General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)
GENERAL PURPOSE COMPUTER PROGRAM FOR INTERACTING SUPERSONIC CONFIGURATIONS

PROGRAMMER'S MANUAL

BY

W. CRILL
B. DALE

Prepared under Contract No. NAS1-13986

BY

BELL AEROSPACE TEXTRON
Post Office Box One
Buffalo, New York 14240

FOR

NASA

National Aeronautics and Space Administration
Summary

This manual contains details for the program ISCON, an abbreviation for "Interacting Supersonic Configuration". The programmer will find an accompanying User's Manual necessary to execute test cases.

The work was accomplished by Bell Aerospace Textron under contract NAS1-13986 with the National Aeronautics and Space Administration, Langley Research Center, Hampton, Virginia.

The program was written in Fortran IV for the CDC 6400/6600 series computers at NASA, LRC.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.1</td>
</tr>
<tr>
<td>2</td>
<td>2.1</td>
</tr>
<tr>
<td>3</td>
<td>3.1</td>
</tr>
<tr>
<td>4</td>
<td>4.1</td>
</tr>
<tr>
<td>5</td>
<td>5.1</td>
</tr>
<tr>
<td>6</td>
<td>6.1</td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>
INTRODUCTION

This manual describes the computer program in support of the problem to generate a numerical procedure for the determination of unsteady aerodynamic forces on arbitrary interacting wings and tails in supersonic flow. See Reference 1.

The information presented here is geared to the programmer. It is sufficient to fully describe the program logic and the required peripheral storage. Figure 1 gives an overview of the entire program, and is the basis for the control program. A separate section is devoted to the definition of external files. Program limitations and convergence criteria are discussed. Individual subroutine write-ups are presented along with the complete Fortran source listing.

All User oriented information is contained in the ISCON User's Manual. For a presentation of input format and test case results this manual should be consulted.
Section 1

Computer Program Flow

Figure 1 illustrates the computer program flow of the ISCON program. The program is divided into two phases - mesh generation phase and solution.

Mesh Generation Phase

This phase reads and processes all input, generates the mesh, prints the grid and determines dynamic storage area constants. The downwash coefficients are computed by LOOPW for all frequencies and computes wake effects if desired.

Solution Phase

In the solution phase, each frequency is selected for solution.

The downwash is generated for each mode by using DIAG and ITRATE. Wake effects are iterated if present. The output is displayed and the program now cycles to the next case.
ENTRY
DEFINE ALL WORK SPACE
DEFINE ALL CONTROL VARIABLES
DEFINE LOGICAL I/O UNITS
READ AND PRINT TITLE

CALL INPI ——— READ AND PROCESS RUN CARD
CALL WING IN ——— READ AND PROCESS GEOMETRIC DATA
CALL GRID IN
OR MESH CL
OR MESH
CALL PT GRID ——— PRINT GRID
CALL SONS ——— DETERMINE SECTION INFLUENCE

RESTART?

YES

TAPE 15 WAKE EFFECTS COEFFICIENTS

NO

TAPE 8 DOWN WASH COEFFICIENTS

CALL LOOP W ——— GENERATE DOWN WASH COEFFICIENT MATRICES FOR ALL FREQUENCIES

INPUT

OUTPUT

A

Figure 1. Computer Program Flow
CALL RD MODE
SELECT FREQUENCY
CALL DIAG
SELECT MODE
GENERATE INPUT DOWN WASH
CALL ITRATE

CONVERGENCE ?
\[ \phi \]
YES \NO

WAKE EFFECT ?
\[ \phi \]
YES \NO

DISPLAY OUTPUT
GO TO NEXT CASE
END OF JOB

READ MODES - SPLINE TO AERO GRID
DECOMPOSE W MATRIX
SOLVE FOR POTENTIAL GRADIENT

Figure 1. Computer Program Flow (contd)
Figure 1. Computer Program Flow (contd)
Section 2
External File Structure

This program uses eleven (11) files during the course of operation. The delivery version of the ISCON program defines the unit designations with a block data statement which assigns values to COMMON TAPE. These may be altered if desired.

<table>
<thead>
<tr>
<th>UNIT NAME</th>
<th>UNIT ID</th>
<th>TYPE</th>
<th>USAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>I9</td>
<td>8</td>
<td>(S,P)</td>
<td>Store downwash coefficients for all frequencies</td>
</tr>
<tr>
<td>I10</td>
<td>10</td>
<td>S</td>
<td>Card images for 1 case, diagonal blocks of coefficients</td>
</tr>
<tr>
<td>I11</td>
<td>11</td>
<td>S</td>
<td>Off diagonal blocks of coefficients</td>
</tr>
<tr>
<td>I12</td>
<td>12</td>
<td>(S,W)</td>
<td>Wake coefficients for 1 frequency</td>
</tr>
<tr>
<td>I13</td>
<td>13</td>
<td>S</td>
<td>Store ETA and dETA/dx</td>
</tr>
<tr>
<td>I14</td>
<td>14</td>
<td>S</td>
<td>Store ETA and dETA/dx</td>
</tr>
<tr>
<td>J9</td>
<td>9</td>
<td>S</td>
<td>Time Print BCD tape</td>
</tr>
<tr>
<td>I15</td>
<td>15</td>
<td>(S,P,W)</td>
<td>Wake Element effects for 1 frequency</td>
</tr>
<tr>
<td>I5</td>
<td>5</td>
<td>P</td>
<td>Standard Card input</td>
</tr>
<tr>
<td>I6</td>
<td>6</td>
<td>P</td>
<td>Standard Line Printer</td>
</tr>
</tbody>
</table>

(Not defined in Fortran) P Standard Plot Tape

Where S = Scratch, W = Wake, P = Permanent

If the restart feature is desired, then units designated as (P) must be permanent files. Units I12 and I15 are only required when WAKE effect is desired. When Plot is requested, the standard plot tape is needed.
Section 3

PROBLEM SIZE LIMITATIONS

At delivery date, this program contains the following limitations (these limitations may change):

1. Number of wings 4
2. Number of sections/wing 3
3. Number of modes 10
4. Number of spans 100
5. Number of corner points/wing 8
6. Number of frequencies 12
7. Number elements/span 70
8. Number of elements: -

The maximum number of elements is based on the amount of dummy storage made available NDUM = the number of storage set when the program is compiled. At delivery, NDUM is set so that the program is capable of handling 300 elements. Since the number of elements is defined by the mesh generator, the mesh is generated by the program first. Then the amount of required storage is determined. If enough storages are not available in the program, the run is terminated, and the program cycles to the next case.
Section 4

CONVERGENCE CRITERIA

Convergence criteria are set up in the program using block data statements. These constants may be changed by altering their value in the block data subprogram.

1. Criteria of Convergence for both wake and potential gradient iteration is used by the subroutine IRELE. This uses a norm type of convergence based on EPS = .01. EPS is set up in common block CEPS.

2. The maximum number of iterations used for potential gradient iterations in subprogram ITRATE is assigned the variable name ITMAX. This variable is set at 10 in the program delivery version. It is included in the common block ITERAT.

3. The parameter PERC is required by LOOPW. It is used to improve computational economy in computing the velocity influence matrix. PERC represents the ratio of $W_{ij}/W_{ii}$, that is, the ratio of far field influencing coefficient to the receiving element self coefficient. This variable is set equal to .005 and is contained in common block PEREPS.
Section 5
Subroutine Write-Ups

The main computer program COMMON blocks are defined in this section. These blocks are referenced by the subroutine write-ups which are presented alphabetically for each subprogram.
# COMMON BLOCKS

<table>
<thead>
<tr>
<th>NAME</th>
<th>VARIABLES</th>
<th>DEFINITIONS</th>
<th>SUBROUTINE REFERENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASIC</td>
<td>MACH</td>
<td>Mach Number SQRT (MACH^2-1)</td>
<td>FGEN, MECH, MESHCL,</td>
</tr>
<tr>
<td></td>
<td>BEETA</td>
<td></td>
<td>WINTGR</td>
</tr>
<tr>
<td>CEPS</td>
<td>EPS</td>
<td>Convergence criteria for ITRATE, IRELE</td>
<td>IRELE, ITRATE</td>
</tr>
<tr>
<td>EEW</td>
<td>XUP, YUP</td>
<td>X and Y coordinates of the upper end of the influencing line</td>
<td>WINTGRR, WVINT</td>
</tr>
<tr>
<td></td>
<td>XLW, YLW</td>
<td>X and Y coordinates of the lower end of the influencing line</td>
<td></td>
</tr>
<tr>
<td>EREØNE</td>
<td></td>
<td>Logical variable set to .TRUE. if a line (in the right hand sense) has influence on a given receiving point</td>
<td></td>
</tr>
<tr>
<td>EMIRRO</td>
<td></td>
<td>Logical variable set equal to .TRUE. if the mirror image of the line has influence on a given receiving point</td>
<td></td>
</tr>
<tr>
<td>RR</td>
<td></td>
<td>Average hyperbolic radius of the endpoints of an influencing line with respect to given receiving point</td>
<td></td>
</tr>
<tr>
<td>RL</td>
<td></td>
<td>Average hyperbolic radius of the mirror image of the endpoints of an influencing line with respect to a given receiving point</td>
<td></td>
</tr>
<tr>
<td>RTR</td>
<td></td>
<td>The product of the hyperbolic radii at the endpoints of an influencing line with respect to a given receiving point</td>
<td></td>
</tr>
<tr>
<td>RTL</td>
<td></td>
<td>The product of the hyperbolic radii at the endpoints of the mirror image of an influencing line with respect to a given receiving point</td>
<td></td>
</tr>
<tr>
<td>NAME</td>
<td>VARIABLES</td>
<td>DEFINITIONS</td>
<td>SUBROUTINE REFERENCE</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>FQ1</td>
<td>NFREQ</td>
<td>Number of frequencies that are input for the current run</td>
<td>MAIN,FGEN,FREQW,LOOPW,WAKET,WINTGR</td>
</tr>
<tr>
<td>CF</td>
<td></td>
<td>An array, a list of 30 frequency coefficients for each frequency</td>
<td></td>
</tr>
<tr>
<td>FREQ</td>
<td></td>
<td>A list of the frequencies input</td>
<td></td>
</tr>
<tr>
<td>FREQP</td>
<td></td>
<td>A list of modified frequencies $FREQP(I)=FREQ(I)*MACH/BEETA$</td>
<td></td>
</tr>
<tr>
<td>FRTE$T$</td>
<td></td>
<td>The largest modified frequency</td>
<td></td>
</tr>
<tr>
<td>NTMXX</td>
<td></td>
<td>The maximum number of non-zero frequency coefficients</td>
<td></td>
</tr>
<tr>
<td>ITERAT</td>
<td>ITMAX</td>
<td>Maximum number iteratives</td>
<td>ITRATE</td>
</tr>
<tr>
<td>ITG</td>
<td>IGUESS</td>
<td>Guess vector indicator</td>
<td>ITRATE</td>
</tr>
<tr>
<td>K1112</td>
<td>KST11</td>
<td>Maximum element numbers for non-zero coefficients</td>
<td>ITRATE,DIAG,WAKET,WRITEO,WTEPHT</td>
</tr>
<tr>
<td>KST12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCOM</td>
<td>CFTABLE</td>
<td>Table of equations coefficients for mode polynomial</td>
<td>RDMODE</td>
</tr>
<tr>
<td>MODEF</td>
<td>EM</td>
<td>Storage for modal values for a single point</td>
<td>MFUN,RDMODE</td>
</tr>
<tr>
<td></td>
<td>DM</td>
<td>Storage for values of the derivative of the modes for a single point</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>$X$ coordinate of the point</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td></td>
<td>$Y$ coordinate of the point</td>
<td></td>
</tr>
<tr>
<td>J1</td>
<td></td>
<td>Wing number that the point is on</td>
<td></td>
</tr>
<tr>
<td>J2</td>
<td></td>
<td>Section of WING J1 that the point is on</td>
<td></td>
</tr>
<tr>
<td>NEXTCS</td>
<td>IFLUSH</td>
<td>Set equal to if an error is found that is serious enough to cause termination of run</td>
<td>MAIN,INPI,WINTGR</td>
</tr>
<tr>
<td>NAME</td>
<td>VARIABLES</td>
<td>DEFINITIONS</td>
<td>SUBROUTINE REFERENCE</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>PARAM</td>
<td>NWING</td>
<td>Total number elements in structure</td>
<td>FREQW</td>
</tr>
<tr>
<td>RRLL</td>
<td>TVW</td>
<td>Transformation matrix to change the velocity influence coefficients from the receiving point to the influencing point</td>
<td>LOOPW, RTOI, WINTGR</td>
</tr>
<tr>
<td></td>
<td>YRO, YLO</td>
<td>Y coordinate of the receiving element in the system of the influencing element, YLO is in the system of the mirror image of the influencing element</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZR</td>
<td>Z coordinate of the receiving element in the system of the influencing element</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZRZR</td>
<td>ZR*ZR</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZL</td>
<td>Z coordinate of the receiving element in the system of the mirror image of the influencing element</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZLZL</td>
<td>ZL*ZL</td>
<td></td>
</tr>
<tr>
<td>TAPE</td>
<td>I9, I10, I11, I12, I13, I14, J9, I15</td>
<td>File numbers set up by block data subprogram</td>
<td>MAIN, DIAG, ITRATE, LOOPW, TIMOUT, WRITE0</td>
</tr>
<tr>
<td>TIMER</td>
<td>ITIME</td>
<td>Time point array</td>
<td>MAIN</td>
</tr>
<tr>
<td>VELCOM</td>
<td>NMAX</td>
<td>Maximum dimension of D</td>
<td>MAIN, DIAG, ITRATE</td>
</tr>
<tr>
<td></td>
<td>PRINT</td>
<td>Print control</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NWBLOCK</td>
<td>Number blocks for ITRATE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NWROW</td>
<td>Number rows in block</td>
<td></td>
</tr>
<tr>
<td>WAKEUP</td>
<td>WTE5</td>
<td>Array of wake effects for elements</td>
<td>MAIN, DIAG, LOOPW, WAKET, WINTGR</td>
</tr>
<tr>
<td></td>
<td>TREDGE</td>
<td>Trailing edge indicator</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WAKE1</td>
<td>Wake indicator</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WAKE</td>
<td>Wake indicator</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WAKENZ</td>
<td>Wake indicator for non-zero term</td>
<td></td>
</tr>
</tbody>
</table>

-5.4-
<table>
<thead>
<tr>
<th>NAME</th>
<th>VARIABLES</th>
<th>DEFINITIONS</th>
<th>SUBROUTINE REFERENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>WV1</td>
<td>NTERM</td>
<td>Number of terms taken in the finite series approximation of the velocity influence expression</td>
<td>WINTGR, WVINT</td>
</tr>
<tr>
<td>V(30)</td>
<td></td>
<td>Array of sidewash contributions for up to thirty terms</td>
<td></td>
</tr>
<tr>
<td>W(30)</td>
<td></td>
<td>Array of upwash contributions for up to thirty terms</td>
<td></td>
</tr>
<tr>
<td>WW1</td>
<td>XO,YO,ZD</td>
<td>The X, Y and Z coordinates of a receiving point</td>
<td>LØØPW, WVINT, WINTGR</td>
</tr>
<tr>
<td>ZDZD</td>
<td></td>
<td>ZD*ZD</td>
<td></td>
</tr>
<tr>
<td>SYMK</td>
<td></td>
<td>Symmetry code of the wing that the influencing element is in</td>
<td></td>
</tr>
<tr>
<td>NINSID</td>
<td></td>
<td>A counter to tell how many elements in the current span, from the bottom up, that have been found to have influence on the current receiving point, so far</td>
<td></td>
</tr>
<tr>
<td>XYSCAL</td>
<td>XX</td>
<td>X coordinate of a point</td>
<td>ZFDZ</td>
</tr>
<tr>
<td></td>
<td>YY</td>
<td>Y coordinate of a point</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XO</td>
<td>Offset value in X direction currently taken to be 0.0.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>YO</td>
<td>Offset value in Y direction, currently taken to be 0.0.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BREF</td>
<td>Scalar transformation value, currently set to 1.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZZZZ</td>
<td>Value of first mode input</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ITEST</td>
<td>Set equal to 1 if mode is a constant for the entire grid</td>
<td></td>
</tr>
<tr>
<td>ZFDZIO</td>
<td>I5</td>
<td>Input unit numbers</td>
<td>MAIN,READAB, READXY</td>
</tr>
<tr>
<td></td>
<td>I6</td>
<td>Output unit numbers</td>
<td></td>
</tr>
<tr>
<td>NAME</td>
<td>VARIABLES</td>
<td>DEFINITIONS</td>
<td>SUBROUTINE REFERENCE</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
<td>------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>ZFUNNY</td>
<td>N</td>
<td>Number points on structural grid for spline</td>
<td>ROMODE, READAB, ZFDZ</td>
</tr>
<tr>
<td></td>
<td>IERF</td>
<td>Error indicator number</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B1,B2,B3</td>
<td>Spline constants</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NFUNMX</td>
<td>Maximum N value</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Function</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>----------------------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>MAIN</td>
<td>Main calling program</td>
<td>5.10</td>
<td></td>
</tr>
<tr>
<td>BLOCK</td>
<td>Define COMMON constants</td>
<td>5.12</td>
<td></td>
</tr>
<tr>
<td>CABSO</td>
<td>Compute absolute value of a complex number</td>
<td>5.14</td>
<td></td>
</tr>
<tr>
<td>DECOM</td>
<td>Decompose matrix for solution</td>
<td>5.15</td>
<td></td>
</tr>
<tr>
<td>DIAG</td>
<td>Determine matrix blocks</td>
<td>5.17</td>
<td></td>
</tr>
<tr>
<td>ECHO</td>
<td>Provide input deck echo print</td>
<td>5.19</td>
<td></td>
</tr>
<tr>
<td>EFPLLOT</td>
<td>Calls LRC CALPLT plot routine to end plot tape</td>
<td>5.21</td>
<td></td>
</tr>
<tr>
<td>EONE</td>
<td>Generate element influence</td>
<td>5.22</td>
<td></td>
</tr>
<tr>
<td>FGEN</td>
<td>Generate frequency coefficients</td>
<td>5.23</td>
<td></td>
</tr>
<tr>
<td>FREQW</td>
<td>Determines coefficients for 1 freq.</td>
<td>5.25</td>
<td></td>
</tr>
<tr>
<td>GETTIM</td>
<td>Transforms CPU time to integer</td>
<td>5.26</td>
<td></td>
</tr>
<tr>
<td>GRIDIN</td>
<td>Reads grid input data</td>
<td>5.28</td>
<td></td>
</tr>
<tr>
<td>INP1</td>
<td>Reads and analyzes RUN card</td>
<td>5.30</td>
<td></td>
</tr>
<tr>
<td>IRELE</td>
<td>Test for convergence</td>
<td>5.34</td>
<td></td>
</tr>
<tr>
<td>ITRATE</td>
<td>Iteration equation solver</td>
<td>5.35</td>
<td></td>
</tr>
<tr>
<td>LOOPW</td>
<td>Evaluate W, Wake coefficients</td>
<td>5.36</td>
<td></td>
</tr>
<tr>
<td>MESH</td>
<td>Grid for section without control lines</td>
<td>5.40</td>
<td></td>
</tr>
<tr>
<td>MESHCL</td>
<td>Grid for section with control lines</td>
<td>5.42</td>
<td></td>
</tr>
<tr>
<td>MFUN</td>
<td>Generate modes by polynomial</td>
<td>5.44</td>
<td></td>
</tr>
<tr>
<td>MXERR</td>
<td>Prints error messages</td>
<td>5.45</td>
<td></td>
</tr>
<tr>
<td>OUTP1</td>
<td>Prints velocity potentials, pressures</td>
<td>5.46</td>
<td></td>
</tr>
<tr>
<td>OUTP2</td>
<td>Prints total lift and total pressures</td>
<td>5.47</td>
<td></td>
</tr>
<tr>
<td>OUTP3</td>
<td>Prints aerodynamic coefficients</td>
<td>5.48</td>
<td></td>
</tr>
<tr>
<td>PHIL</td>
<td>Generate velocity potential</td>
<td>5.49</td>
<td></td>
</tr>
<tr>
<td>PLOT</td>
<td>Calls LRC CALPLT and NFRAME plot routines</td>
<td>5.51</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Function</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>PLOTGD</td>
<td>Plots aerodynamic grid</td>
<td>5.52</td>
<td></td>
</tr>
<tr>
<td>PTGRID</td>
<td>Prints aerodynamic grid</td>
<td>5.54</td>
<td></td>
</tr>
<tr>
<td>RDETA</td>
<td>Read ETA from files 14, 15</td>
<td>5.56</td>
<td></td>
</tr>
<tr>
<td>RDMODE</td>
<td>Read and generate mode data</td>
<td>5.57</td>
<td></td>
</tr>
<tr>
<td>READAB</td>
<td>Read spline coefficients</td>
<td>5.60</td>
<td></td>
</tr>
<tr>
<td>READXY</td>
<td>Read Structural Grid</td>
<td>5.62</td>
<td></td>
</tr>
<tr>
<td>RSTART</td>
<td>Reads and checks restart files</td>
<td>5.64</td>
<td></td>
</tr>
<tr>
<td>RTOI</td>
<td>Generates transformation matrices</td>
<td>5.66</td>
<td></td>
</tr>
<tr>
<td>SOLVE</td>
<td>Backsubstitution for solution</td>
<td>5.58</td>
<td></td>
</tr>
<tr>
<td>SONS</td>
<td>Finds influence of sections</td>
<td>5.70</td>
<td></td>
</tr>
<tr>
<td>SONSPT</td>
<td>Prints influence of sections</td>
<td>5.72</td>
<td></td>
</tr>
<tr>
<td>SYMBOL</td>
<td>Calls LRC plot routine NOTATE</td>
<td>5.73</td>
<td></td>
</tr>
<tr>
<td>TIMOUT</td>
<td>Time print on file J9</td>
<td>5.74</td>
<td></td>
</tr>
<tr>
<td>TMAX</td>
<td>Computes the maximum elements in a real array</td>
<td>5.76</td>
<td></td>
</tr>
<tr>
<td>TRANS</td>
<td>Transformation of coordinates</td>
<td>5.77</td>
<td></td>
</tr>
<tr>
<td>WAKET</td>
<td>Write wake effects on file Tl2</td>
<td>5.78</td>
<td></td>
</tr>
<tr>
<td>WINGCK</td>
<td>Check course points for compatibility</td>
<td>5.80</td>
<td></td>
</tr>
<tr>
<td>WINGIN</td>
<td>Read wing input data</td>
<td>5.82</td>
<td></td>
</tr>
<tr>
<td>WINTGR</td>
<td>Computes W coefficients for 2 elements</td>
<td>5.87</td>
<td></td>
</tr>
<tr>
<td>WRETA</td>
<td>Write ETA in file Il3 and Il4</td>
<td>5.89</td>
<td></td>
</tr>
<tr>
<td>WRITEO</td>
<td>Write W coefficients on file Il1</td>
<td>5.90</td>
<td></td>
</tr>
<tr>
<td>WRTIZ</td>
<td>Write W coefficients in file Il0</td>
<td>5.91</td>
<td></td>
</tr>
<tr>
<td>WRTIZW</td>
<td>Write Wake effects on file Il2</td>
<td>5.92</td>
<td></td>
</tr>
<tr>
<td>WTEPNT</td>
<td>Generate wake effects</td>
<td>5.93</td>
<td></td>
</tr>
</tbody>
</table>
Subroutine Write-Ups (contd)

