(IP+NLHPC) 00442
ISH 0084  X3 = ZPIP(IP) - ZPIP(IP+NLHPC) 00443
ISH 0085  Z1 = 0. 00444
ISH 0086  Z2 = YPIP(IP+1) - YPIP(IP) 00445
ISH 0087  Z3 = ZPIP(IP+1) - ZPIP(IP) 00446
ISH 0088  GO TO 10 00447
ISH 0089  9  X1 = XPIP(IP) - XPIP(IP+NLHPC) 00448
ISH 0090  X2 = YPIP(IP) - YPIP(IP+NLHPC) 00449
ISH 0091  X3 = ZPIP(IP) - ZPIP(IP+NLHPC) 00450
ISH 0092  Z1 = 0. 00451
ISH 0093  Z2 = YPIP(IP) - YPIP(IP+1) 00452
ISH 0094  Z3 = ZPIP(IP) - ZPIP(IP+1) 00453
ISH 0095  10 CONTINUE 00454
C
C HQD FORM THE UNIT VECTORS, FIRST LOCAL Z THEN Y AND FINALLY X
C
ISH 0096  ZMAG =DSQRT(Z1**2 + Z2**2 + Z3**2) 00455
ISH 0097  Z1 = Z1 / ZMAG 00456
ISH 0098  Z2 = Z2 / ZMAG 00457
ISH 0099  Z3 = Z3 / ZMAG 00458
ISH 0100  XMAG =DSQRT(X1**2 + X2**2 + X3**2) 00459
ISH 0101  X1 = X1 / XMAG 00460
ISH 0102  X2 = X2 / XMAG 00461
ISH 0103  X3 = X3 / XMAG 00462
C
C LOCAL Y UNIT VECTOR, Z CROSS X
ISH 0104  Y1 = Z2*X3 - Z3*X2 00463
ISH 0105  Y2 = Z1*X3 + Z3*X1 00464
ISH 0106  Y3 = Z1*X2 - Z2*X1 00465
ISH 0107  YMAG =DSQRT(Y1**2 + Y2**2 + Y3**2) 00466
ISH 0108  Y1 = Y1 / YMAG 00467
ISH 0109  Y2 = Y2 / YMAG 00468
ISH 0110  Y3 = Y3 / YMAG 00469
C
C LOCAL X UNIT VECTOR, Y CROSS Z
ISH 0111  X1 = Y2*X3 - Y3*X2 00470
ISH 0112  X2 = Y1*X3 + Y3*X1 00471
ISH 0113  X3 = Y1*X2 - Y2*X1 00472
ISH 0114  A1 = 0 00473
ISH 0115  A2 = 0 00474
ISH 0116  A3 = 0 00475
ISH 0117 200 CONTINUE
ISH 0118 IF(ICHORD .EQ. 1) GO TO 201
ISH 0120 GO TO 202
ISH 0121 201 CLOCAL(2,1,IP) = 0.
ISH 0122 CLOCAL(2,2,IP) = 1.
ISH 0123 CLOCAL(2,3,IP) = 0.
C
ISH 0124 II = IP
ISH 0125 IF(I .EQ. ISPAN) II = IP - 1
ISH 0127 III = II + 1
ISH 0128 CALL BCOORD(II,III,BLOCAL)
C
ISH 0129 DO 400 K = 1,3
ISH 0130 DO 400 L = 1,3
ISH 0131 400 CLOCAL(K,L,IP) = BLOCAL(K,L)
ISH 0132 GO TO 302
C
ISH 0133 202 CONTINUE
C
ISH 0134 IIP = IP+1000
ISH 0135 CLOCAL(1,1,IP) = X1
ISH 0136 CLOCAL(1,2,IP) = X2
ISH 0137 CLOCAL(1,3,IP) = X3
ISH 0138 CLOCAL(2,1,IP) = Y1
ISH 0139 CLOCAL(2,2,IP) = Y2
ISH 0140 CLOCAL(2,3,IP) = Y3
ISH 0141 CLOCAL(3,1,IP) = Z1
ISH 0142 CLOCAL(3,2,IP) = Z2
ISH 0143 CLOCAL(3,3,IP) = Z3
ISH 0144 302 CONTINUE
C
ISH 0145 101 IP = IP + 1
ISH 0146 100 CONTINUE
ISH 0147 RETURN
ISH 0148 END

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOHMAP NOFORMAT GOSTHXT NOXREF NOALC NOMSF 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
SUBROUTINE ESTIFF(ISEC,NC,NR,I)

This routine determines the material stiffness matrix, $E$, for each beam evaluated at each element.

IMPLICIT REAL*8(A-H,O-Z)

C$E(8,6)$
C$Q(3,3,7)$
CONMON /I\ST$E(7,25),BMASS(25),BSPAN(25),BWIDTH(25)
C$INPUT/X(1000),Y(1000),Z(1000),T(1000)$
CONVOM /BLK/SHB(21),ITTS(18),VARI(235),THAX(21),HALPHA(21)
DINMSION Y(8)

C The layer thicknesses
$Y(I) = -(\text{TH}(1,I) + \text{TH}(2,I) + \text{TH}(3,I) + \text{TH}(4,I)/2.)$

C FIND A11,A22
A11 = 0.
A22 = 0.

C FIND B11,B13
B11 = 0.
B13 = 0.

C ROUND OFF BIJ VALUES TO ZERO IF LESS THAN 10.
CHECK = 10.
IF(DABS(B11) .LT. CHECK) B11 = 0.
IF(DABS(B13) .LT. CHECK) B13 = 0.

C FIND D11,D33,D13
D11 = 0.
D13 = 0.
D33 = 0.

C D11 = D11 + Q(I,I,J) * (Y(J+1) - Y(J))

C D22 = D22 + Q(2,2,I,J) * (Y(J+1) - Y(J))

C A11 = A11 * BWIDTH(I)

C B11 = B11 + Q(I,1,J) * (Y(J+1)**2 - Y(J)**2)

C B13 = B13 + Q(I,3,J) * (Y(J+1)**2 - Y(J)**2)

C B33 = B33 + Q(3,3,J) * (Y(J+1)**2 - Y(J)**2)

C B11 = B11 * BWIDTH(I)

C B13 = B13 * BWIDTH(I)

C B33 = B33 * BWIDTH(I)

C ROUND OFF BIJ VALUES TO ZERO IF LESS THAN 10.
CHECK = 10.
IF(DABS(B11) .LT. CHECK) B11 = 0.
IF(DABS(B13) .LT. CHECK) B13 = 0.

C FIND D11,D33,D13
D11 = 0.
D13 = 0.
D33 = 0.

C D11 = D11 + Q(I,1,J) * (Y(J+1)**3 - Y(J)**3)

C D13 = D13 + Q(I,3,J) * (Y(J+1)**3 - Y(J)**3)

C D33 = D33 + Q(3,3,J) * (Y(J+1)**3 - Y(J)**3)

C D11 = D11 * BWIDTH(I)

C D13 = D13 * BWIDTH(I)

C D33 = D33 * BWIDTH(I)

C D11 = D11 * BWIDTH(I)

C D13 = D13 * BWIDTH(I)

C D33 = D33 * BWIDTH(I)
C ROUND OFF D13 TO ZERO IF LESS THAN CHECK = 10.
C IF(OABS(D13 ) .LT. CHECK) D13 = 0.
C WRITE(6,1) I,A11,A22,B11,B13,D11,D13,D33
C FORMAT(5X,'ESTIFF FOR BEAM',15,5X,'A11,A22,B11,B13,D11,D13,D33',
C 1/,5X,7E10.3)
C FORM THE E MATRIX
C
C DO 105 J = 1,6
C 105 E(J,J) = 0.
C
C ISN 0042 IF(1,(J,J) .LT. 0) RETURN
C
C ISN 0043 105 E(J,J) = A11
C ISN 0044 E(1,2) = 2. * B13
C ISN 0045 E(1,3) = -B11
C ISN 0046 E(2,1) = B13
C ISN 0047 TOTAL = Y(0) - Y(1)
C ISN 0048 E(2,2) = 4. * D33
C
C ISN 0049 E(2,3) = -D13
C ISN 0050 E(3,1) = A11
C ISN 0051 E(3,2) = 2. * B13
C ISN 0052 E(3,3) = -B11
C ISN 0053 E(4,1) = B13
C ISN 0054 E(4,2) = E(2,2)
C
C ISN 0055 E(4,4) = -D13
C ISN 0056 E(5,1) = -B11
C ISN 0057 E(5,2) = -2. * D13
C ISN 0058 E(5,3) = D11
C ISN 0059 E(6,1) = E(4,1) = B13
C ISN 0060 E(6,2) = -2. * D13
C ISN 0061 E(6,3) = D11
C ISN 0062 E(6,4) = B11
C ISN 0063 E(7,5) = BWIDTH(I)*W2/12. * A11
C ISN 0064 E(7,6) = E(7,5)
C
C ROOT WARPING RESTRAINED ( (2,2) TERM NOT CHANGED)
C
C IF(ISEC .GT. 1) RETURN
C
C ISN 0065 E(1,1) = E(1,2) / 2.
C ISN 0066 E(2,1) = E(2,2) / 2.
C
C CALL MATPRIHE(8,6,' E ')
C
C ISN 0067 RETURN
C
C ISN 0072 RETURN
C
C OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
C OPTIONS IN EFFECT*SOURCE EBCDIC NODIST NODECK OBJECT NONAP NOFORMAT GOSTHT NOXREF NOALC NOASNF TERM IBM FLAG(I)
C *STATISTICS* SOURCE STATEMENTS = 72, PROGRAM SIZE = 1290, SUBPROGRAM NAME =ESTIFF
C *STATISTICS* NO DIAGNOSTICS GENERATED
C ******** END OF COMPILATION *******

3004K BYTES OF CORE NOT USED
SUBROUTINE EXNECK ( B )

EXTENDED NECK STIFFNESS - RECTANGULAR - TITANIUM

15 % THICK NECK

COMMON /STIF/ SK(12,12),SL(12,21),SW(25,21)
COMMON /STIFF/ SK(12,12),SI(12,21),SIM(6,6,21)
COMMON /BLKAA/ AA(21),XSC(21),YSC(21),XXX(42)
COMMON /BLKAB/ XBAX(21),XMAX(42),YBAR(21),TLTA(21)
DIMENSION RBE(12,12),RBBT(12,12),DL(12,12)

T = 0.15 * B
RIMIN = B * T**3 / 12.
RIMAX = B**3 * T**3 / 12.
TORS = B * T**3 / 12. * 0.21 * T**4 * (1. - T**4/12./B**4)
TORS = RIMIN + RIMAX

E = 16.1E6
G = 6.0E6
BL = SL(1,1)
AREA = B * T
WRITE(6,10) B,T,RIMIN,RIMAX,TORS,E,G,BL,AREA
C0 FORTRAN,'B,T,RIMIN,RIMAX,TORS,E,G,BL,AREA',/,'5X,SE12.5
C
FILL IN STIFFNESS ARRAY, SK

DO 100 I = 1,12
DO 100 J = 1,12
SK(I,J) = 0.
SK(I,1) = AREA * E / BL
SK(I,7) = -SK(I,1)
SK(2,2) = 12. * E * RIMAX / BL**3
SK(2,6) = 6. * E * RIMAX / BL**2
SK(2,8) = -SK(2,2)
SK(2,12) = SK(2,6)
SK(3,3) = 12. * E * RIMIN / BL**3
SK(3,5) = -6. * E * RIMIN / BL**2
SK(3,9) = -SK(3,3)
SK(3,11) = SK(3,5)
SK(4,4) = TORS * G / BL
SK(4,10) = -SK(4,4)
SK(5,5) = 4. * E * RIMIN / BL
SK(5,9) = -SK(3,11)
SK(5,11) = 2. * E * RIMIN / BL
SK(6,6) = 4. * E * RIMAX / BL
SK(6,8) = -SK(2,12)
SK(6,12) = 2. * E * RIMAX / BL
SK(7,7) = SK(1,1)
SK(8,8) = SK(2,2)
SK(8,12) = -SK(2,12)
ISH 0042  SK(9,9) = SK(3,3) 00675*39
ISH 0043  SK(9,11) = -SK(3,11) 00676*39
ISH 0044  SK(10,10) = SK(4,4) 00677*39
ISH 0045  SK(11,11) = SK(5,5) 00678*39
ISH 0046  SK(12,12) = SK(6,6) 00679*39

C   SYMMETRY CONDITION
C
ISH 0047  DO 110 I = 1,12
ISH 0048  DO 110 J = 1,12
ISH 0049  IF(J .LE. I) GO TO 110
ISH 0051  SK(I,J) = SK(J,I)
ISH 0052  110 CONTINUE

C   CALL MATPRN(SK,12,12,'SKRT')
C   TRANSLATE TO CG LOCATION

ISH 0053  DZ = XSC(2) - XBARX(2)
ISH 0054  DY = YBARY(2) - YSC(2)
ISH 0055  WRITE(6,11) DZ,DY
C   FORMAT(5X,'DZ,DY',2E1.5)
ISH 0056  DO 200 I = 1,12
ISH 0057  DO 200 J = 1,12
ISH 0058  IF(I .EQ. J) RBE(I,J) = 0.
ISH 0059  200 CONTINUE

C   CALL MATPRN(RBE,12,12,'RBE ')
C   RBE TRANSPOSE , RBET

ISH 0061  RBE(1,5) = DZ
ISH 0062  RBE(1,6) = -DY
ISH 0063  RBE(2,4) = -DZ
ISH 0064  RBE(3,4) = DY
ISH 0065  RBE(7,11) = DZ
ISH 0066  RBE(7,12) = -DY
ISH 0067  RBE(8,10) = -DZ
ISH 0068  RBE(9,10) = DY

C   CALL MATPRN(RBET,12,12,'RBET')

ISH 0069  DO 210 I = 1,12
ISH 0070  DO 210 J = 1,12
ISH 0071  210 RBET(I,J) = RBE(J,I)
C   CALL MATPRN(RBET,12,12,'RBET')

ISH 0072  CALL MATPYPY(SK,RBE,01,12,12,12)
ISH 0073  CALL MATPYPY(RBET,01,SK,12,12,12)

C   CALL MATPRN(SK,12,12,'SKRT')

ISH 0074  RETURN
ISH 0075  END

*OPTIONS IN EFFECT=NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(OBL4)
*OPTIONS IN EFFECT=SOURCE EBCDIC NODIST NODIECK OBJECT NOHAP NOFORMAT GOSTMT NOXREF NODLC NOANSF TERM IBM FLAG(1)
*STATISTICS* SOURCE STATEMENTS = 74, PROGRAM SIZE = 4850, SUBPROGRAM NAME =EXNECK
*STATISTICS* NO DIAGNOSTICS GENERATED
******* END OF COMPILATION *******
* 3004K BYTES OF CORE NOT USED*
SUBROUTINE FRPI1(J,IP,X,Y,BMASS,RPM,FBEAM,BLOCAL)

DOUBLE PRECISION X(1),Y(1),BMASS(1),RPM,FBEAM(6,1),BLOCAL(3,3)

DOUBLE PRECISION F(3,1),FF(3,1),A(6)

DO 100 K = 3,6
    A(K) = 0.

100

FBEAM(K,1) = 0.

NONZERO ACCELERATION COMPONENTS

A(1) = RPM**2 * X(IP)
A(2) = RPM**2 * Y(IP)

FORCES

F(1,1) = BMASS(J)/2. * A(1)
F(2,1) = BMASS(J)/2. * A(2)
F(3,1) = 0.

WRITE(6,20) J,IP
FORMAT(5X,'** IN FRP1-GLOBAL RPM LOADS AT BEAM, NODE',215)

CALL MATPRNF(BLOCAL,F,FF,3,3,1)

TRANSFORM TO LOCAL BEAM COOR. SYSTEM

CALL MATPRNF(BLOCAL,F,FF,3,3,1)

INSERT INTO FBEAM

DO 200 K = 1,3
    FBEAM(K,1) = FF(K,1)

200

WRITE(6,20) J,IP
FORMAT(5X,'** IN FRP1',215)

CALL MATPRNF(FBEAM,6,1,'FBEAM')

RETURN

END
SUBROUTINE HOLZERILL(NR,NC,RPM)

THIS ROUTINE CALCULATES ALL SECTION STIFFNESSES AND MASSES AND THEN TRANSFORMS THESE INTO HOLZER MATRICES.

IMPLICIT REAL*8(A-H.O-Z)

COMMON /STIFF/ SK(12,12),SKP(12,12,21),SMH(6,6,21)
COMMON /REORDR/ PT(12,12)
COMMON /STIFF/ SKP(12,12),SKPP(12,12,21)
COMMON /INPUT/ XI(1000),Y(1000),Z(1000),T(1000)
DIMENSION SKII(6,6),SK12(6,6),SK21(6,6),SK22(6,6)
DIMENSION RL(6),RM(6),SI(6,6),DI(12,12),DD(6,6)

BEGIN THE SECTION STIFFNESS SOLUTION MATRIX SK

NR11 = NR - 1
ISKIP = 1
GO TO 99
GO TO 98

GENERATE THE SECTION STIFFNESS SK

SECTION MASS SN

THE BEAM THICKNESSES MASSES AND LENGTHS ARE CALCULATED

THE WARPING FUNCTION WILL ALSO BE EVALUATED

CALL THICK(I,NR,NC,X,Y,Z,T,ISKIP)
IF (I.EQ. 1) CALL WARPII,NR,NC)
II = I + 1
CALL WARP(II,NR,NC)
CALL SECTII(NR,NC,RPM)
GO TO 97

THIS SECTION IS FOR SECOND STATIC ANALYSIS WITH PRESTRESS

CONTINUE

DO 100 J = 1,12
DO 100 K = 1,12
SK(J,K) = SKK(J,K)
CONTINUE

CALL HATPRNH(SK,12,12,'SK')
PARTITION THE SECTION STIFFNESS MATRIX ,SK, INTO:
SK1 , SK12 , SK21 , SK22

CALL PARTNH(SK,12,12,SK11,SK12,SK21,SK22,6,6)

INVERT SK12
CALL MINV(SK12,6,SR,RL)
FORM THE SKP MATRIX FROM PARTITIONED SK COMPONENTS

CALL HATMPY(SK12,SK11,SK1,6,6,6)
I.

(*)

CHANGE SIGN ON UPPER QUARTER OF SKP

C 0031
C
DO 115 K = 1,6
C 0032
DO 115 L = 1,6
C 0033
115
SKP(K,L) = -SK1(K,L)
C
ISN 0034
DO 116 K = 1,6
ISN 0035
DO 116 L = 7,12
ISN 0036
LM = L - 6
ISN 0037
116
SKP(K,L) = SK12(K,LM)
C
THE SIGN ON SKP21 AND SKP22 ARE NEGATIVE FOR EQUIL.
C
ISN 0038
CALL MATMPY(SK22,SK1,SK11,6,6,6)
ISN 0039
DO 117 K = 7,12
ISN 0040
KK = K - 6
ISN 0041
DO 117 L = 7,12
ISN 0042
117
SKP(K,L) = SK11(KK,L) - SK21(KK,L)
C
ISN 0043
CALL MATMPY(SK22,SK12,SK1,6,6,6)
ISN 0044
DO 125 K = 7,12
ISN 0045
KK = K - 6
ISN 0046
JJ = 1
ISN 0047
DO 125 L = 7,12
ISN 0048
SKP(K,L) = -SK1(KK,JJ)
ISN 0049
125
JJ = JJ + 1
C
FILL THE SKPP ARRAY WITH THE SKP MATRIX, THIS SAVES SKP
C
ISN 0050
DO 165 K = 1,12
ISN 0051
DO 165 L = 1,12
ISN 0052
165
SKPP(K,L,I) = SKP(K,L)
C
WRITE(6,101) I
C
101
FORMAT(/,5X,'SKP IS THE HOLZER STIFFNESS FOR SECTION',I5)
C
CALL MATPRH(SKP,12,12,'SKP ')
C
ISN 0053
200
CONTINUE
C
ISN 0054
RETURN
ISN 0055
END

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOUBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC HOLIST NODECK OBJECT NOFORMAT GOSTHT 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
SUBROUTINE INPUT (NR,NC)

MATERIAL PROPERTIES AND GEOMETRY INPUT

ISN 0003

IMPLICIT REAL*8(A-H,O-Z)

ISN 0004

COMMON /ZCDEF/ E11(7),E22(7),E33(7),G12(7),G13(7),G23(7),

ISN 0005

V12(7),V13(7),V23(7)

COMMON /PLY1/ PLY(21,25,7),THETA(7),RHO(7)

COMMON /FAIL/ X1(7),X1C(7),X2T(7),X2C(7),S6P(7),S6M(7),TSAI(8).25)

C

WRITE(6,1)

FORMAT(5X,'IS THIS AN ISOTROPIC BLADE, I=NO/1=YES')

READ(8,*) ISO

IF(ISO .NE. 1) GO TO 2

WRITE(6,3)

FORMAT(5X,'INPUT-ELASTICITY, V AND DENSITY(LB/IN3),FREE FORMAT')

READ(8,*) E,V,R

E = 16100000.

V = .33

R = .16

DO 100 I = 1,7

X1(I) = YIELD

X1C(I) = YIELD

X2T(I) = YIELD

X2C(I) = YIELD

S6P(I) = SHEAR

S6M(I) = SHEAR

E11(I) = E

E22(I) = E

E33(I) = E

G12(I) = E / 2. / (1. + V)

G13(I) = G12(I)

G23(I) = G12(I)

V12(I) = V

V13(I) = V

V23(I) = V

RHO(I) = R / 386.4

DO 100 I = 1,7

100

INPUT MATERIAL STUFF FOR TI/BORON LAYER

C

E11(2) = 33.2E6
**Isis 0037 EIN(6) = EIN(2)**

**Isis 0033 E22(2) = 29.0E6**

**Isis 0039 E22(6) = E22(2)**

**Isis 0040 E33(2) = E33(2)**

**Isis 0041 E33(6) = E33(2)**

**Isis 0042 G12(2) = 22.5E6**

**Isis 0043 G12(6) = G12(2)**

**Isis 0044 G13(2) = G12(2)**

**Isis 0045 G13(6) = G13(2)**

**Isis 0046 G23(2) = G12(2)**

**Isis 0047 G23(6) = G23(2)**

**Isis 0048 V12(2) = 0.3**

**Isis 0049 V12(6) = 0.3**

**Isis 0050 RHO(2) = .000336**

**Isis 0051 RHO(6) = RHO(2)**

**Isis 0052 V13(2) = V12(2)**

**Isis 0053 V13(6) = V13(2)**

**Isis 0054 V23(2) = V12(2)**

**Isis 0055 V23(6) = V23(2)**

**C THETA(2) = 35./180. * 3.14159265**

**C THETA(6) = THETA(2)**

**Isis 0056 YIELD1 = 170000.**

**Isis 0057 YIELD2 = 50000.**

**Isis 0058 YIELDS = 30000.**

**Isis 0059 X1T(2) = YIELD1**

**Isis 0060 X1T(6) = YIELD1**

**Isis 0061 X1C(2) = YIELD1**

**Isis 0062 X1C(6) = YIELD1**

**Isis 0063 X2T(2) = YIELD2**

**Isis 0064 X2T(6) = YIELD2**

**Isis 0065 X2C(2) = YIELD2**

**Isis 0066 X2C(6) = YIELD2**

**Isis 0067 S6P(2) = YIELDS**

**Isis 0068 S6P(6) = YIELDS**

**Isis 0069 S6H(2) = YIELDS**

**Isis 0070 S6H(6) = YIELDS**

**C**

**C**

**C**

**Isis 0071 RETURN**

**Isis 0072 END**

*Options in effect: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)*

*Options in effect: SOURCE EBCDIC NOLIST NODECK OBJECT NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(1)*

*Statistics* SOURCE STATEMENTS = 71, PROGRAM SIZE = 936, SUBPROGRAM NAME = INPUT

*Statistics* NO DIAGNOSTICS GENERATED

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

3004K BYTES OF CORE NOT USED
SUBROUTINE KOGG(ISEC,FF,FOUT1,DSKK)

CALCULATES THE DIFFERENTIAL STIFFNESS MATRIX FOR SECTION ISEC, DSKK. (SEE THE NASTRAN THEORETICAL MANUAL FOR EXPLANATION 7-2-8)

IMPLICIT REAL*8(A-H,O-Z)
COMMON /SC/XSC(21),YSC(21),ZSC(21),ALPHA(21),SECIP(21),SECA(21)
DIMENSION FF(12,1),FOUT1(12,1),DSKK(12,12,20)
REAL MYA,MYB,MZA,MZB

CALL SECTION LENGTH,SL.

X = XSC(ISEC+1) - XSC(ISEC)
Y = YSC(ISEC+1) - YSC(ISEC)
Z = ZSC(ISEC+1) - ZSC(ISEC)
SL = DSQRT(X**2 + Y**2 + Z**2)

DEFINE TERMS CONSISTANT WITH NASTRAN TERMINOLOGY

FX = FOUT1(7,1)
FY = FOUT1(8,1)
FZ = FOUT1(9,1)
MY = -FF(11,1)
HYA = -FOUT1(11,1)
HZA = -FOUT1(12,1)

THE STIFFNESS TERMS IN THE DSKK MATRIX HAVE BEEN SET TO ZERO BECAUSE THE MOMENT OF INERTIA TERM IS UNKNOWN FOR A COMPOSITE

DO 100 I = 1,12
DO 100 J = 1,12
DSKK(I,J,ISEC) = 0.

DSKK(2,2,ISEC) = 6./SL * FX
DSKK(2,4,ISEC) = -MY / SL
DSKK(2,6,ISEC) = FX / 10.
DSKK(2,8,ISEC) = -DSKK(2,2,ISEC)
DSKK(2,10,ISEC) = -HYA / SL
DSKK(2,12,ISEC) = FX / 10.
DSKK(3,3,ISEC) = DSSK(2,2,ISEC)
DSKK(3,4,ISEC) = -HZA / SL
DSKK(3,5,ISEC) = -FX / 10.
DSKK(3,9,ISEC) = -DSKK(2,2,ISEC)
DSKK(3,10,ISEC) = -HZA / SL
DSKK(3,11,ISEC) = -FX / 10.
DSKK(4,4,ISEC) = SECP(ISEC) * FX / SECA(ISEC) / SL
DSKK(4,5,ISEC) = -SL * FY / 6.
DSKK(4,6,ISEC) = -SL * FZ / 6.
DSKK(4,8,ISEC) = MY / SL
DSKK(4,9,ISEC) = HZA / SL
DSKK(4,10,ISEC) = -DSKK(4,4,ISEC)
DSKK(4,11,ISEC) = SL * FY / 6.
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) = MZA / 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
ISN 0060  DO 110 I = 1,12 01041
ISN 0061  DO 110 J = 1,12 01042
ISN 0062  IF( J.LE. I ) GO TO 110 01043
ISN 0064  DSKK(J,I,ISEC) = DSKK(I,J,ISEC) 01045
ISN 0065   110 CONTINUE
C
ISN 0066  DO 200 I = 1,12 01046
ISN 0067  DO 200 J = 1,12 01047
ISN 0068    200 DSKK(I,J) = DSKK(I,J,ISEC) 01048
C  CALL MATPRN(DSKK,12,12,'DSKK') 01049
C
ISN 0059  RETURN 01050
ISN 0070  END 01051

*OPTIONS IN EFFECT* NAME(HMAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTO=BL(DBL4)
*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NOIAP 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
C SUBROUTINE LAMINA
C THIS ROUTINE GENERATES THE LAMINA STRESS-STRAIN MATRIX, Q
C
C IMPLICIT REAL*8(A-H,O-Z)
C COMMON /COEF/ E11(7), E22(7), E33(7), G12(7), G23(7), G13(7),
V12(7), V13(7), V23(7)
C COMMON /PLY1/ PLY1(1,2,5,7), THETA(7), RHO(7)
C COMMON /QIJ/ Q(3,3,7)
C
C N = 7
C DD 100 I = 1,N
C WRITE(6,1) I
C
C CHECK FOR HOLLOW LAYER
C IF(E11(I) .EQ. 0.) GO TO 200
C DETERMINE THE POISSON RATIOS
V21 = V12(I) * E22(I) / E11(I)
V31 = V13(I) * E33(I) / E11(I)
V32 = V23(I) * E33(I) / E22(I)
C DET = 1. - V12(I)*V21 - V13(I)*V31 - V23(I)*V32 -
1 V21*V13(I)*V32 - V31*V12(I)*V23(I)
C WRITE(6,2) V21, V31, V32, DET
C
C WRITE(6,3) C11, C12, C13
C WRITE(6,4) C22, C23
C WRITE(6,5) C33
C
C ROTATE THRU ANGLE THETA(I) ABOUT THE 3 AXIS
C
C =DCOS( THETA(I) )
C S =DSIN( THETA(I) )
C
C CP11 = C**4*C11 + 2.*C**2*S**2*C12 + S**4*C66
C CP12 = C**2*S**2*C11 + C12-4.*C66
C CP13 = C**2*C11 + S**2*C66
C CP16 = -C*S*(C**2*C11 + S**2*C22) + C*S*(C**2-S**2)*C12 + 2.*C66
C CP22 = S**4*C11 + 2.*C**2*S**2*(C12+2.*C66) + C**4*C22
CP23 = S**2*C13 + C**2*C23
CP26 = -C*S*(S**2*C11-C**2*C22) - S*C*(C**2-5**2)*(C12+2.*C66)
CP33 = C33
CP36 = C*S*(C23-C13)
CP44 = C**2*C44 + S**2*C55
CP45 = C*S*(C44-C55)
CP55 = C**2*C55 + S**2*C44
CP66 = C**2*S**2*(C11+C22-2.*C12) + (C**2-5**2)*C66

C

CHECK = 10.
IF(DABS(CP16).LT.CHECK) CP16 = 0.
IF(DABS(CP26).LT.CHECK) CP26 = 0.
IF(DABS(CP36).LT.CHECK) CP36 = 0.
IF(DABS(CP45).LT.CHECK) CP45 = 0.
WRITE(6,6)
FORMAT(1X,3E12.5,24X,E12.5)
WRITE(6,7) CP11,CP12,CP13,CP16
WRITE(6,8) CP22,CP23,CP26
WRITE(6,9) CP33,CP36
WRITE(6,10) CP55
WRITE(6,11) CP66
WRITE(6,12) CP11,CP12,CP13,CP16
WRITE(6,13) CP22,CP23,CP26
WRITE(6,14) CP33,CP36
WRITE(6,15) CP55
WRITE(6,16) CP66
WRITE(6,17) CP11,CP12,CP13,CP16
WRITE(6,18) CP22,CP23,CP26
WRITE(6,19) CP33,CP36
WRITE(6,20) CP55
WRITE(6,21) CP66
Z1 = (CP23 * CP13 - CP16 * CP33) / (CP22 * CP33 - CP23**2)
Z2 = (CP26 * CP33 - CP23 * CP36) / (CP22 * CP33 - CP23**2)
Z3 = (-CP13 - CP23 * Z1) / CP33
Z4 = (CP36 - CP23 * Z2) / CP33
Q(1,1,I) = CP11 + CP13 * Z3 + CP12 * Z1
Q(1,2,I) = 0.
Q(1,3,I) = CP13 * Z4 + CP12 * Z2 - CP16
Q(2,1,I) = 0.
Q(2,2,I) = CP55
Q(2,3,I) = 0.
Q(3,1,I) = Q(1,3,I)
Q(3,2,I) = 0.
Q(3,3,I) = -CP36 * Z4 - CP26 * Z2 + CP66

C HOLLOW LAYER CONDITION
C
C
GO TO 201
VERSION 1.3.0 (01 MAY 80)  LAMINA  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 02.141/10.48.54  PAGE 3

*VERSIOU 1.3.0 (01 MAY 80) LAMINA SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 02.141/10.48.54 PAGE 3

ISH 0094  RETURN
ISH 0095  END

*OPTIONS IN EFFECT=NAME(MAIN) OPTIMIZE(3) LINECOUNT(160) SIZE(MAX) AUTOBBL(DBL4)
*OPTIONS IN EFFECT=SOURCE EBCDIC NOLIST NODECK OBJECT NOHAP NOFORMAT GOSTMT NOREF 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
SUBROUTINE LIN11 (TSKIN, TCEnt, PBT, PGE, NC, NR)

INPUT: TSKIN = TITANIUM'SKIN THICKNESS IN INCHES
TCENT = Ti CENTER LAYER THICKNESS IN INCHES
PBT = PERCENTAGE OF REAMAINING THICKNESS OF BORON/TI
PGE = PERCENTAGE OF REMAINING THICKNESS OF GRAPHITE/EPoxy

NOTE: PBT + PGE = 1.0

IMPLICIT REAL*8(A-H,O-Z)
COMMON /PLY1/ PLY(21,25,7),THETA(7),RHO(7)
COMMON /INPUT/ X(1000),Y(1000),Z(1000),T(1000)

NR1 = NR - 1
DO 100 I = 1,NR1

TAVE = (T(NODE) + T(NODE+NC)) / 2.

CHECK ON TOTAL TITANIUM LAYER THICKNESS

TTI = 2. * TSKIN + TCEnt

TAVE = TAVE - TTI

INITIALIZE THICKNESS:

TS = TSKIN
TC = TCEnt
TBT = 0.
TGE = 0.

IF(TAVE) 110,110,120

NO G/E OR TBT LAYERS DUE TO MINIMUM THICKNESS EXCEEDED

TAVE = TAVE - 2. * TSKIN

IF(TAVE) 111,111,112

TC = 0.
GO TO 200

TC = TAVE - 2. * TS
GO TO 200

INSERT G/E AND TBT LAYERS

TBT = PBT * TAVE / 2.
TGE = PGE * TAVE / 2.
CONTINUE

FILL THE PLY ARRAY

PLY(I,J,1) = TS
PLY(I,J,2) = TBT
I.

C = WRITE(6,300) I,J,NODE,TAVE,TS,TBT,TGE,TC
C300 FORMAT(5X,'I,J,NODE,TAVE,TS,TBT,TGE,TC',/5X,
C 1     315,5E12.5)

C = TOTAL = 2. * (TS + TBT + TGE) + TC
C = WRITE(6,301) TOTAL
C301 FORMAT(5X,'TOTAL SUMMED THICKNESS =',13.E12.5)

I00 NODE = NODE + 1

ISN 0036 RETURN

ISN 0037 END

*OPTIONS IN EFFECT* NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NOHMAP NOFORMAT GOSTHT NOXREF NOAHC NOAHSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 36, PROGRAM SIZE = 922, SUBPROGRAM NAME = LAHINI
*STATISTICS* NO DIAGNOSTICS GENERATED
******* END OF COMPILATION *******

3016K BYTES OF CORE NOT USED
SUBROUTINE LOAD2(NR,FF)
C
C PRODUCE A SINGLE LOAD VECTOR, FF, FROM ALL
C THE STIFFNESSES-LOAD VECTOR PRODUCTS IN THE HOLZER METHOD
C
C IMPLICIT REAL*(A-H,O-Z)
C COMMON /STATC/ SH(12,12,20),F(12,21)
C DIMENSION SI(12,12), FF(12,12), FI(12,12)
C
C INITIALIZE FF, AND INSERT THE LOAD VECTOR AT NODE NR
C
WRITE(6,10)
DO 50 I = 1,12
FF(I,1) = F(I,NR)
CALL MATPRN(FF,12,1, ' FF ')
C
NRM1 = NR - 1
C
IF(NRM1 .EQ. 1) RETURN
C
DO 100 I = 2,NRM1
C
FILL IN DUMMY ARRAYS
DO 110 J = 1,12
FH(J,1) = F(J,NR)
DO 110 K = 1,12
SI(J,K) = SH(J,K,NR-I)
WRITE(6,10)
FORMAT(/,5X,'*** IN LOAD2 ****')
CALL MATPRN(SI,12,12, ' SI ') 
CALL MATPRN(FI,12,1, ' FI ')
C
PERFORM MATRIX MULTIPLICATION
CALL MATMPY(SI,FI,F2,12,12,1)
CALL MATPRN(F2,12,1, ' F2 ')
C
SUM INTO THE FF ARRAY
DO 120 J = 1,12
FF(J,1) = FF(J,1) + F2(J,1)
C
CONTINUE
CALL MATPRN(FF,12,1, ' FF ')
C
RETURN
C
END
C
*OPTIONS IN EFFECT* NAME(HAIN) 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
SUBROUTINE HATADO(A,B,C,I,J,S)

MATRICES A, B, C

DO 100 K = 1, I
DO 100 L = 1, J
C(K,L) = A(K,L) + S * B(K,L)

RETURN

DOUBLE PRECISION A(I,J), B(I,J), C(I,J), S

DO 100 K = 1, I
DO 100 L = 1, J
C(K,L) = A(K,L) + S * B(K,L)

RETURN

*OPTIONS IN EFFECT* NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
*STATISTICS* SOURCE STATEMENTS = 7, PROGRAM SIZE = 436, SUBPROGRAM NAME = MATADD
*STATISTICS* NO DIAGNOSTICS GENERATED

END OF COMPILATION

3016K BYTES OF CORE NOT USED
SUBROUTINE MATMPY(A,B,C,I,J,K)

C

THIS ROUTINE MULTIPLIES MATRICES A * B = C

C

DOUBLE PRECISION A(I,J), B(J,K), C(I,K)

C

DO 100 L = 1,I

C

DO 100 M = 1,K

C

C(L,H) = 0.

C

DO 100 N = 1,J

C

100 C(L,H) = C(L,H) + A(L,N) * B(N,M)

C

RETURN

C

END

*OPTIONS IN EFFECT* NAME(MAIN) OPTIMIZE(3) LINENUMBER(60) SIZE(MAX) AUTOCLASS(DBL4)

*OPTIONS IN EFFECT* SOURCE EBCDIC NOTLIST NODECK OBJECT NOHAP NOFORMAT EOSTIT NOXREF NOALC NOANSF TERM IBM FLAG(I)

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

*STATISTICS* NO DIAGNOSTICS GENERATED

****** END OF COMPILED ******

3016K BYTES OF CORE NOT USED
SUBROUTINE MATPRN(A,I,J,TITLE)

DOUBLE PRECISION A(I,J)

MATRIX OUTPUT

WRITE(6,1) TITLE

1 FORMAT(/,5X,'MATRIX OUTPUT FOR ',A4,/)  

WRITE(6,2)

2 FORMAT(5X,'ROW',5X,'COL',5X,'VALUE',/)

DO 100 K = 1,1

DO 100 L = I,J

IF(A(K,L).EQ.0.) GO TO 100

WRITE(6,3) K,L,A(K,L)

100 CONTINUE

3 FORMAT(5X, I5,3X,I5,3X,E12.5)

RETURN

END
SUBROUTINE MINV

PURPOSE

INVERT A MATRIX

USAGE

CALL MINV(A,N,D,L,M)

DESCRIPTION OF PARAMETERS

A - INPUT MATRIX, DESTROYED IN COMPUTATION AND REPLACED BY
RESULTANT INVERSE.
N - ORDER OF MATRIX A
D - RESULTANT DETERMINANT
L - WORK VECTOR OF LENGTH N
H - WORK VECTOR OF LENGTH N

REMARKS

MATRIX A MUST BE A GENERAL MATRIX
SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
NONE
METHOD

THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT
IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT
THE MATRIX IS SINGULAR.

DIMENSION A(I),L(I),M(I)

IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE
C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION
STATEMENT WHICH Follows.

DOUBLE PRECISION A,D,BIGA,HOLD
THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS
APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS
ROUTINE.
THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO
CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS. DBDS IN STATEMENT
10 MUST BE CHANGED TO DDABS.
Search for largest element

I = 1.0
DO 20 K = 1, N
K = K + 1
BIGA = A(KK)
DO 20 I = K, N
IJ = IZ + I
IF (ABS(BIGA) > DABS(A(IJ))) 15, 20, 20
BIGA = A(IJ)
K = I
J = K
INTERCHANGE ROWS
DO 30 I = I, N
KI = K - I
HOLD = A(KI)
A(KI) = A(JI)
A(JI) = HOLD
30 CONTINUE

INTERCHANGE COLUMNS
J = L(K)
IF (J - K) 35, 35, 25
KI = K - N
DO 35 I = 1, N
KI = K + I
35 A(KI) = HOLD

DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS CONTAINED IN BIGA)

DO 40 I = 1, N
40 A(JI) = HOLD
40 CONTINUE

REDUCE MATRIX

DO 65 I = 1, N
65 EXIT
**FORTRAN EXTENDED**

**VERSIO 1.3.0**

**SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)**

**DATE** 82.14/10.49.22

---

ISH 0049  \( \text{IK}=\text{IK}+1 \)

ISH 0050  \( \text{HOLD}=\text{A(IK)} \)

ISH 0051  \( \text{IJ}=\text{IJ}+1 \)

ISH 0052  \( \text{DO} 60 \text{ J}=1, \text{N} \)

ISH 0053  \( \text{IK}=\text{IK}+1 \)

ISH 0054  \( \text{IF}(\text{IK}) \geq 60, \text{STOP} \)

ISH 0055  \( \text{IF}(\text{IK}) = 60, \text{STOP} \)

ISH 0056  \( \text{KJ}=\text{IJ}+1 \)

ISH 0057  \( \text{A(IIJ)}=\text{HOLD} \cdot \text{A(KJ)}+\text{A(IIJ)} \)

ISH 0058  \( 65 \text{ CONTINUE} \)

---

**C**

**DIVIDE ROW BY PIVOT**

**C**

ISH 0059  \( \text{KJ}=\text{K}-1 \)

ISH 0060  \( \text{DO} 75 \text{ J}=1, \text{N} \)

ISH 0061  \( \text{KJ}=\text{KJ}+1 \)

ISH 0062  \( \text{IF}(\text{KJ}) \geq 70, \text{STOP} \)

ISH 0063  \( \text{A(IIJ)}=\text{A(KJ)} / \text{BIGA} \)

ISH 0064  \( 75 \text{ CONTINUE} \)

---

**C**

**PRODUCT OF PIVOTS**

**C**

ISH 0065  \( \text{D}=\text{D} \cdot \text{BIGA} \)

---

**C**

**REPLACE PIVOT BY RECIPROCAL**

**C**

ISH 0066  \( \text{A(KK)}=1.0 / \text{BIGA} \)

ISH 0067  \( 80 \text{ CONTINUE} \)

---

**C**

**FINAL ROW AND COLUMN INTERCHANGE**

**C**

ISH 0068  \( \text{K}=\text{N} \)

ISH 0069  \( 100 \text{ K}=-(\text{K}-1) \)

ISH 0070  \( \text{IF}(\text{K}) \geq 150, \text{STOP} \)

ISH 0071  \( 105 \text{ I}=\text{L}(\text{K}) \)

ISH 0072  \( \text{IF}(\text{I}) \geq 120, \text{STOP} \)

ISH 0073  \( 108 \text{ JQ}=\text{H}(\text{I}-1) \)

ISH 0074  \( \text{JR}=\text{H}(\text{I}-1) \)

ISH 0075  \( \text{DO} 110 \text{ J}=1, \text{N} \)

ISH 0076  \( \text{JK}=\text{JQ}+\text{J} \)

ISH 0077  \( \text{HOLD}=\text{A(IIJ)} \)

ISH 0078  \( \text{JI}=\text{JR}+\text{J} \)

ISH 0079  \( \text{A(IIJ)}=-\text{A(IIJ)} \)

ISH 0080  \( 110 \text{ A(IIJ)}=\text{HOLD} \)

ISH 0081  \( 120 \text{ JK}=\text{JQ}+\text{J} \)

ISH 0082  \( \text{IF}(\text{JK}) \geq 100, \text{STOP} \)

ISH 0083  \( 125 \text{ KI}=\text{K}-1 \)

ISH 0084  \( \text{DO} 130 \text{ I}=1, \text{N} \)

ISH 0085  \( \text{KJ}=\text{KJ}+1 \)

ISH 0086  \( \text{HOLD}=\text{A(KI)} \)

ISH 0087  \( \text{JI}=\text{K}-\text{KJ} \)

ISH 0088  \( \text{A(KI)}=-\text{A(KI)} \)

ISH 0089  \( 130 \text{ A(KI)}=\text{HOLD} \)

ISH 0090  \( \text{GO TO} 100 \)

ISH 0091  \( 150 \text{ RETURN} \)

ISH 0092  \( \text{END} \)

---

**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 = 91, PROGRAM SIZE = 1502, SUBPROGRAM NAME = MINV**
SUBROUTINE MODMAS(I,EX,NR,RMASS)

IMPLICIT REAL*8(A-H,O-Z)

C MODAL MASS FOR THE I MODE

COMMON /STIFF/ SK(12,12),SKK(12,12,21),SMN(6,6,21)

DIMENSION EX(12,21,2),RMASS(3),DISP(6,1),SMN(6,6),DISPT(1,6)

DIMENSION OL(6,1)

RMASH(I) = 0.

DO 100 J = 1,NR

C CONVERT THE W137 EIGENVECTOR, EX, TO STABLE FORMAT

CALL SHAPE(J,EX,DISP)

NOW MULTIPLY - DISPT * SMM * DISP

110 SMM(K,L) = SMM(K,L,J)

DO 120 K = 1,6

120 DISPT(I,K) = DISP(K,1)

CALL HATMPY(SMM,DISP,D1,6,6,1)

CALL HATMPY(DISPT,D1,D2,1,6,1)

RMASH(I) = RMASH(I) + D2

RETURN

END
C SUBROUTINE MODSTR (I, NC, NR) 
C IMPLICIT REAL*8(A-H,O-Z) 
C THIS ROUTINE IS THE DRIVER FOR THE MODAL STRESS CALCULATION 
C COMMON /CALCA/ FILL(1468),MS(2),FILL(588),EX(12,21,2),SS(12,2) 
C COMMON /STRS/ RE(12,12,25,21),OUT(12,21),SL(25,21),SH(25,21) 
C COMMON /MODAL/ SIGMA(3,6,25) 
C COMMON /MODAL1/ STRESS(3,3,6,25),FACTOR(3),RMASS(3) 
C COMMON /LM/ DIAGON(100),FIN(100),FILLI(300) 
C DIMENSION DISP(6,1),BFORCE(6,1),DISP(1,6),PLOAD(1,1) 
C CONVER#137 EIGENVECTOR TO STABEl FORMAT FOR STATIONS 1 AND 2 
C CALL SHAPE(2,EX,DISP) 
C CALL MATFRN(DISP,6,1,'DISP') 
C FOUT(J,2) = DISP(J,1) 
C CALL SHAPE(3,EX,DISP) 
C CALL MATFRN(DISP,6,1,'DISP') 
C FOUT(J,3) = DISP(J,1) 
C CALCULATE ROOT STRESS FOR MODE I 
C CALL ZSTRES(2,NC,SHAX) 
C STORE THE STRESS VALUES FOR MODE I IN STRESS(3,3,6,25) 
C DO 100 J = 1, NC 
C DO 100 J = 1,6 
C DO 120 JJ = 1, NC 
C DO 120 JJ = 1,6 
C STRESS(I,K,J,JJ) = SIGMA(K,J,JJ) 
C CALCULATE THE MODAL MASS FOR MODE I 
C CALL MODMAS (I,EX,NR,RMASS) 
C CALCULATE THE BIRD FORCE AND MOMENT 
C CALL BIRD(NR,NC,ISTA,BFORCE,SQUASH) 
C FIND THE STABEl MODE SHAPE AT STATION ISTA 
C CALL SHAPE (ISTA,EX,DISP) 
C CALL MATFRN(DISP,6,1,'DISP') 
C DO 130 J = 1,6 
C DISP(J,1) = DISP(J,1) 
C MODAL LOAD, PLOAD 
C
CALL MATHPY (DISPT,BFORCE,PLOAD,1,6,1)
CALL MATPRN (PLOAD,1,1,'PLOAD')
PARTICIPATION FACTOR, FACTOR
PI = PLOAD(1,1)
TO = SQUASH
FREQ = FN1(I) * 2. * 3.141593
FACTOR(I) = PI * TO / RMASS(I) / FREQ
CALL MATPRN(FN1,100,1,'FN1 ')
FACT = FN1(I)
CALL MATPRN(RMASS,3,1,'RMASS')
CALL MATPRN(FACTOR,3,1,'FACT')
RETURN
END
SUBROUTINE PARTN(SK,II,JJ,SK11,SK12,SK21,SK22,KK,LL)

THIS ROUTINE PARTITIONS THE SK MATRIX INTO SK11,SK12,
SK21 AND SK22

DIMENSION SK(II,JJ),SK11(II,LL),SK12(II,LL),SK21(II,LL),SK22(II,LL)

DO 100 I = 1,6
   DO 100 J = 1,6
      SK11(I,J) = SK(I,J)
   CONTINUE

DO 101 I = 1,6
   L = 1
   DO 101 J = 7,12
      SK12(I,L) = SK(I,J)
      L = L + 1
   CONTINUE

DO 102 I = 7,12
   K = I - 6
   DO 102 J = 1,6
      SK21(K,J) = SK(I,J)
   CONTINUE

DO 103 I = 7,12
   K = I - 6
   DO 103 J = 7,12
      SK22(K,L) = SK(I,J)
      L = L + 1
   CONTINUE

CALL MATPRH(SK11,6,6,'SK11')
CALL MATPRH(SK12,6,6,'SK12')
CALL MATPRH(SK21,6,6,'SK21')
CALL MATPRH(SK22,6,6,'SK22')

RETURN
END
SUBROUTINE PPl(ii)
THIS ROUTINE SETS UP THE REORDER MATRICES
P AND P1 ARE CONSTANT AND DO NOT CHANGE

IMPLICIT REAL*8(A-H,O-Z)
COMMON /REORDER/ P(12,12),PT(12,12)

INITIALIZE
DO 100 I = 1,12
DO 100 J = 1,12
100 P(I,J) = 0.
IF(ii .GT. 1) GO TO 50
SET NON-ZERO VALUES OF P
P(1,1) = 1.
P(2,5) = 1.
P(3,9) = 1.
P(4,2) = 1.
P(5,10) = 1.
P(6,6) = 1.
P(7,3) = 1.
P(8,7) = 1.
P(9,11) = 1.
P(10,4) = 1.
P(11,12) = 1.
P(12,8) = 1.
GO TO 51
50 P(1,7) = 1.
P(2,6) = 1.
P(3,12) = -1.
P(4,9) = -1.
P(5,11) = 1.
P(6,10) = 1.
P(7,1) = 1.
P(8,6) = -1.
P(9,2) = 1.
P(10,5) = 1.
P(11,3) = -1.
P(12,4) = 1.
GO TO 51
NOW DETERMINE THE TRANSPOSE MATRIX, PT
DO 200 I = 1,12
DO 200 J = 1,12
PT(I,J) = P(J,I)
200 CONTINUE
CALL MATPRN(PT,12,12,' PT ')
CALL MATPRN(P,12,12,' P ')

+VERSION 1.3.0 (01 MAY 80)  PPI  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 82.141/10.49.51  PAGE 2
ISH 0040  RETURN
ISH 0041  END

*OPTIONS IN EFFECT*NAME(HAIN) OPTIMIZE(3) LIONCOUNT(60) SIZE(MAX) AUTOLOB(0BL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOCPAP NOFORMAT GOSTHT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS =  48, PROGRAM SIZE =  534, SUBPROGRAM NAME =  PPI
*STATISTICS* NO DIAGNOSTICS GENERATED
****** END OF COMPILATION ******

3016K BYTES OF CORE NOT USED
SUBROUTINE RBE2(X,Y,Z,ISEC,IBEAM,NC,BLOCAL)
THIS ROUTINE CALCULATES THE RIGID BODY TRANSFORMATION
MATRICES RBL12 AND RBL1
RBL12 IS WRITTEN IN THE LOCAL BEAM COORD. SYSTEM
IMPLICIT REAL*8(A-H,O-Z)
COMMON /RBE/ RBL12(12,12),RBL4(12,12)
COMMON /HARP/ HS(26,21),RH(25,21),RT(25,21)
COMMON /SC/ XSC(21),YSC(21),ZSC(21),ALPHA(21),SECIP(21),SECA(21)
DIMENSION RBL(6,6),RL(6),RH(6)
DIMENSION X1(1),Y1(1),Z1(X3),XG(3),XH(3)
DIMENSION XA(3),XB1(3),XB2(3),XGB(3),BLOCAL(3,3)
DIMENSION V1(3),V2(3),VL1(3),VL2(3)
FORMING RBL12
TRANSFORM GLOBAL COORD. POINTS TO SECTION SYSTEM
LABEL 1-BEAM COORD IS XAI,SECTION COORD IS XBL
IP = (ISEC-1) * NC + IBEAM
POSITION VECTORS FOR LAYERS 1 AND 2 , V1 AND V2
V1(1) = X(IP) - XSC(IP)
V1(2) = Y(IP) - YSC(IP)
V1(3) = Z(IP) - ZSC(IP)
V2(1) = X(IP+NC) - XSC(IP+1)
V2(2) = Y(IP+NC) - YSC(IP+1)
V2(3) = Z(IP+NC) - ZSC(IP+1)
ROTATE FROM GLOBAL SYSTEM TO BEAM LOCAL
CALL MATMPY(BLOCAL,V1,VL1,3,3,1)
CALL MATMPY(BLOCAL,V2,VL2,3,3,1)
X1 = VL1(1)
Y1 = VL1(2)
Z1 = VL1(3)
X2 = VL2(1)
Y2 = VL2(2)
Z2 = VL2(3)
XL = XSC(IP+1) - XSC(IP)
DO 102 I = 1,12
DO 102 J = 1,12
RBL1(I,J) = 0.
RBL12(I,J) = 0.
DO 103 I = 1,12
RBL1(I,I) = 1.
**+VERSION 1.3.0 (01 MAY 80) RBE2 SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)**

**ISH 0033**  
103 RBL12(I,J) = 1.

* C

**ISH 0034**  
RBL12(1,4) = -WS(Ibeam,Isec) / XL

**ISH 0035**  
RBL12(1,5) = Z1

**ISH 0036**  
RBL12(1,6) = -Y1

**ISH 0037**  
RBL12(1,10) = -RBL12(1,4)

**ISH 0038**  
RBL12(2,4) = -Z1

**ISH 0039**  
RBL12(2,6) = X1

**ISH 0040**  
RBL12(3,4) = Y1

**ISH 0041**  
RBL12(3,5) = -X1

**ISH 0042**  
RBL12(5,4) = RT(Ibeam,Isec) / XL

**ISH 0043**  
RBL12(5,10) = -RBL12(5,4)

**ISH 0044**  
RBL12(6,4) = RN(Ibeam,Isec) / XL

**ISH 0045**  
RBL12(6,10) = -RBL12(6,4)

**ISH 0046**  
RBL12(7,4) = -WS(Ibeam,Isec+1) / XL

**ISH 0047**  
RBL12(7,10) = -RBL12(7,4)

**ISH 0048**  
RBL12(7,11) = Z2

**ISH 0049**  
RBL12(7,12) = -Y2

**ISH 0050**  
RBL12(6,10) = -Z2

**ISH 0051**  
RBL12(6,12) = X2

**ISH 0052**  
RBL12(6,9) = Y2

**ISH 0053**  
RBL12(9,11) = -X2

**ISH 0054**  
RBL12(11,4) = RT(Ibeam,Isec+1) / XL

**ISH 0055**  
RBL12(11,10) = -RBL12(11,4)

**ISH 0056**  
RBL12(12,4) = RN(Ibeam,Isec+1) / XL

**ISH 0057**  
RBL12(12,10) = -RBL12(12,4)

* C

**ISH 0058**  
CALL MATPRMN(RBL12,12,12,'RBL12')

**ISH 0059**  
RBL1(1,5) = RBL1(1,5)

**ISH 0060**  
RBL1(1,6) = RBL1(1,6)

**ISH 0061**  
RBL1(2,4) = RBL1(2,4)

**ISH 0062**  
RBL1(2,6) = RBL1(2,6)

**ISH 0063**  
RBL1(3,4) = RBL1(3,4)

**ISH 0064**  
RBL1(3,5) = RBL1(3,5)

**ISH 0065**  
RBL1(7,11) = RBL1(7,11)

**ISH 0066**  
RBL1(7,12) = RBL1(7,12)

**ISH 0067**  
RBL1(8,10) = RBL1(8,10)

**ISH 0068**  
RBL1(9,10) = RBL1(9,10)

**ISH 0069**  
RBL1(9,11) = RBL1(9,11)

* C

**ISH 0070**  
IF( Isec .EQ. 2 ) GO TO 300

**ISH 0071**  
IF( Isec .GT. 1 ) RETURN

* C

**ISH 0072**  
RESET RBL12 FOR EXTENDED NECK REGION - NO WARPING

* C

**ISH 0073**  
DO 200 I = 1,12

**ISH 0074**  
DO 200 J = 1,12

**ISH 0075**  
DO 200 J = 1,12

**ISH 0076**  
DO 200 I = 1,12

**ISH 0077**  
DO 200 I = 1,6

**ISH 0078**  
DO 200 J = 1,12

**ISH 0079**  
DO 305 I = 1,6

**ISH 0080**  
DO 305 J = 1,12

**ISH 0081**  
DO 305 I = 1,6

**ISH 0082**  
DO 305 J = 1,12

**ISH 0083**  
DO 305 I = 1,6

**ISH 0084**  
DO 305 J = 1,12

**ISH 0085**  
RETURN

* C

**ISH 0086**  
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 NOSREF NOSLF TERM IBM FLAG(I)
+VERSION 1.3.0 (01 MAY 80) RBE2 SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 82.141/10.49.55 PAGE 3

*STATISTICS* SOURCE STATEMENTS = 61, PROGRAM SIZE = 1496, SUBPROGRAM NAME = RBE2

*STATISTICS* NO DIAGNOSTICS GENERATED

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

3004K BYTES OF CORE NOT USED
C SUBROUTINE SECTH(NR,NC,NRM)
C THIS ROUTINE CALCULATES THE SECTION STIFFNESS MATRIX, SKK AND SK
C THE SECTION STIFFNESSES ARE SUMMED FROM THE LOCAL BEAMS
C THE SECTION MASS IS ALSO DETERMINED FROM THE BEAM MASSES
C ALSO THE SECTION CENTRIFUGAL RESTORING MATRIOS,CHNG, IS FOUND.
C
C IMPLICIT REAL*8(A-H,O-Z)
C COMMON /SC,XSC(21),YSC(21),ZSC(21),ALPHA(21),SECP(21),SEC(21)
C COMMON /RBE/RBL(12,12),RBIL(12,12)
C COMMON /RBE/RBE(12,12,25,21),FOUT(12,21),SL(25,21),SH(25,21)
C COMMON /COORD//CLOCAL(3,3,100)
C COMMON /BSTIFF//BK(12,12),EM(12,12)
C COMMON /STATC//SH(12,12,20),F(12,21)
C COMMON /LAYR//TH(7,25),BHASS(25),BSPAN(25),BHWIDTH(25)
C COMMON /STFS//RASS(12,12),R1(12,12),R1(12,12)
C COMMON /STIFF//SK(12,12),SKK(12,12,21),CM(6,6,21)
C COMMON /DSTIFF//CHNG(12,12,20),DSK(12,12,20)
C COMMON /INPUTA//X(1000),Y(1000),Z(1000),T(1000)
C COMMON /BLK &//SHM(21),ITLLE(18),VARI(235),TMAX(21),HALPHA(21)
C COMMON /ANAL33//DUN(6),NCD,HCD,DUN(4)
C DIMENSION BLOCAL(3,3),SHM(6,6),SM(6,6),SMOLD(6,6)
C COMMON /SHM(12,12),FBEAHM(6,1),FBEAMH(6,1),FOLD(6,1)
C COMMON /TRANS//TS(3,3,21)
C DIMENSION D1(12,12),D2(12,12),SKKOLD(12,12),TSI(3,3),TS(3,3)
C DIMENSION FF(12,12),FSEC(12,12),FOTO(12,12),FOTO(12,12)
C DIMENSION RESG(12,12),CMG(12,12),TBS1(3,3),TBS2(3,3)
C DIMENSION RF(12,12)
C BEGIN THE SECTION STIFFNESS SOLUTION FOR SECTION I
C FORM THE SECTION COORDINATE SYSTEM ,TS
C TSI FOR LAYER ONE
C TSZ FOR LAYER TSO
C LOCAL X UNIT VECTOR
C
C IP = (I-1) * NC + 1
C IP = IP + NC / 2
C IPP = IP + NC
C NRM = NR - 1
C
C XSY = XSC(I+1) - XSC(I)
C YSY = YSC(I+1) - YSC(I)
C ZSY = ZSC(I+1) - ZSC(I)
C WRITE(6,6) XSY(II),YSC(II),ZSC(II),XSC(II+1),YSC(II+1),ZSC(II+1)
C FORMAT(5X,6E12.5)
C XSMAG = DSGRT( XSY**2 + YSY**2 + ZSY**2 )
C WRITE(6,4) XSMAG,II
C FORMAT(5X,'MAGNITUDE',E12.5,E15)
C XSY = XSY / XSMAG
LOCAL Z UNIT VECTOR (INITIALLY)

ZSX = 0.

LOCAL Y UNIT VECTOR YS = ZS CROSS XS

YSX = ZSY * XSZ - ZSZ * XSY
YSZ = ZSX * XSY - ZSY * XSX

WRITE(6,4) YSHAG, I, II

REDEFINE THE LOCAL ZS VECTOR, ZS = XS CROSS YS

IF(I .EQ. NRMI .AND. NRM1 .NE. 1) II = II + 1

98 TS(5,1,II) = XSY * YSZ - XSZ * YSX
TS(5,2,II) = XSZ * YSX - XSX * YSZ
TS(5,3,II) = XSY * YSX - XSX * YSZ

FILL THE TS ARRAY

TS(1,1,II) = XSX
TS(1,2,II) = XSY
TS(1,3,II) = XSZ
TS(2,1,II) = YSX
TS(2,2,II) = YSY
TS(2,3,II) = YSZ

DO 300 L = 1,3
DO 300 M = 1,3
300 IF(DABS(TS(L,M,II)) .LE. .001) TS(L,M,II) = 0.

WRITE(6,2) I
FORMAT(5X,'TS MATRIX FOR SECTION',I5,/) II = I+1
C
DD 161 L = I,II
WRITE(6,162) L
C
C162 FORMAT(5X,'SECTION NODE',I5)
C161 WRITE(6,1) ((TS(J,K,L),K=1,3),J=1,3)
C1 FORMAT(IX,3(E12.5,2X),/) INITIALIZE THE SK AND SM MATRIX AND LOAD VECTORS, FTOT1,FTOT2

DO 100 J = 1,12
DO 100 K = 1,12
CHG(J,K) = 0.
SMJ,K) = 0.
I,  

BEGIN LOOP FOR EACH BEAM 

DO 200 J = 1,NC 

CALCULATE THE BEAM COORDINATE SYSTEM ,BLOCAL(3,3) 

IP = (I-1) * NC + J 

IPP = IP + NC 

CALL BCOORD(IP,IPP,BLOCAL) 

CALL MATRN(BLOCAL,3,3,'BLOC') 

CALL THE BEAM MATERIAL STIFFNESS MATRIX , E 

BEGIN LOOP FOR EACH BEAM 

IP = (I-I) * NC + J 

IPP = IP + NC 

CALL ESTIFF(IP,NC,IP,J) 

DETERMINE THE BEAM STIFFNESS MATRIX , BK 

THE BEAM CENTRIFUGAL RESTORING MATRIX, BCI"GG 

CALL BEAMNC,J,BLOCAL,BCI"GG,RPtD 

THE BEAM STIFFNESS MATRIX IS BUILT AND MUST 

BE SUMMED INTO THE SECTION (I) STIFFNESS MATRIX ,SK 

CALL THE TBS TRANSFORMATION MATRIX 

TBS = BLOCAL * TS(TRANSPOSE) 

DO 103 K = 1,3 

DO 103 L = 1,3 

WRITE(6,1) I 

WRITE(6,3) I 

WRITE(6,14) 

THE RIGID BODY TRANSFORMATION MATRIX , RBL1 

DO NOT INCLUDE WARPING- ONLY EQUILIBRIUM , RBL1 

DO NOT INCLUDE WARPING- ONLY EQUILIBRIUM , RBL1 

DO NOT INCLUDE WARPING- ONLY EQUILIBRIUM , RBL1 

READ INPUT IF I .EQ. 1) GO TO 190 

RESTRANIED WARPING AT THE ROOT 

IF (I .EQ. 1) GO TO 190 

GO TO 191 

IWARP = 1 

IF (IWARP .GT. 1) GO TO 191 

WRITE(6,77) 

FOR(I = 1) GO TO 200 

RETURN 

RBL12(1,4) = 0. 

RBL12(1,10) = 0. 

RETURN
DEFINE THE TBSS MATRIX

C

DO 105 K = 1,12
DO 105 L = 1,12
TBSS(K,L) = 0.

C

IROW = 1
DO 106 K = 1,4
DO 106 L = 1,3

C

ICOL = (K-1) * 3 + 1

C

DO 108 M = 1,3

C

TBS(IRA,W,ICOL) = TBS1(L,M)

C

IF(K.GT.2) TBSS(IRW,ICOL) = TBS2(L,M)

C

ICOL = ICOL + 1

C

WRITE(6,51 I
C
C5 FORMAT(15X,'TBSS HATRIX FOR SECTION',15)
C
CALL MATPRN(TBSS,12,12,'TBSS')

C

CALCULATE THE R MATRIX, R = RBL12 * TBSS

C

CALL MATMPY(RBL12,TBSS,R,12,12,12)

C

CALL MATPRN(R,12,12,' R')

C

FILL THE RBE ARRAY FOR LATER USE IN THE STATIC ANALYSIS

C

ALSO STORE THE INFO FOR BEAM DEDIMENSIONS

C

DO 80 K = 1,12
DO 80 L = 1,12
RBE(K,L,J,I) = R(K,L)

C

SL(J,I) = BSPAN(J)

C

SW(J,I) = BWIDTH(J)

C

CALCULATE R1 = TSS * RBL1 * TBB

C

IT HAS ALSO BEEN SHOWN THAT R1 = R(TRANSPOSE)

C

ALSO TRANSPOSE RF AND STORE IN RBL1

C

DO 201 K = 1,12
DO 201 L = 1,12
RBL1(K,L) = RF(L,K)

C

CALL MATPRN(R1,12,12,'R1 ')

C

CALL MATMPY(RBL1,12,12,'RF ')

C

BEAM(J) CONTRIBUTION TO SECTION (I) STIFFNESS IS:

C

SK(K,L) = R1(K,M) * BK(M,N) * R(N,L)

C

BEAM(J) CONTRIBUTION TO SECTION (I) MASS , SM
DATE 02.141/10.50.02  PAGE 5

C SH(K,L) = RI(K,M) * BM(N,N) * R(N,L)
C
C ISN 0133 CALL MATMPY(BM,R,01,12,12,12)
C ISN 0134 CALL MATMPY(R1,01,02,12,12,12)
C C BEAM(J) CONTRIBUTION TO SECTION(I) CENTRIFUGAL RESTORING STIFFNESS MATRIX, CMGG
C
C ISN 0135 CALL MATMPY(BCMGG,R,01,12,12,12)
C ISN 0135 CALL MATMPY(R1,01,BCMGG,12,12,12)
C C BEAM(J) CONTRIBUTION TO SECTION(I) - RPM LOAD END 1
C
C ISN 0137 IP = (I-1) * NC + J
C ISN 0138 CALL FRPM(J,IP,X,Y,BMASS,RPM,FBEAM1,BLOCAL1)
C C END 2
C
C ISN 0139 IP = IP * NC
C ISN 0140 CALL FRPM(J,IP,X,Y,BMASS,RPM,FBEAM2,BLOCAL1)
C C TRANSFER TO SECTION COORDINATE SYSTEM
C
C ISN 0141 DO 170 K = 1,6
C ISN 0142 FSEC(K,1) = FBEAM1(K,1)
C ISN 0143 170 FSEC(K+6,1) = FBEAM2(K,1)
C
C ISN 0144 CALL MATMPY(RBL1,FSEC,FF,12,12,12)
C ISN 0145 DO 175 K = 1,6
C ISN 0146 FTOT1(K,1) = FTOT1(K,1) + FF(K,1)
C ISN 0147 175 FTOT2(K,1) = FTOT2(K,1) + FF(K+6,1)
C C SUM UP CONTRIBUTION OF BEAM(J) FOR SECTION(I)
C
C ISN 0148 DO 101 K = 1,12
C ISN 0149 DO 101 L = 1,12
C ISN 0150 CMG(K,L) = CMG(K,L) + BCMGG(K,L)
C ISN 0151 SK(K,L) = SK(K,L) + D2(K,L)
C ISN 0152 101 SK(K,L) = SK(K,L) + SKOLD(K,L)
C
C WRITE(6,13) I,J
C C13 FORMAT(1X,'THE SECTION STIFFNESS AND MASS FOR SECTION',I5,',',/)
C C 15X,'FOR BEAMS ONE THRU ',IS5)
C C CALL MATPRN(SK,12,12,' SK ')
C C WRITE(6,21)
C C21 FORMAT(1X,'SECTION MASS ',SM,/,)
C C CALL MATPRN(SH,12,12,' SM ')
C C
C ISN 0153 200 CONTINUE
C C CALL MATPRN(CMG,12,12,' CMG ')
C C CALL MATPRN(FTOT1,6,1,'FT1 ')
C C WARPING FUNCTION UPDATE TO ACCOUNT FOR RESTRAINT EFFECTS
C C
A = 0.5 * SBS(2)

ISN 0155

XDIST = XSC(I) - XSC(2)

ISN 0156

ARG = - XDIST / A

ISN 0157

WARP1 = 1. - DEXP(A) 

ISN 0158

IF(I .LE. 2) WARP1 = 1.

ISN 0159

XDIST = XSC(I+1) - XSC(2)

ISN 0160

ARG = - XDIST / A

ISN 0161

WARP2 = 1. - DEXP(A)

ISN 0162

IF(I .EQ. 1) WARP2 = 1.

ISN 0163

C

SK(4,4) = SK(4,4) / WARP1

ISN 0165

SK(4,10) = SK(4,10) / WARP1

ISN 0166

SK(10,4) = SK(10,4) / WARP2

ISN 0167

SK(10,10) = SK(10,10) / WARP2

C

REPLACE EXTENDED NECK STIFFNESS WITH RECTANGULAR SECTION NECK

ISN 0169

C

IF(I .EQ. 1 .AND. NCO .EQ. 1) CALL EXNECK(SBS(2))

C

C INSERT SK INTO SKK FOR STORAGE

ISN 0171

DO 110 J = 1,12

DO 110 K = 1,12

CNGG(J,K,I) = CMG(J,K)

110

SKK(J,K,I) = SK(J,K)

C

FILL IN THE MASS MATRIX, SH1, AND THE LOAD VECTOR, F.

ISN 0173

DO 115 J = 1,6

DO 115 K = 1,6

SM1(J,K) = SM(J,K)

115

SM2(J,K) = SM(J,K)

C

WRITE(6,30)

C30 FORMAT(/,5X,'PUTTING MASS INTO SM')

C CALL MATFRN(SM1,6,6,'SH1 ')

C CALL MATFRN(SM2,6,6,'SH2 ')

C

HR1 = HR - 1

ISN 0174

IF(I .NE. 1) GO TO 125

C

DO 116 J = 1,6

F(J,I) = 0.

ISN 0175

F(J+6,I) = FBEAM1(J,1)

ISN 0176

FOLD(J,1) = FBEAM2(J,1)

ISN 0177

DO 116 K = 1,6

SM1(J,K,I) = SM1(J,K)

116

SM2OLD(J,K) = SM2(J,K)

ISN 0178

GO TO 126

ISN 0179

125 CONTINUE

ISN 0180

DO 118 J = 1,6

F(J,I) = 0.

ISN 0181

F(J+6,I) = FOLD(J,1) + FBEAM1(J,1)

ISN 0182

FOLD(J,1) = FBEAM2(J,1)

ISN 0183

DO 118 K = 1,6

ISN 0184

END

+VERSION 1.3.0 (01 MAY 80) SECTN SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 02.14/10.50.02 PAGE 6

142
*VERSION 1.3.0 (01 MAY 80) SECTN SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 82.141/10.50.02 PAGE 7

ISH 0201      SIM(J,K,I) = SH2OLD(J,K) + SH1(J,K)   02169
ISH 0202      118 SH2OLD(J,K) = SH2(J,K)            02169
ISH 0203      126 IF(I .NE. NRH1) RETURN            02190
ISH 0205      DO 119 J = 1,6                        02192
ISH 0206      DO 119 K = 1,6                        02193
ISH 0207      F(J,I+1) = 0.                          02194
ISH 0208      F(J+6,I+1) = FBEAM2(J,1)               02195
ISH 0209      119 SHH(J,K,I+1) = SH2(J,K)            02196
              C
              C ADD A TIP MOMENT                         02197
              C
              C F(10,I+1) = 1000.                         02198
              C
              C  RETURN                                    02199
ISH 0210      RETURN                                 02200
ISH 0211      END                                    02201

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NODLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALLOC NOAHSF 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
C

SUBROUTINE SHAPE (NODE, EX, DISP)
IMPLICIT REAL*8(A-H,O-Z)

GIVEN THE WI37 EIGENVECTOR CALCULATE THE STABLE SYSTEM EIGENVECTOR, FOR A GIVEN NODE

COHOM /REORD/ P(12,12),PT(12,12)
DIMENSION EX(12,21,2),DISP(6,1),WI37(6,1),PPT(6,6)

EXTRACT THE WI37 EIGENVECTOR COMPONENT FOR THE NODE

DO 100 I = 1,6
WI37(I,1) = EX(I+6,NODE,1)
CALL MATPRN(WI37,6,1,'WI37')

TRANSFORMATION ARRAY, PPT

CALL PPI(2)

DO 110 I = 1,6
DO 110 J = 1,6
PPT(I,J) = PT(I,J+6)
CALL MATPRN(PPT,6,6,' PPT')

CALL MATMPY(PPT, WI37, DISP, 6, 6, 1)
RETURN
END

*OPTIONS IN EFFECT* NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOOBL(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
SUBROUTINE SHMNT(NR)

A SLAVE ROUTINE TO PERFORM THE HOLZER STIFFNESS MATRIX STRING MULTIPLICATION FOR THE STATIC ANALYSIS

IMPLICIT REAL*8(A-H,O-Z)

COMMON /STATC/ SH(12,12,201),F(12,21)

COMMON /HSTIFF/ SKP(12,12),SKPP(12,12,21)

DIMENSION D1(12,12),D2(12,12),D3(12,12)

C INITIALIZE SH MATRIX

NR = NR - 1

DO 100 I = 1,12
    DO 100 J = 1,12
        SH(I,J,1) = SKPP(I,J,NR)

100 CONTINUE

IF(NR .EQ. 1) RETURN

DO 200 I = 2,NR
    IP = NR - I + 1
    DO 110 J = 1,12
        DO 110 K = 1,12
            SH(I,J,K) = SH(J,K,IP)

110 CONTINUE

CALL MATMPY(D1,D2,D3,12,12,12)

CALL MATMPH(D1,D2,D3,12,12,12)

WRITE(6,10)

10 FORMAT(5X,'IN SHMNT FOR STRING OPERATION')

C CALL MATMPH(D12,12,12,'D3')

C

CONTINUE

RETURN
SUBROUTINE STABEUNR(NC,RPM1,RPM2,RPM3,FREQ,LOOP,ISERCH)

C THIS PROGRAM IS THE DRIVER FOR A STACKED BEAM ANALYSIS
C A STATIC ANALYSIS IS PERFORMED WITH THE HOLZER METHOD
C WRITTEN BY NICK MARTIN AND KEN BROWN

IMPLICIT REAL*8(A-H,O-Z)
COMION /INPUTT/ XC 1000, YH 1000, ZC 1000, T(1000)
COMION /SC/ CC6,6), RPM
COMION /STIFF/ SK(12,12), SKK(12,12,21), SM(6,6,21)
COMION /DSTIFF/ CG(12,12,20), DSK(12,12,20)
COMION /SC/ XSC(21), YSC(21), ZSC(21), ALPHA(21), SECIP(21), SECA(21)
COMION /HSTIFF/ SKP(12,12), SKPP(12,12,21)
COMION /BLKAC/ AA(12,12,20), BB(12,12,20), CC(12,12,20),

1 STAHAT(12,12,20)

COMMON /REORD/ F(12,12), PT(12,12)
COMMON /STATC/ SH(12,12,20), F(12,21)
COMMON /HOLE/ HOLE
COMMON /SHMX/ SHMX(8), SMX2(8), SMX3(8), SMAXE(8), SMAXTE(8)
COMMON /MOCS/ SIGMA(3,6,25)
COMMON /FAIL/ XXX(42), TAI(8,25)
DIMENSION SKPP(12,12,20), SKPP2(12,12,20)
DIMENSION DS(12,12), CN(12,12), SKK(12,12,20), FOLD(12,21)

C IF( LOOP .GT. 1 ) GO TO 999
LOOP = 0
IF( INPUT .GT. 1 ) GO TO 999
RPM = RPM1

FOR CHECK OUT RUNS IN W137 THIS CALL TO INPUT IS MADE, REMOVE
AFTER SUITABLE INPUT IS AVAILABLE FROM W137

CALL INPUT(NR,NC)

DEFINE THE LOCAL NODAL COORDINATE SYSTEMS

CALL CORD2R(X,Y,Z,NR,NC)

CALCULATE THE LAMINA STRESS-STRAIN RELATION ,QIJ

CALL LAMINA

DEFINE THE REORDER MATRICES P AND P1

CALL PP1(1)

SET UP THE BOUNDARY CONDITIONS FOR A FIXED ROOT CONDITION

DO 50 I = 1,6
DO 50 J = 1,6
C(I,J) = 0.0

FIND EACH SECTION STIFFNESS AND CORRESPONDING HOLZER STIFFNESS

ALSO DETERMINE THE NODAL LUMPED MASS MATRICES
CALL HOLZER(1,NR,NC,RPM1)
CALL HOLZER(2,NR,NC,RPM1)

PERFORM THE STATIC ANALYSIS
WRITE(6,100) RPM1
WRITE(6,150) RPM1
FORM(/,5X, "STATIC ANALYSIS, RPM =',E12.5, ,/")
WRITE(6,100) RPM1
WRITE(6,150) RPM1
CALL STATIC(NR,NC,1)

SET THE INITIAL VALUE OF SKKOLD - THE ORIGINAL STIFFNESSES
NRML = NR - 1
DO 201 I = 1,NRML
DO 201 J = 1,12
DO 201 K = 1,12
SKKOLD(J,K,I) = SKK(J,K,II

LOCAL BIRD INGESTION PARAMETER CALCULATION
CALL FOD( NR,NC,RPH1,ALPHA I
IF(ISERCH .GT. II GO TO 998
UPDATE THE STIFFNESS MATRICES FOR DIFFERENTIAL STIFFNESS
AND CENTRIFUGAL RESTORATION - ALSO UPDATE THE LOAD VECTOR, F
LOOP = LOOP + 1
RATIO = 1.
WRITE(6,500) ISERCH,LOOP,RATIO
FOR RPM1 - HOLZER STIFFNESS MATRIX IS SKPP1(12,12,20)
FOR RPM2 - HOLZER STIFFNESS MATRIX IS SKPP2(12,12,20)

CALL HOLZER(2,NR,NC,RPH1)
FOR RPM1 - HOLZER STIFFNESS MATRIX IS SKPP1(12,12,20)
FOR RPM2 - HOLZER STIFFNESS MATRIX IS SKPP2(12,12,20)

DO 400 I = 1,NRML
DO 400 J = 1,12
DO 400 K = 1,12
IF(LOOP .EQ. 1) SKPP1(J,K,I) = SKPP1(J,K,I)
IF(LOOP .EQ. 2) SKPP2(J,K,I) = SKPP(J,K,I)
CONTINUE
IF(LOOP .EQ. 1) GO TO 888
RESET THE HOLZER MATRIX FOR THE STATIC ANALYSIS

WRITE(5,501)

FORMAT(/,5X,** RESET SKPP MATRIX **,/)             

DO 410 I = 1,1NRU1
     DO 410 J = 1,12
     DO 410 K = 1,12
410  SKPP(I,K,I) = SKPP1(I,J,K)

CALL STATICANL(NR,NC)

CALCULATE TSAI-WU FOR LE AND TE ROOT STRESS - SMAXL,SMAXT

CALL TSAIWU(SIGMA,HC)

DO 330 I = 1,8
     SMAXL(I) = TSAI(I,NC)
330  SMAXT(I) = TSAI(I,NC)

IF(IHOLE .NE. 0) CALL ZSTRES(IHOLE,NC,SMAX2)

IHOLE1 = IHOLE + 1

IF(IHOLE .NE. 0) CALL ZSTRES(IHOLE1,NC,SMAX3)

IF(IHOLE .EQ. 0) GO TO 501

DO 300 I = 1,8
     IF(SMAX3(I) .GT. SMAX2(I)) SMAX2(I) = SMAX3(I)
300 CONTINUE

501 CONTINUE

FORM MASS * STIFFNESS PRODUCTS TO BE SENT TO W137 SOLVER

-MUST CHANGE SIGN ON 21 AND 22 QUADRANTS OF SKPP FIRST

SKPP = HOLZER STIFFNESS MATRICES FOR EACH SECTION
C FORM MASS MATRIX AND STORE IT IN SK ARRAY

C

ISH 0106 DO 305 I = 1,12
ISH 0107 DO 305 J = 1,12
ISH 0108 305 SK(I,J) = 0.
ISH 0109 DO 306 I = 1,6
ISH 0110 SK(I,J) = 1.
ISH 0111 306 SK(I+6,J+6) = -1.
C
C HOW INSERT SMM INTO THE 21 QUADRANT OF THE MASS MATRIX SK
C ACTUALLY THE PRODUCT FREQ**2 * SMM IS INSERTED
C
ISH 0112 DO 310 I = 1,6
ISH 0113 DO 310 J = 1,6
ISH 0114 310 SK(I+6,J) = SK(I+6,J) + FREQ2 * SMM(I,J,I SEC+1)
C
ISH 0115 CALL MATHPY(SK,SKP,DS1,12,12,12)
C CALL MATPRN(DS1,12,12, 'DS1')
C REORDER THE MASS STIFFNESS PRODUCT MATRIX TO AGREE WITH
C THE W137 ORDER
C
ISH 0116 CALL PP1(2)
ISH 0117 CALL MATHPY(DS1,PT,CH,12,12,12)
ISH 0116 CALL MATHPY(PT,CH,DS1,12,12,12)
C CALL MATPRN(DS1,12,12,'DSCP')
C FILL IN THE SK MATRIX
C
ISH 0119 DO 320 I = 1,12
ISH 0120 DO 320 J = 1,12
ISH 0121 320 STAMAT(I,J,I SEC) = DS(I,J)
C
ISH 0122 300 CONTINUE
C
ISH 0123 RETURN
C
ISH 0124 END

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

2984k BYTES OF CORE NOT USED
SUBROUTINE STATICINR,NC.ISKIPI

THIS ROUTINE PERFORMS THE STATIC SOLUTION WITH THE HOLZER METHOD

IMPLICIT REAL*8(A-H,O-Z)

COMMON /STIFF/ SK(12,12),SKK(12,12,21),SHI(6,6,21)

COMMON /STIFF/ CHGG(12,12,20),DSKK(12,12,20)

COMMON /HSTIFF/ SKP(12,12),SKPP(12,12,21)

COMMON /STATIC/ SH(12,12,20),F(12,21)

COMMON /EC/ CI(6,6),RPI

COMMON /STRS/ RBE(12,12,25,21),FOUT(12,21),SL(25,21),SW(25,21)

DIMENSION SHII(12,6,6),SHII(6,6,6),SHI2(6,6)

DIMENSION SHII(12,12),FI(12,6,6),RI(6,6),RHI(6)

DIMENSION FF1(12,21),FI(6,1),FFI(6,1),FFI(6,1)

DIMENSION SDSPL(12,1),SDSPL(12,1),SDSPL(12,1),BDSPL(12,1)

DIMENSION RB(12,12)

FORM THE HOLZER STIFFNESS MATRIX STRING MULTIPLICATION

CALL SHULT(NR)

CALL LOAD2 (NR,FF)

PARTITION THE SH(12,12,NR-1) MATRIX FOR SOLUTION

THE SH IS A DUMMY MATRIX

DO 50 I = 1,12

DO 50 J = 1,12

SHII(I,J) = SH(I,J,NR-1)

CALL MATPRN(SHII,12,12,'SHII ')

CALL PARTN(SHII,12,12,SHII,SHII,SHII,SHII,6,6)

SOLVE FOR THE FORCE VECTOR AT NODE 1

F1 = (SHII(+1) + SHII) * FF1(-12)

WITH -1- MEANING THE INVERSE

DO 100 I = 1,6

FFI(I,1) = -FFI(I+6,1)

CALL MATPRN(FFI,6,1,'FFI ')

CALL MATPRN(FFI,6,1,6,6,1)

CALL MATADD(D1,SHII,SHII,SHII,SHII,6,6,1,1)

CALL MATPRN(SHII,6,6,'SHII ')

CALL MATADD(D1,SHII,SHII,SHII,SHII,6,6,1,1)

CALL MATPRN(FFI,6,1,'FFI ')

SOLVE FOR THE DISPLACEMENT AT NODE,NR
DO 110 \textbf{I} = 1,6
\textbf{I}10 \textbf{F}1(\textbf{I},1) = \textbf{F}1(1,1)
\textbf{C}

\textbf{ISN 0029}
CALL \text{MAT}HY(\text{SH}11,C,\text{SH}21,6,6,6)
\textbf{ISN 0030}
CALL \text{MAT}ADD(\text{SH}21,\text{SH}12,\text{SH}11,6,6,1,1)
\textbf{ISN 0031}
CALL \text{MAT}HY(\text{SH}11,F1,\text{FF}2,6,6,1,1)
\textbf{ISN 0032}
CALL \text{MAT}ADD(\text{FF}2,\text{FF}1,\text{U},6,6,1,1,1)
\text{CALL} \text{MAT}PRN(\text{U},6,1,1,1,1,1)
\textbf{C}

\text{THE VECTOR OF DISPL. AND FORCES AT EACH NODE WILL BE STORED}
\text{IN} \text{FOUT}(12,21) - \text{DISPL. IN} 1-6 \text{AND FORCES IN} 7-12.
\text{FIRST SOLVE FOR DEFLECTIONS AND FORCES AT NODE 2}
\text{THIS SECTION ALSO GENERATES THE DIFFERENTIAL STIFFNESS MATRIX}
\text{AND THE CENTRIFUGAL RESTORING STIFFNESS MATRIX}

\textbf{DO 120} \textbf{I} = 1,6
\textbf{F}OUT(I,NR) = \text{U}(I,1)
\text{F}OUT(I+6,NR) = \text{F}(I+6,NR)
\text{F}OUT(I,1) = 0.
\text{F}OUT(I+6,1) = \text{F}(I,1)
\textbf{FF}(I,1) = 0.
\textbf{FF}(I+6,1) = \text{F}(I+6,1)
\textbf{120} \text{FF}(I+6,1) = \text{F}(I+6,1)
\textbf{C}

\text{GET THE HOLZER STIFFNESS FOR SECTION 1}
\text{DO 130} \textbf{I} = 1,6
\text{DO 130} \textbf{J} = 1,6
\text{SP}1(I,J) = \text{SKPP}(I,J,1)
\text{SKP}(I,J+6) = \text{SKPP}(I,J+6,1)
\text{SKP}(I+6,J) = -\text{SKPP}(I+6,J,1)
\text{SKP}(I+6,J+6) = -\text{SKPP}(I+6,J+6,1)
\textbf{130} \text{SKP}(I+6,J+6) = -\text{SKPP}(I+6,J+6,1)
\textbf{C}

\text{CALL} \text{MAT}PY(\text{SKP},\text{FF},\text{FOUT}1,12,12,1)
\text{CALL} \text{MAT}PY(\text{FF},12,1,1,1,1,1)
\text{CALL} \text{MAT}PY(\text{FOUT}1,12,1,1,1,1,1)
\text{C}

\text{THE DIFFERENTIAL STIFFNESS FOR SECTION 1}
\text{IF(ISKIP .LE. 1) CALL KDGG1,1,FF,FOUT1,DSKK)
\text{DO 135} \textbf{I} = 1,12
\textbf{F}OUT(I,2) = \text{FOUT}(I,1)
\textbf{C}

\text{BEGIN LOOP TO SOLVE FOR THE REMAINING NODES 3 - (NR-1)}
\text{NRH1} = \text{NR} - 1
\text{IF(NR}1 \text{.EQ. 1) GO TO 201}
\text{DO 200} \textbf{I} = 3,\text{NRH1}
\textbf{C}

\text{FILL IN THE DUMMY DISPL. AND FORCE VECTOR AT NODE ( I-1 )}
\text{THE EQUILIBRIUM EQ WILL BE USED}
\text{DO 210} \textbf{J} = 1,6
\textbf{FF}(J,1) = \text{FOUT}(J,1-1)
\textbf{FF}(J+6,1) = -\text{FOUT}(J+6,1-1) + \text{F}(J+6,1-1)
\textbf{210} \text{FF}(J+6,1) = -\text{FOUT}(J+6,1-1) + \text{F}(J+6,1-1)
\textbf{C}

\text{GET THE HOLZER STIFFNESS MATRIX FOR SECTION ( I-1 )}
**VERSION 1.3.0 (01 MAY 80) STATIC SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 02.14/10.50.26 PAGE 3**

C

ISH 0058 DO 220 J = 1, 6 02599
ISH 0059 DO 220 K = 1, 6 02600
ISH 0060 SKP(J,K) = SKPP(J,K,1-1) 02601
ISH 0061 SKP(J,K+6) = SKPP(J,K+6,1-1) 02602
ISH 0062 SKP(J+6,K) = -SKPP(J+6,K,1-1) 02603
ISH 0063 DO 220 SKP(J+6,K+6) = -SKPP(J+6,K+6,1-1) 02604
ISH 0064 CALL MATPY(SKP,FF,FOUT1,12,12,1) 02605
ISH 0065 CALL MATRN(FF,12,1,'FF') 02606
ISH 0066 CALL MATRN(FOUT1,12,1,'FOUT1') 02607
ISH 0067 C THE DIFFERENTIAL STIFFNESS FOR SECTION I-1 02608
ISH 0068 DO 225 J = 1, 12 02609
ISH 0069 FOUT(J,I) = FOUT(I,J) 02610
ISH 0070 CONTINUE 02611
ISH 0071 THE DIFFERENTIAL STIFFNESS FOR SECTION NR-1 02612
ISH 0072 DO 200 NR = 1, 1 02613
ISH 0073 FOUT(I,I) = FOUT(I,NR) 02614
ISH 0074 FOUT(I+6,I) = FOUT(I+6,NR) 02615
ISH 0075 FF(I,I) = FOUT(I,ISEC) 02616
ISH 0076 IF(SKIP .LE. I) CALL KDGG(ISEC,FF,FOUT1,DSKK) 02617
ISH 0077 CONTINUE 02618
ISH 0078 THE LOCAL BEAM DEFLECTIONS AND FORCES FOR EACH SECTION 02619
ISH 0079 DO 300 I = 1, NRH1 02620
ISH 0080 IP = (I-1) * NC + 1 02621
ISH 0081 WRITE(6,305) I 02622
ISH 0082 FORMAT(/,5X,'DEFLECTIONS AND FORCES FOR SECTION',I5,/) 02623
ISH 0083 DEFINE SECTION ARRAYS FOR DISPL. AND FORCES 02624
ISH 0084 DO 301 J = 1, 6 02625
ISH 0085 SDISPL(J,I) = FOUT(J,I) 02626
ISH 0086 SDISPL(J+6,I) = FOUT(J+6,I) 02627
ISH 0087 SFORCE(J,I) = FOUT(J+6,I) 02628
ISH 0088 SFORCE(J+6,I) = FOUT(J+6,I) 02629
ISH 0089 C DEFLECTIONS AND FORCES FOR EACH BEAM 02630
ISH 0090 DO 400 J = 1, NC 02631
ISH 0091 C 02632
ISH 0092 DO 401 K = 1, 12 02633
ISH 0093 DO 401 L = 1, 12 02634
**ISN 0094**

```fortran
        401 RB(K,L) = RBE(K,L,J,I)
```

**ISN 0095**

```fortran
        CALL MATOPY(RB,SDISPL,EDISPL,12,12,1)
```

**ISN 0096**

```fortran
        CALL MATOPY(RB,SFORCE,BFORCE,12,12,1)
```

**ISN 0097**

```fortran
        IPNC = IP + NC
```

**ISN 0098**

```fortran
        WRITE(6,404) J,IP,IPNC
```

**ISN 0099**

```fortran
        FORMAT(5X,'BEAM',15,'DEFLECTIONS, END 1=',IS,' END 2=',IS)
```

**ISN 0100**

```fortran
        WRITE(6,402) (EDISPL(K,1),K=1,6)
```

**ISN 0101**

```fortran
        FORMAT(2X,'END 1',6E12.5)
```

**ISN 0102**

```fortran
        WRITE(6,403) (BDISPL(K,1),K=7,12)
```

**ISN 0103**

```fortran
        FORMAT(2X,'END 2',6E12.5)
```

**ISN 0104**

```fortran
        WRITE(6,405)
```

**ISN 0105**

```fortran
        FORMAT(14X,'FORCES')
```

**ISN 0106**

```fortran
        WRITE(6,406) (EFORCE(K,1),K=1,6)
```

**ISN 0107**

```fortran
        WRITE(6,407) (EFORCE(K,1),K=7,12)
```

**ISN 0108**

```fortran
        IP = IP + 1
```

**ISN 0109**

```fortran
        CONTINUE
```

**ISN 0110**

```fortran
        WRITE(6,240)
```

**ISN 0111**

```fortran
        FORMAT(5X,'THE STATIC SOLUTION',/5X,'DISPL. AND ROTATIONS',/15X,'U',10X,'V',10X,'W',10X,'RX',10X,'RY',10X,'RZ')
```

**ISN 0112**

```fortran
        DO 250 I = 1,HR
```

**ISN 0113**

```fortran
        WRITE(6,260) (FOUT(I,J),J=1,6)
```

**ISN 0114**

```fortran
        WRITE(6,261) (FOUT(I,J),J=1,6)
```

**ISN 0115**

```fortran
        WRITE(6,262) (FOUT(I,J),J=1,6)
```

**ISN 0116**

```fortran
        FORMAT(5X,'FORCES AND MOMENTS',/5X,'FX',10X,'FY',10X,'FZ',10X,'MX',10X,'MY',10X,'MZ')
```

**ISN 0117**

```fortran
        DO 280 I = 1,HR
```

**ISN 0118**

```fortran
        WRITE(6,266) (FOUT(I,J),J=7,12)
```

**ISN 0119**

```fortran
        RETURN
```

**ISN 0120**

```fortran
        END
```

---

**OPTIONS IN EFFECT**

```fortran
*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTO32(DBL4)*
```

**OPTIONS IN EFFECT**

```fortran
*SOURCE EBCDIC NOLIST NODECK OBJECT NMAP NFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)*
```

**STATISTICS**

```fortran
SOURCE STATEMENTS = 119, PROGRAM SIZE = 7468, SUBPROGRAM NAME =STATIC
```

**STATISTICS**

```fortran
NO DIAGNOSTICS GENERATED
```

**END OF COMPI LATION ******

---

2988K BYTES OF CORE NOT USED
C SUBROUTINE ZSTRES (ISEC, NC, SMAX)
C THIS ROUTINE PERFORMS THE STRESS ANALYSIS FOR EACH COMPOSITE
C BEAM IN THE SECTION

C IMPLICIT REAL*8(A-H,O-Z)
COMMON /STRS/ RBE(12,12,25,21),FOUT(12,21),SL(25,21),SW(25,21)
COMMON /PLY/ PLY(21,25,7),THETA(7),RHO(7)
COMMON /ZCQEF/ E11(7),E22(7),E33(7),G12(7),G23(7),G13(7),
1 V12(7),V13(7),V23(7)
COMMON /RECORD/ P(12,12),PT(12,12)
COMMON /FAIL/ VV(42),TSAI(8,25)
COMMON /MDATA/ SIGMA(3,6),SIGMA(3,25)
DIMENSION BDISPL(12,11),SDISPL(12,11),STRN(6,1),STRN(6,1)
DIMENSION R(12,12),TRANS(6,6),STR(3,1),STR(3,1)
DIMENSION DI3(3,1),DI4(3,1),Q3(3,3),SIGMA(3,1),SIGMA(3,1)
DIMENSION TT(3,3),SHAX(6)
DIMENSION TTRC(3,3),TTRC(3,3),QB(3,3),QB(3,3)

TRANS(6,5) = -1.

DO 100 J = 1,NC
C TRANSFER THE SECTION NODE DISPLACEMENTS IN THE FOUT ARRAY
C TO BEAM I
C
DO 110 J = 1,12
C CALL MATPRNC(RBE,J,K,ISEC)
C CALL MATPRNC(SDISPL,J,J,1, 'SDIS')
C CALL MATPRNC(BDISPL,J,J,1, 'BDIS')
C
DO 120 J = 1,6
C CALL MATPRN(SDISPL,J+6,1) = FOUT(J,ISEC)
C CALL MATPRN(BDISPL,J,1) = FOUT(J,ISEC+1)
C
DO 110 K = 1,12
C CALL MATMPY(R,J,K,1,1, 'R')
C CALL MATMPYSDISPL,BDISPL,12,12,1)
C CALL MATMPY(BDISPL,BDISPL,12,12,1)
C
C SHAPE FUNCTION COEFFICIENT EVALUATION

C BL = SL(I,ISEC)
C UI = BDISPL(1,1)
C V1 = BDISPL(2,1)
C W1 = BDISPL(3,1)
C RX1 = BDISPL(4,1)
C RY1 = BDISPL(5,1)
C RZ1 = BDISPL(6,1)
C U2 = BDISPL(7,1)
C V2 = BDISPL(8,1)
C W2 = BDISPL(9,1)
C RX2 = BDISPL(10,1)
C RY2 = BDISPL(11,1)
C RZ2 = BDISPL(12,1)
C
CALCULATE THE DEFLECTIONS AND SLOPES AT 10% OF BEAM LENGTH

X = 0.25 * SL(I, ISEC)

BEGIN LOOP FOR THE SEVEN LAYERS IN EACH BEAM

T = 0.
DO 51 K = 1,7
T = T + PLY(ISEC, I, K)
Y = T / 2.
Z = SWI, ISEC) / 2.

DETERMINE THE RADIAL DEFLECTIONS AT THE BEAM CORNERS

U1, U2, U3, U4

WRITE (6, 500) I, U1, U2, U3, U4

FORMAT SX, '** BEAM, CORNER DEFLECS. ',I5,4E12.5)

CORRESPONDING FILAMENT STRAINS , SX1, SX2, SX3, SX4

WRITE (6, 501) SL(I, ISEC), SX1, SX2, SX3, SX4

FORMAT SX, 'LENGTH,CORNER STRAINS',5E12.5)

SLOPE OF STRAIN-THICKNESS LINE - SIDE Z = -SW/2 - SLOPE1
Z = +SW/2 - SLOPE2

LOOP TO SOLVE FOR LAYER STRESSES

DO 200 LAYER = 1,7
TEFF = T/2. + Y
PROCEDURE ISN 0066
SIDE1 = SLOPE1 * TEFF + SX1
SIDE2 = SLOPE2 * TEFF + SX2
WRITE(6,503) I,LAYER,T,Y,TEFF,SIDE1,SIDE2
FORMAT(SX,'BEAM,LAYER,T,Y,TEFF,S1,S2',/,S5,S2,S5,S2,S5,S5,S2,S5)
C
ISH 0068
YOLD = Y
ISH 0069
IF(LAYER .GT. 7) GO TO 201
ISH 0071
Y = Y + PLY(ISEC,I,LAYER)
ISH 0072
201 Y = LAYER
ISH 0073
IF(LAYER .GE. 5) II = II - 1
ISH 0074
ISH 0075
C DETERMINE THE LAMINA GEOMETRIC STRAIN COMPONENTS ,EY AND EXY
ISH 0076
C TRANSFORMATION MATRIX IS TT
ISH 0077
C TTI = TT INVERSE
ISH 0078
C TTR = TT TRANSPOSE
ISH 0079
C
ISH 0080
CC = COS(TETA(I) )
ISH 0081
SS = SIN(TETA(I) )
ISH 0082
C2 = CC * CC
ISH 0083
S2 = SS * SS
ISH 0084
CS = CC * SS
ISH 0085
TT(1,1) = CC
ISH 0086
TT(1,2) = SS
ISH 0087
TT(1,3) = 2. * CS
ISH 0088
TT(2,1) = S2
ISH 0089
TT(2,2) = C2
ISH 0090
TT(2,3) = -2. * CS
ISH 0091
TT(3,1) = -CS
ISH 0092
TT(3,2) = C2
ISH 0093
TT(3,3) = C2 - S2
ISH 0094
DO 150 KK = 1,3
ISH 0095
DO 150 LL = 1,3
ISH 0096
TTI(KK,LL) = TT(KK,LL)
ISH 0097
CALL MATPRNT(3,3,TTI)
ISH 0098
CALL MATPRNT(3,3,TT)
ISH 0099
CALL MATPRNT(3,3,TTR)
ISH 0100
IF(EII(I) .EQ. 0 .OR. PLY(ISEC,I,I) .EQ. 0) GO TO 300
ISH 0101
V21 = V12(I) * E22(I) / E11(I)
ISH 0102
FACTOR = 1 - V12(I) * V21
ISH 0103
Q(1,1) = E11(I) / FACTOR
ISH 0104
Q(1,2) = V21 * Q(1,1)
C ROTATED LAMINA STIFFNESS MATRIX, QB(I,J)
C
C CALL MATMPY(Q,TTR,QB1,3,3,3)
C CALL MATMPY(TTI,QB1,QB,3,3,3)
C
C QF1 = QB(1,2) - QB(2,3) * QB(1,3) / QB(3,3)
C QF2 = QB(2,2) - QB(2,3) * QB(2,3) / QB(3,3)
C
C STR1(1,1) = SIDE1
C STR1(2,1) = -QF1 / QF2 * SIDE1
C STR1(3,1) = (QB(2,3)/QB(3,3))*QF1/QF2-QB(1,3)/QB(3,3)) * SIDE1
C
C STR2(1,1) = SIDE2
C STR2(2,1) = -QF1 / QF2 * SIDE2
C STR2(3,1) = (QB(2,3)/QB(3,3))*QF1/QF2-QB(1,3)/QB(3,3)) * SIDE2
C
C HOLLOW LAYER - ZERO Q(I,J)
C
C GO TO 301
C 300 CONTINUE
C
C DO 310 KK = 1,3
C
C CALL MATMPY(Q,TTR,STR1,3,3,1)
C CALL MATMPY(TTR,STR2,03,3,3,1)
C
C CALL MATMPY(Q,TTR,STR1,3,3,1)
C CALL MATMPY(TTR,STR2,03,3,3,1)
C
C AVERAGE THE SIDE STRESSES - STORE IN SIGI1A
C 02924*35
ISH 0136 DO 210 J = 1,3
C
ISH 0137 210 SIGMA(J,LAYER,I) = (SIGMA(J,1) + SIGMA2(J,1)) / 2.
C
C WRITE(6,212) (SIGMA(JJ,LAYER,I),JJ=1,3)
C212 FORMAT(5X,3E12.5)
C
ISH 0138 200 CONTINUE
ISH 0139 100 CONTINUE
C
C CALCULATE THE TSAI-WU FAILURE CRITERION FOR EACH LAYER OF EACH
C BEAM FOR THE SECTION
C
ISH 0140 CALL TSAIWU (SIGMA,NC)
C
C PICK OUT THE MAX TSAI-WU VALUE PER LAYER FOR ALL BEAMS
C
ISH 0141 DO 220 I = 1,8
ISH 0142 SMAX(I) = 0.
ISH 0143 DO 220 J = 1,NC
ISH 0144 220 IF(SMAX(I) .LT. TSAI(J,I)) SMAX(I) = TSAI(I,J)
C
ISH 0146 RETURN
ISH 0147 END

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

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

2984K BYTES OF CORE NOT USED
SUBROUTINE THICKCI,NR,NC,X,Y,Z,THK,ISKIP)

C DETERMINE THE LAYER THICKNESS, TH, ASSUMING THICKNESS SYMMETRY ABOUT THE CENTER AXIS

IMPLICIT REAL*8(A-H,O-Z)

COMMON IPLY1,21,25,7),THETAC7),RHOC7)
COMMON ILAYER1, THC 7,25),BMASS(25),BSPAN(25),BWIDTH(25)
COMMON WIEG1H(7)
COMMON IANAL331 DUM6(6),NCO,NCK,DUM14(4)
COMMON /BLK81 S1BC 21),ITTLEC 18),DUM11C 277)

DIMENSION X(1),Y(1),Z(1),THK(7),TP(7),T(1000),DIST(25)
DIMENSION PLY(7)

DO 100 I = 1,NC
   IP = IP + 1
   THK(1) = (THK(IP) + THK(IP+1))/2.

C AVERAGE THE X VALUES FOR BOTH ENDS TO FIND LAYER THICKNESS

TOTAL = THK(IP)
DO 101 J = 1,7
   TH(J,I) = PPLY(IP,J)

C DETERMINE THE BEAM LENGTH

XX = X(IP) - X(IP+1)
YY = Y(IP) - Y(IP+1)
ZZ = Z(IP) - Z(IP+1)
BSPAN(I) =DSQRT(XX**2 + YY**2 + ZZ**2)

C AVERAGED WIDTH

XX = X(IP) - X(IP+1)
YY = Y(IP) - Y(IP+1)
ZZ = Z(IP) - Z(IP+1)
DIST(I) =DSQRT(XX**2 + YY**2 + ZZ**2)

C BEAM WIDTHS

NCM1 = NC - 1
IP(NCM1,EQ.0) GO TO 201
BWIDTH(I) = DIST(I)

100 CONTINUE
SUBROUTINE TBLK(ISKIP, NC, I, TH)
  USES, AREA, BMASS, BSPAN, BWIDTH, DIST, NCD

  ! Risk rank K
  ! Initial area, $A_0 = B \cdot \sum_{i} (B - A_0)$
  ! NCD = number of cards
  ! $\rho = \sqrt{\frac{A}{2}}$

  I = 1, NC
  BMASS(I) = 0.
  AREA = BWIDTH(I) * BSPAN(I)

  J = 1, NCD
  BMASS(I) = BMASS(I) + AREA * TH(J,I) * $\rho(J)$

  RETURN
END
SUBROUTINE TSTAIWU (SIGMA, NC)

IMPLICIT REAL*8(A-H,O-Z)

THE TSAI-WU FAILURE CRITERION IS EVALUATED

COMMON /FAIL/ XI(T),XIC(T),X2(T),X2C(T),S6P(T),S6M(T),TSAI(0,25)

DIMENSION SIGMA(3,6,25),FI(8),F2(8),F6(8),FI1(8),F22(8),F66(8)

C DETERMINE THE COEFFICIENTS

L = 0

DO 100 I = 1,8
L = L + 1
IF(l.EQ. 5) L = L - 1

F1(I) = 1./XI(T(L)) - 1./XIC(L)
F2(I) = 1./X2(T(L)) - 1./X2C(L)
F6(I) = 1./S6P(L) - 1./S6M(L)

100 F12(I) = -DSQRT(F1(I)*F22(I))

CALL MATPRH(Fl, 6, I, 'F1 ')
CALL MATPRN(F2, 8, I, 'F2 ')
CALL MATPRN(Fl1, 8, I, 'F11 ')
CALL MATPRN(F22, 8, I, 'F22 ')
CALL MATPRN(F66, 8, I, 'F66 ')
CALL MATPRN(F12, 8, I, 'F12 ')

BEGIN LOOP FOR EACH LAYER OF EACH BEAM

DO 110 J = 1, NC
DO 110 I = 1, 8

TSAI(I,J) = F1(I) * SIGMA(1,I,J) + F2(I) * SIGMA(2,I,J) +
           F6(I) * SIGMA(3,I,J) + FI1(I) * SIGMA(1,1,J) +
           2*F22(I) * SIGMA(2,1,J) +
           3*F66(I) * SIGMA(3,1,J) +
           F12(I) * SIGMA(1,1,J) * SIGMA(2,1,J)

END

RETURN

*OPTIONS IN EFFECT* NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOOBl(DBL4)
*OPTIONS IN EFFECT* SOURCE EBCDIC NONLIST NODECK OBJECT NOHAP NOFORMAT GOSTMT NOXREF NOALC NOHASF TERM IBM FLAG(1)
*STATISTICS* SOURCE STATEMENTS = 22, PROGRAM SIZE = 1112, SUBPROGRAM NAME = TSTAIWU
*STATISTICS* NO DIAGNOSTICS GENERATED

****** END OF COMPILATION ******
SUBROUTINE WARP(ISEC, NR, NC)

THIS ROUTINE CALCULATES THE WARPING TERMS REQUIRED FOR THE WARPING FUNCTION OF A THIN SECTION - THE VALUES WILL BE USED IN SUBROUTINE RBE - (SEE THE THEORY OF ELASTIC STABILITY, TIMOSHENKO AND GERE)

IMPLICIT REAL*8IA-H, O-Z)

COMMON /SCXSEC(21), YSEC(21), ZSEC(21), ALPHA(21), SECIP(21), SECA(21)

COMMON /INPUT/ X(1000), Y(1000), Z(1000), T(1000)

COMMON /LAYER/ TH(7,25), BHASS(25), BSPAN(25), BIWTH(25)

COMMON /WARPP/ HS(26,21), RII(25,21), RTT(25,21)

COMMON /COORD1/ CLOCAL(3,3,1000)

DIMENSION RT(26), THETA(26), SS(25), HSS(25), ROUT(26)

DIMENSION SIG(25)

C ARC LENGTH VARIABLE , S , AND SS.
C SS= CUMULATIVE ARC LENGTH FOR ELEMENT INTERVALS
C
SS(1) = BWIDTH(1)

DO 60 I = 2, NC

60 SS(I) = SS(I-1) + BWIDTH(I)

CALL MATPRM(SS,25,1,1, 'SS ')

C DETERMINE THE VALUE , HS , BEGINNING AT THE LEADING EDGE
C - MUST ALSO CALCULATE THE PERPENDICULAR DISTANCES FROM THE TANGENT, RT AND FROM THE NORMAL, RN TO THE SHEAR CENTER
C WS = EVALUATED AT NODE POINTS
C WSS = EVALUATED AT INTERVAL ENDS
C
NODE = (ISEC-1) * NC + 1

DO 100 I = 2, NC

BOTTOM LAYER OF SECTION ISEC

RX = XSEC(ISEC) - X(NODE)

RY = YSEC(ISEC) - Y(NODE)

RZ = ZSEC(ISEC) - Z(NODE)

RMAG = DSQRT( RX**2 + RY**2 + RZ**2 )

IF(RMAG.EQ.0.) GO TO 50

RX = RX / RMAG

RY = RY / RMAG

RZ = RZ / RMAG

C ANGLE BETWEEN VECTORS R AND LOCAL Z , THETA
C
ARG = RX * CLOCAL(3,1,NODE) + RY * CLOCAL(3,2,NODE) + RZ * CLOCAL(3,3,NODE)

THETA(I) = DARCOS( ARG )

C DETERMINE THE DIRECTION OF SHEEP, CW OR CCH, FOR THE SIGN ON WSS - NOTE: THIS CHECK USES THE COORDINATE POINTS NOT THE SIDES ON THE BEAM, SO THE CHECK IS APPROXIMATE
C
IF(I.EQ. NC) SIGN(I) = SIGN(I-1)

IF(I.EQ. NC) GO TO 400

C V1 CROSS V2 - THE RESULTANT X COMPONENT SETS SIGN OF, SIGN.
C XCOMP GT. 0. - SIGN = 1
C XCOMP LT. 0. - SIGN = -1

C

C XCOMP = V1Y * V2Z - V1Z * V2Y
C SIGN(I) = 1.
C IF(XCOMP .LT. 0.) SIGN(I) = -1.
C WRITE(6,111) I,NC,SIGN(I)
C111 FORMAT(5X,'I,NC,SIGN(I)'),2I5,E12.5)

C

C RN(I,ISEC) = -RMAG * DCOS( THETA(I) )
C RT(I) ) = RMAG * DSIN( THETA(I) )
C RTT(I,ISEC) = RT(I) * SIGN(I)
C ROUT(I) = RN(I,ISEC)

C

C IF( I .EQ. 1 ) GO TO 101
C WSS(I) = WSS(I-1) + SIGN(I) * RT(I) * (SS(I) - SS(I-1))
C GO TO 100
C
C101 CONTINUE

C

C WSS(I) = SIGN(I) * RT(I) * SS(I)
C WRITE(6,112) I,SIGN(I),WSS(I)
C112 FORMAT(5X,'I,SIGN(I),WSS(I)'),IS,2E12.5)

C

C100 NODE = NODE + 1
C WRITE(6,52) (WSS(III),II=1,NC)
C52 FORMAT(5X,E10.4)
C CALL MATPRN(ROUT,26,1,' RN ')
C CALL MATPRN(RT,26,1,' RT ')
C CALL MATPRN(THETA,26,1,'THET ')
C CALL MATPRN(SIGN,25,1,'SIGN ')
C CALCULATE THE AREA
C

C AREA = 0.
C NODE = (ISEC-1) * NC + 1
C DO 110 I = 1,NC
C AREA = AREA + T(NODE) * BWIDTH(I)
C110 NODE = NODE + 1
C WRITE(6,53) AREA
C53 FORMAT(5X,'AREA=',E12.5)
C AVERAGE WARING , WSBAR
C WSBAR = 0.
C NODE = (ISEC-1) * NC + 1
C DO 200 I = 2,NC
C NODE = NODE + 1
C SSS = SS(I) - SS(I-1)
C200 WSBAR = WSBAR + T(NODE) * (WSS(I) + WSS(I-1)) / 2. * SSS
C ADD THE FIRST TERM AND DIVIDE BY THE TOTAL AREA
C HERE
C NODE = (ISEC-1) * NC + 1
I.

\[ WS = WS\bar - \sigma; \]  

\[ WS = WS\bar - WS(1) / 2. \]  

\[ WS(1) = WSS(1) / 2. \]  

\[ WS(1, ISEC) = WS\bar - WSS(I) / 2. \]  

\[ WS(I, ISEC) = WS\bar - (WSS(I) + WSS(I-1)) / 2. \]  

\[ WRITE(6, 52) (WS(I, ISEC), II=I, NC) \]  

\[ RETURN \]  

\[ END \]  

*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 NOXREF NOXREF NOXREF 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
SUBROUTINE HOLLOW(DLE,DTE,DROOT,DTIP,TTI,TBT,NC,NR)

THIS ROUTINE CALCULATES THE LAYER THICKNESSES FOR EACH BEAM. Also found are the beam length, BSPAN, and the width, BWIDTH.

DIMENSION 0151(25)

A = CHORDWISE DISTANCE OF HOLLOW SECTION PENETRATION OF A BEAM
B = SAME AS A BUT IN THE SPAN DISRECTION
RBOT = RADIAL DISTANCE TO BOTTOM OF CAVITY
RTIP = RADIAL DISTANCE TO TOP OF CAVITY

CHECK FOR THE CAVITY EXISTENCE

IHOLE = 0

CAVITY = DLE + DTE + DROOT + DTIP
IF (CAVITY .EQ. 0.) GO TO 150

GO TO 151

150 NR1 = NR - 1
DO 160 I = 1,NR1
NODE = (I-1) * NC + 1
C
DO 110 J = 1,NC

IHOLE = IHOLE + 1
RETURN

151 CONTINUE

RBOT = X(NC/2) + DROOT
RTIP = X(NC*NR-NC/2) - DTIP
WRITE(6,130) RBOT,RTIP

130 FORMAT(5X,'RBOT,RTIP',2E12.8)

DO 100 I = 1,NR1
NODE = (I-1) * NC + 1
DO 110 J = 1,NC

PLY(I,J,4) = 0.
PLY(I,J,5) = 0.
PLY(I,J,6) = 0.
PLY(I,J,7) = TAVE / 2.

RETURN

100 CONTINUE

110 CONTINUE
C BEAM LENGTH, BSPAN

C

ISN 0037
XX = X(NODE) - X(NODE+NC)

ISN 0038
YY = Y(NODE) - Y(NODE+1)

ISN 0039
ZZ = Z(NODE) - Z(NODE+1)

ISN 0040
BSPAN(J) = DSQRTE(XX**2 + YY**2 + ZZ**2)

C CALL MATRNX(BSPAN,ZS,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) = DSQRTE(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) + DSQRTE(XX**2 + YY**2 + ZZ**2)) / 2.

C

ISN 0049
110 NODE = NODE + 1

ISN 0050
NODE = NODE - 1

C

ISN 0051
NCH1 = NC - 1

ISN 0052
IF(NCH1 .EQ. 0) GO TO 201

ISN 0053
BWIDTH(1) = DIST(1)

ISN 0054
BWIDTH(NC) = DIST(NC-1)

ISN 0055
DO 111 JJ = 2,NCH1

ISN 0056
BWIDTH(JJ) = (DIST(JJ-1) + DIST(JJ)) / 2.

C CALL MATRNX(BWIDTH,ZS,1,'WIDTH')

C

ISN 0057
111 BWIDTH(JJ) = (DIST(JJ-1) + DIST(JJ)) / 2.

C

ISN 0058
CONTINUE

ISN 0059
DO 112 JJ = 1,N

ISN 0060
CHORD = CHORD + BWIDTH(JJ)

C WRITE(6,131) J,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 0064
IF(X(NODE+NC) .GT. RBT0 .AND. X(NODE+NC) .LE. RTIP) GO TO 20

ISN 0065
IF(X(NODE+NC) .GT. RTIP) GO TO 30

C

C THIS RADIAL SECTION IS SOLID

C

ISN 0066
10 B = 0.

ISN 0067
GO TO 50

C

C THIS RADIAL SECTION IS HOLLOW / SOLID

C

ISN 0068
20 B = BSPAN(J)

ISN 0069
IF(X(NODE) .LT. RBT0) B = X(NODE+NC) - RBOT

ISN 0070
IF(INHOLE .EQ. 0) INHOLE = I

ISN 0071
GO TO 50

C

C THIS RADIAL SECTION IS NEAR THE TIP, MAY INCLUDE SOME HOLLOW

C

ISN 0072
30 B = 0.