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>WVINT</td>
<td>Evaluates equations for influence coefficients</td>
<td>5.95</td>
</tr>
<tr>
<td>ZFDZ</td>
<td>Evaluate model functions and derivatives</td>
<td>5.96</td>
</tr>
</tbody>
</table>
1. Subroutine Name: MAIN

2. Purpose:
The main organization of the ISCON program is defined here.
See Section 1 for a description of this subprogram and program flow.

3. Subroutine Required:
- DIAG
- SONS
- RDETA
- MESHCL
- ECHO
- LOOPW
- WRETA
- PLOTGD
- FGEN
- MXERR
- EFPLOT
- PTGRID
- INPL
- OUTP1
- GETTIM
- RDMUDE
- MESH
- OUTP2
- GRIDIN
- RSTART
- PHIL
- OUTP3
- ITRATE
- SONSPT
- TIMEOUT
- WINGIN
- WTEPHT

See Section 6 for the function of each one of these subroutines.

4. Files Used:
I5, I6, I9, I10, I11, I12, I13, I14, J9, I15
These files are defined in Section 2.

5. Common Blocks Used:
- BASIC: Contains Mach number variables
- FQ1: Contains frequency variables
- MXSTOR: Contains maximum parameters
- NEXTCU: Contains error control
- TIMEP: Contains time variable
- Param: Contains no. elements
- TAPE: Contains tape definitions
- WAKEUP: Contains WAKE parameters
- VELCOM: Contains iteration parameters
WAKE Contains wake controls
ZFDZIO Contains ECHO tape def.
ZFUNNY Contains mode function parameters

6. Calling Sequence: None
1. Subroutine Name: BLOCK DATA

2. Purpose:
   Set up constants. Common blocks used in tape definitions, convergence criteria, print controls, iteration maximums, and storage limits.

3. Common Blocks:
   - ITERAT: Maximum no. iterations
   - MXSTOR: Maximum storage limits
   - TAPE: File definitions
   - VELCOM: Maximum block size, print control

4. Definitions:
   1. ITERAT: ITMAX=10=maximum no. iterations for solution
   2. MXSTOR: NWINGS = no. wings = 4
      NSECTN = no. sections = 3
      NMODES = no. nodes = 10
      NSPANS = no. spans = 100
      NCORNR = no. corner pts. = 8
      NFRQUN = no. frequencies = 12
      NEPSPN = no. elements/span = 70
      NMXFDZ = no. mode storage = 300
   3. TAPE: I9 = 8
      I10 = 10
      I11 = 11
      I12 = 12
      I13 = 13
\[ I_{14} = 14 \]
\[ J_9 = 9 \]
\[ I_{15} = 15 \]

See Section 2 for file descriptions.

4. VELCOM
   NMAX = maximum solution block size = 10
   PRINT = print control = 2

5. Calling Sequence: None
1. **Function Name:** CABS0
2. **Purpose:**
   Compute absolute value of complex number.
3. **Input Arguments:**
   \[ A = \text{complex number} \]
4. **Output Argument:**
   \[ \text{CABS0} = \text{value of function} \]
5. **Subroutine User:** IRELE
6. **Subroutine Used:** SQRT
7. **Calling Sequence:**
   \[ \text{CABS0} = \text{CABS0}(A) \]
1. **Subroutine Name:** DECOM

2. **Purpose:**
   Factorization of the matrix A into a product of a lower triangular matrix L and an upper triangular matrix U. L has a unit diagonal which is not stored.

3. **Equations and Procedure:**
   Matrix triangularization by Gaussian elimination. See Algorithm 423, "Collected Algorithms from CACM", by Cleve Moler.

4. **Input Arguments:**
   - **N** Order of the Matrix A
   - **NDIM** First dimension of A declared in calling program. If A is singly subscripted in calling program, set NDIM=N.
   - **A** On input the matrix to be factored.

5. **Output Arguments:**
   - **A** On output A(I,J) I.LE.J contains the upper triangle U, A(I,J) I.GT.J contains (I-L), where I is the identity matrix and L is the lower triangle.
   - **IP** IP(K) K.LT.N contains the row interchange information. IP(N) contains (-1)**(number of interchanges) or 0.

6. **Common Blocks Used:** None

7. **Error Returns:** None
8. **Calling Sequence:**
   Call DECOM(N,NDIM,A,IP)

9. **Input Tapes:** None
10. **Output Tapes:** None
11. **Scratch Tapes:** None
12. **Storage Required:** 347 words
13. **Subroutines Used:** None
14. **Subroutine User:** MAIN
15. **Remarks:**
   1. If IP(N)=0 then matrix A is singular.
   2. Use DECOM in conjunction with subroutine "solve" to obtain the solution of the linear system A*X = B.
   3. Determinant(A) = IP(N)*A(1,1)*A(2,2)*...*A(N,N).
   4. The row interchange information stored in IP is not easy to interpret. However, it is used properly by "solve".
1. **Subroutine Name:** DIAG

2. **Purpose:**

   Process coefficient matrix for 1 frequency.

   The incore blocks are determined based on the W coefficient matrices stored on file I9. After decomposing, the diagonal blocks are stored on file I10. The off-diagonal terms are stored in compressed form on file I11. If wake effects have been computed, the wake coefficients are stored in compressed form on file I12. WAKENZ is set .FALSE. if no wake effects or else .TRUE. on return.

3. **Input Arguments**

   - **IF** frequency number
   - **LRE** logical record length on tape I9
   - **NNCH** row number of trailing edge elements
   - **MC** maximum dimension of incore matrix
   - **NET** number of elements in total system
   - **FRQ** frequency value
   - **NSPT** No. trailing edges

4. **Output Arguments**

   - **WROW** W coefficients work storage from file I9
   - **D** decomposed block (incore) stored on file I10
   - **AROW** off diagonal block stored on file I11
   - **IZ** non zero element numbers array
   - **IP** decomposing information array
   - **CWROW** wake coefficients work storage from file I15
   - **CAROW** wake effects for 1 frequency stored on file I12
5. Subroutines Used:
   WRITEO
   WAKET
   DECOM
   FREQW

6. Subroutine User:
   Main program

7. Calling Sequence:
   CALL DIAL (WROW, IF, LRECL, NNCH, MC, D, NET, AROW, IZ, IP,
   NSPT, FRQ, CAROW, CWROW)

8. Files Used:
   I9 = (input) file containing W coefficients, for all
       frequencies
   I10 = diagonal block, decomposed for 1 frequency (output)
   I11 = off diagonal terms (output)
   I15 = (input) wake coefficients
   I13 = wake effects for 1 frequency in compressed form
1. **Subroutine Name:** ECHO

2. **Purpose:**
   To read the input for a given case, and print out an echo check of the input data.

3. **Equations and Procedures:**
   The input deck is read from unit I5, and placed on unit I11 and printed with format control on unit I6. If an end of file is encountered, the subroutine sets the variable KONTRL equal to 1.

4. **Input Arguments:** None

5. **Output Arguments:**
   - KONTRL Integer variable set equal to 1 if an end of file is encountered on reading the input stream.

6. **Common Blocks:**
   /TAPE/
   I11 Unit number of file to have card images of input stream.

   /ZFDZIO/ 
   J5 Unit number of input stream - card reader 
   J6 Unit number for printed output

7. **Error Returns:** None

8. **Calling Sequence:**
   Call ECHO(KONTRL)

9. **Input Tapes:** None
10. **Output Tapes:**
   Ill Contains card images of input deck.
11. **Scratch Tapes:** None
12. **Storage Required:** 225 words
13. **Subroutine Required:** None
14. **Subroutine User:** MAIN
15. **Remarks:**
   The test for the end of file differs from IBM to CDC. Make sure that the correct coding is present for the correct installation.
1. **Subroutine Name:** EFPLOT

2. **Purpose:**
   To end the plot tape on the CDC version.

3. **Equations and Procedure:**
   CDC plot routine is called to end the tape, CALL CALPLT (0.0, 0.0, 999). This routine replaces the IBM version of the CALCOMP routine EFPLOT.

4. **Input Arguments:**
   A dummy argument, not used, but present to keep the call to the routine identical to the IBM-CALCOMP version of EFPLOT. (In that version this argument represents four characters to be printed out on the plot just before ending the plot tape.)

5. **Output Arguments:** None

6. **Common Blocks Used:** None

7. **Error Returns:** None

8. **Calling Sequence:** Call EFPLOT(A)

9. **Input Tapes:** None

10. **Output Tapes:** None

11. **Scratch Tapes:** None

12. **Storage Required:** relatively very small

13. **Subroutine Required:** CALPLT

14. **Subroutine User:** MAIN

15. **Remarks:**
   This subroutine is not used in the IBM version. It is replaced by the standard CALCOMP routine of the same name.
1. **Subroutine Name**: EONE

2. **Purpose**:
   This routine determines if an element and its mirror image are in the Mach cone.

3. **Equations and Procedure**:
   This routine examines the end points of a line. If either one end point or the other is in the Mach cone, the line is in the mach cone and EMIRRO is set equal to .TRUE. The mirror image of the line is found and the same test is made. If the mirror image of the line is inside the MACH cone then EMIRRO is set equal to .TRUE. This routine is used to test only one line per call.

4. **Input Arguments**: None

5. **Output Arguments**: None

6. **Common Blocks Used**:
   /EEW/
   /RRLL/
   /WWL/

7. **Error Returns**: None

8. **Calling Sequence**: CALL EONE

9. **Input Tapes**: None

10. **Output Tapes**: None

11. **Scratch Tapes**: None

12. **Storage Required**: 275 words

13. **Subroutines Required**: None

14. **Subroutine User**: WINTGR

15. **Remarks**: None
1. **Subroutine Name**: FGEN

2. **Purpose**: To calculate KAPPA and the frequency coefficients for each frequency.

3. **Equations and Procedure**: For each frequency, K, the modified frequency KAPPA is defined by:

   \[ KAPPA = \frac{K \times M}{\sqrt{M \times M - 1}} \]

   where \( M \) is the MACH NUMBER.

   (The list of frequencies is stored in \text{FREQ}, and the modified frequencies are stored in \text{FREQP}).

   The frequency coefficients are calculated from:

   \[ C_1 = 1.0 \]

   \[ C_J = -C_{J-1} \left( \frac{KAPPA^{*2}}{(2*N)(2*N-1)} \right) \]

   for \( J > 1 \), where \( N = J - 1 \).

   Currently a maximum of 30 terms are taken (\( J = 30 \)) for each frequency. A check on the exponent is made to avoid an exponent underflow. A scalar, XUNDER, is defined to be -77.5 on the IBM machine and -292.0 on the CDC machine. This is close to the largest negative exponent allowed each machine.

4. **Input Arguments**: None

5. **Output Arguments**: None

6. **Common Blocks Used**: \text{/BASIC/} and \text{/FQ1/}

-5.23-
7. **Error Returns:** None
8. **Calling Sequence:** Call FGEN
9. **Input Tapes:** None
10. **Output Tapes:** None
11. **Scratch Tapes:** None
12. **Storage Required:** 263 words
13. **Subroutine User:** MAIN
14. **Subroutine Required:** None
15. **Remarks:** None
1. **Subroutine Name:** FREQW

2. **Purpose:**
Read one record W coefficient for all frequencies from
file 18. The coefficients for the input frequency are
then determined and stored in AROW.

3. **Input Arguments:**
   I8 = file number containing coefficients
   NF = frequency number
   LRECL = length of file record on file 18

4. **Output Arguments:**
   WROW = work storage for coefficients all frequencies
   AROW = output coefficients record - one frequency

5. **Files Used:**
   I8 = input file containing W coefficient

6. **Subroutine User:** DIAG

7. **Calling Sequence:**
   Call FREQW (I8, AROW, WROW, NF, LRECL)
1. **Subroutine Name:** GETTIM

2. **Purpose:**
   This is a FORTRAN routine to be used on the CDC machine only. It replaces an identically named system routine on the IBM 360-65 at Bell Aerospace. This routine calculates the elapsed CPU time in milliseconds and stores it as an integer.

3. **Equations and Procedure:**
   This routine depends on the CDC routine $ECOND$, which returns the elapsed CPU time in seconds as a floating point number. Subroutine GETTIM multiplies this number by 1000 and stores it in second location of an integer array, eight words in length. This will mimic the IBM system routine GETTIM.

4. **Input Arguments:**
   ITIME Integer array of length 8.

5. **Output Argument:**
   ITIME Integer array of length 8, time is put in second location.

6. **Common Blocks Used:** None

7. **Error Returns:** None

8. **Calling Sequence:**
   CALL GETTIM (ITIME)

9. **Input Tapes:** None

10. **Output Tapes:** None

11. **Scratch Tapes:** None
12. Storage Required: Small

13. Subroutines Required:
   SECOND, a CDC routine that gives elapsed CPU seconds.

14. Subroutine User: MAIN

15. Remarks: None
1. **Subroutine Name:** GRIDIN

2. **Purpose:**
   To read in specific grid information for one section and generate mesh for that section.

3. **Procedure:**
   a) GRID card is read and checked.
   b) SPAN card is read and checked.
   c) CHORD cards are read for every span.
   Cards are checked to make sure elements are not crossed.
   d) The next SPAN card is read, along with its CHORD cards for every span.
   e) Next input card is read.

4. **Input Arguments:**
   - I5: Input unit (usually card reader 5)
   - I6: Output unit (printer 6)
   - ERROR: Logical that comes in false, may be set true
   - BEETA: Sort (Mach*Mach-1) used to transform CORNX and CL
   - NWING: No. wings in structure
   - NSECT: Array no. sections in each wing
   - LAB, ID, FD information on last card read
     (Both input and output)

5. **Output Arguments:**
   - CORNX, CORNY, CORNZ: X, Y, and Z coordinates of corner points 1, 2, 3, 4, in global system
   - NSP: Desired number of spans for section (Input by user)
   - XYZ: 4 X and 2 Y local coord. for each element of section
Z coordinate for all elements in section (from trans)
TRS 2 by 2 trans matrix to go from local to global
NNCH Element number (W.R.T. total structure of last element in span. (Used for labeling plot)
SW Span width
XLE Average X coordinate of leading edge per span
NE Number of elements in section
ICL ICL(I,J) Code for section I of wing J
ICL(I,J) = -1 specified grid information is input here ICL is updated if another 'GRID' section is encountered after read present 'GRID' section
NETSV Counter of total number of elements in structure

6. Common Blocks Used: None
7. Error Returns:
   Input cards are checked. If an error is found, ERROR is set equal to .TRUE. and an error message is written.
8. Calling Sequence:
   SUBROUTINE GRIDIN(CORNX, CORNY, CORNZ, XYZ, Z, TRS, NNCH, SW, XLE,
   1 NSP, NE, NETSV, REFLEM, MFREQ, FREQ,
   1 NWING, NSECT, ICL, ERROR, BEETA, LBA, ID, FD, I5, I6)
9. Input Tapes: None
10. Output Tapes: None
11. Scratch Tapes: None
12. Storage Required: 1244
13. Subroutine User: MAIN
14. Subroutine Required: TRANS
15. Remarks: None
1. **Subroutine Name:** INP

2. **Purpose:**

   To read and check the RUN data card.

3. **Equations and Procedure:**

   The RUN data card is read with a format of \((A^4,2X,I2,4E12.0)\).

   a) A check is made to be sure that it is a RUN card.

   b) The first integer field contains the run type code and is stored in scalar ITRUN.

   c) The second integer field contains the plot request code. If there is a '1' in this field (column 10), a plot of the structure is to be generated, and the logical variable PLOTR is set equal to .TRUE..

   d) The first floating point field contains the mach number and is stored in the real variable MACH. A check is made to see that it is greater than 1.0.

   e) The third integer field contains the code to consider wake effects and is stored in the variable IWTE.

   f) The first floating point field contains the mach number and is stored in the real variable MACH. A check is made to see that it is greater than 1.0.

   g) The second floating point field contains the Reference length and is stored in REFLNK. A check is made to be sure that it was not input as 0.0.

   h) The third floating point field contains the pitching axis, and is stored in the variable XPIN.
1) The fourth floating point field contains the element aspect ratio. This is stored in the variable EARO.

4. **Input Arguments:**

   I5  Logical unit number containing the input deck - usually the card reader.

   I6  Logical unit number for output unit - usually the system printer.

   TITLE Array containing title as character information.

   LAB Input and output of character information in first 4 columns of current input card.

   FD  Floating point array 4 words long to store floating point fields of current input card.

   ID  Integer array 9 words long to store the integer fields of the current input card.

5. **Output Arguments:**

   ITRUN Run type requested.

   ITRUN=1, the run is a complete execution attempt.

   ITRUN<>1, the run is a "check run".

   MACH Mach number (a floating point number).

   REFLEN Reference length.

   XP  Pitching moment axis after being normalized by BEETA and REFLEN.

   XPIN Pitching moment axis as input.

   PLOTR Logical variable set to .TRUE. if plotting is requested. Otherwise it is .FALSE.
BEETA = BEETA = $\phi$SORT(MACH*MACH-1.0).

ERROR = SET TO .TRUE. if error condition is encountered.

EARO = Element aspect ratio

IWTE = Code set to 0 if wake effect is not considered greater than 0 if wake effects are considered.