ISN 0073
IF(X(NODE) .LT. RTIP) B = RTIP - X(NODE)

C
50 CONTINUE
C NOW BEGIN CHORDWISE SEARCH FOR - A - DIMENSION
C
ISH 0001 IF (B .GT. 0.) GO TO 60
ISH 0003 GO TO 61
C THIS IS A HOLLOW BEAM
C
ISH 0004 60 CDIST = 0.
ISH 0005 DO 120 JJ = 1, J
ISH 0006 120 CDIST = CDIST + BWIDTH(JJ)
C CDIST = DISTANCE FROM LE TO BEAM (J)
C CTE = DISTANCE FROM LE TO CHORDWISE END OF CAVITY
C
ISH 0007 CTE = CHORD - DTE
C
ISH 0008 IF (CDIST .LE. DLE) GO TO 61
ISH 0009 IF (CDIST .GT. DLE .AND. CDIST .LE. CTE) GO TO 62
ISH 0010 CDIST1 = CDIST - BWIDTH(J)
ISH 0011 IF (CDIST1 .LT. CTE .AND. CDIST1 .LE. CTE) GO TO 63
ISH 0012 IF (CDIST1 .GT. CTE) GO TO 64
C
ISH 0007 CTE = CHORD - DTE
C
ISH 0008 IF (CDIST1 .LE. DLE) GO TO 61
ISH 0009 IF (CDIST1 .GT. DLE .AND. CDIST1 .LE. CTE) GO TO 62
ISH 0010 CDIST1 = CDIST1 - BWIDTH(J)
ISH 0011 IF (CDIST1 .LT. CTE .AND. CDIST1 .LE. CTE) GO TO 63
ISH 0012 IF (CDIST1 .GT. CTE) GO TO 64
C
ISH 0007 SOLID LE SECTION
C
ISH 0007 A = 0.
ISH 0009 GO TO 65
C HOLLOW / SOLID SECTION - LE
C
ISH 0009 A = CDIST - DLE
ISH 0100 IF (A .GT. BWIDTH(J)) A = BWIDTH(J)
ISH 0102 GO TO 65
C HOLLOW / SOLID SECTION - TE
C
ISH 0103 A = CTE - CDIST1
ISH 0104 GO TO 65
C SOLID TE SECTION
C
ISH 0105 A = 0.
ISH 0106 CONTINUE
C DEFINE THICKNESS OF EACH LAYER IN BEAM (J)
C
ISH 0107 IF (B .EQ. 0. .OR. A .EQ. 0.) GO TO 80
C
ISH 0109 TAVE = (TINODE + TINODE+HC) / 2.
ISH 0110 TCHECK = TAVE - 2. * TTI - 2. * TBT
ISH 0111 IF (TCHECK) 90, 90, 95
C NO HOLLOW SECTION DUE TO MINIMUM THICKNESS
C
ISH 0112 TCIK1 = TAVE - 2. * TTI
ISH 0113 IF (TCIK1) 91, 91, 92
ISH 0114 TTI1 = TAVE / 2.
ISH 0115 TBT1 = 0.
HOLLOW SECTION THICKNESS IS NONZERO

C

ISH 0122 95 THOL = A * B / BWIDTH(J) / BSPAN(J) * TCHECK
ISH 0123 TBTI = A * B / BWIDTH(J) / BSPAN(J) * TBT
ISH 0124 TTII = TAVE - THOL - 2. * TBTI
ISH 0125 TTII = TTII / 2.
ISH 0126 GO TO 200
C
C SOLID SECTION - ALL TITANIUM
C
ISH 0127 80 TTII = (T(NODE) + T(NODE+NC)) / 4.
ISH 0128 TBTI = 0.
ISH 0129 THOL = 0.
C
ISH 0130 200 CONTINUE
C
WRITE(6,132) I,J,A,B
C132 FORMAT(5X,'STATION,BEAM,A,B',2I5,2E12.5)
ISH 0131 PLY(I,J,1) = TTII
ISH 0132 PLY(I,J,2) = TBTI
ISH 0133 PLY(I,J,3) = 0.
ISH 0134 PLY(I,J,4) = THOL
ISH 0135 PLY(I,J,5) = 0.
ISH 0136 PLY(I,J,6) = TBTI
ISH 0137 PLY(I,J,7) = TTII
C
WRITE(6,300) I,J,(PLY(I,J,K),K=1,7)
C300 FORMAT(5X,'STATION,BEAM,PLY',2I5,/,5X,7E12.5)
C
ISH 0138 100 CONTINUE
C
ISH 0139 RETURN
ISH 0140 END

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOBL(DBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC HOLIST NODECK OBJECT NOHAP 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
C DATA SET U490DBMFIT AT LEVEL 003 AS OF 11/02/81
C DATA SET T926BMFFAt AT LEVEL 002 AS OF 05/25/79

ISON 0002 SUBROUTINE BMFIT2 ( X, Y, YP1, YPN, N, IP, A, B, C, D )
ISON 0003 DIMENSION X(N), Y(N), A(N), B(N), C(N), D(N)
ISON 0004 IF ( N - 2 ) 10, 30, 40
ISON 0005 WRITE (6,20) H
ISON 0006 FORMAT (/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,'/,'/42I3,
GO TO 160

D(I) = 1.0

A(I) = ( (Y(2)-Y(1))/(C(I) - YP1 ) * (6. / C(I) )

IF ( IP - 12 ) 170, 160, 190

D(I) = -YP1 - YP1

A(I) = 0.0

IF ( IP - 2 ) 150, 160, 190

GOTO 150 IF ( IP ) 300, 160, 170

C PERIODIC SPLINE

C LAMBDA(N) = H(1) / ( H(1) + H(N-1) )

C SD(N) = 6. *(((Y(2)-Y(N))/H(1) - (Y(N)-Y(N-1))/H(N-1))/(H(N-1)+H(1)))

C A(N) = 6. *(((Y(2)-Y(N))/C(1) - (Y(N)-Y(N-1))/C(N1) )/(C(N1)+C(1)))

B(2) = ( D(2) - 1. ) / 2

A(2) = A(2) / 2

D(2) = -D(2) / 2

C S(I) = S(I-1) + LAMBDAg(i-1) / ( 2.*W(I-1)*LAMBDAg(i-1))

C U(I) = (SD(I)-U(I-1)*LAMBDAg(i) / ( 2.*W(I-1)*LAMBDAg(i))

C Q(I) = -LAMBDAg(i) / ( 2.*W(I-1)*LAMBDAg(i))

C DO 310 I = 3, N1

B(I) = (B(I-1)*D(I-1)) / ( 2.+D(I-1)*1.-(D(I)-1))

A(I) = (A(I)-A(I-1)*D(I)) / ( 2.+D(I-1)*(1.-D(I)))

D(I) = -D(I) / ( 2.+D(I-1)*(1.-D(I)))

C DO 310 I = 3, N1

T(N-1) = Q(N-1) + S(N-1)

C V(N-1) = U(N-1)

C (T(I) = Q(I) * T(K+1) + S(K)

C V(K) = Q(K) * V(K+1) + U(K)

B(N1) = D(N1) + B(N1)

D(N1) = A(N1)

DO 320 I = 2, N2

K = N - I

B(K) = D(K) * B(K+1) + B(K)

D(K) = D(K) * D(K+1) + A(K)

C CONTINUE

C MIN) = ( SD(N) - V(N-1) + LAMBDAg(N) * ( V(N-1) - V(2) ) / ( 2. +

C T(N-1) = LAMBDAg(N) * ( T(N-1) - T(2) )

C H(1) = MIN)

C A(N) = (A(N)-D(N)+D(N)*D(N1)-D(2)) / ( 2.+B(N1)-D(N)*B(N1)-B(2))

C A(1) = A(N)

DO 330 I = 1, N2

K = N - I

C M(K) = T(K) * M(N) + V(K)

C A(K) = B(K) * A(N) + D(K)

C CONTINUE

C
+VERSION 1.3.0 (01 MAY 80) BMFIT2 SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 82.141/10.51.11 PAGE 3

ISH 0072 330 CONTINUE
ISH 0073 GO TO 225
C
C SET RIGHT END CONDITION
C TAU = B(N)
C SD(N) = A(N)
C
ISH 0074 160 B(N) = -YFN - YPN
ISH 0075 A(N) = 0.0
ISH 0076 GO TO 200
ISH 0077 170 B(N) = 1.0
ISH 0078 A(N) = ( YPN - (Y(N)-Y(N1))/C(N1) ) * ( 6. / C(N1) )
ISH 0079 GO TO 200
ISH 0080 180 B(N) = 0.0
ISH 0081 A(N) = YFN + YPN
ISH 0082 GO TO 200
ISH 0083 190 B(N) = -2. * ( ( C(N1) + C(N-2) ) / C(N-2) )
ISH 0084 A(N) = 0.0
ISH 0085 D(N) = ( C(N1) + C(N1) ) / C(N-2)
C
C G(I) = (LAMBD(I1)*SD(2)-2.*SD(1)) / (LAMBD(I1)*C(N1)-SD(2)-2.*C(N1))
C H(I) = (LAMBD(2)-LAMBD(I1)*SD(2)-2.*SD(1)) / (LAMBD(2)-LAMBD(I1)*C(N-2)-2.*C(N-2))
C G(2) = (SD(2)-G(1))*(1.-LAMBD(2)) / 2.
C H(2) = (LAMBD(2)-H(I))*(1.-LAMBD(2)) / 2.
C
ISH 0086 200 A(1) = (D(1)*A(2)-A(1)-A(1)) / (D(1)*C(N-2)-4.)
ISH 0087 B(1) = (D(1)*D(2)-C(N1)*C(N1)) / (D(1)*D(N1)-4.)
ISH 0088 A(2) = (A(2)-A(1)*(1.-D(2))) / 2.
ISH 0089 B(2) = (D(2)-B(1)*(1.-D(2))) / 2.
C
C G(I) = (SD(I1)-G(I-1)+(1.-LAMBD(I1)) / (2-W(I-1)+(1.-LAMBD(I1)))
C H(I) = LAMBD(I1) / (2-W(I-1)+(U-LAMBD(I1)))
C
ISH 0090 DO 210 I = 3, N1
ISH 0091 A(I) = (A(I1)-A(I-2)*(1.-D(I))) / (2.-B(I-1)*(1.-D(I)))
ISH 0092 B(I) = D(I) / (2.-B(I-1)*(1.-D(I)))
ISH 0093 210 CONTINUE
C
C M(N) = (SD(N)*S*(G(N-2)-W(N-2)+M(N-1)) / (2.(TAU-S)*H(N-2)*W(N-2)));(M(N-1))
C MK = G(K) - W(K) * M(K+1)
C
ISH 0094 A(N) = ( A(N) - D(N) ) * ( A(N2) - B(N2) * A(N1) ) - B(N) * A(N1) )
ISH 0095 1 / ( 2. - ( B(N) - D(N) * B(N2) ) * B(N1) )
ISH 0096 DO 220 I = 1, N2
ISH 0097 K = N - I
ISH 0098 A(K) = A(K1) - B(K) * A(K1)
ISH 0099 220 CONTINUE
C
C M(1) = G(1) - W(1) * M(1)
C
ISH 0099 A(1) = A(1) - B(1)*A(3)
C
C COMPUTE COEFFICIENTS
C
ISH 0100 225 DO 230 I = 1, N1
ISH 0101 C D(I) = ( M(I+1) - M(I) ) / ( 6. * H(I) )
ISH 0102 C B(I) = ( Y(I+1) - Y(I) ) / ( H(I) - H(I) ) * ( ( M(I) + 2 * M(I) ) ) / 6.00172
C C(I) = M(I) / 2 00173
C A(I) = Y(I) 00174
C
ISH 0101 D(I) = (A(I+1) - A(I)) / (6. * C(I)) 00175
ISH 0102 B(I) = (Y(I+1) - Y(I)) / C(I) - (A(I+1) + A(I) + A(I)) * (C(I) / 6.) 00176
ISH 0103 C(I) = A(I) / 2 00177
ISH 0104 A(I) = Y(I) 00178
ISH 0105 230 CONTINUE 00179
ISH 0106 240 RETURN 00180
ISH 0107 END 00181

*OPTION IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOCALL(DBL4)
*STATISTICS* SOURCE STATEMENTS = 106, PROGRAM SIZE = 2398, SUBPROGRAM NAME =BMFIT2
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILED ********

*STATISTICS* NO DIAGNOSTICS THIS STEP

2984K BYTES OF CORE NOT USED
F64-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED MAP, LIST, LET, SIZE=(256K, 36K)

VARIABLE OPTIONS USED - SIZE=(262144, 36864)

<table>
<thead>
<tr>
<th>NAME</th>
<th>ORIGIN</th>
<th>LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>STABLE</td>
<td>00</td>
<td>22ASC</td>
</tr>
<tr>
<td>BBIP</td>
<td>22A60</td>
<td>1864</td>
</tr>
<tr>
<td>BIRD</td>
<td>242C8</td>
<td>42C</td>
</tr>
<tr>
<td>ECOORD</td>
<td>246FB</td>
<td>2CE</td>
</tr>
<tr>
<td>BEAM</td>
<td>249CB</td>
<td>2706</td>
</tr>
<tr>
<td>CONDOR</td>
<td>271A0</td>
<td>1AC</td>
</tr>
<tr>
<td>ESTIFF</td>
<td>276C0</td>
<td>50A</td>
</tr>
<tr>
<td>EXHE1CK</td>
<td>280D0</td>
<td>12F2</td>
</tr>
<tr>
<td>FRP1</td>
<td>293C8</td>
<td>2C2</td>
</tr>
<tr>
<td>HOLZER</td>
<td>29690</td>
<td>AE8</td>
</tr>
<tr>
<td>INPUT</td>
<td>2A178</td>
<td>3A8</td>
</tr>
<tr>
<td>KDG5</td>
<td>2A520</td>
<td>9F0</td>
</tr>
<tr>
<td>L3HINA</td>
<td>2AF10</td>
<td>9FA</td>
</tr>
<tr>
<td>L3M110</td>
<td>2B910</td>
<td>39A</td>
</tr>
<tr>
<td>LOKG2</td>
<td>2BCB8</td>
<td>77E</td>
</tr>
<tr>
<td>NATADD</td>
<td>2C430</td>
<td>1B4</td>
</tr>
<tr>
<td>NATIHPY</td>
<td>2C5E8</td>
<td>232</td>
</tr>
<tr>
<td>MATPRN</td>
<td>2C620</td>
<td>21E</td>
</tr>
<tr>
<td>MINV</td>
<td>2C440</td>
<td>50E</td>
</tr>
<tr>
<td>MODHAS</td>
<td>2D020</td>
<td>40A</td>
</tr>
<tr>
<td>MODSTR</td>
<td>2D430</td>
<td>4C2</td>
</tr>
<tr>
<td>PARTN</td>
<td>2DEFB</td>
<td>37A</td>
</tr>
<tr>
<td>PPI</td>
<td>2DC78</td>
<td>216</td>
</tr>
<tr>
<td>RBE2</td>
<td>2DE90</td>
<td>50B</td>
</tr>
<tr>
<td>SECTN</td>
<td>2E468</td>
<td>4390</td>
</tr>
<tr>
<td>SHAPE</td>
<td>327F8</td>
<td>340</td>
</tr>
<tr>
<td>SHULT</td>
<td>32E30</td>
<td>100E</td>
</tr>
<tr>
<td>STABLE</td>
<td>33846</td>
<td>12E58</td>
</tr>
<tr>
<td>STATIC</td>
<td>4B9A0</td>
<td>102C</td>
</tr>
<tr>
<td>ZSTRES</td>
<td>4E3D0</td>
<td>1440</td>
</tr>
<tr>
<td>THICK</td>
<td>4B810</td>
<td>25FE</td>
</tr>
<tr>
<td>T3AI1U</td>
<td>4C110</td>
<td>458</td>
</tr>
<tr>
<td>3ARP</td>
<td>4C5E8</td>
<td>1A6</td>
</tr>
<tr>
<td>HOLL0W</td>
<td>4CF80</td>
<td>98B</td>
</tr>
<tr>
<td>E3FIT2</td>
<td>4D939</td>
<td>98E</td>
</tr>
<tr>
<td>IHOLASCN*</td>
<td>4E290</td>
<td>267</td>
</tr>
<tr>
<td>IHOLATH2*</td>
<td>4E4F8</td>
<td>278</td>
</tr>
<tr>
<td>IHOLCOS*</td>
<td>4E770</td>
<td>280</td>
</tr>
</tbody>
</table>

IH$DARCS 4E290  DARCOS 4E290  IH$DARSN 4E2A8  DARSIN 4E2A8
DATAN2 4E4F8  IH$DATA2 4E4F8  DATAN 4E510  IH$DATA3 4E510
DCOS 4E770  IH$DCOS 4E770
<table>
<thead>
<tr>
<th>NAME</th>
<th>ORIGIN</th>
<th>LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>IHOLSIN</td>
<td>4E9F0</td>
<td>2F0</td>
</tr>
<tr>
<td>IHOLDRT</td>
<td>4ECE0</td>
<td>170</td>
</tr>
<tr>
<td>IHOLINTC</td>
<td>4EE50</td>
<td>2F0</td>
</tr>
<tr>
<td>FIGEXEC</td>
<td>4F140</td>
<td>14EE</td>
</tr>
<tr>
<td>FIOPRET</td>
<td>50630</td>
<td>3500</td>
</tr>
<tr>
<td>FIGCOMPL</td>
<td>53B30</td>
<td>2846</td>
</tr>
<tr>
<td>IHODCOMN</td>
<td>56378</td>
<td>E54</td>
</tr>
<tr>
<td>FIAP#</td>
<td>571D0</td>
<td>61C</td>
</tr>
<tr>
<td>IHOFCVT#</td>
<td>582C8</td>
<td>CEA</td>
</tr>
<tr>
<td>IHOETN#</td>
<td>59FB8</td>
<td>800</td>
</tr>
<tr>
<td>IHOEXP</td>
<td>597D0</td>
<td>3DC</td>
</tr>
<tr>
<td>IHOEIOS#</td>
<td>59898</td>
<td>118C</td>
</tr>
<tr>
<td>IHOFOS#</td>
<td>5A028</td>
<td>642</td>
</tr>
<tr>
<td>IHODRE#</td>
<td>5B370</td>
<td>624</td>
</tr>
<tr>
<td>INQUATBL</td>
<td>5B998</td>
<td>328</td>
</tr>
<tr>
<td>IHOUOPT</td>
<td>5BCC0</td>
<td>4C8</td>
</tr>
<tr>
<td>IHOFCVN#</td>
<td>5C168</td>
<td>416</td>
</tr>
<tr>
<td>IHOFCV#</td>
<td>5C5A0</td>
<td>8B8</td>
</tr>
<tr>
<td>IHOET#</td>
<td>5CE58</td>
<td>2AE</td>
</tr>
<tr>
<td>PHPOSTDP#</td>
<td>5D108</td>
<td>128</td>
</tr>
<tr>
<td>IHOFTEN#</td>
<td>5D230</td>
<td>220</td>
</tr>
<tr>
<td>BLK1</td>
<td>5D450</td>
<td>6904</td>
</tr>
<tr>
<td>BLK2</td>
<td>63D58</td>
<td>20</td>
</tr>
<tr>
<td>BLK4</td>
<td>63D78</td>
<td>2C</td>
</tr>
<tr>
<td>BLK7</td>
<td>63D98</td>
<td>4EC</td>
</tr>
<tr>
<td>BLK12</td>
<td>64298</td>
<td>348</td>
</tr>
<tr>
<td>BLKAA</td>
<td>645E0</td>
<td>348</td>
</tr>
<tr>
<td>BLKB</td>
<td>64928</td>
<td>998</td>
</tr>
</tbody>
</table>
DIAGNOSTIC MESSAGE DIRECTORY

IEN0132 ERROR - SYMBOL PRINTED IS AN UNRESOLVED EXTERNAL REFERENCE.
PRATT & WHITNEY AIRCRAFT ENGINEERING DIV
VER 10.0
PAGE 1
SERIAL 037212

THE PROGRAM MANAGEMENT AND SECURITY SYSTEM

PROGRAMS AND ALL SUPPORTING MATERIALS COPYRIGHT 1975 BY PAN SOPHIC 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,U498SNH983
***** ABOVE ACTION SATISFACTORILY COMPLETED *****

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

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

+WRITE WORK,U498SRDFUN
***** ABOVE ACTION SATISFACTORILY COMPLETED *****

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

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

+WRITE WORK,U498SNT137
***** 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,U477HAIN
***** ABOVE ACTION SATISFACTORILY COMPLETED *****
<table>
<thead>
<tr>
<th>SOURCE STATEMENTS</th>
<th>PROGRAM SIZE</th>
<th>SUBPROGRAM NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>105</td>
<td>4294</td>
<td>F0D</td>
</tr>
<tr>
<td>187</td>
<td>3568</td>
<td>ANALIZ</td>
</tr>
<tr>
<td>70</td>
<td>1244</td>
<td>RDODATA</td>
</tr>
<tr>
<td>42</td>
<td>1032</td>
<td>CALTH</td>
</tr>
<tr>
<td>77</td>
<td>1402</td>
<td>ITT983</td>
</tr>
<tr>
<td>19</td>
<td>464</td>
<td>NDT983</td>
</tr>
<tr>
<td>18</td>
<td>558</td>
<td>RDT983</td>
</tr>
<tr>
<td>110</td>
<td>3472</td>
<td>RDB137</td>
</tr>
<tr>
<td>33</td>
<td>1048</td>
<td>RSFUNC</td>
</tr>
<tr>
<td>58</td>
<td>1098</td>
<td>THX</td>
</tr>
<tr>
<td>49</td>
<td>1206</td>
<td>ITT983</td>
</tr>
<tr>
<td>114</td>
<td>3002</td>
<td>WTH137</td>
</tr>
<tr>
<td>58</td>
<td>1994</td>
<td>OBJTV</td>
</tr>
<tr>
<td>87</td>
<td>1730</td>
<td>CN3201</td>
</tr>
<tr>
<td>40</td>
<td>654</td>
<td>CN3202</td>
</tr>
<tr>
<td>249</td>
<td>3520</td>
<td>CN3203</td>
</tr>
<tr>
<td>105</td>
<td>1754</td>
<td>CN3204</td>
</tr>
<tr>
<td>122</td>
<td>2548</td>
<td>CN3205</td>
</tr>
<tr>
<td>493</td>
<td>6746</td>
<td>CN3206</td>
</tr>
<tr>
<td>55</td>
<td>936</td>
<td>CN3207</td>
</tr>
<tr>
<td>84</td>
<td>1390</td>
<td>CN3208</td>
</tr>
<tr>
<td>830</td>
<td>15164</td>
<td>CN3210</td>
</tr>
<tr>
<td>706</td>
<td>15790</td>
<td>COPE01</td>
</tr>
<tr>
<td>41</td>
<td>1352</td>
<td>COPE02</td>
</tr>
<tr>
<td>68</td>
<td>1246</td>
<td>COPE03</td>
</tr>
<tr>
<td>230</td>
<td>3454</td>
<td>COPE04</td>
</tr>
<tr>
<td>71</td>
<td>2210</td>
<td>COPE05</td>
</tr>
<tr>
<td>46</td>
<td>1118</td>
<td>COPE06</td>
</tr>
<tr>
<td>52</td>
<td>1470</td>
<td>COPE07</td>
</tr>
<tr>
<td>SOURCE STATEMENTS</td>
<td>PROGRAM SIZE</td>
<td>SUBPROGRAM NAME</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>63</td>
<td>846</td>
<td>COPE08</td>
</tr>
<tr>
<td>544</td>
<td>9868</td>
<td>COPE09</td>
</tr>
<tr>
<td>69</td>
<td>2038</td>
<td>COPE10</td>
</tr>
<tr>
<td>57</td>
<td>1134</td>
<td>COPE11</td>
</tr>
<tr>
<td>125</td>
<td>2844</td>
<td>COPE12</td>
</tr>
<tr>
<td>23</td>
<td>566</td>
<td>COPE13</td>
</tr>
<tr>
<td>95</td>
<td>2632</td>
<td>COPE14</td>
</tr>
<tr>
<td>63</td>
<td>1728</td>
<td>COPE15</td>
</tr>
<tr>
<td>29</td>
<td>878</td>
<td>COPE16</td>
</tr>
<tr>
<td>26</td>
<td>680</td>
<td>COPE17</td>
</tr>
<tr>
<td>49</td>
<td>1408</td>
<td>COPE18</td>
</tr>
<tr>
<td>157</td>
<td>3780</td>
<td>SIICON</td>
</tr>
<tr>
<td>248</td>
<td>3980</td>
<td>MAIN</td>
</tr>
</tbody>
</table>
DATA SET U498SFOD AT LEVEL 006 AS OF 03/01/82
DATA SET U498SFOD AT LEVEL 004 AS OF 02/08/82
DATA SET U498SFOD AT LEVEL 002 AS OF 11/16/81

SUBROUTINE FODUIR,NC,RPIH1,ALPHAI

C
   C LOCAL STRESS BIRD INGESTION PARAMETER

C
   IMPLICIT REAL*8(A-H,O-Z)
   REAL*4 VALT, BRSU, TIS
   COMMON /BLK 3/ NH, BLADES, BETA, THR, NAT, NC
   COMMON /BLKAA/ ALPHAI(21), XSC(21), YSC(21), XXY(21), YYY(21)
   COMMON /BLK 8/ NSB(21), INTE(21), VAR(21), THAX(21), RHALPHA(21)
   COMMON /LAY/ TH(7,25), BMAS(25), BPAH(25), BHDTH(25)
   COMMON /BIRD/ BLSMAX(21)
   COMMON /PLYL/ PLY(21,25,7), THETA(7), RHTH(7)
   COMMON /STR/ RBE(12, 12, 25, 21), FOUT(12, 21), SL(25, 21), SH(25, 21)

   DIMENSION TB(25), AREA(25), AREAF, TORSI(25), CT25(21)
   DIMENSION TB25(21), BIPI25(21), XXI25(21), ALPHAI(1)

C
   WRITE(6, 270)

C
   VIELD/100, 'LSBIP ANALYSIS **', //

C
   DO 500 KK = 1, NR
C
   XBIRD = 180.0
   NB = BLADES
   PI = 3.14159265
   ALPHA = ALPHAI(NSTA)
   RPM = RPM1 * 30. / PI
   VI = .42 * 12. * VFIELD * DCOS( ALPHA )
   NODE = (NSTA-1) * NC + 1
   V2 = PI * X(NODE) * RPM / 30. * DSINH( ALPHA )

C
   WRITE(6, 400) NSTA, NC, RPM, ALPHA, NB, VFIELD, XSC(NSTA)

C
   1 /P, 19X, 'RPM(RPM/REV/MIN) = ', RFO(7.2., 19X), 'ALPHA CHG(SI2RADIAN) = ',
   2 , E12.5. / 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
   ZERO MATRICES

C
   DO 260 I = 1,26
   TB1(I) = 0.
   TB2(I) = 0.
   TB3(I) = 0.
C DISTANCE TO CENTER OF LOAD APPLICATION FROM LE
C
ISH 0039
XF = 2. * PI * X(NODE) / NB / (V2/V1-1) / DCOS( ALPHA )
C
ISH 0040
P = IMPACT LOAD
C
ISH 0041
WRITE(6,401) XF,P
ISH 0042
401 FORMAT(19X,'XF DISTANCE = ',E12.5,1,19X,'P LOAD = ',E12.5,/)
CALL MATPFRN(TORS,25,1,'TORS')

C DISTANCE FROM L.E., XX
C
XX(I) = EWIDTH(I)/2.
DO 210 I = 2,NC


XX(I) = XX(I-1) + EWIDTH(I-1)/2. + EWIDTH(I)/2.
C CALL MATPFRN(XX,25,1,'XX')
C
C XBAR = DISTANCE FROM LOAD CENTER TO C.G.
C
TDX = 0.
DO 350 I = 1,NC


TDX = TDX + XX(I) * TS(I)
XBAR = TDX / AREA(NC) - XF/2.
C CALL MATPFRN(XX,25,1,'XX')
C
C SEARCH FOR MAXIMUM BIP FOR SECTION KK
C
SLSMAX(KK) = 0.
DO 405 I = 1,NC


IF(BIP(I) > SLSMAX(KK)) SLSMAX(KK) = BIP(I)
405 CONTINUE
C CALL MATPFRN(BIP,25,1,'BIP')
C
C BLMAX(KK) = 0.
DO 405 I = 1,NC


IF(BIP(I) < BLMAX(KK)) BLMAX(KK) = BIP(I)
405 CONTINUE
C

DATE 82.141/10.52.17 PIN.186**6
00114**6
00115
00116
00117
00118
00119
00120
00121
00122
00123
00124
00125
00126
00127
00128**6
00129**6
00130**6
00131**6
00132**6
00133**6
00134**6
00135**6
00136**6
00137**6
00138**6
00139
00140
00141
00142**6
00143
00144
00145
00146
00147
00148
00149
00150
00151
00152
00153
00154
00155
00156
00157
00158
00159
00160
00161
00162
00163
00164
00165
00166
00167
00168
00169
00170
00171
00172
00173
00174
00175
00176
00177
00178
00179
00180
00181
00182
00183
00184
00185
00186
00187
00188
00189
00190
00191
00192
00193
00194
00195
00196
00197
00198
00199
00200
00201
00202
00203
C 00173
ISN 0105 RETURN
ISN 0106 END

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

*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOMAPRINT NOSHTMT 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
REQUESTED OPTIONS:

+VERSION 1.3.0 (01 MAY 80)

SOURCE,NOLIST,NODECK,OPT(3),AUTOBL(INC),NOW

OPTIONS IN EFFECT: NAME(HAME) OPTIMIZE(3) LINFOCOUNT(60) SIZE(INCH) AUTOBL(INC)

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

C DATA SET U988SAHLIZ AT LEVEL 019 AS OF 03/16/82 00001
C DATA SET U988SAHLIZ AT LEVEL 016 AS OF 03/15/82 00002
C DATA SET U988SAHLIZ AT LEVEL 017 AS OF 03/15/82 00003
C DATA SET U988SAHLIZ AT LEVEL 016 AS OF 03/12/82 00004

ISH 0002 SUBROUTINE ANALIZ (ICALC) 00005
ISH 0003 COMMON /KHUI37/ BLADWT, FREQ(100) 00006
ISH 0004 COMMON /KU80/ INS, INS 00007
ISH 0005 COMMON /ANL03/ RPM, FIS, DFM, TOLL, ROOT, RPMN, RPMMN 00008
ISH 0006 COMMON /ANL04/ HSTA, MST1, MST2, NOACR, ILT, LGOM, ISTE 00009
ISH 0007 COMMON /ANL05/ R(21), BGRS(21), TBG(21), THSTH(21), ALPH(21), 00100
ISH 0008 COMMON /ANL17/ XPS(3,53,21), THX(21), TCC(21), BMX(21) 00101
ISH 0009 COMMON /ANL30/ NTIS, NRF, IST(21), VALT(21), BRSV 00102
ISH 0010 COMMON /ANL33/ DLED, DTED, DROOT, DTIPD, TTID, TLTD, HCD 00103
ISH 0011 COMMON /ANL54/ ND(5), DLSV(5) 00104
ISH 0012 COMMON /PILY1/ PLY(21,25,7), THETA(7), RHOD(7) 00105
ISH 0013 COMMON /GLOB/ OVF, FH(5), DLAR(5), THVAL(21), RF(5,4) 00106
ISH 0014 COMMON /PRP1/ PRP(21), D5, D6, D7, D8, D9, D10, D11, D12 00107
ISH 0015 COMMON /VALT1/ VALT(21) 00108
ISH 0016 COMMON /BCC/ BCC(21) 00109
ISH 0017 COMMON /ANL30/ NTIS, NRF, IST(21), VALT(21), BRSV 00110
ISH 0018 COMMON /ANL54/ ND(5), DLSV(5) 00111
ISH 0019 COMMON /ANL54/ ND(5), DLSV(5) 00112
ISH 0020 COMMON /PILY1/ PLY(21,25,7), THETA(7), RHOD(7) 00113
ISH 0021 COMMON /GLOB/ OVF, FH(5), DLAR(5), THVAL(21), RF(5,4) 00114
ISH 0022 COMMON /PRP1/ PRP(21), D5, D6, D7, D8, D9, D10, D11, D12 00115
ISH 0023 COMMON /VALT1/ VALT(21) 00116
ISH 0024 COMMON /BCC/ BCC(21) 00117
ISH 0025 COMMON /ANL54/ ND(5), DLSV(5) 00118
ISH 0026 COMMON /ANL54/ ND(5), DLSV(5) 00119
ISH 0027 COMMON /PILY1/ PLY(21,25,7), THETA(7), RHOD(7) 00120
ISH 0028 COMMON /GLOB/ OVF, FH(5), DLAR(5), THVAL(21), RF(5,4) 00121
ISH 0029 COMMON /PRP1/ PRP(21), D5, D6, D7, D8, D9, D10, D11, D12 00122
ISH 0030 COMMON /VALT1/ VALT(21) 00123
ISH 0031 COMMON /BCC/ BCC(21) 00124
ISH 0032 COMMON /ANL54/ ND(5), DLSV(5) 00125
ISH 0033 COMMON /PILY1/ PLY(21,25,7), THETA(7), RHOD(7) 00126
ISH 0034 COMMON /GLOB/ OVF, FH(5), DLAR(5), THVAL(21), RF(5,4) 00127
ISH 0035 COMMON /PRP1/ PRP(21), D5, D6, D7, D8, D9, D10, D11, D12 00128
ISH 0036 COMMON /VALT1/ VALT(21) 00129
ISH 0037 COMMON /BCC/ BCC(21) 00130
ISH 0038 COMMON /ANL54/ ND(5), DLSV(5) 00131
ISH 0039 COMMON /PILY1/ PLY(21,25,7), THETA(7), RHOD(7) 00132
ISH 0040 COMMON /GLOB/ OVF, FH(5), DLAR(5), THVAL(21), RF(5,4) 00133
ISH 0041 COMMON /PRP1/ PRP(21), D5, D6, D7, D8, D9, D10, D11, D12 00134
IF (NRF .NE. 0) CALL RDT983 (ICALC) 00055
CALL THAX (ICALC) 00056
CALL WTHI37 (ICALC) 00057
DO 150 I = 1, NTIS 00058
THVAL(I) = VALT(I) 00059
150 CONTINUE 00060
BRCC = BRSV 00061
CALL MYTIME (ITIME) 00062
ITIMEC = 3 00063
WRITE (26,3000) ITIMEC, ITIME 00064
RETURN 00065
200 IF (ICALC .GT. 2) GO TO 300 00066
CALL MYTIME (ITIME) 00067
ITIMEC = 4 00068
WRITE (26,3000) ITIMEC, ITIME 00069
NBR = ERS + .05 00070
IF (NCD .NE. 2) GO TO 210 00071
HLRTIO = (DTIP + DROOT) / (R(NSTA) - R(1)) 00072
ECRTIO = (DLE + DTE) / BRCC 00073
WRITE (I61,1000) 00074
WRITE (I61,238) DLE, DTE, DROOT, DTIP, TTI, TLT 00075
1, HLRTIO, ECRTIO 00076
GO TO 220 00077
WRITE (I61,1030) 00078
VAL = (2.0 * TIS + TIC) / (THKVAL(I)) 00079
WRITE (I61,238) TIS, TIC, PCBA, PCGE, VAL 00080
CALL THI37 (ICALC) 00081
CALL CALCTH 00082
CALL WTHI37 (ICALC) 00083
CALL THAX (ICALC) 00084
CALL WTHI37 (ICALC) 00085
CALL MTHI37 00086
CALL RSFRC 00087
CALL OBJTV 00088
WRITE (I61,930) 00089
WRITE (I61,238) OBJF, OBJFU 00090
WRITE (I61,930) 00091
WRITE (I61,238) TOVB(I), I = I, NTIS 00092
WRITE (I61,930) 00093
WRITE (I61,930) 00094
WRITE (I61,930) 00095
WRITE (I61,930) 00096
WRITE (I61,930) 00097
WRITE (I61,930) 00098
WRITE (I61,930) 00099
WRITE (I61,930) 00100
WRITE (I61,930) 00101
WRITE (I61,930) 00102
WRITE (I61,930) 00103
WRITE (I61,930) 00104
WRITE (I61,930) 00105
WRITE (I61,930) 00106
WRITE (I61,930) 00107
WRITE (I61,930) 00108
WRITE (I61,930) 00109
WRITE (I61,930) 00110
WRITE (I61,930) 00111
WRITE (I61,930) 00112
WRITE (I61,930) 00113
+VERSION 1.3.0 (01 MAY 60) ANALIZ SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 82.141/10.52.24 PAGE 3

ISH 0105 FOMAX = AMAX1(FOMAX,BMAX)
ISH 0106 IF (FOMAX .EQ. BMAX) RMAX = R(K)
ISH 0108 WRITE (16I,230) FOMAX, RMAX
ISH 0109 234 CONTINUE
ISH 0110 WRITE (16I,970)
ISH 0111 WRITE (16I,238) (FOMAX(I), I=NRBS,NSTA)
ISH 0112 236 FORMAT (/8(3X,E12.5))
ISH 0113 238 FORMAT (4(3X,E12.5))
ISH 0114 DO 237 I = 1,8
ISH 0115 SNA1S(I) = SMAX1(I)
ISH 0116 SNA2S(I) = SMAX2(I)
ISH 0117 SNA3S(I) = SMAX3(I)
ISH 0118 SNA4S(I) = SMAX4(I)
ISH 0119 SMAX1S(I) = SMAX1(I)
ISH 0120 237 CONTINUE
ISH 0121 WRITE (16I,940)
ISH 0122 WRITE(16I,238) (SMAX1S(I), I=1,8)
ISH 0123 WRITE (16I,950)
ISH 0124 WRITE(16I,238) (SNA2S(I), I=1,8)
ISH 0125 WRITE (16I,960)
ISH 0126 WRITE(16I,238) (SNA3S(I), I=1,8)
ISH 0127 WRITE (16I,980)
ISH 0128 WRITE (16I,238) (SNA4S(I), I=1,8)
ISH 0129 WRITE (16I,990)
ISH 0130 WRITE (16I,238) (SMAX1S(I), I=1,8)
ISH 0131 CALL MYTIME (ITIME)
ISH 0132 ITIMEC = 5
ISH 0133 WRITE (26,3000) ITIMEC, ITIME
ISH 0134 IF (NRF .EQ. 0) GO TO 240
ISH 0136 CALL IT903 (ICAC)
ISH 0137 WRITE (16I,920)
ISH 0138 DO 240 I = 1,NRF
ISH 0139 WRITE (16I,910) ND(I), DLSV(I)
ISH 0140 DLAR(I) = DLSV(I)
ISH 0141 240 CONTINUE
ISH 0142 CALL MYTIME (ITIME)
ISH 0143 ITIMEC = 6
ISH 0144 WRITE (26,3000) ITIMEC, ITIME
ISH 0145 RETURN
ISH 0146 300 CONTINUE
ISH 0147 CALL MYTIME (ITIME)
ISH 0148 ITIMEC = 7
ISH 0149 WRITE (26,3000) ITIMEC, ITIME
ISH 0150 CALL RNH137 (ICAC)
ISH 0151 CALL CALTCH
ISH 0152 CALL TMAX (ICAC)
ISH 0153 CALL WTH137 (ICAC)
ISH 0154 ICAC = 4
ISH 0155 C CALL WTH137 (ICAC)
ISH 0156 ICAC = 3
ISH 0157 IF (NRF .EQ. 0) GO TO 350
ISH 0158 INAVE = -1
ISH 0159 IND = 4
ISH 0160 MODE = 1
ISH 0161 CALL WT903 (ICAC,MODE,IND,INAVE)
ISH 0162 MODE = 2
ISH 0163 CALL WT903 (ICAC,MODE,IND,INAVE)
ISH 0164 MODE = 3
ISH 0165 CALL WT903 (ICAC,MODE,IND,INAVE)
ISH 0166 350 CONTINUE
ISH 0167 NBR5 = ERS + .05
ISH 0168 WRITE (I6I,900) (TIX(I),I=HBR5,NSTA)
ISH 0169 WRITE (I6I,900) (TCC(I),I=HBR5,NSTA)
ISH 0170 900 FORMAT (/3I2X,F10.6))
ISH 0171 910 FORMAT (5X,15,5X,E12.5)
ISH 0172 920 FORMAT (/6X,'NO-SAVE DELAERO-SAVE')
ISH 0173 930 FORMAT (/6X,'HEIGHT OBJECT FUNCTION')
ISH 0174 940 FORMAT (/5X,'SHAX1(I) I = 1,8')
ISH 0175 950 FORMAT (/5X,'SHAX2(I) I = 1,8')
ISH 0176 960 FORMAT (/5X,'SHAX3(I) I = 1,8')
ISH 0177 970 FORMAT (/5X,'LSBIP(I) I = 1,NTIS')
ISH 0178 980 FORMAT (/5X,'SHAXLE(I) I = 1,8')
ISH 0179 990 FORMAT (/5X,'SHAXTE(I) I = 1,8')
ISH 0180 1000 FORMAT (/6X,'BLE',12X,'LTE',11X,'DROGT',10X,'OTIP',
ISH 0181 1 11X,'TII',11X,'TBT',12X,'HlRAT',O6X,'ECONF')
ISH 0182 1010 FORMAT (/5X,'THETA(I) I = 1,7')
ISH 0183 1020 FORMAT (/5X,'T/BI(I) I = 1,NTIS')
ISH 0184 1030 FORMAT (/6X,'TIS',12X,'TIC',11X,'PCBA',11X,'PCGE',9X,'T RATIO')
ISH 0185 CALL MYTIME (ITIME)
ISH 0186 ITIMEC = 8
ISH 0187 WRITE (26,3000) ITIMEC, ITIME
ISH 0188 RETURN
ISH 0189 END

*OPTIONS IN EFFECT=NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOIDL(NONE)
*OPTIONS IN EFFECT=SOURCE EBCDIC NOLIST NODECK OBJECT NOAP NOFORMAT GOSTMT NODEREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS= SOURCE STATEMENTS = 167, 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),AUTOCLONE,NOLC
OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOCLONE(NONE)

SOURCE EBCDIC NOLIST NODECK OBJECT NOLMAP NOLIST COSTH SOURCE EXTRASF NOLC NOLANSF TERM IBM FLAG(I)

DATA SET U498SRDATA AT LEVEL 005 AS OF 03/15/82 00001
DATA SET U498SRDATA AT LEVEL 004 AS OF 02/24/82 00002
DATA SET U498SRDATA AT LEVEL 003 AS OF 02/23/82 00003
DATA SET U498SRDATA AT LEVEL 002 AS OF 01/29/82 00004
DATA SET U498SRDATA AT LEVEL 001 AS OF 11/12/81 00005
DATA SET U500RDATA AT LEVEL 003 AS OF 07/16/81 00006
DATA SET U500RDATA AT LEVEL 002 AS OF 06/10/81 00007
DATA SET U500RDATA AT LEVEL 001 AS OF 06/06/81

SUBROUTINE RODATA

COMMON /UISO/ ISI , I6I
COMMON /ANAL5/ ND(5) , DLSV(5)
COMMON /ANAL30/ NTIS , NRF , IST(21) , VALT(21) , BRSV
COMMON /ANAL31/ NSFA , ISTH(21)
COMMON /ANAL32/ RPMIN , RPMAX , EORD(4) , NRORD
COMMON /ANAL33/ DLED , DTED , DROOT , DTIPD , TTID , TLT , NC
COMMON /ANAL04/ NSTA , NHT , BST2 , NACR , ILT , LONG , ISTE , I1DEN , EB , POB , ERC , BRS , BTA , IOPP
COMMON /PLY/ PLY(21,25,7) , THETAD(7) , RHI(7)
COMMON /GLCSCH/ OBJF , FCH(5) , DLAR(5) , THKVAL(21) , RF(5,4)
1 , DRCC , FODLSB(21) , DLE , DTE , DROOT , DTIP , TLT , TLT
2 , OBJFUN , SMASIS(8) , SMASIS(8)
3 , SMAXSIS(8) , SMAXSIS(8) , THETA(7) , HLRRT0 , ECRT0 , TQVAL(21)
4 , FMAX , TIS , TIC , PCBA

DOUBLE PRECISION DLED , DTED , DROOT , DTIPD , TTID , TLT

READ NTIS : NUMBER OF THICKNESS INPUT STATIONS (MAX OF 21) 000027
READ NRF : NUMBER OF ROOTS CALCULATED BY FLUTTER ANALYSIS (MAX OF 5) 000029
READ NSFA : NUMBER OF W137 OUTPUT STATIONS USED FOR THE FLUTTER ANALYSIS (MAX OF 21) 000030
READ NRORD : NUMBER OF ORDERS FOR RESONANCE FUNCTION CALCULATION (MAX OF 4) 000031
READ (ISI,900) NTIS , NRF , NSFA , NRORD , BRSV 000032
READ NTIS VALUES OF STATION NUMBER AND THICKNESS 000033
READ (ISI,901) (IST(I),VALT(I) , I = 1,NTIS) 000034
READ NRF VALUES OF NODAL DIAMETER (1 PER ROOT) FOR INITIAL STARTING VALUES FOR THE FLUTTER CALCULATION 000035
READ W137 OUTPUT STATION NUMBERS TO BE USED 000036
READ IN THE FLUTTER ANALYSIS
CONTINUE
READ (151,903) (ISTH!, I = 1,NSFA)
RBS = RBS + .05
NCK = NCK - NCRS + 1
IF (NCK .NE. NSFA) GO TO 70
DO 65 I = 1,NSFA
65 CONTINUE

CONTINUE
READ (151,904) RPMIN, RPMIX, EORD(1), EORD(2)
1, EORD(3), EORD(4)

READ CODE TO DETERMINE IF FOIL IS HOLLOW OR COMPOSITE
NCD = 1 (HOLLOW)
NCD = 2 (COMPOSITE)

READ (151,905) NCO
IF (NCO .EQ. 2) GO TO 90

READ DATA ASSOCIATED WITH HOLLOW FOIL

DLE = DISTANCE TO HOLE FROM LEADING EDGE
DTE = DISTANCE TO HOLE FROM TRAILING EDGE
DROOT = DISTANCE TO HOLE FROM FOIL ROOT
DTIP = DISTANCE TO HOLE FROM FOIL TIP
TTI = THICKNESS OF TITANIUM
TLT = THICKNESS OF EORCIC TITANIUM

READ (151,906) DLE, DTE, DROOT, DTIP, TTI, TLT

READ (151,907) (THETA(I), I=1,7)
DO 100 I = 1,7
THETAD(I) = THETA(I)
100 CONTINUE

READ DATA ASSOCIATED WITH COMPOSITE FOIL

TIS = TITANIUM SKIN THICKNESS
TIC = TITANIUM CENTER THICKNESS
PCBA = PERCENT OF BORON ALUMINUM

READ (151,908) TIS, TIC, PCBA
TISD = TIS
TICD = TIC
PCBAD = PCBA
*VERSION 1.3.0 (01 MAY 80) SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 02.14/10.52.33 PAGE 1*

**REQUESTED OPTIONS:** SOURCE, NOMAP, NOXREF, NOLIST, NODECK, OPT(3), AUTOOBJ, NONE, XREF

**OPTIONS IN EFFECT:** NAME(HAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOOBJ INCHIE

**SOURCE EBCDIC NOLIST NODECK OBJECT NOPAP NOFORMAT GOSTH IFXREF NOALC NOANSF TERM IBM FLAG(I)**

```fortran
DATA SET U498SCALCT AT LEVEL 010 AS OF 01/29/82 00001
DATA SET U498SCALCT AT LEVEL 009 AS OF 01/28/82 00002
DATA SET U498SCALCT AT LEVEL 008 AS OF 10/19/81 00003
DATA SET U500CALCTH AT LEVEL 007 AS OF 07/23/81 00004
DATA SET U500CALCTH AT LEVEL 006 AS OF 07/17/81 00005
DATA SET U500CALCTH AT LEVEL 005 AS OF 07/16/81 00006
DATA SET U498SCALCT AT LEVEL 004 AS OF 06/17/81 00007
DATA SET U498SCALCT AT LEVEL 003 AS OF 06/16/81 00008
DATA SET U498SCALCT AT LEVEL 002 AS OF 06/15/81 00009
DATA SET U498SCALCT AT LEVEL 001 AS OF 06/14/81 00010

SUBROUTINE CALCTH

C DETERMINE RADIUS AT EACH INPUT THICKNESS STATION

C

C WRITE (I6I,910) NISTA, BRS
DO 10 I = 1,NTIS
    N = IST(I)
    RIST(I) = R(N)
    THKVAL(I) = THKVAL(I)
    WRITE (I6I,910) I, THKVAL(I), THKVAL(I)
    WRITE (I6I,910) I, RIST(I)

10 CONTINUE

C CALCULATE THICKNESS AT EACH W137 INPUT RADIUS USING
C A LINEAR FIT OF THKVAL

C

C N = 2
C WRITE (I6I,920)
ISN 0016
ISN 0017
ISN 0018
920 FORMAT (I5,2F12.8)

C N = RIST(I)
C WRITE (I6I,910)
ISN 0019
ISN 0020
TCC(NBRS) = THKVAL(N)
ISN 0021
ECC(NBRS) = BRCC
ISN 0022
WRITE (I6I,910) NBRS, TCC(NBRS), BCC(NBRS)
ISN 0023
IF (N .GT. NTIS) GO TO 30
ISN 0024
N = NBRS + 1
ISN 0025
DO 20 I = N,NSTA
20 CONTINUE

16 IF(R(I) .GE. RIST(I)) GO TO 16
ISN 0026
15 IF(R(I) .LE. RIST(N)) GO TO 16
ISN 0027
N = N + 1
ISN 0028
IF (N .GT. NTIS) GO TO 30
ISN 0029
16 TCC(I) = THKVAL(N) - (THKVAL(N) - THKVAL(N) * R(I) - RIST(N))
1 / (RIST(N) - RIST(N))
ISN 0030
WRTIE (I6I,910) I, TCC(I), BCC(I)
ISN 0031
ISN 0032
ISN 0033
ISN 0034
ISN 0035
ISN 0036
ISN 0037
20 CONTINUE
```

```fortran
DATA SET U498SCALCT AT LEVEL 001 AS OF 06/14/81 00010
DATA SET U498SCALCT AT LEVEL 000 AS OF 06/13/81 00001
DATA SET U498SCALCT AT LEVEL 000 AS OF 06/10/81 00002
DATA SET U498SCALCT AT LEVEL 000 AS OF 06/09/81 00003
DATA SET U498SCALCT AT LEVEL 000 AS OF 06/08/81 00004
DATA SET U498SCALCT AT LEVEL 000 AS OF 06/07/81 00005
DATA SET U498SCALCT AT LEVEL 000 AS OF 06/06/81 00006
DATA SET U498SCALCT AT LEVEL 000 AS OF 06/05/81 00007
DATA SET U498SCALCT AT LEVEL 000 AS OF 06/04/81 00008
DATA SET U498SCALCT AT LEVEL 000 AS OF 06/03/81 00009
DATA SET U498SCALCT AT LEVEL 000 AS OF 06/02/81 00010
```

**END OF DOCUMENT**
C   WRITE (I6I,910) N , THKVEL(I)
00055
ISN 0039       RETURN 00056
ISN 0040       30 WRITE (I6I,900) N , NTIS 00057
ISN 0041       900 FORMAT (' N = ',15,' NTIS = ',15,' N EXCEEDS NTIS'/) 00058
ISN 0042       STOP 00059
ISN 0053       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 GOSTMT 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
+VERSION 1.3.0 (01 MAY 60) SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 02.141/10.52.36 PAGE 1
REQUESTED OPTIONS: SOURCE,NOMAP,NOSREF,NOLIST,NODECK,OPT(3),AUTOenburg(NONE),NOALC
OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOenburg(NONE)
SOURCE EBCDIC NOLIST NODECK NOSAP FORMAT GOSTINT NOANSF TERM IBM FLAG(I)

C DATA SET U4985IT983 AT LEVEL 009 AS OF 03/10/62
C DATA SET U498SIT983 AT LEVEL 008 AS OF 11/09/61
C DATA SET US5005IT983 AT LEVEL 007 AS OF 06/09/61
C DATA SET US5005IT983 AT LEVEL 006 AS OF 06/03/61

ISN 0002 SUBROUTINE IIT983 (ICALC) 00004
ISN 0003 COMMON /USOS/ I51, I61 00005
ISN 0004 COMMON /ANAL3/ NTS, NFS, IST(21), VALT(21) 00006
ISN 0005 COMMON /ANAL5/ ND(5), DLSV(5) 00007
ISN 0006 COMMON /SOLK/ FLAP(21,21), THIST(21,21), ACOEF(21,4), ZODALD(21) 00008
1 RED(21), DELAER(21), F2HSD, F2PSP 00009
ISN 0007 COMMON /JHFCxK/ JHP(5), MODE 00010
ISN 0008 COMMON /ANAL11/ HAY, MAX, ISTR, N8B, DENH, DENA, ED 00011
1 DIAH, BcLADES, POID 00012
ISN 0009 NHIN = 2 00013
ISN 0010 NHIN = -2 00014
ISN 0011 NMAX = (BLADES + .05 / 2. ) -2. 00015
ISN 0012 NMAX = -NMAX 00016
ISN 0013 LIN = NFS 00017
ISN 0014 DO 5 I = 1,5 00018
ISN 0015 JHP(I) = 0 00019
ISN 0016 1 CONTINUE 00020
ISN 0017 WRITE (161,110) 00021
ISN 0018 MODE = 1 00022
ISN 0019 10 INC = 1 00023
ISN 0020 IHAVE = 1 00024
ISN 0021 IF (ND(MODE) .NE. NHIN) GO TO 13 00025
ISN 0022 ND(MODE) = NHIN 00026
ISN 0023 GO TO 17 00027
ISN 0024 13 IF (ND(MODE) .EQ. MNMAX .OR. ND(MODE) .EQ. NMIN) 00028
1 ND(MODE) = ND(MODE) + 1 00029
ISN 0025 IF (ND(MODE) .EQ. MNMAX .OR. ND(MODE) .EQ. NMIN) 00030
1 ND(MODE) = ND(MODE) - 1 00031
ISN 0026 17 IF (ND(MODE) .LT. 0 ) IHAVE = -1 00032
ISN 0027 NODE = IABS (ND(MODE)) 00033
ISN 0028 CALL WTT983 (ICALC,MODE,NODE,IHAVE) 00034
ISN 0029 CALL MNT983 00035
ISN 0030 DELSAX = DELAER(1) 00036
ISN 0031 NSAV = ND(MODE) 00037
ISN 0032 WRITE (161,100) DELSAX, DELAER(1), NSAV, ND(MODE) 00038
ISN 0033 QOD = ND(MODE) 00039
ISN 0034 CALL NDT983 (NOD,NODE,IHAVE,INC) 00040
ISN 0035 ND(MODE) = NODE * IHAVE 00041
ISN 0036 CALL WTT983 (ICALC,MODE,NODE,IHAVE,INC) 00042
ISN 0037 CALL MNT983 00043
ISN 0038 IF (DELAER(1) .EQ. DELSAX) GO TO 40 00044
ISN 0039 IF (DELAER(1) .LT. DELSAV) GO TO 20 00045
ISN 0039 INC = -1 00046
ISN 0040 WRITE (161,100) DELSAX, DELAER(1), NSAV, ND(MODE) 00047
ISN 0041 ND(MODE) = ND(MODE) + INC 00048
ISN 0042 IF (ND(MODE) .EQ. 1) ND(MODE) = NHIN 00049
ISN 0043 GO TO 30 00050
ISN 0044 20 DELSAV = DELAER(1) 00051
ISN 0045 NSAV = ND(MODE) 00052
ISN 0046 WRITE (161,100) DELSAX, DELAER(1), NSAV, ND(MODE) 00053
ISN 0047 30 ND = ND(MODE) 00054

192
ISH 0056 CALL NDT983 (NCD,MODE,INAVE,INC) 00055
ISH 0057 ND(MODE) = NODE * INAVE 00056
ISH 0058 CALL WTT983 (ICALC,MODE,NODE,INAVE) 00057
ISH 0059 CALL MRT983 00058
ISH 0060 IF (DELAER(1) .GE. DELSAV) GO TO 40 00059
ISH 0062 DELSAV = DELAER(1) 00060
ISH 0063 NSAV = ND(MODE) 00061
ISH 0064 WRITE (16I,100) DELSAV, DELAER(1), NSAV, ND(MODE) 00062
ISH 0065 GO TO 30 00063
ISH 0066 40 CONTINUE 00064
ISH 0067 WRITE (16I,100) DELSAV, DELAER(1), NSAV, ND(MODE) 00065
ISH 0068 ND(MODE) = NSAV 00066
ISH 0069 DLSV(MODE) = DELSAV 00067
ISH 0070 JP(MODE) = 0 00068
ISH 0071 MODE = MODE + 1 00069
ISH 0072 IF (MODE .GT. LIMIT) GO TO 50 00070
ISH 0074 GO TO 10 00071
ISH 0075 50 RETURN 00072
ISH 0076 100 FORMAT (5X,E12.5,5X,E12.5,5X,I5,5X,I5) 00073
ISH 0077 110 FORMAT (5X,E12.5,5X,E12.5,5X,9X,'NO SAVE',4X,'NO') 00074
ISH 0078 END 00075
*OPTIONS IN EFFECT* NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOCLASS(NONE)
*OPTIONS IN EFFECT* SOURCE EDCOIC NOLIST NODECK NOEPIF NOFORMAT GOSTMT NOXREF NOALIGN 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
SUBROUTINE NDT983 (NOD, NODE, NAVE, INC)

THIS ROUTINE SELECTS A NODAL DIAMETER (NOD) BETWEEN \((-\frac{NB}{2} - 2\)) AND \((-2\)) AND BETWEEN 2 AND \((\frac{NB}{2} - 2\)) AND SETS NODE TO THE \(\text{ABS}(NOD)\) AND THE WAVE DIRECTION EQUAL TO THE SIGN OF NOD. THIS ROUTINE WEIGHS BY 10 INC UNTIL DELTA AERO IN THE CALLING ROUTINE IS MINIMIZED.

COMMON /ANAL11/ MAY, NAX, ISTR, HBB, DENG, DENA, ED

1, DIAND, BLADES, POID

COMMON /JMPCHK/ JMP(5), MODE

NMIN = 2

NMAX = \((\text{BLADES} - 0.05) / 2.\) - 2.

IF ( IABS(NOD) .GT. NMIN .AND. IABS(NOD) .LT. NMAX ) GO TO 10

IF ( JMP(MODE) .GT. 0 ) GO TO 10

JMP(MODE) = 1

NOD = NOD \((-1\))

GO TO 20

10 NOD = NOD + INC

20 NAVE = 1

IF ( NOD .LT. 0 ) NAVE = -1

RETURN

END

*OPTIONS IN EFFECT* NAME(main) OPTIMIZE(3) LINCOUNT(60) SIZE(MAX) AUTOCLIB(NCHE)

*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NOFORMAT GOSTIT NOLREF NOAHC NOAHSF TERM IBM FLAG(I)

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

*STATISTICS* NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****
DATA SET U498SRD983 AT LEVEL 002 AS OF 06/04/81
DATA SET U500RDT983 AT LEVEL 001 AS OF 05/12/81

READ TITLE CARD
READ (IXI,900) TITL
READ CONTROL CARD
READ (IXI,901) MACH,MAERO,NUMA,MHI37,INH,IBETA,IGUST
READ (IXI,900) TIAERO
READ AERO DATA
DO 20 I = 1,MAERO
READ (IXI,902) RAERO(I),ANACH(I),VEL1(I),TEMP1(I)
1,PRES(I),PSI1(I),BETAI(I)
20 CONTINUE
RETURN

*OPTIONS IN EFFECT=SOURCE EBCDIC NOLIST NODECK OBJECT NCHAR NOFORMAT GOSTHT NOSRF NOALC NOANSF TERM IBM FLAG(I)
*OPTIONS IN EFFECT=SOURCE EBCDIC NOLIST NODECK OBJECT NCHAR NOFORMAT GOSTHT NOSRF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 18, PROGRAM SIZE = 558, SUBPROGRAM NAME = RDT983
*STATISTICS* NO DIAGNOSTICS GENERATED

******* END OF COMPILATION *******
3016K BYTES OF CORE NOT USED
DATA SET U99SDRD137 AT LEVEL 004 AS OF 02/01/82
DATA SET U99SRRD137 AT LEVEL 003 AS OF 11/23/81
DATA SET U99SDRD137 AT LEVEL 002 AS OF 05/28/81
DATA SET U905SRD137 AT LEVEL 001 AS OF 04/06/81

SUBROUTINE RCH137 (ICALC)
COMMON /AUGS/ I5I, I5I
COMMON /ALAN0/ NTEST
COMMON /ALAN02/ TLTI, I61
COMMON /ALAN03/ RFN, FN, DFN, TOLL, ROOT, DRPM, RPMH, RPMX
COMMON /ALAN04/ NSTA, NST1, NST2, NOACR, I1L, IONG, ISTE
1 DEHB, GB, POIB, BR, SERIES, BSR, BTS, IGPP
COMMON /ALAN05/ R(31), BCBR(31), TOSH(31), ALPH(31), 1
1 SDB(31), RLE(31), RTE(31), ADWT(31), O(31)
COMMON /ALAN06/ TLTA(31), TLTL(31), XADNT(31), YADNT(31)
1 AKX(31), AKY(31), AKZ(31)
COMMON /ALAN07/ XP(3,53,21), HPH(21)
COMMON /ALAN08/ THE, THER, AKXH, AKMAX, ERANG
COMMON /ALAN09/ WTS1, XSSH1, YSSH1, XSHI, YSHI, 1
1 AKSH1, SHPOI, AKSH2, XSSH2, 1
COMMON /ALAN10/ WT22, XSSH2, YSSH2, ARSH2, XISH2, YISH4,
1 AKSH2, SHFOI, AKSH2
COMMON /ALAN11/ MAY, NAX, ISTR, Nb, DENB, DHA, ED
1 DIAH, BLADES, POIB
COMMON /ALAN12/ RADS(32), THE(32)
COMMON /ALAN13/ SIGDOM, RADST(32), TANST(32)
COMMON /ALAN14/ ELP, WP, XB, DNP, NNP, DNP, EL, W
1 HS, DS, WZ, DT, RZ, THZ, TTH, DF
2 NT, EMU, DLI, RRT, BNP, NBP, W, RDR
COMMON /ALAN15/ SA, SB, SC, SD, SE, SF, SG, SH, SI, SJ
1 SK, SL, SH, REF, W1, T11, R2, D2, D3, D4, D5, D6, D7
2 DLR, T1, T2, T3, T4, RA, SETD
COMMON /ALAN16/,XSH(30),VAL(30,12), III
DIMENSION X(3,53)
READ FIRST FOUR CARDS OF THE W137 INPUT
CARD 1 --- NEW CASE CONTROL CARD
CARD 2 --- TITLE CARD AND RIM AND RIM PROPERTIES OPTIONS
CARD 3 --- SPEED, FREQUENCIES, ROOT, ECT.
CARD 4 --- W137 CONTROL CARD
IXI = I5I
IF (ICALC .GT. 1) IXI = 24
IF (ICALC .EQ. 5) IXI = 5
READ (IXI,900) NTEST
READ (IXI,901) ITTL, IOP1, IOP2, IOP3
READ (IXI,902) RFN, FN, DFN, TOLL, ROOT, DRPM, RPMH, RPMX
READ (IXI,903) NSTA, NST1, NST2, NOACR, I1L, IONG, ISTE, DENB, EB,
1 POIB, BR, SERIES, BTS, IBPP
DO 20 I = 1, NSTA
READ (IXI,904) R(1), BCBR(I), TOSH(I), ALPH(I), SDB(I), 1
1 RLE(I), RTE(I), ADWT(I), O(I)
READ TILT AND SPRING CARD IF:
1 TILT IS INPUT AT EACH STATION (ILT = 1)
C 2--- TILT IS INPUT AT THE TIP STATION (ILT = 2 ) AND THE TIP STATION IS BEING READ ( I = NO. OF STATIONS )
C 3--- ADDED WEIGHT HAS BEEN INPUT

ISN 0032 IF (ILT .EQ. 1 ) GO TO 5
ISN 0034 IF (ILT .EQ. 2 ) AND I .EQ. NSTA GO TO 5

ISN 0035 5 READ (IXI,905) TLTA(I) , TLTT(I) , XADNT(I) , YADNT(I) , AKX(I) , AKY(I) , AKZ(I)

ISN 0036 IF T/B IS POSITIVE --- STOP READING AIRFOIL DATA
ISN 0037 IF T/B IS NEGATIVE --- READ COORDINATES

ISN 0039 IF (TBI .GT. 0.0 ) GO TO 20

ISN 0040 N = O(I) + .05

ISN 0041 DO 10 JJ = 1,N

ISN 0042 READ (IXI,905) (XI(JI,IN) , IN = 1,N)

ISN 0043 10 CONTINUE

ISN 0044 DO 15 M = 1,N

ISN 0045 XP(1,M,I) = X(1,M)
ISN 0046 XP(2,M,I) = X(2,M)
ISN 0047 XP(3,M,I) = X(3,M)

ISN 0048 15 CONTINUE

ISN 0049 NPI(I) = N

ISN 0050 20 CONTINUE

C READ ROOT AND TIP ANGLE , M/O , AND ENTRAP ANGLE

ISN 0051 READ (IXI,906) THET , THER , AKMIN , AKMAX , BRANG

C READ SHROUD DATA (HST1 = 0,NO SHROUD )

ISN 0052 IF (HST1 .LT. 1 ) Go TO 30

ISN 0053 30 CONTINUE

ISN 0054 READ (IXI,907) WTS1 , XSHR1 , YSHR1 , ARSH1 , XISH1 , XISH3 , AKSH1 , SHPOI1 , ANGS1 , NSL

ISN 0055 IF (HST2 .LT. 2 ) Go TO 30

ISN 0056 30 CONTINUE

ISN 0057 READ (IXI,907) WTS2 , XSHR2 , YSHR2 , ARSH2 , XISH2 , XISH4 , 1 AKSH2 , SHPOI2 , ANGS2

C READ DISC CONTROL DATA

ISN 0058 30 CONTINUE

ISN 0059 READ (IXI,908) MAY , NAX , ISTR , NDB , DEND , DENA , ED 1 , DIAND , BLADES , POID

C READ DISC RADII AND THICKNESSES IF MAY .NE. 0

ISN 0060 IF (MAY .EQ. 0 ) Go TO 40

ISN 0061 40 CONTINUE

C READ BORE STRESS , STRESS AT EACH STATION , OR NO STRESS

ISN 0062 IF (ISTR) 50,70,60

ISN 0063 50 READ (IXI,910) SIGBOR

ISN 0064 GO TO 70

ISN 0065 70 CONTINUE
C  READ DOVETAIL INPUT OR SKIP TO RING INPUT
C  IF ( I0P1 .EQ. 0 ) GO TO 80
C  ISN 0070
C  ISN 0071
C  READ (IXI,902) EL,P,HP,DP,HP,DP,EL,W
C  ISN 0072
C  READ (IXI,902) WS,DS,HZ,DT,RZ,THZ,THZ,THZ
C  ISN 0073
C  READ (IXI,902) WT,ENH,DLX,RTT,ENP,TPP,HR
C  ISN 0074
C  IF ( HR .NE. 0. ) READ (IXI,910) RDR
C  ISN 0075
C  80 CONTINUE
C  ISN 0076
C  READ DISC RIM INPUT I0P2 = 1 OR 2
C  ISN 0077
C  ISN 0078
C  IF ( I0P2 - 1 ) 110,100,90
C  ISN 0079
C  90 READ (IXI,902) SA , SB , SC , SD , SE , SF , SG , SH
C  ISN 0080
C  READ (IXI,902) XI , SJ , SK , SL , SH, REF, WI
C  ISN 0081
C  IF ( WI ) 95,110,95
C  ISN 0082
C  95 READ (IXI,902) T11, R2
C  ISN 0083
C  GO TO 110
C  ISN 0084
C  100 READ (IXI,902) D2, D3, D4, D5, D6, DLR, WI
C  ISN 0085
C  READ (IXI,902) T1, T2, T3, T4, RA, BETD
C  ISN 0086
C  IF ( WI ) 105,110,105
C  ISN 0087
C  105 READ (IXI,902) T11, R2
C  ISN 0088
C  110 CONTINUE
C  ISN 0089
C  ISN 0090
C  READ SPRING AND RING PROPERTY INFORMATION
C  ISN 0091
C  I = 1
C  ISN 0092
C  120 READ (IXI,911) JXH(I) ,(VALII,L),L=1,12
C  ISN 0093
C  IF ( JXH(I) .EQ. 0 ) GO TO 130
C  ISN 0094
C  I = I + 1
C  ISN 0095
C  GO TO 120
C  ISN 0096
C  130 I = I + 1
C  ISN 0097
C  III = I - 2
C  ISN 0098
C  RETURN
C  ISN 0099
C  900 FORMAT(72X,II)
C  ISN 0100
C  901 FORMAT(16A4,1X,3I1)
C  ISN 0101
C  902 FORMAT(6F8.0)
C  ISN 0102
C  903 FORMAT(7I2,2X,7F8.0,1X,II)
C  ISN 0103
C  904 FORMAT(9F8.0,1X,F4.0)
C  ISN 0104
C  905 FORMAT(9F8.0)
C  ISN 0105
C  906 FORMAT(5E8.0)
C  ISN 0106
C  907 FORMAT(9F8.0,1X,II)
C  ISN 0107
C  908 FORMAT(4I2,6X,6F8.0)
C  ISN 0108
C  909 FORMAT(6F8.0)
C  ISN 0109
C  910 FORMAT(F8.2)
C  ISN 0110
C  911 FORMAT(12,6,1F3.0,F8.0))
C  ISN 0111
C  END
C
*OPTIONS IN EFFECT=MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT=SOURCE EEDIC NOLIST NODECK OBJECT HMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 110, PROGRAM SIZE = 3472, SUBPROGRAM NAME =RD137
*STATISTICS* NO DIAGNOSTICS GENERATED
******** END OF COMPILATION ********

2986K BYTES OF CORE NOT USED
DATA U498SRSFUN AT LEVEL 003 AS OF 01/28/82
DATA U498SRSFUN AT LEVEL 002 AS OF 06/22/81
DATA U500SRSFUN AT LEVEL 001 AS OF 06/22/81

C SAVE THE LARGER VALUE OF RFMAX AND RFMIN FOR EACH
C ORDER AND EACH ROOT
C
ISN 0009 NROOT = ROOT + .05
ISN 0010 WRITE (161,910)
ISN 0011 910 FORMAT(/,5X,'FREQUENCY',6X,'SPEED',10X,'BETA')
ISN 0012 DO 3 I = 1,NROOT
ISN 0013 WRITE (161,900) FNRQ(I), SPEED(I), BTA(I)
ISN 0014 3 CONTINUE
ISN 0015 900 FORMAT(5(3X,E12.5))
ISN 0016 WRITE (161,920)
ISN 0017 920 FORMAT(1,6X,'RPM-HIH',8X,'RPtlX',9X,'NU/I(l)',
1 10X,'ND(2)',10X,'ND(3)',10X,'ND(4)')
ISN 0018 WRITE (161,900) RPIHIN, RPtlX, EGRD(I), EGRD(2), EGRD(3)
ISN 0019 WRITE (161,930)
ISN 0020 930 FORMAT(/,7X,'DMAX',11X,'DMIN',11X,'RFMAX',10X,'RFMIN',11X,'RF')
ISN 0021 DO 10 I = 1,NROOT
ISN 0022 DO 5 J = 1,NORD
ISN 0023 DMAX = FNRQ(I)**2 + BTA(I) * (RFMAX**2 - SPEED(I)**2)
ISN 0024 DMIN = FNRQ(I)**2 + BTA(I) * (RFMIN**2 - SPEED(I)**2)
ISN 0025 RFMAX = 60. * SQRT(DMAX) / EORD(J) / RPtlX - 1.0
ISN 0026 RFMIN = 1.0 - 60. * SQRT(DMIN) / EORD(J) / RPtlX
ISN 0027 RF(I,J) = RFMAX
ISN 0028 IF (RFMIN .GT. RFMAX) RF(I,J) = RFMIN
ISN 0029 WRITE (161,900) DMAX, DMIN, RFMAX, RFMIN, RF(I,J)
ISN 0030 10 CONTINUE
ISN 0031 5 CONTINUE
ISN 0032 10 CONTINUE
ISN 0033 RETURN
ISN 0034 END

*OPTIONS IN EFFECT*NAME=MAIN OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*NAME=MAIN OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*STATISTICS* SOURCE STATEMENTS = 33, PROGRAM SIZE = 1048, SUBPROGRAM NAME =RSFUCN
*STATISTICS* NO DIAGNOSTICS GENERATED
* * * * * END OF COMPILATION * * * * *

3012K BYTES OF CORE NOT USED
C DATA SET U498STMAX AT LEVEL 003 AS OF 02/01/82
C DATA SET U498STMAX AT LEVEL 002 AS OF 07/16/81
C DATA SET U500STMAX AT LEVEL 001 AS OF 04/06/81

ISN 0002 SUBROUTINE TMAX (ICALC) 00001
ISN 0003 COMMON /UOS/ I51, I61 00002
ISN 0004 COMMON /ANAL04/ HSTA, HST1, HST2, NOACR, ILT, LONG, ISTE, 00003
I DEMA, EB, PGOB, BR, SERIES, BRS, BT5, IOPP 00004
ISN 0005 COMMON /ANAL05/ XP(3,53,21), NP(21) 00005
ISN 0006 COMMON /ANAL17/ XPS(3,53,21), TXI(21), TCC(21), BMX(21) 00006

1, ECC(21) 00007
ISN 0007 NBRS = NRS + .05 00008
ISN 0008 IF (ICALC .EQ. 3) GO TO 70 00009
ISN 0010 IF (ICALC .EQ. 1 OR ICALC .EQ. 2) GO TO 10 00010
ISN 0012 WRITE (161,900) ICALC 00011
ISN 0013 STOP 00012
ISN 0014 10 IF (ICALC .EQ. 2) GO TO 70 00013

C DETERMINE MAX THICKNESS OF EACH INPUT STATION AND SAVE COORDINATES00014
C
ISN 0016 DO 30 I = NBRS, HSTA 00015
ISN 0017 NPP = NP(I) 00016
ISN 0018 TMAX = XP(2,1,I) - XP(3,1,I) 00017
ISN 0019 BMX(I) = XP(1,NPP,I) - XP(1,1,I) 00018
ISN 0020 DO 20 J = 1, NPP 00019
ISN 0021 T = XP(2,J,I) - XP(3,J,I) 00020
ISN 0022 TMAX = AMAX1 (TMAX,T) 00021
ISN 0023 20 CONTINUE 00022
ISN 0024 TMX(I) = TMAX 00023
ISN 0025 30 CONTINUE 00024
ISN 0026 DO 60 I = NBRS, HSTA 00025
ISN 0027 NPP = NP(I) 00026
ISN 0028 DO 50 J = 1, NPP 00027
ISN 0029 DO 40 K = 1, 3 00028
ISN 0030 XPS(K,J,I) = XP(K,J,I) 00029
ISN 0031 40 CONTINUE 00030
ISN 0032 50 CONTINUE 00031
ISN 0033 60 CONTINUE 00032
ISN 0034 GO TO 999 00033
ISN 0035 70 CONTINUE 00034

C RATIO THE Y-UPPERS AND Y-LOIERS TO REFLECT THE NEW 00035
C THICKNESS FROM COPES-COMIN 00036
C
ISN 0036 DO 90 I = NBRS, HSTA 00037
ISN 0037 NPP = NP(I) 00038
ISN 0038 PER = TCCI(I) / THXI(I) 00039
ISN 0039 DO 80 J = 1, NPP 00040
ISN 0040 T = XPS(2,J,I) - XPS(3,J,I) 00041
ISN 0041 XS = T/THXI(I) 00042
ISN 0042 XPS(2,J,I) = XPS(2,J,I) * (PER - 1.) * .5 * T * XS 00043
ISN 0043 XPS(3,J,I) = XPS(3,J,I) - (PER - 1.) * .5 * T * XS 00044
ISN 0044 80 CONTINUE 00045
ISN 0045 90 CONTINUE 00046

C RATIO THE Y-UPPERS AND Y-LOIERS TO REFLECT THE NEW 00047
C
ISN 0048
C CHORD FROM COPES-CONSIN
C

ISH 0046 DO 110 I = NBRSR,NSTA
ISH 0047 NPP = NP(I)
ISH 0048 IF (BCC(I) .EQ. 0.0 .OR. BMX(I) .EQ. 0.0) GO TO 110
ISH 0050 PER = BCC(I) / BMX(I)
ISH 0051 DO 100 J = 1 , NPP
ISH 0052 XP(1,J,I) = XP(1,J,I) + (PER - 1.0) * XP(1,J,I)
ISH 0053 XP(2,J,I) = XP(2,J,I) + (PER - 1.0) * XP(2,J,I)
ISH 0054 XP(3,J,I) = XP(3,J,I) + (PER - 1.0) * XP(3,J,I)
ISH 0055 100 CONTINUE
ISH 0056 110 CONTINUE
ISH 0057 900 FORMAT (' ICALC = ',I10)
ISH 0058 999 RETURN
ISH 0059 END

*OPTIONS IN EFFECT* MAIN OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOOML(NONE)
*OPTIONS IN EFFECT* SOURCE EBCDIC NOEBCDIC NODECK OBJECT NOOBJECT NOFORMAT NOSTMT 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, NOLAP, NOXREF, NOLIST, NODECK, OPT(3), AUTOCL(NONE), NOALC
OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOCL(NONE)
SOURCE EBCDIC NOLIST NODECK OBJECT NOLAP NOFORMAT GOSTHT NOXREF NOALC NOANSF TERM IBM FLAG(I)
&VERSION 1.3.0 (01 MAY 80)  WTT983  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 08.141/10.53.00  PAGE 2

ISH 0032
  DO 10 I = 1,MAERO
  00055
ISH 0033
  WRITE (IYI,902) RAERO(I), AMACH(I), VEL1(I), TEMP1(I)
  00056
  1, PRES(I), PSI2(I), BETA1(I)
  00057
ISH 0034
  10 CONTINUE
  00058
C
C  WRITE W137 CONTROL CARD -- CARD 7
C
ISH 0035
  DIA = IND
  00062
ISH 0036
  ANBLD = BLADES * BMX(BER) / BCC(BER)
  00063
ISH 0037
  WRITE (IYI,903) SPEED(MODE), DIA, IRNO(MODE), FRF(MODE),
  00064
  1 TRM(MODE), ANBLD, NSFA
  00065
C
C  WRITE (26,903) SPEED(MODE), DIA, IRNO(MODE), FRF(MODE),
C
ISH 0038
  1 TRM(MODE), ANBLD, NSFA
  00066
C
C  WRITE W137 MODE SHAPE -- CARD 8
C
ISH 0039
  DO 20 J = 1,NSFA
  00071
ISH 0040
  I = ITHN(J)
  00072
ISH 0041
  WRITE (IYI,904) RAD(MODE,I), CHD(MODE,I), SCT(MODE,I),
  00073
  1 ALPH(MODE,I), PFDF(MODE,I), PFAG(MODE,I), PHDF(MODE,I),
  00074
  2 PHAG(MODE,I)
  00075
ISH 0042
  20 CONTINUE
  00076
ISH 0043
  IF (IYI.EQ.5) REWIND 5
  00077
ISH 0044
  RETURN
  00078
ISH 0045
  900 FORMAT (20A4)
  00079
ISH 0046
  901 FORMAT (5I5,13.12,6I5)
  00080
ISH 0047
  902 FORMAT (2F10.5,3F10.3,2F10.4)
  00081
ISH 0048
  903 FORMAT (2F10.2,1I10,1F10.4,2E10.4,1I10)
  00082
ISH 0049
  904 FORMAT (4F10.5,E10.4,F10.4,E10.4,F10.4)
  00083
ISH 0050
  END

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

300CK BYTES OF CORE NOT USED
NEW CASE CONTROL CARD
101137 CONTROL CARD

OPTIONS EFFECT:
NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOBBL(NONE)
SOURCE ECCDIC NOLIST NNODECK NOHAP NOFORMAT GOSTTH NOXREF NOALC NOANSF TERM IBM FLAG(I)

DATA SET U98SHT137 AT LEVEL 005 AS OF 03/11/82
DATA SET U98SHT137 AT LEVEL 004 AS OF 01/28/82
DATA SET U98SHT137 AT LEVEL 003 AS OF 12/02/81
DATA SET U98SHT137 AT LEVEL 002 AS OF 05/28/81
DATA SET U98SHT137 AT LEVEL 001 AS OF 04/06/81

SUBROUTINE WTH137 (ICALC)
COMMON /UOS/ ISE, ISE
COMMON /ANAL0/ HTEST
COMMON /ANAL02/ ITITLE(16), IOP1, IOP2, IOP3
COMMON /ANAL03/ RPM, FN, DFN, TOLL, ROOT, DRPM, RPMN, RPMX
COMMON /ANAL04/ NSTA, HST1, HST2, NOACR, ILT, LNS, ISTE
REQUESTED VERSION 1.3.0 (01 MAY 80) SYSTEM/370 FORTRAN

1ST! 0026
ISH 0034
ISN 0030
ISN 0028
ISN 0027
ISN 0026
ISN 0025
ISN 0024
ISN 0023
ISN 0022
ISN 0021
ISN 0020
ISN 0019
ISN 0018
ISN 0017
ISN 0016
ISN 0015
ISN 0014
ISN 0013
ISN 0012
ISN 0011
ISN 0010
ISN 0009
ISN 0008
ISN 0007
ISN 0006
ISN 0005
ISN 0004
ISN 0003
ISN 0002
ISN 0001

WRITE FIRST FOUR CARDS OF THE H137 INPUT
WRITE AIRFOIL DATA

WRITE NEW CASE CONTROL CARD
WRITE TITLE CARD AND RIM PROPERTIES OPTIONS
WRITE H137 CONTROL CARD

IF (ICALC.EQ. 1) IYI = 24
IF (ICALC.EQ. 2) IYI = 4
IF (ICALC.EQ. 4) IYI = 7
IF (IYI.EQ. 5 OR. IYI.EQ. 24) REWIND IYI
WRITE (IYI900) NTEST
WRITE (IYI901) ITITLE, IOP1, IOP2, IOP3
WRITE (IYI902) RPM, FN, DFN, TOLL, ROOT, DRPM, RPMN, RPMX
WRITE (IYI903) NSTA, HST1, HST2, NOACR, ILT, LNS, ISTE

DO 20 I = 1, NSTA

WRITE FIRST FOUR CARDS OF THE XI37 INPUT
DATA SET U98SHT137 AT LEVEL 005 AS OF 03/11/82
DATA SET U98SHT137 AT LEVEL 004 AS OF 01/28/82
DATA SET U98SHT137 AT LEVEL 003 AS OF 12/02/81
DATA SET U98SHT137 AT LEVEL 002 AS OF 05/28/81
DATA SET U98SHT137 AT LEVEL 001 AS OF 04/06/81

SUBROUTINE WTH137 (ICALC)
COMMON /UOS/ ISE, ISE
COMMON /ANAL0/ HTEST
COMMON /ANAL02/ ITITLE(16), IOP1, IOP2, IOP3
COMMON /ANAL03/ RPM, FN, DFN, TOLL, ROOT, DRPM, RPMN, RPMX
COMMON /ANAL04/ NSTA, HST1, HST2, NOACR, ILT, LNS, ISTE
REQUESTED VERSION 1.3.0 (01 MAY 80) SYSTEM/370 FORTRAN

1ST! 0026
ISH 0034
ISN 0033
ISN 0032
ISN 0031
ISN 0030
ISN 0029
ISN 0028
ISN 0027
ISN 0026
ISN 0025
ISN 0024
ISN 0023
ISN 0022
ISN 0021
ISN 0020
ISN 0019
ISN 0018
ISN 0017
ISN 0016
ISN 0015
ISN 0014
ISN 0013
ISN 0012
ISN 0011
ISN 0010
ISN 0009
ISN 0008
ISN 0007
ISN 0006
ISN 0005
ISN 0004
ISN 0003
ISN 0002
ISN 0001
WRITE (IYI,904) R(I), DCOR(I), TOG(I), THSTH(I), ALPHA(I), 00056
1 S0B(I), RLE(I), RTE(I), ADWT(I), 0(I)

WRITE TILT AND SPRING CARD IF:
1 ---- TILT IS INPUT AT EACH STATION (ILT = 1)
2 ---- TILT IS INPUT AT THE TIP STATION (ILT = 2) AND THE TIP STATION IS BEING READ (I = NO. OF STATIONS)
3 ---- ADDED WEIGHT HAS BEEN INPUT

IF (ILT.EQ.1) GO TO 5
IF (ILT.EQ.1 .AND. I.EQ.1) GO TO 5
IF (ADWT(I)) 5,6,5
WRITE (IYI,905) TLTA(I), TLT(I), XADWT(I), YADWT(I), AKX(I), AKY(I), AKZ(I)

IF T/B IS POSITIVE --- STOP READING AIRFOIL DATA
IF T/B IS NEGATIVE --- READ COORDINATES

6 IF (TOB(I).GT.0.0) GO TO 20
N = O(I) + .05
DO 10 J = 1, 3
WRITE (IYI,906) (XP(J,IH,I), IN = 1, H)
10 CONTINUE
CONTINUE

WRITE ROOT AND TIP ANGLE, M/O, AND BROACH ANGLE
WRITE (IYI,907) THET, THER, AKMIN, AKMAX, ERANG
WRITE SHROUD DATA (HST1 = 0, NO SHROUDS)

IF (MST1.LT.1) GO TO 30
WRITE (IYI,908) HTS1, XSHRI, YSHRI, ARSH1, XISH1, XISH3, AKSH1, SHPOI1, ANGSI, NSL
IF (MST2.LT.2) GO TO 30
WRITE (IYI,909) HTS2, XSHRI, YSHRI, ARSH2, XISH2, XISH4, AKSH1, SHPOI2, ANGSI
30 CONTINUE

WRITE DISC CONTROL DATA
BLADE = BLADES
IF (ICALC.EQ.1) GO TO 35
NSR = BRS + .05
BLADE = BLADES * BHX(NER) / BCC(NER)
WRITE (IYI,909) MAY, NAX, ISTR, NBB, DEND, DENA, ED, DIAHD, BLADE, POID
WRITE DISC RADIUS AND THICKNESSES IF MAY .NE. 0
IF (MAY.EQ.0) GO TO 40
WRITE (IYI,910) (RAD(I), THE(I), I = 1, MAY)
40 CONTINUE
WRITE BORE STRESS, STRESS AT EACH STATION, OR NO STRESS
IF (ISTR) 70, 60
50 WRITE (IYI,911) SIGBOR
*VERSION 1.3.0 (01 MAY 80)  WTH137  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 02.141/10.53.04  PAGE 3

I3N 0069  GO TO 70  001114
I3N 0070  60 J = MAX + 1  001115
I3N 0071  WRITE (I3I,911) (RADST(I), TAHST(I), I = J, MAY)  001116
I3N 0072  70 CONTINUE  001117

C WRITE DOVETAIL INPUT OR SKIP TO RING INPUT

I3N 0073  IF (IOP1 .EQ. 0) GO TO 80  001119
I3N 0075  WRITE (I3I,910) ELP, WP, WBP, DXP, WR  001120
I3N 0076  WRITE (I3I,910) MT, EMU, DXL, RTP, DNP, THP, WR  001122
I3N 0077  IF (HR .NE. 0.) WRITE (I3I,910) RDR  001124
I3N 0080  60 CONTINUE  001126

C WRITE DISC RIM INPUT IF IOP2 = 1 OR 2

I3N 0081  IF (IOP2 - 1) 110,100,90  001127
I3N 0082  90 WRITE (I3I,910) SA, S8, SC, SD, SE, SF, SG, SH  001130
I3N 0083  WRITE (I3I,910) SI, SJ, SK, SL, SH, REF, WI  001132
I3N 0084  IF (HI) 95,110, 95  001133
I3N 0085  WRITE (I3I,910) T1, R2  001134
I3N 0086  GO TO 110  001135
I3N 0087  100 WRITE (I3I,910) D2, D3, D4, D5, D6, DLR, WI  001136
I3N 0088  WRITE (I3I,910) T1, T2, T3, T4, RA, BEND  001137
I3N 0089  IF (HI) 105,110,105  001138
I3N 0090  105 WRITE (I3I,910) T1, T2  001139
I3N 0091  110 CONTINUE  001140

C WRITE SPRING AND RING PROPERTY INFORMATION

I3N 0092  DO 120 I = 1, III  001141
I3N 0093  WRITE (I3I,912) JXN(I), (VALCI,L), L = 1, I2  001142
I3N 0094  120 CONTINUE  001143
I3N 0095  IXN = 0  001144
I3N 0096  WRITE (I3I,913) IXN  001145
I3N 0097  WRITE (I3I,913) IXN  001146
I3N 0098  IF (I3I .EQ. 5) REWIND I3I  001147
I3N 0100  RETURN  001148

I3N 0101  900 FORMAT (72X,I1)  001149
I3N 0102  901 FORMAT (18A4,1X,3I1)  001150
I3N 0103  902 FORMAT (3F8.1,F0.5,4F8.1)  001151
I3N 0104  903 FORMAT (72X,F8.5,E8.5,3F8.5,3F8.1,1X,I1)  001152
I3N 0105  904 FORMAT (7F8.5,2F8.4,4F8.5,1X,F4.0)  001153
I3N 0106  905 FORMAT (3E8.5,4F8.5,3E6.1)  001154
I3N 0107  906 FORMAT (9F8.5)  001155
I3N 0108  907 FORMAT (2F8.4,2F8.0,F8.4)  001156
I3N 0109  908 FORMAT (3F8.5,F0.8.6,4F8.7,F0.4.1X,I1)  001157
I3N 0110  909 FORMAT (4F8.5,F8.4,F8.4,2F8.5,F8.3,3F8.4)  001158
I3N 0111  910 FORMAT (8F8.5)  001159
I3N 0112  911 FORMAT (6F8.1)  001160
I3N 0113  912 FORMAT (I2,F3.0,E8.3,5F3.0,F0.8.5))  001161
I3N 0114  913 FORMAT (7X,I1)  001162
I3N 0115  914 FORMAT (7X,I1)  001163
I3N 0116  915 END  001164

*OPTIONS IN EFFECT=ENAME(MAIN) OPTIMIZE(3) LINEDOCT(60) SIZE(MAX) AUTOBLINQUE
*OPTIONS IN EFFECT=SOURCE EBCDIC NOLIST NOOPT NOOBJ NOFORMAT GOSTHM NOXREF NOSFCT NOIND TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 114, PROGRAM SIZE = 3002, SUBPROGRAM NAME =WTH137
*STATISTICS* NO DIAGNOSTICS GENERATED
****** END OF COMPILATION ******

2984K BYTES OF CORE NOT USED
SUBROUTINE OBJTV

CALCULATE THE OBJECTIVE FUNCTION FOR A HOLLOW FOIL WITH BORSIC TITANIUM INLAY OR A SUPERHYBRID FOIL

COMMON /GLOBCM/ OBJF, FN(5), DLAR(5), THKVAL(21), RF(5,4)

1 , ERCC , FODLSB(21), DLE , DTE , DROOT , DTIP , TTI , TLT

2 , OBJFUN

COMMON /PLY1/ PLY(21,25,7) , THETA(7) , RHO(7)

ISN 0003

COMMON /ANAL33/ DLED, DTED, DROOT, DTIPD, TTD, TLT, NCD

COMMON /ANAL17/ XPS(3,5,3,21), TIC(21), TCC(21), BHX(21)

1 , ERCC(21)

ISN 0004

COMMON /VOLS/ ISI , I6I

ISN 0005

COMMON /ANAL04/ HSTA , MSTA , MSTR , NOACR , ILT , LGHT , ISTE , 1 , DENS , EB , POID , BR , SERIES , BRS , BTS , IOPP

ISN 0006

COMMON /ANAL05/ R(21), BOBR(21), TOS(21), TISTH(21), ALPHA(21),

1 , SODB(21), REL(21), RTE(21), ADIFF(21), O(21)

ISN 0007

DOUBLE PRECISION PLY , Theta , RHO , WL , DLED , DTED , DROOT

DOUBLE PRECISION PLY , TTHETA , RHO , WLI

NCO = 1 (HOLLOW FOIL)

NCO = 2 (SUPERHYBRID FOIL)

ISN 0010

IF (BRCC .EQ. 0.0) RC = BMX(1)

RT = THKVAL(1)

ISN 0011

TIVOL = (HL(1) + HL(7)) / RHO(1)

BTVOl = (HL(2) + HL(6)) / RHO(2)

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

EMC = 2000. * (.00958 * TIVOL + 1.183 * (BTVOl/116.6)**1.08 )


EDC = 260.42 * (EMC + ELC) / RC + 3700. * RC

ISN 0021

WRTE (16I,500) RC , RT , TIVOL , BTVOl , EMC , ELC , EDC

ISN 0022

900 FORMAT (7(3X,E12.5/))

ISN 0024

EMC = MATERIAL COST

ELC = LABOR COST

EDC = DELTA COST RELATIVE TO E3

ISN 0025

TR = R(NSTA)

ISN 0026

NBSR = BRS + .05
RR = R(NGRS) 00055
ARATIO = (TR**2 - (DROOT - 2.)**2) / (TR**2 - RR**2) 00056
IF ((DROOT - 2.) .LT. RR) ARATIO = 1.0 00057
EDLE = DLE 00058
IF ( (DLE -.001) .LT. 0. ) EDLE = 20.0 00059
EDMHC = (EMC + ELC) * (4.0 + 6.8 * ARATIO/EDLE) * 220.0 00060
1 / 250000.0 / RC 00061
TR = TIP RADIUS 00062
RR = ROOT RADIUS 00063
EDIMC = DELTA MAINTENCE AND MATERIAL COST 00064
HI = WU1 + WU2 + WU4 + WU7 00065
OIIT = 99.5 * FW - 1.233 * FW**2 00066
OBJFUN = .54 * EDC/I00000. + .80 * EDMHC/I0. + .52 * DWT/1000. 00067
FW = FOIL WEIGHT 00068
DWT = DELTA WEIGHT RELATIVE TO E3 00069
OBJFUN = OBJECTIVE FUNCTION 00070
TIVOL = VOLUME OF TITANIUM 00071
BAVOL = VOLUME OF BORON ALUMINIUM 00072
GEVOL = VOLUME OF GRAPHITE EPOXY 00073
TIVOL = (WL(1) + WL(2) + WL(5) + WL(6) ) / 3 00074
BAVOL = (WL(2) + WL(6) ) / RHO(2) 00075
GEVOL = (WL(3) + WL(5) ) / RHO(3) 00076
EMC = MATERIAL COST 00077
ELC = LABOR COST 00078
EDC = DELTA COST RELATIVE TO E3 00079
EDMHC = DELTA MAINTENCE AND MATERIAL COST 00080
EMC = 2000.0 * (.483 * (GEVOL / 116.6)**1.05 00081
1 + .637 * ( (BAVOL + TIVOL ) / 116.6 )**1.08 00082
2 + .1445 * ( (RC / 9.25)**1.774) 00083
ELC = 2000.0 * (.063 * (RC / 9.25)**1.15 00084
1 + .121 * (RT / .925)**.82) 00085
EDC = 246.42 * (EMC + ELC) / RC + 3700.0 * RC 00086
EDMHC = 222.0 * (EMC + ELC) / (12500.0 * RC) 00087
FW = FOIL WEIGHT 00088
DWT = DELTA WEIGHT RELATIVE TO E3 00089
OBJFUN = OBJECTIVE FUNCTION 00090
FW = (WL(1) +WL(2) +WL(3) +WL(4) +WL(5) +WL(6) +WL(7)) / 3 00091
DWT = 99.5 * FW - 1.233 * FW**2 00092
OBJFUN = .54 * EDC/I00000. + .60 * EDMHC/I0. + .52 * DWT/1000. 00093
WRITE (I61,900) (RHO(i),1=1.71 00094
WRITE (I61,900) RHIT, WC, ELC, EDMHC, FW, DWT, OBJFUN 00095
50 CONTINUE
*OPTIONS IN EFFECT* NAME (MAIN) OPTIMIZE (3) LINECOUNT (60) SIZE (MAX) AUTOCBL (NONE)

*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NOXREF NOFORMAT GOSTIT NODECK NOALC NODATE TERM IEM FLAG (I)

*STATISTICS* SOURCE STATEMENTS = 59, PROGRAM SIZE = 1994, SUBPROGRAM NAME = OBJTIV

*STATISTICS* NO DIAGNOSTIC GENERATED

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

3000K BYTES OF CORE NOT USED
DATA SET U77CHNNO1 AT LEVEL 001 AS OF 02/13/81
DATA SET 9156CHNNO1 AT LEVEL 001 AS OF 07/10/80
SUBROUTINE CUNNO1 (JGOTO,X,DF,G,ISC,IC,A,G1,VLB,VUB,SCAL,C,NCAL,DX00002
1,DXI,FI,XI,HI,NI,NI,HI,NI,NI,NI)
COMMON /CHNNO1/ DELFUN,DABFUN,FDCH,FDCHM,CT,CMTH,CTL,CMTHM,ALPHAX00003
1,ABFUN,THTA,OBJ,NDV,NCON,HSIDE,IFINT,NFDG,NSCAL,LINOBJ,ITHAX,IT00004
2RHI,ICHDIR,JGOTO,NAC,INFO,INFOG,ITER
DIMENSION X(N1), DF(N1), G(N2), ISC(N2), IC(N3), A(N1,N3), G1(N2),00005
1 VLB(N1), VUB(N1), SCAL(N1), NCAL(2), CN(4)
ROUTINE TO CALCULATE GRADIENT INFORMATION BY FINITE DIFFERENCE. 00009
C BY G. H. VANDERPLAATS  JUNE, 1972. 00010
C NASA-AERES RESEARCH CENTER,  NDFFETT FIELD,  CALIF. 00011
IF (JGOTO.EQ.1) GO TO 10 00012
IF (JGOTO.EQ.2) GO TO 70 00013
IF (NC(II).EQ.O.AND.G(II).LT.CT) GO TO 20 00014
NAC=NAC+1 00015
IF (NCAC.GE.N3) RETURN 00016
IC(NACI)=I 00017
CONTINUE 00018
IF (JGOTO.EQ.2.AND.NAC.EQ.O) RETURN 00019
IF (INFO.GE.10) RETURN 00020
IF (INFO.GE.10) RETURN 00021
IF (INFO.GE.10) RETURN 00022
IF (INFO.GE.10) RETURN 00023
IF (INFO.GE.10) RETURN 00024
DO 10 I=1,NCON 00025
Gl(I)=G(I) 00026
CONTINUE 00027
JGOTO=0 00028
IF (INFO.GE.10) RETURN 00029
IF (INFO.GE.10) RETURN 00030
IF (INFO.GE.10) RETURN 00031
IF (INFO.GE.10) RETURN 00032
IF (INFO.GE.10) RETURN 00033
IF (INFO.GE.10) RETURN 00034
IF (INFO.GE.10) RETURN 00035
IF (INFO.GE.10) RETURN 00036
DO 20 I=1,NCON 00037
CONTINUE 00038
G1(I)=G(I) 00039
CONTINUE 00040
JGOTO=0 00041
IF (INFO.GE.10) RETURN 00042
IF (INFO.GE.10) RETURN 00043
IF (INFO.GE.10) RETURN 00044
IF (INFO.GE.10) RETURN 00045
IF (INFO.GE.10) RETURN 00046
CONTINUE 00047
STORE VALUES OF CONSTRAINTS IN G1 00048
CONTINUE 00049
DO 30 I=1,NCON 00050
CONTINUE 00051
G1(I)=G(I) 00052
CONTINUE 00053
JGOTO=0 00054
IF (INFO.GE.10) RETURN 00055
IF (INFO.GE.10) RETURN 00056
CONTINUE 00057
CONTINUE 00058
CONTINUE 00059
FUNCTION EVALUATION

------------------------------------------------------------------
C DETERMINE GRADIENT COMPONENTS OF ACTIVE CONSTRAINTS

------------------------------------------------------------------
C STORE CURRENT CONSTRAINT VALUES BACK IN G-VECTOR

------------------------------------------------------------------
END OF COMPILATION

300K BYTES OF CORE NOT USED
DATA SET U477CN002 AT LEVEL 001 AS OF 02/13/81
DATA SET 9167CN002 AT LEVEL 001 AS OF 07/10/80

SUBROUTINE CNMN02 (NCALC, SLOPE, DFTDF1, DF, S, NI)

COMMON /CHNL/ DELFUN, D2FUN, FDC, FDCH, FDCHM, CT, CTMIN, ALPHAX, IC
1, ABOBJ, JTHETA, DS, NCH, NSIDE, FPRINT, NFDG, NHCAL, LINOBJ, ITHAX, ITG0004
2RM, IMCHDIR, IGO, IAC, INFOG, ITER

DIMENSION DF(N1), S(N1)

ROUTINE TO DETERMINE CONJUGATE DIRECTION VECTOR OR DIRECTION
OF STEEPEST DESCENT FOR UNCONSTRAINED FUNCTION MINIMIZATION.
BY G. N. VANDERPLAATS
APRIL, 1972.
NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.
NCALC = CALCULATION CONTROL.
NCALC = 0, S = STEEPEST DESCENT.
NCALC = 1, S = CONJUGATE DIRECTION.
CONJUGATE DIRECTION IS FOUND BY FLETCHER-REEVES ALGORITHM.

CALCULATE NORM OF GRADIENT VECTOR

DFTDF=0.
DO 10 I=1,NDV
DFI=DF(I)
DFTDF=DFTDF+DFI*DFI
CONTINUE
NCALC=0

CALCULATE DIRECTION OF STEEPEST DESCENT
DO 40 I=1,NDV
S(I)=-DF(I)
SLOPE=-DFTDF
CONTINUE

NORMALIZE S TO MAX ABS VALUE OF UNITY
S1=0.
DO 60 I=1,NDV
S2=ABS(S(I))
IF (S2.GT.S1) S1=S2
CONTINUE

END
0033 IF (Sl.LT.1.0E-20) S1=1.0E-20
0035 S1=1./S1
0036 DFTDF1=DFTDF*S1
0037 DO 70 I=1,NV
0038 S(I)=S1*S(I)
0039 SLOPE=S1*SLOPE
0040 RETURN
0041 END

*OPTIONS IN EFFECT*MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOLIST NOFORMAT 60STMT NOXREF NOALC NOANSF TERM IBM FLAG(1)
*STATISTICS* SOURCE STATEMENTS = 40, PROGRAM SIZE = 654, SUBPROGRAM NAME = CNMN02
*STATISTICS* NO DIAGNOSTICS GENERATED

***** END OF-compilation *****

3012K BYTES OF CORE NOT USED
SUBROUTINE CN003 (X,S,SLOPE,ALP,FFF,A1,A2,A3,A4,F1,F2,F3,F4,APP,J00003
11,NCAL,KOUNT,JGOTO) 00004

COMMON /CHM1/ DELFUN,DASFUN,FDCH,FDCHM,CT,CTMN,CTL,CTLMIN,ALPHA00005
1,ABDUX,THETA,OBJ,NDF,NCON,INSIDE,ITPRINT,NFDG,NCAL,LINEJ,ITMAX,IT00006
2N,ISCO,DIR,IGOTO,NAC,INFO,INFOG,ITER 00007

COMMON /UIOS/ ISI, I6I 00008
DIMENSION X(N1), S(N1), NCAl(2) 00009

ROUTINE TO SOLVE ONE-DIMENSIONAL SEARCH IN UNCONSTRAINED 00010
MINIMIZATION USING 2-POINT QUADRATIC INTERPOLATION, 3-POINT 00011
CUBIC INTERPOLATION AND 4-POINT CUBIC INTERPOLATION. 00012
BY G. N. VANDERPLAATS APRIL, 1972. 00013
NASA-AMES RESEARCH CENTER, Moffett Field, Calif. 00014
ALP = PROPOSED MOVE PARAMETER. 00015
SLOPE = INITIAL FUNCTION SLOPE = S-TRANSPOSE TIMES DF. 00016
SLOPE MUST BE NEGATIVE. 00017
OBJ = INITIAL FUNCTION VALUE. 00018
ZRO=0. 00019
IF (SLOPE.LT.O.) GO TO 20 00020
A1P=0. 00021
RETURN 00022
CONTINUE 00023
IF (IPRINT.GT.4) WRITE (161,370) AP 00024
IF (IPRINT.GT.4) WRITE (161,380) (XCI)' 1=1,NOV) 00025
NCAL(1)=NCAL(1)+1 00026
JGOTO=1 00027
RETURN 00028
F2=OBJ 00029
IF (F2.LT.F1) GO TO 120 00030
10 IF (SLOPE.LT.0.) GO TO 20 00031
ALP=0. 00032
RETURN 00033
CONTINUE 00034
DO 40 I=1,NDV 00035
X(I)=X(II+AP*SII) 00036
IF (IPRINT.GT.4) WRITE (161,370) AP 00037
IF (IPRINT.GT.4) WRITE (161,380) (XCI)' 1=1,NOV) 00038
NCAL(1)=NCAL(1)+1 00039
JGOTO=1 00040
RETURN 00041
CONTINUE 00042
DO 40 I=1,NDV 00043
X(I)=X(I)+AP*H(I) 00044
IF (IPRINT.GT.4) WRITE (161,370) AP 00045
IF (IPRINT.GT.4) WRITE (161,380) (XCI)' 1=1,NOV) 00046
NCAL(1)=NCAL(1)+1 00047
JGOTO=1 00048
RETURN 00049
CONTINUE 00050
F2=OBJ 00051
IF (F2.LT.F1) GO TO 120 00052
CONTINUE 00053
F2=OBJ 00054
IF (F2.LT.F1) GO TO 120 00055
CHECK FOR ILL-CONDITIONING

IF (KOUNT.GT.5) GO TO 60

FF=2.*ABS(F1)

IF (F2.LT.FF) GO TO 90

FF=5.*ABS(F1)

IF (F2.LT.FF) GO TO 60

A2=.5*A2

A3=A2

A2=.5*A2

IF (IPRINT.GT.4) WRITE (161,390) F2

PROCEED TO CUBIC INTERPOLATION.

GO TO 160

JJ=1

II=1

CALL CNMNO4 (II,APP,ZRO,A1,F1,SLOPE,A2,F2,ZRO,ZRO,ZRO,ZRO)

IF (APP.LT.ZRO.OR.APP.GT.A2) GO TO 120

F3=F2

A3=A2

A2=APP

JJ=0

UPDATE DESIGN VECTOR AND FUNCTION VALUE

AP=A2-ALP

ALP=A2

DO 70 I=1,NDV

X(I)=X(I)+AP*S(I)

IF (IPRINT.GT.4) WRITE (161,370) A2

IF (IPRINT.GT.4) WRITE (161,380) (X(I),I=1,NDV)

NCAL(I)=NCAL(I)+1

JGOTO=2

RETURN

CONTINUE

F2=OBJ

IF (IPRINT.GT.4) WRITE (161,390) F2

PROCEED TO CUBIC INTERPOLATION.

GO TO 160

CONTINUE

2-POINT QUADRATIC INTERPOLATION

JJ=1

II=1

CALL CNMNO4 (II,APP,ZRO,A1,F1,SLOPE,A2,F2,ZRO,ZRO,ZRO,ZRO)

IF (APP.LT.ZRO.OR.APP.GT.A2) GO TO 120

F3=F2

A3=A2

A2=APP

JJ=0

UPDATE DESIGN VECTOR AND FUNCTION VALUE

AP=A2-ALP

ALP=A2

DO 100 I=1,NDV

X(I)=X(I)+AP*S(I)

IF (IPRINT.GT.4) WRITE (161,370) A2

IF (IPRINT.GT.4) WRITE (161,380) (X(I),I=1,NDV)

NCAL(I)=NCAL(I)+1

JGOTO=3

RETURN

CONTINUE

F2=OBJ

IF (IPRINT.GT.4) WRITE (161,390) F2
GO TO 150
120 A3=2.*A2
C UPDATE DESIGN VECTOR AND FUNCTION VALUE
C
AP=A3-ALP
ALP=A3
DO 130 I=1,NDV
130 XU(I)=XU(I)+AP*S(I)
IF (IPRINT.GT.4) WRITE (161,370) A3
IF (IPRINT.GT.4) WRITE (161,380) (XU(I),I=1,NDV)
NCAL(I)=NCAL(I)+1
JGOTO=4
RETURN
140 CONTINUE
F3=OBJ
IF (IPRINT.GT.4) WRITE (161,390) F3
150 CONTINUE
IF (F3.LT.F2) GO TO 190
160 CONTINUE
C 3-POINT CUBIC INTERPOLATION
C
II=3
CALL CHNNO4 (II,APP,ZRO,A1,F1,SLOPE,A2,F2,A3,F3,ZRO,ZRO)
IF (APP.LT.ZRO.OR.APP.GT.A3) GO TO 190
C UPDATE DESIGN VECTOR AND FUNCTION VALUE.
AP=APP
AP=AP-ALP
ALP=AP
DO 170 II=1,NDV
170 XIII=II+AP*S(II)
IF (IPRINT.GT.4) WRITE (161,370) ALP
IF (IPRINT.GT.4) WRITE (161,380) (X(I),I=1,NDV)
NCAL(I)=NCAL(I)+1
JGOTO=5
RETURN
180 CONTINUE
IF (UPRWT.GT.4) WRITE (161,390) OBJ
C CHECK CONVERGENCE
C
AA=I-APP/A2
AB2=ABS(F2)
AB3=ABS(OBJ)
AB=AB2
IF (AB.LT.1.0E-15) AB=1.0E-15
AB=(AB2-AB3)/AB
IF (ABS(AB).LT.1.0E-15.AND.ABS(AA).LT.1.0E-15) GO TO 330
A4=A3
F4=F3
A3=APP
IF (A3.GT.A2) GO TO 230
A3=A2
F3=F2
*VERSION 1.3.0 (01 MAY 80)*  
**SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)**  
DATE 02.141/10.53.20  
PAGE 4

**1. 3.0 (01 MAY 80)**  
**CUHL03**  
**SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)**  
DATE 02.141/10.53.20  
PAGE 4

**ISN 0150**  
A2=APP 00173

**ISN 0159**  
F2=OBJ 00174

**ISN 0160**  
GO TO 230 00175

**ISN 0161**  
190 CONTINUE 00176

**C**  
---------------------------------------------------------------

**C**  
**4-POINT CUBIC INTERPOLATION**  
---------------------------------------------------------------

**C**  
---------------------------------------------------------------

**ISN 0162**  
200 CONTINUE 00179

**ISN 0163**  
A4=2.*A3 00180

**C**  
UPDATE DESIGN VECTOR AND FUNCTION VALUE.  
00181

**ISN 0164**  
AP=A4-APL 00182

**ISN 0165**  
ALP=A4 00183

**ISN 0166**  
DO 210 I=1,NDV 00184

**ISN 0167**  
210 X(I)=X(I)+APL*SCI(I) 00185

**ISN 0168**  
IF (IPRINT.GT.4) WRITE (161,370) ALP 00186

**ISN 0169**  
IF (IPRINT.GT.4) WRITE (161,380) (X(I),I=1,NOV) 00187

**ISN 0170**  
NCAL(I)=NCAL(I)+1 00188

**ISN 0171**  
JGOTO=6 00189

**ISN 0172**  
RETURN 00190

**ISN 0173**  
220 CONTINUE 00191

**ISN 0174**  
F4=08J 00192

**ISN 0175**  
IF (IPRINT.GT.4) WRITE (161,390) F4 00193

**ISN 0176**  
IF (F4.GT.F3) GO TO 230 00194

**ISN 0177**  
A1=A2 00195

**ISN 0178**  
F1=F2 00196

**ISN 0179**  
A2=A3 00197

**ISN 0180**  
F2=F3 00198

**ISN 0181**  
A3=A4 00199

**ISN 0182**  
F3=F4 00200

**ISN 0183**  
GO TO 200 00201

**ISN 0184**  
GO TO 200 00202

**ISN 0185**  
230 CONTINUE 00203

**ISN 0186**  
II=4 00204

**ISN 0187**  
CALL CN204 (II,APP,A1,A1,F1,SLOPE,A2,F2,A3,F3,A4,F4) 00205

**ISN 0188**  
IF (APP.GT.AI) GO TO 250 00206

**ISN 0189**  
AP=AI-ALP 00207

**ISN 0190**  
ALP=AI 00208

**ISN 0191**  
OBJ=FI 00209

**ISN 0192**  
DO 240 I=1,NOV 00210

**ISN 0193**  
240 X(I)=X(I)+APL*SCI(I) 00211

**ISN 0194**  
GO TO 280 00212

**ISN 0195**  
250 CONTINUE 00213

**C**  
UPDATE DESIGN VECTOR AND FUNCTION VALUE  
00214

**C**  
---------------------------------------------------------------

**ISN 0190**  
AP=APP-ALP 00215

**ISN 0191**  
ALP=APP 00216

**ISN 0192**  
DO 260 I=1,NDV 00217

**ISN 0193**  
260 X(I)=X(I)+APL*SCI(I) 00218

**ISN 0194**  
IF (IPRINT.GT.4) WRITE (161,370) ALP 00219

**ISN 0195**  
IF (IPRINT.GT.4) WRITE (161,380) (X(I),I=1,NOV) 00220

**ISN 0196**  
NCAL(I)=NCAL(I)+1 00221

**ISN 0197**  
JGOTO=7 00222

**ISN 0198**  
RETURN 00223

**ISN 0200**  
RETURN 00224

**ISN 0201**  
270 CONTINUE 00225

**ISN 0202**  
IF (IPRINT.GT.4) WRITE (161,390) OBJ 00226

**ISN 0203**  
GO TO 280 00227

**C**  
CHECK FOR ILL-CONDITIONING  
00228

**C**  
---------------------------------------------------------------

**C**  
---------------------------------------------------------------
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 0225 OBJ=F3 00240
ISN 0227 AP=A3-ALP 00241
ISN 0228 ALP=A3 00242
ISN 0229 GO TO 310 00243
ISN 0230 FORMATS 00244
ISN 0231 OBJ=F2 00245
ISN 0232 ALP=A2 00246
ISN 0233 310 CONTINUE 00247
C UPDATE DESIGN VECTOR 00248
C DO 320 I=I,NDV 00251
C X(I)=X(I)+AP*S(I) 00252
C CONTINUE 00253
C CHECK FOR MULTIPLE MINIMA 00254
C IF (OBJ.LE.FFF) GO TO 350 00255
C INITIAL FUNCTION IS MINIMUM. 00256
C DO 340 I=I,NDV 00257
C X(I)=X(I)-ALP*S(I) 00258
C ALP=O. 00259
C OBJ=FFF 00260
C CONTINUE 00261
C JGOTO=O 00262
C RETURN 00263
C FORMATS 00264
C END 00265
C FORMAT (////5X,40H* * UNCONSTRAINED ONE-DIMENSIONAL SEARCH INFO*) 00266
C IRATION * * *) 00267
C ISN 0246 360 FORMAT (////5X,40H* * UNCONSTRAINED ONE-DIMENSIONAL SEARCH INFO) 00268
C ISN 0247 370 FORMAT (/5X,7HALPHA =,EI4.5/) 00269
C ISN 0248 380 FORMAT (5X,5HOBJ =,EI4.5/) 00270
C ISN 0249 390 FORMAT (/5X,5HOBJ =,EI4.5/) 00271
C ISN 0250 END 00272
*OPTIONS IN EFFECT=NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOFLB(NONE) 00273
*OPTIONS IN EFFECT=SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOASF TERM IBM FLAG(I) 00274
*STATISTICS* SOURCE STATEMENTS = 249, PROGRAM SIZE = 3520, SUBPROGRAM NAME =CN203 00275
*STATISTICS* NO DIAGNOSTICS GENERATED 00276
***** END OF COMPILATION ***** 00277

2972K BYTES OF CORE NOT USED
SUBROUTINE CNMN04 (II, XBAR, EPS, X1, Y1, SLOPE, X2, Y2, X3, Y3, X4, Y4)

ROUTINE TO FIND FIRST XBAR,GE.EPS CORRESPONDING TO A MINIMUM OF A ONE-DIMENSIONAL REAL FUNCTION BY POLYNOMIAL INTERPOLATION.

BY G. N. VANDERPLAATS APRIL, 1972.

NASA-AMES RESEARCH CENTER, Moffett Field, Calif.

II = CALCULATION CONTROL.
  1: 2-POINT QUADRATIC INTERPOLATION, GIVEN X1, Y1, SLOPE, X2 AND Y2.
  2: 3-POINT QUADRATIC INTERPOLATION, GIVEN X1, Y1, X2, Y2, X3, Y3.
  3: 3-POINT CUBIC INTERPOLATION, GIVEN X1, Y1, SLOPE, X2, Y2, X3, Y3.
  4: 4-POINT CUBIC INTERPOLATION, GIVEN X1, Y1, X2, Y2, X3, Y3, X4, Y4.

EPS MAY BE NEGATIVE.

If required minimum on y does not exists, or the function is ill-conditioned, xbar = eps-1.0 will be returned as an error indicator.

If desired interpolation is ill-conditioned, a lower order will be attempted, and ii will be changed accordingly.

II = 1: 2-POINT QUADRATIC INTERPOLATION

II = 2: 3-POINT QUADRATIC INTERPOLATION

II = 3: 3-POINT CUBIC INTERPOLATION

II = 4: 4-POINT CUBIC INTERPOLATION

DATA SET U477CNMN04 AT LEVEL 001 AS OF 02/13/81

DATA SET 9188CNMN04 AT LEVEL 001 AS OF 07/10/80

SUBROUTINE CNMN04 (II, XBAR, EPS, X1, Y1, SLOPE, X2, Y2, X3, Y3, X4, Y4)

II = CALCULATION CONTROL.
  1: 2-POINT QUADRATIC INTERPOLATION, GIVEN X1, Y1, SLOPE, X2 AND Y2.
  2: 3-POINT QUADRATIC INTERPOLATION, GIVEN X1, Y1, X2, Y2, X3, Y3.
  3: 3-POINT CUBIC INTERPOLATION, GIVEN X1, Y1, SLOPE, X2, Y2, X3, Y3.
  4: 4-POINT CUBIC INTERPOLATION, GIVEN X1, Y1, X2, Y2, X3, Y3, X4, Y4.

EPS MAY BE NEGATIVE.

If required minimum on y does not exits; or the function is ill-conditioned, xbar = eps-1.0 will be returned as an error indicator.

If desired interpolation is ill-conditioned, a lower order will be attempted, and ii will be changed accordingly.

II = 1: 2-POINT QUADRATIC INTERPOLATION

II = 2: 3-POINT QUADRATIC INTERPOLATION

II = 3: 3-POINT CUBIC INTERPOLATION

II = 4: 4-POINT CUBIC INTERPOLATION

DATA SET U477CNMN04 AT LEVEL 001 AS OF 02/13/81

DATA SET 9188CNMN04 AT LEVEL 001 AS OF 07/10/80

SUBROUTINE CNMN04 (II, XBAR, EPS, X1, Y1, SLOPE, X2, Y2, X3, Y3, X4, Y4)

II = CALCULATION CONTROL.
  1: 2-POINT QUADRATIC INTERPOLATION, GIVEN X1, Y1, SLOPE, X2 AND Y2.
  2: 3-POINT QUADRATIC INTERPOLATION, GIVEN X1, Y1, X2, Y2, X3, Y3.
  3: 3-POINT CUBIC INTERPOLATION, GIVEN X1, Y1, SLOPE, X2, Y2, X3, Y3.
  4: 4-POINT CUBIC INTERPOLATION, GIVEN X1, Y1, X2, Y2, X3, Y3, X4, Y4.

EPS MAY BE NEGATIVE.

If required minimum on y does not exits; or the function is ill-conditioned, xbar = eps-1.0 will be returned as an error indicator.

If desired interpolation is ill-conditioned, a lower order will be attempted, and ii will be changed accordingly.
ISH 0034  \( BB = (Y2-Y1)/X21 - AA*(X1+X2) \)
ISH 0035  \( XBAR = -0.5l(BB/AA \)
ISH 0036  IF (XBAR.LT.EPS) XBAR=XBAR1
ISH 0037 RETURN
ISH 0038 30 CONTINUE
ISH 0039 IF (NSLOP.EQ.0) RETURN
ISH 0040 GO TO 10
ISH 0041 40 CONTINUE

C------------------------------------------------------------------
C 11=3: 3-POINT CUBIC INTERPOLATION
C------------------------------------------------------------------
ISH 0044
ISH 0045 X21=X2-X1
ISH 0046 X31=X3-X1
ISH 0047 X32=X3-X2
ISH 0048 Q=X21*X31*X32
ISH 0049 IF (ABS(Q).LT.1.0E-20) RETURN
ISH 0050 X11=X1*X1
ISH 0051 DNOM=X21*X21*X11-X31*X32-X3*X3*X21
ISH 0052 IF (ABS(DNOM).LT.1.0E-20) GO TO 20
ISH 0053 AA=((X31*X31*(Y2-Y11-X21*X21*(Y3-Y11))/(X31*X21)-SLOPE*X32)/DNOM
ISH 0054 BB=((Y2-Y11)/X21-SLOPE-AA*(X2*X2+X1*X2-2.*X11))/X21
ISH 0055 CC=SLOPE-3.*AA*X11-2.*BB*X1
ISH 0056 BAC=BB*BB-3.*AA*CC
ISH 0057 IF (BAC.LT.0.) GO TO 20
ISH 0058 BAC=SQRT(BAC)
ISH 0059 XBAR=(BAC-BBJ/(3.*AAJ
ISH 0060 IF (XBAR.LT.EPS) XBAR=EPS
ISH 0061 RETURN
ISH 0062 50 CONTINUE

C------------------------------------------------------------------
C 11=4: 4-POINT CUBIC INTERPOLATION
C------------------------------------------------------------------
ISH 0064 X21=X2-X1
ISH 0065 X31=X3-X1
ISH 0066 X41=X4-X1
ISH 0067 X32=X3-X2
ISH 0068 X42=X4-X2
ISH 0069 X11=X1*X1
ISH 0070 X22=X2*X2
ISH 0071 X33=X3*X3
ISH 0072 X44=X4*X4
ISH 0073 Q1=/(X11*X32-X22*X31)*X3*X3*X21
ISH 0074 Q2=/(X41*X22-X22*X41)*X4*X4*X21
ISH 0075 Q3=/(X41*X21-X21*X41)*X4*X4*X21
ISH 0076 Q4=/(X31*X22-X22*X31)*X3*X3*X21
ISH 0077 Q5=/(X11*X32-X22*X31)*X3*X3*X21
ISH 0078 Q6=/(X41*X22-X22*X41)*X4*X4*X21
ISH 0079 Q7=/(X41*X21-X21*X41)*X4*X4*X21
ISH 0080 Q8=/(X31*X22-X22*X31)*X3*X3*X21
ISH 0081 IF (ABS(Q2).LT.1.0E-30) RETURN
ISH 0082 Q1=/(X11*X32-X22*X31)*X3*X3*X21
ISH 0083 Q2=/(X41*X22-X22*X41)*X4*X4*X21
ISH 0084 Q3=/(X41*X21-X21*X41)*X4*X4*X21
ISH 0085 Q4=/(X31*X22-X22*X31)*X3*X3*X21
ISH 0086 Q5=/(X11*X32-X22*X31)*X3*X3*X21
ISH 0087 Q6=/(X41*X22-X22*X41)*X4*X4*X21
ISH 0088 Q7=/(X41*X21-X21*X41)*X4*X4*X21
ISH 0089 Q8=/(X31*X22-X22*X31)*X3*X3*X21
ISH 0090 IF (ABS(Q2).LT.1.0E-30) GO TO 60
ISH 0091 AA=((Q1+Q2-Q3-Q4)/DNOM
ISH 0092 BB=((Q1-Q3-Q4)/QQ1
ISH 0093 CC=/(Y2-Y11-AAM/(X22-X11))1/X21-BB/(X1+X2)
ISH 0094 BAC=BB*BB-3.*AA*CC
ISH 0095 IF (ABS(AA).LT.1.0E-20 OR BAC.LT.0.) GO TO 60
ISH 0097 BAC=SQR(BAC) 00114
ISH 0098 XBAR=(BAC-GB)/(3.*AA) 00115
ISH 0099 IF (XBAR.LT.EPS) XBAR=XBAR1 00116
ISH 0100 RETURN 00117
ISH 0101 CONTINUE 00118
ISH 0102 IF (NSDOP.EQ.1) GO TO 40 00119
ISH 0103 GO TO 20 00120
ISH 0104 END 00121

*OPTIONS IN EFFECT* NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOBLINDONE)
*OPTIONS IN EFFECT* SOURCE EBCDIC NODIAG NODECK OBJECT NODAP NOFORMAT GOSTMT NOXREF NDARC NDANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 105, PROGRAM SIZE = 1754, SUBPROGRAM NAME = CN3H04
*STATISTICS* NO DIAGNOSTICS GENERATED

***** END OF COMPIILATION ***** 3004K BYTES OF CORE NOT USED
SUBROUTINE CNNH05 (G,DF,A,S,B,C,SLP,PHA,ISC,ISCI,N1,N2,N5,N7)

READ DATA SET U477CNMH05 AT LEVEL 002 AS OF 03/13/81
READ DATA SET U477CNMH05 AT LEVEL 001 AS OF 02/13/81
READ DATA SET 918SCNIN05 AT LEVEL 001 AS OF 07/10/80
DATA SET U477CNMH05 AT LEVEL 002 AS OF 03/13/81
DATA SET U477CNMH05 AT LEVEL 001 AS OF 02/13/81
DATA SET 918SCNIN05 AT LEVEL 001 AS OF 07/10/80

SUBROUTINE CNNH05 (G,DF,A,S,B,C,SLP,PHA,ISC,ISCI,N1,N2,N5,N7)

COMMON /CHMN1/ DELFUN,DABFUN,FDCH,FDCHN,CT,CTHIN,CTL,CTLN,ALPHA00005
1,ABOBIJ,THETA,OBJ,NOV,NCON,NSIDE,IPRINT,NDFG,NDFC,NUMB,IMAX,IIMAX,IT00006
2RMT,ICHDIR,IGOTO,MAC,INFO,ITER

COMMON /UOS/ ID, I6I

DIMENSION DF(NL), G(N2), ISC(N2), ISCI(N3), A(N1,N3), S(N1), C(N4)

ROUTINE TO SOLVE DIRECTION FINDING PROBLEM IN MODIFIED METHOD OF
FEASIBLE DIRECTIONS.
BY G. N. VOERPLAATS MAY, 1972.
NASA-AIRES RESEARCH CENTER, MOFFETT FIELD, CALIF.
NORM OF S VECTOR USED HERE IS S-TRANSPOSE TIMES S.LE.1.

IF NVC = 0 FIND DIRECTION BY ZOUTENDIJK'S METHOD. OTHERWISE
FIND MODIFIED DIRECTION.

----------------------------------
*** NORMALIZE GRADIENTS, CALCULATE THETA'S AND DETERMINE NVC ***
----------------------------------

NDV1=NDV+1
NDV2=NDV+2
NAC1=NAC+1
NVC=0
THNAX=0.

CTA=ABS (CT)
CTC=CTAM
IF (NCJ.LE.0) GO TO 10
CTC=CTBH
CTD=CT2
IF (Cl.GT. CTC) NVC=HVC+1
THT=0.
GG=I.+CTD*Cl
IF (HCJ.EQ.O.OR.Cl.GT. CTC) THT=THETA*GG*GG
IF (THT.GT.50.1 THT=50. 0
IF (THT.GT.THNAX) THNAX=THT
A(NDV1,II=THT
----------------------------------
NORMALIZE GRADIENTS OF CONSTRAINTS
----------------------------------
A(NDV2,II=I.

CALCULATE THETA
NCI=IC(I)
NC=1
NCJ=1
IF (NCJ.LE.0) NCJ=ISC(NCI)
C1=G(NCI)
CTD=CT1
CTC=CTAM
IF (NCJ.LE.0) GO TO 10
CTC=CTH
CTD=CIM

IF (C1.GT.CTC) NVC=NVC+1
THT=0.
EG=1.+CTD*C1
IF (NCJ.EQ.0.OR.C1.GT.CTC) THT=THETA*EG*GG
THT=THETA*GG
IF (THT.GT.50.) THT=50.
IF (THT.GT.THZAX) THZAX=THT
A(NDV1,II=I."
IF (NCI.GT.NCON) GO TO 40
A1=0.
DO 20 J=1,NDV
A1=A1+A(J,I)**2
20 CONTINUE
IF (A1.LT.1.0E-20) A1=1.0E-20
A1=SQRT(A1)
A(NDV2,I)=A1
A1=1./A1
DO 50 J=1,NDV
A(J,I)=A1*A(J,I)
40 CONTINUE

C------------------------------------------------------------------
C NORMALIZE GRADIENT OF OBJECTIVE FUNCTION AND STORE IN NAC+1
C COLUMN OF A
C------------------------------------------------------------------
A1=0.
DO 50 I=1,NDV
A1=A1+DF(I)**2
50 CONTINUE
IF (A1.LT.1.0E-20) A1=1.0E-20
A1=SQRT(A1)
A1=1./A1
DO 60 I=1,NDV
60 A(I,HAC1)=A1*DF(I)
C BUILD VECTOR.