6. Common Blocks Used:

/NEXTCS/

IFLUSH = Set equal to 1 if run card is not found. This will cause the main routine to terminate the current case.

7. ERROR RETURN:

See description of IFLUSH above.

Logical ERROR is set equal to .TRUE. if any of several errors exist in reading the input. The error condition generates an appropriate error message. All error message format statements are numbered between 2000 and 2026. The following is a list of error messages in this routine. (All messages are preceded by the statement ERROR IN READING INPUT).

a) RUN CARD IS MISSING OR OUT OF ORDER. RUN CARD MUST IMMEDIATELY FOLLOW THE TITLE CARD AND BE THE SECOND CARD IN THE INPUT DECK. JOB IS TERMINATED.

b) MACH NUMBER MUST BE GREATER THAN 1.0.

MACH NUMBER WAS READ AS __________.

MACH NUMBER WILL BE SET EQUAL TO 2.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA.

-5.32-
c) REFERENCE LENGTH WAS READ AS 0.0. THIS IS NOT ALLOWED.
REFERENCE LENGTH WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA.
REFERENCE LENGTH MUST BE INPUT AS A NON-ZERO FLOATING POINT NUMBER IN COLUMNS 37 THROUGH 48 OF THE RUN DATA CARD.

d) OFF-DIAGONAL PERCENTAGE, IF ENTERED, MUST BE GREATER THAN OR EQUAL TO 0.0 AND LESS THAN 1.0.
OFF-DIAGONAL WAS INPUT AS ________.

8. Calling Sequence:
CALL INP1(15,16,TITLE,ITRUN,MACH,REFLEN,XP,XPIN,PL0TR,IWTE,
EARO,BEETA,LAB,ID,FD,ERROR).

9. Input Tapes: None
10. Output Tapes: None
11. Scratch Tapes: None
12. Storage Required: 688
13. Subroutine User: MAIN
14. Subroutine Required: None
15. Remarks: None
1. Subroutine Name: IRELE
2. Purpose:
   Check relative error between A and B.
   If Relative error is less than Eps, IRELE is set = 0 otherwise IRELE is set =1. If KRELE = 1, do not test for convergence.
3. Input Arguments:
   A = argument #1
   B = argument #2
   KRELE = code to signify former convergence
4. Calling Sequence:
   Function IRELE (A,B,KRELE)
5. Output Arguments:
   IRELE = 1 means A, B has not converged.
   IRELE = 0 means A, B has converged
6. Subroutine User: ITRATE,WAKET
1. **Subroutine Name:** ITRATE

2. **Purpose:**
   
   Downwash equation solver using block iteration with successive over relaxation. See Reference 3.

3. **Input Arguments:**
   
   I9 = file containing coefficients off diagonal
   NWR = input downwash - real
   NWI = input downwash - imaginary
   MC = block dimension

4. **Output Arguments:**
   
   GW = output solution vector

5. **Work Storage:**
   
   D = diagonal block storage

6. **Common Block:**
   
   CEPS defines convergence
   ITERATE control variables for ITRATE
   ITG defines ELEMENT DATA on file
   TAPE file unit numbers
   VELCOM define block information

7. **Calling Sequence:**
   
   CALL ITRATE (I9, NWR, NWI, GW, GT, MC, D, IP, DNWR, DNWI, RWR, RWI, NET, A, IZ, GWESS, WW)

8. **Subroutines:**

   SOLVE

9. **Used By:**

   MAIN program
1. **Subroutine Name:** LOOPW

2. **Purpose:**
   To organize the calculation of the velocity influence matrix.

3. **Equations and Procedure:**
   Procedure is outlined in the accompanying flow chart.

4. **Input Arguments:**
   Input arguments are defined in the comment statements at the beginning of the routine.

5. **Output Arguments:**
   Output arguments are also listed in comment cards in subroutine.

6. **Common Blocks Used:**
   `/FQ1/
   `/RRLL/
   `/TAPE/
   `/WAKEUP/
   `/WL/

7. **Error Returns:** None

8. **Calling Sequence:**
   Call LOOPW(NET,NWING,XYZ,NSP,SYM,TRS,NNCH,NE,NSECT,ZSECT, ISONS,I6,TAPE8,XCEN,AREA,WROW,LRECL,WTEROW,LRWTE,PERC, ERROR,IWTE,REFLEN)

9. **Input Tapes:** None

10. **Output Tapes:**
    Two tapes are generated in the subroutine. The first is on logical unit TAPE8. The first record on this tape contains
the "restart record". This record contains the mach number, number of elements, reference length, wake element code, number of frequencies, and a list of the frequencies. Following is a record for every element on the tape. These records contain the element number and the array of influence coefficients for every frequency at every influencing element (all other elements).

If wake effect is requested, a second tape is written. This tape is defined on logical unit 115. It also begins with a restart record as defined above. It has a record only for those elements that have some nonzero wake coefficients. The record begins with the receiving element number and has a list of wake effects (complex numbers) for every frequency for every trailing edge number.

11. Scratch Tapes: None

12. Storage Required: 1110 words

13. Subroutines Required: RTOI and WINTRGR

14. Subroutine User: MAIN

15. Remarks: Domain of Effective Far Field Elements - Contribution to the velocity component at a receiving point, from far field elements (i.e. when is large) is small. Computational economy can be regained by truncating the domain of influence, when the order of the magnitude of the influence coefficient $W_{ij}$ is less than say (E) of $W_{ii}$. The parameter E is fixed in the program at .005 and called PERC. The programmer can update this quality by changing the variable PERC in the BLOCK DATA routine.
BEGIN
LOOP ON RECEIVING ELEMENTS

FIND INFLUENCE COEFFICIENTS FOR RECEIVING ELEMENT ON ITSELF

LOOP ON INFLUENCING WING AND SECTIONS

DOES SECTION INFLUENCE THE SECTION OF THE RECEIVING ELEMENT

YES
FIND TRANSFORMATION MATRIX FOR THIS SECTION

LOOP ON SPANS OF THIS SECTION

LOOP ON ELEMENTS OF SPAN FROM BOTTOM TO TOP

FIND INFLUENCE COEFFICIENTS FOR THIS PAIR OF ELEMENT (ALSO POSSIBLE WAKE EFFECT)

FLOW CHART OF SUBROUTINE LOOPW
A

TEST MAGNITUDE OF INFLUENCE AGAINST DIAGONAL SELF INFLUENCE

SIGNIFICANT

PUT VALUE IN THE REMAINING ELEMENTS OF SPAN UPDATE COUNTER

YES

TEST IF ASYMPTOTIC VALUE FOR SPAN HAS BEEN REACHED

NO

PUT INFLUENCE COEFFICIENT AND POSSIBLE WAKE IN CORRECT LOCATION

END OF LOOP ON ELEMENTS OF SPAN

END LOOP ON INFLUENCING SPANS

END LOOP ON INFLUENCING SECTIONS AND WING

WRITE RECORD FOR 1 RECEIVING ELEMENT, MAYBE WAKE RECORD ALSO

END LOOP ON RECEIVING ELEMENTS

TEST IF WAKE WAS FOUND

UPDATE ELEMENT COUNTERS, PLACING 0.0 IN REMAINING ELEMENT IN THIS SPAN

Flow Chart of Subroutine LOOPQ (Cont.)
1. **Subroutine Name**: MESH

2. **Purpose**: To generate a mesh (grid) for a section without control lines.

3. **Procedure**: Corner points of are input. These corner points are transformed to the aerodynamic reference system in subroutine TRANS. The number of spans (strip of the wing running parallel to the X axis) is input. This determines the span width. The height to width ratio of the elements is given by CRATIO.

4. **Input Arguments**:
   - CORNX, CORNY, CORNZ: X, Y, and Z coord. of corner pts. 1, 2, 3, 4 in global system.
   - NSP: Desired number of spans for section (input by user).

5. **Output Arguments**:
   - XYZ: 4 X and 2 Y local coord. for each element of section.
   - Z: Z coordinate for all elements in section (from trans).
   - TRS: 2 by 2 trans matrix to go from local to global.
   - NNCH: Element number (W.R.T. total structure) of last element in span. (Used for labeling plot).
   - SW: Span Width.
XLE  Average X coordinate of leading edge per span.
NE  Number of elements in section.
NETSV  Counter of total number of elements in structure.

6. **Common Blocks Used:**
/BASIC/

7. **Error Returns:**  None

8. **Calling Sequence:**
SUBROUTINE MESH(CORNX,CORNY,CORNZ,XYZ,Z,TRS,NNCH,SW,XLE,NSP,
NE,NETSV).

9. **Input Tapes:**  None
10. **Output Tapes:**  None
11. **Scratch Tapes:**  None
12. **Storage Required:**  459 words
13. **Subroutine User:**  MAIN
14. **Subroutine Required:**  TRANX
15. **Remarks:**  None
1. **Subroutine Name:** MESHCL

2. **Purpose:**
To generate a mesh (grid) for a section with 1 or 2 control lines.

3. **Procedure:**
The procedure is similar to subroutine MEXH except that element size is determined by control lines.

4. **Input Arguments:**
- CORNX, CORNY, CORNZ: X, Y and Z coord. of corner pts. 1, 2, 3, 4 in global system.
- NSF: Desired number of spans for section (input by user).
- ICL: Number of control lines in section.
- CL(1), CL(2): x coord. of first control line.
- CL(3), CL(4): x coord. of second control line.

5. **Output Arguments:**
- XYZ: 4 X and 2 Y local coord. for each element of section.
- Z: Z coordinate for all elements in section (from trans).
- TRS: 2 by 2 trans matrix to go from local to global.
- NNCH: Element number (W.R.T. total structure) of last element in span. (Used for labeling plot).
- SW: Span width.
XLE  Average X coordinate of leading edge per span.
NE  Number of elements in section.
NETSV  Counter of total number of elements in structure.

6. Common Blocks Used:
/BASIC/

7. Error Returns: None

8. Calling Sequence:
Subroutine MESHCL(CORNX,CORNY,CORNZ,XYZ,Z,TRS,NNCH,SW,XLE,
NSP,NE,NETSV,ICL,CL).

9. Input Tapes: None
10. Output Tapes: None
11. Scratch Tapes: None
12. Storage Required: 829 words.
13. Subroutine User: MAIN
14. Subroutine Required: TRANZ
15. Remarks: None
1. **Subroutine Name:** MFUN

2. **Purpose and Equations:**
   a. Generate modes by using a quadratic polynomial
   b. The coefficients are supplied by a table - CTABLE
   c. The wing number and mode number are stored in IFTABL
   d. The derivative is formed and stored in DM
   e. \[ EM = C_0 + C_x X + C_y Y + C_{xy} XY + C_{xx} X^2 + C_{yy} Y^2 \]
      \[ DM = C_x + C_{xy} Y + 2C_{xx} X^2 \]

3. **Input Arguments:**
   JMODE = number modes

4. **Common Blocks:**
   a. Common/MFUN/CTABLE,IFTABL,NTABL,NFQF
      This common block contains input:
      1) CTABL = coefficient tables
      2) IFTABL = table containing wing number and mode number
      3) NEQF = number equations
   b. Common/MODEF/EM,DM,X,Y,J1,J2
      1) EM = mode output
      2) DM = derivative = output
      3) X = value of x (input)
      4) Y = value of y (input)
      5) J1 = wing number (input)
      6) J2 = mode number (input)

5. **Calling Sequence:**
   Call MFUN (JMODE)

6. **Subroutine User:** RDMODE
1. **Subroutine Name:** MXERR

2. **Purpose:**
   Whenever a program limit has been exceeded, this subroutine will write an error message to inform the user.

3. **Input Arguments:**
   - **IERR** = identifies error
   - **MXNO** = program limit
   - **NOIN** = value in error
   - **I6** = output tape for printing

4. **Calling Sequence:**
   Call MXERR(IERR,MXNO,NOIN, I6)

5. **Output:**
   An error message is printed for the following parameters if their limit is exceeded:
   - Number wings
   - Number sections
   - Number modes
   - Number spaces
   - Number corner points
   - Number frequencies
   - Number elements/span
1. Subroutine Name: OUTP1

2. Purpose:
   To output the LIFT AND PRESSURE MOMENT PER UNIT SPAN FOR
   WING - and the VELOCITY POTENTIALS AND ELEMENT PRESSURES FOR
   WING - tables.

3. Equations and Procedures:
   Information is printed out in tabular form.

4. Input Arguments:
   Defined by comment statements at the beginning of subroutine.

5. Output Arguments: None

6. Common Blocks: None

7. Error Returns: None

8. Calling Sequence:
   Call OUTP1 (MACH,FREQ,JMODE,JWING,NS,NSP,SW,DLSAP,PMSPAN,
   XP,INDXI,PRES,IPW,I6,LINE,LMAX,NNCH,NE,TITLE,NSPT)

9. Input Tapes: None

10. Output Tapes: None

11. Scratch Tapes: None

12. Storage Required: 650 words

13. Subroutine User: MAIN

14. Subroutine Required: None

15. Remarks: None
1. **Subroutine Name:** OUTP2

2. **Purpose:**
   Print the Total Lift and Pitching Moment tables.

3. **Equations and Procedures:**
   Information is printed out in tabular form.

4. **Input Arguments:**
   - **I6:** Logical unit number of output device (printer)
   - **NWING:** number of wings in structure
   - **TL(I,J):** Total lift on wing I due to mode J
   - **TM(I,J):** Total moment on wing I due to mode J
   - **JMODE:** Total number of modes for run
   - **FREQ:** Frequency
   - **MACH:** Mach number
   - **LINE:** Current line count
   - **LMAX:** Maximum number of lines per page

5. **Output Arguments:** None

6. **Common Blocks Used:** None

7. **Error Returns:** None

8. **Calling Sequence:**
   Call OUTP2 (I6,NWING,XP,TL,JMODE,FREQ,MACH,LINE,LMAX)

9. **Input Tapes:** None

10. **Output Tapes:** None

11. **Scratch Tapes:** None

12. **Storage Required:** 260 words

13. **Subroutine User:** MAIN

14. **Subroutine Required:** None

15. **Remarks:** None
1. Subroutine Name: OUTP3

2. Purpose:
   Write the table of GENERALIZED AERODYNAMIC COEFFICIENT MATRIX
   IN AGARD DEFINITION.

3. Procedure:
   Information is printed out in tabular form.

4. Input Arguments:
   I6   Logical unit number of printer
   MACH  Mach number
   FREQ   Frequency
   Q  Aerodynamic coefficient matrix
   LINE  Current line
   LMAX  Total number of lines allowed on page
   TITLE  Print title information, 4 characters per word, 56 characters in total

5. Output Arguments: None

6. Common Blocks Used: None

7. Error Returns: None

8. Calling Sequence:
   Call OUTP3 (I6,MACH,GREQ,JMODE,Q,LINE,LMAX,TITLE)

9. Input Tapes: None

10. Output Tapes: None

11. Scratch Tapes: None

12. Storage Required: 430 words

13. Subroutine User: MAIN

14. Subroutine Required: None

15. Remarks: None
1. Subroutine Name: PHIL
2. Purpose: Integration of velocity potentials.
3. Equations and Procedure:
The velocity potentials are integrated separately for each span. The equation and method is given in detail in Reference #1. This routine also determines the convergence of the velocity potentials when wake effect is considered.
4. Input Arguments:
   XLE  X coordinate of the center of the leading edge line for the span.
   XCEN X coordinates of the center of the elements of the span.
   DPDX Functional values to be integrated (COMPLEX).
   NEPS The number of elements per span.
   A Complex constant used as exponent.
       A = (0.0, K'*M).
   CXP Complex expotentials taken at the center of the element. CXP(I) = CEXP (0.0, XCENLI)*K'*M).
5. Output Arguments:
   PHI Velocity potential for the span.
   PHIW This array is used as both input and output to test for wake convergence. On input it is the velocity potential of the last iteration. On output it is identical to PHI.
6. Common Blocks Used: /WAKE/
7. Error Returns: None
8. **Calling Sequence:**
   Call PHIL(XLE, XCEN, DPDX, NEPS, PHI, A, CXP, PHIW)

9. **Input Tapes:** None

10. **Output Tapes:** None

11. **Scratch Tapes:** None

12. **Storage Required:** 914 words

13. **Subroutines Required:** None

14. **Subroutine User:** MAIN

15. **Remarks:** None
1. **Subroutine Name:** PLOT

2. **Purpose:**
   This routine positions the pen of the plotter and skips plot frames on the CDC version of the plotter.

3. **Equations and Procedure**
   The Langley plotting routines, CALPLT and NFRAME, are called in place of the standard CALCOMP routine PLOT.

4. **Input Arguments**
   X, Y, IPEN These are the arguments of the standard CALCOMP routine PLOT.

5. **Output Arguments:** None

6. **Common Blocks Used:** None

7. **Error Returns:** None

8. **Calling Sequence:**
   Call PLOT(X,Y, IPEN)

9. **Input Tapes:** None

10. **Output Tapes:** None

11. **Scratch Tapes:** None

12. **Storage Required:** Relatively small

13. **Subroutines Required:** CALPLT, NFRAME

14. **Subroutine User:** PLOTGD

15. **Remarks:**
   This routine is not used in the IBM version of the program. Instead, the standard CALCOMP routine PLOTS, with entry point PLOT is used.
1. **Subroutine Name:** PLOTGD

2. **Purpose:**
   Generate a plot of each section of a wing.

3. **Equations and Procedure:**
   Standard CALCOMP plotting techniques are used.

4. **Input Arguments:**
   - NE: Number of elements in each section of the wing
   - XYZ: X and Y coordinates of each element (aerodynamic Reference System)
   - NNCH: Array containing last element number of each span
   - NSP: Number of spans in each section of the wing
   - NSECT: Number of sections in the wing
   - TITLE: 56 characters of title information. 14 words, 4 characters in each word
   - JWING: Wing number
   - ZSECT: Z coordinate of all elements in the section
   - AR: Aspect Ratio
   - MACH: Mach Number

5. **Output Arguments:** None

6. **Common Blocks Used:** None

7. **Error Returns:** None

8. **Calling Sequence:**
   Call PLOTGD(NE,XYZ,NNCH,NSP,NSECT, TITLE,JWING,ZSECT,AR,MACH)

9. **Input Tapes:** None

10. **Output Tapes:** None

11. **Scratch Tapes:** None
12. Storage Required: 740 words

13. Subroutine Required:
This routine requires the following CALCOMP routines:
NUMBER, SYMBOL, LINE, PLOT

14. Subroutine User: MAIN

15. Remarks: None
1. **Subroutine Name**: PTGRID

2. **Purpose**: 
   To print out the aerodynamic grid and calculate the X coordinate of the center of each element.

3. **Equations and Procedure**: 
   The coordinates of each point are found in the structural reference (or rotated structural reference) system. These are then printed out in tabular form along with the X and Y coordinates of the center of the element. The X coordinate of the center of the element is stored (in the aerodynamic system) in the array XCEN.

4. **Input Arguments**: 
   - **I6**: Logical unit number of output device. 
   - **TITLE**: Array containing title information (56 characters, 14 words, 4 characters each). 
   - **MACH**: Mach number 
   - **NWING**: Number of wings 
   - **NZECT**: Number of sections in each wing. 
   - **NP**: Number of spans in section I of wing J-NP(I,J). 
   - **NNCH**: Array containing the element number of the last element of each span. 
   - **XYZ**: X and Y coordinates of the structure in the aerodynamic system. 
   - **ZECT**: Z coordinates of each section. 
   - **REFLEN**: Input reference length. 
   - **BETA**: BETA=SQR(MACH**2-1.0).
5. **Output Arguments:**
   
   XCEN \hspace{1em} X coordinates of the center of each element

6. **Common Blocks Used:** None

7. **Error Returns:** None

8. **Calling Sequence:**
   
   Subroutine PTGRID (I6,TITLE,MACH,NWING,NSECT,NSP,NNCH,XYZ,
   ZSECT,REFLEN,BEETA).

9. **Input Tapes:** None

10. **Output Tapes:** None

11. **Scratch Tapes:** None

12. **Storage Required:** 406 words

13. **Subroutine User:** MAIN

14. **Subroutine Required:** None

15. **Remarks:** None
1. Subroutine Name:   RDETA

2. Purpose:  
   Read DETADX, ETA arrays from Tape II3

3. Input Arguments:  
   II3 = Tape number  
   NET = number of elements

4. Output Arguments:  
   DETADX = array on tape II3  
   ETA = array on tape II3

5. Scratch Tapes:   II3

6. Subroutine User:  
   Main program

7. Calling Sequence:  
   Call RDETA (II3,DETADX,ETA,NET)
1. **Subroutine Name:** RDMODE

2. **Purpose:**
   Read and define modal input.