IF (NYC.GT.0) GO TO 80
C------------------------------------------------------------------
C BUILD FOR CLASSICAL METHOD
C------------------------------------------------------------------
NDB=HAC1
A(NDV1,NDB)=1.
DO 70 I=1,NDB
70 C(I)=-A(TL,DV1,I)
GO TO 110
80 CONTINUE
C------------------------------------------------------------------
C BUILD FOR MODIFIED METHOD
C------------------------------------------------------------------
NDB=NAC
A(NDV1,NAC1)=-PHI
C------------------------------------------------------------------
SCALE THETA'S SO THAT MAXIMUM THETA IS UNITY
C------------------------------------------------------------------
IF (THMAX.GT.0.0001) THMAX=1./THMAX
DO 90 I=1,NDB
A(NDV1,I)=A(NDV1,I)*THMAX
90 CONTINUE
DO 100 I=1,NDB
100 C(I)=C(I)+A(J,I)*A(J,NAC1)
110 CONTINUE
C------------------------------------------------------------------
C BUILD B MATRIX
C------------------------------------------------------------------
DO 120 I=1,NDB
120 DO 120 J=1,NDV
B(I,J)=0.
DO 120 J=1,NDV
B(I,J)=0.
ISH 0089  DO 120 K=1,NDV1
ISH 0090  B(I,J)=B(I,J)-A(K,I)*A(K,J)
C
---------------------------
C  SOLVE SPECIAL L. P. PROBLEM
C
---------------------------
ISH 0091  CALL CNMN08 (ND,B,NC,VH,B,NC,VH,B,NC,VH,B,NC,VH)
ISH 0092  IF (PRINT.GT.1.AND.NER.GT.0) WRITE (161,160)
ISH 0093  CALL RESULTING DIRECTION VECTOR, S.
ISH 0094  SLOPE=0.
ISH 0095  DO 140 I=I,NDV1
ISH 0096  SI=0.
ISH 0097  IF (VC.GT.0) SI=-AII,HASII
ISH 0098  DO 130 J=I,NOB
ISH 0099  SI=SI-A(I,J)*C(J)
ISH 1000  SLOPE=SLOPE+SI*DF(I)
ISH 1001  140 S(I)=SI
ISH 1002  SINDVI1=1.
ISH 1003  IF (VC.GT.0) 5(NDV1)=1.
ISH 1004  S(NDV1)=1.
ISH 1005  DO 150 J=I,NOB
ISH 1006  150 SINOV11=-AINDV1,J )*C(J)
C
---------------------------
C  USABLE-FEASIBLE DIRECTION
C
---------------------------
ISH 1007  160 CONTINUE
ISH 1008  S=0.
ISH 1009  DO 160 I=I,NDV1
ISH 1010  AI=ABS(S(I))
ISH 1011  IF (AI.GT.SI) SI=AI
ISH 1012  CONTINUE
ISH 1013  DO 170 J=I,NOB
ISH 1014  IF (SI.LT.1.0E-10) RETURN
ISH 1015  SI=1./SI
ISH 1016  RETURN
ISH 1017  DO 170 I=1,NDV1
ISH 1018  S(I)=SI*SII
ISH 1019  SLOPE=SI*SLOPE
ISH 1020  S(NDV1)=SI*S(NDV1)
ISH 1021  RETURN
C
---------------------------
C  FORMATS
C
---------------------------
ISH 1022  FORMAT (/5X,46H* DIRECTION FINDING PROCESS DID NOT CONVERGE/5X,0157
ISH 1023  129H* S-VECTOR MAY NOT BE VALID)
ISH 1024  END
C
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOBL(KNONE)
*OPTIONS IN EFFECT*SOURCE ECCDIC KOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NDANSF TERM ISG FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 122, PROGRAM SIZE = 2548, SUBPROGRAM NAME =CHNN05
*STATISTICS* NO DIAGNOSTICS GENERATED
****** END OF COMPILATION ******

299K BYTES OF CORE NOT USED
SUBROUTINE CHMN06 (X,V1B,VUB,G,SCAl,OF,S,Gl,G2,CTAM,CTBN,SlOPE,AlP00003
1, A2, A3, A4, F1, F2, F3, CV1, CV2, CV3, CV4, ALPCA, ALPFES, ALPN, ALFIN, ALPCO00004
2, ALPSAV, ALPSID, ALPTOT, ISCIN2), HCAU2) 00001
C ROUTINE TO SOLVE ONE-DIMENSIONAL SEARCH PROBLEM FOR CONSTRUIED 00013
C FUNCTION MINIMIZATION. 00016
C BY G. N. VANDERPLAATS 00017
C NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF. 00018
C OBJ = INITIAL AND FINAL FUNCTION VALUE. 00019
C ALP = MOVE PARAMETER. 00020
C SLOPE = INITIAL SLOPE. 00021
C ALPSID = MOVE TO SIDE CONSTRAINT. 00024
C ALPFES = MOVE TO FEASIBLE REGION. 00025
C ALPNC = MOVE TO NEW NON-LINEAR CONSTRAINT. 00026
C ALPC = MOVE TO RE-ENCOUNTER CURRENTLY ACTIVE CONSTRAINT. 00029
C ALPIN = MOVE TO MINIMIZE FUNCTION. 00030
C ALPTOT = TOTAL MOVE PARAMETER. 00031
C TOLERANCES. 00034
C CTAM = ABS(CTMIN) 00037
C CTBN = ABS(CTLMIN) 00038
C PROPOSED MOVE. 00039
C CONTINUE 00040
C **** BEGIN SEARCH OR IMPOSE SIDE CONSTRAINT MODIFICATION **** 00041
C *** BEGIIH SEARCH OR IMPOSE SIDE CONSTRAINT MODIFICATION *** 00042
C AI=ALPSAV 00043
C ICOUNT=ICOUNT+1 00044
C ALPSID=1.0E+20 00045
C INITIAL ALPHA AND OBJ. 00046
C ALP=0. 00047
C F1=OBJ 00048
C KSID=0 00049
C IF (NSIDE.EQ.0) GO TO 70 00050
C FIND MOVE TO SIDE CONSTRAINT AND INSURE AGAINST VIOLATION OF 00051
C SIDE CONSTRAINTS 00052
C

DO 60 I=1,NDV
SI=S(I)
C ITH COMPONENT OF S IS SMALL. SET TO ZERO.
IF (ABS(S(I).GT.1.0E-20) GO TO 30
S(I)=0.
SLOPE=SLOPE-SI*DF(I)
GO TO 60
CONTINUE
XI=X(I)
XI=1./S1
IF (SI.GT.O.) GO TO 40
LCWER BOUND.
XI2=VLB(I)
XI1=ABS(XI2)
IF (XIl.LT.1.1 XI1=1.
CONSTRAINT VALUE.
GI=(XI2-XII)/XI1
IF (GI.GT.-1.0E-6) GO TO 50
PROPOSED MOVE TO LOWER BOUND.
ALPA=(XI2-XII)*SI
IF (ALPA.LT.ALPSID I ALPSID=ALPA
GO TO 60
CONTINUE
UPPER BOUND.
XI2=VUB(I)
XI1=ABS(XI2)
IF (Xl1.LT.1.1 XI1=1.
CONSTRAINT VALUE.
GI=(XII-XII)/XI1
IF (GI.GT.-1.0E-6) GO TO 50
PROPOSED MOVE TO UPPER BOUND.
ALPA=(XI2-XII)*SI
IF (ALPA.LT.ALPSID I ALPSID=ALPA
GO TO 60
CONTINUE
MOVE WILL VIOLATE SIDE CONSTRAINT. SET S(I)=0.
SLOPE=SLOPE-SI*DF(I)
S(I)=0.
KSID=KSID+1
CONTINUE
ALPSID IS UPPER BOUND ON ALPHA.
IF (A2.GT.ALPSID) A2=ALPSID
CONTINUE
CHECK ILL-CONDITIONING
------------------------------------------------------------------
IF (KSID.EQ.NDV.OR.ICOUNT.GT.101 GO TO 7
IF (NVC.EQ.O.AND.SLOPE.GT.O.) GO TO 7
ALPFEES=-l.
ALPMIN=-l.
ALPLN=1.1*ALPSID
ALPCA=ALPSID
IF (ICOUNT.EQ.O) GO TO 90
STORE CONSTRAINT VALUES IN GI.
DO 80 I=1,NCON
G1(I)=GI
CONTINUE
------------------------------------------------------------------
DO 60 I=1,NDV
SLOPE=SLOPE-SI*DF(I)
S(I)=0.
KSID=KSID+1
CONTINUE
ALPSID IS UPPER BOUND ON ALPHA.
IF (A2.GT.ALPSID) A2=ALPSID
CONTINUE
CHECK ILL-CONDITIONING
------------------------------------------------------------------
IF (KSID.EQ.NDV.OR.ICOUNT.GT.101) GO TO 710
IF (NVC.EQ.O.AND.SLOPE.GT.O.) GO TO 710
ALPFEES=-l.
ALPMIN=-l.
ALPUN=1.1*ALPSID
ALPNC=ALPSID
ALPCA=ALPSID
IF (ICOUNT.EQ.O) GO TO 90
STORE CONSTRAINT VALUES IN GI.
DO 80 I=1,NCON
GI(I)=GI(I)
CONTINUE
CONTINUE
CONTINUE
CONTINUE
C MOVE A DISTANCE A2*S

DO 100 I=1,NDV

XII)=X(I)+A2*SII)

100 CONTINUE

IF (IPRINT.LT.5) GO TO 130

WRITE (161,740) A2

IF (INSCAL.EQ.O) GO TO 120

DO 110 I=1,NOV

110 GII)=SCAL(I)*XII) WRITE (161,750) (G(Il),I=1,NDV)

GO TO 130

WRITE (161,750) (XII),I=1,NOV

C UPDATE FUNCTION AND CONSTRAINT VALUES

130 NCAL(I)=NCAL(I)+1

JGOTO=1

RETURN

F2=OBJ

IF (IPRINT.GE.5) WRITE (161,760) F2

IF (IPRINT.LT.5.OR.NCON.EQ.O) GO TO 150

WRITE (161,750) (G(I),I=1,NCON)

150 CONTINUE

C IDENTIFY ACCAPTABLE OF DESIGNS F1 AND F2

IGOOD1=0

IGOOD2=0

CV1=0.

CV2=0.

NVC1=0

IF (NCON.EQ.O) GO TO 170

DO 160 I=1,NCON

CC=CTAM

IF (ISC(I).GT.0) CC=CTBM

160 CONTINUE

IF (C2.GT.0.) IG002=1

IF (C1.GT.CV1) CV1=C1

IF (C2.GT.CV2) CV2=C2

170 CONTINUE

IF (CV1.GT.0.) IG002=1

IF (CV2.GT.0.) IG002=1

ALP=A2

OBJ=F2

C IF F2 VIOLATES FEWER CONSTRAINTS THAN F1 BUT STILL HAS CONSTRAINT

C VIOLATIONS RETURN

C IF (NVC1.LT.NVC.AND.NVC1.GT.0) GO TO 710
IDENTIFY BEST OF DESIGNS F1 ANF F2

IBEST CORRESPONDS TO MINIMUM VALUE DESIGN.

IF CONSTRAINTS ARE VIOLATED, IBEST CORRESPONDS TO MINIMUM CONSTRAINT VIOLATION.

IF (IGOOD1.EQ.0.AND.IGOOD2.EQ.0) GO TO 180

VIOLATED CONSTRAINTS. PICK MINIMUM VIOLATION.

IBEST=1

IF (CV1.GE.CV2) IBEST=2

GO TO 190

CONTINUE

C NO CONSTRAINT VIOLATION. PICK MINIMUM F.

IBEST=1

IF (F2.LE.F1) IBEST=2

CONTINUE

IF (IIGOOD1.EQ.0.AND.IGOOD2.EQ.0) GO TO 230

PROJECTED MOVE

AT LEAST FAR ENOUGH TO OVERCOME CONSTRAINT VIOLATIONS.

A3=ALPFES

MOVE TO MINIMIZE FUNCTION.

IF (ALPHIN.GT.A3) A3=ALPHIN

IF A3.LE.0, SET A3 = ALPSID.

IF (A3.LE.0.) A3=ALPSID
C LIMIT MOVE TO NEW CONSTRAINT ENCOUNTER.

C IF (A3.GT.ALPNC) A3=ALPNC
C MAKE A3 NON-ZERO.
C IF (A3.LE.1.0E-20) A3=1.0E-20
C IF A3=A2=ALPSID AND F2 IS BEST, GO INVOKE SIDE CONSTRAINT MODIFICATION.

C ISN 0192
C ISN 0194
C ISN 0196
C PRECISION
C C C C C
C ISN 0198
C ISN 0199
C ISN 0200
C ISN 0201
C ISN 0203
C ISN 0204
C ISN 0205
C ISN 0206
C ISN 0207
C ISN 0208
C ISN 0209
C ISN 0210
C ISN 0211
C ISN 0212
C ISN 0213
C ISN 0215
C ISN 0216
C ISN 0217
C ISN 0218
C ISN 0219
C ISN 0220
C ISN 0221
C ISN 0222
C ISN 0223
C ISN 0224
C ISN 0225
C ISN 0226
C ISN 0227
C ISN 0228
C ISN 0229
C ISN 0230
C ISN 0231
C ISN 0232
C ISN 0233
C ISN 0234
C ISN 0235
C ISN 0236
C ISN 0237
C ISN 0238
C ISN 0239
C ISN 0240
C ISN 0241
C ISN 0242
C ISN 0243
C ISN 0244
C ISN 0245
C ISN 0246
C ISN 0247
C ISN 0248
C ISN 0249
C ISN 0250
C ISN 0251
C ISN 0252
C ISN 0253
C ISN 0254
C ISN 0255
C ISN 0256
C ISN 0257
C ISN 0258
C ISN 0259
C ISN 0260
C ISN 0261
C ISN 0262
C ISN 0263
C ISN 0264
C ISN 0265
C ISN 0266
C ISN 0267
C ISN 0268
C ISN 0269
C ISN 0270
C ISN 0271
C ISN 0272
C ISN 0273
C ISN 0274
C ISN 0275
C ISN 0276
C ISN 0277
C ISN 0278
C ISN 0279
C ISN 0280
C ISN 0281
C ISN 0282
C ISN 0283
C ISN 0284
C ISN 0285
C ISN 0286
C ISN 0287
C ISN 0288
C ISN 0289
C ISN 0290
ISH 0248 CV3=0.
ISH 0249 IGOOD3=0
ISH 0250 NVC1=0
ISH 0251 IF (NCON.EQ.0) GO TO 340
ISH 0253 DO 330 I=1,NCON
ISH 0254 CC=CTAH
ISH 0255 IF (ISC(I).GT.0) CC=CTBH
ISH 0256 CI=G(I)-CC
ISH 0258 IF (CI.GT.CV3) CV3=CI
ISH 0260 IF (CI.GT.0.) NVC1=NVC1+1
ISH 0262 330 CONTINUE
ISH 0263 IF (CV3.GT.0.) IGOOD3=1
ISH 0265 340 CONTINUE

C DETERMINE BEST DESIGN.
ISH 0266 IF (IBEST.EQ.2) GO TO 360
ISH 0268 IF (IGOOD1.EQ.0.AND.IGOOD3.EQ.0) GO TO 350
ISH 0270 IF (CV1.GE.CV3) IBEST=3
ISH 0272 GO TO 360
ISH 0273 350 IF (F3.LE.F1) IBEST=3
ISH 0275 GO TO 360
ISH 0276 360 CONTINUE
C CHOOSE BETWEEN F1 AND F3.
ISH 0277 IF (IGOOD2.EQ.0.AND.IGOOD3.EQ.0) GO TO 370
ISH 0279 IF (CV2.GE.CV3) IBEST=3
ISH 0281 GO TO 360
ISH 0282 370 IF (F3.LE.F2) IBEST=3
ISH 0284 380 CONTINUE
ISH 0285 ALP=A3
ISH 0286 OBJ=F3
C IF F3 VIOLATES FEWER CONSTRAINTS THAN F1 RETURN.
ISH 0287 IF (NVC1.LT.NVC) GO TO 710
ISH 0289 IF (LINOB.NE.0.AND.NUNL.EQ.NCON) GO TO 710
C IF A3 = ALPNSD AND F3 IS BOTH GOOD AND BEST RETURN.
ISH 0291 ALPB=1.-ALPNSD/A3
ISH 0292 IF (ABS(ALPB).LT.1.0E-20.AND.IBEST.EQ.3) GO TO 710
C IF A3 = ALPNSD AND BEST, GO INVOKE SIDE CONSTRAINT MODIFICATION.
ISH 0294 ALPA=1.-ALPNSD/A3
ISH 0295 IF (ABS(ALPA).LT.1.0E-20.AND.IBEST.EQ.3) GO TO 20
C ** ****** 3 - POINT INTERPOLATION **********
C IPI1=I.-ALPNSD/A3
ISH 0297 ALPNSD=ALPNSD
ISH 0298 ALPCA=ALPCA
ISH 0299 ALPFES=-1.
ISH 0300 ALPHIN=-1.
ISH 0301 IF (NCON.EQ.0) GO TO 440
ISH 0303 III=0
ISH 0304 390 III=III+1
ISH 0305 CI=G(I)(III)
ISH 0306 C2=G2(III)
ISH 0307 C3=G3(III)
ISH 0308 IF (ISC(III).EQ.0) GO TO 440
C LINEAR CONSTRAINT. FIND ALPFES ONLY. ALPNSD SAME AS BEFORE.
IF (C1.LE.CTBM) GO TO 430
11=1
CALL CNMN07 (II,ALP,ZRO,ZRO,C1,A2,C2,A3,C3)
IF (ALP.GT.ALPFES) ALPFES=ALP
GO TO 430
CONTINUE
C
NON-LINEAR CONSTRAINT
C
II=2
CALL CNMN07 (II,ALP,ZRO,ZRO,C1,A2,C2,A3,C3)
IF (ALP.LE.ZROI) GO TO 430
IF (C1.GE.CT.AND.C1.LE.0.) GO TO 410
IF (C1.GT.CTAM.OR.C1.LT.0.) GO TO 420
ALP IS MINIMUM MOVE. UPDATE FOR NEXT CONSTRAINT ENCOUNTER.
ALPA=ALP
GO TO 430
CONTINUE
C
NON-LINEAR CONSTRAINT
C
II=3
IF (A2.GT.A3.AND.(IGOOD2.EQ.0.AND.IBEST.EQ.2)) II=2
CALL CNMN04 (II,ALPMIN,ZRO,ZRO,F1,SLOPE,A2,F2,A3,F3,ZRO,ZROI)
C
PROPOSED MOVE
C
MOVE AT LEAST ENOUGH TO OVERCOME CONSTRAINT VIOLATIONS.
A4=ALPFES
MOVE TO MINIMIZE FUNCTION.
IF (ALPMIN.GT.A4) A4=ALPMIN
C
IF A4.LE.0, SET A4 = ALPSID. 
C
IF (C1.GT.A4) A4=ALPSID
LIMIT MOVE TO NEW CONSTRAINT ENCOUNTER.
IF (A4.GT.ALPFES) A4=ALPFES
C
LIMIT MOVE TO RE-ENCOUNTER CURRENTLY ACTIVE CONSTRAINT.
C
IF (A4.GT.ALPFSA) A4=ALPFSA
C
LIMIT MOVE TO RE-ENCOUNTER CURRENTLY ACTIVE CONSTRAINT.
C
IF (C1.GT.(5.*A3)) A4=5.*A3
C
UPDATE DESIGN.
C
IF (IBEST.NE.3.OR.NCON.EQ.0) GO TO 470
C
STORE CONSTRAINT VALUES IN G2. F3 IS BEST. F2 IS NOT.
DO 460 I=1,NCON
C
IF A4=A3 AND IGOOD1=0 AND IGOOD3=1, SET A4=.9*A3.
C
IF (IGOOD1.EQ.0.AND.IGOOD3.EQ.1.AND.ABS(ALP).LT.1.0E-20) A4=.9*A00408
*VERSION 1.3.0 (01 MAY 80) CNN06 SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 82.141/10.53.39  PAGE 8

13  

C  --------  \------------------------  
C  MOVE A DISTANCE A4*S  \------------------------  
C  --------  

C  \------------------------  

ISN 0369  
ALP=A4-A3  

ISN 0370  
ALPTOT=ALPTOT+ALP  

ISN 0371  
DO 460 I=1,NDV  

ISN 0372  
X(I)=X(I)+ALP*S(I)  

ISN 0373  
460 CONTINUE  

ISN 0374  
IF (IPRINT.LT.5) GO TO 510  

ISN 0375  
WRITE (I61,720)  

ISN 0376  
WRITE (I61,740)  

ISN 0377  
WRITE (I61,740) A4  

ISN 0378  
IF (NSCAL.EQ.0) GO TO 500  

ISN 0379  
DO 490 I=1,NDV  

ISN 0380  
490 G(I)=SCAL(I)*X(I)  

ISN 0381  
WRITE (161,750) (G(I),I=I,NDV)  

ISN 0382  
WRITE (161,750) (X(I),I=I,NDV)  

ISN 0383  
CONTINUE  

ISN 0384  
500 WRITE (I61,750) (X(I),I=I,NDV)  

ISN 0385  
510 CONTINUE  

C  \------------------------  
C  UPDATE FUNCTION AND CONSTRAINT VALUES  \------------------------  
C  \------------------------  

ISN 0386  
NCAL(I)=NCAL(I)+1  

ISN 0387  
JGOTO=3  

ISN 0388  
RETURN  

ISN 0389  
520 CONTINUE  

ISN 0390  
F4=OBJ  

ISN 0391  
IF (IPRINT.GE.5) WRITE (I61,750) F4  

ISN 0392  
IF (IPRINT.LT.5.FOR.NCON.EQ.0) GO TO 530  

ISN 0393  
WRITE (I61,750) (G(I),I=I,NCON)  

ISN 0394  
CONTINUE  

ISN 0395  
DETERMINE ACCPAPTABILITY OF F4.  

ISN 0396  
IGOOD4=0  

ISN 0397  
CV4=0.  

ISN 0398  
IF (NCON.EQ.0) GO TO 550  

ISN 0399  
DO 540 I=1,NCON  

ISN 0400  
CC=CTAM  

ISN 0401  
IF (ISC(I).LT.0) CC=CTBM  

ISN 0402  
C1=G(C)-CC  

ISN 0403  
IF (C1.GT.CV4) CV4=C1  

ISN 0404  
530 CONTINUE  

C  \------------------------  
C  DETERMINE BEST DESIGN  \------------------------  
C  \------------------------  

ISN 0405  
GO TO (560,610,660),IBEST  

ISN 0406  
550 CONTINUE  

C  \------------------------  
C  CHOOSE BETWEEN F1 AND F4.  \------------------------  
C  \------------------------  

ISN 0407  
IF (IGOOD4.EQ.0.AND.IGOOD4.EQ.0) GO TO 570  

ISN 0408  
IF (C4.GT.0.) IGOOD4=1  

ISN 0409  
CONTINUE  

ISN 0410  
IF (F4.LE.F1) GO TO 710  

ISN 0411  
CONTINUE  

C  \------------------------  
C  F1 IS BEST.  \------------------------  

ISN 0412  
CONTINUE  

C  \------------------------  
C  DETERMINE BEST DESIGN  \------------------------  
C  \------------------------  

ISN 0413  
ALP=A4  

ISN 0414  
OBJ=F4  

ISN 0415  
Determine best design.
ISN 0486 720 FORMAT (/5X,25HTHREE-POINT INTERPOLATION) 00527
ISN 0487 730 FORMAT (/5X,50H* * CONstrained ONE-DIMENSIONAL SEARCH INFORMATION) 00528
ISN 0488 740 FORMAT (/5X,15HPROPOSED DESIGN/5X,7HLHALPHA =,E12.5/5X,6HVECTOR) 00530
ISN 0489 750 FORMAT (1X,6E12.4) 00531
ISN 0490 760 FORMAT (/5X,5HOBJ =,E13.5) 00532
ISN 0491 770 FORMAT (/5X,17HCONSTRAINT VALUES) 00533
ISN 0492 780 FORMAT (/5X,23HTWO-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) AUTOCBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOAISF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 493, PROGRAM SIZE = 6746, SUBPROGRAM NAME =CIMN06
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPIlation *****

2924K BYTES OF CORE NOT USED
SUBROUTINE CMN07 (II,XBAR, EPS,X1,Y1,X2,Y2,X3,Y3)

ROUTINE TO FIND FIRST XBAR.GE.EPS CORRESPONDING TO A REAL ZERO
OF A ONE-DIMENSIONAL FUNCTION BY POLYNOMIAL INTERPOLATION.

BY G. N. VANDERPLAATS APRIL, 1972.
NASA-AHES RESEARCH CENTER, MOFFETT FIELD, CALIF.

II = CALCULATION CONTROL.

1: 2-POINT LINEAR INTERPOLATION, GIVEN X1, Y1, X2 AND Y2.

2: 3-POINT QUADRATIC INTERPOLATION, GIVEN X1, Y1, X2, Y2, X3 AND Y3.

EPS MAY BE NEGATIVE.

IF REQUIRED ZERO ON Y DOES NOT EXIST, OR THE FUNCTION IS
ILL-CONDITIONED, XBAR = EPS-1.0 WILL BE RETURNED AS AN ERROR INDICATOR.

IF DESIRED INTERPOLATION IS ILL-CONDITIONED, A LOWER ORDER
INTERPOLATION, CONSISTENT WITH INPUT DATA, WILL BE ATTEMPTED AND
II WILL BE CHANGED ACCORDINGLY.

II = 1: EPS-1.0

II = 0

CONTINUE

II = 1: 2-POINT LINEAR INTERPOLATION

II = 2: 3-POINT QUADRATIC INTERPOLATION
ISH 0040  
BB=(Y2-Y1)/X21-AA*(X1+X2)  
ISH 0041  
CC=Y1-X1*(AA*AA+BB)  
ISH 0042  
BAC=BB+BB-4.*AA*CC  
ISH 0043  
IF (BAC.LT.0.) GO TO 10  
ISH 0045  
BAC=SQRT(BAC)  
ISH 0046  
AA=.5/AA  
ISH 0047  
XBAR=AA*(BAC-BB)  
ISH 0048  
XB2=-AA*(BAC+BB)  
ISH 0049  
IF (XBAR.LT.EPS) XBAR=XB2  
ISH 0051  
IF (XB2.LT.XBAR.AND.XB2.GT.EPS) XBAR=XB2  
ISH 0053  
IF (XBAR.LT.EPS) XBAR=XBAR1  
ISH 0055  
RETURN  
ISH 0056  
END  
*OPTIONS IN EFFECT*NAME(NAMEN) OPTIMIZE(3) LINESIZE(60) AUTODEF(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 =CN07  
*STATISTICS* NO DIAGNOSTICS GENERATED  
***** END OF COMPILATION *****  
3016K BYTES OF CORE NOT USED
SUBROUTINE CNNN08 (NOB, NER, C, MS1, B, N3, N4, N5) 00002
DIMENSION C(N4), B(N3,N3), MS(N5) 00003
ROUTINE TO SOLVE SPECIAL LINEAR PROBLEM FOR IMPOSING S-TRANSPOSE TIMES S.BOUNDS IN THE MODIFIED METHOD OF FEASIBLE DIRECTIONS. 00004
BY G. N. VANDERPLAATS
APRIL, 1972. 00006
NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF. 00007
VECTOR MS1 IDENTIFIES THE SET OF BASIC VARIABLES. 00016
 **** BEGIN NEW ITERATION 000040
C ********** 000041
C ********** 000042
C ********** 000043
C ********** 000044
C ********** 000045
ISH 0043  30 CONTINUE 00055
ISH 0044   IF (CBMAX.LT.CBMIN) GO TO 70 00056
ISH 0046   IF (ICHK.EQ.0) GO TO 70 00057
ISH 0048        C UPDATE VECTOR MS1.
ISH 0049        JJ=ICHK 00059
ISH 0050        IF (ICHK.EQ.0) JJ=ICHK+NDB 00060
ISH 0051         KK=JJ+NDB 00061
ISH 0052         IF (KK.GT.M21) KK=JJ-NDB 00062
ISH 0054         MS1(KK)=ICHK 00063
ISH 0055         MS1(JJ)=0 00064
ISH 0056         C 00065
ISH 0057         C PIVOT OF B(I=ICHK,I=ICHK) 00066
ISH 0058         C 00067
ISH 0059         BB=1./B(I=ICHK,I=ICHK) 00068
ISH 0060         DO 40 J=1,1NDB 00069
ISH 0061         DO 40 B(I=ICHK,J)=BB*B(I=ICHK,J) 00070
ISH 0062         C(I=ICHK)=CBMAX 00071
ISH 0063         B(I=ICHK,I=ICHK)=BB 00072
ISH 0064         C 00073
ISH 0065         C ELIMINATE COEFFICIENTS ON VARIABLE ENTERING BASIS AND STORE 00074
ISH 0066         C COEFFICIENTS ON VARIABLE LEAVING BASIS IN THEIR PLACE. 00075
ISH 0067         DO 50 I=1,NDB 00076
ISH 0068         B(I,1)=B(I,ICHK) 00077
ISH 0069         BB1=B(I,ICHK) 00078
ISH 0070         DO 50 B(I,J)=B(I,J)-BB1*B(I=ICHK,J) 00079
ISH 0071         C(I)=C(I)-BB1*CBMAX 00080
ISH 0072         CONTINUE 00081
ISH 0073         NER=0 00082
ISH 0074         C 00083
ISH 0075         C STORE ONLY COMPONENTS OF U-VECTOR IN 'C'. USE B(I,1) FOR 00084
ISH 0076         C TEMPORARY STORAGE 00085
ISH 0077         C 00086
ISH 0078         DO 80 I=1,NDB 00087
ISH 0079         B(I,1)=C(I) 00088
ISH 0080         CONTINUE 00089
ISH 0081         DO 80 I=1,NDB 00090
ISH 0082         CONTINUE 00091
ISH 0083         CONTINUE 00092
ISH 0084         Go TO 20 00093
ISH 0085         CONTINUE 00094
ISH 0086         END 00095
ISH 0087         C 00096
ISH 0088         END 00097
ISH 0089         C 00098
ISH 0090         C 00099
ISH 0091         C OPTIONS IN EFFECT=NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOBL(NONE) 00100
ISH 0092         OPTIONS IN EFFECT=SOURCE EBCDIC NOLIST NODECK OBJECT NOMETADATA NOSLIST NOSTMT NOXREF NOALG NOSPL NOASF TERM IBM FLAG(I) 00101
ISH 0093         STATISTICS= SOURCE STATEMENTS = 84, PROGRAM SIZE = 1359, SUBPROGRAM NAME =CNMN08 00102
ISH 0094         STATISTICS= NO DIAGNOSTICS GENERATED 00103
ISH 0095         END OF COMPILATION 00104
ISH 0096         3004K BYTES OF CORE NOT USED 00105
SUBROUTINE COIIN (X,VLB,VUB,G,SCAL,DF,A,SC,IC,HS1,N)

C COMMON /CMN1/ DELFUN,DABFUN,FDCM,CTMIN,CTLMN,ALPHAM0006
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100

COMMON /CMN0/ U477CONM1H,ATLEVEL,ASOF

DATA SET U477CONM1H AT LEVEL 0001 AS OF 03/17/81
DATA SET U477CONM1H AT LEVEL 0002 AS OF 03/13/81
DATA SET U477CONM1H AT LEVEL 0003 AS OF 02/13/81
DATA SET U477CONM1H AT LEVEL 0004 AS OF 07/10/80

SUBROUTINE COIIN (X,VLB,VUB,G,SCAL,DF,A,SC,IC,HS1,N)

C ROUTINE TO SOLVE CONSTRAINED OR UNCONSTRAINED FUNCTION

C BY G. N. VANDERPLAATS APRIL, 1972.
C
C NASA-AMES RESEARCH CENTER, Moffett Field, Calif.
C
C REFERENCES:
C
C 1. NASA-AMES RESEARCH CENTER - A FORTRAN PROGRAM FOR CONSTRAINED FUNCTION

C STORAGE REQUIREMENTS:
C
C 2. PROGRAM - 7000 DECIMAL WORDS (CDC COMPUTER)
C 3. ARRAYS - APPROX. 2*(NDV**2)+27*NDV+4#CON,
C 4. WHERE N3 = NDV+2.

C RE-SCALE VARIABLES IF REQUIRED.
C
C IF (SCAL.EQ.0.OR.IGOTO.EQ.0) GO TO 20

C DO 10 I=1,NDV,1
C X(I)=C(I)
C
C CHECK FOR UNBOUNDED SOLUTION
C
C IF (OBJ IS LESS THAN -1.0E+40) STOP
C IF (OBJ.GT.-1.0E+40) GO TO 30

C WRITE (161,980) 00045
C GO TO 810
C CONTINUE
C
C CHECK FOR UNBOUNDED SOLUTION
C
C STOP IF OBJ IS LESS THAN -1.0E+40
C IF (OBJ.GT.-1.0E+40) GO TO 30
C WRITE (I61,980) 00045
C GO TO 810
C CONTINUE
C
C SAVE INPUT CONTROL PARAMETERS
C
C IF (PRINT.GT.0) WRITE (I61,1220)
C IF (LINOBJ.EQ.0.OR.(NCON.GT.0.OR.NSIDE.GT.0)) GO TO 50
C
C TOTALLY UNCONSTRAINED FUNCTION WITH LINEAR OBJECTIVE.
C SOLUTION IS UNBOUNDED.
WRITE (161,970) LINOBJ,NCON,NSIDE
RETURN
CONTINUE

C DEFAULTS
IF (ITRM .LE. 0) ITRM=3
IF (ITMAX .LE. 0) ITMAX=20
NDV1=NDV+1
IF (ICHDIR.EQ.0) ICNDIR=NDV1
IF (DELFUN .LE. 0.) DELFUN=.001
CT=-ABS(CT)
IF (CT.GE.0.) CT=-.1
CTMIN=ABS(CTMIN)
IF (CTMIN .LE. 0.) CTMIN=.1

C INITIALIZE INTERNAL PARAMETERS
INFOG=0
ITER=0
JDIR=0
IOBJ=0
KOBJ=0
NDV2=NDV+2
KCOUNT=0
NCAL(1)=0
NCAL(2)=0
NAC=0
NPEAS=0
MSCAL=NSCAL
CT1=ITRM
CT=1./CT1
DCT=CTMIN/ABS(CT)**CT1

C
C CALCULATE NUMBER OF LINEAR CONSTRAINTS, NLNC.

C CHECK TO BE SURE THAT SIDE CONSTRAINTS ARE SATISFIED

C INITIALIZE SCALING VECTOR, SCAL
**** Calculate Initial Function and Constraint Values ****

C

C

ISN 0151
INFO=1
00175
ISN 0152
NCAL(1)=1
00176
ISN 0153
IGOTO=1
00177
ISN 0154
GO TO 950
00178
ISN 0155
CONTINUE
00179
ISN 0156
OBJ=OBJ
00180
ISN 0157
IF (DABFUN.LE.0.) DABFUN = 0.01*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

C

ISN 0163
IF (IPRINT.LE.1) GO TO 230
00184
ISN 0165
IF (HSIDE.EQ.0.AND.NCON.EQ.0) WRITE (I61,1290)
00185
ISN 0167
IF (HSIDE.NE.0.OR.NCON.GT.0) WRITE (I61,1250)
00186
ISN 0169
WRITE (I61,1240) IPRINT,ITMX,NCON,NSIDE,ICHDIR,NSCAL,NFAC,LNDB00190
11,ITRM,NS,N,M,N,N
00187
ISN 0170
WRITE (I61,1260) CT,CTMIN,CT,CTMIN,THETA,PHI,DELFUN,DABFUN
00188
ISN 0171
WRITE (I61,1250) FDCH,FODH,ALPHAX,ABOBJ
00189
ISN 0172
IF (HSIDE.EQ.0) GO TO 190
00190
ISN 0174
WRITE (I61,1270)
00191
ISN 0175
DO 170 I=1,NDV,6
00192
ISN 0176
M1=MINDNV,I+5
00193
ISN 0177
170 WRITE (I61,1010) I,(VLB(I,J)=I,M1)
00194
ISN 0178
WRITE (I61,1280)
00195
ISN 0179
DO 180 I=1,NDV,6
00196
ISN 0180
M1=MINDNV,I+5
00197
ISN 0181
180 WRITE (I61,1010) I,(IG1(I,J)=I,M1)
00198
ISN 0182
190 CONTINUE
00199
ISN 0183
IF (NSCAL.GE.0) GO TO 200
00200
ISN 0185
WRITE (I61,1300)
00201
ISN 0186
WRITE (I61,1460) (SCAL(I),I=1,NDV)
00202
ISN 0187
200 CONTINUE
00203
ISN 0188
IF (NCON.EQ.0) GO TO 230
00204
ISN 0189
IF (NLNC.EQ.0.OR.NLNC.EQ.NCON) GO TO 220
00205
ISN 0192
WRITE (I61,1020)
00206
ISN 0193
DO 210 I=1,NCON,15
00207
ISN 0194
M1=MINDNCON,I+14
00208
ISN 0195
210 WRITE (I61,1030) I,(ISC(J)=I,M1)
00209
ISN 0196
GO TO 230
00210
ISN 0197
220 IF (NLNC.EQ.NCON) WRITE (I61,1040)
00211
ISN 0199
IF (NLNC.EQ.0) WRITE (I61,1050)
00212
ISN 0201
230 CONTINUE
00213
ISN 0202
WRITE (I61,1440) OBJ
00214
ISN 0203
WRITE (I61,1450)
00215
ISN 0204
DO 240 I=1,NDV
00216
ISN 0205
XI=1.
00217
ISN 0206
IF (NSCAL.NE.0) XI=SCAL(I)
00218
ISN 0208
240 GI(I)=XI(I)*XI
00219
ISN 0209
DO 250 I=1,NDV,6
00220
ISN 0210
M1=MINDNV,I+5
00221
ISN 0211
250 WRITE (I61,1010) I,(GI(J)=I,M1)
00222
ISN 0212
IF (NCON.EQ.0) GO TO 270
00223
ISN 0214
WRITE (I61,1470)
00224
ISN 0215
DO 260 I=1,NCON,6
00225
ISN 0216
M1=MINDNCON,I+5
00226
ISN 0217
260 WRITE (I61,1010) I,(GI(J)=I,M1)
00227
**VERSION 1.3.0 (01 MAY 80)  CONMIN  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 82.141/10.53.56  PAGE 5**

ISTN 0210  270 CONTINUE  00230
ISTN 0219  IF (IPRINT.GT.1) WRITE (I6I,1360)  00233
C  ------------------------------------ BEGIN MINIMIZATION ------------------------------------  00234
C  ------------------------------------  00235
ISTN 0221  280 CONTINUE  00236
ISTN 0222  ITER=ITER+1  00237
ISTN 0223  IF (ABOBJLT.0..0001) ABOBJL=.0001  00239
ISTN 0225  IF (ABOBJLT.2) ABOBJL=2  00240
ISTN 0227  IF (ALPHAX.GT.1.) ALPHAX=1.  00241
ISTN 0229  IF (ALPHAXLT.0001) ALPHAX=.001  00242
ISTN 0231  IF (IPRINT.GT.2) WRITE (I6I,1310) ITER  00243
ISTN 0233  IF (IPRINT.GT.3.AND.NCON.GT.0) WRITE (I6I,1320) CT,CTL,PHI  00244
ISTN 0235  CTA=ABS(CT)  00245
ISTN 0236  IF (NCOBJ.EQ.0) GO TO 340  00246
C  ------------------------------- END OF INITIALIZATION -------------------------------  00247
C  NO MOVE ON LAST ITERATION. DELETE CONSTRAINTS THAT ARE NO  00248
C  LONGER ACTIVE.  00249
C  ------------------------------------  00250
ISTN 0238  NNAC=NAC  00251
ISTN 0239  DO 290 I=1,NNAC  00252
ISTN 0240  IF (IC(I).GT.NCON) NAC=NAC-1  00253
ISTN 0242  290 CONTINUE  00254
ISTN 0243  IF (NAC.LE.0) GO TO 420  00255
ISTN 0245  NNAC=NAC  00256
ISTN 0246  DO 330 I=1,NNAC  00257
ISTN 0247  300 NIC=IC(I)  00258
ISTN 0248  CTL=CT  00259
ISTN 0249  IF (ISC(NIC).GT.0) CTL=CTL  00260
ISTN 0251  IF (G(NIC).GT.CTL) GO TO 330  00261
ISTN 0253  NAC=NAC-1  00262
ISTN 0254  IF (I.GT.NAC) GO TO 420  00263
ISTN 0256  DO 320 K=1,NAC  00264
ISTN 0257  II=K+1  00265
ISTN 0258  DO 310 J=1,NDV2  00266
ISTN 0259  310 AJ(K)=AJ(J,II)  00267
ISTN 0260  320 IC(K)=IC(II)  00268
ISTN 0261  GO TO 300  00269
ISTN 0262  330 CONTINUE  00270
ISTN 0263  GO TO 420  00271
ISTN 0264  340 CONTINUE  00272
ISTN 0265  IF (NSCAL.LT.NSCAL.OR.NSCAL.EQ.0) GO TO 360  00273
ISTN 0266  IF (NSCAL.TE.0.AND.KCOUNT.LT.ICNDIR) GO TO 360  00274
ISTN 0267  MSCAL=0  00275
ISTN 0270  KCOUNT=0  00276
C  ------------------------------- SCALE VARIABLES -------------------------------  00277
C  ------------------------------------  00278
ISTN 0271  DO 350 I=1,NDV  00279
ISTN 0272  SI=SCAL(I)  00280
ISTN 0273  XI=SI*XI(I)  00281
ISTN 0274  SIB=SI  00282
ISTN 0275  IF (NSCAL.GT.0) SIB=ABS(SI)  00283
ISTN 0277  IF (SI.LT.1.0E-10) GO TO 350  00284
ISTN 0279  SCAL(I)=SI  00285
ISTN 0280  SI=1./SI  00286
ISTN 0281  X(I)=XI*SI  00287
ISTN 0282  IF (HSIDE.EQ.0) GO TO 350  00288
ISTN 0284  VLB(I)=SIB*SI*VLB(I)  00289
ISTN 0285  IF (IPRINT.GT.3.AND.NCON.GT.0) WRITE (I6I,1320) CT,CTL,PHI  00290
+VERSION 1.3.0 (01 MAY 80) CONMIN SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 82.141/10.53.56 PAGE 6

ISH 0285  VUB(I)=SIB*SI*VUB(I)  00291
ISH 0286  350 CONTINUE  00292
ISH 0287  IF (IPRINT.LT.4.OR.(NSCAL.LT.0.AND.ITER.GT.1)) GO TO 360  00293
ISH 0289  WRITE (I61,1330)  00294
ISH 0290  WRITE (I61,1460) (SCAL(I),I=1,NDV)  00295
ISH 0291  360 CONTINUE  00296
ISH 0292  NSCAL=MSCAL+1  00297
ISH 0293  NAC=0  00298
ISH 0294  END
ISH 0295  IF (INFDG.NE.1) GO TO 370  00299
ISH 0296  IGOTO=2  00300
ISH 0297  GO TO 950  00301
ISH 0300  370 CONTINUE  00302
ISH 0301  IGOTO=0  00303
ISH 0302  380 CONTINUE  00304
ISH 0303  CALL CNNOI (JGOTO,X,DF,G,ISC,IC,AGI,VLB,VUB,SCAL,C,NCAL,DX,DX1,FOO3101)
ISH 0304  IGOTO=3  00312
ISH 0305  IF (JGOTO.GT.0) GO TO 950  00313
ISH 0306  390 CONTINUE  00314
ISH 0307  INFO=1  00315
ISH 0308  IF (NAC.GE.N3) GO TO 810  00316
ISH 0309  IF (NSCAL.EQ.0.OR.NFDG.EQ.0) GO TO 420  00317
ISH 0310  C SCALE GRADIENT OF OBJECTIVE FUNCTION.  00318
ISH 0311  DO 400 I=1,NDV  00319
ISH 0312  DF(I)=DF(I)*SCAL(I)  00320
ISH 0313  IF (INFDG.EQ.2.OR.NAC.EQ.0) GO TO 420  00321
ISH 0314  SCAL(I)=SCAL(I)*SC(J)  00322
ISH 0315  DO 410 J=1,HAC  00323
ISH 0316  AI(JI)=AI(JI)*SC(J)  00324
ISH 0317  END
ISH 0318  DO 410 J=1,HAC  00325
ISH 0319  END
ISH 0320  410 A(JI)=A(JI)*SC(J)  00326
ISH 0321  420 CONTINUE  00327
ISH 0322  IF (IPRINT.LT.3.OR.NCON.EQ.0) GO TO 470  00328
ISH 0323  PRINT  00329
ISH 0324  C PRINT ACTIVE AND VIOLATED CONSTRAINT NUMBERS.  00330
ISH 0325  M1=0  00331
ISH 0326  M2=NS  00332
ISH 0327  IF (NAC.EQ.0) GO TO 450  00333
ISH 0328  DO 440 J=1,HAC  00334
ISH 0329  J=J+1  00335
ISH 0330  IF (J.GT.NCON) GO TO 440  00336
ISH 0331  GI=G(J)  00337
ISH 0332  CI=CTAN  00338
ISH 0333  CI=CTAN  00339
ISH 0334  IF (ISC(J).GT.0) CI=CTAN  00340
ISH 0335  GI=GI-CI  00341
ISH 0336  IF (GI.GT.0.) GO TO 430  00342
ISH 0337  C ACTIVE CONSTRAINT.  00343
ISH 0338  M1=M1+1  00344
ISH 0339  NS1(M1)=J  00345
ISH 0340  END

C OBTAIN GRADIENTS OF OBJECTIVE AND ACTIVE CONSTRAINTS
C

C SCALE GRADIENTS
C

C SCALE GRADIENT OF OBJECTIVE FUNCTION.

DO 400 I=1,NDV  00318
400 DF(I)=DF(I)*SCAL(I)  00319
C SCALE GRADIENTS OF ACTIVE CONSTRAINTS.

DO 410 J=1,HAC  00320
410 AI(JI)=AI(JI)*SC(J)  00321
C

C PRINT
C

C PRINT ACTIVE AND VIOLATED CONSTRAINT NUMBERS.
ISH 0341 GO TO 440
ISH 0342 430 M2=M2+1
ISH 0343 C VIOLATED CONSTRAINT.
ISH 0344 440 CONTINUE
ISH 0345 450 H3=H2-H3
ISH 0346 WRITE (I6I,1060) M1
ISH 0347 430 t12=N2+1
ISH 0348 C VIOLATED CONSTRAINT.
ISH 0349 WRITE (I6I,1070)
ISH 0350 WRITE (I6I,1480) (MS1(I),I=1,M1)
ISH 0351 460 WRITE (I6I,1080) M3
ISH 0352 IF (M3.EQ.0) GO TO 470
ISH 0353 M3=N3+1
ISH 0354 WRITE (I6I,1480) (MS1(I),I=M3,H2)
ISH 0355 470 CONTINUE
ISH 0356 WRITE (I6I,1070) 00357
ISH 0357 470 CONTINUE
C CALEULATE GRADIENTS OF ACTIVE SIDE CONSTRAINTS
C
ISH 0358 IF (HSIDE.EQ.0) GO TO 530
ISH 0359 MCNI=MCNI+1
ISH 0360 M1=0
ISH 0361 DO 510 I=1,NDV
ISH 0362 C LOWER BOUND.
ISH 0363 XI=X(I)
ISH 0364 XID=VLB(I)
ISH 0365 X12=ABS(XID)
ISH 0366 IF (X12.LT.1.) X12=1.
ISH 0367 GI=(XID-XI)/X12
ISH 0368 IF (GI.LT.-1.0E-6) GO TO 490
ISH 0369 M1=M1+1
ISH 0370 MSI(M1)=I
ISH 0371 NAC=NAC+1
ISH 0372 IC(NAC)=NCH1
ISH 0373 G(MCN1)=GI
ISH 0374 IF (MCNI.EQ.0) GO TO 810
ISH 0375 MCNI=MCNI+1
ISH 0376 DO 490 J=1,NDV
ISH 0377 490 A(J,HAC)=0.
ISH 0378 A(I,HAC)=-1.
ISH 0379 480 A(J,NAC)=-1.
ISH 0380 IC(NAC)=MCNI
ISH 0381 G(MCN1)=GI
ISH 0382 ISC(MCN1)=1
ISH 0383 510 CONTINUE
C UPPER BOUND.
ISH 0384 IF (MCNI.EQ.0) GO TO 810
ISH 0385 DO 480 J=1,NDV
ISH 0386 480 A(J,HAC)=0.
ISH 0387 490 A(I,HAC)=-1.
ISH 0388 480 A(J,NAC)=0.
ISH 0389 IC(NAC)=MCNI
ISH 0390 G(MCN1)=GI
ISH 0391 ISC(MCN1)=1
ISH 0392 CONTINUE
PRINT ACTIVE SIDE CONSTRAINT NUMBERS.

IF (IPRINT.LT.3) GO TO 530
WRITE (161,1090) HI
IF (H1.EQ.0) GO TO 530
WRITE (161,1100)
WRITE (161,1480) (HSl(J),J=I,Hll)
CONTINUE

PRINT GRADIENTS OF ACTIVE AND VIOLATED CONSTRAINTS.

IF (IPRINT.LT.4) GO TO 570
WRITE (161,1340)
DO 540 I=I,HDV,6
m=MINO(I,HDV,I+5) 540
WRITE (161,1010) I,(DF(J),J=I,M)
IF (HAC.EQ.O) GO TO 570
WRITE (161,1350)
DO 560 I=1,HAC
Hl=IC(I)
H2=HI-HCON
H3=0
IF (H2.GT.0) H3=IABS(HSl(H2))
IF (H2.LE.0) WRITE (161,990) HI
IF (H2.GT.0) WRITE (161,1000)
DO 550 K=1,NOV,6
m=MINO(HOV,K+5) 550
WRITE (161,1010) K,(A(J,I),J=K,M)
560
570
CONTINUE

DETERMINE SEARCH DIRECTION

ALP=1.0E+20
IF (NAC.GT.O) GO TO 580

UHCONSTRAINED FUNCTION

NVC=O
NFEAS=O
KCOUNT=KCOUNT+1
IF (KCOUNT.GT.ICNDIR) RESTART CONJUGATE DIRECTION ALGORITHM.
IF (KCOUNT.EQ.l) JOIR=O
IF JOIR =O FUIO DIRECTION OF STEEPEST DESCENT.
CALL CNIH02 (JDIR,SLOPE,OFTOF1,DF,S,N1) 630
GO TO 630
CONTINUE

CONSTRAINED FUNCTION

FINO USABLE-FEASIBLE DIRECTION.

KCOUNT=O
JDIR=O
PHI=10.*PHI
IF (PHI.GT.1000.) PHI=1000.
CALL CNIH05 (G,DF,A,S,B,C,SLOPE,PHI,ISC,IC,MSI,NVC,N1,N2,N3,N4,N5) 047
IF (IPRINT.LT.3) GO TO 660 00468
WRITE (161,1210) I,(A(NDV1,J),J=I,M1) 00472
WRITE (161,1210) S(NDV1) 00473
CONTINUE 00474
------------------------------------------------------------------00475
C ****************** ONE-DIMENSIONAL SEARCH ************************00476
------------------------------------------------------------------00477
IF (S(NDV1).LT.1.0E-6.AND.NVC.EQ.0) GO TO 710 00478
C FIHD ALPHA TO OBTAIN A FEASIBLE DESIGN 00480
C ------------------------------------------------------------------00481
IF (HVC.EQ.0) GO TO 630 00482
ALP=-I. 00483
DO 620 I=I,NAC 00484
NCI=IC(I) 00485
Cl=G(NCI) 00486
CTC=CTAM 00487
IF (ISC(CCI).GT.0) CTC=CTBM 00488
IF (Cl.LE.CTC) GO TO 620 00489
ALPL=0. 00490
DO 610 J=I,NDV 00491
ALPL=ALPL+S(J)*A(J,I) 00492
ALPL=ALPL*A(NDV2,I) 00493
IF (ABS(ALPL).LT.1.0E-20) GO TO 620 00494
ALPL=-CI/ALPL 00495
IF (ALPL.GT.ALP) ALP=ALPL 00496
620 CONTINUE 00497
630 CONTINUE 00498
C ------------------------------------------------------------------00499
C LIMIT CHANGE TO AOOBJ 00500
C ------------------------------------------------------------------00501
ALP1=1.0E+20 00502
DO 640 I=I,NDV 00513
SI=ABS(S(I)) 00514
XI=ABS(X(I)) 00515
IF (SI.LT.1.0E-10 .OR. XI.LT.0.1) GO TO 640 00516
ALPL=ALPHAX*XI/SI 00517
IF (ALPL.LT.ALP1) ALP1=ALPL 00518
640 CONTINUE 00519
IF (NVC.GT.0) ALP1=10.*ALP1 00520
IF (ALP1.LT.ALP) ALP=ALP1 00521
IF (ALP.GT.1.0E+20) ALP=1.0E+20 00522
IF (ALP.LE.1.0E-20) ALP=1.0E-20 00523
CONTINUE 00524
C ------------------------------------------------------------------00525
C LIMIT CHANGE IN VARIABLE TO ALPHAX 00526
C ------------------------------------------------------------------00527
ALP1=1.0E+20 00528
DO 600 I=I,NDV 00539
SI=ABS(S(I)) 00540
XI=ABS(X(I)) 00541
IF (SI.LT.1.0E-10 .OR. XI.LT.0.1) GO TO 600 00542
ALPL=ALPHAX*XI/SI 00543
IF (ALPL.LT.ALP1) ALP1=ALPL 00544
600 CONTINUE 00545
IF (NVC.GT.0) ALP1=10.*ALP1 00546
IF (ALP1.LT.ALP) ALP=ALP1 00547
IF (ALP.GT.1.0E+20) ALP=1.0E+20 00548
IF (ALP.LE.1.0E-20) ALP=1.0E-20 00549
CONTINUE 00550
C ------------------------------------------------------------------00551
C
DO ONE-DIMENSIONAL SEARCH FOR UNCONSTRAINED FUNCTION

JGOTO = 0

CONTINUE

CALL CN2002 (X,SLOPE,ALP,F,ALPHA,ALPHA2,ALPHA3,ALPHA4,APP,N1,NCAL,NCAP)

IGOTO = 4

IF (JGOTO GT 0) GO TO 950

JDIR = 1

GO TO 700

SOLVE ONE-DIMENSIONAL SEARCH PROBLEM FOR CONSTRAINED FUNCTION

JGOTO = 0

CONTINUE

CALL CN2006 (X,VLB,VLB,6,SCAL,OF,S,G1,G2,CTAH,CTAH,SLOPE,ALP,A1,A2,A3,A4,F1,F2,F3,F4,APP1,N1,NCAL,NCAP)

IGOTO = 5

IF (JGOTO GT 0) GO TO 950

IF (NAC.EQ.0) JDIR = 1

Print

UPDATE ALPHAX

ALP11 = 0.

DO 720 I = 1, NDV

SI = ABS(S(I))

XI = ABS(X(I))

IF (XI LT 1.0E-10) GO TO 720

ALP11 = ALP11 + SI / XI

IF (ALP11 GT 1.0) ALP11 = 1.0

ALP = 0.5 * ALP11 + (ALP11 - 1.0)

ALPHAX = ALP

NCOBJ = NCOBJ + 1

ABSOLUTE CHANGE IN OBJECTIVE.

OBJ1 = OBJ1 - OBJ

OBJ1 = ABS(OBJ1)

IF (OBJ1 LT 1.0E-10) OBJ1 = 0.

IF (NAC.EQ.0 OR OBJ1 = 0) NCOBJ = 0

IF (NCOBJ = 1) NCOBJ = 0

PRINT
C PRINT MOVE PARAMETER, NEW X-VECTOR AND CONSTRAINTS.

C

ISN 0570 IF (IPRINT.LT.3) GO TO 730
ISN 0571 WRITE (I61,1590) ALP
ISN 0572 IF (IPRINT.LT.2) GO TO 800
ISN 0573 IF (OBJB.GT.0.) GO TO 740
ISN 0574 IF (IPRINT.EQ.2) WRITE (I61,1400) ITER,OBJ
ISN 0575 IF (IPRINT.GT.2) WRITE (I61,1410) OBJ
ISN 0576 GO TO 760
ISN 0577 740 IF (IPRINT.EQ.2) GO TO 750
ISN 0578 WRITE (I61,1420) OBJ
ISN 0579 WRITE (I61,1430) ITER,OBJ
ISN 0580 DO 770 I=1,NDV
ISN 0581 IF (INSCAL.NE.0) FF1=SCAL(I)
ISN 0582 GII=FF1*XII
ISN 0583 DO 780 I=I,NDV,6
ISN 0584 M1=MH0(NDV,I+5)
ISN 0585 WRITE (161,1010) I,(GII),J=I,M1)
ISN 0586 IF (NCOFL.EQ.0) GO TO 800
ISN 0587 WRITE (161,1470)
ISN 0588 DO 790 I=1,NCON,6
ISN 0589 HII=MH0(NCON,I+5)
ISN 0590 WRITE (161,1010) I,(HII),J=I,M1)
ISN 0591 CONTINUE
C CHECK FEASABILITY
C
ISN 0592 IF(NCOFL.LE.0) GO TO 808
ISN 0593 DO 804 I=I,NCON
ISN 0594 CI=CTAM
ISN 0595 IF(ISCII).GT.0) CI=CTBM
ISN 0596 IF(GII).LE.Cl) GO TO 804
ISN 0597 NFEAS=NFEAS+1
ISN 0598 GO TO 806
ISN 0599 CONTINUE
ISN 0600 IF(NFEAS.GT.0) ABOBJ=.05
ISN 0601 NFEAS=0
ISN 0602 PHI=5.
ISN 0603 IF(HFEAS.GE.101 GO TO 810
ISN 0604 CONTINUE
C CHECK CONVERGENCE
C
ISN 0605 IF(NFEAS.LE.0) GO TO 808
ISN 0606 DO 804 I=1,NCON
ISN 0607 CI=CTAM
ISN 0608 IF(ISCII).LT.0) CI=CTBM
ISN 0609 IF(GII).LE.Cl) GO TO 804
ISN 0610 NFEAS=NFEAS+1
ISN 0611 GO TO 806
ISN 0612 CONTINUE
ISN 0613 804
ISN 0614 IF(NFEAS.GT.0) ABOBJ=.05
ISN 0615 NFEAS=0
ISN 0616 PHI=5.
ISN 0617 IF(NFEAS.LE.10) GO TO 810
ISN 0618 CONTINUE
ISN 0619 C STOP IF ITER EQUALS ITMAX.
ISN 0620 IF (ITER.GE.ITMAX) GO TO 810
ISN 0621 C ABSOLUTE CHANGE IN OBJECTIVE
ISN 0622 OBJB=ABS(OBJD)
ISN 0623 KOBJ=KOBJ+1
ISN 0624 IF (OBJB.GE.DABFUN.OR.NFEAS.GT.0) KOBJ=0
ISN 0625 C RELATIVE CHANGE IN OBJECTIVE
ISN 0626 AB.OBJ=ABS(OBJD)/ABS(OBJJ)
ISN 0627 IF (AB.OBJ.GT.1.0E-10) OBJD=OBJD/AB.OBJ
**ABOBJ=ABS(OBJD) 00645**
**IOBJ=IOBJ+1 00646**
**IF(NFEAS.GT.O.OR.OBJD.GE.DELFUH) IOBJ=O 00647**
**IF(IOBJ.GE.ITRH.OR.KOBJ.GE.ITRH) GO TO 810 00648**
**OBJ1=OBJ 00649**

---

**REDUCE CT IF OBJECTIVE FUNCTION IS CHANGING SLOWLY 00650**

---

**IF(IOBJ.LT.0R.NAC.EQ.0) GO TO 280 00653**

---

**CT=DCT*CT 00656**
**CTL=CTL*DCTL 00657**

---

**IF(ABS(CT).LT.CTHIN) CT=-CTMIN 00660**
**IF(ABS(CTL).LT.CTLMIN) CTL=-CTLMIN 00661**

---

**GO TO 280 00664**

---

**CONTINUE 00667**

---

**IF(NAC.GE.H3) WRITE(161,1490) 00670**

---

**FINAL FUNCTION INFORMATION 00673**

---

**UN-SCALE THE DESIGN VARIABLES. 00676**
**DO 820 I=1,HDV 00677**
**XI=SCALI(XI) 00678**

---

**IF(NSIDE.EQ.0) GO TO 820 00681**
**VLB(I)=XI*VLB(I) 00682**
**VUB(I)=XI*VUB(I) 00683**
**X(I)=XI*X(I) 00684**

---

**PRINT FINAL RESULTS 00687**

---

**IF(IPRINT.EQ.0.OR.NAC.GE.N3) GO TO 940 00690**
**WRITE (161,1500) OBJ 00693**
**WRITE(161,1420) OBJ 00696**
**WRITE(161,1450) OBJ 00699**
**DO 840 I=1,NDV,6 00702**
**M1=INO(NDV,1+5) 00705**
**WRITE (161,1010) I,(X(J),J=I,M1) 00708**

---

**IF(NCON.EQ.0) GO TO 900 00711**
**WRITE (161,1470) N1=INO(NCON,1+5) 00714**
**WRITE (161,1010) I,(G(J),J=I,N1) 00717**

---

**DETERMINE WHICH CONSTRAINTS ARE ACTIVE AND PRINT. 00720**
**NAC=0 00723**
**NVC=0 00726**
**DO 870 I=1,NCON 00729**
**CTA=CTAM 00732**
**IF(ISCI(I).GT.0) CTA=CTEM 00735**
**GI=GI(I) 00738**

---

**IF(GI.GT.CTA) GO TO 860 00741**
**IF(GI.LT.CTA.AND.ISCI(I).EQ.0) GO TO 870 00744**
**IF(GI.LT.CTA.LAND.ISCI(I).GT.0) GO TO 870 00747**
**NAC=NAC+I 00750**
**ICIN./;C)=I 00753**
**NVC=NVC+1 00756**
**MSlCNVC)=I 00759**
**WRITE (161,1060) NAC 00762**

---

**PRINT 00765**

---

**IF(ISCI(I).LT.0) CTA=CTAM 00768**
**IF(GI.GT.CTA) GO TO 860 00771**
**IF(GI.LT.CTA.AND.ISCI(I).EQ.0) GO TO 870 00774**
**IF(GI.LT.CTA.LAND.ISCI(I).GT.0) GO TO 870 00777**
**NAC=NAC+I 00780**
**ICIN./;C)=I 00783**
**NVC=NVC+1 00786**
**MSlCNVC)=I 00789**
**WRITE (161,1060) NAC 00792**

---

**CONTINUE 00795**

---

**WRITE (161,1060) NAC 00800**
IF (NAC.EQ.0) GO TO 880

WRITE (161,1480) (IC(J),J=1,NAC)

WRITE (161,1070) NAC

IF (HVC.EQ.0) GO TO 890

WRITE (161,1480) (HS1(J),J=1,NVC)

WRITE (161,1070) NVC

CONTINUE

IF (NFEAS.EQ.0) GO TO 930

C DETERMINE WHICH SIDE CONSTRAINTS ARE ACTIVE AND PRINT.

NAC=0

DO 920 I=1,NBV

XI=X(I)

XID=VUB(I)

X12=ABS(XI)

IF (X12.LT.1.) X12=1.