3. **Equations and Procedure:**
   Mode type of input is determined. (MDWING or MDPOLY)
   If MDPOLY is present, the coefficient of the polynomials are read, and the modal values and their derivatives at each element is found using MFUN. If MDWING data is present, the spline routine obtained from Robert Desmarais, is used. Cards from program XX31 are read by subroutines READAB and READXY. Note that this routine is really ZFUN with minor changes. The next data card is read. If it is another MDWING card, the above procedure is repeated. If it is a RIGM card, the rigid mode data is read. This is done until an END card is reached.

4. **Input Arguments:**
   - NWING: Number of wings in structure
   - NSECT: Number of sections in each wing.
   - NE: Number of elements in each section of each wing.
   - XCEN: X coordinate of the center of each element in the aerodynamic grid in the aerodynamic reference system.
   - XYZ: X and Y coordinates of the structure in the aerodynamic system.
   - BEETA: SQRT(MACH*MACH-1.0)
   - REFLEN: Reference length
NET  Total number of elements.
LAB  Label on last card read.
ID   Integer field of last card read.
I5   Unit number of device containing the input deck.
I6   Unit number of device for printout.

5. **Output Arguments:**

ETA  The modes, for all modes and all elements.
DETADX The derivative of the mode at every point.
ERROR Logical variable set equal to .TRUE. if an error is encountered.

6. **Common Blocks Used:**

/MODE/
/MODEF/
/XYCAL/

7. **Error Returns:**
The program returns the logical variable ERROR equal to .TRUE. if an error condition is present in reading the input.

8. **Calling Sequence:**
Call RDMODE(JMODE,NWING,NSECT,NE,XCEN,XYZ,EEITA,REFLEN,NET,
LAB,ID,ERROR,ETA,DETADX,I5,I6).

9. **Input Tapes:** None
10. **Output Tapes:** None
11. **Scratch Tapes:** None
12. **Storage Required:** 1896 words
13. **Subroutines Required:**
ZFDZ, READAB, MFUN and READXY.
14. Subroutine User: MAIN
15. Remarks: None
1. **Subroutine Name:** READAB

2. **Purpose:**
To read in the spline coefficients used for finding mode information.

3. **Equations and Procedure:**
This reads in the surface spline coefficients and the modal values as punched from program $x31$. This program was received from and written by Robert Desmarais. The modes are read and used only to see if the mode is constant for all points in the grid. For this reason the modes and the spline coefficients may have the same storage unless the user of the routine wishes to use both the modal values at the structural grid and the spline coefficients.

4. **Input Arguments:**
- Z  
  Storage to read in modal values.
- A  
  Storage to read in surface spline coefficients.

5. **Output Arguments:**
- Z  
  Modal values for the structural grid.
- A  
  Surface spline coefficients.

6. **Common Blocks Used:**
- /ZFUNNY/ and /ZFDZI0/

7. **Error Returns:** None

8. **Calling Sequence:**
CALL READXY(Z, A)

(Note that if the modes at the structural grid are not of interest, Z may share the same storage as A).
9. **Input Tapes:** None
10. **Output Tapes:** None
11. **Scratch Tapes:** None
12. **Storage Required:** 230 words
13. **Subroutines Required:** None
14. **Subroutine User:** RDMODE
15. **Remarks:**

   This subroutine corresponds to entry point READZAB of the function subprogram ZFUN written by Robert Desmarais of Langley Research Center. READAB is to be used in conjunction with subroutines READXY and ZFDZ.
1. Subroutine Name: READXY

2. Purpose:
   To read in the structural grid used for spline data.

3. Equations and Procedure:
   This reads in the structural grid for the spline routine.
   This is a modified version of subroutine ZFUN. It reads
   in the number of points in the structural grid and the X
   and Y coordinates of the structural grid, exactly as it was
   punched from Program $231. ($231 was written by
   Robert Desmarais of Langley Research Center).

4. Input Arguments: None

5. Output Arguments:
   X    X coordinates of the structural surface.
   Y    Y coordinates of the structural surface.

6. Common Blocks Used: /ZFUNNY/ and /ZFDZIO/

7. Error Returns
   IERF is set equal to 680 if the number of points to be read
   exceeds the maximum allowed by the program. IERF is in
   common /ZFUNNY/

8. Calling Sequence
   CALL READ(X,Y)

9. Input Tapes: None

10. Output Tapes: None

11. Scratch Tapes: None

12. Storage Required: 210 words

13. Subroutines Required: None
14. **Subroutine User:** RDMODE

15. **Remarks:**

This subroutine corresponds to entry point READZXY of the function subprogram ZFUN, written by Robert Desmarais.

READXY is to be used in conjunction with subroutines READAB and ZFDZ.
1. **Subroutine Name:** RESTART

2. **Purpose:**
   Read and check restart tape.

3. **Equations and Procedure:**
   The first record of the "restart" tape is read and compared against the current run. If the tape is compatible with the current run, execution proceeds. If not, a table is printed out giving the information on the restart tape and for the current run. If wake elements are requested, this same test is made on the wake effect tape, IL5.

4. **Input Arguments:**
   - **MACH** Mach number
   - **NET** Total number of elements
   - **NFREQ** Number of frequencies
   - **FREQ** List of frequencies
   - **REFLEN** Reference length
   - **IWTE** Wake effect code
   - **I8** Logical unit number of restart tape containing velocity influence coefficients.
   - **IL5** Logical unit number of restart tape containing wake effects.
   - **I6** Logical unit number of output device.
   - **FREQR** Space used to read frequencies from restart tape.
   - **XYZ** X and Y coordinates of elements.

5. **Output Arguments**
   - **AREA** Area of each element.
6. **Common Blocks Used**
   /NEXTCX/ (see Error Returns)

7. **Error Returns**
   IFLUSH in COMMON/NEXTCX/ is set equal to 1 if either one of the restart tapes are not compatible with the present run. This forces termination of the program. The program will not cycle to the next case to avoid writing over the restart tape.

8. **Calling Sequence:**
   Call RSTART(MACH, NET, NFREQ, FREQ, REFLEN, IWTE, I8, I15, I6, FREQR, XYZ, AREA).

9. **Input Tapes:**
   Logical unit I8 and I15, the restart tapes. I15 is optional and needed only when wake is required in the current run.

10. **Output Tapes:** None

11. **Scratch Tapes:** None

12. **Storage Required:** 450 words

13. **Subroutines Required:** None

14. **Subroutine User:** MAIN

15. **Remarks:** None
1. Subroutine Name: RTOI

2. Purpose:
This routine transforms a pair of coordinates from the reference system of the receiving element to the reference system of the influencing system.

3. Equations and Procedures:
The point is first transformed from the receiving system to the reference system. This new set of coordinates is then put in the influencing system. The transformation matrix to relate the velocity influence from the influencing system to the receiving system is also defined.

4. Input Arguments:
TR  Transformation matrix to go from receiving system to the reference system.
TI  Transformation matrix to go from the influencing system to the reference system.
YCR  Y coordinate in the receiving system.
XCR  Z coordinate in the receiving system.

5. Output Arguments:
ZRO  Z coordinate of the receiving point in the influencing system.
ZLO  Z coordinate of the mirror image of the receiving point in the influencing system.

6. Common Blocks Used:
/RRLL/

7. Error Returns: None
8. Calling Sequence:
   Call RTOI (TR, TI, YCR, ZCR, ZRO, ZLO)

9. Input Tapes: None

10. Output Tapes: None

11. Scratch Tapes: None

12. Storage Required: 200 words

13. Subroutines Required: None

14. Subroutine User: LOOPW

15. Remarks: None
1. **Subroutine Name:** SOLVE

2. **Purpose:**
   
   Solution of the linear system of equations $C \times X = B$.

3. **Equations and Procedure:**

   Back substitution based on the factored form of the coefficient matrix. See Algorithm 423, "Collected Algorithms from CACM", by Cleve Moler.

4. **Input Arguments:**

   - $N$: Order of the Matrix $A$
   - $NDIM$: First dimension of a declared in calling program. If $A$ is singly subscripted in calling program, $SFT \ NDIM - N$.
   - $A$: Contains the triangular factors of the matrix $C$ (as determined by subroutine "DECOM").
   - $B$: On input, the RHS vector
   - $IP$: Vector of dimension $N$ containing Row interchange information (as determined by subroutine "DECOM").

5. **Output Arguments:**

   - $B$: The solution vector

6. **Common Blocks Used:** None

7. **Error Returns:** None

8. **Calling Sequence:**

   Call SOLVE($N$,$NDIM$,$A$,$B$,$IP$)

9. **Input Tapes:** None
10. **Output Tapes:** None
11. **Scratch Tapes:** None
12. **Storage Required:** 249 words
13. **Subroutines Required:** None
14. **Subroutine User:**
   MAIN
15. **Remarks:**
   Subroutine SOLVE must be used in conjunction with subroutine DECOM.
1. **Subroutine Name:** SONS

2. **Purpose:**
   Determines if the elements of one section influence the elements of any other section.

3. **Equations and Procedure:**
   Each section is treated as having two receiving points and two influencing points. The receiving points are then taken to be the corner at the trailing edge of the section. The influencing points are the corner points of the section at the leading edge. Each section is examined to see if any other sections "influencing points" have any influence on its receiving points. The Boolean matrix $\text{ISONS}(K1,K2)$ is determined such that $\text{ISONS}(K1,K2)=1$ if section $K2$ has influence of section $K1$, $\text{ISONS}(K1,K2) = 0$ if section $K2$ has no influence on section $K1$.

4. **Input Arguments:**
   - NWING Number of wings
   - NSECT Number of sections in each wing
   - CORNX Array of X, Y, and Z coordinates for the wing corner points
   - CORNY
   - CORNZ

5. **Output Arguments:**
   - ISONS See Equations and Procedures section above.

7. **Common Blocks Used:** None

8. **Calling Sequence:**
   Call SONS(NWING,NSECT,CORNX,CORNY,CORNZ,ISON)

9. **Input Tapes:** None

10. **Output Tapes:** None
11. **Scratch Tapes:** None

12. **Storage Required:** 430 words

13. **Subroutine User:** MAIN

14. **Subroutine Required:** None

15. **Remarks:** None
1. **Subroutine Name:** \texttt{XONXPT}

2. **Purpose:**
   Print out a table to show if any section is influenced by any other section.

3. **Procedure:**
   \texttt{IXON} (see Subroutine \texttt{XON} write-up) is printed in tabular form.

4. **Input Arguments:**
   \texttt{IXON} See write up, (Equations and Procedure Section) for Subroutine \texttt{XON}.
   \texttt{NWING} Number of wings.
   \texttt{N\_ECT} Number of sections in each wing.
   \texttt{MACH} Mach number (REAL)
   \texttt{I6} Logical unit number of prints.

5. **Output Arguments:** None

6. **Common Blocks Used:** None

7. **Error Returns:** None

8. **Calling Sequence:**
   Call \texttt{XONXPT(IXON,NWING,N\_ECT,MACH,I6)}

9. **Input Tapes:** None

10. **Output Tapes:** None

11. **Scratch Tapes:** None

12. **Storage Required:** 325 words

13. **Subroutine User:** MAIN

14. **Subroutine Required:** None

15. **Remarks:** None
1. **Subroutine Name**: SYMBOL

2. **Purpose**: To plot character information in the CDC version of the program.

3. **Equations and Procedure**
   This routine calls the Langley plotting routine NOTATE.

4. **Input Arguments**: The input arguments are identical to either the standard CALCOMP routine SYMBOL or the Langley plot routine NOTATE.

5. **Output Arguments**: None

6. **Common Blocks Used**: None

7. **Error Returns**: None

8. **Calling Sequence**: Call SYMBOL (X, Y, SIZE, CHAR, ANG, NCHAR)

9. **Input Tapes**: None

10. **Output Tapes**: None

11. **Scratch Tapes**: None

12. **Storage Required**: None

13. **Subroutine Required**: NOTATE

14. **Subroutine User**: PLOTGD

15. **Remarks**: This routine is only used in the CDC version of the program. In the IBM version, this routine is replaced by the standard CALCOMP routine SYMBOL.
1. **Subroutine Name:** TIMOUT

2. **Purpose:**
   To print in tabular form the elapsed time from the last call to this routine.

3. **Equations and Procedure:**
   Procedure is fairly obvious. Note that eight scalars are used to receive title information. Scalars were used instead of an array to permit the same routine to be used on both the CDC and IBM machines. Scalars were needed because of the different word length of the two machines.

4. **Input Arguments:**
   - **MSEC**
     Integer value of current elapsed time in milliseconds.
   - **A, B, C, D, E, F, G, and H**
     Eight scalars used to input 4 characters of title information each.

5. **Output Arguments:** None

6. **Common Blocks Used:**
   /TAPE/

7. **Error Returns:** None

8. **Calling Sequence:**
   Call TIMOUT(MSEC,A,B,C,D,E,F,G,H)

9. **Input Tapes:** None

10. **Output Tapes:** None

11. **Scratch Tapes:** None
12. Storage Required: 124 words
13. Subroutine Required: None
14. Subroutine User: MAIN
15. Remarks:

This routine prints the time on logical unit J9, the seventh scalar (integer) in COMMON/TAPE/. It is suggested that this output device be separate from the output device of the rest of the run.
1. **Subroutine Name:** TMAX

2. **Purpose:**
   1. Generate maximum value in array.
   2. Test each element in the array and determine whether it is zero relative to the maximum.
   3. Let element = 0.0 if it is relatively = 0.0.

3. **Input Requirements:**
   - NET = number elements in array
   - A = array of real numbers

4. **Output Arguments:**
   - A = array which has been set equal to zero in those locations which the original element = relatively small.
   - GBAR = maximum value

5. **Subroutine User:** ITRATE

6. **Calling Sequence:**
   - Call TMAX (NET, A, GBAR)

7. **Subroutine Used:** CABSO
1. **Subroutine Name:** TRANS
2. **Purpose:**
   To find the transformation matrix to transform coordinates from one rectangular system to another. This routine is used to find the transformation matrix from the reference system of the section to the main reference system.
3. **Equations and Procedure:**
   The equations are found in Reference 1.
4. **Input Arguments:**
   \( X, Y, Z \)
   \( X, Y, Z \) coordinates of the corner points of the structure
5. **Output Arguments**
   \( X_P, Y_P \)
   \( X \) and \( Y \) coordinates of the corner points of the section in the local reference system of the section
   \( Z_C \)
   The local \( Z \) coordinate of all points in the section
6. **Common Blocks Used:** None
7. **Error Returns:** None
8. **Calling Sequence:**
   Call TRANS \( (X, Y, Z, T, X_P, Y_P, Z_C) \)
9. **Input Tapes:** None
10. **Output Tapes:** None
11. **Scratch Tapes:** None
12. **Storage Required:** 266 words
13. **Subroutines Required:** None
14. **Subroutine User:**
   MESH, MESHCL and GRIDIN
15. **Remarks:** None
1. **Subroutine Name:** WAKET

2. **Purpose:**
   Write Wake element effects on tape 112 for 1 frequency

3. **Input Arguments:**
   - FRQ = frequency
   - I12 = output tape number
   - IF = frequency number
   - LRECLW = length of record on tape I8
   - K1 = number frequencies

4. **Output Arguments:**
   - IZ = non zero elements in record
   - AROW = elements selected for given frequency stored on tape I12
   - WROW = record from tape I8 containing wake effects coefficients
   - NNCH = row numbers of trailing edge elements

5. **Calling Sequence:**
   Call WAKET (I12,AROW,WROW,IF,LRECLW,NSPT,18,IZ,K1,NNCH,FRQ)

6. **Files:**
   - I12 output tape containing effects for 1 frequency
   - I8 input tape contains effects for all frequencies

7. **Subroutines Used:** WRITZW

8. **Subroutine User:** DIAG

9. **Calling Sequence:**
   Call DIAG (WROW,IF,LRECL,NNCH,MC,D,NET,AROW,IZ,IP,NSPT,FRQ,CWROW,CAROW)
10. **Common Blocks:**
   - COMMON/TAPE/ Defines tape storage
   - COMMON/VELCOM/ NW block and NWROW
   - COMMON/WAKE/ Defines wake effect controls
   - COMMON/WAKEUP Defines wake effects variables

11. **Files:**
   - I9 = Input file containing W coefficients
   - II5 = Input file containing Wake effects
   - I10 = Output file containing D block compound
   - I11 = Output file containing block off diagonals
   - I12 = Output file containing WAKE effects

12. **Subroutines Used:**
   1. FREQW
   2. WAKET
   3. WRITED
   4. DECOM
1. **Subroutine Name:** WINGCK

2. **Purpose:**
   Check on the validity of the corner points of a wing.

3. **Equation and Procedure**
   The corner points of a given wing are checked in pairs. The leading corner point must have an X coordinate that is less than or equal to the trailing corner point. The Y and Z coordinates of the leading corner point must be the same as the Y and Z coordinates of the trailing corner point, respectively.

4. **Input Arguments**
   - **CORNX**
   - **Corny**
   - **CORNZ**
     Input X, Y, and Z coordinates of the wing.
   - **NWING**
     Wing number, used in error message printout.
   - **ICN**
     Number of corner points.
   - **I6**
     Logical unit number of output device.

5. **Output Arguments**
   - **ERROR**
     Error code, set equal to .TRUE. if an error condition exists.

6. **Common Blocks Used:** None

7. **Error Returns:**
   See Equations and Procedure and definition of ERROR above.

8. **Calling Argument:**
   Call WINGCK(CORNX, CORNY, CORNZ, NWING, ICN, ERROR, I6)

9. **Input Tapes:** None
10. Output Tapes: None
11. Scratch Tapes: None
12. Storage Required: 175 words
13. Subroutines Required: None
14. Subroutine User: WINGIN
15. Remarks: None
1. **Subroutine Name**: WINGIN

2. **Purpose**: To read and check the geometric input. All geometric input label cards with the exception of the SPAN and CHORD sections are read here.

3. **Equations and Procedure**: Each label card is read with the same format: READ (I5,1000) LAB,ID,FD; 1000 FORMAT (A4,2X,9I2,4E12.0), where ID and FD are dimensional 9 and 4 respectively. The program then branches to the section that processes that label section.

4. **Input Arguments**:
   - I5: Logical unit number containing the input deck - usually the card reader
   - I6: Logical unit number of the output unit. Usually the system printer
   - BEETA: SQRT (MACH*MACH-1)
   - REFLLEN: Reference length

5. **Output Arguments**: 
   - NWING: Number of wings in structure
   - NSECT: Array containing the number of sections for each wing
   - AR: Aspect ratio of each wing - array
   - NSP: NSP(I,J) is the number of spans for section I of wing J
   - CORNX: The X, Y, and Z coordinates of the corner points of each wing XORNX(I,J) is the X coordinate of corner point I of wing J.
SYM Array, SYM(J) is the symmetry code of wing J

ICL Code for section I of wing J ICL(I,J) = 1, specific grid information is input.
ICL(I,J) = 0, MESH generated, no control lines.
ICL(I,J) = 1, or 2, MESH generated with 1 or 2 control lines.

CL X coordinate of control lines for each section.
CL(K,I,J) for section I of wing J.
K = 1, and 2 for first and second coordinates of first control line of the section.
K = 3, 4 for first and second coordinates of second control line in section.