GI=(XID-XI)/X12

IF (GI.LT.-1.0E-6) GO TO 910

HAC=NAC+1

HSl(NAC)=-1

910

CONTINUE

WRITE (161,1100)

WRITE (161,1480) (HSl(J),J=1,NAC)

930

WRITE (161,1100)

IF (ITER.GE.ITHAX) WRITE (161,1160)

IF (NCON.GT.0.AND.NFDG.EQ.11) WRITE (161,1540)

C RE-SET BASIC PARAMETERS TO INPUT VALUES

ITR=IDT1

ITHAX=IOH2

ICON=IDH3

DABF=ODH2

C=DAT3

CT=DAT4

C=DAT5

IF (NCON.GT.0.AND.NFDG.EQ.11) WRITE (161,1540)
I

IN

UN

+VERSION 1.3.0 (01 MAY 80) CONMIN SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 02.141/10.53.56  PAGE 14

ISN 0762  FDCIOM=DM10  00763
ISN 0763  ADOUSI=ON11  00764
ISN 0764  ALPHAEXPDM12  00765
ISN 0765  IGOT0=0  00766
ISN 0766  CONTINUE  00767
ISN 0767  IF (NSCAL.EQ.0.OR.IGOTO.EQ.0) RETURN  00768
C UI=SCALE VARIABLES.
ISN 0769  DO 950 I=1,NDV  00770
ISN 0770   C(I)=X(I)  00771
ISN 0771   X(I)=X(I)*SCAL(I)  00772
ISN 0772   RETURN  00773
C --------------------------------------------------------
C FORMATS
C --------------------------------------------------------
ISN 0773  FORMAT (//,5X,30HCOMPLETELY UNCONSTRAINED FUNCTION WITH A LINEAR OBJECTIVE IS SPECIFIED//10X,6HMONOBJ =,IS10X,6HCON =,IS10X,6HINSIDE =,IS15/5X,35HCONTROL RETURNED TO CALLING PROGRAM)  00774
ISN 0774  FORMAT (//,5X,30HCONMIN HAS ACHIEVED A SOLUTION OF OBJ LESS THAN -00782
   11.0E+4//5X,35HREVENGE OF OPTIMIZATION 00783
ISN 0775  215 TERMINATED)  00784
ISN 0776  1000 FORMAT (5X,2HCONSTRAINT NUMBER,IS)  00785
ISN 0777  1010 FORMAT (3X,IS,1H),2X,6E13.5)  00786
ISN 0778  1020 FORMAT (/5X,35HLINEAR CONSTRAINT IDENTIFIERS IISC)/5X,36HNOZERO INDICATES LINEAR CONSTRAINT)  00787
ISN 0779  1030 FORMAT (3X,IS,1H),2X,6E13.5)  00788
ISN 0780  1040 FORMAT (5X,26HALL CONSTRAINTS ARE LINEAR)  00789
ISN 0781  1050 FORMAT (5X,26HALL CONSTRAINTS ARE NONLINEAR)  00790
ISN 0782  1060 FORMAT (5X,26HTHERE ARE,IS,19HACTIVE CONSTRAINTS)  00791
ISN 0783  1070 FORMAT (5X,26HCONSTRAINT NUMBERS ARE)  00792
ISN 0784  1080 FORMAT (5X,34HACTIVE SIDE CONSTRAINTS)  00793
ISN 0785  1090 FORMAT (5X,34HACTIVE SIDE CONSTRAINTS)  00794
ISN 0786  1100 FORMAT (5X,43HDECISION VARIABLES AT LOWER OR UPPER BOUNDS,30H IMIN00797
   15S INDICATES LOWER BOUND))  00798
ISN 0787  1110 FORMAT (/5X,21HDECISION VARIABLES AT LOWER OR UPPER BOUNDS,30H IMIN00797
   15S INDICATES LOWER BOUND))  00799
ISN 0788  1120 FORMAT (/5X,15HSEARCH//5X,15HINITIAL SLOPE =,E12.4,2X,8HPROP05ED ALPH0 =,E12.4)  00800
ISN 0789  1130 FORMAT (/5X,41H* CONMIN DETECTS VLB(I) =,E12.4/5X,41H* CONMIN DETECTS VUB(I) =,E12.4)  00801
ISN 0790  1140 FORMAT (/5X,21HTERMINATION CRITERION)  00802
ISN 0791  1150 FORMAT (/5X,43HMINSUM(I,J) =,E12.4,2X,8HSUM(I,J) =,E12.4)  00803
ISN 0792  1160 FORMAT (10X,15HITER EQUALS ITMAX)  00804
ISN 0793  1170 FORMAT (10X,62HTEN CONSECUTIVE ITERATIONS FAILED TO PRODUCE A FEASIBLE DESIGN)  00805
ISN 0794  1180 FORMAT (10X,43HABS(1-CBJ(I,J)-OBJ(I,J)) LESS THAN DELFUN FOR,IS,11H ITERATIONS)  00806
ISN 0795  1190 FORMAT (10X,43HABS(OBJ(I,J)-OBJ(I,J-1)) LESS THAN DABFUN FOR,IS,11H ITERATIONS)  00807
ISN 0796  1200 FORMAT (/5X,25HNUMBER OF ITERATIONS =,IS)  00808
ISN 0797  1210 FORMAT (/5X,20HCONSTRAINT PARAMETER, BET0 =,E14.5)  00809
ISN 0798  1220 FORMAT (1H1,///12X,27(1H*),12X,1H*,51X,1H*/12X,1H*,20X,1H* CO N 00810
   1M I N12X,1H*12X,1H*,51X,1H*/12X,1H*,15X,21H FORTRAN PROGRAM FOR 00811
   2.15X,1H*/12X,1H*,51X,1H*/12X,1H*,9X,33HCONSTRAINED FUNCTION MINIM00812
   15
C DATA SET U477COPE01 AT LEVEL 003 AS OF 05/17/81 00001
C DATA SET U477COPE01 AT LEVEL 002 AS OF 05/13/81 00002
C DATA SET U477COPE01 AT LEVEL 001 AS OF 02/13/81 00003
C DATA SET U477COPE01 AT LEVEL 001 AS OF 07/10/80 00004
C SUBROUTINE COPE01 (RA,IA,NORA,NIA) 00005
C CALL COPE03/ /GNOP3/,NCALC,IG3OBJ:NSV,NSOBJ:NCA,N2VX,N2VY,M00006
C 1,ABOBJ,THETA,OBJ,NOV,NCON,NSIDE,IPRINT,NFDG,NSC1,ITMC00007
C 2RM,ICDNDR,ICO3OBJ,INFO,INFOG,ITER 00008
C C 3.NAN2,HAN3,NP2AX,NP3AX,JNOH,MAXTRM 00009
C ENDI/1H/E/,END2/1HHI,END3/1HDI 00010
C DATA ENDU/1HE/,END2/1HHI,END3/1HDI 00011
C**************************************************************00012
C ROUTINE TO READ CONTROL INPUT FOR COPES. 00013
C**************************************************************00014
C BY G. H. VANDERPLAATS 00015
C NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF. 00016
C**************************************************************00017
C**************************************************00018
C ROUTINE TO READ CONTROL INPUT FOR COPES. 00019
C**************************************************************00020
C C**************************************************************00021
C**************************************************************00022
C**************************************************************00023
C**************************************************************00024
C**************************************************************00025
C**************************************************************00026
C**************************************************************00027
C**************************************************************00028
C**************************************************************00029
C**************************************************************00030
C**************************************************************00031
C**************************************************************00032
C**************************************************************00033
C**************************************************************00034
C**************************************************************00035
C**************************************************************00036
C**************************************************************00037
C**************************************************************00038
C**************************************************************00039
C**************************************************************00040
C**************************************************************00041
C**************************************************************00042
C**************************************************************00043
C**************************************************************00044
C**************************************************************00045
C**************************************************************00046
C**************************************************************00047
C**************************************************************00048
C**************************************************************00049
C**************************************************************00050
C**************************************************************00051
C**************************************************************00052
C**************************************************************00053
C**************************************************************00054
C**************************************************************00055
C**************************************************************00056
C**************************************************************00057
C**************************************************************00058
C**************************************************************00059
C**************************************************************00060
C**************************************************************00061
C**************************************************************00062
C**************************************************************00063
C**************************************************************00064
C**************************************************************00065
C**************************************************************00066
C**************************************************************00067
C**************************************************************00068
C**************************************************************00069
C**************************************************************00070
C**************************************************************00071
C**************************************************************00072
C**************************************************************00073
C**************************************************************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
C**************************************************************00114
C**************************************************************00115
C**************************************************************00116
C**************************************************************00117
C**************************************************************00118
C**************************************************************00119
C**************************************************************00120
C**************************************************************00121
C**************************************************************00122
C**************************************************************00123
C**************************************************************00124
DATA BLOCK A. READ (ISCR1,1190) (ATITLE(I),I=1,20)

C CONTROL PARAMETERS. C --- DATA BLOCK B. READ (ISCR1,1200) HCALC,NOV,HSV,N2VAR,HXAPRX,IPNPUT,IPDBG

IF (HCALC.LT.0.OR.NCALC.GT.61) WRITE 161,1220) HCALC
IF (HCALC.LT.0.OR.NCALC.GT.61) RETURN
IF (IPNPUT.GT.1) GO TO 100
WRITE (161,970) 970
WRITE (161,980) 980
WRITE (161,990) (ATITLE(I),I=1,20)

C --- DATA BLOCK C. READ (ISCR1,1200) IPRINT,ITMAX,ICNDIR,NASCAL,ITRM,LINOBJ,NACMX1,NFD0000

IF (IPNPUT.GT.0) GO TO 90
WRITE (161,870) 870
WRITE (161,880) 880
DO 80 I=1,ICARD
READ (ISCR2,590) NCARDS,(RA(I),J=1,80)
WRITE (161,890) NCARDS,(RA(J),J=1,80)
REWIND ISCR2
CONTINUE
WRITE (161,1000) (ATITLE(I),I=1,20)
WRITE (161,1010) NCALC,IIDV,NSV,H2VAR,HXAPRX,IPNPUT,IPDBG

END

C OPTIMIZATION INFORMATION

C OPTIMIZATION CONTROL VARIABLES. - COMMON DEPENDENT.

DATA BLOCK C. READ (ISCR1,1210) IPRINT,ITMAX,ICNDIR,NASCAL,ITRM,LINOBJ,NACMX1,NFD000110

C DATA BLOCK D.

READ (ISCR1,1210) FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,DELPHUN,00113
### Data Block E

- **ISN 0007**: READ (ISCR1, 920) NDVTOT, IOBJ, SGNOPT
- **ISN 0008**: IF (NDVTOT.LT.NDV) NDVTOT=NDV
- **ISN 0009**: IF (HCACI.EQ.6 .AND. HACXI.EQ.0) HACXI=2*NDV+2
- **ISN 0002**: IF (HACXI.LE.0) HACXI=NDV+2
- **ISN 0004**: IF (IP十八届.EQ.2) GO TO 110
- **ISN 0006**: IF (ABS(SGNOPT).LT.1.E-10) SGNOPT=-1.
- **ISN 0008**: WRITE (I6I, 1010) IOBJ, SGNOPT
- **ISN 0009**: WRITE (I6I, 760) IMPRINT, IMPHAX, IMORG, IMSCAL, INTRIB, LINDOG, HACXI, NFDG
- **ISN 0100**: WRITE (I6I, 1770) FDCH, FDCHN, CT, CTHII, CTL, CTLMIN, THETA, PHI, DELFUN, DAI

### Data Block F

- **ISN 0101**: N2=NDV+3
- **ISN 0102**: N3=N2+NDV+2
- **ISN 0103**: N4=N3+NDV+2

### Data Block G

- **ISN 0104**: IF (IP十八届.LT.2) WRITE (I6I, 1000)
- **ISN 0105**: N5=N4+NDV+2
- **ISN 0106**: IF (N5.LE.NDRA) GO TO 120
- **ISN 0109**: WRITE (I6I, 780)
- **ISN 0110**: Write (I6I, 790)
- **ISN 0111**: LOCRI(25)=N5
- **ISN 0112**: GO TO 550
- **ISN 0113**: NSIDE=0
- **ISN 0115**: DO 130 I=1,NDV
- **ISN 0116**: IF (RAIN2).GT.-1.0E+15 .AND. RAIN3).LT.1.0E+15) NSIDE=1
- **ISN 0117**: IF (RAIN2).LE.-1.0E+15) RAIN2)=-1.1E+15
- **ISN 0118**: IF (RAIN3).GE.1.0E+15) RAIN3)=1.1E+15
- **ISN 0119**: IF (IP十八届.LT.2) WRITE (I6I, 1090) I, RAIN2, RAIN3, RAIN4, TITLE(J), J=1,5
- **ISN 0120**: N2=N2+1
- **ISN 0121**: N3=N3+1
- **ISN 0122**: N4=N4+1
- **ISN 0128**: N2=N2+1
- **ISN 0129**: CONTINUE

### Data Block H

- **ISN 0129**: IF (IP十八届.LT.2) WRITE (I6I, 930)
- **ISN 0130**: N5=4*NDV+9
- **ISN 0131**: M2=NDV+1
- **ISN 0132**: M3=NDV+2
- **ISN 0133**: IF (RAIN2).LT.-1.0E+15 .AND. RAIN3).LT.1.0E+15) NSIDE=1
- **ISN 0134**: IF (RAIN2).LE.-1.0E+15) RAIN2)=-1.1E+15
- **ISN 0135**: IF (RAIN3).GE.1.0E+15) RAIN3)=1.1E+15
- **ISN 0136**: IF (IP十八届.LT.2) WRITE (I6I, 1090) I, RAIN2, RAIN3, RAIN4, TITLE(J), J=1,5
- **ISN 0137**: WRITE (I6I, 1780)
- **ISN 0138**: WRITE (I6I, 1790)
- **ISN 0139**: LOCRI(25)=N5
- **ISN 0140**: GO TO 550
- **ISN 0141**: NSIDE=0
- **ISN 0142**: DO 130 I=1,NDV
- **ISN 0143**: IF (RAIN2).LT.-1.0E+15 .AND. RAIN3).LT.1.0E+15) NSIDE=1
- **ISN 0144**: WRITE (I6I, 1090) I, RAIN2, RAIN3, RAIN4, TITLE(J), J=1,5
- **ISN 0145**: WRITE (I6I, 1790)
- **ISN 0146**: NSIDE=0
- **ISN 0147**: DO 130 I=1,NDV
- **ISN 0148**: IF (RAIN2).LT.-1.0E+15 .AND. RAIN3).LT.1.0E+15) NSIDE=1
- **ISN 0149**: WRITE (I6I, 1090) I, RAIN2, RAIN3, RAIN4, TITLE(J), J=1,5
- **ISN 0150**: CONTINUE

---

**Note:** The above code snippet contains FORTRAN language statements related to data block handling, optimization, and variable manipulation. The specific context and purpose of the code are not detailed within the snippet.
**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) IAH(M2),IA(I),RA(NS)
```

**ISN 0151**
```
IF (ABS(RA(NS)).LT.1.0E-20) RA(NS)=1.0E-20
```

**ISN 0155**
```
M2=M2+1
```

**ISN 0156**
```
NS=NS+1
```

**ISN 0157**
```
160 CONTINUE
```

**ISN 0158**
```
NCOH=0
```

**ISN 0159**
```
C --- DATA BLOCK H.
```

**ISN 0160**
```
C NUMBER OF CONSTRAINT SETS.
```

**ISN 0161**
```
READ (ISCR1,920) NCONS
```

**ISN 0162**
```
IF (INPUT.LT.2) WRITE (161,1100)
```

**ISN 0163**
```
H2=H2+1
```

**ISN 0164**
```
NS=NS+1
```

**ISN 0165**
```
160 CONTINUE
```

**ISN 0166**
```
NCOH=0
```

**ISN 0167**
```
C --- DATA BLOCK I.
```

**ISN 0168**
```
DO 240 I=1,NCONS
```

**ISN 0169**
```
IF (INPUT.LT.2) WRITE (161,1120)
```

**ISN 0170**
```
M6=2*NDVT+NDVTOT
```

**ISN 0171**
```
M4=2*NDVTOT+NCOH
```

**ISN 0173**
```
L=L+1
```

**ISN 0174**
```
C --- DATA BLOCK H.
```

**ISN 0175**
```
C NUMBER OF CONSTRAINT SETS.
```

**ISN 0176**
```
READ (ISCR1,920) ICONI,JCOHI,LCONI
```

**ISN 0177**
```
C LB, NORM, UB, NORM.
```

**ISN 0178**
```
READ (ISCR1,1200) (RA(I)),I=1,NNN
```

**ISN 0179**
```
IF (INPUT.LT.2) WRITE (161,1140)
```

**ISN 0180**
```
J1=0
```

**ISN 0181**
```
IF ( INPUT.LT.2 ) WRITE (161,1150)
```

**ISN 0182**
```
NVAR=NVAR-1
```

**ISN 0183**
```
DO 190 J=1,NVAR
```

**ISN 0184**
```
IA(1)=LCONI
```

**ISN 0185**
```
190 CONTINUE
```

**ISN 0186**
```
C HOW MANY CONSTRAINTS?
```

**ISN 0187**
```
IF (J1.EQ.0) GO TO 160
```

**ISN 0188**
```
C ADD LINEAR CONSTRAINT IDENTIFIERS TO ISC.
```

**ISN 0189**
```
DO 170 J=1,NCOH
```

**ISN 0190**
```
IA(1)=LCONI
```

**ISN 0191**
```
M4=M4+1
```

**ISN 0192**
```
M9=M9+1
```

**ISN 0193**
```
M1=M1+1
```

**ISN 0194**
```
C ADD LB, UB AND SCAL TO BLU IF NVAR.GT.1.
```

**ISN 0195**
```
IF (NVAR.GE.1) WRITE (161,1170)
```

**ISN 0196**
```
IF (NVAR.LT.1) NVAR=1
```

**ISN 0197**
```
NCOH=NCOH+1
```

**ISN 0198**
```
C --- DATA BLOCK H.
```

**ISN 0199**
```
C NUMBER OF CONSTRAINT SETS.
```

**ISN 0200**
```
READ (ISCR1,920) IAH(M2),IA(I),RA(NS)
```

**ISN 0201**
```
IF (ABS(RA(NS)).LT.1.0E-20) RA(NS)=1.0E-20
```

**ISN 0202**
```
C --- DATA BLOCK H.
```

**ISN 0203**
```
C NUMBER OF CONSTRAINT SETS.
```

**ISN 0204**
```
READ (ISCR1,920) IAH(M2),IA(I),RA(NS)
```

**ISN 0205**
```
IF (ABS(RA(NS)).LT.1.0E-20) RA(NS)=1.0E-20
```

**ISN 0206**
```
M2=M2+1
```

**ISN 0207**
```
M9=M9+1
```

**ISN 0208**
```
M1=M1+1
```

**ISN 0209**
```
C ADD LB, UB AND SCAL TO BLU IF NVAR.GT.1.
```

**ISN 0210**
```
IF (NVAR.GE.1) WRITE (161,1170)
```

**ISN 0211**
```
IF (NVAR.LT.1) NVAR=1
```

**ISN 0212**
```
C --- DATA BLOCK H.
```

**ISN 0213**
```
C NUMBER OF CONSTRAINT SETS.
```

**ISN 0214**
```
READ (ISCR1,920) IAH(M2),IA(I),RA(NS)
```

**ISN 0215**
```
IF (ABS(RA(NS)).LT.1.0E-20) RA(NS)=1.0E-20
```

**ISN 0216**
```
C --- DATA BLOCK H.
```

**ISN 0217**
```
C NUMBER OF CONSTRAINT SETS.
```

**ISN 0218**
```
READ (ISCR1,920) IAH(M2),IA(I),RA(NS)
```

**ISN 0219**
```
IF (ABS(RA(NS)).LT.1.0E-20) RA(NS)=1.0E-20
```

**ISN 0220**
```
C ADD LB, UB AND SCAL TO BLU IF NVAR.GT.1.
```

**ISN 0221**
```
IF (NVAR.GE.1) WRITE (161,1170)
```

**ISN 0222**
```
IF (NVAR.LT.1) NVAR=1
```

**ISN 0223**
```
C --- DATA BLOCK H.
```

**ISN 0224**
```
C NUMBER OF CONSTRAINT SETS.
```

**ISN 0225**
```
READ (ISCR1,920) IAH(M2),IA(I),RA(NS)
```

**ISN 0226**
```
IF (ABS(RA(NS)).LT.1.0E-20) RA(NS)=1.0E-20
```

**ISN 0227**
```
C --- DATA BLOCK H.
```

**ISN 0228**
```
C NUMBER OF CONSTRAINT SETS.
```

**ISN 0229**
```
READ (ISCR1,920) IAH(M2),IA(I),RA(NS)
```

**ISN 0230**
```
IF (ABS(RA(NS)).LT.1.0E-20) RA(NS)=1.0E-20
```

**ISN 0231**
```
C --- DATA BLOCK H.
```
ISt! 0220 RAIN6+6)=RAIN6+2)
ISH 0221 RAIN6+7)=RAIN6+3)
ISH 0222 N6=N5+4
ISH 0223 190 CONTINUE
ISH 0224 200 CONTINUE
C ADD CONSTRAINED VARIABLE GLOBAL IDENTIFIERS TO ICON.
ISH 0225 ICON1=ICONI
ISH 0226 M99=M9+NVAR-1
ISH 0227 IF (MM.GT.INDIA) GO TO 260
ISH 0229 DO 230 J=1,NVAR
ISH 0230 IF (J.EQ.1) GO TO 220
C SHIFT ISC VECTOR.
ISH 0232 L1=M4+1
ISH 0233 L2=M4
ISH 0234 DO 210 K=M4,M4
ISH 0235 IAI L1)=IAI L2)
ISH 0236 L1=L1-1
ISH 0237 210 L2=L2-1
ISH 0238 M4=M4+1
ISH 0239 M4=M4+1
ISH 0240 IA(M3)=ICONI
ISH 0241 ICON1=ICON1+1
ISH 0242 230 M3=M3+1
ISH 0243 IF (INPUT.LT.2) WRITE (161,l100) I,ICON1,CONI1,LCON1,RA(N5),RA(N6+100255)
ISH 0244 N6=N6+4
ISH 0246 L1=ICON1+1
ISH 0247 240 CONTINUE
ISH 0248 IF (INPUT.LT.2) WRITE (161,900) NCDNA
ISH 0249 GO TO 270
ISH 0250 WRITE (161,910)
ISH 0252 WRITE (161,920)
ISH 0253 LOC1(25)=tltlt1
ISH 0254 GO TO 550
ISH 0256 LOC1(25)=tltlt1
ISH 0258 GO TO 550
ISH 0259 270 CONTINUE
C STARTING LOCATIONS FOR APPROXIMATION INFORMATION.
ISH 0260 NAPR=4*NDV+NDVTOT+4*NCONA+9
ISH 0261 NAPR=2*(NDV+NCONA)+2*NDVTOT+NCONA+1
ISH 0262 NF0
ISH 0263 KMAX=0
ISH 0264 NPTOT=0
ISH 0265 MAXTRM=0
ISH 0266 IF (NXAPRX.LE.0) GO TO 450
C *------------------------------------------------------------------
C APPROXIMATE ANALYSIS/DESIGN
C *------------------------------------------------------------------
C DATA BLOCK J.
C CONTROL PARAMETERS.
ISH 0268 READ (ISCR1,1200) NF,NPS,NPF5,NPA,INCH,ISCRX,ISCRXF,IPAPRX
ISH 0269 IF (NPA NE.0) NPA=1
ISH 0271 IF (NPS.EQ.0.AND.NPFS.EQ.0) NPA=1
ISH 0273 IF (ISCRX.EQ.0) ISCRX=NPA
ISH 0276 IF (ISCRX.EQ.0) ISCRX=5
ISH 0277 IF (INPUT.LT.2) WRITE (161,960) HF,NPS,NPF5,NPA,INCH,ISCRX,ISCRXF,IPAPRX
ISH 0278 IF (INPUT.LT.2) WRITE (161,960) HF,NPS,NPF5,NPA,INCH,ISCRX,ISCRXF,IPAPRX
**VERSION 1.3.0 (01 MAY 80) COPE01 SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)**

```fortran
HPSFS=NPS+HPFS 00291
IF (HPSFS.LT.2) HPA=1 00292
NPTOT=HPS+HPFS+HPA 00293
IF (NPTOT.LT.2) HPTOT=2 00294
READ (ISCR1,1200) KMIN,KMAX,NMAX,JNOM,INXLOC,INFLOC,MAXTRM 00295
IF (INXLOC.EQ.0) NXAPRX=NDVTOT 00296
H=NXAPRX+1 00297
IF (NPMAX.LE.0) NPMAX=2*M 00298
IF (KMAX.EQ.0) KMAX=3*M-HPTOT+1 00299
IF (KMIN.EQ.0) KMIN=2*NDV-HPTOT+1 00300
IF (KMIN.LT.0) KHIN=0 00301
IF (KMAX.GT.0.AND.KMIN.EQ.0) KMIN=KHIN 00302
READ (ISCR1, 1210) (RAII),I=H7,HH7) 00316
IF (IPNPUT.LT.2) WRITE 1161,640) XFACT1,XFACT2 00347
```

---

**DATA BLOCK K, PART 1.**

```fortran
C DELX BOUNDS ON APPROXIMATE OPTIMIZATION. 00310
IF (NDV.LE.0) GO TO 290 00309
N7=NDV 00311
N7=N7+NDV-1 00312
IF (N7.LE.NDIA) GO TO 290 00313
WRITE (161,7801 00331
WRITE (161,630) LOCII251=MMS 00332
GO TO 550 00333
300 CONTINUE 00334
IF (IHLOC.EQ.0) GO TO 310 00335
READ (ISCR1,1200) (IAII)'I=M5,tlM5) 00337
310 CONTINUE 00338
C X-LOCATIONS ARE DEFAULTED TO DESIGN VARIABLE LOCATIONS. 00339
320 I=I,NXAPRX 00340
IA(MS)=IA(I) 00341
320 M5=MS+1 00342
330 CONTINUE 00343
```

---

**DATA BLOCK L.**

```fortran
C GLOBAL LOCATIONS OF X-VARIABLES. 00327
MS=MSI 00328
M5=MS+NXAPRX-1 00329
IF (M5.LE.NDIA) GO TO 300 00330
WRITE (161,7801 00331
WRITE (161,630) LOCII251=MMS 00332
GO TO 550 00333
305 CONTINUE 00334
306 CONTINUE 00335
IF (INXLOC.EQ.0) GO TO 310 00336
310 CONTINUE 00337
READ (ISCR1,1200) (IAII),I=MS,M5) 00338
320 CONTINUE 00339
330 CONTINUE 00340
C X-LOCATIONS ARE DEFAULTED TO DESIGN VARIABLE LOCATIONS. 00341
340 GO 320 I=1,NXAPRX 00342
IA(MS)=IA(I) 00343
340 M5=MS+1 00344
350 CONTINUE 00345
360 CONTINUE 00346
IF (IPNPUT.LT.2) WRITE (161,640) (IAII),I=M5,M5) 00347
```

---

**DATA BLOCK M.**

```fortran
C GLOBAL LOCATIONS OF FUNCTIONS. 00349
```

---

**DATA BLOCK N.**

```fortran
```

---

**DATA BLOCK O.**

```fortran
```

---

**DATA BLOCK P.**

```fortran
```

---

**DATA BLOCK Q.**

```fortran
```

---

**DATA BLOCK R.**

```fortran
```

---

**DATA BLOCK S.**

```fortran
```
**VERSION 1.3.0 (01 MAY 80) COPE01 SYSTEM/370 FORTRAN H EXTENDED (EISHANCED) DATE 82.141/10.54.12 PAGE 7**

**0350**