IWP Print control for each wing
ERROR Set equal to .TRUE. if an error is found.
NFREQ Number of frequencies
FREQ List of frequencies
LAB Label of last card read
ID Integer field of last card read
FD Floating point field of last card read

6. Common Blocks Used: None

7. ERROR Returns:
Logical ERROR is set equal to .TRUE. if any of several errors exist in reading the input. The error condition generates an appropriate error message. All error message format statements are numbered between 2000 and 2022. (All messages are preceeded by the statement ERROR IN READING INPUT.)
Messages a through f have to do with errors on the WING data card, and are preceded by the message:

WING DATA CARD IS INCORRECT.

a) DATA FOR WING - WAS READ WHEN DATA FOR WING - WAS EXPECTED BY THE PROGRAM.
(WING data cards must appear in consecutive order in the deck. Each WING data card is the first card of a WING data group.)

b) WING NUMBER MUST BE GREATER THAN ZERO AND LESS THAN OR EQUAL TO 4. IT WAS READ AS ____.
IT WILL BE TREATED AS WING 1 FOR CHECK PURPOSE ONLY.

c) NO MORE THAN FOUR WING DATA CARDS MAY BE INPUT.

d) THE NUMBER OF SECTIONS INPUT FOR WING -- IS ____.
THIS IS GREATER THAN THE MAXIMUM PER WING OF 3.

e) NUMBER OF CORNER POINTS, ____ IS NOT CONSISTANT WITH THE NUMBER OF SECTION ____ FOR WING ____.
(A wing with 1 section may have only 3 or 4 corner points. A wing with 2 sections, 5 or 6 corner points. A wing with 3 sections, 7 or 8 corner points.)

f) SYMMETRY CODE FOR WING ____, IS INPUT AS ____.
SYMMETRY CODE MAY ONLY EQUAL -1, 0, OR 1.

g) "CORNER" DATA CARD IS MISSING FOR WING ________.
(The CORNER cards defining a wing must immediately follow the WING card for that wing.)

Messages h through l have to do with errors on the LINE data card.
They are preceded by the following message
ERROR ON CONTROL "LINE" DATA CARD FOR WING ___.

h) MORE THAN TWO CONTROL LINES ARE SPECIFIED FOR
SECTION ____.

i) CONTROL LINES ARE NOT INPUT IN ORDER FOR SECTION ____.

j) CONTROL LINE IS SPECIFIED FOR SECTION ____ OF WING ____.
   THIS WING HAS ____ SECTIONS (CONTROL LINE IGNORED).

k) CONTROL LINE ____ FOR SECTION ____ IS NOT DEFINED
   TO BE INSIDE OF THE SECTION.

l) CONTROL LINE 2 OF SECTION ____ IS ABOVE THE FIRST
   CONTROL LINE.(In a section with 2 control lines, the
   leading control line must always be defined first
   and designated as control line number 1).

m) MODE DATA IS MISSING OR OUT OF ORDER.

n) FREQUENCY DATA IS MISSING OR OUT OF ORDER.

o) A LABEL CARD IS EXPECTED BUT NOT PRESENT.
   CARD WILL BE PRINTED ON THE FOLLOWING LINE AND THE
   NEXT CARD WILL BE READ.

p) ERROR ON GRID LABEL CARD GRID DATA PRESENT FOR
   WING ____.
   SECTION ____
   THIS SECTION HAS NOT BEEN DEFINED FOR THE STRUCTURE.

8. **Calling Sequence:**
   Call WINGIN( I5, I6, NWING, NSECT, AR, NSP, CORNX, CORNY, CONRZ,
   SYM, ICL, CL, ERROR, BEETA, IPW, NFREQ, FREQ, REFLEN, LAB, ID, PD)

9. **Input Tapes:** None
10. Output Tapes: None
11. Scratch Tapes: None
12. Storage Required: 2022
13. Subroutine User: MAIN
14. Subroutine Required: WINGCK
15. Remarks: None
1. **Subroutine Name:** WINTGR

2. **Purpose:**
   To organize the computation of velocity influence coefficients of a pair of elements.

3. **Equations and Procedure:**
   The influence of a single line and its mirror image is determined with respect to a given receiving point. This influence is combined with the influence of the preceding (lower) line in the span, (if any). Subroutine EONE is used to determine if there is any influence at all, and subroutine WVINT calculates the frequency independent terms. Wake effects are also calculated.

4. **Input Arguments:**
   X,Y  X and Y coordinates of points of the influencing element.

5. **Output Arguments:**
   WROW  Array of influence coefficients for all frequencies.

6. **Common Blocks Used:**
   
   /EEW/
   /EXCEED/
   /FQ1/
   /FKTEST/
   /RRL/
   /WAKE/
   /WAKEUP/
   /WVL/
   /WWl/
7. Error Returns: None
8. Calling Sequence:
   Call WINTGR(X,Y,WROW)
9. Input Tapes: None
10. Output Tapes: None
11. Scratch Tapes: None
12. Storage Required: 1630 words
13. Subroutine Required: WVINT
14. Subroutines User: LØØPW
15. Remarks: None
1. **Subroutine Name:** WRETA

2. **Purpose of Procedures:**
   WDETA, WETA arrays are stored on tape I13 - each mode is a row.

3. **Input Arguments:**
   - NET = no. of elements
   - JMODE= no. modes
   - WDETA= WDETA array
   - WETA = WETA array
   - I13 = tape no.

4. **Calling Sequence:**
   Call WRETA(NET,JMODE,WDETA,WETA,I13)

5. **Scratch Tapes:** I13

6. **Subroutine User:** MAIN program
1. Subroutine Name: WRITEO

2. Purpose and Procedures:
Test for nonzero elements in a row and store the nonzero elements and the corresponding row numbers on tape. The record is not written if NNZ = 0.

3. Input Arguments:
K = row numbers
AROW = array of numbers to be tested
NET = total number elements

4. Output Arguments:
IZ = array containing row numbers of nonzero element.
ANEW = array of nonzero elements

5. Calling Sequence:
Call WRITEO (K,AROW,IZ,ANEW,NET)

6. Scratch Tapes:
Ill = tape on which records are written

7. Subroutines Required:
WRITZ

8. Subroutine User: DIAG

9. Common Blocks:
/TAPE/I9,I10,Ill
1. **Subroutine Name:** WRITZ

2. **Purpose:**
   Write nonzero elements of coefficient matrix

3. **Equations and Procedures:**
   WRITE (112), K, NNZ, IZ, ANEW

4. **Input Arguments:**
   - Ill = tape number
   - K = row number
   - NNZ = number nonzero element numbers
   - IZ = array of element number
   - ANEW = nonzero elements of W. This array is real

5. **Calling Sequence:**
   Call WRITZ(Ill, K, NNZ, IZ, ANEW)

6. **Scratch Tape:**
   Ill = output tape containing ANEW records

7. **Subroutine User:** WRITEO

8. **Remarks:**
   This subroutine is different from WRITZW because ANEW is real.
1. **Subroutine Name:** WRITZW

2. **Purpose:**
   Write records for wake element coefficient matrix

3. **Equations and Procedures:**
   Write (I12) K,NNZ,IZ,ANEW

4. **Input Arguments:**
   - I12 = tape number
   - K = row number
   - NNZ = number of nonzero element numbers
   - IZ = array of element number
   - ANEW = nonzero elements of WTE. This array is complex.

5. **Calling Sequence:**
   Call WRITZW (I12,K,NNZ,IZ,ANEW)

6. **Scratch Tapes:**
   I12 = output tape containing ANEW records.

7. **Subroutine User:** WAKET

8. **Remarks:**
   This subroutine is different from WRITZ because ANEW is complex.
1. Subroutine Name: WTEPHT

2. Purpose:
Generate wake effect in computation of dn/dt.

3. Equations and Procedures:
\[ \frac{dn}{dt} = \frac{dn}{dt} - W_{TE}\phi \]
a) IZ and \( W_{TE} \) are read from tape I12
b) If the record is appropriate for the desired element, the nonzero terms are determined from the IZ array.
c) \( W_{TE} \) is post multiplied by \( \phi \) elements and subtracted from DEDT.
d) The result from step c) is stored back on DEDT.
Steps a), b), c), and d) are repeated for each element.

4. Input Arguments:
- \( DEDT \) = dh/dt array
- \( NET \) = number of elements
- \( PHIW \) = \( \phi \) array of length \( NET \)
- \( I12 \) = tape containing \( W_{TE} \) arrays
- \( W_{TE} \) = \( W_{TE} \) work array - only nonzero terms

5. Output Arguments:
- \( DEDT \) = revised \( dn/dt \)

6. Error Return: None

7. Calling Sequence:
Call WTEPHT(DEDT,NET,PHIW,IZ,I12,WTE)

8. Scratch Tapes:
\( I12 \) = tape containing \( NET \) records.
Each record is of form:
\( KEL,NNZ,IZ,WTE \).
9. Common Statements: None
10. Subroutine User: MAIN
11. Subroutines Required: None
1. **Subroutine Name:** WVINT

2. **Purpose:**
   To evaluate the velocity influence of a line on a point.

3. **Equations and Procedure:**
   The equations for this section are given in Reference 1.
   A Gaussian quadrature is used to evaluate the higher order terms of the H expressions.

4. **Input Arguments:** None

5. **Output Arguments:** None

6. **Common Blocks Used:**
   `/EEW/
   `/WVI/
   `/WWI/

7. **Error Returns:** None

8. **Calling Sequence:**
   Call WVINT

9. **Input Tapes:** None

10. **Output Tapes:** None

11. **Scratch Tapes:** None

12. **Storage Required:** 1384 words

13. **Subroutine Required:** None

14. **Subroutine User:** WINTGR

15. **Remarks:** None
1. Subroutine Name: ZFDZ

2. Purpose:
   To evaluate modal functions and its derivatives.

3. Equations and Procedure:
   This subroutine is a modified version of ZFUN, the routine written by Robert Desmarais of Langley Research Center.
   It calculates the modal values and the derivative for 1 aerodynamic grid point and one mode, given the coordinates of the structural grid system and the surface spline coefficients.

4. Input Arguments:
   X, Y  Structural grid coordinates
   A    Surface spline coefficients

5. Output Arguments
   ZFUN   Modal value at aerodynamic grid point
   DZDX   Derivative of the function (modal value) at the aerodynamic grid point.

6. Common Blocks Used:
   /XYCAL/ and /ZFUNNY/

7. Error Returns: None

8. Calling Sequence:
   CALL ZFDZ(ZFUN, DFDZ, X, Y, A)

9. Input Tapes: None

10. Output Tapes: None

11. Scratch Tapes: None
12. **Storage Required:** 151 words

13. **Subroutines Required:** None

14. **Subroutine User:** RDMODE

15. **Remarks:**

   This subroutine corresponds to the main entry of the function subprogram ZFUN, written by Robert Desmarais. It also includes the evaluation of the derivative done at entry point DZDX of ZFUN. Subroutine ZFDZ is to be used in conjunction with subroutines READXY and READAB. These two subroutines read the spline input data that is punched by program %31. (Program %31 was also written and received from Robert Desmarais).
Section 6
Source Program Listings

The source program listings for each subroutine are included in this section. The routines are indexed by deck number. A list of deck names and numbers can be found on the next page.

The program consists of 51 subprograms each with a unique sequence number. Columns 73, 74, 75 contain the "Deck" number and columns 76 through 80 contain the card sequence number for that subprogram. The first card is always ----0010 with successive increments of 10.
## Subroutine Deck Numbers

<table>
<thead>
<tr>
<th>Name</th>
<th>Deck No.</th>
<th>Name</th>
<th>Deck No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAIN</td>
<td>1</td>
<td>PLOTGD</td>
<td>26</td>
</tr>
<tr>
<td>BLOCK</td>
<td>2</td>
<td>PTGRID</td>
<td>27</td>
</tr>
<tr>
<td>CABSO</td>
<td>3</td>
<td>RDETA</td>
<td>28</td>
</tr>
<tr>
<td>DECOM</td>
<td>4</td>
<td>RDMODE</td>
<td>29</td>
</tr>
<tr>
<td>DIAG</td>
<td>5</td>
<td>READAB</td>
<td>30</td>
</tr>
<tr>
<td>ECHO</td>
<td>6</td>
<td>READXY</td>
<td>31</td>
</tr>
<tr>
<td>EFPLLOT</td>
<td>7</td>
<td>RSTART</td>
<td>32</td>
</tr>
<tr>
<td>EONE</td>
<td>8</td>
<td>RTOI</td>
<td>33</td>
</tr>
<tr>
<td>FGEN</td>
<td>9</td>
<td>SOLVE</td>
<td>34</td>
</tr>
<tr>
<td>FREQW</td>
<td>10</td>
<td>SONS</td>
<td>35</td>
</tr>
<tr>
<td>GETTIM</td>
<td>11</td>
<td>SONSPT</td>
<td>36</td>
</tr>
<tr>
<td>GRIDIN</td>
<td>12</td>
<td>SYMBOL</td>
<td>37</td>
</tr>
<tr>
<td>INPL</td>
<td>13</td>
<td>TIMOUT</td>
<td>38</td>
</tr>
<tr>
<td>IRELE</td>
<td>14</td>
<td>TMAX</td>
<td>39</td>
</tr>
<tr>
<td>ITRATE</td>
<td>15</td>
<td>TRANS</td>
<td>40</td>
</tr>
<tr>
<td>LOOPW</td>
<td>16</td>
<td>WAKET</td>
<td>41</td>
</tr>
<tr>
<td>MESH</td>
<td>17</td>
<td>WINGCK</td>
<td>42</td>
</tr>
<tr>
<td>MESCL</td>
<td>18</td>
<td>WINGIN</td>
<td>43</td>
</tr>
<tr>
<td>MFUN</td>
<td>19</td>
<td>WINTGR</td>
<td>44</td>
</tr>
<tr>
<td>MXERR</td>
<td>20</td>
<td>WRETA</td>
<td>45</td>
</tr>
<tr>
<td>OUTP1</td>
<td>21</td>
<td>WRITEO</td>
<td>46</td>
</tr>
<tr>
<td>OUTP2</td>
<td>22</td>
<td>WROITZ</td>
<td>47</td>
</tr>
<tr>
<td>OUTP3</td>
<td>23</td>
<td>WROITZW</td>
<td>48</td>
</tr>
<tr>
<td>PHIL</td>
<td>24</td>
<td>WTEPHT</td>
<td>49</td>
</tr>
<tr>
<td>PLOT</td>
<td>25</td>
<td>WVINT</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ZMDZ</td>
<td>51</td>
</tr>
</tbody>
</table>
**REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR**

```plaintext
DISPLAY...FILE DECK 01

<table>
<thead>
<tr>
<th>LINE</th>
<th>C</th>
<th>Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C</td>
<td>MAIN</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>C</td>
<td>CONSTANT POTENTIAL GRADIENT MEASUREMENT</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>CHANGE STORAGE TO 300, 70, 100 FROM 200, 43, 50</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>COMPLEX CARO (100)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>COMPLEX GE, PRES(300)</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>COMPLEX CREA</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>COMPLEX CROW (1200)</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>COMPLEX CKM</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>COMPLEX EXP (300)</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>COMPLEX DC WORK (900)</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>COMPLEX JET, DPOX</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>COMPLEX DLS, PH, DLOAD</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>COMPLEX OLSPN (100), P4PSPN (100)</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>COMPLEX ESWKC (11)</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>COMPLEX GT (300)</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>COMPLEX Q (10, 10)</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>COMPLEX PH (70)</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td>COMPLEX PHIA (300)</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td>COMPLEX SWOKC (1671)</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>COMPLEX TL(4, 10), TM(4, 10)</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
<td>COMPLEX WTE (100)</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td>COMPLEX WTER (1200)</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td>COMPLEX WTES</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td>COMPLEX WFL (300)</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>REAL IBUFF</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td>REAL IRCH</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td></td>
<td>INTEGER PRINT</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td>INTEGER TAPE3</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
<td>LOGICAL ERROR</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>LOGICAL PLOT</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td></td>
<td>LOGICAL PLOTR</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td>LOGICAL TECO</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td></td>
<td>LOGICAL TREDGE</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td></td>
<td>LOGICAL WAKE, WAKE1, WAKENZ</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td></td>
<td>DIMENSION AR(4), ISONS (12, 12)</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td></td>
<td>DIMENSION AREA (300)</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td></td>
<td>DIMENSION AROW (300)</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td></td>
<td>DIMENSION CWORK (4529)</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td></td>
<td>DIMENSION DWORK (3901)</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td>DIMENSION D1(60, 60), IP (60), DWNR (50), DNR (50), RN1 (60)</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td></td>
<td>DIMENSION DR(300), DJ(300), DPOX (300), DEDT (300)</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td></td>
<td>DIMENSION ECWORK (11)</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td></td>
<td>DIMENSION ECWKR (1)</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td>DIMENSION EMISC (1)</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td></td>
<td>DIMENSION ESMARK (1)</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td></td>
<td>DIMENSION ETA (300), OCTAOX (300)</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td></td>
<td>DIMENSION EWXR (1)</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td></td>
<td>DIMENSION FEDR (12)</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td></td>
<td>DIMENSION IBUFF (2000)</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td></td>
<td>DIMENSION IC(4, 3, 4), IC3 (4, 3, 4)</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td></td>
<td>DIMENSION ID(9), II(4)</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td></td>
<td>DIMENSION IP (4)</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td></td>
<td>DIMENSION I7 (300)</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td></td>
<td>DIMENSION ITIME (4)</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td></td>
<td>DIMENSION NE (3, 4)</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td></td>
<td>DIMENSION NSEX (4), CORX (8, 4), CORY (8, 4), CORZ (8, 4), NSP (3, 4)</td>
<td></td>
</tr>
</tbody>
</table>
```

**ORIGINAL PATTERN OF POOR QUALITY**

-6.3-
<table>
<thead>
<tr>
<th>LINE</th>
<th>DIMENSION</th>
<th>DESCRIPTION</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>57</td>
<td>W(4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>W(100), XLE(100), NNC(100)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>W(844)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>W(120)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>SYM(4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>TITLE(24)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>TRS(3, 4), ZSEC(3, 4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>MET(1300, 10), NETA(300)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>W(1300)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>66</td>
<td>W(3600)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>67</td>
<td>XCN(300)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>XYZ(1600)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>XZFD(300, 3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>KAK, IMPRINT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>W(300, 20), FREQ(12), HFP(12), FRT(12), NTMAX</td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>72</td>
<td>ITG, IGESS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>MKSTOR, WINGS, NSECIN, NMODE, NPSANS, NCONR, NFRQ, NPSN</td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>74</td>
<td>1, NMXFDZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>MAXIMUM N WINGS</td>
<td></td>
<td>40</td>
</tr>
<tr>
<td>76</td>
<td>MAXIMUM NSECTIONS</td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>77</td>
<td>MAXIMUM NMODES</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>78</td>
<td>MAXIMUM NPSANS</td>
<td></td>
<td>50</td>
</tr>
<tr>
<td>79</td>
<td>MAXIMUM NCONR PTS</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>80</td>
<td>MAXIMUM N FREQUENCIES</td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>81</td>
<td>MAXIMUM N ELEMENTS / SPAN</td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>82</td>
<td>NEXICS / IFUSH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>83</td>
<td>NPARM / NET</td>
<td></td>
<td></td>
</tr>
<tr>
<td>84</td>
<td>PERSP / PERC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>TAPE / 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>86</td>
<td>VTIME / ITIME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>87</td>
<td>VELCOM, NMAX, PRINT, NMROW(20)</td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>88</td>
<td>WAKE, WAKE, WAKE, WAKE, WAKE, WAKE</td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>89</td>
<td>WAKE / WAKE, WAKE, WAKE, WAKE, WAKE, WAKE</td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>90</td>
<td>ZFD(10), J5, J6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>ZFUNNY, NIERF, B1, B2, B3, NFRQ, NPSN, NPSN</td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>92</td>
<td>ROW(1), GARK(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>93</td>
<td>MSEC, ITIME(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>94</td>
<td>WARK(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>WARK(13)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>WARK(25)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>WORK(29)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>98</td>
<td>WORK(329)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>99</td>
<td>WORK(429)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>WORK(729)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>WORK(829)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>102</td>
<td>WORK(929)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>103</td>
<td>WORK(4529)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>104</td>
<td>DWORK(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>105</td>
<td>DWORK(101)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>106</td>
<td>DWORK(201)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>107</td>
<td>DWORK(501)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>108</td>
<td>DWORK(601)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>109</td>
<td>DWORK(751)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>DWORK(1001)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>DWORK(1011)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>112</td>
<td>DWORK(1012)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>113</td>
<td>DWORK(1013)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>114</td>
<td>DWORK(1014)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>115</td>
<td>DWORK(1015)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>116</td>
<td>DWORK(1016)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>117</td>
<td>DWORK(1017)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>118</td>
<td>DWORK(1018)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ORIGINAL PAGE 2 OF POOR QUALITY**
LINE 119 EQUIVALENCE ( SWORK(1), WORKM(1) )
LINE 120 EQUIVALENCE ( SWORK(121), SWORKC(1) )
LINE 121 EQUIVALENCE ( SWORK(1), DW(1) )
LINE 122 EQUIVALENCE ( SWORKC(1101), P(1) )
LINE 123 EQUIVALENCE ( SWORKC(171 ), PHIM(1) )
LINE 124 EQUIVALENCE ( SWORKC(471 ), DEN(1) )
LINE 125 EQUIVALENCE ( SWORKC(771), ONM(1) )
LINE 126 EQUIVALENCE ( SWORKC(1071), CXP(1) )
LINE 127 EQUIVALENCE ( SWORKC(1371), W(1) )
LINE 128 EQUIVALENCE ( SWORKC(671), ESWRK(1) )
LINE 129 EQUIVALENCE ( SWORK(1), OR(1) )
LINE 130 EQUIVALENCE ( SWORKC(301), OR(1) )
LINE 131 EQUIVALENCE ( SWORKC(601), IW(1) )
LINE 132 EQUIVALENCE ( SWORKC(901), ARGW(1) )
LINE 133 EQUIVALENCE ( SWORKC(1201), ESWRK(1) )
LINE 134 EQUIVALENCE ( TAPES Y(5), AR(1) )
LINE 135 EQUIVALENCE ( WORKM(5), CL(1) )
LINE 136 EQUIVALENCE ( WORKM(53), FD(1) )
LINE 137 EQUIVALENCE ( WORKM(57), ID(1) )
LINE 138 EQUIVALENCE ( WORKM(166), IC(1) )
LINE 139 EQUIVALENCE ( WORKM(78), IT(1) )
LINE 140 EQUIVALENCE ( WORKM(126), CORNL(1) )
LINE 141 EQUIVALENCE ( WORKM(158 ), CORN(1) )
LINE 142 EQUIVALENCE ( WORKM(190), CORM(1) )
LINE 143 EQUIVALENCE ( WORKM(222), ICN(1) )
LINE 144 EQUIVALENCE ( WORKM(366), ISECT(1) )
LINE 145 EQUIVALENCE ( WORKM(378), IMEX(1) )
LINE 146 EQUIVALENCE ( WORKM(401), XYZ(1) )
LINE 147 EQUIVALENCE ( wjorkm(2201), WTEROW(1) )
LINE 148 EQUIVALENCE ( WORKM(4601), WETA(1) )
LINE 149 EQUIVALENCE ( WWORKM(7601), WETA(1) )
LINE 150 EQUIVALENCE ( WWORKM(110601), ESWRK(1) )
LINE 151 EQUIVALENCE ( WWORKM(130601), ESJWRK(1) )
LINE 152 EQUIVALENCE ( WWORKM(131091), WTEROW(1) )
LINE 153 EQUIVALENCE ( WWORKM(131091), WETA(1) )
LINE 154 DATA I3/67/
LINE 155 DATA I5, I6, I11, 6/
LINE 156 DATA LMODE / 4HMODE /
LINE 157 DATA ITITLE / 4HTITLE /
LINE 158 DATA LMAX / 62/
LINE 159 DATA TWP(1) / 6.283185308/
LINE 160 DATA XP / 0.0/
LINE 161 C
LINE 162 J5 = 15
LINE 163 J6 = 16
LINE 164 PLOTR = .FALSE.
LINE 165 CONTINUE [ERR = 0]
LINE 166 CALL ECHC(KONTRL)
LINE 167 IF(KJNTRL .EQ. 1) GO TO 10
LINE 168 PLOTR = .FALSE.
LINE 170 IFLUSH = 0
LINE 171 NETSV = 0
LINE 172 CALL GETTIM (ITIME)
LINE 173 C
LINE 174 C
LINE 175 C
LINE 176 C
LINE 177 C
LINE 178 C
LINE 179 C
LINE 180 C

00111150
001111200
001111210
001111220
001111220
001111240
001111250
001111260
001111270
001111280
001111290
001111300
001111310
001111320
001111330
001111340
001111350
001111360
001111370
001111380
001111390
001111400
001111410
001111420
001111430
001111440
001111450
001111460
001111470
001111480
001111490
001111500
001111510
001111520
001111530
001111540
001111550
001111560
001111570
001111580
001111590
001111600
001111610
001111620
001111630
001111640
001111650
001111660
001111670
001111680
001111690
001111700
001111710
001111720
001111730
001111740
001111750
001111760
001111770
001111780
001111790
001111800
001111810
001111820
001111830
001111840
001111850
001111860
001111870
001111880
001111890
001111900
001111910
001111920
001111930
001111940
001111950
001111960
001111970
001111980
001111990
001112000
001112010
001112020
001112030
LINE 181  C  READ( 5, 1006 ) TITLE
LINE 182   6  CONTINUE
LINE 183   7  READ( 15, 1JC6, END = 10 ) LABEL, TITLE
LINE 184  CDC  READ( 15, 1006 ) LABEL, TITLE
LINE 185  CDC  IF( I3(I39) ) 10, 8
LINE 186  CDC  CONTINUE
LINE 187   IF( LABEL .NE. TITLE ) GO TO 6
LINE 188  WRITE( 1, 1006 ) TITLE
LINE 189  C  NHNG NO. WINGS BETWEEN LAND 4
LINE 190  C  LINE 191  CALL INPL( 15, 16, TITLE, ITRUN, MACH, REFLEN, XP, XPIN, PLTR, 1
LINE 192   1  LTE, NAR, BEETA, LAB, 1D, FD, ERROR )
LINE 193   2  IF( IFLUSH .NE. 0 ) GO TO 500
LINE 194  C  CALL MINGIV( 15, 16, NHNG, NSECT, AR, NSP, CORNX, CORNY, CURNZ, 1
LINE 195   1  SYM, ICL, CL, ERROK, BEEET, IPW, NFR, FREQ, 1
LINE 196   2  REFLEN, LAB, 1D, FD )
LINE 197  C  MAIN 158
LINE 198  C  IF(NHNGS .LT. NHNG ) GO TO 610
LINE 199  C  GENERATE AFRO GRID
LINE 200  C  VIEW TOTAL NUMBER OF ELEMENTS IN THE STRUCTURE
LINE 201  C  VIEW TOTAL NUMBER OF ELEMENTS IN THE WING
LINE 202  C  TOTAL NUMBER OF SPANS IN A WING
LINE 203  C  IF(IPLTR) PLOTR1 = TRUE.
LINE 204  CDC  REPLACES
LINE 205  CDC  IF(IPLTR) CALL PSEUDO
LINE 206  CDC  LOOP ON WINGS J
LINE 207  NSPT = 1
LINE 208  NET = 0
LINE 209  C  main 158
LINE 210  C  IF(INWINGS .LT.NNHNG ) GO TO 610
LINE 211  C  GENERATE AERO GRID
LINE 212  C  VIEW TOTAL NUMBER OF ELEMENTS IN THE STRUCTURE
LINE 213  C  VIEW TOTAL NUMBER OF ELEMENTS IN THE WING
LINE 214  C  TOTAL NUMBER OF SPANS IN A WING
LINE 215  C  IF(IPLTR) PLOTR1 = TRUE.
LINE 216  CALL 4ESH( CORNX(JCN, J), CORNY( JCN, J), CURNZ(JCN, J), XYZ(NINDEX), J)
LINE 217  CALL 4ESH( CORNX(JCN, J), CORNY( JCN, J), CURNZ(JCN, J), XYZ(NINDEX), J)
LINE 218  CALL 4ESH( CORNX(JCN, J), CORNY( JCN, J), CURNZ(JCN, J), XYZ(NINDEX), J)
LINE 219  CALL 4ESH( CORNX(JCN, J), CORNY( JCN, J), CURNZ(JCN, J), XYZ(NINDEX), J)
LINE 220  CALL 4ESH( CORNX(JCN, J), CORNY( JCN, J), CURNZ(JCN, J), XYZ(NINDEX), J)
LINE 221  CALL 4ESH( CORNX(JCN, J), CORNY( JCN, J), CURNZ(JCN, J), XYZ(NINDEX), J)
LINE 222  IF(NSECTN .LT. NSPT) GO TO 620
LINE 223  LOOP ON SECTIONS OF EACH WING I
LINE 224  DO 19 1 = 1, NSPT
LINE 225  DO 19 1 = 1, NSPT
LINE 226  DO 19 1 = 1, NSPT
LINE 227  DO 19 1 = 1, NSPT
LINE 228  DO 19 1 = 1, NSPT
LINE 229  BRANCH TO CORRECT MESH GENERATING ROUTINE
LINE 230  IF( ICL(I, J) ) 15, 16, 17
LINE 231  CALL GRIDIV( CORNX(JCN, J), CORNY(JCN, J), CURNZ(JCN, J), XYZINDEX, J)
LINE 232  1  ZSECT(I, J), TRSL(I, J), NACH(NSPT), SW(NSPT), XLEN(NSPT)
LINE 233  2  ZSP(I, J), NEM(I, J), NETSY, REFLEN, NFRQ, FREQ
LINE 234  3  NW, NSECT, ICL, ERROK, BEEET, LAB, 1D, FD, 15, 16
LINE 235  GO TO 18
LINE 236  CALL MESH1( CORNX(JCN, J), CORNY(JCN, J), CURNZ(JCN, J), XYZINDEX, J)
LINE 237  1  ZSECT(I, J), TRSL(I, J), NACH(NSPT), SW(NSPT), XLEN(NSPT)
LINE 238  2  ZSP(I, J), NEM(I, J), NETSY, EARK
LINE 239  GO TO 18
LINE 240  CALL MESH1( CORNX(JCN, J), CORNY(JCN, J), CURNZ(JCN, J), XYZINDEX, J)
LINE 241  1  ZSECT(I, J), TRSL(I, J), NACH(NSPT), SW(NSPT), XLEN(NSPT)
LINE 243   2  NSP(I,J), NE(I,J), NETSX, ICLIJ,J, CL(I,J), EARS)
LINE 244   18 CONTINUE
LINE 245   C
LINE 246   VNSP = NSPV & NSP(I,J)
LINE 247   NSPT = NSPV & NSP(I,J)
LINE 248   NEW = NE(I,J)
LINE 249   C END LOOP ON SECTIONS OF WING
LINE 250   19 CONTINUE
LINE 251   C
LINE 252   VINDEX = 6#NET & 1
LINE 253   C JCN = NUMBER OF COUNTER POINTS TO DEFINE THE WING
LINE 254   JCN = 2*(NSEC(J)-1) & 4
LINE 255   C IF(IPLOTR) CALL PLOT(D(NE(I,J), XYZ INDEX), NSCH(I,J), NSP(I,J),
LINE 256   C)
LINE 257   IF(IPLOTR) CALL PLOT(D(NE(I,J), XYZ INDEX), NSCH(I,J), NSP(I,J),
LINE 258   C)
LINE 259   IF(IPLOTR) CALL PLOT(D(NE(I,J), XYZ INDEX), NSCH(I,J), NSP(I,J),
LINE 260   C)
LINE 261   C END LOOP ON SECTIONS OF WING
LINE 262   20 CONTINUE
LINE 263   IF(IPLOTR) CALL ENDPLT
LINE 264   C
LINE 265   NSPT = NSPV - 1
LINE 266   IF(IPLOTR) CALL ENDPLT
LINE 267   C WRITE( 6, 1014 ) ( NWCH(I), I=1,NSPT )
LINE 268   C WRITE( 6, 1015 ) ( XLE(I), I=1,NSPT )
LINE 269   C WRITE( 6, 1016 ) ( XLE(I), I=1,NSPT )
LINE 270   C WRITE( 6, 1017 ) ( XLE(I), I=1,NSPT )
LINE 271   C WRITE( 6, 1018 ) ( XLE(I), I=1,NSPT )
LINE 272   C WRITE( 6, 1019 ) ( XLE(I), I=1,NSPT )
LINE 273   C WRITE( 6, 1020 ) ( XLE(I), I=1,NSPT )
LINE 274   C WRITE( 6, 1021 ) ( XLE(I), I=1,NSPT )
LINE 275   C WRITE( 6, 1022 ) ( XLE(I), I=1,NSPT )
LINE 276   C WRITE( 6, 1023 ) ( XLE(I), I=1,NSPT )
LINE 277   C WRITE( 6, 1024 ) ( XLE(I), I=1,NSPT )
LINE 278   C WRITE( 6, 1025 ) ( XLE(I), I=1,NSPT )
LINE 279   C WRITE( 6, 1026 ) ( XLE(I), I=1,NSPT )
LINE 280   C WRITE( 6, 1027 ) ( XLE(I), I=1,NSPT )
LINE 281   C WRITE( 6, 1028 ) ( XLE(I), I=1,NSPT )
LINE 282   C WRITE( 6, 1029 ) ( XLE(I), I=1,NSPT )
LINE 283   C WRITE( 6, 1030 ) ( XLE(I), I=1,NSPT )
LINE 284   C WRITE( 6, 1031 ) ( XLE(I), I=1,NSPT )
LINE 285   C WRITE( 6, 1032 ) ( XLE(I), I=1,NSPT )
LINE 286   C WRITE( 6, 1033 ) ( XLE(I), I=1,NSPT )
LINE 287   C WRITE( 6, 1034 ) ( XLE(I), I=1,NSPT )
LINE 288   C WRITE( 6, 1035 ) ( XLE(I), I=1,NSPT )
LINE 289   C WRITE( 6, 1036 ) ( XLE(I), I=1,NSPT )
LINE 290   C WRITE( 6, 1037 ) ( XLE(I), I=1,NSPT )
LINE 291   C WRITE( 6, 1038 ) ( XLE(I), I=1,NSPT )
LINE 292   C WRITE( 6, 1039 ) ( XLE(I), I=1,NSPT )
LINE 293   C WRITE( 6, 1040 ) ( XLE(I), I=1,NSPT )
LINE 294   C WRITE( 6, 1041 ) ( XLE(I), I=1,NSPT )
LINE 295   C WRITE( 6, 1042 ) ( XLE(I), I=1,NSPT )
LINE 296   C WRITE( 6, 1043 ) ( XLE(I), I=1,NSPT )
LINE 297   C WRITE( 6, 1044 ) ( XLE(I), I=1,NSPT )
LINE 298   C WRITE( 6, 1045 ) ( XLE(I), I=1,NSPT )
LINE 299   C WRITE( 6, 1046 ) ( XLE(I), I=1,NSPT )
LINE 300   C WRITE( 6, 1047 ) ( XLE(I), I=1,NSPT )
LINE 301   C WRITE( 6, 1048 ) ( XLE(I), I=1,NSPT )
LINE 302   C WRITE( 6, 1049 ) ( XLE(I), I=1,NSPT )
LINE 303   C WRITE( 6, 1050 ) ( XLE(I), I=1,NSPT )
LINE 304   C WRITE( 6, 1051 ) ( XLE(I), I=1,NSPT )

---

6.1
LINE 305 WRITE((9, 4001))
LINE 306 4001 FORMAT(16I1, 5X, 7H SECTION, 30X, 16HPJ SECUNS USF, 5X, 9MLEAL CPU, //1, 10I5)
LINE 307 C
LINE 308 CALL TIMOUT(MSEC, 4HREAD, 4H I N D, 4H AHD, 4H AHD, 4HLNER, 4HATE, 4HGRID, 013J3080)
LINE 309 1 4H C
LINE 310 C LINE 311 CALL PGEN
LINE 312 L AEC = NET*NFRx
LINE 313 C LINE 314 CALL
LINE 315 C LINE 316 C IF THERE IS ONLY ONE FREQUENCY, AND IT = 0.0, SET INT= J
LINE 317 CALL WRITE(1, 4H) WRITE = 1
LINE 318 IF( IFREQ .EQ. 1 .AND. FREQU(1) .EQ. 0.0 ) WRITE = 0
LINE 319 IF( IFREQ .NE. J ) WRITE = NSPET*FREQ
LINE 320 IF( INTM .EQ. 1 ) INTM = WRITE
LINE 321 IF( IFREQ .EQ. 1 ) INTM = WRITE
LINE 322 IF( ERROR ) GO TO 600
LINE 323 IF( IFREQ .EQ. 1 ) WRITE = NWPET
LINE 324 C WRITE(6, 3002) INTM, INTM
LINE 325 C
LINE 326 CALL LUOPN( NET, NWPET, XYZ, NSP, SYM, TRS, NNCN, NE, 013J3260)
LINE 327 1 NSEC, 3ZEC, 180NS, 16, TAPET, XECN, ARE, WRT, LREEJ
LINE 328 2 WRT, WRT, WRT, WRT, WRT, WRT, WRT, WRT
LINE 329 C LINE 330 IF( IFREQ .EQ. 1 ) WRITE(16, 1023) NTMAX, NCVFL, RKE, FREEST
LINE 331 C WRITE(16, 1022) NTMAX, RKE
LINE 332 GO TO 110
LINE 333 C LINE 334 C IF( IFREQ .EQ. 1 ) WRITE(16, 1021) NTMAX, NCVEI, RKEX, FREEST
LINE 335 C IF( IFREQ .EQ. 1 ) WRITE(16, 1022) NTMAX, RKEX
LINE 336 GO TO 110
LINE 337 C LINE 338 60 CONTINUE
LINE 339 IF( IFREQ .EQ. 1 ) WRITE(16, 1021) NTMAX, NCVFL, RKE, FREEST
LINE 340 CALL ROUTINE TO READ MODES IN UNNORMALIZED STRUCTURAL REF. SYST.
LINE 341 IF( IFREQ .EQ. 1 ) WRITE(16, 1022) NTMAX, RKEX
LINE 342 GO TO 110
LINE 343 C LINE 344, 110 CONTINUE
LINE 345 C LINE 346 C
LINE 347 C READ MODE INPUT
LINE 348 C MODE CARD HAS BEEN READ
LINE 349 C CALL ROUTINE TO READ MODES IN UNNORMALIZED STRUCTURAL REF. SYST.
LINE 350 C WRITE(16, 2006) TITLE
LINE 351 C IF( LAB .NE. LMODE ) GO TO 132
LINE 352 C WRITE(16, 2006) TITLE
LINE 353 C LMODE ) WRITE((16, 2006) TITLE
LINE 354 C IF( LAB .NE. LMODE ) GO TO 132
LINE 355 C CALL LMODE(NMODE, NWIND, NSEC, NE, XECN, XYZ, BETA, REFLEJ
LINE 356 1 NET, LAB, ID, ERROR, NFAT, NWAT, 15, 15, XYZFOL(1, 1), XYZFOL(1, 2), 013J3560
LINE 357 2 XYZFOL(1, 1), 31, MXFOL, IERI
LINE 358 C LINE 359 C IF( IFREQ .EQ. 1 ) GO TO 680
LINE 360 C LINE 361 C IF( IFREQ .EQ. 1 ) GO TO 680
LINE 362 C IF( IFREQ .EQ. 1 ) GO TO 680
LINE 363 C IF( IFREQ .EQ. 1 ) GO TO 680
LINE 364 C IF( IFREQ .EQ. 1 ) GO TO 680
LINE 365 C IF( IFREQ .EQ. 1 ) GO TO 680
LINE 366 1 4H C
LINE 367 C IF( LTRUN .EQ. 3 ) GO TO 132
LINE 370 C IF( LTRUN .EQ. 1 ) GO TO 132.
LINE 371 C WRITE( 16, 3031 )
LINE 372 C IF( XLOTR ) WRITE( 16, 3032 )
LINE 373 C GO TO 4
LINE 374 132 CONTINUE
LINE 375 IF( NOT. ERROR ) GO TO 135
LINE 376 WRITE( 16, 3033 )
LINE 377 GO TO 4
LINE 378 135 CONTINUE
LINE 379 CALL WETA( NET,JMODE,DETA, ETA, 113).
LINE 380 CALL WETA( NET,JMODE,DETA, ETA, 114)
LINE 381 C LOOP ON FREQUENCIES
LINE 382 C SET LINE .EQ. LMAX & 1 TO FORCE NEW PAGE WITH FREQ. HEADING
LINE 383 C
LINE 384 C
LINE 385 C LOOP ON FREQUENCIES
LINE 386 C DO 300 IF=1,NFREQ
LINE 387 C SET LINE .EQ. LMAX & 1 TO FORCE NEW PAGE WITH FREQ. HEADING
LINE 388 C
LINE 389 C
LINE 390 C
LINE 391 C
LINE 392 C
LINE 393 C
LINE 394 C
LINE 395 C
LINE 396 C
LINE 397 C
LINE 398 C
LINE 399 C
LINE 400 C CALL DIAG( WR, IF, LRECL, NNCH, MC, D, NET, ARW, IZ, IP, NSPT, FRQ, C)
LINE 401 C CALL GETTIM( ITIME )
LINE 402 C WRITE(J9,3025)
LINE 403 C CALL TIMOUT(MSEC,4HREAD,4H AND,4H DEC,4HOMPO,4HOMPO,4H. M, C)
LINE 404 C CALL TIMOUT(MSEC,4HREAD,4H AND,4H DEC,4HOMPO,4HOMPO,4H. M, C)
LINE 405 C 4HTIX )
LINE 406 C FIND POSITIVE EXPONENTIAL
LINE 407 C DO 168 I=1,NFREQ
LINE 408 C
LINE 409 C
LINE 410 C
LINE 411 C
LINE 412 C
LINE 413 C
LINE 414 C
LINE 415 C
LINE 416 C
LINE 417 C
LINE 418 C
LINE 419 C
LINE 420 C
LINE 421 C
LINE 422 C
LINE 423 C
LINE 424 C
LINE 425 C
LINE 426 C
LINE 427 C
LINE 428 C

LOOP ON FREQUENCIES

ZERO OUT GENERALIZED COEFFICIENTS (Q).