```plaintext
MS=NAPI+NXAPRX
```

**0351**

```plaintext
MS=MH+MF+1
```

**0352**

```plaintext
IF (MN6.LE.MHIA) GO TO 340
```

**0354**

```plaintext
WRITE (I61,780)
```

**0355**

```plaintext
WRITE (I61,650)
```

**0356**

```plaintext
LOCRI(25)=MH
```

**0357**

```plaintext
GO TO 550
```

**0358**

```plaintext
CONTINUE
```

**0359**

```plaintext
IF (INLOC.EQ.0) GO TO 350
```

**0361**

```plaintext
READ (ISCR1,1200) (IA(I),I=MH,MH6)
```

**0362**

```plaintext
GO TO 380
```

**0363**

```plaintext
CONTINUE
```

**0364**

```plaintext
C FUNCTION LOCATIONS ARE DEFAULTED TO OBJECTIVE AND CONSTRAINT
```

**0365**

```plaintext
NF1=1
```

**0366**

```plaintext
M3=MV+MVTOT+1
```

**0367**

```plaintext
IA(MH)=IOBJ
```

**0368**

```plaintext
IF (INCONA.EQ.0) GO TO 370
```

**0369**

```plaintext
DO 550 I=1,NCONA
```

**0370**

```plaintext
IF (IA(MS).EQ.IOBJ) GO TO 360
```

**0371**

```plaintext
IA(MH)=IA(MH6)
```

**0372**

```plaintext
GO TO 550
```

**0373**

```plaintext
M6=MH+1
```

**0374**

```plaintext
IA(MH)=IA(MH3)
```

**0375**

```plaintext
M3=MH3+1
```

**0376**

```plaintext
NF=NFS
```

**0377**

```plaintext
MS=NAPI+NXAPRX
```

**0378**

```plaintext
M6=MH+MF+1
```

**0379**

```plaintext
IF (INPUL.LT.2) WRITE (I61,660)
```

**0380**

```plaintext
C --- DATA BLOCK N.
```

**0381**

```plaintext
IF (INPUL.LT.2) WRITE (I61,1180) (IA(I),I=MH,MH6)
```

**0382**

```plaintext
C READ INPUT X-VECTORS AND STORE ON UNIT ISCR2.
```

**0383**

```plaintext
RETHIND ISCR2
```

**0384**

```plaintext
IF (NFS.EQ.0) GO TO 410
```

**0385**

```plaintext
N7=NAPR+NDV
```

**0386**

```plaintext
N77=N7+NXAPRX-1
```

**0387**

```plaintext
NF=1+NF+1
```

**0388**

```plaintext
M6=MH+MF+1
```

**0389**

```plaintext
IF (INPUL.LT.2) WRITE (I61,660)
```

**0390**

```plaintext
C --- DATA BLOCK O.
```

**0391**

```plaintext
C READ INPUT X-F PAIRS AND STORE ON UNIT ISCR2.
```

**0392**

```plaintext
IF (NFS.EQ.0) GO TO 440
```

**0393**

```plaintext
N7=NAPR+NDV
```

**0394**

```plaintext
N77=N7+NXAPRX-1
```

**0395**

```plaintext
C BINARY READ IF ISCRX.NE.5.
```

**0396**

```plaintext
C FORMATTED READ IF ISCRX.EQ.5.
```

**0397**

```plaintext
WRITE (I61,780)
```

**0398**

```plaintext
WRITE (I61,650)
```

**0399**

```plaintext
LOCAL(25)=N7
```

**0400**

```plaintext
GO TO 550
```

**0401**

```plaintext
CONTINUE
```

**0402**

```plaintext
IF (INPUL.LT.2) WRITE (I61,660)
```

**0403**

```plaintext
GO TO 440
```

**0404**

```plaintext
CONTINUE
```

**0405**

```plaintext
IF (INPUL.LT.2) WRITE (I61,710) I,I
```

**0406**

```plaintext
READ (ISCR2) (IA(IJ),I=N7,NH7)
```

**0407**

```plaintext
C --- DATA BLOCK O.
```

**0408**

```plaintext
READ (ISCR2) (R(AJ),J=N7,NH7)
```

**0409**

```plaintext
GO TO 440
```

**0410**

```plaintext
CONTINUE
```
IF (NN8.LE.N1RA) GO TO 420
WRITE (I61,780) NN8
SB =NN8+NF-1
GO TO 550
420 CONTINUE
IF (IPNPUT.LE.2) WRITE (I61,700) ISCRXF
GO TO 550
430 CONTINUE
440 CONTINUE
450 CONTINUE
NSOBJ=O
NSVTOT=O
C STARTING LOCATIONS FOR SENSITIVITY INFORMATION.
NSVR=N.SPR+NDV
NSVI=NAPI+APRX+NF
IF (NSV.LE.O) GO TO 500
C------------------------------------------------------------------
C SENSITIVITY INFORMATION
C------------------------------------------------------------------
IF (IPNPUT.LE.2) WRITE (I61,1020)
C --- DATA BLOCK P, PART 1.
C NSOBJ, IPSENS
READ (ISCR1,1200) NSOBJ,IPSENS
C --- DATA BLOCK P, PART 2.
C NSSENS.
M15=NSVI
M16=M15+NSOBJ-1
GO TO 460
WRITE (I61,810)
WRITE (I61,830)
LCI(25)=M15
GO TO 550
460 CONTINUE
READ (ISCR1,1200) (IA(I),I=M15,MM15)
C --- DATA BLOCK P, PART 3.
C NSSENS.
M15=NSVR
M16=NSVI+NSOBJ
C DATA BLOCK Q. PART 1.
C SENS, NSENS.
ISH 0460 READ (ISCR1,1200) IIA(H16),NN1
ISH 0461 NN1=NN1+NN1-1
ISH 0462 IF (NN1.LE.NDRA) GO TO 470
ISH 0464 WRITE (I61,790)
ISH 0465 WRITE (I61,840)
ISH 0466 LOCRI(25)=HH15
ISH 0467 GO TO 550
ISH 0468 470 CONTINUE
C DATA BLOCK Q. PART 2.
C SENS.
ISH 0469 READ (ISCR1,1210) (RA(J),J=NI5,NN1)
ISH 0470 IF (INPINPUT.GE.2) GO TO 480
ISH 0472 JJ=NI5+5
ISH 0473 IF (JJ.GT.NN15) JJ=NN15
ISH 0475 WRITE (I61,1040) IIA(H16),RA(J),J=NN15,JJ
ISH 0476 JJ=JJ+1
ISH 0477 IF (JJ.LE.NH15) WRITE (I61,1050) RA(J),J=JJ,NN15
ISH 0478 480 CONTINUE
ISH 0480 NSVTOT=NSVTOT+NN1
ISH 0481 IAI(MI7)=NN1
ISH 0482 NIS=NIS+1
ISH 0483 M16=M16+1
ISH 0484 M17=M17+1
ISH 0485 490 CONTINUE
ISH 0486 500 CONTINUE
ISH 0487 M2VX=0
ISH 0488 M2VY=0
C STARTING LOCATIONS FOR TWO-VARIABLE FUNCTION SPACE INFORMATION.
ISH 0489 N2VR=NSVR+NSVTOT
ISH 0490 N2VI=NSVI+NSOBJ+2*NSV
ISH 0492 IF (N2VAR.LE.0) GO TO 540
ISH 0494 ND2=MM20
ISH 0495 540 CONTINUE
C DATA BLOCK S.
C variable numbers and number of values of X and Y.
ISH 0496 READ (ISCR1,1200) N2VX,M2VX,H2VY,M2VY,IP2VAR
ISH 0498 N2O=N2VR
ISH 0499 N16=N16+1
ISH 0501 M2O=H2VI
ISH 0502 M120=M120+H2VAR-1
ISH 0503 IF (M120.LE.ND1A) GO TO 510
ISH 0505 WRITE (I61,840)
ISH 0506 WRITE (I61,850)
ISH 0507 LOCRI(25)=MM20
ISH 0508 GO TO 550
ISH 0509 510 CONTINUE
ISH 0510 550 CONTINUE
C DATA BLOCK T.
C GLOBAL VARIABLE NUMBERS CORRESPONDING TO FUNCTIONS OF X AND Y.
ISH 0511 READ (ISCR1,1200) (IA(I),I=H20,MI20)
ISH 0512 IF (IPINPUT.LT.2) WRITE (I61,1170) IP2VAR
ISH 0514 IF (IPINPUT.LT.2) WRITE (I61,1160) (IA(I),I=H20,MI20)
ISH 0516 GO TO 520
ISH 0518 520 CONTINUE
ISH 0520 530 CONTINUE
C VALUES OF X COMPONENTS.
ISH 0521 NN20=N20+M2VX-1
ISH 0522 IF (NN20.LE.NDRA) GO TO 520
ISH 0524
WRITE (161,780) 00527
WRITE (161,740) 00528
LOCR(25)=NH20 00529
GO TO 550 00530
520 READ (ISCR1,1210) (RA(I),I=N20,HN20) 00531
IF (INPUT.LT.12) WRITE 1161,1140) H2VX 00532
IF (INPUT.LT.2) WRITE (161,1150) RA(I),I=N21,HN21) 00533
C --- DATA BLOCK U.
C VALUES OF Y COMPONENTS.
N21=N20+N2VX 00534
NN21=N21+H2VY 00535
IF (NN21.LE.NDRA) GO TO 530 00536
WRITE (161,780) 00537
WRITE (161,750) 00538
LOCR(25)=NN21 00539
GO TO 550 00540
530 CONTINUE 00541
IN20=NN21 00542
READ (ISCR1,1210) (RA(I),I=N21,NN21) 00543
IF (INPUT.LT.12) WRITE (161,1150) N2VY 00544
IF (INPUT.LT.2) WRITE (161,1160) RA(I),I=N21,NN21) 00545
540 CONTINUE 00546
C ------------------------------------------------------------------
C DYNAMIC STORAGE ALLOCATION 00547
C ------------------------------------------------------------------
NDV2=NDV+2 00548
C REAL VARIABLES. 00549
LOCR(1)=1 00550
VLB. 00551
LOCR(2)=NDV+3 00552
VUB. 00553
LOCR(3)=LOCR(2)+NDV2 00554
SCAL. 00555
LOCR(4)=LOCR(3)+NDV2 00556
AJULT. 00557
LOCR(5)=LOCR(4)+NDV2 00558
BLU. 00559
LOCR(6)=LOCR(5)+NDVTOT 00560
DEIX. 00561
LOCR(7)=LOCR(6)+NDVTOT 00562
NCNA 00563
LOCR(8)=LOCR(7)+NDV 00564
SENS. 00565
LOCR(15)=LOCR(8) 00566
XHV. 00567
LOCR(20)=LOCR(15)+NSVTOT 00568
YH2. 00569
LOCR(21)=LOCR(20)+NSVTOT 00570
LOCR(21)=LOCR(20)+H2VX 00571
LOCR(22)=LOCR(21)+H2VY 00572
START OF EXECUTION STORAGE. 00573
LOCR(23)=LOCR(22) 00574
C INTEGER VARIABLES. 00575
IDSGN. 00576
LOCR(1)=1 00577
C NDSGN. 00578
LOC(2)=NDVTOT+1 00579
C ICON. 00580
LOC(3)=LOC(2)+NDVTOT 00581
C ISC.
C START OF EXECUTION STORAGE.
LOC(23)=LOC(21)
EXECUTION STORAGE REQUIREMENTS.
NRI=NDV
IF (NACMX1.GT.NRI) NRI=NACMX1
NRI2=NACMX1+2*NDV+NCON
NRI3=NSV
IF (NSOBJ.GT.NRI3) NRI3=NSOBJ
NRI4=N2VAR
NRI5=NR2+NR3
NRI6=NCONA+NXAPRX
NRI7=(KHAX+1-HPTOT1*NXAPRX+NF+1)
IF (NRI.LT.NR2) NRI=NR2
IF (NRI.LT.NRI2) NRI=NRI2
START OF TEMPORARY STORAGE.
LOC(24)=LOC(23)
LOC(25)=LOC(24)
LOC(26)=LOC(25)
LOC(27)=LOC(26)
LOC(28)=LOC(27)
LOC(29)=LOC(28)
LOC(30)=LOC(29)
LOC(31)=LOC(28)
LOC(32)=LOC(27)
LOC(33)=LOC(26)
LOC(34)=LOC(25)
LOC(35)=LOC(24)
LOC(36)=LOC(23)
LOC(37)=LOC(22)
LOC(38)=LOC(21)
LOC(39)=LOC(20)
LOC(40)=LOC(19)
LOC(41)=LOC(18)
LOC(42)=LOC(17)
LOC(43)=LOC(16)
LOC(44)=LOC(15)
LOC(45)=LOC(14)
LOC(46)=LOC(13)
LOC(47)=LOC(12)
LOC(48)=LOC(11)
LOC(49)=LOC(10)
LOC(50)=LOC(9)
LOC(51)=LOC(8)
LOC(52)=LOC(7)
LOC(53)=LOC(6)
LOC(54)=LOC(5)
LOC(55)=LOC(4)
LOC(56)=LOC(3)
LOC(57)=LOC(2)
LOC(58)=LOC(1)
LOC(59)=LOC(0)
CONTINUE
17X,1H,5X,15HSINGLE ANALYSIS/7X,1H,5X,12HOPTIMIZATION/7X,1H3,5X,00705
21HSENSITIVITY/7X,1H4,5X,27HVAR-VARIABLE FUNCTION SPACE/7X,1H5,5X,00706
31HOPTIMUM SENSITIVITY/7X,1H6,5X,24HAPPROXIMATE OPTIMIZATION 00707

FORMAT (210,F10.2) 00708

FORMAT (/5X,16HDESIGN VARIABLES/1X,5H0D.V.,5X,6HGLOBAL,4X,11HMUL00709
1TYPING/5X,21HD,5X,3HIND.,5X,4HRVAR. NO.,5X,6HFACITOR) 00710

FORMAT (217,5X,15,E12.5) 00711

FORMAT (5X,6,IS) 00712

FORMAT (/5X,3HPRINT CONTROL, IPSENS =,15/5X,34HNUMBER 00713
1OF SENSITIVITY OBJECTIVES =,15/5X,53HGLOBAL NUMBERS ASSOCIATED 00714
2TH SENSITIVITY OBJECTIVES) 00715

(1H1,1H,5X,47HCCCCCCCC 000000 P000000 EEEE00000 P000000 00716
1SSSSSS/5X,5,47HC 0 O P E S /5X,47HC00717
2HC O O P E S /5X,47HC00718
3 O PPPPPP EEE 00719
4 E 5/5X,47HC O O P 00720
5 S/5X,47HCCCCCCCC000000 PEEEEEE00721
6) 00722

FORMAT (/14X,21HC CONTROL PROGRAM/26X,5HF OR/8X00723
1.41HE ENGINEERING SYNTHESIS) 00724

FORMAT (/14X,9HT I T L E /5X,20A4) 00725

FORMAT (1H1,4X,6HTITLE;5X,20A4) 00726

FORMAT (/5X,19HC CONTROL PARAMETERS/5X,42HCALCULATION CONTROL, 00727
1 NCALC =,15/5X,42HNUMBER OF GLOBAL DESIGN VARIABLES, 00728
2NDY =,15/5X,42HNUMBER OF SENSITIVITY VARIABLES, NSV =,15/5X,4200729
3NUMBER OF FUNCTIONS IN TWO-SPACE, NVAR =,15/5X,42HNUMBER OF APP00730
4ROXIMATING VAR., NKAPPX =,15/5X,42HINITIAL INFORMATI00731
5E, IPINIT =,15/5X,42HDEBUG PRINT CODE, IFDBG 00732
6=,15) 00733

FORMAT (/5X,27H* SENSITIVITY INFORMATION) 00734

FORMAT (/14X,6HGLOBAL,4X,7HNUMER/5X,6HNUMBER,2X,8HVARIABLE,4X,00735
1VALUE,6X,18HOFF-HOMINAL VALUES) 00736

FORMAT (5X,4,15,5X,12.5,1X,5E11.4) 00737

FORMAT (3X,5E11.4) 00738

FORMAT (4F10.2,1OA4) 00739

FORMAT (/5X,28H* OPTIMIZATION INFORMATION/5X,35HGLOBAL VARI00740
1BLE NUMBER OF OBJECTIVE,10X,H=,15/5X,46HMULTIPLIER (NEGATIVE 00741
2CATES MINIMIZATION) =,15) 00742

FORMAT (/5X,52DVARIABLE INFORMATION/5X,56HWN-ZERO INITIAL 00743
1VALUE WILL OVER-RIDE MODULE INPUT/5X,5HD.IP. 5X,5HGLOBAL,10X,5HUPPER00744
2R,9X,7HINITIAL/5X,3HND.,7X,5HSOUND,10X,5HBOUND,10X,5HVARIABLE,10X,5H00745
3SCALE) 00734

FORMAT (5X,4X,6HGLOBAL,4X,7HNUMER/5X,6HNUMBER,2X,8HVARIABLE,4X,00735
1VALUE) 00746

FORMAT (5X,12.5,1X,5E11.4) 00747

FORMAT (3X,5E11.4) 00748

FORMAT (4F10.2,1OA4) 00749

FORMAT (/5X,9HTHERE ARE,13,16H CONSTRAINT SETS) 00750

FORMAT (5X,5HGLOBAL,2X,5HGLOBAL,2X,5HLINEAR,6X,5HGLOBAL,6X,13H00751
1NORMALIZATION,5X,5HUPPER,6X,13HNORMALIZATION,5X,2HID,3X,6HVARI. 1,2,6X00752
2VAR. 2,4X,2HID,8X,5HSOUND,9X,6HFACITOR,10X,5HBOUND,9X,6HFACITOR) 00753

FORMAT (/5X,4HGLOBAL VARIABLE NUMBER CORRESPONDING TO X, N2VX =,00754
115/5X,20HVARIABLES OF X-VARIABLE) 00755

FORMAT (/5X,56HGLOBAL VARIABLE NUMBER CORRESPONDING TO Y, N2VY =,00756
115/5X,20HVARIABLES OF Y-VARIABLE) 00757

FORMAT (3X,5E12.4) 00758

FORMAT (/5X,51H* THE-VARIABLE FUNCTION SPACE MAPPING INFORMAT00759
1ION/5X,23HPRINT CONTROL, IPVAR =,15/5X,52HGLOBAL VARIABLE HUME00760
2RS ASSOCIATED WITH F(X,Y), MVZ) 00756

FORMAT (5X,10IS) 00762
ISN 0703 1190 FORMAT (20A4) 00763
ISN 0704 1200 FORMAT (8I10) 00764
ISN 0705 1210 FORMAT (8F10.2) 00765
ISN 0706 1220 FORMAT (/'3X.26H* * INPUT ERROR, NCAlC =,I2,2X,4I1S LT.0 OR GT0766
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 NOFORMAT GOSTMT NOXREF NDALC NDANSF 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
DATA SET U477COPE02 AT LEVEL 001 AS OF 02/13/81

SUBROUTINE COPE02 (ARRAY,RA,IA,ARRAY,NDRA,NDIA)

DIMENSION ARRAY(AARRAY), RA(NDRA), IA(NDIA)

**********************************************************
ROUTINE TO CONTROL OPTIMIZATION.
**********************************************************

BY G. H. VANDERPLAATS

NASA-AIRES RESEARCH CENTER, MOFFETT FIELD, CALIF.

-----------------------------------------------

ARRAY DIMENSIONS

-----------------------------------------------

M1=NDV+2
N2=2*NOV+NCON
N3=NACMX1

IF (NDV.GT.HH4) HH4=HDV

N=2*NN4

N5=HH3

IF (NOV.GT.HH4) HH4=NOV

N6=HH4

N7=HCDA

-----------------------------------------------

ARRAY STARTING LOCATIONS

-----------------------------------------------

X, VLH, VUB, DF, A, S, G1, G2, C, B, SCAL, ISC, IC, MS1

-----------------------------------------------

SOURCE EBCDIC HOLIST NODDECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
CALL CONMIN (X,VLB,VUB,G,SCAL,D,F,A,S,G1,G2,B,C,ISC,IC,MS1,N1,N2,N3)
C *,NCt,N5)
CONTINUE
CALL CONMIN (RANX,RA(NLB),RANVUB),RA(NC),RA(NSCAL),RA(NDF),RA(NDG1,RA(NG1),RA(NG2),RA(NB),RA(NC),IA(NISC),IA(NIC),IA(NS1000)
C ANALYZE.
CALL COPE03 (ARRAY,NARRAY,RA(NX),RA(NH),RA(NMULT),RA(NNU),IA(NID),N1,NN2,NN3,NN4,NN5)
IF (IGOTO.GT.0) GO TO 10
RETURN
END
*OPTIONS IN EFFECT*NAME = OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOOBL(NONE)
*OPTIONS IN EFFECT*SOURCE ESCDIC NOLIST NODECK OBJECT NOHAP NOFORMAT GOSTIT NOXREF NOALC NOAHSF 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
SUBROUTINE COPE03 (ARRAY,NARRAY,G,AMULT,BLU,ISIGN,NSIGN,ICON,NH)  00003
COMMON /COPE3S/ SGNOPT,IOBJ,NSV,NCOA,NCNA,NCGX,NCGX,NSV,NSV,NSV,NSV  00004
12Y,N2VAR,IPSNS,IPESVAR,IPDBG,NACHB,NDVHT,LCR(25),LCT(25),ISCR(25),ISCR(25)  00005
21,LSG,LSAPRX,NPS,HPF3,HPF4,HPF4,HPF5,HPF5,HPF5,HPF5,HPF5,HPF5,HPF5,HPF5  00006
3,NSIGN,NSIGN,NSIGN,NSIGN,NSIGN,NSIGN,NSIGN,NSIGN,NSIGN,NSIGN,NSIGN  00007
COMMON /JOS/ 151, 161  00008
DIGIN = 150, 160  00009
DIMENSION Array(NARRAY), X(NN1), G(NN2), AMULT(NN3), BNU(4)  00010
DIMENSION NSIGN(NN1), NSIGN(NN2), ICON(NN7)  00011
***********************************************************************  00012
BUFFER BETWEEN COMMON AND COPE3 FUNCTION EVALUATION.  00013
********************************************  00014
BY G. N. VANDERPLAATS  00015
NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.  00016
INITIAL ANALYSIS HAS BEEN DONE. IF ITER = 0, GO EVALUATE  00017
OBJECTIVE AND CONSTRAINTS.  00018
IF (ITER.EQ.0) ITER=0  00019
IF (ITER.LE.1) GO TO 40  00020
-------------------------------------------------------------------  00021
PRINT OUTPUT IF DEBUG CONTROL IS TURNED ON.  00022
-------------------------------------------------------------------  00023
DEBUG OUTPUT AS REQUIRED.  00024
IF (IPDBG.LT.I) GO TO 20  00025
IF (ITER.EQ.ITER1) OR (ITER.LE.1) GO TO 20  00026
N5 = LOCRI(5)  00027
M2 = LOCRI(2)  00028
DO 10 I=1, NDVTOT  00029
CALL ANALIZ (ICALC)  00030
WRITE (161,70)  00031
ITER1=ITER  00032
X = XSAV2  00033
CONTINUE  00034
-------------------------------------------------------------------  00035
TRANSFER DESIGN VARIABLES TO USER ARRAY  00036
-------------------------------------------------------------------  00037
IF (N.GT.0) ARRAY(M) = X(N)*AMULT(I)  00038
CONTINUE  00039
ICALC = 3  00040
N3 = N3+1  00041
CALL ANALIZ (ICALC)  00042
WRITE (161,70)  00043
ITER=ITER  00044
X = XSAV2  00045
CONTINUE  00046
-------------------------------------------------------------------  00047
TRANSFER DESIGN VARIABLES TO USER ARRAY  00048
-------------------------------------------------------------------  00049
N5 = LOCRI(5)  00050
M2 = LOCRI(2)  00051
DO 30 I=1, NDVTOT  00052
CALL ANALIZ (ICALC)  00053
WRITE (161,70)  00054
ITER=ITER  00055
X = XSAV2  00056
CONTINUE  00057
-------------------------------------------------------------------  00058
C ISN 0040 ICAlC=2 00055
C ISN 0041 NANE=NANE+1 00057
C ISN 0042 CALL ANALIZ (ICALC) 00058
C ISN 0043 SAVE X(1). 00059
XSAV1=X(1) 00060
C ISN 0044 CONTINUE 00061
C ISN 0045 OBJ=-SGNOPT*ARRAY(IOBJ) 00062
C ISN 0046 IF (NCONA.EQ.0) RETURN 00063
C ISN 0047 OBJECTIVE 00064
C ISN 0048 4O CONTINUE 00065
C ISN 0049 OBJ=-SGNOPT*ARRAY(IOBJ) 00066
C ISN 0050 IF (NCONA.EQ.0) RETURN 00067
C ISN 0051 NCONA=ICONI) 00068
C ISN 0052 CC=ARRAY(NN) 00069
C CONSTRAINT VALUES 00070
C ISN 0053 BB=BLU(l,1) 00071
C ISN 0054 IF (BB.LT.-1.0E+15) GO TO 50 00072
C ISN 0055 C NORMALIZATION FACTOR. 00073
C ISN 0056 CI=BLU(2,1) 00074
C CONSTRAINT VALUE. 00075
C ISN 0057 NCONA=ICONI) 00076
C ISN 0058 G(N)=(BB-CC)/CI 00077
C ISN 0059 50 CONTINUE 00078
C ISN 0060 BB=BLU(3,1) 00079
C ISN 0061 CI=BLU(4,1) 00080
C ISN 0062 IF (BB.GT.1.0E+15) GO TO 60 00081
C FORMATS 00082
C ISN 0063 RETURN 00083
C ISN 0064 70 FORMAT (I1) 00084
C ISN 0065 0100 END 00085
C OPTIONS IN EFFECT *NAME(NAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOBEGIN(NONE)
C OPTIONS IN EFFECT *SOURCE EBCDIC NOLIST NODECK OBJECT NOAPI NOFORMAT GOSTMT NOXREF NOALC NODANSF TERM IBM FLAG(I)
C STATISTICS* SOURCE STATEMENTS = 68, PROGRAM SIZE = 1246, SUBPROGRAM NAME =COPE03
C STATISTICS* NO DIAGNOSTICS GENERATED
C *** END OF COMPILATION *** 3004K BYTES OF CORE NOT USED
C DATA SET U477COPE04 AT LEVEL 001 AS OF 02/13/81
C DATA SET 9166COPE04 AT LEVEL 001 AS OF 07/10/80

C SUBROUTINE COPE04 (ARRAY,NARRAY,SENS,NSENS,ISENS,NSENS,TMP,NNB)

C N4,NN10,RA,JA,IA,IA(IA)

C COMMON /COPES3/ TITLE(20)
C COMMON /COPES5/ SNOPT,HCALC,IDXV,NSV,JNO,NCRA,NPV,HDPV,NSV,JS
C COMMON /COPES7/ NSENS,NSW,NSN,CMP_MAX,HDPMAX,HDPMAX,NDV,NDV
C COMMON /COPES1/ TITLE(20)
C COMMON /COPES3/ SGTHM,HCALC,IDXV,NSV,JNO,NCRA,NPV,HDPV,NSV,JS
C COMMON /COPES7/ NSENS,NSW,NSN,CMP_MAX,HDPMAX,HDPMAX,NDV,NDV

C SOURCE EBBCD NOLIST NODECK OBJECT NAME NOFORMAT GOSTMT NOLIST TERM IBM FLRAG(I)

C ROUTINE TO PROVIDE SENSITIVITY INFORMATION WITH RESPECT TO
C A PRESCRIBED SET OF DESIGN VARIABLES.
C STORE OUTPUT ON UNIT ISCRI.
C
C TITLE.
C WRITE (ISCRI,3301) TITLE(I),I=1,20
C NCALC, NSV, NSOBJ
C WRITE (ISCRI,3401) NCALC,NSV,NSOBJ
C ISENS(I),I=1,NSV.
C WRITE (ISCRI,3401) ISENS(I),I=1,NSV.
C NSENSZ(I),I=1,NSOBJ.
C WRITE (ISCRI,3401) NSENSZ(I),I=1,NSOBJ.
C JCALC=3
C ICALC=2
C NDVSAV=NDV
C
C *************** NOMINAL ***************
C
C IF (NCALC.EQ.5) GO TO 10
C STANDARD SENSITIVITY.
C NANS=NANS+1
C CALL ANALIZ (JCALC)
C IF (IPSENS.GT.0) NANS=NANS+1
C IF (IPSENS.GT.0) CALL ANALIZ (JCALC)
C GO TO 130
C CONTINUE
C OPTIMUM SENSITIVITY.
C SAVE X, VLB, VUB AND SCAL IN TEMPORARY STORAGE.
C N=4+NDV+8
C L=LOCRI(I)
C DO 20 I=1,N
C RA(I)=RA(I)
C SAVE IDSIGN AND NDSGN IN TEMPORARY STORAGE.
C
N=2*NDVTOT
DO 30 I=1,N
IA(I)=IA(I-1)
L=L+1
C SHIFT DESIGN VARIABLE INFORMATION IF ANY SENSITIVITY VARIABLE IS
C ALSO A DESIGN VARIABLE.

M=LOC(I(2)
DO 40 J=1,NDVTOT
L=IA(J)
IF (L.EQ.N) GO TO 50
M=M2+1
C SENSITIVITY VARIABLE IS NOT A DESIGN VARIABLE.
GO TO 90
CONTINUE
SENSITIVITY VARIABLE IS ALSO A DESIGN VARIABLE.

M2=10C(21
DO 40 J=1,NDVTOT
L=IA(J)
IF (L.EQ.N) GO TO 50
M=M2+1
C SENSITIVITY VARIABLE IS NOT A DESIGN VARIABLE.
GO TO 90
CONTINUE
SENSITIVITY VARIABLE IS ALSO A DESIGN VARIABLE.

M2=10C(21
DO 40 J=1,NDVTOT
L=IA(J)
IF (L.EQ.N) GO TO 50
M=M2+1
C SENSITIVITY VARIABLE IS NOT A DESIGN VARIABLE.
GO TO 90
CONTINUE
SENSITIVITY VARIABLE IS ALSO A DESIGN VARIABLE.

M2=10C(21
DO 40 J=1,NDVTOT
L=IA(J)
IF (L.EQ.N) GO TO 50
M=M2+1
C SENSITIVITY VARIABLE IS NOT A DESIGN VARIABLE.
GO TO 90
CONTINUE
SENSITIVITY VARIABLE IS ALSO A DESIGN VARIABLE.
C WRITE NOMINAL RESULTS ON UNIT ISCR1
C
C SENS(I,I).  

C GLOBAL LOCATION OF SENSITIVITY VARIABLE.  

C NUMBER OF SENSITIVITY VARIABLES, NSINV.  
C WRITE ISNSV AND NSNSV-1 ON UNIT ISCR1.  
C WRITE ISNSV, NSNSV-1 ISNSV, NSNSV  
C IF (NSNSV.LE.1) GO TO 320  
C IF (NCALC.NE.51 GO TO 210  
C SAVX IS A DESIGN VARIABLE.  
C SAVE NSVH.NSVV AND SCAL FOR THIS DESIGN VARIABLE AND SHIFT  
C REMAINING VARIABLES.  
C SAVE.
+VERSION 1.3.0 (01 MAY 80) COPE04 SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 82.141/10.54.31 PAGE 4

I  N1U1'1VERSIOH 1.3.0

ISH 0127
SAL=RA(I)
K=K+NDV2
ISH 0128
ISH 0129
SHFT
K=K+NDV2
ISH 0131
SAV=RA(I)

ISH 0132
IF (ID1.GT.NDV) GO TO 190
ISH 0133
DO 180 I=ID1,NDV
RA(I)=RA(I+1)
ISH 0135
K=K+NDV2
ISH 0136
RA(K)=RA(K+1)
ISH 0137
K=K+NDV2
ISH 0138
RA(K)=RA(K+1)
ISH 0139
K=K+NDV2
ISH 0140
K=K+NDV2
ISH 0141
RA(K)=RA(K+1)
ISH 0142
SHFT
K=I+NDV2
ISH 0143
RA(K)=RA(K+1)
ISH 0144
K=I+NDV2
ISH 0145
RA(K)=RA(K+1)
ISH 0146
K=I+NDV2
ISH 0147
RA(K)=RA(K+1)
ISH 0148
K=I+NDV2
ISH 0149
RA(K)=RA(K+1)
ISH 0150
K=I+NDV2
ISH 0151
RA(K)=RA(K+1)
ISH 0152
K=I+NDV2
ISH 0153
K=I+NDV2
ISH 0154
K=I+NDV2
ISH 0155
K=I+NDV2
ISH 0156
K=I+NDV2
ISH 0157
K=I+NDV2
ISH 0158
K=I+NDV2
ISH 0159
K=I+NDV2
ISH 0160
K=I+NDV2
ISH 0161
K=I+NDV2
ISH 0162
K=I+NDV2
ISH 0163
K=I+NDV2
ISH 0164
K=I+NDV2
ISH 0165
K=I+NDV2
ISH 0166
K=I+NDV2
ISH 0167
K=I+NDV2
ISH 0168
K=I+NDV2
ISH 0169
K=I+NDV2
ISH 0170
K=I+NDV2
ISH 0171
K=I+NDV2
ISH 0172
K=I+NDV2
ISH 0173
K=I+NDV2
ISH 0174
K=I+NDV2
ISH 0175
K=I+NDV2
ISH 0176
K=I+NDV2
ISH 0177
K=I+NDV2
ISH 0178
K=I+NDV2
ISH 0179
K=I+NDV2
ISH 0180
K=I+NDV2
ISH 0181
K=I+NDV2
ISH 0182
K=I+NDV2
ISH 0183
K=I+NDV2
ISH 0184
K=I+NDV2
ISH 0185
K=I+NDV2
ISH 0186
K=I+NDV2
ISH 0187
K=I+NDV2
ISH 0188
K=I+NDV2
ISH 0189
K=I+NDV2
ISH 0190
K=I+NDV2
ISH 0191
K=I+NDV2
ISH 0192
K=I+NDV2
ISH 0193
K=I+NDV2
ISH 0194
K=I+NDV2
ISH 0195
K=I+NDV2
ISH 0196
K=I+NDV2
ISH 0197
K=I+NDV2
ISH 0198
K=I+NDV2
ISH 0199
K=I+NDV2
ISH 0200
K=I+NDV2
ISH 0201
K=I+NDV2
ISH 0202
K=I+NDV2
ISH 0203
K=I+NDV2
ISH 0204
K=I+NDV2
ISH 0205
K=I+NDV2
ISH 0206
K=I+NDV2
ISH 0207
K=I+NDV2
ISH 0208
K=I+NDV2
ISH 0209
K=I+NDV2
ISH 0210
K=I+NDV2
ISH 0211
K=I+NDV2
ISH 0212
K=I+NDV2
ISH 0213
K=I+NDV2
ISH 0214
K=I+NDV2
ISH 0215
K=I+NDV2
ISH 0216
K=I+NDV2
ISH 0217
K=I+NDV2
ISH 0218
K=I+NDV2
ISH 0219
K=I+NDV2
ISH 0220
K=I+NDV2
ISH 0221
K=I+NDV2
ISH 0222
K=I+NDV2
ISH 0223
K=I+NDV2
ISH 0224
K=I+NDV2
ISH 0225
K=I+NDV2
ISH 0226
K=I+NDV2
ISH 0227
K=I+NDV2
ISH 0228
K=I+NDV2
ISH 0229
K=I+NDV2
ISH 0230
K=I+NDV2
ISH 0231
K=I+NDV2

C MODIFY NDSGN.
M2=LOCRI(2)
DO 200 I=I,NDVTOT
IF (IA(M2).EQ.ID11 IA(M2)=0
IF (IA(M2).GT.ID11 IA(H21=IA(M2)-1
H2=M2+1
200 CONTINUE
C_____________________________________________________________________
VARY THE VALUE OF THE SENSITIVITY PARAMETER
C_____________________________________________________________________
NSVAL1=NSVAL1+1
NSVALN=NSVAL1
DO 280 JJ=2,NSENSV
NSVAL1=NSVAL1+1
ARRAYISENSV)=SENS(INSVAL1)
WRITE IISCRI,3S0 SENS(INSVAL1)
CANALIZE.
IF (NCALC.EQ.51 GO TO 220
C STANDARD SENSITIVITY.
NAN2=NAN2+1
CALL ANALIZ (ICALC)
IF (IPSENS.GT.0) NAN3=NAN3+1
IF (IPSENS.GT.0) CALL ANALIZ (JCALC)
GO TO 260
C O P T I M U M S E N S I T I V I T Y .
IF (NDV.EQ.NDVSAV) GO TO 240
SET LINKED DESIGN VARIABLE VALUES TO PRESCRIBED VALUE.
M2=LOCRI(2)
DO 230 I=1,NDVTOT
L=IA(I)
M5=LOCRI(5)+I-1
ARRAY(L)=ARRAY(ISENSV)*RA(M5)
230 M2=M2+1
240 CONTINUE
NAN2=NAN2+1
CALL ANALIZ (ICALC)
IF (NDV.LE.0) GO TO 250
CALL COPE02 (ARRAY,RA,IA,NARRAY,NDRA,NDIA)
250 CONTINUE
IF (IPSENS.GT.0) NAN3=NAN3+1
ISN 0106    IF (IPSENS.GT.0) CALL ANALIZ (JCALC)
ISN 0108   260   CONTINUE
            C WRITE SENSITIVITY RESULTS ON UNIT ISCRI
            C OBJZ.
            C
            ISN 0169    DO 270 I=1,NSOBJ
            ISN 0170     N=NSENSZ(I)
            ISN 0171   270   TEI1PII)=ARRAY(M)
            ISN 0172     WRITE (ISCRI,350) (TEMP(I),I=1,NSOBJ)
            ISN 0173   280   CONTINUE
            ISN 0174     ARZAYIISENSV)=SENSINSVALU
            ISN 0175     IF (NCALC.NE.5.0R.IDL.EQ.0) GO TO 320
            C RESTORE X, VLB, VUB AND SCAL.
            ISN 0197     NOV=NOVSAV
            ISN 0198     IF (IDL.EQ.NDV) GO TO 300
            ISN 0200    L=HOV-1
            ISN 0201     L=I
            ISN 0202    DO 290 I=ID1,L1
            ISN 0203     RA(L+1)=RA(L)
            ISN 0204     K=IDV+ID2
            ISN 0205     RA(K+1)=RA(K)
            ISN 0206     K=K-IDV2
            ISN 0207     RA(K+1)=RA(K)
            ISN 0208     K=K+IDV2
            ISN 0209     RA(K+1)=RA(K)
            ISN 0210   290   L=L-1
            ISN 0211     RA(ID1)=SAVX
            ISN 0212     K=ID1+IDV2
            ISN 0213     RA(K)=SAVL
            ISN 0214     K=K-IDV2
            ISN 0215     RA(K)=SAVU
            ISN 0216     K=K+IDV2
            ISN 0217     RA(K)=SAVS
            ISN 0218   300   CONTINUE
            C RESTORE NDSGN.
            ISN 0219     M2=LOC1(2)
            ISN 0220    DO 310 I=1,NDVTOT
            ISN 0221     IF (IA(M2).GE.ID1) IA(M2)=IA(M2)+I
            ISN 0223    IF (IA(M2).GE.ID1) IA(M2)=ID1
            ISN 0225   310   M2=M2+I
            ISN 0226   320   CONTINUE
            ISN 0227    RETURN
            C FORMAT (5EI5.8)
            C
            ISN 0228   330   FORMAT (20A4)
            ISN 0229   340   FORMAT (1615)
            ISN 0230   350   FORMAT (SE15.8)
            ISN 0231   END
*OPTIONS IN EFFECT=NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODEL(NOHE)
*OPTIONS IN EFFECT=SOURCE EBCDIC NOLIST NDECK OBJECT NCHAP NGFORMAT GOSTMT NDXRES 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
DATA SET U477COPEOS AT LEVEL 002 AS OF 03/16/81 00001
DATA SET U477COPEOS AT LEVEL 001 AS OF 02/13/81 00002
DATA SET 9183COPEOS AT LEVEL 001 AS OF 07/10/80 00003

SUBROUTINE COPEOS (RA,IA,NDRA,IA,ISCR1) 00004
COMMON /UOS/ ISI, I6I 00005
DIMENSION RA(NDRA), IA(NIA) 00006
******************************************************************************00007
ROUTINE TO PRINT SENSITIVITY INFORMATION STORED ON UNIT ISCR1. 00008
******************************************************************************00009
BY G. N. VANDERPLAATS JULY, 1974. 00010
NASA-JAMES RESEARCH CENTER, MOFFETT FIELD, CALIF. 00011

-----------------------------------------------------------------------------00012
GENERAL INFORMATION 00013
-----------------------------------------------------------------------------00014
TITLE. 00015
READ (ISCR1,70) (RA(I),I=1,20) 00016
NCALC, NSV, NSOBJ 00017
READ (ISCR1,80) NCALC,NSV,NSOBJ 00018
IF (NCALC.EQ.3) WRITE (161,90) 00019
WRITE (161,100) NSV,NSOBJ 00020
READ (ISCR1,80) (IA(I),I=1,NSV) 00021
WRITE (161,110) 00022
WRITE (161,120) (IA(I),I=1,NSV) 00023
IRENSII),I=1,NSOBJ. 00024
READ (ISCR1,140) (RA(I),I=1,NSOBJ) 00025
WRITE (161,130) 00026
WRITE (161,140) (RA(I),I=1,NSOBJ) 00027
-----------------------------------------------------------------------------00028
NOMINAL INFORMATION 00029
-----------------------------------------------------------------------------00030
SENS(I),I=1,NSV. 00031
READ (ISCR1,140) (RA(I),I=1,NSV) 00032
WRITE (161,150) 00033
WRITE (161,160) (RA(I),I=1,NSV) 00034
OBJJ(I)=1,NSOBJ. 00035
READ (ISCR1,140) (RA(I),I=1,NSOBJ) 00036
WRITE (161,170) 00037
WRITE (161,180) (RA(I),I=1,NSOBJ) 00038
-----------------------------------------------------------------------------00039
SENSITIVITY INFORMATION 00040
-----------------------------------------------------------------------------00041
DO 40 ISENS=1,NSV 00042
ISENSI, NSENSI 00043
READ (ISCR1,80) ISENSI,NSENSI 00044
WRITE (161,190) ISENSI 00045
IF (NSENSI.EQ.0) WRITE (161,200) 00046
WRITE (161,210) 00047
DO 30 J=1,NSENSI 00048
C SENSI(J). 00049
ISN 0037      READ (ISCR1,140) SENSIJ
               C
             0055               C
ISN 0038      READ (ISCR1,140) (RA(I),I=1,NSOBJ)
ISN 0039      N=NSOBJ/14
ISN 0040      WRITE (161,210) SENSIJ,(RA(I),I=1,N)
ISN 0041      N=(NSOBJ-1)/4
ISN 0042      IF (N.LT.1) GO TO 20
ISN 0044      L1=L1+5
ISN 0045      DO 10 I=1,N
               L2=L1+3
               L2=MIN0(L2,NSOBJ)
               WRITE (161,220) (RA(J),J=L1,L2)
               10
ISN 0049      10
ISN 0050      20 CONTINUE
ISN 0051      30 CONTINUE
ISN 0052      40 CONTINUE
ISN 0053      RETURN
               C
               C
               C
               C
ISN 0054      50 FORMAT (IH,4X,46HOPTIMUM SENSITIVITY ANALYSIS RESULTS (NCALC=5)) 0075
ISN 0055      60 FORMAT (4X,20A4)
ISN 0056      70 FORMAT (20A4)
ISN 0057      80 FORMAT (16I8)
ISN 0058      90 FORMAT (11I,4X,47HSTANDARD SENSITIVITY ANALYSIS RESULTS (NCALC=3)) 0079
ISN 0059      100 FORMAT (/5X,36HNUMBER OF SENSITIVITY VARIABLES, NSV,9X,1H=,IS/SX, 0050
               13HNUMBER OF SENSITIVITY OBJECTIVES, NSOBJ,4X,1H=,IS)
               0081
ISN 0060      110 FORMAT (/5X,52HGLOBAL NUMBERS ASSOCIATED WITH SENSITIVITY VARIABLES) 0032
               0083
ISN 0061      120 FORMAT (5X,10I5)
ISN 0062      130 FORMAT (/5X,53HGLOBAL NUMBERS ASSOCIATED WITH SENSITIVITY OBJECTIVES) 0085
               15X)
               0036
ISN 0063      140 FORMAT (5E15.8)
ISN 0064      150 FORMAT (/5X,26HGLOBAL NUMERICAL DESIGN INFORMATION,6X,3IVALUES OF SENSITIVITY VARIABLES) 0033
               15ITY VARIABLES)
               0089
ISN 0065      160 FORMAT (5X,5E13.5)
ISN 0066      170 FORMAT (/5X,4IVALUES OF SENSITIVITY OBJECTIVE FUNCTIONS)
               0091
ISN 0067      180 FORMAT (/5X,2CHSENSITIVITY ANALYSIS RESULTS)
               0092
ISN 0068      190 FORMAT (/5X,15HGLOBAL VARIABLE,IS//10X,1HX,20X,4HFF(X))
               0093
ISN 0069      200 FORMAT (/5X,35THENOMINAL VALUE IS THE ONLY VALUE/5X,27HSPECIFIED0094
               1 FOR THIS VARIABLE)
               0095
ISN 0070      210 FORMAT (/5X,1E12.4,3X,4E13.4)
ISN 0071      220 FORMAT (10X,4E13.4)
ISN 0072      END
               C
               C
               C
               C
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOGBLINGONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOCP MTFORMAT COSTNT NOREF NOCAL MDSN P 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
SUBROUTINE COPE06 (ARRAY,RA,IA,HARRAY,NDRA,NDIA)  
COMMON /COPES1/ TITLE(20),  
COMMON /COPES2/ SGUOPT,NCALC,ISGJ,NSV,NSOBJ,HCNONA,N2VX,M2VX,NSVY,H2VY,HN2VY,  
12VY,NEVAR,IPSENS,IP2VAR,IPBO,HIINF,H2INF,NV0TOT,LCOCR(25),LCOCR(25),ISCRI0005  
21,ISC2,IXAFRX,NP,SNP,FN,FN,INON1,IPAFRX,H2INF,KMIN,KMAX,XFACT1,XFACT200006  
3,NH2,NH3,NH4,NVPTOT,JH2,H2TR,  

c !SH0002  
ISH 0003  
ISH 0004  

c C ROOT ROUTINE TO CALCULATE FUNCTIONS OF TWO DESIGN  
C VARIABLES FOR ALL COMBINATIONS OF A SET OF PRESCRIBED VALUES OF THESE VARIABLES.  
C BY G. N. VANDERPLAATS AUG., 1974.  
C NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.  

c UNIT ISCR1 WRITE  
------------------------------------------------------------------  
WRITE (ISCR1,40) TITLE(I),I=1,20)  
WRITE (ISCR1,50) NeAlC,N2VAR,M2VX,H2VY,M2VY,N2VZ.  
WRITE (ISCR1,50) M20=lOCR(20)  
WRITE (ISCR1,50) M21=lOCR(21)-1  
------------------------------------------------------------------  
TWO-VARIABLE FUNCTION SPACE  
------------------------------------------------------------------  
ICAlC=2  
KCAlC=3  
IGN=1  
N20=lOCR(20)  
N21=lOCR(21)-1  
DO 20 J=1,H2VY  
N21=N21+SIGN  
20  ARRAY(N2VY)=RAIN21)  
ANAlIZE.  
NAN2=NAN2+1  
CALL ANAlIZ (ICAlC)  
IF (IP2VAR.GT.0) CALL ANAliz (KCAlC)  
IF (IP2VAR.GT.0) NAN3=NAN3+1  
------------------------------------------------------------------  
UNIT ISCR1 WRITE  
------------------------------------------------------------------  
WRITE X, Y, VALUES.  
N=IA(H20)  
WRITE (ISCR1,60) RA(N20),RA(N21)  
F(X,Y) VALUES.
ISN 0034  RA(N24)=ARRAY(N)
ISN 0035  N24=N24+1
ISN 0036  M20=M20+1
ISN 0037  10 CONTINUE
ISN 0038  N24=N24+N2VAR-1
ISN 0039  WRITE (ISCR1,60) (RA(K),K=N23,N24)
ISN 0040  20 CONTINUE
ISN 0041  N21=N21+ISIGN
ISN 0042  N20=N20+1
ISN 0043  ISIGN=-ISIGN
ISN 0044  30 CONTINUE
ISN 0045  RETURN

C------------------------------------------------------------------
C FORMATS
C------------------------------------------------------------------
C
ISH 0046  40 FORMAT (20A4)
ISH 0047  50 FORMAT (16I5)
ISH 0048  60 FORMAT (5E15.8)
ISH 0049  END

**END OF COMPILED **
SUBROUTINE COPE07 (RA,IA,NORA,NOIA,ISCR1)

**GENERAL INFORMATION**

TITLE.

READ (ISCR1,60) (RA(I),I=1,20)

READ (ISCR1,70) (IA(I),I=1,N2VAR)

**TWO-VARIABLE FUNCTION SPACE INFORMATION**

DO 30 I=1,M2VAR

WRITE (161,100) XX,YY,(RA(K),K=1,H)

DO 30 J=1,H2VY

WRITE (161,110) (RA(KK),KK=H,L1

CONTINUE

CONTINUE

RETURN
C FORMATS 00055
C
------------------------------------------------------------------00056
ISN 0041  40 FORMAT (//5X,SHITLE/5X,20A4) 00057
ISN 0042  50 FORMAT (1H1,4X,35HTWO-VARIABLE FUNCTION SPACE RESULTS) 00058
ISN 0043  60 FORMAT (20A4) 00059
ISN 0044  70 FORMAT (1615) 00060
ISN 0045  80 FORMAT (5X,10I5) 00061
ISN 0046  90 FORMAT (/15X,E12.4,3X,4E13.4) 00062
ISN 0047 100 FORMAT (/3X,2E12.4,3X,4E13.4) 00063
ISN 0048 110 FORMAT (30X,4E13.4) 00064
ISN 0049 120 FORMAT (///5X,49GLOBAL NUMBER ASSOCIATED WITH X-VARIABLE, N2VX =00065
1,15/5X,49GLOBAL NUMBER ASSOCIATED WITH Y-VARIABLE, N2VY =,IS) 00066
ISN 0050 130 FORMAT (/5X,37GLOBAL NUMBERS ASSOCIATED WITH F(X,Y)) 00067
ISN 0051 140 FORMAT (/10X,1HX,11X,1HY,20X,6HF(X,Y)) 00068
ISN 0052 150 FORMAT (5E15.8) 00069
ISN 0053   END 00070
*OPTIONS IN EFFECT*NAME=MAIN OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOSAP NOSKMT NOSXREF NOSALC NOSNSF 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
DATA SET U477COPEOS AT LEVEL 001 AS OF 02/13/81
DATA SET 9188COPEOS AT LEVEL 001 AS OF 07/10/80

SUBROUTINE COPEOS (A,B,IFORM,NFLD)
DIMENSION A(1), B(1), C(10)
DATA COMMA/IH,/,BLANK/IH
ROUTINE TO CONVERT UNFORMATTED DATA TO FORMATTED DATA IN FIELDS OF 10 EACH FIELD RIGHT JUSTIFIED.
NASAAMES RESEARCH CENTER, MOFFETT FIELD, CALIF.
--- INPUT.
A -- ARRAY OF DATA SEPARATED BY COMMAS. MINIMUM DIMENSION OF IS 0002
C --- OUTPUT.
B -- ARRAY OF DATA IN FIELDS OF 10 AND RIGHT JUSTIFIED.
C --- NOTE.
1) DATA IS ASSUMED TO BE REAL OR INTEGER DATA WITH NO EMBEDDED BLANKS WITHIN A GIVEN FIELD.
2) DATA IS CONSIDERED UNFORMATTED IF
   A) A COMMA IS FOUND
   B) LAST NON-BLANK CHARACTER IS IN COLUMN 1-10 AND THERE IS NO DECIMAL AND IT IS NOT RIGHT JUSTIFIED.
C SEARCH FOR LAST NON-BLANK CHARACTER AND SEARCH FOR COMMA.
C CALCULATE NUMBER OF NON-BLANK SETS.

DO 10 I=1,80
   IF (A(I).EQ.COMMA) GO TO 20
   JNON=JNON
   IF (A(I).EQ.BLANK) JNON=0
   IF (A(I).NE.BLANK) JNON=1
   IF (JNON.GT.JNON) KNON=KNON+1
   IF (A(I).NE.BLANK) I=I
10 CONTINUE
C NO COMMA HAS FOUND. DATA MAY BE FORMATTED.
   IF (LIST.EQ.10) GO TO 90
   IF (KNON.GT.1) GO TO 90
C DATA IS UNFORMATTED.
   K=10
C 30 CONTINUE
   Nfld=0
   I=0
30 CONTINUE
   IF (I.GT.80) GO TO 110
C IGNORE LEADING BLANKS.
   IF (A(I).EQ.BLANK) GO TO 30
C 034 C CALCULATE NUMBER OF NON-BLANK CHARACTERS IN THIS FIELD.
   C 053 C
ISH  0036  JJ0=0
ISH  0037  DO 40 J=1,60
ISH  0038  IF (A(J).EQ.COMMA.OR.A(J).EQ.BLANK) GO TO 50
ISH  0040  JJ=JJ\+1
ISH  0041  40  C(JJJ)=A(J)
ISH  0042  50  NFLO=NFLO\+1
ISH  0043  I=I\+JJ
ISH  0044  C  BLANK FIELD NFLO OF B.
ISH  0045  K1=K2=9
ISH  0046  60  B(K)=BLANK
ISH  0047  C  STORE C IN FIELD NFLO OF B, RIGHT JUSTIFIED.
ISH  0048  IF (JJ.EQ.0) GO TO 60
ISH  0049  JJ=JJ\+1
ISH  0050  K=K\+1
ISH  0051  DO 70 L=I,JJ
ISH  0052  70  B(KL)=C(J1)
ISH  0053  K=K\+1
ISH  0054  80  J1=J1\+1
ISH  0055  90  CONTINUE
ISH  0057  C  FORMATTED INPUT. STORE A DIRECTLY IN B.
ISH  0058  IFORM=1
ISH  0059  NFLO=8
ISH  0060  DO 100 I=1,80
ISH  0061  100  B(I)=A(I)
ISH  0062  110  CONTINUE
ISH  0063  RETURN
ISH  0064  END

*OPTIONS IN EFFECT* NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOCOL(NONE)
*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NOAP NOFORMAT GOSTMT NOXREF NOALC NDANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 63, PROGRAM SIZE = 646, SUBPROGRAM NAME =COPE08
*STATISTICS* NO DIAGNOSTICS GENERATED
 **** END OF COMPIILATION *****

3012K BYTES OF CORE NOT USED
REQUESTED OPTIONS: SOURCE, NOLIST, NODECK, OPT(3), AUTOCOLL(ONE), NOASM

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MIX) AUTOCOLL(ONE)

SOURCE EGCICD NLIST NODECK OBJECT NOLIST NOFORMAT GOSTIT NOXREF NDARC NOANSF TERM IGM FLAG(I)

C DATA SET U477COPE09 AT LEVEL 002 AS OF 03/16/81
C DATA SET U477COPE09 AT LEVEL 002 AS OF 02/13/81
C DATA SET 9108COPE09 AT LEVEL 002 AS OF 11/14/80

ISN 0002 SUBROUTINE COPE09
ISN 0003 COMMON /CHMIN/, DELFUN, DABFUN, FDCHM, CT, CHMIN, CTLMIN, ALPHAX00004
1,ABS3J, THETA, DEJ, NDV, NCON, HSIDE, IPINT, NFDS, NSCAL, LMOBJ, ITHMX, IT00005
2R, INDIR, GTOQ, NAC, INFO, INFOG, ITER
ISN 0004 COMMON /COPEPS2/ R(AI000), IA(1000)
00007
ISN 0005 COMMON /COPEPS3/ NSIGNPT, NSCAL, JOBJ, NSV, NSOBJ, NCCNA, NEVX, NEVY, MEDO008
12VY, NEVAR, IPSENS, IPFVAR, IPGDS, NCCX1, NTOTV, LOCRI(25), LOCRI(25), ISCR00009
21, ISCR2, NAPRX, NPS, NPF, NAC, IAPRX, KMIN, NMAX, XFACT1, XFACT0010
3, NAX, NAX1, NMAX, NPTOT, JNO, MAXTRN
ISN 0006 COMMON /GLOBCH/ ARRAY(1500)
ISN 0007 COMMON /JOIN/ I51, I61

C ***************************************************************
C ROUTINE TO DO APPROXIMATE OPTIMIZATION.
C ***************************************************************
C
C BY G. H. VANDERPLAATS
C NASA Ames Research Center, Moffett Field, Calif.
C
C JAN., 1979.
C
C
ISN 0008 NFDSV = NFDS
ISN 0009 NFDS = 1
ISN 0010 IF (NFDSV .LT. 0) NFDS = 0
ISN 0011 CHMIN = ABS(CHMIN)
ISN 0012 CTLMIN = ABS(CTLMIN)
ISN 0013 IF (CHMIN .LE. 0.) CHMIN = .005
ISN 0014 IF (CTLMIN .LE. 0.) CTLMIN = .001
ISN 0015 IF (ABS(CT) .LE. 0.) CT = .01
ISN 0016 IF (ABS(CTL) .LE. 0.) CTL = .001
ISN 0017 IF (DELFUN .LE. 0.) DELFUN = .001
ISN 0018 IF (IPRINT .LT. 1) IPRINT = 0
ISN 0019 IF (IMAX .LT. 50) IMAX = 50
ISN 0020 HSIDE = 1
ISN 0021 KOUNT = 0

C **************************************** 0000
C ARRAY STARTING LOCATIONS.
C
ISN 0022 NN1 = VXAPRX + 2
ISN 0023 NKV = LOCRI(23)
ISN 0024 NVLB = NVLB + NN1
ISN 0025 NMOCH = NVLB + NN1
ISN 0026 VDX = VXAPRX + NN1
ISN 0027 NFXOM = VDX + NDV
ISN 0028 NFNEW = NFXOM + 1F
ISN 0029 NSTAY = NFNEW + NF
ISN 0030 NDR = VXAPRX + (VXAPRX*VXAPRX + 11)/2
ISN 0031 IF (MAXTRN .LT. 3) NDR = MAXTRN*VXAPRX
ISN 0032 NTSP = NSTAY + NDR*NF
ISN 0033 NB3U = LOCRI(6)
ISN 0034 HAIC = LOCRI(4)
ISN 0035 HNFGH = LOCRI(23)
ISN 0036 NIDV = NIFGH + NCCNA
ISN 0037 IF (KMAX .LT. 0) GO TO 160

PAGE 1
NEX
VERSIOH 1.3.0 (01 MAY 80) COPE09 SYSTEM/370 FORTRAN M EXTENDED (ENHANCED) DATE 82.141/10.54.59 PAGE 2

C COMMINT ARRAYS.
C
C DIMENSIONS.
C
ISN 0049
NN1 = NDV + 2
ISN 0050
NN2 = 2*NDV + NCONH
ISN 0051
NN3 = NACMX1
ISN 0052
NN4 = NN3
ISN 0053
IF (NDV.GT. NN4) NN4 = NDV
ISN 0055
NNS = 2*NN4

C SCAL, DF, G, A, S, G1, G2, C, B, ISC, IC, MS1.

ISN 0056
NNSCAL = LCCR(4)
ISN 0057
NDF = NTHP
ISN 0058
NS1 = NDF + NN1
ISN 0059
NA = NS + NN2
ISN 0060
NS = NA + NN1*NN3
ISN 0061
NG1 = NS + NN1
ISN 0062
NG2 = NG1 + NN2
ISN 0063
NC = NSG + NN1 + NN2
ISN 0064
NB = NC + NN4
ISN 0065
NISC = LOCI(4)
ISN 0066
NIC = NIVD + NXAPRX
ISN 0067
NMS1 = NIC + NN3

------------------------------------------------------------------
OTHERLINE IOBJA, ARRAYS IGFN AND IDV.
------------------------------------------------------------------

IOBJA.
M6 = LOCI(6)

DO 10 I=1, NF
J = IA(I)
IF (J .EQ. IOBJ) GO TO 20
M6 = M6 + 1

10 CONTINUE
ERROR - IOBJA NOT FOUND.

IF (NCOHA .EQ. 0) GO TO 60

IGFN ARRAY.
M3 = LOCI(3)
M23 = LOCI(23)

DO 50 I=1,NCCNA
J = IA(I)
M3 = M3 + 1
M23 = M23 + 1
50 CONTINUE

COMMENT: -----GLOBAL LOCATIONS OF CONSTRAINED PARAMETERS.----------------
J = IA(M3)
M3 = M3 + 1

COMMENT: -----LOCAL VARIABLE, F, LOCATION.------------------
M6 = LOCI(6)

DO 30 K=1, NF
L = IA(M6)
IF (L .EQ. J) GO TO 40
30 CONTINUE
M6 = M6 + 1

COMMENT: ERROR - CONSTRAINED VARIABLE IS NOT AN APPROXIMATE FUNCTION.

DO 40 K=1, NF
L = IA(M6)
IF (L .EQ. J) GO TO 40
40 CONTINUE
M6 = M6 + 1

COMMENT: -----IDV ARRAY.----------------------------------------
N3 = NIVD
N5 = LOCI(5)
**Comment:** This approximating variable is not a design variable.

**Comment:** Check to be sure each independent design variable is also an approximating variable.

**Comment:** Print initial information.

**Comment:** Title.

**Comment:** Print initial information.

**Comment:** Print initial information.

**Comment:** Objective function.
IF (NCONA .EQ. 0) GO TO 140

COMMENT:--------------- CONSTRAINTS.

WRITE (I6I,560) OC175
N1 = NIDV
N2 = N1 + NCONA - 1
WRITE (I6I,570) (IA(I),I=N1,N2)

140 CONTINUE

COMMENT:--------------- DESIGN VARIABLES.

WRITE (I6I,580) IIIDV
H2 = HI t HP - 1
WRITE (I6I,590) (IA(I),I=HI,N2)

160 CONTINUE

COMMENT:--------------- ITERATION NUMBER.

WHITE (I6I,680) KOUNT
H2 = NPTOT - 1

C SET UP ARRAYS XNOM, FNOM, XI, Y.
C
NX1 = NTHP
NX = NX1 + NXAPRX*NPTOT
NY = NY + NF*NPTOT
MIGHT = NY + HF*NPTOT

IF (IKOUNT .LT. 2 .AND. DABFUH .LE. 0.) DABFUH = .001*ABS(H1OBJA)
IF (IPAPRX .LT. 1 .OR. IPAPRX .EQ. 3) GO TO 170

COMMENT:----------------------------------------------------------------
CALL COPE12 (RAINXI, RAINY), NX1, HP, NI, RA1 NBTAY, RA1 NAA, RA1 HFF)

C LEAST SQUARES FIT FOR TAYLER SERIES EXPANSION.

NX1 = NXAPRX
M = NX1 + (NX1*(NX1 + 1))/2
IF (M .GT. NP) M = NP
IF (NP .LT. NXAPRX) M = NXAPRX
MMAX = NXAPRX*MXTXM
IMAX = MAXTRM
IF (MXTXM .LT. 2) MMAX = M
IF (M .GT. MMAX) M = MMAX

C LEAST SQUARES FIT FOR TAYLER SERIES EXPANSION.

COMMENT:----------------------------------------------------------------
IRR(NG),NKAPRX,NF,NER,RA(NIGHT),NER)
00232

COMMENT:----------------------------------------- 00233

ISH 0199
  IF (NER .GT. 0) WRITE (I6I,590) 00234
ISH 0201
  IF (KMAX .LT. 0) GO TO 530 00235
ISH 0203
  IF (IPAPRX .LT. 2 .AND. IPAPRX .NE. 4) GO TO 150 00236

COMMENT:----------------------------------------- 00237

ISH 0205
  WRITE (I6I,720) 00238
ISH 0206
  N3 = LOC6(6) 00239
ISH 0207
  N1 = NDTAY 00240
ISH 0209
  DO 180 J=1,NF 00241
    N2 = N1 + M - 1 00242
ISH 0210
  WRITE (I6I,600) J,IA(N3) 00243
ISH 0211
  WRITE (I6I,700) (RA(I),I=N1,N2) 00244
ISH 0213
  N3 = N3 + 1 00245
ISH 0214
  180 CONTINUE 00246
ISH 0215
  190 CONTINUE 00247
ISH 0216
  IF (KCOUNT .GT. KMAX) GO TO 470 00248

C
C
C

ISH 0218
  N1 = NXV 00249
ISH 0219
  DO 200 I=1,NKAPRX 00250
    RA(N1) = 0. 00251
    N1 = N1 + 1 00252
ISH 0222
  200 CONTINUE 00253
ISH 0223
  N2 = NVLB 00254
ISH 0224
  N3 = NVUB 00255
ISH 0225
  N4 = NNHSM 00256
ISH 0226
  N5 = LOC2(2) 00257
ISH 0227
  N6 = LOC2(3) 00258
ISH 0228
  N7 = LOC2(7) 00259
ISH 0229
  N8 = NDX 00260
ISH 0230
  L1 = M - NDV 00261
ISH 0231
  L2 = L1 - NDV 00262
ISH 0232
  ICK = ICK1 + ICK2 + ICK3 00263
ISH 0233
  DO 210 I=1,NDV 00264
    RA(N8) = 0. 00265
    XFAC = 1. 00266
ISH 0234
    IF (I .LE. L1) XFAC = XFAC1 00267
ISH 0235
    L2 = L2 - NDV + I 00268
ISH 0236
    IF (L2 .GE. 0) XFAC = XFAC2 00269

COMMENT:----------------------------------------- 00270

ISH 0237
  IF (ICK .GT. 0) XFAC = .5 00271
ISH 0238
  DX = XFAC*RA(N7) 00272
ISH 0239
  XX = RA(N4) 00273
ISH 0240
  DNL = XX - RA(N5) 00274
ISH 0241
  IF (DNL .GT. DXL) DXL = DX 00275
ISH 0242
  DXL = RA(N6) - XX 00276
ISH 0243
  IF (DXL .GT. DXU) DXU = DX 00277
ISH 0244
  RA(N1) = -DXL 00278
ISH 0245
  RA(N2) = -DXL 00279
ISH 0246
  RA(N3) = DXU 00280
ISH 0247
  N2 = N2 + 1 00281
ISH 0248
  N3 = N3 + 1 00282
ISH 0249
  N4 = N4 + 1 00283
ISH 0250
  N5 = N5 + 1 00284
ISH 0251
  N6 = N6 + 1 00285
ISH 0252
  N7 = N7 + 1 00286
ISH 0253
  N8 = N8 + 1 00287

PAGE 5
CONTINUE 00291
IF (IPAPRX.LT.2 .AND. IPAPRX .NE. 4) GO TO 220
00292
WRITE (161,610)
00293
N1 = NVVB + NDV - 1
00294
WRITE (161,700) (RA(I), I=NVVB,N1)
00295
WRITE (161,620)
00296
N1 = NVUB + NDV - 1
00297
WRITE (161,700) (RA(I), I=NVUB,N1)
00298
CONTINUE 00299
C
OPTIMIZE APPROXIMATE FUNCTION.
00300
C
COMMENT:---------------- 
00301
CONTINUE 00302
C
IGOTO = 0
00303
CALL CONMIN (X,VLB,VUB,G,SCAL,DF,A,S,Gl,G2,B,C,ISC,IC,MS1,N1,N2,H3)
00304
*H4,N5)
230 CONTINUE
00305
COMMENT:----------------
00306
COMMENT:----------------
00307
COMMENT:----------------
00308
COMMENT:----------------
00309
COMMENT:----------------
00310
COMMENT:----------------
00311
CONTINUE 00312
N1 = NLVB + NOV - 1
00313
WRITE (161,700) (RA(I), I=NLVB,N1)
00314
WRITE (161,620)
00315
N1 = NLB + NOV - 1
00316
WRITE (161,700) (RA(I), I=NLB,N1)
00317
CONTINUE 00318
C
APPROXIMATE ANALYSIS.
00319
COMMENT:----------------
00320
COMMENT:----------------
00321
COMMENT:----------------
00322
COMMENT:----------------
00323
COMMENT:----------------
00324
COMMENT:----------------
00325
COMMENT:----------------
00326
COMMENT:----------------
00327
COMMENT:----------------
00328
COMMENT:----------------
00329
COMMENT:----------------
00330
COMMENT:----------------
00331
COMMENT:----------------
00332
COMMENT:----------------
00333
COMMENT:----------------
00334
COMMENT:----------------
00335
COMMENT:----------------
00336
COMMENT:----------------
00337
COMMENT:----------------
00338
COMMENT:----------------
00339
COMMENT:----------------
00340
COMMENT:----------------
00341
COMMENT:----------------
00342
COMMENT:----------------
00343
COMMENT:----------------
00344
COMMENT:----------------
00345
COMMENT:----------------
00346
COMMENT:----------------
00347
COMMENT:----------------
00348
COMMENT:----------------
00349
DO 270 I = 1, NXAPRX
RA(N1) = RA(N2) + RA(N3)
N1 = N1 + 1
N2 = N2 + 1
N3 = N3 + 1

DO 270 CONTINUE

COMMENT:--------- READ X-VECTORS ONE AT A TIME AND COMPARE TO XNOM.

REMIND ISCR2
N1 = NTHP + NXAPRX
N2 = N1 + NXAPRX - 1
N3 = N2 + 1
N4 = N3 + HP - 1
DO 290 J = 1, HPTOT

COMMENT:--------- READ (ISCR2) (RA(I),I=N1,N2)

X-VECTOR. NOT USED. READ TO POSITION ISCR2.

READ (ISCR2) (RA(I),I=N3,N4)

COMMENT:--------- COMPARE X WITH XNOM.

N5 = N1
N6 = NTHP
SUM = 0.
DO 280 I = 1, NXAPRX
SUM = SUM + (RA(N5) - RA(N3))**2
N5 = N5 + 1
N6 = N6 + 1

IF (SUM.LT.1.0E-10) GO TO 300

CONTINUE

IF (SUM.LT.1.0E-10) GO TO 300

CONTINUE

GO TO 300

CONTINUE

C THIS DESIGN IS SAME AS PREVIOUS DESIGN.

C MODIFY DELTA-X VECTOR.

N8 = NTHP + NXAPRX - 1
IF (IPAPRX.LT.1.0R.IPAPRX.EQ.3) GO TO 310
WRITE (161,630)
WRITE (161,740)
WRITE (161,700)
WRITE (161,750)
N9 = NTHP + HXAPRX - 1
WRITE (161,640)

IF (SUM.LT.1.0E-10) GO TO 300

CONTINUE

AMULT = .01*FLOAT(IJJJ)
DO 320 I = 1, NDV
BU = RA(N6)
BL = RA(N8)
IF (IBL.LT.-1.0E+15) BL = 0.
IF (IBU.GT.1.0E+15) BU = 0.
DB = ABS(BU-BL)
IF (DB.GT.1.0E-6) DB = 1.
DO 350 I = 1, NDV
DX = RA(N7) + AMULT*DB
IF (DX.GT.RA016) DX = DX - 2.*AMULT*DB
DX = DX + 1.5*AMULT*DB

CONTINUE

AMULT = .01*FLOAT(IJJJ)
DO 320 I = 1, NDV
BU = RA(N6)
BL = RA(N8)
IF (IBL.LT.-1.0E+15) BL = 0.
IF (IBU.GT.1.0E+15) BU = 0.
DB = ABS(BU-BL)
IF (DB.GT.1.0E-6) DB = 1.
DO 350 I = 1, NDV
DX = RA(N7) + AMULT*DB
IF (DX.GT.RA016) DX = DX - 2.*AMULT*DB
DX = DX + 1.5*AMULT*DB
**+VERSION 1.3.0 (01 MAY 80) C0PE09 SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 02.14.81/10.54.59 PAGE 8**

**ISN 0363**  
RA(N7)=DX  
00409

**ISN 0364**  
N5=N5+1  
00410

**ISN 0365**  
N7=N7+1  
00411

**ISN 0366**  
N8=N8+1  
00412

**ISN 0367**  
320 CONTINUE  
00413

**ISN 0368**  
CALL COPE15 (RA(HV),RA(N5),RA(N7),RA(NA),IA(HISC),IA(NIC),NN1,RA00414
1(NBLU),NX1,OBJA,M,RA(NHOM),RA(NHEW),RA(NHST),NBR,IA(HFHE),CT,00415
2CT1,JINFO,NAC,NCOHA,MDV,NF,OBJ,SGOPT)  
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.NPTOTI) GO TO 340  
00421

**ISN 0372**  
KK=KK+1  
00422

**ISN 0373**  
DO 330 J=KK,NPTOT  
00423

**ISN 0374**  
READ (ISCR2) (RA(I),I=NI,N2)  
00424

**ISN 0375**  
330 READ (ISCR2) (RA(I),I=NI,N2)  
00425

**ISN 0376**  
CONTINUE  
00426

**ISN 0377**  
IF (IPAPRX.LT.1.OR.IPAPRX.EQ.3) GO TO 350  
00427

**ISN 0378**  
IF (JJJ.EQ.4) WRITE (161,650)  
00428

**ISN 0379**  
CONTINUE  
00429

**ISN 0380**  
CONTINUE  
00430

------------------------------------------------------------------

UPDATE ANALYSIS.  
------------------------------------------------------------------

**ISN 0381**  
XNOM=  
00431

**ISN 0382**  
n1=N1  
00432

**ISN 0383**  
n2=N2  
00433

**ISN 0384**  
DO 370 I=I,IPAPRX  
00434

**ISN 0385**  
RAIN11=RAIN11+RAIN2)  
00435

**ISN 0386**  
HH1=N1+1  
00436

**ISN 0387**  
N2=N2+1  
00437

**ISN 0388**  
GLOBAL VARIABLES.  
00438

**ISN 0389**  
N3=N3  
00439

**ISN 0390**  
n4=N4  
00440

**ISN 0391**  
DO 410 I=1,IPAPRX  
00441

**ISN 0392**  
II=IA(N4)  
00442

**ISN 0393**  
IF (II.EQ.0) GO TO 400  
00443

**ISN 0394**  
DESIGN VARIABLE NUMBER.  
00444

**ISN 0395**  
II=II+1  
00445

**ISN 0396**  
IF (II.EQ.0) GO TO 400  
00446

**ISN 0397**  
DESIGN VARIABLE UPDATE.  
00447

**ISN 0398**  
II=II+1  
00448

**ISN 0399**  
UPDATE VARIABLE J.  
00449

**ISN 0400**  
JJ=IA(N1)  
00450

**ISN 0401**  
ARRAY(JJ)=RA(N3)*RA(N5)  
00451

**ISN 0402**  
360 N1=N1+1  
00452

**ISN 0403**  
N2=N2+1  
00453

**ISN 0404**  
CONTINUE  
00454

**ISN 0405**  
CONTINUE  
00455

**ISN 0406**  
CONTINUE  
00456

**ISN 0407**  
CONTINUE  
00457

**ISN 0408**  
CONTINUE  
00458

**ISN 0409**  
CONTINUE  
00459

**ISN 0410**  
CONTINUE  
00460

**ISN 0411**  
CONTINUE  
00461

**ISN 0412**  
CONTINUE  
00462

**ISN 0413**  
CONTINUE  
00463

**ISN 0414**  
CONTINUE  
00464

**ISN 0415**  
CONTINUE  
00465

**ISN 0416**  
CONTINUE  
00466

**ISN 0417**  
CONTINUE  
00467

**ISN 0418**  
CONTINUE  
00468

**ISN 0419**  
CONTINUE  
00469
ISH 0415 WRITE (161,750) 00466
ISH 0416 N2=NXNOM+NXAPRX-1 00469
ISH 0417 WRITE (161,700) (RA(I),I=NXNOM,N2) 00470
ISH 0418 WRITE (161,760) 00471
ISH 0419 N2=NFNEW+NF-1 00472
ISH 0420 WRITE (161,700) (RA(I),I=NFNEW,N2) 00473
ISH 0421 CONTINUE 00474
ISH 0422 IF ((ICK1.GE.2.AND.KOUNT.GE.KMIN).AND.(IPAPRX.GT.0.AND.IPAPRX.NE.300475
1)) WRITE (161,540) 00476
ISH 0423 IF (ICK1.GE.2.AND.KOUNT.GE.KMIN) GO TO 460 00477
ISH 0424 ICALC=2 00478
ISH 0425 NAN2=NAN2+1 00479
ISH 0426 CALL ANALIZ (ICALC) 00480
ISH 0427 NEW FUNCTION VALUES. 00481
ISH 0428 N1=NFNON 00482
ISH 0429 M6=LOCII6) 00483
ISH 0430 DO 430 I=I,NF 00484
ISH 0431 RAIH )=ARRAYI II) 00485
ISH 0432 430 H1=N1+l 00486
ISH 0433 IF IIIPDBG.LT.l) GO TO 440 00489
ISH 0434 N1B=N1B+1 00491
ISH 0435 ICALC=3 00492
ISH 0436 CALL ANALIZ (ICALC) 00493
ISH 0437 440 CONTINUE 00494
ISH 0438 IF (IPAPRX.LT.1.0R.IPAPRX.EQ.3) GO TO 450 00495
ISH 0439 ICALC=3 00496
ISH 0440 NEW OBJECTIVE. 00497
ISH 0441 N1=NXAPRX+1 00498
ISH 0442 OBJECTIVE. 00499
ISH 0443 NPTOT=NPTOT+1 00502
ISH 0444 WRITE ISCR2) (RA(I),I=NXNOM,N1) 00505
ISH 0445 WRITE ISCR2) (RA(I),I=N1,HI) 00508
ISH 0446 WRITE ISCR2) (RA(I),I=N1B,HI) 00511
ISH 0447 CONTINUE 00514
ISH 0448 OBJ=-RAIN1)*SG1OPT 00517
ISH 0449 NPTOT=NPTOT+1 00520
ISH 0450 WRITE ISCR2) (RA(I),I=NFNON+1) 00523
ISH 0451 WRITE ISCR2) (RA(I),I=N1B,HI) 00526
ISH 0452 UPDATE PARAMETERS. 00529
ISH 0453 NPTOT=NPTOT+1 00532
ISH 0454 IF (J.JJ.LT.2.OR.KOUNT.LT.KMIN) INGH=NPTOT 00535
ISH 0455 CONVERGENCE CHECK. 00538
ISH 0456 IF (KOUNT.LT.KMIN) GO TO 130 00541
ISH 0457 ICK2=ICK2+1 00544
ISH 0458 ICK3=ICK3+1 00547
ISH 0459 DEL=ABS(OBJ) 00550
ISH 0460 IF (DEL.LT.1.0E-6) DEL=1.0E-6 00553
ISH 0461 DEL=ABS(OBJ1)/DEL 00556
ISH 0462 DEL=ABS(OBJ1) 00559
ISH 0463 IF (DEL.GT.DELFUN) ICK2=0 00562
ISH 0464
ISH 0468  DEL=ABS(OBJ-OBJSAV) 00527
ISH 0469  IF (DEL.GT.DABFUN) ICK3=0 00528
ISH 0471  IF (ICK2.GE.2.OR.ICK3.GE.2) GO TO 460 00529
ISH 0473  IF (KOUNT.LT.KMAX) GO TO 130 00530
ISH 0475 460 CONTINUE
ISH 0476  IF (IPAPRX.GT.0.AHD.IPAPRX.NE.3) WRITE (161,660) 00531
ISH 0478  INOM=0 00532
ISH 0479  KOUNT=KMAX+1 00533
ISH 0480  IF (KOUNT.LT.JHOM) KOUNT=JHOM+1 00534
ISH 0482  CTSAV=CT 00535
ISH 0483  CTLSAV=CTL 00536
ISH 0484  IF (ABS(CT).LT.1.0E-10) CT=-.004 00537
ISH 0486  IF (ABS(CTL).LT.1.0E-10) CTL=-.001 00538
ISH 0488  GO TO 160 00539
ISH 0489  470 CONTINUE
ISH 0490  CT=CTSAV 00540
ISH 0491  CTL=CTLSAV 00541
ISH 0492  STORE FINAL VALUES OF XNOM IN GLOBAL ARRAY. 00542
ISH 0493  N3=INOM 00543
ISH 0494  N4=NIVD 00544
ISH 0495  DO 510 I=I,NXAPRX 00545
ISH 0496  II=IA(N4) 00546
ISH 0497  IF (II.EQ.O) GO TO 500 00547
ISH 0498  ARRAY(I)=RA(H3)*RA(H5) 00548
ISH 0499  510 CONTINUE
ISH 0500  S10 H4=N4+1 00549
ISH 0501  STORE FINAL VALUES OF FNUM IN GLOBAL ARRAY. 00550
ISH 0502  M6=LOCI(6) 00551
ISH 0503  IF (IA(H2).NE.II) GO TO 480 00552
ISH 0504  JJ=IA(IH) 00553
ISH 0505  ARRAY(JJ)=RA(H1) 00554
ISH 0506  520 H1=N1+1 00555
ISH 0507  H2=H2+1 00556
ISH 0508  H5=H5+1 00557
ISH 0509  STORE FINAL VALUES OF FKOM IN GLOBAL ARRAY. 00558
ISH 0510  M6=LOCII(6) 00559
ISH 0511  N4=N4+1 00560
ISH 0512  IF (IA(H2).NE.II) GO TO 480 00561
ISH 0513  JJ=IA(IH) 00562
ISH 0514  ARRAY(JJ)=RA(N3)*RA(N5) 00563
ISH 0515  530 CONTINUE
ISH 0516  M6=LOCII(6) 00564
ISH 0517  N1=N1+1 00565
ISH 0518  ID=IA(N6) 00566
ISH 0519  ARRAY,ID=RA(N1) 00567
ISH 0520  RETURN
ISH 0521  540 FORMAT (/5X,71HTWO CONSECUTIVE APPROXIMATE OPTIMIZATIONS HAVE PRO0553
ISH 0522  IDUCED THE SAME DESIGN/5X,23OPTIMIZATION TERMINATED) 00554
ISH 0523  FORMAT (/5X,15,17H IS THE OBJECTIVE) 00555

FORMATS

---
**APPROXIMATING FUNCTIONS ASSOCIATED WITH CONSTRAINTS**

1) $00586$

**DESIGN VARIABLE NUMBERS ASSOCIATED WITH APPROXIMATION**

1) $00588$

**LEAST SQUARES FIT TO APPROXIMATION DATA IS SINGULAR**

2) $00592$

**FUNCTION NUMBER, IS, GLOBAL VARIABLE NUMBER, 15/2**

3) $00593$

**SIDE CONSTRAINTS ON APPROXIMATE OPTIMIZATION**

4) $00595$

**UPPER BOUNDS**

5) $00597$

**OPTIMIZATION WILL CONTINUE WITH MOST RECENT X-VECTOR**

6) $00600$

**FINAL RESULT OF APPROXIMATE OPTIMIZATION**

7) $00603$

**BEGIN ITERATION N U M B E R = 15**

8) $00605$

**FUNCTION VALUES**

9) $00610$

**APPROXIMATE FUNCTION VALUES**

10) $00616$

**END OF COMPIIATION**

11) $00617$
DATA SET U477COPE10 AT LEVEL 001 AS OF 02/13/81
DATA SET 9188COPE10 AT LEVEL 001 AS OF 07/10/80

SUBROUTINE COPE10 (XI,Y,NOM,POINT,KOUNT,BlU,IGN,INOM,ISC,NO0002
1XAPRX,NF,NCHNA,ENDOPT,CTHIN,CTUHNI,ISCR2,NOMTH,HPMAX,JNOM) 00003
DIMENSION XI(NXAPRX), Y(NF), NCHNI(I), BlU(4,1), IGN(100004
11), ISC(1), NCHTH(1) 00005

C ******************************************************************00006 ROUTINE TO SET 00007
up ARRAYS FOR TAYLER SERIES EXPANSION. 00007
C ******************************************************************00008

BY G. N. VANDERPLAATS 00009 NASA Ames Research Center, Moffett Field, Calif. 00010

C ISN 0004 REHIND ISCR2 00011
C ISN 0005 DO 10 J=1,NPTOT 00014
C ISN 0006 READ (ISCR2) (XI(I,J),I=1,NXAPRX) 00015
C ISN 0007 10 READ (ISCR2) (Y(I,J),I=1,NF) 00016

C FIND BEST NOMINAL IF REQUIRED. 00017
C------------------------------------------------------------------00018
C IF (KOUNT.LE.1.AND.INOM.GT.0) GO TO 20 00019
C IF (KOUNT.GT.1.AND.KOUNT.LE.JNOM) GO TO 20 00020
C CALL COPE11 (NPTOT,Y,NF,NOM,BlU,NCHNA,IGN,INOM,ISC,NO0022
1IOM,ISC) 00023
C CONTINUE 00024

C CREATE XNOM AND FNC:1. 00025
C------------------------------------------------------------------00026
C DO 30 I=1,NXAPRX 00027
C XNOM(I)=XI(I,INOM) 00028
C DO 40 I=1,NF 00029
C Y(I,INOM)=Y(I,INOM) 00030
C CONTINUE 00031

C REPLACE XI BY DELTA-XI AND Y BY DELTA-Y. 00032
C------------------------------------------------------------------00033
C DO 80 J=1,NOM 00034
C XI(I,J)=XI(I,J)-XNOM(I) 00035
C Y(I,J)=Y(I,J)-FNOM(I) 00036
C CONTINUE 00037

C WEIGHTING FACTORS. 00038
C------------------------------------------------------------------00039
C DO 120 J=1,NOM 00040
C X(I,J)=XI(I,J)-XNOM(I) 00041
C Y(I,J)=Y(I,J)-FNOM(I) 00042
C CONTINUE 00043

C
C 00055

ISH 0034  SMAX=1.0E-10 00056
ISH 0035  DO 130 J=1,NP 00057
ISH 0036  SUM=0. 00058
ISH 0037  DO 120 I=1,NXAPRX 00059
ISH 0038  SUM=SUM+XI(I,J)**2 00060
ISH 0039  IF (SUM.GT.SMAX) SMAX=SUM 00061
ISH 0040  DO 120 I=1,NP 00062
ISH 0041  120 SUI1=SUI1+XI(I,J)**2 00063
ISH 0042  SMAX=SQRT(SMAX) 00064
ISH 0043  IF (NP.LE.NPMAX) RETURN 00065
ISH 0044  130 WIGHT(JI)=SQRT(SUI1) 00066
ISH 0045  STIAX=SQRT(SSTIAX) 00067

C 00058

C REDUCE THE NUMBER OF DESIGNS TO NPMAX. 00059

C 00060

ISH 0047  NPMAX=NP+1 00061
ISH 0048  NPSAV=NP 00062
ISH 0049  DO 200 II=NPMAX,NPSAV 00063
C FIND DESIGN WITH MINIMUM WEIGHTING FACTOR. 00064
ISH 0050  WMIN=WIGHT(JI) 00065
ISH 0051  IM1=1 00066
ISH 0052  DO 150 I=2,NP 00067
ISH 0053  IF (WIGHT(JI).GE.WMIN) GO TO 150 00068
ISH 0054  IM1N=I 00069
ISH 0055  IM1=I 00070
ISH 0056  CONTINUE 00071
ISH 0057  IF (IM1N.EQ.NP) GO TO 190 00072
C SHIFT XI, Y AND WIGHT. 00073
ISH 0058  NP=NP-1 00074
ISH 0059  150 CONTINUE 00075
ISH 0060  XI(I,J)=XI(I,J+1) 00076
ISH 0061  Y(I,J)=Y(I,J+1) 00077
ISH 0062  WGHT(I)=WGHT(I+1) 00078
ISH 0063  160 XI(I,J)=XI(I,J+1) 00079
ISH 0064  DO 170 I=1,NF 00080
ISH 0065  Y(I,J)=Y(I,J+1) 00081
ISH 0066  WGHT(I)=WGHT(I+1) 00082
ISH 0067  170 Y(I,J)=Y(I,J+1) 00083
ISH 0068  180 XI(I,J)=XI(I,J+1) 00084
ISH 0069  RETURN 00085
ISH 0070  END 00086

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOCEIL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NODECK OBJECT NMAP NOFORMAT GOSTIT HOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 69, PROGRAM SIZE = 2038, SUBPROGRAM NAME = COPE10
*STATISTICS* NO DIAGNOSTICS GENERATED
****** END OF COMPILATION ******
DATA SET U477COPE11 AT LEVEL 001 AS OF 02/13/81
DATA SET 9188COPE11 AT LEVEL 001 AS OF 07/10/80

SUBROUTINE COPE11 (NPTOT, Y, NVR, INCN, BLU, NCONA, IGFN, IOBJA, SGNOPT, CT)

DIMENSION (NVR1, BLU4, IGFNI, ISC1)

ROUTINE TO DETERMINE NOMINAL DESIGN FOR APPROXIMATE OPTIMIZATION.

BY G. N. VANDERPLAATS

NASA Ames Research Center, Moffett Field, Calif.

NOMINAL DESIGN IS THE ONE WITH LOWEST OBJECTIVE SATISFYING ALL CONSTRAINTS.

THE LEAST VIOLATION IS FOUND.

II IS DESIGN WITH LOWEST MAXIMUM CONSTRAINT VALUE.

II IS THE DESIGN WITH THE LOWEST OBJECTIVE SATISFYING ALL CONSTRAINTS.

GMAX=1.0E+20

11=1

DD 10 J=2, NPTOT

OBJ=-Y(IOBJA, J)*SGNOPT

DO 10 J=1, NPTOT

OBJ=-Y(IOBJA, J)*SGNOPT

OBJ=-Y(IOBJA, J)*SGNOPT

CONTINUE
ISH 0030   CT=CT1
ISH 0039   IF (ISCICON).GT.0) CT=CT1
ISH 0041   G=(GS-BLU(3,I))/BLU(4,I)-CT
ISH 0042   IF (G.GT.GL) GL=G
ISH 0044   30 CONTINUE
ISH 0045   IF (Gl.LT.0..OR.GI.GT.GMAX) GO TO 40
ISH 0047   II=J
ISH 0048   GMAX=Gl
ISH 0049   40 IF (OBJ.GT.OBJMAX. OR.G1.GT.O.) GO TO 50
ISH 0051   I2=J
ISH 0052   OBJMAX=OBJ
ISH 0053   50 CONTINUE
ISH 0054   INOM=II
ISH 0055   IF (I2.GT.O) INOM=I2
ISH 0057   RETURN
ISH 0058   END

*OPTION IN EFFECT=NAME(MAIN) OPTIMIZE(3) LINESCOUNT(60) SIZE(MAX) AUTOBL(NONE)
*OPTION IN EFFECT=SOURCE EBCDIC NOISTECK OBJECT NOHAP NOFORMAT COSTHT NOXREF NOALC NOHSSF TERM IBM BLG(I)
*STATISTICS= NO DIAGNOSTICS GENERATED
******* END OF COMPILE *******
SUBROUTINE COPE12

****  ROUTINE TO PERFORM A LEAST SQUARES FIT OF AN ARBITRARY FUNCTION OF NV VARIABLES.  
Y = F(X1,X2,...,XNX) = B(1)*F(1) + B(2)*F(2) + ... + B(M)*F(M)  
****

BY G. N. VANDERPLAATS

NASA AMES RESEARCH CENTER, MOUNT FIELD, CALIF.

ARGUMENTS.

X,Y - INPUT ARRAYS OF OBSERVATIONS OF NP POINTS
X(NX,NP), Y(NF,NP)
NX - NUMBER OF INDEPENDENT VARIABLES OF WHICH Y IS A FUNCTION.
NP - NUMBER OF OBSERVATION POINTS.
NF - NUMBER OF SEPARATE CURVE FITS BEING DONE SIMULTANEOUSLY.
THIS IS THE NUMBER OF SETS OF Y VALUES.
M - NUMBER OF COMPONENTS OF THE FUNCTIONS TO BE FITTED.
B - ARRAY OF M COEFFICIENTS OF FUNCTIONAL FIT TO DATE.
A - (M+M+1)/2 WORK VECTOR.
F - WORK VECTOR - F(I).
G - WORK VECTOR - G(I).
NXR - DIMENSIONED ROWS OF X.
NYR - DIMENSIONED ROWS OF Y.
NSR - DIMENSIONED ROWS OF B.
NIGHT - ARRAY OF WEIGHTING FACTORS - NIGHT(NP).
NER - ERROR FLAG. IF NER.LT.0, DIAGONAL ELEMENT NER OF A IS LESS THAN 1.0E-10.

USER SUPPLIED SUBROUTINE, COPE13.

USAGE
CALL COPE13(XI,F,NX,M)

ROUTINE TO EVALUATE COMPONENTS F(1),...,F(M) WHICH ARE TO BE FITTED TO DATA.

TO DATA, ROUTINE EVALUATES THE FUNCTIONS FOR A SINGLE VECTOR OF XI AND STORES THE RESULTING VALUES IN VECTOR F.

ARGUMENTS.

XI - VECTOR OF INDEPENDENT VARIABLES AT WHICH FUNCTIONS ARE TO BE EVALUATED.
F - VECTOR OF FUNCTION VALUES.
NX - NUMBER OF INDEPENDENT VARIABLES OF WHICH Y IS A FUNCTION.
M - NUMBER OF FUNCTION COMPONENTS, ALSO REQUIRED DIMENSION OF F.

SUBROUTINE COPE12 (X,Y,NX,NP,NF,N,M,B,A,F,G,NXR,NYR,NSR,NIGHT,NER)
DIMENSION X(NXR,1), Y(NYR,1), B(NR,1), A(1), F(I), G(I)

IF (NX.LE.NP) GO TO 50
NXI=NX+1
AP=FLOAT(NP)
DO 20 I=1,NX
XI(I,NP1)=AP
DO 10 J=1,NP
10 XI(I,J)=XI(I,J-1)
20 CONTINUE
**VERSION 1.3.0 (01 MAY 60) COPE12 SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 02.141/10.55.15 PAGE 2

ISH 0012 IF (ABS(X(I,J)).GT.1.0E-10) GO TO 10 00055
ISH 0014 X(I,NP1)=X(I,NP1)-1. 00056
ISH 0015 X(I,J)=1.0E+20 00057
ISH 0016 CONTINUE 00058
ISH 0017 IF ((X(I,NP1)).LT.1.) X(I,NP1)=1. 00059
ISH 0019 CONTINUE 00060
ISH 0020 DO 40 I=1,NX 00061
ISH 0021 DO 30 J=1,NF 00062
ISH 0022 B(I,J)=0. 00063
ISH 0023 DO 30 K=1,NP 00064
ISH 0024 30 B(I,J)=B(I,J)+Y(J,K)/X(K) 00065
ISH 0025 DO 40 I=1,NP 00066
ISH 0026 NER=0 00067
ISH 0027 RETURN 00068
ISH 0028 CONTINUE 00069
ISH 0029 GENERAL CASE. DO LEAST SQUARES FIT. 00070
ISH 0030 A=B=O. 00071
ISH 0031 DO 60 J=1,NF 00072
ISH 0032 DO 60 I=1,M 00073
ISH 0033 B(I,J)=0. 00074
ISH 0034 L=(M(M+1))/2 00075
ISH 0035 DO 70 J=1,L 00076
ISH 0036 A(I,J)=0. 00077
ISH 0037 LOWER TRIANGLE OF A IN SYMMETRIC MODE. 00078
ISH 0038 DO 100 K=1,NP 00079
ISH 0039 WGTK=WGTK(K) 00080
ISH 0040 CALL COPE13 (X(K),F,NX,M) 00081
ISH 0041 L=1 00082
ISH 0042 DO 80 I=1,J 00083
ISH 0043 L=L+1 00084
ISH 0044 A(I,J)=A(I,J)*F(I)*F(J)*WGTK 00085
ISH 0045 C Y*F 00086
ISH 0046 80 A(I,J)=A(I,J)+F(I)*F(J)*WGTK 00087
ISH 0047 CONTINUE 00088
ISH 0048 C SOLVE FOR B. 00089
ISH 0049 IF (M.LE.1) GO TO 200 00090
ISH 0050 C LDU DECOMPOSITION. 00091
ISH 0051 MMI=M-1 00092
ISH 0052 K=0 00093
ISH 0053 NER=K 00094
ISH 0054 K=KK 00095
ISH 0055 IF (ABS(A(KK)).LT.1.0E-20) GO TO 220 00096
ISH 0056 FACT=1./A(KK) 00097
ISH 0057 A(KK)=FACT 00098
ISH 0058 KPI=K+1 00099
ISH 0059 KJ=KK 00100
ISH 0060 DO 60 J=KPI,M 00101
ISH 0061 KJ=KJ+1 00102
ISH 0062 GO=A(KJ)*FACT 00103
ISH 0063 K2=KK 00104
ISH 0064 KJ=KK 00105
ISH 0065 DO 60 I=KPI,KJ 00106
ISH 0066 IJ=KJ 00107
ISH 0067 IJ=IJ+1 00108
ISH 0068 KI=KI+1 00109
ISH 0069 CONTINUE 00110
ISH 0070 IF (X(I,NP1)).LT.1. X(I,NP1)=1. 00111
ISH 0072 CONTINUE 00112
ISH 0073 RETURN 00113
ISH 0069  A(I,J)=A(I,J)-A(K,K)*GSS
ISH 0070  110 CONTINUE
ISH 0071  KI=KK+1
ISH 0072  NER=1
ISH 0073  IF (ABS(A(K,K)).LT.1.0E-20) GO TO 220
ISH 0074  A(K,K)=1./A(K,K)
ISH 0075  C FORWARD SUBSTITUTION
ISH 0076  MPI=HI+1
ISH 0077  KI=0
ISH 0078  DO 130 K=1,KM
ISH 0079  KPI=K+1
ISH 0080  KK=KK+K
ISH 0081  AK=AI(K,K)
ISH 0082  DO 120 L=1,NF
ISH 0083  120 B(K,L)=B(K,L)*AKK
ISH 0084  KI=KK
ISH 0085  DO 130 I=KPI,1M
ISH 0086  KI=KI+1
ISH 0087  DO 120 I=KI,1M
ISH 0088  B(I,K)=B(I,K)-AK*K
ISH 0089  130 CONTINUE
ISH 0090  DO 140 I=1,KM
ISH 0091  KK=KK+M
ISH 0092  AK=AI(K,K)
ISH 0093  DO 140 J=1,NF
ISH 0094  140 B(I,J)=B(I,J)*AKK
ISH 0095  DO 190 I=5,1M
ISH 0096  J=MP1-I
ISH 0097  J=J+(J+1)/2
ISH 0098  K=J
ISH 0099  JP1=J+1
ISH 0100  DO 150 L=1,NF
ISH 0101  150 G(L)=0.
ISH 0102  DO 170 K=JP1,1M
ISH 0103  J=J+K-1
ISH 0104  AJK=AI(J,K)
ISH 0105  DO 150 L=1,NF
ISH 0106  160 G(L)=G(L)+AJ*K*BJ(L)
ISH 0107  170 CONTINUE
ISH 0108  AJJ=AI(J,J)
ISH 0109  DO 160 L=1,NF
ISH 0110  180 B(J,L)=B(J,L)-AJJ*G(L)
ISH 0111  190 CONTINUE
ISH 0112  NE=0
ISH 0113  RETURN
ISH 0114  200 CONTINUE
ISH 0115  AKK=AI(J,J)
ISH 0116  NE=1
ISH 0117  IF (ABS(AKK)).LT.1.0E-20) GO TO 220
ISH 0118  A(KK)=1./AKK
ISH 0119  DO 210 J=1,NF
ISH 0120  210 B(I,J)=B(I,J)*AKK
ISH 0121  NE=0
ISH 0122  RETURN
ISH 0123  RETURN
ISH 0124  220 CONTINUE
ISH 0125  RETURN
ISH 0126  END

*OPTIONS IN EFFECT*NAME=MAIN OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOBLK(NONE)
*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 NODECK NOFORMAT GOSTMT 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
SUBROUTINE COPE13 (X,F,NX,N)

** Diagonal Elements. **

\[
\text{H COEF. } = \frac{1}{2} (X + \frac{X^2}{X})
\]

** Off-Diagonal Elements. **

\[
F(II) = X(I) \times X(J)
\]
DATA SET U477COP14 AT LEVEL 002 AS OF 03/16/81
DATA SET U477COP14 AT LEVEL 001 AS OF 02/13/81
DATA SET 91SSCOP14 AT LEVEL 001 AS OF 03/16/81

SUBROUTINE COPE14 (NXAPRX, NF, NPTOT, RA, IA, LCR, LCC, TITLE, INO, NDV)
COMMON /LIOG/ IS1, I61
COMMON /UC6/ IS1, I61

**ROUTINE TO PRINT RESULTS OF APPROXIMATE ANALYSIS/OPTIMIZATION.**

BY G. N. VANDERPLAATS JAN., 1979.
NASA AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.

------------------------------------------------------------------
TITLE.
------------------------------------------------------------------
WRITE (161,90) (TITLE(I), I=1,20)
------------------------------------------------------------------
GLOBAL LOCATION OF X AND FIX
------------------------------------------------------------------
GLOBAL LOCATIONS OF X.
M5=LCCI(5)
M6=M5+NXAPRX-1
WRITE (161,100) (IA(I), I=M5,M6)
------------------------------------------------------------------
X-VALUES AND FUNCTIONS, F(X)
------------------------------------------------------------------
X-VALUES.
N1=LCC(23)+3*NXAPRX+6
N2=N1+NXAPRX-1
WRITE (161,120) (RA(I), I=N1,N2)
------------------------------------------------------------------
TAYLOR SERIES COEFFICIENTS.
------------------------------------------------------------------
ISN 0031 WRITE (161,80) JJ,N6
C LINEAR TERMS.
ISN 0032 N2=NXAPRX
ISN 0033 IF (NP,LT,N2) N2=NP
ISN 0035 N2=N2+NBTAY-1
ISN 0036 WRITE (161,160)
ISN 0037 WRITE (161,150) (RA(I),I=NBTAY,N2)
ISN 0038 IF (NP,LE,NXAPRX) GO TO 20
ISN 0040 IF (MAXTRM.LT.2) GO TO 20
C LINEAR TERMS.
ISN 0042 N1=N2+1
C N2 = LOCATION OF LAST DIAGONAL ELEMENT.
ISN 0043 N2=NXAPRX
ISN 0044 IF (N2,GT,NP) N2=NP
ISN 0046 N2=N2+N1-1
ISN 0047 N3=LOCATION OF FIRST OFF-DIAGONAL ELEMENT.
N3=N2+1
ISN 0048 N4=LOCATION OF LAST OFF-DIAGONAL ELEMENT.
ISN 0049 N4=NBR
ISN 0051 N4=N4-2*NXAPRX+N3-1
C LL = LOCATION OF LAST OFF-DIAGONAL ELEMENT - THIS RCH.
ISN 0052 WRITE (161,190)
C II=1
ISN 0053 DO 10 I=H1,H2
ISN 0055 WRITE (I61,150)
C WRITE (161,140) II
ISN 0056 II=II+1
ISN 0057 LL=N3+NXAPRX-II
ISN 0058 IF (LL,GT,N4) LL=N4
ISN 0059 IF (LL,LT,N3) WRITE (161,130) RA(I)
ISN 0060 IF (LL,GT,N3) WRITE (161,130) RA(I),(RA(J),J=H3,LL)
10 H3=LL+1
ISN 0065 CONTINUE
ISN 0066 NBTAY=NBTAY+NBR
ISN 0067 CONTINUE
ISN 0068 N3=II+1
ISN 0069 CONTINUE
ISN 0070 IF (NXAPRX.LT.3) RETURN
C X-VECTOR.
ISN 0071 WRITE (161,50)
ISN 0072 DO 40 I=1,NPTOT
C WRITE (161,50)
ISN 0073 IF (IPAPRX.LT.3) RETURN
ISN 0074 WRITE (161,50)
ISN 0075 WRITE (161,130) (RA(J),J=1,NXAPRX)
C FUNCTION VALUES.
ISN 0076 READ (ISC2) (RA(J),J=1,NF)
ISN 0077 WRITE (161,70)
ISN 0078 WRITE (161,130) (RA(J),J=1,NF)
40 CONTINUE
ISN 0079 RETURN
C -----------------------------------------------
C FORMATS

+VERSION 1.3.0 (01 MAY 80) COPE14 SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 82.14/10.55.24 PAGE 2
**VERSION 1.3.0 (01 MAY 80) COPE14 SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 02.141/10.55.24 PAGE 3**

*a* **OPTIONS IN EFFECT**
- NAME(Main) 
- OPTIMIZE(3) 
- LINECOUNT(60) 
- SIZE(MAX) 
- AUTOUBL(NONE)

*a* **OPTIONS IN EFFECT**
- SOURCE EBCDIC NOLIST NODECK OBJECT NOHMAP NOFORMAT GOSTMT NOXREF NOALC NOABS TERM IBM FLAG(I)

*a* **STATISTICS**
- SOURCE STATEMENTS = 95, PROGRAM SIZE = 2632, SUBPROGRAM NAME =COPE14

*a* **STATISTICS**
- NO DIAGNOSTICS GENERATED

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

2996K BYTES OF CORE NOT USED
SUBROUTINE COPE15 (XV, G, DF, A, ISC, IC, NN, BLU, NX, IOA, M, F, HEW, IGN, H, FNEH, FHOE)

DIMENSION XV(NX), FNEH(N), FHOE(N), A(NN,1), BTAY(NBR,1), DF(1), IGN(1), IC(1), BLU(4,1)

FUNCTION EVALUATION FOR APPROXIMATE OPTIMIZATION.

BY G. N. VANDERPLAATS
NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.

OBJECTIVE.

CALL COPE16 (NX1, XV, NF, FNEH, FHOE, BTAY, NBR, M)
OBJ=-FHEW(IOBA)*SGHOPT

IF (HFOE.EQ.1) GO TO 20

GRADIENT OF OBJECTIVE.

CALL COPE17 (NX1, XV, IOBA, BTAY, NBR, M, DF)

DO 10 I=1, HVE
DF(IJ=-DF(II)*SGHOPT

CONTINUE

IF (HCONA.LE.O) GO TO 80

CONSTRAINTS.

IF (INFO.EQ.2) HAC=0
ICOH=0
DO 70 I=1, NCONA
J=IGNH(I)
GG=FHOE(J)

LAUER SOUND.
IF (BLU(I,J).LT.-1.0E+15) GO TO 40
ICON=ICON+1
G(ICON)=(GG-BLU(3,J))/(BLU(4,J)

IF (INFO.EQ.1) GO TO 40

IS THIS CONSTRAINT ACTIVE OR VIOLATED.

CTI=CT
IF (ISC(ICON,GT.O) CTI=CTL
IF (GG(ICON).LT.CTI) GO TO 40

ACTIVE CONSTRAINT. CALCULATE GRADIENT.

HAC=NAC+1
IC(HACJ=ICOH

IF (INFO.EQ.1) GO TO 40

ACTIVE CONSTRAINT. CALCULATE GRADIENT.

HAC=NAC+1
IC(HACJ=ICOH

IF (INFO.EQ.1) GO TO 40

ACTIVE CONSTRAINT. CALCULATE GRADIENT.

HAC=NAC+1
IC(HACJ=ICOH

IF (INFO.EQ.1) GO TO 40

ACTIVE CONSTRAINT. CALCULATE GRADIENT.

HAC=NAC+1
IC(HACJ=ICOH

IF (INFO.EQ.1) GO TO 40

ACTIVE CONSTRAINT. CALCULATE GRADIENT.

HAC=NAC+1
IC(HACJ=ICOH

IF (INFO.EQ.1) GO TO 40

ACTIVE CONSTRAINT. CALCULATE GRADIENT.
ISN 0046  CTI=CT  00055
ISN 0047  IF (ISC(ICON).GT.0) CTI=CTI  00056
ISH 0049   IF (G(ICON).LT.CTI) GO TO 60  00057
          C ACTIVE CONSTRAINT. CALCULATE GRADIENT.  00058
ISH 0051  NAC=NAC+1  00059
ISH 0052  INC(NAC)=ICON  00060
ISH 0053   NM=M  00061
ISH 0054  IF (ISC(ICON).GT.0) NM=NDV  00062
ISH 0056  CALL COPE17 (HX1,XV,J,BTAY,NBR,HM,A(1,NAC))  00063
ISH 0057      FF=1./BLU(4,1)  00064
ISH 0058   DO 50 K=1,NDV  00065
ISH 0059  50  A(K,NAC)=A(K,NAC)*FF  00066
ISH 0060   60 CONTINUE  00067
ISH 0061   70 CONTINUE  00068
ISH 0062   80 CONTINUE  00069
ISH 0063   RETURN  00070
ISH 0064   END  00071

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

*OPTIONS IN EFFECT* SOURCE EBCDIC NO LIST OBJECT NOFORMAT NOXREF NOALC NOANSF TERM IBM FLAG(I)

*STATISTICS* SOURCE STATEMENTS = 63, PROGRAM SIZE = 1728, SUBPROGRAM NAME = COPE15
*STATISTICS* NO DIAGNOSTICS GENERATED

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

3004K BYTES OF CORE NOT USED
ROUTINE TO EVALUATE FUNCTIONS APPROXIMATED BY TAYLOR SERIES EXPANSION UP TO SECOND ORDER.


**ARGUMENTS.**

**NX** - NUMBER OF INDEPENDENT VARIABLES CONTAINED IN X.
**X** - VECTOR OF DELTA VARIABLES X-X(0). DIMENSIONED X(NX).
**NF** - NUMBER OF FUNCTIONS TO BE EVALUATED.
**FNOM** - INITIAL FUNCTION VALUES ABOUT WHICH TAYLOR SERIES EXPANSION WAS DONE.
**FNEW** - NEW APPROXIMATED VALUES. - OUTPUT. DIMENSIONED FNEW(NF).
**B** - MATRIX OF TAYLOR SERIES COEFFICIENTS.
**B(I,J)** CONTAINS DEL F, I=1,NX.
**B(NX+I,J)** CONTAINS DEL2 TERMS, I = 1,NX*(NX+1)/2.
**MINIMUM DIMENSIONS** - B(M,NF).
**NBR** - DIMENSIONED ROWS OF B.
**M** - TOTAL NUMBER OF COEFFICIENTS CURRENTLY USED.

---

**DATA SET 577COPE16 AT LEVEL 001 AS OF 02/13/81**

**DATA SET 916COPE16 AT LEVEL 001 AS OF 07/10/80**

**SUBROUTINE COPE16 (NX,X,NF,FNOM,FNEW,B,NBR,M)**

**FUNCTIONAL TESTS.**

**DO 50 J=1,NF**

---

**CONSTANT TERM.**

**F=FNOM(J)**

**DO 30**

**SECOND ORDER TERMS.**

**II=II+1**

**DIAGONAL ELEMENTS.**

**DO 10 I=1,NX**

**OFF-DIAGONAL ELEMENTS.**

**DO 20 I=1,NX**

---

**SOURCE EBCDIC NODECK OBJECT NOMAP NOFORMAT GOSTLIT NOXREF NOALC NOANSF TERM IBM FLAG(I)**
**VERSION 1.3.0** (01 MAY 80) COPE16 SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 02.14/10.55.31 PAGE 2

ISH 0020   IP1=I+1  00055
ISH 0021   XX=X(I)  00056
ISH 0022   DO 30 K=IP1,NX  00057
ISH 0023   II=II+1  00058
ISH 0024   IF (II .GT. M) GO TO 40  00059
ISH 0025   30   F=F+B(II,J)*XX*X(K)  00060
ISH 0026   40   CONTINUE  00061
ISH 0027   50   FHEH(J)=F  00062
ISH 0028   RETURN  00063
ISH 0030   END  00064

*OPTIONS IN EFFECT*NAME(HAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOOBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOLIST NOFORMAT GOSTMT NOXREF NOALC NDANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 29, PROGRAM SIZE = 876, SUBPROGRAM NAME = COPE16
*STATISTICS* NO DIAGNOSTICS GENERATED

****** END OF COMPIIATION *****

30K BYTES OF CORE NOT USED
SUBROUTINE COPE17 (NX, X, J, B, NBR, M, GRAD)

**DIMENSIONS**
- `X(NX)`: Number of independent variables contained in `X`.
- `DELV`: Vector of delta variables `X-XIW`, defined in `X`.
- `F`: Function for which gradient information is calculated.
- `B`: Matrix of Taylor series coefficients.

**ARGUMENTS**
- `NX`: Number of independent variables contained in `X`.
- `X`: Vector of delta variables `X-XIW`, defined in `X`.
- `J`: Function for which gradient information is calculated.
- `B`: Matrix of Taylor series coefficients.
- `NBR`: Number of rows of `B`.
- `M`: Total number of coefficients currently used.

**FIRST ORDER TERMS.**

DO 10 I = 1, NX

IF (I .GT. M) GO TO 40

GRAD(I) = B(I, J)

10 CONTINUE

**SECOND ORDER TERMS.**

IF (NX .LT. 2) GO TO 40

NXM1 = NX - 1

DO 30 I = 1, NXM1

IP1 = I + 1

DO 30 K = IP1, NX

II = II + 1

IF (II .GT. M) GO TO 40

GRAD(I) = GRAD(I) + B(II, J) * X(K)

GRAD(K) = GRAD(K) + B(II, J) * X(K)

30 CONTINUE

CONTINUE

**DIAGONAL ELEMENTS.**

DO 20 I = 1, NX

II = II + 1

IF (II .GT. M) GO TO 40

GRAD(I) = GRAD(I) + B(II, J) * X(I)

20 CONTINUE

CONTINUE

**OFF-DIAGONAL ELEMENTS.**

IF (NX .LT. 2) GO TO 40

NXM1 = NX - 1

DO 10 I = 1, NXM1

IP1 = I + 1

DO 10 K = IP1, NX

II = II + 1

IF (II .GT. M) GO TO 40

GRAD(I) = GRAD(I) + B(II, J) * X(K)

GRAD(K) = GRAD(K) + B(II, J) * X(K)

10 CONTINUE

CONTINUE

RETURN

* DATE 82.141/10.55.33 PAGE 1

**MAIN OPTIMIZE (3) AUTODS U NotL1E I, NOAle**

**SOURCE EBCDIC NOLIST NODERE CXXREF FOSTH NODERE NOANCF TERM IBM FLAG(I)**
**VERSION 1.3.0 (01 MAY 80) COPE17 SYSTEM/370 FORTRAN H EXTENDED (ENHANCED) DATE 02.141/10.55.33 PAGE 2

ISN 0027 END

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

*OPTIONS IN EFFECT* SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT NOXREF NOALIGN 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
DATA SET U477COPE18 AT LEVEL 002 AS OF 03/16/81
DATA SET U477COPE18 AT LEVEL 001 AS OF 02/13/81
DATA SET U9185COPE18 AT LEVEL 001 AS OF 07/10/80

SUBROUTINE COPE18 (IDSB, NDVTOT, NCONA, RA, IA, LOCR, LOCI, ARRAY)

**********************************************************************
ROUTINE TO PRINT OPTIMIZATION RESULTS
**********************************************************************

BY G. N. VANDERPLAATS MARCH, 1979
NASA AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.

CGN0N /UIS/ 151, 161
DlMENSION RA(1), IA(1), LOCR(1), LOCI(1), ARRAY(1)

OBJECTIVE FUNCTION AND DESIGN VARIABLES.

WRITE (161,30) IOBJ, ARRAY(IOBJ)

N2=LOCR(2)
N3=LOCR(3)
N5=LOCR(5)
N2=LOCI(2)
DO 10 I=1,NDVTOT

DESIGN VARIABLE NUMBER.

IV=IA(M2)
N2=LOCR(2)+IDV-1
N3=LOCR(3)+IDV-1
M2=M2+1

GLOBAL LOCATION.

IG=IA(I)

MULTIPLIER.

AMULT=RA(N5)
N5=N5+1

LOWER BOUND.

BL=AMULT*RA(N5)
N6=N6+4

IDENTIFICATION NUMBER.

WRITE (161,40) I,IDV,IG,BL,XX,BU
CONTINUE

IF (NCONA.EQ.0) RETURN

CONSTRAINTS.

WRITE (161,50) I,IV,IG,BL,XX,BU

CONTINUE

IF (NCONA.NE.0) RETURN

GLOBAL LOCATION.

IG=IA(M3)
M3=M3+1

LOWER BOUND.

BL=RA(N6)

VALUE.

XX=ARRAY(IG)

UPPER BOUND.

BU=RA(N6+2)

DATA SET U477COPE18 AT LEVEL 001 AS OF 02/13/81
DATA SET U9183COPE18 AT LEVEL 001 AS OF 07/10/80
ISH 0038  JD=ID
ISH 0039  IF (BL.GT.-1.0E+15) ID=ID+1
ISH 0041  IF (BU.LT.1.0E+15) ID=ID+1
ISH 0043  WRITE (161,60) JD,ID,BL,XX,BU
ISH 0044  20 CONTINUE
ISH 0045  RETURN

C  -------------------------------------------------------------------
C  FORMATS
C  -----------------------------------------------------------------
ISH 0046  30 FORMAT (1H1,4X,20HOPTIMIZE RESULTS///5X,12HOBJECTIVE FUNCTION///00064
15X,15HGLOBAL LOCATION,15,5X,14HFUNCTION VALUE,12.5///5X,16HDESIGN00065
2 VARIABLES///14X,5HID. V.,5X,6HGLOBAL,7X,5HLOWER,23X,5HUPPER/8X,2HID00066
3,5X,3HNO.,5X,6HVAR. NO.,6X,5HBOUND,9X,5HVALUE,9X,5HBOUND)
ISH 0047  40 FORMAT (110,17,111,3X,3E14.5)
ISH 0048  50 FORMAT (///5X,16HDESIGN CONSTRAINTS///15X,6HGLOBAL,7X,5HLOWER,23X,500069
1UPPER/9X,2HID,4X,6HVAR. NO.,6X,5HBOUND,9X,5HVALUE,9X,5HBOUND)
ISH 0049  60 FORMAT (110,19,13X,3E14.5)
ISH 0050  END

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOGBL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOLAP 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 COMPIllATION *****

3012K BYTES OF CORE NOT USED
SUBROUTINE SIMCON (NOV, X, VLB, VUB, NCON, ITMAX, IPRINT, ISCAL, DELFUN, QA00002
                   IBFUN, FUN, WK, IWK, UK, IUWK, 03J, G, IER)

00003
------------------------------------------------------------------00004
SIMCON 00005
ROUTINE FOR SIMPLIFIED CONMIN
USAGE
------------------------------------------------------------------00006

BY G. N. VANDERPLAATS
NASA AEROS RESEARCH CENTER, MOFFETT FIELD, CALIF.

PURPOSE

MINIMIZE OBJ AS A FUNCTION OF X(I), I = 1, NOV
SUBJECT TO

G(J) .LE. 0, J = 1, NCON
V LB(I) .LE. X(I) .LE. V UB(I), I = 1, NDV

NOTES

N CON MAY BE ZERO.
VLB(I) IS IGNORED IF VLB(I) .LT. -1.0E+15.
V UB(I) IS IGNORED IF V UB(I) .GT. 1.0E+15.

IF NCON = 0 AND IT IS NOT ESSENTIAL TO LIMIT THE VARIABLES, X(I)
THEN SET V LB(I) = -1.0E+16 AND V UB(I) = 1.0E+16. THIS WILL IMPROVE
THE EFFICIENCY OF THE UNCONSTRAINED OPTIMIZATION.

REFERENCE

G. N. VANDERPLAATS
CONMIN - A FORTRAN PROGRAM FOR CONSTRAINED FUNCTION MINIMIZATION

ARGS

---

INPUT
NOV - NUMBER OF INDEPENDENT DESIGN VARIABLES.
X - ARRAY OF INITIAL DESIGN VARIABLES.
VLB - ARRAY OF LOWER BOUNDS ON X.
V UB - ARRAY OF UPPER BOUNDS ON X.
NCON - NUMBER OF CONSTRAINT VALUES STORED IN ARRAY G.
ITMAX - MAXIMUM PERMISSIBLE ITERATIONS IN CONMIN.
DEFAULT = 20.
IF ITMAX IS NOT SPECIFIED, IT IS TAKEN TO BE 20.
IPRINT - CONMIN PRINT CONTROL. IPRINT = 0, NO PRINT.
DEFAULT = 5.
IPRINT = 5, MAXIMUM PRINT.
SUGGESTED VALUE IS IPRINT = 3.
ISCAL - SCALING PARAMETER. IF ISCAL = 0 NO SCALING IS DONE.
DEFAULT = 1.
ISCAL = 1 THE DESIGN VARIABLES ARE SCALING DURING
OPTIMIZATION. RECOMMENDED ISCAL = 1.
DEL FUN - CONVERGENCE TOLERANCE ON FRACTIONAL CHINE IN OBJECTIVE
FUNCTION. IF OBJ DOES NOT CHANGE BY MORE THAN DELFUN
FOR THREE CONSECUTIVE ITERATIONS, OPTIMIZATION IS TERMINATED.

IF DELFUN = 0 IS INPUT, DELFUN = 0.0001 IS USED.

FOR CONVERGENCE TOLERANCE ON ABSOLUTE CHANGE IN OBJECTIVE FUNCTION, IF OBJ DOES NOT CHANGE BY CODE THEN DELFUN IS IN MAGNITUDE FOR THREE CONSECUTIVE ITERATIONS, OPTIMIZATION IS TERMINATED.

IF DABFUN = 0 IS INPUT, DABFUN = 0.0001ABS(INITIAL OBJ) IS USED.

NAME OF EXTERNAL SUBROUTINE WHICH EVALUATES OBJECTIVE AND CONSTRAINT FUNCTIONS.

REAL WORK ARRAY.

DIMErNSION OF IXK ARRAY.

MINIMUM PERMISSIBLE DIMENSION.

NIXK = 16*NDV + 2*(NDV**2) + 2*NCON + 16

SPECIAL CASE - NCON = 0, VLB(I).LT.-1.0E+15 AND VUB(I).GT.1.0E+15 FOR I = 1,NDV.

NIXK = 5*NDV + 11

INTEGER WORK ARRAY.

DIMENSION OF IXK ARRAY.

MINIMUM PERMISSIBLE DIMENSION.

NIXK = 5*NDV + NCON + 6

SPECIAL CASE - NCON = 0, VLB(I).LT.-1.0E+15 AND VUB(I).GT.1.0E+15 FOR I = 1,NDV.

NIXK = 3

ARRAY CONTAINING DESIGN VARIABLES DEFINING THE OPTIMUM.

OPTIMUM OBJECTIVE FUNCTION VALUE.

ARRAY OF CONSTRAINT VALUES AT THE OPTIMUM.

REQUIRED DIMENSION = NCON + 2*NDV.

SPECIAL CASE - NCON = 0, VLB(I).LT.-1.0E+15 AND VUB(I).GT.1.0E+15 FOR I = 1,NDV.

DIMENSION = 1.

ERROR CODE.

IER = 0, NO STORAGE ERROR.

IER = 1, ARRAY WK OR IXK IS NOT DIMENSIONED LARGE ENOUGH.

EXTERNAL Routines to Evaluate Objective and Constraint Functions.

NDV, X, NCON, OBJ, G

INPUT

NUMBER OF INDEPENDENT DESIGN VARIABLES.

ARRAY CONTAINING CURRENT VALUES OF THE NDV DESIGN VARIABLES.

DIMENSION = NDV + 2.

NUMBER OF CONSTRAINT VALUES STORED IN G.

NCON MAY BE ZERO.

... OUTPUT

OBJECTIVE FUNCTION ASSOCIATED WITH X.

ARRAY CONTAINING THE NCON CONSTRAINT VALUES ASSOCIATED WITH X. IF NCON = 0, G IS NOT CALCULATED.

DIMENSION = NCON + 2*NDV IF NCON.GT.0.

DIMENSION = 1 IF NCON = 0.

EXTERNAL
C INITIALIZE COMM PARAMETERS

C ---

ISN 0006 NDV1=NDV
ISN 0007 NCON1=NCON
ISH 0008 IPRINT=IPRINT
ISH 0009 ITHMX=ITHMX
ISH 0010 DELFUN=DELFUN
ISH 0011 DABFUN=DABFUN
ISH 0012 CI=0.
ISH 0013 CTMIN=0.
ISH 0014 CTL=0.
ISH 0015 CTLAX=0.
ISH 0016 ALPHAX=0.
ISH 0017 AGOBJ=0.
ISH 0018 THETA=0.
ISH 0019 NFDG=0.
ISH 0020 LINOBJ=0.
ISH 0021 ITHAX=0.
ISH 0022 ICNO=0.
ISH 0023 NSCAL=NDV+1
ISH 0024 IF (ISCAL.EQ.0) NSCAL=0
ISH 0026 NSIDE=0
ISH 0027 DO 10 I=1,NDV
ISH 0028 IF (VLB(I).GT.-1.0E+15.OR.VUB(I).LT.1.0E+15) NSIDE=1
ISH 0030 10 CONTINUE
ISH 0031 FOCH=.001
ISH 0032 FDCH=.001
ISH 0033 IF (IPRINT.LT.1) GO TO 30

C ---

ISN 0035 WRITE (6,170)
ISH 0036 WRITE (6,160) NDV,NCON,ITHMX,IPRINT,ISCAL,DELFUN,DABFUN
ISH 0037 WRITE (6,150)
ISH 0038 DO 20 I=1,NDV
ISH 0039 20  WRITE (6,200) I,VLB(I),X(I),VUB(I)
ISH 0040 30 CONTINUE

C ---

ISH 0041 IER=1
ISH 0042 N1=NDV+2
ISH 0043 IF (NCON.EQ.0.AND.NSIDE.EQ.0) GO TO 40
ISH 0044 N2=NCON+2*NDV
ISH 0045 N3=N1
ISH 0046 N4=N1
ISH 0047 N5=N1
ISH 0048 N5=2*N1
ISH 0049 GO TO 50
ISH 0050 40 CONTINUE

C SPECIAL CASE. NCON = NSIDE = 0.
ISH 0051 N2=1
ISH 0052 N3=1
ISH 0053 N4=NDV
ISH 0054 N5=1
ISH 0055 50 CONTINUE
ISH 0056 NREQR=3*N1+2*N2+N3*(N1+N3)+N4
C FIND MAXIMUM POSSIBLE N5.

N3SAV=N3
N21=N2+1
DO 60 I=II,N21
II=II+1
N4=N3
IF (N4.LT.NDV) N4=NDV
N5=2*N4
NREQR=3*N1+2*N2+3*N2*(N1+N3)+N4
NREQI=N2+N3+N5
IF (NREQR.GT.NWK.OR.NREQI.GT.NIWK) GO TO 70
N3SAV=I
60 CONTINUE
N3=N3SAV
N4=N3
IF (N4.LT.NDV) N4=NDV
70 CONTINUE
C STORAGE ALLOCATION

NNSCAL=1
NDF=NNNSCAL+N1
NA=NDF+N1
NS=NA+N4+N3
NG1=NS+N1
NG2=NG1+N2
NC=NG2+N3+N3
NISC=1
NIC=IC+1
NHS=IC+1
NRS=HIC+N3
NREQR=NC+N4-1
NREQ1=NS1+N5-1
IF (NREQR.GT.NWK.OR.NREQ1.GT.NIWK) GO TO 130
IF (NCON.LE.0) GO TO 100
C DEFINE ISC ARRAY SO ALL CONSTRAINTS ARE NONLINEAR.
N=NISC
DO 90 I=1,NCOND
90 I=I+1
N=N+1
100 CONTINUE
N=I
IF (IPRINT.LT.1) GO TO 110
WRITE (6,200) NWK,NREQRl,NREQI,NIWK,NREQIl,NREQI
110 CONTINUE
C OPTIMIZATION

IGOTO=0
120 CONTINUE
CALL COMIN (X,VLB,VUB,G,WK(NNNSCAL),WK(NDF),WK(NA),WK(NS),WK(NG1))
C HAS STORAGE BEEN EXCEEDED.  00233

C EVALUATE OBJECTIVE AND CONSTRAINTS.  00235

CALL FUN (HDV,X,NCON,OBJ,G)  00236

C OPTIMIZATION COMPLETE.  00239

IER=0  00240

RETURN  00241

C ------------------------------------------------------------------

C REQUIRED STORAGE EXCEEDS AVAILABLE STORAGE  00243

C ------------------------------------------------------------------

WRITE (6,220)  00247

WRITE (6,230) NWK,NREQR,NREQR,HIWK,NREQI,NREQI  00248

RETURN  00249

C ------------------------------------------------------------------

C FORATS  00265

C ------------------------------------------------------------------

FORMAT (11H1,///////12X,29(2H*,//12X,1H*,55X,1H*/12X,1H*,22X,11HS 00273

1 M C O N,22X,1H*,12X,1H*,55X,1H*/12X,1H*,10X,1SHFORTRAN PROGRAM FCOO274

2R,16X,1H*/12X,1H*,55X,1H*/12X,1H*,16X,23HSIMPLIFIED COMM USAGE,100275

3X,1H*/12X,1H*,55X,1H*/12X,1H*,29H*  )  00276

FORMAT (///17X,16MINPUT PARAMETERS///17X,NH NUMBER OF DESIGN VARIABLES77

1BLES, NDV =,15/17X,16NH NUMBER OF CONSTRAINTS, NCON =,15/100278

27X,16RMAXIMUM ITERATIONS, ITMAX =,15/17X,16HPRINT CONTENOLO279

3X, 3PRINT =,15/17X,16HSCALING PARAMETER, ISCA=0280

4L =,15/17X,16HCONVERGENCE CRITERIA, DELFUN =,E12.5/17X,3HC=0281

ONVERGENCE CRITERIA, DASFUN =,E12.51  00282

FORMAT (///30X,16HDESIGN VARIABLES/24X,16HLOWER,7X,7HINITIAL,5X,SHPO0283

1PER/24X,5THING,4X,5HVALUE,6X,5HBOUND/19X,1H1,4X,6HVLB(I),8X,4HHX100284

21,6x,6HVB(I))  00285

FORMAT (15X,15,3E12.5)  00286

FORMAT (///19X,3HSTORAGE REQUIREMENTS FOR WORK ARRAYS)  00287

1STORAGE IN WORK ARRAY KK OR IKK IS INSUFFICIENT)  00289

FORMAT (/21X,SHARRAY,3X,9HDIMENSION,2X,ENEEDED,3X,4HUSED/20X,7HDO290
FORMAT ('/22X,27HOPTIMIZATION CANNOT PROCEED/22X,29HRETURNING TO 00292
ICALLING PROGRAM//12X,47(IH*)) 00293
ISN 0157 250 FORMAT (/12X,66HAVAILABLE STORAGE FOR GRADIENT INFORMATION EXCEEDS 00294
1AVAILABLE STORAGE//12X,56HARRAY,14X,7HI:K(N1:K),5X,9HI:K(N1:K)//12X,9H00295
2DIMENSION,15X,16.X,16/12X,19HSUGGESTED DIMENSION,15X,16.X,16) 00296
ISN 0158 END 00297
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTODSL(NONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOCAP NOFORMAT GOSTMT NOXREF NOLIC 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
**Copies**—Program for Engineering Synthesis.

**COMMON**

- `/COPESI/
  TITLE(20)`
- `/COPES2/
  RA(5000).IA(1000)`
- `/COPES3/
  SGNOPT, TCAlC, NSV, NSCBJ, NCCNA, N2VX, MVVX, NZVY, N00014`  
- `/COPES4/
  SNSCHT, HACLC, IO3J, NSB0009`  
- `/COPES5/
  NNSCHT, HACLC, IO3J, NSB0009`  
- `/COPES6/
  ARRAYS(1500)`
- `/COPES7/
  150, 161`

**Comment**

- "BY G. N. VANDEPLAATS OCT., 1974."
- "NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.

**Source**

- "ASA-AMES RESEARCH CENTER • MOFFETT FIELD, CALIF."
- "NCAlC OPTIONS:"
  1. READ ALL INPUT AND STOP.
  2. SINGLE PASS ANALYSIS.
  3. OPTIMIZATION.
  4. SENSITIVITY - Z = F(X).
  5. TWO VARIABLE FUNCTION SPACE - Z = F(X,Y).
  6. OPTIMUM SENSITIVITY.
  7. ANALYSIS/OPTIMIZATION USING APPROXIMATION TECHNIQUES.

**Print**

- "DATE 82.141/10.55.45 PAGE 1"
IF (NDRA1.LE.NDRA.AND.NDIA.1.LE.NDIA) GO TO 10
WRITE (I61,360) NDRA,NDRA1,NDIA,NDIA1
GO TO 340
CONTINUE
C READ USER INPUT.
ICALC=1
CALL ANALIZ (ICALC)
IF (ICALC.LT.1.OR.NCALC.GT.6) GO TO 340
*************
ICALC=2
JCALC=3
IF (ICALC.NE.2.AND.NCALC.LT.5) GO TO 60
ABS(X(I).GT.0 OVER-RIDE USER INPUT OF DECISION VARIABLES FOR OPTIMIZATION.
DO 40 I=1,NDV
XX=ABS(X(I))
DO 20 J=I,NDVTOT
H2=IA(J)
IF (XX.LT.1.0E-10) GO TO 30
ARRAY(IIN1=RA(I)*RAtN5
N5=N5+1
XX=RA(I)/ARRAY(I)
CONTINUE
M2=LOCII(2)
N5=LOCRI(5)
DO 50 I=1,NDVTOT
II=IA(M2)
M=IA(I)
ARRAY(NI=RA(NI*RA(N5)
115=N5+1
M2=M2+1
CONTINUE
IF (NCALC.NE.3.AND.NCALC.NE.5) GO TO 80
TRANSFER NOMINAL VALUES OF SENSITIVITY VARIABLES TO ARRAY.
M16=LOCII(16)
M17=LOCII(17)
N15=LOCRI(15)
DO 70 I=1,NSV
91=IA(M16)
M16=M16+1
CONTINUE
IF (NCALC.LT.6) GO TO 290
TRANSFER DESIGN VARIABLES TO ARRAY.
INITIALIZE FOR APPROXIMATE ANALYSIS/OPTIMIZATION

C ANALYZE INPUT DEFINES AN X-VECTOR.

C

IF (HPA.EQ.0) GO TO 130

C

ANALYZE INPUT DEFINES AN X-VECTOR.

C

M5=LOCI(5)

C

N3=LOCI(23)

C

DO 120 I=1,NXAPRX

C

J=IA(MS)

C

IS THIS A DESIGN VARIABLE.

C

DO 99 K=1,NVTOT

C

KK=K

C

IF (IA(K).EQ.J) GO TO 100

C

CONTINUE

C

NO.

C

AMULT=1.

C

GO TO 110

C

YES.

C

100 K=LOCR(K)+KK-1

C

AMULT=RA(K)

C

RA(N3)=ARRAY(J)/AMULT

C

N5=M5+1

C

N23=N23+1

C

CONTINUE

C

IF (TLPS.GT.0 .OR. TPFS.GT.0) GO TO 190

C

ONLY ONE DESIGN VECTOR IS AVAILABLE. CREATE A SECOND X-VECTOR

C

SO OPTIMIZATION CAN PROCEED.

C

H5=LOCR(5)

C

H23=LOCR(23)

C

DO 120 I=1,NXAPRX

C

N5=M5+1

C

GLOBAL LOCATION.

C

IG=IA(M5)

C

PROPOSED X-VALUE.

C

XX=1.1*RA(U23)

C

IF (ABS(XX).LT.1.0E-10) XX=.1

C

IS THIS A DESIGN VARIABLE.

C

H5=LOCR(5)

C

DO 140 J=1,NVTOT

C

J=J

C

AH=RA(N5)

C

IF (IA(J).EQ.IG) GO TO 150

C

NO.

C

N5=N5+1

C

CONTINUE

C

GO TO 170

C

CONTINUE

C

YES. WHICH DESIGN VARIABLE IS IT.

C

ID=LOCI(2)+JJ-1

C

ID=IA(ID)

C

INSURE XX IS WITHIN BOUNDS.

C

N2=LOCR(2)+ID-1

C

N3=LOCR(3)+ID-1

C

BL=RA(N2)*ABS(AH)

C

BU=RA(N3)*ABS(AH)

C

IF (BL.LE.BU) GO TO 160

C

SAV=BL

C

GO TO 171

C

BL=BU

C

GO TO 172
IF (XX.LT.BL) XX=BL
IF (XX.GT.BU) XX=BU
IF (ABS(DX).LT.1.0E-6) XX=I.0E-6
IF (ABS(XX).LT.1.0E-6) XX=.OO1
MAIN24J=XX
N23=123+1
N24=124+1
CONTINUE
REIND ISC2
NPSA=NPS+NPA
IF (NPSA.EQ.O) GO TO 250
IF (NPS.EQ.O) GO TO 210
READ X-VECTORS.
NXI=LOCRI(23)+NPA*NAPRX
DO 200 J=1,NPS
IX=IX+NXAPRX-1
READ (ISCR2) (RA(I),I=NXI,IXJ)
NXI=NXI+NAPRX
CONTINUE
IF (NPFS.LE.O) NPSA=NPFS
NXI=LOCRI(23)
NY=NXI+NAPRX
DO 240 J=1,NPSA
TRANSFER X-VALUES.
MS=LOCRI(5)
II=NXI
DO 220 I=I,NXAPRX
II=1A(MS)
ARRAY(I)=RA(I)
MS=MS+1
CONTINUE
II=II+1
ANALIZE.
NAN2=NAN+1
CALL ANALIZE(ICALC)
M5=LOCRI(6)
II=NY
DO 230 I=1,NF
II=I4(M5)
RA(I)=ARRAY(I)
II=II+1
M5=M5+1
CONTINUE
II=II+1
C ANALIZE.
PUT FUNCTION VALUES IN Y-ARRAY.
NY=NY+1
C READ X AND Y VECTORS.
NXI=NXI+NAPRX
NY=NY+1
CONTINUE
NXI=NXI+NAPRX-1
NY=NY+1
C X-VECTOR.
READ (ISCR2) (RA(I),I=NXI,NXJ)
C Y-VECTOR.
NXI=NXI+NXAPRX 00232
NY=NY+NYAPRX*NPTOT 00233
DO 280 I=1,NPTOT 00234
NXIJ=NXI+NXAPRX-1 00237
NYJ=NY+IIF-1 00241
C X-VECTOR.
WRITE (ISCR2) (RA(J),J=HXI,NXIJ) 00243
C V-VECTOR.
WRITE (ISCR2) (RA(J,J=NY,HYJI 00244
NXI=NXIJ+1 00247
HY=NYJ+1 00247
CONTINUE 00248
GO TO (300,300,310,320,310,330),NCALC 00249
C ONE ANALYSIS 00250
300 NAN2=NAN2+1 00251
CALL AHALIZ (ICALC) 00252
NAN3=NAH3+1 00255
CALL ANALIZ (JCALC) 00256
IF (NCALC.EQ.1) GO TO 340 00257
C OPTIMIZATION. 00258
CALL COPE02 (ARRAY,RA,IA,NARRAY,NDRA,NDIA) 00262
C OUTPUT RESULTS.
CALL COPE18 (OBJ,IVOTOT,NCONA,RA,IA,LOCR,LOCII ARRAY) 00264
CALL ANALIZ (JCALC) 00266
GO TO 340 00267
C SENSITIVITY ANALYSIS 00268
310 CONTINUE 00271
C ARRAY STARTING LOCATIONS.
N1=LOCR(15) 00272
NSOBJ 00273
N2=LOCR(15) 00274
N3=LOCR(16) 00275
N4=LOCR(23) 00276
CALL COPE04 (ARRAY,RA,HRA,IA,IA(1H1),IA(1H2),IA(1H3),IA(1H4),RA(1H5),NNSN=NSV 00277
NNSN=IA(1H6) 00278
NNSN=IA(1H7) 00279
CALL COPE02 (ARRAY,RA,IA,NARRAY,NDRA,NDIA) 00280
C OUTPUT RESULTS.
CALL COPE05 (RA,IA,NRA,NRA,NDIA,ISCR1) 00281
GO TO 340 00282
*VERSION 1.3.0 (01 MAY 80)  MAIN  SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 80.161/10.55.45  PAGE 6

ISH 0227  320 CONTINUE 00291
C -------------------------------00292
C TWO VARIABLE FUNCTION SPACE 00292
C -------------------------------00294
ISH 0228  CALL COPE06 (ARRAY,RA,IA,HARRAY,HDIRA,HTIA) 00295
ISH 0229  CALL COPE07 (RA,IA,HARRAY,NDIA,ISCI) 00297
ISH 0230  GO TO 340 00298
ISH 0231  330 CONTINUE 00299
C -------------------------------00300
C APPROXIMATE ANALYSIS/OPTIMIZATION. 00301
C -------------------------------00302
ISH 0232  CALL COPE09 00303
C OUTPUT RESULTS. 00304
ISH 0233  CALL COPE14 (NXAPRX,NF,NPTOT,RA,IA,LOCR,LOCI,TITLE,NINH,NBV,IPAPRX) 00305
ISH 0234  IF (KHAX.LT.0) GO TO 340 00306
ISH 0235  CALL COPE18 (IOBJ,NVOTOT,NCONA,RA,IA,LOCR,LOCI,ARRAY) 00307
ISH 0236  NAND=NAND+1 00308
ISH 0237  CALL ANALIZ (JCALC) 00309
ISH 0238  CALL ANALIZ (JCALC) 00310
ISH 0239  340 CONTINUE 00311
ISH 0240  WRITE (I61,350) NAND,NAND 00312
ISH 0241  WRITE (I61,350) NAND,NAND 00313
ISH 0242  WRITE (I61,350) NAND,NAND 00314
ISH 0243  CALL MYTIME (ITIME) 00315
ISH 0244  ITIMEC = 9 00316
ISH 0245  WRITE (I61,350) ITIMEC, ITIME 00317
ISH 0246  STOP 00318
C -------------------------------00319
C FORMATS 00320
C -------------------------------00321
ISH 0247  350 FORMAT (1IH1,4X,23HPROGRAI1 CALLS TO AUALIZ/EX,5HICALC,3X,5HCALLS/100322
ISH 0248  360 FORMAT (/5X,6HREQUIRED STORAGE FOR ARRAY RA OR IA EXCEEDS DIMENS00324
ISH 0249  END 00325
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOBSK(LNONE)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 246, PROGRAM SIZE = 3960, SUBPROGRAM NAME = MAIN
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPILATION *****

2968K BYTES OF CORE NOT USED
APPENDIX C

REVISED FORTRAN FOR SUPERHYBRID BLADE DEMONSTRATION CASE
DATA SET U4920STAEB AT LEVEL THP AS OF 05/21/82
DATA SET U4920STAEB AT LEVEL THP AS OF 01/20/82
DATA SET U4920STAEB AT LEVEL THP AS OF 10/16/81
DATA SET U4920STAEB AT LEVEL THP AS OF 03/26/81
DATA SET U4920STAEB AT LEVEL THP AS OF 03/14/81

SUBROUTINE STAEBL 00005
COMMON /BLKCH/ XP(3,53,21), HP(21), IND(21), INP 00006
COMMON /BLK 2/ HOACR, HOACRI, HSTA, HSTA1, POISH, MS2, MTS3 00007
COMMON /BLK 4/ QMP, QMPC, BR, PE, TERN, NDS 00008
COMMON /BLK 7/ AREA(21), CF(21), AA(21), AKG(21), IT(7,21) 00009
COMMON /BLK12/ XBARX(21), XMAX(21), XMINT(21), YBARY(21), 00010
1 TLYA(21) 00011
COMMON /BLKAA/ ALPH(21), XSSC(21), YSSC(21), XX(21), YY(21) 00012
COMMON /BLK/ SPB(21), ITLE(21), VARI(20), TMX(21), HALPHA(21) 00013
COMMON /BLK A/ ALPHA(21), HA(21), HMINT(21), HAMINT(21), 00014
1 HINERT(21) 00015
COMMON /BLK 9/ DELT(21), DELTAZ(21), P(21), R(21), SHRT(21) 00016
COMMON /INPUT/ XSAVE(1000), YSAVE(1000), ZSAVE(1003), TSAVE(1003) 00017
COMMON /GC/ XSCG(V(21)), YSCG(V(21)), ZSCG(V(21)) 00018
1 ALSAVE(21), FOLAR(21), ASAVE(21) 00019
COMMON /ANLS3/ DLED, DED, DRODD, DTPO, TTO, TTD, TLO, NCD 00020
1, HCK, TSKIN, TCENTR, PTB, PGE 00021
DIMENSION YH(53,21), XM(53,21), TM(53,21), T(120), T1(15,21) 00022
1, XM(50,21), Y(50,21), Z(50,21) 00023
3, XSCG(21), YSCG(21), ZSCG(21), ZSCG(21), 00024
3, XG(50,21), YG(50,21), ZG(50,21) 00025
4, XSCG(21), YSCG(21), ZSCG(21) 00026
5, XG(50,21), Y(50,21), Z(50,21) 00027
REAL*4 A(51), B(51), C(51), D(51), E(51), F(51), G(51), 00028
1, A(51), B(51), C(51), D(51), E(51), F(51), G(51), 00029
2, D(51), E(51), F(51), G(51) 00030
REAL*4 X(51), Y(51), Z(51), TM(51), TTHL(51) 00031
REAL*4 X(51), Y(51), Z(51), TM(51), TTHL(51) 00032
C CALCULATE MEAN Y VALUE AND THICKNESS NORMAL TO THE CHORD 00033
CISN 0019 NBR = 1 00036
ISN 0020 NPS = NP(1) 00037
ISN 0021 DO 5 N = 1,NPS 00038
ISN 0022 XP(1,N,1) = XP(1,N,2) 00039
ISN 0023 XP(1,N,1) = XP(1,N,2) 00040
ISN 0024 XP(3,N,1) = XP(3,N,2) 00041
ISN 0025 5 CONTINUE 00042
ISN 0026 DO 20 I = NBR,NBR 00043
ISN 0027 NPS = NP(I) 00044
ISN 0028 DO 10 N = 1,NPS 00045
ISN 0029 YM(N,1) = (XP(2,N,1) + XP(3,N,1)) / 2.0 00046
ISN 0030 XM(N,1) = XP(1,N,1) 00047
ISN 0031 TIC(N,1) = XP(2,N,1) - XP(3,N,1) 00048
ISN 0032 10 CONTINUE 00049
ISN 0033 20 CONTINUE 00050
C CALCULATE THICKNESS NORMAL TO MEANLINE 00051
CISN 0034 DO 40 I = NBR,NBR 00052
C
*VERSION 1.3.0 (01 MAY 80) STAEBL SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 82.141/10.58.59  PAGE 2

ISH 0035  NPS = NP(I)-1 00049
ISH 0036  TNL(I,I) = 0.0 00050
ISH 0037  TNL(NPS+1,I) = 0.0 00051
ISH 0039  DO 30 N = 2,NPS 00052
ISH 0040  T1 = TAH(N1(I),I) - YH(N1(I),I) ) 00053
ISH 0041  T2 = TAH(N2(I),I) - YH(N2(I),I) 00054
ISH 0042  THT = ( T1 + T2 ) / 2.0 00055
ISH 0043  TNL(I,N) = TNL(I,N) * COS(THT) 00056
ISH 0044  CONTINUE 00057
ISH 0045  NSTN = NSTA - NER 00058
ISH 0046  DO 45 I = NSTA,NSTN 00059
ISH 0047  WRITE (6,900) YH(N1(I),I),H=1,NPS 00060
ISH 0048  WRITE (6,900) XH(I,N),I=1,NPS 00061
ISH 0049  WRITE (6,900) YH(N2(I),I),N=1,NPS 00062
ISH 0050  WRITE (6,900) XH(I,N),N=1,NPS 00063
ISH 0051  CONTINUE 00064
ISH 0052  DO 201 K = 1,NPS 00065
ISH 0053  XM(K) = XH(K,I) 00066
ISH 0054  XM(NP(I)) = XM(K,I) 00067
ISH 0055  CALL PMFIT FOR CURVE FIT 00068
ISH 0056  CALL PMFIT(XM,YM,NPS,NP,A,B,C,D,A1,B1,C1,D1,S1) 00069
ISH 0057  CALL PMFIT(XM,YM,NPS,NP,A2,B2,C2,D2,A3,B3,C3,D3,S2) 00070
ISH 0058  SARC = S1/NPS / NP 00071
ISH 0059  MXH SEARCH S1 ARRAY FOR INTERVAL VALUE 00072
ISH 0060  NODE = 1 00073
ISH 0061  NARC = SARC / 2. 00074
ISH 0062  IF(NARC .LE. S1(N)) GO TO 220 00075
ISH 0063  N = N + 1 00076
ISH 0064  GO TO 210 00077
ISH 0065  DIST = (ARC1 - S1(N-1)) / (S1(N) - S1(N-1)) 00078
ISH 0066  SONE = S1(N) - S1(N-1) * DIST 00079
ISH 0067  STN0 = S2(N) - S2(N-1) * DIST 00080
ISH 0068  CALL CUBIC(A(N-1),B(N-1),C(N-1),D(N-1),SONE,YH(NODE)) 00081
ISH 0069  CALL CUBIC(A(N-1),B(N-1),C(N-1),D(N-1),STNO,TTLF(NODE)) 00082
ISH 0070  CALL CUBIC(A(N-1),B(N-1),C(N-1),D(N-1),SONE,YH(NODE)) 00083
ISH 0071  CALL CUBIC(A(N-1),B(N-1),C(N-1),D(N-1),STNO,TTLF(NODE)) 00084
DO 300 K = 1,NPF
   XHF(K,I) = XXMF(K)
   YHF(K,I) = YYMF(K)
   WRITE(6,902) I,K,XHF(K,I),YHF(K,I)
   FORMAT(5X,I,3X,2E12.5)
   TMLF(K,I) = TTHLF(K)
C
C   FILL IN THE XMF,YMF,TMLF VECTORS WITH THE INTERPOLATED VALUES
C
DO 230 K = 1,NPF
   XI1F(K,I) = XXMF(K)
   YXI1F(K,I) = YYMF(K)
   WRITE(6,902) I,K,XXMF(K),YYMF(K)
   FORMAT(5X,I,3X,2E12.5)
C
C   TRANSLATE COORDINATES TO AN ENGINE AXIS SYSTEM
C
DO 130 I = NBR,NSTA
   XCHORD = XMB(I)
   XSCLE = XCHORD - XSCII
   XSCCGII) = XSCLE - (XCHORD - XBARXII))
   YSCCGII) = YSCII - (XCHORD - XBARYII))
   DO 120 N = 1,NPF
      XCGIN,I) = XMF(N,I) - XCHORD + XBARX(I)
      YCGIN,I) = YMF(N,I) - YBARY(I)
   120 CONTINUE
   130 CONTINUE
C
C   SHIF T XY PLANE TO ENGINE AXIS YZ
C
DO 150 I = NBR,NSTA
   XSCG(I) = R(I)
   YSCG(I) = YSCG(I)
   ZSCG(I) = XSCG(I)
   DO 140 N = 1,NPF
      XG(N,I) = R(I)
      YG(N,I) = YCG(N,I)
      ZG(N,I) = XCG(N,I)
   140 CONTINUE
   150 CONTINUE
C
C   ROTATE THRU ALPHA CHORD - 90.0
C
DO 170 I = NBR,NSTA
   ALPHA(I) = 90.0 * .0174533
   ANG = ALPHA(I) + 90.0 * .0174533
   EH = COS(ANG)
   EN = SIN(ANG)
   XSCR(I) = XSCG(I)
   YSCR(I) = YSGG(I) * EM + EN * ZSCG(I)
   ZSCR(I) = ZSGG(I) - EM * YSGG(I)
   WRITE(6,900) ANG, EH, EN, XSCR(I), YSCR(I), ZSCR(I), AREA(I)
   170 CONTINUE
C
DO 160 N = 1,NPF
   XN(I) = XG(N,I)
   160 CONTINUE
*VERSION 1.3.0 (01 MAY 80) STAEGL SYSTEM/370 FORTRAN H EXTENDED (ENHANCED)  DATE 02.14/10.56.59  PAGE  4

ISH 0111  Y(N,I) = YG(N,I)* EM + EN * ZG(N,I)  00167
ISH 0112  Z(N,I) = EM * ZG(N,I) - EN * YG(N,I)  00168
ISH 0113  160 CONTINUE  00169
ISH 0114  170 CONTINUE  00170
C  DO 175 I = NCR,NSTA,NSTND
C  WRITE (6,900) (XCG(N,I),N=1,NPF)
C  WRITE (6,900) (YCG(N,I),N=1,NPF)
C  WRITE (6,900) (ZCG(N,I),N=1,NPF)
C  WRITE (6,900) (ZG(N,I),N=1,NPF)
C  WRITE (6,900) (X(N,I),N=1,NPF)
C  WRITE (6,900) (Y(N,I),N=1,NPF)
C  WRITE (6,900) (Z(N,I),N=1,NPF)
C  WRITE (6,900) (THLF(N,I),N=1,NPF)
C  175 CONTINUE  00180
ISH 0115  IJ = 1  00181
ISH 0116  IF(IJ .EQ. 1) GO TO 811  00182
C  TO3902 GENERATED NODES AND THICKNESSES  00184
ISH 0118  DO 800 I = NCR,NSTA  00185
ISH 0119  DO 801 J = 1,NPF  00186
ISH 0120  801 READ(5,802) Y(J,I),Z(J,I)  00187
ISH 0121  802 FORMAT(/,32X,2F3.0)  00188
ISH 0122  DO 805 J = 1,NPF  00189
ISH 0123  READ(5,803) P1,P2  00190
ISH 0124  803 FORMAT(24X,F8.0)  00191
ISH 0125  805 THLF(J,I) = (PI + P2) / 2.  00192
C  WRITE(6,810) I  00193
C  FORMAT(5X,'TO39 NODS AND THICKNESSES FOR SECTION ',I5)  00194
C  WRITE(6,900) (X(N,I),N=1,NPF)  00195
C  WRITE(6,900) (Y(N,I),N=1,NPF)  00196
C  WRITE(6,900) (Z(N,I),N=1,NPF)  00197
C  WRITE(6,900) (THLF(N,I),N=1,NPF)  00198
ISH 0126  800 CONTINUE  00199
ISH 0127  811 CONTINUE  00200
C  I = 0  00201
ISH 0128  K = 0  00202
ISH 0129  DO 190 I = NBR,NSTA  00203
ISH 0130  K = K + 1  00204
ISH 0131  XSCSV(K) = XSCR(I)  00205
ISH 0132  YSCSV(K) = YSCR(I)  00206
ISH 0133  ZSCSV(K) = ZSCR(I)  00207
ISH 0134  C  ALSAVEK) = ALPHA(I)  00208
ISH 0135  P0LARI(K) = XIMH(I) + XIMAX(I)  00209
ISH 0136  ASAVEK) = AREAI)  00210
ISH 0137  C  DO 160 N = 1,NPF  00211
ISH 0138  J = J +1  00212
ISH 0139  XSAVE(J) = X(N,I)  00213
ISH 0140  YSAVE(J) = Y(N,I)  00214
ISH 0141  ZSAVE(J) = Z(N,I)  00215
ISH 0142  TSAVE(J) = THLF(N,I)  00216
ISH 0143  C  WRITE (7,901) XSAVE(J), YSAVE(J), ZSAVE(J), TSAVE(J)  00217
ISH 0144  180 CONTINUE  00218
ISH 0145  190 CONTINUE  00219
ISN 0146     K = 0
ISN 0147     DO 200 I = NSR,NST,
           00226  
           00227
ISN 0148     K = K + 1
           00228
           00229
           00230
           00231
           00232
           00233
           00234
           00235
           00236
           00237
           00238
           00239
           00240
           00241
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(3) LINECOUNT(60) SIZE(MAX) AUTOOBL(OBL4)
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NCHAP NOFORMAT GOSTTH NOREF NOALC NOANSF TERM IBM FLAG(I)
*STATISTICS* SOURCE STATEMENTS = 157, PROGRAM SIZE = 141976, SUBPROGRAM NAME =STAEBL
*STATISTICS* NO DIAGNOSTICS GENERATED
***** END OF COMPIlATION *****
SUBROUTINE INPUT (NR,NC)

MATERIAL PROPERTIES AND GEOMETRY INPUT

COMMON /ZCOEF/ E11(7),E22(7),E33(7),G12(7),G23(7),G13(7),
V10(7),V13(7),V23(7)

COMMON /PLY/ PLY(21,25,7),THETA(7),RHO(7)

COMMON /FAIL/ X1T(7),X1C(7),X2T(7),X2C(7),S6P(7),S6M(7),TSAI(6,25)

WRITE (8,1)

FORMAT(SX,'IS THIS AN ISO1'0TPIC BLADE,1=YES')

READ(8,*) ISO

IF(ISO .NE. 1) GO TO 2

WRITE(8,3)

3 FORMAT(5X, 'INPUT-E(PSI),V AND DENSITY(LB/IN3),FREE FORMAT')

READ(8,*) E,V,R

E = 1610000.
V = .33
R = .16

DPLY = 1. / 7.

YIELD = 110000.

SHEAR = 0.577 * YIELD

XIT(I) = YIELD
X1C(I) = YIELD
X2T(I) = YIELD
X2C(I) = YIELD
S6P(I) = SHEAR
S6M(I) = SHEAR
E11(I) = E
E22(I) = E
E33(I) = E
G12(I) = E / 2. / (1. + V)
G13(I) = G12(I)
G23(I) = G12(I)
V10(I) = V
V13(I) = V
V23(I) = V

RHO(I) = R / 386.4

THETA(I) = 0.

I = I + 3

I = 2
ISN 0034  I3 = 3

C

ISN 0035  E1BA = 31.0E6

ISN 0036  E1BA = 0.9 * E1BA

ISN 0037  E2BA = 20.0E6

ISN 0038  E2BA = 0.9 * E2BA

ISN 0039  V1CBA = 0.27

ISN 0040  G3A = 8.5E6

ISN 0041  GBA = 0.9 * GBA

ISN 0042  RBA = 0.097

ISN 0043  RSA = 0.9 * RSA

ISN 0044  XITBA = 140000.0

ISN 0045  XITBA = 0.9 * XITBA

ISN 0046  X1CBA = 200000.0

ISN 0047  X1CBA = 0.9 * X1CBA

ISN 0048  X2TBA = 14800.0

ISN 0049  X2TBA = 0.9 * X2TBA

ISN 0050  X2CBA = 32200.0

ISN 0051  X2CBA = 0.9 * X2CBA

ISN 0052  SBA = 14500.0

ISN 0053  SBA = 0.9 * SBA

C

ISN 0054  E1GE = 10.5E6

ISN 0055  E2GE = 1.54E6

ISN 0056  V12GE = 0.3

ISN 0057  G6E = 0.85E6

ISN 0058  RGE = 0.056

ISN 0059  XITGE = 160000.0

ISN 0060  X1CGE = 160000.0

ISN 0061  X2TGE = 7500.0

ISN 0062  X2CGE = 25000.0

ISN 0063  SGE = 10000.0

C

ISN 0064  DO 200 II = 1,2

ISN 0065  E1I(I2) = E1BA

ISN 0066  E22(I2) = E2BA

ISN 0067  E33(I2) = E22(I3)

ISN 0068  G12(I2) = GBA

ISN 0069  G13(I2) = GBA

ISN 0070  G23(I2) = GBA

ISN 0071  V12(I2) = V12BA

ISN 0072  V13(I2) = V12BA

ISN 0073  V23(I2) = V12BA

ISN 0074  RHO(I2) = RBA / 386.4

ISN 0075  XIT(I2) = XITBA

ISN 0076  X1C(I2) = X1CBA

ISN 0077  X2T(I2) = X2TBA

ISN 0078  X2C(I2) = X2CBA

ISN 0079  S6P(I2) = SBA

ISN 0080  S6N(I2) = SBA

C

ISN 0081  E1I(I3) = E1GE

ISN 0082  E22(I3) = E2GE

ISN 0083  E33(I3) = E22(I3)

ISN 0084  G12(I3) = GGE

ISN 0085  G13(I3) = GGE

ISN 0086  G23(I3) = GGE

ISN 0087  V12(I3) = V12GE

ISN 0088  V13(I3) = V12GE
**VERSION 1.3.0 (01 MAY 80)**