LOOP ON PRESSURE MODES

SET LINE=LMAX TO FORCE NEW PAGE FOR JM .GT. 1

IF( JM .GT. 1 .AND. LINE .NE. (LMAX+1) ) LINE=LMAX

IT=0

IF( FRJ .EQ. 0.01 ) JCNVSW = NET

CALL 4DETA( 113,DETMX*ETA,NET)

DO 197 I=1,NET

-6.9-
REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR
CALL PHIL(XLE(NSPI),XCEN(INDXR),LUXI(INDXR),NEPS,PHI,CXKM).
LINE 452 1 CFX(I,JXJ),PHI(W(INDXR))
LINE 453 IF(IT1,EQ.0) GO TO 182
LINE 454 IF(ITE.EQ.0) GO TO 182
LINE 455 IF(VIT,WAKE) GO TO 182
LINE 456 IF(VIT.WAKEN1) GO TO 182
LINE 457 CONTINUE
LINE 458 WRITE(16,3002) J,ISP
LINE 459 DLS = (0.0,0.0)
LINE 460 PMS = (0.0,0.0)
LINE 501 CLOOP ON NUMBER OF ELEMENTS IN SPAN ISP
LINE 502 DO 210 IE=1,NEPS
LINE 503 PRES(INDXR) = (DPDX(INDXR))*CFREQ*(E)*TWOPI
LINE 504 DLS(INDXR) = PRES(INDXR)*AREA(INDXR)
LINE 505 WRITE(16,3004) INDEXR, PHI(IE), PRES(INDXR)
LINE 506 CONTINUE
LINE 507 TL(J,JM) = TL(J,JM) + PMS
LINE 508 TL(J,JM) = TL(J,JM) + PMS
LINE 509 CEND LOOP ON ELEMENT 15 OF SPAN ISP
LINE 510 CEND LOOP ON SPAN ISP OF WING J
LINE 511 CONTINUE
LINE 512 CEND LOOP ON DISPLACEMENT MODE TO CALCULATE Q
LINE 513 DO 270 IM=IP(JM)
LINE 514 WRITE(16,3005) TL(J,JM), TM(J,JM)
LINE 515 CREAD ETA FROM 114
LINE 516 CALL RDETA(XETA,NE,T,E)
LINE 517 CLOOP OVER ALL ELEMENTS BY WINGS
LINE 518 CEND LOOP ON WINGS J
LINE 519 CEND LOOP ON WINGS J
LINE 520 CONTINUE
LINE 521 WRITE(16,3005) TL(J,JM), TM(J,JM)
LINE 522 CREAD ETA FROM 114
LINE 523 IF(ETA(JM) .EQ. 0) GO TO 240
LINE 524 IF(ITE.EQ.0) GO TO 230
LINE 525 IF(VIT.WAKE) GO TO 230
LINE 526 IF(VIT.WAKEN1) GO TO 230
LINE 527 WRITE(16,3004) IM
LINE 528 CONTINUE
LINE 529 CALL JUTPI(MACH,FRQ,JM,J,NSECT(J),NSP,L,J,SW,DLSPAN)
LINE 530 CALL PPMSPN(NSPI), XPIN,INDXI,PRES,IPW,L3,LINE,LMAX,NNCH,NE(I,J)
LINE 531 CEND LOOP ON DISPLACEMENT MODE TO CALCULATE Q
LINE 532 CEND LOOP ON DISPLACEMENT MODE TO CALCULATE Q
LINE 533 CEND LOOP ON DISPLACEMENT MODE TO CALCULATE Q
LINE 534 CONTINUE
LINE 535 IF(ITE.EQ.0) GO TO 245
LINE 536 IF(ITE.EQ.0) GO TO 245
LINE 537 IF(VIT.WAKE) GO TO 245
LINE 538 IF(VIT.WAKEN1) GO TO 245
LINE 539 REWIN(112)
LINE 540 IM = IM+1
LINE 541 CALL WTEPHT(DEUT,NET,PHI,112,112,WTE)
LINE 542 GO TO 169
LINE 543 CONTINUE
LINE 544 CONTINUE
LINE 545 CONTINUE
LINE 546 CLOOP ON DISPLACEMENT MODE TO CALCULATE Q
LINE 547 CLOOP ON DISPLACEMENT MODE TO CALCULATE Q
LINE 548 REWIN(112)
LINE 549 DO 270 IM=1,JMODE
LINE 550 CREAD ETA FROM 114
LINE 551 CALL ADELA(I14,DETDX,ETA,NET)
LINE 552 CLOOP OVER ALL ELEMENTS BY WINGS
LINE 533  INDEX = 0
LINE 534  DO 200 J=1,NWING.
LINE 535  SYMCC = 2.0
LINE 536  C SYMCC 2.0 SYMCCN DEFINED 1.5
LINE 537  IF (SYMCC .EQ. 0.0) SYMCC = 1.5
LINE 538  NS = NSECT(JJ)
LINE 539  NEW = 0
LINE 540  DO 255 I=1,NS
LINE 541  255 NEW = NEW .NE(I,JJ)
LINE 542  NEW = 2.0 NSYMC(JJ) = 0
LINE 543  INDEX = INDEX + 1
LINE 544  DLGA = PKNR(NDXR)*AREA(NDXR) + 0.0
LINE 545  Q(JM,JM) = Q(JM,JM) & ETA(NDXR) 0.040
LINE 546  250 CONTINUE
LINE 547  260 CONTINUE
LINE 548  270 CONTINUE
LINE 549  C END LOOP ON MODE JM
LINE 550  C WRITE (16, 3010) NWING
LINE 551  C WRITE (16, 3011) JMODE
LINE 552  DO 320 I=1,JMODE
LINE 553  WRITE (16, 3006) I, (Q(I,J), J=1,JMODE)
LINE 554  END LOOP ON FREQUENCY IF
LINE 555  C LINE = LMAX
LINE 556  CALL OUTPA(I3, MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE)
LINE 557  CALL GETTIM (TIME)
LINE 558  CALL TIMJUT(MSEC, 4HSSOLV, 4HSDR, 4HRDG, 4HRDO, 4HR3, 4H5, 4H6, 4H7)
LINE 559  CALL OUTP2(I3, NWING, XPIN, TL, TM, JMODE, FREQ, MACH, LINE, LMAX)
LINE 560  GO TO 500
LINE 561  1 4HREQ.
LINE 562  300 CONTINUE
LINE 563  CONTINUE
LINE 564  WRITE (16, 3026)
LINE 565  IF (IRUN .EQ. 3) GO TO 10
LINE 566  WRITE (16, 3030)
LINE 567  C
LINE 568  C
LINE 569  C
LINE 570  C
LINE 571  280 CONTINUE
LINE 572  C
LINE 573  C
LINE 574  C
LINE 575  C
LINE 576  C
LINE 577  C
LINE 578  CALL OUTPA(I3, MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE)
LINE 579  CALL GETTIM (TIME)
LINE 580  CALL TIMJUT(MSEC, 4HSSOLV, 4HSDR, 4HRDG, 4HRDO, 4HR3, 4H5, 4H6, 4H7)
LINE 581  CALL OUTP2(I3, NWING, XPIN, TL, TM, JMODE, FREQ, MACH, LINE, LMAX)
LINE 582  GO TO 500
LINE 583  1 4HREQ.
LINE 584  300 CONTINUE
LINE 585  CONTINUE
LINE 586  WRITE (16, 3026)
LINE 587  IF (IRUN .EQ. 3) GO TO 10
LINE 588  WRITE (16, 3030)
LINE 589  C
LINE 590  C
LINE 591  C
LINE 592  C
LINE 593  CALL MXERR (IERR, NWING, NWING, 16)
LINE 594  GO TO 500
LINE 595  610 IERR = 610
LINE 596  CALL MXERR (IERR, NWING, NWING, 16)
LINE 597  GO TO 500
LINE 598  620 IERR = 620
LINE 599  CALL MXERR (IERR, NWING, NWING, 16)
LINE 600  GO TO 500
LINE 601  630 IERR = 630
LINE 602  CALL MXERR (IERR, NWING, NWING, 16)
LINE 603  GO TO 500
LINE 604  640 IERR = 640
LINE 605  CALL MXERR (IERR, NWING, NWING, 16)
LINE 606  GO TO 500
LINE 607  650 IERR = 650
LINE 608  CALL MXERR (IERR, NWING, NWING, 16)
LINE 609  GO TO 500
LINE 610  660 IERR = 660
LINE 611  CALL MXERR (IERR, NWING, NWING, 16)
LINE 612  GO TO 500
LINE 613  670 IERR = 670
LINE 614  CALL MXERR (IERR, NWING, NWING, 16)
LINE 615 GO TO 500
LINE 616 DO 500
LINE 617 C 10 CONTINUE
LINE 618 10 IF (PLTTR) CALL PLOT(1UOFF, WNO)
LINE 619 IF (PLTTR) CALL EPLOT(1HEND)
LINE 620 WRITE(16, 3328)
LINE 621 3028 FORMAT(1H0/// 3X, 10(1H*), 5X, 10(1H*)
LINE 622 STOP
LINE 623 C
LINE 624 1026 FORMAT(A4, 2X, 14A4, 10A1)
LINE 625 2006 FORMAT(1H1, 20X, 14A4, 10A1//)
LINE 626 3025 FORMAT(1H0)
LINE 627 3026 FORMAT(141)
LINE 628 3030 FORMAT(1H1// 5X, 10(1H*), 5X, 20HEND OF EXECUTION RUN, 5X, 10(1H*)
LINE 629 2 10(1H*)
LINE 630 3040 FORMAT(1H0/// 8X, 33HWAKE EFFECTS VELOCITY POTENTIALS / 8X
LINE 631 1 17HLTERATION NUMBER:
LINE 632 C3004 FORMAT(/ 9H LIFT/DS = 1PZE12.4• 5X, 10H MOMENT/DS = 1PZE12.4 /
LINE 633 C3002 FORMAT(215)
LINE 634 C3003 FORMAT(/ 3(6X, 1PZE12.4 )
LINE 635 C3005 FORMAT(3E19.0)
LINE 636 C3006 FORMAT(/ 20X, 21HNUMBER OF FREQUENCIES //)
LINE 637 C3007 FORMAT(/ 1X, 12(1H*), 5X, 10(1H*)
LINE 638 C3008 FORMAT(/// 1X, 12(1H*)
LINE 639 C3009 FORMAT(/ 1X, 12(1H*) // 1X, 12(1H*)
LINE 640 C3010 FORMAT(/// 1X, 12(1H*)
LINE 641 C3011 FORMAT(/ 1X, 12(1H*) // 1X, 12(1H*)
LINE 642 C3012 FORMAT(/// 1X, 12(1H*)
LINE 643 C3013 FORMAT(/ 1X, 12(1H*) // 1X, 12(1H*)
LINE 644 C3014 FORMAT(/// 1X, 12(1H*)
LINE 645 C3015 FORMAT(/// 1X, 12(1H*)
LINE 646 C3016 FORMAT(/ 1X, 12(1H*)
LINE 647 C3017 FORMAT(/ 1X, 12(1H*)
LINE 648 C3018 FORMAT(/ 1X, 12(1H*)
LINE 649 C3019 FORMAT(/ 1X, 12(1H*)
LINE 650 C3020 FORMAT(/ 1X, 12(1H*)
LINE 651 C3021 FORMAT(/ 1X, 12(1H*) // 5X, 9HMORE THAN, 13, 37 TERMS ARE NEEDED)
LINE 652 C 1ED FOR CONVERGENCE FOR, 15, 19 PAIRS OF ELEMENTS. // 5X, 10(1H*)
LINE 653 C 2 27H THE MAXIMUM RADIUS*KAPPA =, E15.4, 14H, WITH KAPPA =, E15.6///
LINE 654 C 1022 FORMAT(/// 5X, 5H THE GREATEST NUMBER OF TERMS NEEDED FOR CONVERGENCE)
LINE 655 C 1ENCE IS, 13 //5X, 26H THE MAXIMUM RADIUS*KAPPA =, E15.4 ///)
LINE 656 C2007 FORMAT(/// 1H1MAC NUMBER, F22.4 //)
LINE 657 C2008 FORMAT(/// 20X, 21HNUMBER OF FREQUENCIES, //)
LINE 658 C2009 FORMAT(/// 1X, 20X, 19H LIST OF FREQUENCIES, 5F14.4 // 139X, 5F14.4 )
LINE 659 C2010 FORMAT(/// 20X, 21HNUMBER, F22.4 //)
LINE 660 C2011 FORMAT(/// 20X, 21HNUMBER, F22.4 //)
LINE 661 C2012 FORMAT(/// 18H RECEIVING ELEMENT, 12, 5X, 9HNUMBER, 14, 5X
LINE 662 C 1 10DF SECTION, 12, 5X, 7HOF WING // (1X, 10(1H*)
LINE 663 C 1020 FORMAT(/// 5X, 241)
LINE 664 C 1021 FORMAT(/// 1X, 14H OF WING, 12, 8H SECTION, 12 )
LINE 665 C3034 FORMAT(/ 19H MODAL DATA FOR WING, 12, 8H SECTION, 12 )
LINE 666 END
DISPLAY... FILE

DECK 02

LINE 1 BLOCK DATA 00200010
LINE 2 INTEGER PRINT 00200020
LINE 3 COMM/N/CEPS/ EPS 00200030
LINE 4 COMM/J /ITERAT/ I MAX 00200040
LINE 5 CJHMJ /NAXE/ NAXE 00200050
LINE 6 COMM/J/MXSTR/ MXINGS/NSEC TN/NAXE/ NSPANS/NCCNKR/NFKOJN/NEPSPN 00200060
LINE 7 LAN MFDZ 00200070
LINE 8 COMM/J /PEREPS/ PERC 00200080
LINE 9 COMM/J/TAPE/ 19,110,111,112,113,114,19,115 00200090
LINE 10 COMM/J/VELCOM /NMAX,PRINT, NWBLOK, NWROW(20) 00020100
LINE 11 C NWINGS= 4 00200110
LINE 12 C NSEC TN= 3 00200120
LINE 13 C NMODES= 10 00200130
LINE 14 C NSPANS= 100 00200140
LINE 15 C NCORN= 8 00200150
LINE 16 C NFKOJN= 12 00200160
LINE 17 C NEPSPN= 70 00200170
LINE 18 DATA /EPS/0.01/ 00200180
LINE 19 DATA ITMAX /10/ 00200190
LINE 20 DATA 19,110,111 /8,10,11/ 00200200
LINE 21 DATA 112/12/ 00200210
LINE 22 DATA 113/13/ 00200220
LINE 23 DATA 114/14/ 00200230
LINE 24 DATA 115/15/ 00200240
LINE 25 DATA J6/9/ 00200250
LINE 26 DATA NMAX/60/ 00200260
LINE 27 DATA NWINGS,NSEC TN,NMODES,NSPANS,NCORNK,NFKOJN,NEPSPN 00200270
LINE 28 1 NMAXFDZ 00200280
LINE 29 2 /4,3,10,100,8,12,70,300/ 00200290
LINE 30 DATA PERC /3.005/ 00200300
LINE 31 DATA PRINT /2/ 00200310
LINE 32 END 00200320

-6.14-
LINE 1  FUNCTION CABS1(A)
LINE 2  COMPLEX A
LINE 3   AR = A
LINE 4   AI = AIMAG(A)
LINE 5   CABS1 = AR**2 & AI**2
LINE 6   CABS1 = SQRT(CABS1)
LINE 7   RETURN
LINE 8   END

-6.15-
SUBROUTINE DECOM(N,NDIM,A,IP)  
SUBROUTINE DECOM  

PURPOSE  
FACTORIZATION OF THE MATRIX A INTO A PRODUCT OF A LOWER  
TRIANGULAR MATRIX L AND AN UPPER TRIANGULAR MATRIX U. L HAS  
A UNIT DIAGONAL WHICH IS NOT STORED.  

USAGE  
CALL DECOM(N,NDIM,A,IP)  

DESCRIPTION OF PARAMETERS  
N - ORDER OF THE MATRIX A  
NDIM - FIRST DIMENSION OF A DECLARED IN CALLING PROGRAM. IF A IS  
SINGLY SUBSCRIBED IN CALLING PROGRAM, SET NDIM=N.  
A - ON INPUT THE MATRIX TO BE FACTORED.  
ON OUTPUT A(I,J) I.LE.J CONTAINS THE UPPER TRIANGLE U,  
A(I,J) I.GT.J CONTAINS (I-L), WHERE I IS THE  
IDENTITY MATRIX AND L IS THE LOWER TRIANGLE.  

IP - IP(K) K.LT.N CONTAINS THE ROW INTERCHANGE INFORMATION.  
IP(N) CONTAINS (-1)**(NUMBER OF INTERCHANGES) OR 0.  

REMARKS  
(1) IF IP(N)=0 THEN MATRIX A IS SINGULAR.  
(2) USE DECOM IN CONJUNCTION WITH SUBROUTINE 'SOLVE' TO OBTAIN  
THE SOLUTION OF THE LINEAR SYSTEM A*X = B.  
(3) DETERMINANT(A) = IP(N)*A(1,1)*A(2,2)*...*A(N,N).  
(4) THE ROW INTERCHANGE INFORMATION STORED IN IP IS NOT EASY  
TO INTERPRET. HOWEVER, IT IS USED PROPERLY BY 'SOLVE'.  

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED.  