```
ISN 0009  V23(I3) = V12GE
ISN 0090  RHO(I3) = RGE / 386.4
ISN 0091  X1I(I3) = X1IGE
ISN 0092  X1C(I3) = X1CGE
ISN 0093  X2T(I3) = X2TGE
ISN 0094  X2C(I3) = X2CGE
ISN 0095  S1P(I3) = S1GE
ISN 0096  S6M(I3) = S6GE

C
ISN 0097  I2 = I2 + 4
ISN 0098  200  I3 = I3 + 2

C

RETURN
```

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

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

*STATISTICS* SOURCE STATEMENTS = 99, PROGRAM SIZE = 1402, SUBPROGRAM NAME = INPUT

*STATISTICS* NO DIAGNOSTICS GENERATED

****** END OF COMPIILATION ******

3004K BYTES OF CORE NOT USED
REFERENCES


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 RTM-6/S. L. Venneri 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
<table>
<thead>
<tr>
<th>Company</th>
<th>Address</th>
<th>Attn.</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Motors Corporation</td>
<td>Warren, MI 48090</td>
<td>R. J. Trippet</td>
</tr>
<tr>
<td>AVCO Lycoming Division</td>
<td>550 South Main Street, Stratford,</td>
<td>Mr. Herbert Kaehler</td>
</tr>
<tr>
<td>Beech Aircraft Corporation, Plant 1</td>
<td>Wichita, KA 67201</td>
<td>Mr. M. K. O'Connor</td>
</tr>
<tr>
<td>Bell Aerospace</td>
<td>P. O. Box 1, Buffalo, NY 14240</td>
<td>R. A. Gellatly</td>
</tr>
<tr>
<td>Boeing Aerospace Company</td>
<td>Impact Mechanics Lab, Seattle, WA</td>
<td>Dr. R. J. Bristow</td>
</tr>
<tr>
<td>Boeing Commercial Airplane Company</td>
<td>P. O. Box 3707, Seattle, WA 98124</td>
<td>Dr. Ralph B. McCormick</td>
</tr>
<tr>
<td>Boeing Commercial Airplane Company</td>
<td>P. O. Box 3707, Seattle, WA 98124</td>
<td>Mr. David T. Powell</td>
</tr>
<tr>
<td>Boeing Commercial Airplane Company</td>
<td>P. O. Box 3707, Seattle, WA 98124</td>
<td>Dr. John H. Gerstle</td>
</tr>
<tr>
<td>Boeing Company</td>
<td>Wichita, KA 67201</td>
<td>Mr. C. F. Tiffany</td>
</tr>
<tr>
<td>McDonnell Douglas Aircraft Corporation</td>
<td>P. O. Box 516, Lambert Field, MO</td>
<td>Library</td>
</tr>
<tr>
<td>Douglas Aircraft Company</td>
<td>3855 Lakewood Blvd, Long Beach, CA</td>
<td>Mr. M. A. O'Connor, Jr.</td>
</tr>
</tbody>
</table>
Garrett AiResearch Manufacturing Co.
111 S. 34th Street
P. O. Box 5217
Phoenix, AZ 85010
Attn: L. A. Matsch

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

General Dynamics
P. O. Box 748
Fort Worth, TX 76101
Attn: Library

General Dynamics/Convair Aerospace
P. O. Box 1128
San Diego, CA 92112
Attn: Library

General Electric Company
Interstate 75, Bldg. 500
Cincinnati, OH 45215
Attn: Dr. L. Beitch
Dr. M. Roberts
Dr. V. Gallardo

General Electric Company
Aircraft Engine Group
Lynn, MA 01902
Attn: Mr. Herbert Garten

Grumman Aircraft Engineering Corp.
Bethpage, Long Island, NY 11714
Attn: Library
H. A. Armen

IIT Research Institute
Technology Center
Chicago, IL 60616
Attn: Library

Lockheed California Company
P. O. Box 551
Dept. 73-31, Bldg. 90, PL. A-1
Burbank, CA 91520
Attn: Mr. D. T. Pland
MARC Analysis Research Corporation
260 Sheridan Avenue, Suite 314
Palo Alto, CA 94306
Attn: P. V. Marcel

United Technologies Research Center
East Hartford, CT 06108
Attn: Dr. A. Dennis

Ohio State University
Columbus, OH 43210
Attn: A. W. Leissa

University of California
Mechanics & Structures Department
School of Engineering & Applied Sciences
Los Angeles, CA 90024
Attn: L. A. Schmit, Jr.

Columbia University
New York, NY 10027
Attn: R. Vaicaitis

Georgia Institute of Technology
School of Civil Engineering
Atlanta, GA 30332
Attn: S. N. Atluri

Georgia Institute of Technology
225 North Avenue
Atlanta, GA 30332
Attn: G. J. Simitsis

Lawrence Livermore Laboratory
P. O. Box 808, L-421
Livermore, CA 94550
Attn: M. L. Wilkins

Lehigh University Institute of Fracture
and Solid Mechanics
Bethlehem, PA 18015
Attn: G. T. McAllister

Materials Science Corporation
1777 Walton Road
Blue Bell, PA 19422
Attn: W. B. Rosen

National Bureau of Standards
Engineering Mechanics Section
Washington, DC 20234
Attn: R. Mitchell
Purdue University
School of Aeronautics & Astronautics
West Lafayette, IN 47907
Attn: C. T. Sun

University of Dayton Research Institute
Dayton, OH 45409
Attn: F. K. Bogner

Texas A&M University
Aerospace Engineering Department
College Station, TX 77843
Attn: W. E. Haisler
     J. M. Vance

V. P. I. and State University
Department of Engineering Mechanics
Blacksburg, VA 24061
Attn: R. H. Heller

University of Arizona
College of Engineering
Tucson, AZ 87521
Attn: R. H. Gallagher
     J. C. Heinrich

University of California
Department of Civil Engineering
Berkeley, CA 94720
Attn: E. Wilson

University of Kansas
School of Engineering
Lawrence, KS 66045
Attn: R. H. Dodds

University of Virginia
School of Engineering & Applied Science
Charlottesville VA 22901
Attn: E. J. Gunter
End of Document