METHOD  
MATRIX TRIANGULARIZATION BY GAUSSIAN ELIMINATION. SEE  
ALGORITHM 423, 'COLLECTED ALGORITHMS FROM CACM', BY CLEVE MOLER  

DIMENSION A(NDIM,1),IP(1)  

DO 1 N = 1,N  
M = K  
IF (ABS(A(I,K))) .GT. ABS(A(M,K)) M = I  
1 CONTINUE  

DO 2 J = K+1,N  
IF (T .EQ. 0.) GO TO 5  
2 CONTINUE  

DO 4 J = K+1,N  
IF (IP(K)) M = K  
4 CONTINUE  

DO 5 I = 1,N  
IF (IP(I)) M = I  
5 CONTINUE  

DO 6 K = 1,N  
IF (IP(K)) M = K  
6 CONTINUE  

DO 8 N = 1,N  
IF (IP(N)) M = N  
8 CONTINUE  

DO 10 I = 1,N  
IF (IP(I)) M = I  
10 CONTINUE  

DO 12 I = 1,N  
IF (IP(I)) M = I  
12 CONTINUE  

DO 14 I = 1,N  
IF (IP(I)) M = I  
14 CONTINUE  

END
T = A(M,J)
A(M,J) = A(K,J)
A(K,J) = T
IF (T .EQ. 0.) GO TO 4
DO 3 I = KP1,N
IF (A(I,J) .EQ. 0.) GO TO 21
C = (A(I,J) & A(I,K) * T) / A(I,J)
C = A38(C)
IF (C .LE. .0001) A(I,J) = 0.0
IF (C .LE. .0001) GO TO 3
CONTINUE
A(I,J) = A(I,J) & A(I,K) * T
CONTINUE
IF (A(K,K) .EQ. 0.) IP(N) = 0
CONTINUE
RETURN
END
DISPLAY ... FILE

LINE 1 SUBROUTINE DIAG (AROW, IF, LRECL, NNCH, MC, D, NET, AROW(1), IP)
LINE 2 NAMEST, FRQ, CAROW, CHROW
LINE 3 COMPLEX CAROW(1), CHROW(1)
LINE 4 COMPLEX WES
LINE 5 INTEGER PRINT
LINE 6 LOGICAL TREDGE
LINE 7 LOGICAL WAKE, WAKE1, WAKENZ
LINE 8 DIMENSION AROW(NET), ZINET)
LINE 9 DIMENSION D(MC, MC), IP(MC)
LINE 10 DIMENSION NNCH(1)
LINE 11 DIMENSION WROW(LRECL)
LINE 12 COMMON /K1112/ KST11, KST12
LINE 13 COMMON /TAPE/ I9, I10, I11, I12, I13, I14, J9, I15
LINE 14 COMMON /VELCOM/ NMAX, PRINT, NMAX, LROW, LROW(20)
LINE 15 COMMON /WAKE/ ITM, ITM, ICNVT, EPSW, JCNYGW, LRWTE, KAWTE
LINE 16 COMMON /WAKE/wp, WTE(12), TREDGE, WAKE1, WAKE, WAKENZ, KAWTE
LINE 17 REWIND 111
LINE 18 REWIND 119
LINE 19 REWIND 110
LINE 20 IB = 19
LINE 21 IF (IWTE .EQ. 0) REWIND 112
LINE 22 IF (WAKE) REWIND 115
LINE 23 READ (19)
LINE 24 IF (WAKE) READ (115)
LINE 25 IBT = 0
LINE 26 I=0
LINE 27 K=0
LINE 28 WAKENZ = .FALSE.
LINE 29 DO 40 IB = 1, NROW
LINE 30 NROW = NROW + 1
LINE 31 DO 20 I = 1, NROW
LINE 32 K = K + 1
LINE 33 CALL FREQW(I, AROW, WROW, IF, LRECL)
LINE 34 DO 15 J = 1, NROW
LINE 35 M = J + IBT
LINE 36 IF (I J) = AROW(M)
LINE 37 AROW(M) = 0.0
LINE 38 CONTINUE
LINE 39 IF (NROW > NHT, NNCH)
LINE 40 CALL WITED(K, AROW, IZ, WRCK, NET)
LINE 41 IF (IWTE .EQ. 0) GO TO 20
LINE 42 IF (NOT. WAKE ) GO TO 20
LINE 43 IF ( K .GT. KAWTE ) GO TO 20
LINE 44 IF ( K .GT. KAWTE ) GO TO 20
LINE 45 IF ( FRQ .EQ. 0.0) GO TO 20
LINE 46 CALL AKET(112, CAROW, CHROW, IF, LRWTE, NSPT, TREDGE, WAKE1, WAKE, WAKENZ)
LINE 47 IF (I15, IZ, K1, NNCH, FRQ, K)
LINE 48 CONTINUE
LINE 49 IBT = IBT + NROW
LINE 50 CALL MAPRT( D, NROW, NROW, 'D MATRIX'
LINE 51 CALL DECOM(NROW, NMAX, D, IP)
LINE 52 WRITE (110) D, IP
LINE 53 CONTINUE
LINE 54 END_FILE 110
LINE 55 END_FILE 111
LINE 56 REWIND 110
LINE 57 REWIND 111
LINE 58 IF(1WTE .NE. 0) END FILE 112
LINE 59 IF(1WTE .NE. 0) REWIND 112
LINE 60 RETURN
LINE 61 END
CISPLAY...FILE  DECK 06

LINE 1  SUBROUTINE ECHOI KONTRL.
LINE 2  DIMENSION CARD(20)
LINE 3  COMMON /TAPE/ J5,113,111,112,113,114,J9,115
LINE 4  COMMON /FD010/ J5,J6
LINE 5  DATA CEND(4)HEND /
LINE 6  KONTRL = 0
LINE 7  REWIND 111
LINE 8  LINES = 0
LINE 9  WRITE(J6,500)
LINE 10  10 READ(J5,2000,END=100) CARD
LINE 11  CDC10 READ ( J5,2000) CARD
LINE 12  CDC IF I END(J5) 100,11
LINE 13  11 CONTINUE
LINE 14  LINES = LINES + 1
LINE 15  IF( LINES .LT. 47 ) GO TO 50
LINE 16  WRITE(J6,700)
LINE 17  WRITE(J6,500)
LINE 18  LINES = 0
LINE 19  50 WRITE(J11,2000) CARD
LINE 20  WRITE(J6,600) CARD
LINE 21  IF( CARD(1) .EQ. CEND) GO TO 90
LINE 22  GO TO 10
LINE 23  C END OF ONE CASE
LINE 24  90 WRITE(J6,700)
LINE 25  END FILE 111
LINE 26  REWIND 111
LINE 27  RETURN
LINE 28  C END OF JOB
LINE 29  100 KONTAL = 1
LINE 30  RETURN
LINE 31  500 FORMAT(1H1,41X, 46H* * * I SCON ANALYSIS INPUT DATA DECK * * * ,00600010
LINE 32  1 1H0,24X,1H ,9X,1H1,9X,1H2,9X,1H3,7X,1H4,9X,1H5,9X,1H6,9X,1H7,00600020
LINE 33  2 9X,1H8 /25X,01H 123456789012345678901234567890 /25X,9X,9X,1H3,9X,1H1,9X,1H2,9X,1H3,9X,1H7,00600030
LINE 34  3012345673501234567890123456789012345678901234567890 /25X,9X,9X,1H3,9X,1H1,9X,1H2,9X,1H3,9X,1H7,9X,9X,1H8,00600040
LINE 35  400 FORMAT(1H1,1H,25X,204).
LINE 36  700 FORMAT(1H7,81H 123456789012345678901234567890123456789012345678901234567890 /25X,9X,9X,1H3,9X,1H1,9X,1H2,9X,1H3,9X,1H7,9X,1H8,00600050
LINE 37  101234567890123456789012345678901234567890 /25X,1H ,9X,1H1,9X,1H2,9X,1H3,9X,1H7,9X,1H8,00600060
LINE 38  2H4,9X,1H5,9X,1H6,9X,1H7,9X,1H8,00600070
LINE 39  2000 FORMAT(20A4).
LINE 40  END
<table>
<thead>
<tr>
<th>LINE</th>
<th>Code</th>
<th>Description</th>
<th>Line Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>SUBROUTINE EFPLT(A)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>CALL CALPLT(C, 0, 0, 0.999)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>RETURN</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>
SUBROUTINE EONE

LOGICAL EREONE, EMIRRO

COMMON / EEW / XUP, YUP, XLW, YLW, EREONE, EMIRRO, RR, RL, RTR, RFL

COMMON / RRL / TVW(4), YRO, YLO, ZR, ZZR, ZL, ZZL

COMMON / W11 / XO, YO, ZD, ZDD, SYMK

LINE 8 C TEST IF ELEMENT AND MIRROR IMAGE OF ELEMENT ARE IN MACH CONE
LINE 9 C EREONE TRUE IF ELEMENT IS IN MACH CONE. IF EREONE=FALSE THEN EREONE=TRUE.
LINE 10 C EMIRRO TRUE IF SYMMETRY CODE .NE. 0 AND MIRROR IMAGE IS IN MACH CONE. IF EMIRRO=FALSE THEN EMIRRO=TRUE.
LINE 12 C XO, YO COORDINATES OF RECEIVING PT. (CENTER OF REC. ELE.)
LINE 13 C XLW, YLW 1ST PT. ON LINE OF INFLUENCING ELE.
LINE 14 C XUP, YUP 2ND PT. ON LINE OF INFLUENCING ELE.
LINE 15 C SYMK SYMMETRY CODE (0.0 FOR NO SYMMETRY)
LINE 16 C RR AVERAGE R OF RIGHT-HAND ELEMENT
LINE 17 C RL AVERAGE R OF LEFT-HAND ELEMENT
LINE 18 C NOTE THIS ROUTINE IS ALSO USED TO TEST TRAILING EDGE LINE OF ELEM.
LINE 19 C ELEMENT WHEN XL, YL, AND X2, Y2 ARE ON TRAILING EDGE AND SYM=0.
LINE 20 C
LINE 21 C
LINE 22 C EREONE = .FALSE.
LINE 23 C EMIRRO = .FALSE.
LINE 24 C RR = 0.0
LINE 25 C RL = 0.0
LINE 26 C ZETA1 = XO - XLW
LINE 27 C ZETA2 = XO - XUP
LINE 28 C IF (ZETA1 .LE. 0.0 .AND. ZETA2 .LE. 0.0) RETURN
LINE 29 C ETA1 = YRO - YLW
LINE 30 C ETA2 = YRO - YUP
LINE 31 C T1 = SQRT(ETA1*ETA1 + ZRZR) T2 = SQRT(ETA2*ETA2 + ZRZR)
LINE 32 C CONTINUE
LINE 33 10 CONTINUE
LINE 34 C R1 = ZETA1 - T1
LINE 35 C R2 = ZETA2 - T2
LINE 36 C IF (R1 .LE. 0.0 .AND. R2 .LE. 0.0) RETURN
LINE 37 C R1 = R1*ZETA1 + T1
LINE 38 C R2 = R2*ZETA2 + T2
LINE 39 C IF (EREONE) GO TO 90
LINE 40 C EREONE = .TRUE.
LINE 41 C RTR = R1*R2
LINE 42 C IF (R1 .LT. 0.0) R1 = 0.0
LINE 43 C IF (R2 .LT. 0.0) R2 = 0.0
LINE 44 C R1 = SQRT(R1)
LINE 45 C R2 = SQRT(R2)
LINE 46 C RR = 0.5*(R1 + R2)
LINE 47 C IF (SYMK.EQ. 0.0) RETURN
LINE 48 C ETA1 = YLO - YLW
LINE 49 C ETA2 = YLO - YUP
LINE 50 C T1 = SQRT(ETA1*ETA1 + ZZL)
LINE 51 C T2 = SQRT(ETA2*ETA2 + ZZL)
LINE 52 C GO TO 10
LINE 53 90 CONTINUE
LINE 54 C EMIRRO = .TRUE.
LINE 55 C RTL = R1*R2
LINE 56 C IF (R1 .LT. 0.0) R1 = 0.0

-6.22-
LINE 57  IF ( R2 .LT. 0.0 ) R2 = 0.0
LINE 58  R1 = SORT( R1 )
LINE 59  R2 = SORT( R2 )
LINE 60  RL = 0.5*( R1 + R2 )
LINE 61  RETURN
LINE 62  END
DISPLAY...FILE DECK 09

LINE  1 SUBROUTINE FGEN
LINE  2      REAL MACH   09000010
LINE  3 COMMON / BASIC/MACH, BEETA
LINE  4 COMMON / FOI / NFREQ, CF(12,30), FREQ12, FREQP12, FRTES12, NTMX 09000020
LINE  5 C       NTMX THE MAXIMUM NUMBER OF TERMS USED IN MINGR, MINT
LINE  6 C       XUNDER MACHINE DEPENDENT NUMBER, LARGEST NEGATIVE EXPONENT
LINE  7 C       NTMX = 30 09000030
LINE  8 CDC XUNDER =-292.0 09000040
LINE  9 CDC XUNDER =-77.5 09000050
LINE 10 C WRITE( 6, 1001 ) NFREQ, ( FREQ(I), I=1, NFREQ )
LINE 11 C FRTES12= -1.0E+65 09000060
LINE 12 DO 5 I=1, NFREQ 09000070
LINE 13 C FREQP(I) = FREQ(I)*MACH/BEETA 09000080
LINE 14 C WRITE( 6, 1001 ) NFREQ, ( FREQP(I), I=1, NFREQ ), FRTES 09000090
LINE 15 C NTMX = 0 09000100
LINE 16 C NTMX = 1 09000110
LINE 17 C NTMX = 1 09000120
LINE 18 DO 40 I = 1, NFREQ 09000130
LINE 19 C CF(I,1) = 1.0 09000140
LINE 20 XKK = FREQP(I)*FREQP(I) 09000150
LINE 21 IF( XKK .GT. 0.0 ) GO TO 15 09000160
LINE 22 DO 30 J = 2, NTMX 09000170
LINE 23 10 CF(I,J) = 0.0 09000180
LINE 24 GO TO 40 09000190
LINE 25 15 CONTINUE 09000200
LINE 26 DO 30 J = 2, NTMX 09000210
LINE 27 Y = J - 1 09000220
LINE 28 N2 = N2
LINE 29 FACTOR = XKK/FLOAT( N2*(N2-1) ) 09000230
LINE 30 A10 = ABS4 CF(I,N) 09000240
LINE 31 A10 = ALOG10( A10 ) C ALOG10( FACTOR ) 09000250
LINE 32 IF( A10 .GE. XUNDER ) GO TO 25 09000260
LINE 33 DO 20 J1 = J, NTMX 09000270
LINE 34 20 CF(I,J1) = 0.0 09000280
LINE 35 NTMXX = MAX0( NTMXX, N ) 09000290
LINE 36 GO TO 40 09000300
LINE 37 25 CONTINUE 09000310
LINE 38 C CF(I,J) = -CF(I,N)*FACTOR 09000320
LINE 39 30 CONTINUE 09000330
LINE 40 NTMXX = NTMXX 09000340
LINE 41 C 09000350
LINE 42 40 CONTINUE 09000360
LINE 43 C 09000370
LINE 44 C DO 50 I = 1, NFREQ 09000380
LINE 45 C WRITE( 6, 1001 ) I, ( CF(I,J), J=1, NTMX ) 09000390
LINE 46 C 50 CONTINUE 09000400
LINE 47 C 50 CONTINUE 09000410
LINE 48 RETURN 09000420
LINE 49 END 09000430
<table>
<thead>
<tr>
<th>LINE</th>
<th>CODE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUB</td>
<td>SUBROUTINE FREQ(I8,AROW,WROW,NFREQLRECL)</td>
</tr>
<tr>
<td>2</td>
<td>DIM</td>
<td>DIMENSION WROW(LRECL), AROW(NWING)</td>
</tr>
<tr>
<td>3</td>
<td>COM</td>
<td>COMMON /FQ1/ NFREQ</td>
</tr>
<tr>
<td>4</td>
<td>COM</td>
<td>COMMON/PARAM/ NWING</td>
</tr>
<tr>
<td>5</td>
<td>EQU</td>
<td>EQUIVALENCE (NET,NWING)</td>
</tr>
<tr>
<td>6</td>
<td>READ</td>
<td>READ(18) K1, WROW</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>K = NF</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>DO 100 J = 1, NWING</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>AROW(J) = WROW(K)</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>K = K + NFREQ</td>
</tr>
<tr>
<td>11</td>
<td>100</td>
<td>CONTINUE</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>RETURN</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>END</td>
</tr>
</tbody>
</table>

-6.25-
DISPLAY...FILE

LINE 1  SUBROUTINE GETTIM ( ITIME )
LINE 2  DIMENSION ITIME ( 8 )
LINE 3  CALL SECOND ( A )
LINE 4  ITIME ( 2 ) = A * 1000.0
LINE 5  RETURN
LINE 6  END

-6.26-
SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z, TRS, NNCH, SW, XLE, NSP, NE, NETSV, REFLEN, NFREQ, FREQ, NWS, XLE, ID, FD, IS, I6 )

PURPOSE  GENERATE MESH FOR 1 SECTION WHEN OPTIONAL DATA FOR A PARTICULAR SECTION IS PRESENT, "GRID" DATA

INPUT

I5  INPUT UNIT ( USUALLY CARD READER 5 )
I6  OUTPUT UNIT ( PRINTER 6 )
I7  ERROR LOGICAL THAT COMES IN FALSE, MAY BE SET TRUE
I8  NSP, XWING, NSECT, ICL, ERROR, ID, FD, ID, FREQ
I9  LAB, NE, NTSECT, ERROR, ETA, ID, FD, I5, I6

OUTPUT

CORNX, CORNY, CORNZ, X, Y, AND Z COORD. OF CORNER POINTS OF CORNER P1S 1, 2, 3, 4 IN GLOBAL SYSTEM

NSP  DESIRED NUMBER OF SPANS FOR SECTION (INPUT BY USER) IN GLOBAL SYSTEM
XYZ  4 X AND 2 Y LOCAL COORD. FOR EACH ELEMENT OF SECTION IN GLOBAL SYSTEM
TRS  2 BY 2 TRANS MATRIX TO GC FROM LOCAL TO GLOBAL
NNCH  ELEMENT NUMBER (w.r.t. TOTAL STRUCTURE) OF LAST ELEMENT IN SPAN
SW  SPAN WIDTH
XLE  AVERAGE X COORDINATE OF LEADING EDGE PER SPAN
NE  NUMBER OF ELEMENTS IN SECTION
ICL(I,J) CODE FOR SECTION I OF WING J
ICL(I,J) = -1 SPECIFIC GRID INFORMATION IS INPUT HERE ICL IS UPDATED IF ANOTHER "GRID" SECTION IS ENCOUNTERED AFTER READ PRESENT "GRID" SECTION
NETSV  COUNTER OF TOTAL NUMBER OF ELEMENTS IN STRUCTURE
LOGICAL ERROR
LOGICAL FREQUENCY

DATA LB, LE, LC, X, Y, Z / 4HEND, 4HCHOR /
DATA LG, LS, LM, LF / 4HGRID, 4HSWAN, 4HMUDE, 4HFREQ /
DATA LINE

LINE 1  SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z, TRS, NNCH, SW, XLE, NSP, NE, NETSV, REFLEN, NFREQ, FREQ, NWS, XLE, ID, FD, IS, I6 ) 01200010
LINE 2  1 NSP, NE, NETSV, REFLEN, NFREQ, FREQ, NWS, XLE, ID, FD, IS, I6 ) 01200020
LINE 3  1 NSP, NE, NETSV, REFLEN, NFREQ, FREQ, NWS, XLE, ID, FD, IS, I6 ) 01200030
LINE 4  C I5  INPUT UNIT ( USUALLY CARD READER 5 ) 01200040
LINE 5  C I6  OUTPUT UNIT ( PRINTER 6 ) 01200050
LINE 6  C I7  ERROR LOGICAL THAT COMES IN FALSE, MAY BE SET TRUE 01200060
LINE 7  C NSP, XWING, NSECT, ICL, ERROR, ID, FD, IS, I6 01200070
LINE 8  C LAB, NE, NTSECT, ERROR, ETA, ID, FD, I5, I6 01200080
LINE 9  C LAB, NE, NTSECT, ERROR, ETA, ID, FD, I5, I6 01200090
LINE 10 C I5  INPUT UNIT ( USUALLY CARD READER 5 ) 01200940
LINE 11 C I6  OUTPUT UNIT ( PRINTER 6 ) 01200950
LINE 12 C I7  ERROR LOGICAL THAT COMES IN FALSE, MAY BE SET TRUE 01200960
LINE 13 C NSP, XWING, NSECT, ICL, ERROR, ID, FD, IS, I6 01200970
LINE 14 C LAB, NE, NTSECT, ERROR, ETA, ID, FD, I5, I6 01200980
LINE 15 C LAB, NE, NTSECT, ERROR, ETA, ID, FD, I5, I6 01200990
LINE 16 C LAB, NE, NTSECT, ERROR, ETA, ID, FD, I5, I6 01200990
LINE 17 C LAB, NE, NTSECT, ERROR, ETA, ID, FD, I5, I6 01200990
LINE 18 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 19 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 20 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 21 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 22 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 23 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 24 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 25 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 26 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 27 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 28 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 29 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 30 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 31 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 32 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 33 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 34 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 35 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 36 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 37 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 38 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 39 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 40 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 41 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 42 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 43 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 44 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 45 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 46 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 47 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 48 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 49 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 50 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 51 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 52 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 53 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 54 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 55 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990
LINE 56 C SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYX, Y, Z ) 01200990

-6.27-
LINE 57  Y1 = Y(1) * REFLEN
LINE 58  C  INITIALIZE ELEMENT AND SPAN COUNTERS (NE AND ISP)
LINE 59  NE = 0
LINE 60  ISP = 0
LINE 61  C  WRITE( I6, 3000 ) JW, IS
LINE 62  C  10 CONTINUE
LINE 63  C  READ( IS, 1000 ) LAB, ID, FD
LINE 64  C  WRITE( I6, 1600 ) 10 CONTINUE
LINE 65  C  IF( LAB .EQ. LS ) GO TO 20
LINE 66  C  IF( LAB .EQ. LG ) GO TO 80
LINE 67  C  IF( LAB .EQ. LF ) GO TO 50
LINE 68  C  IF( LAB .EQ. LM ) GO TO 60
LINE 69  C  IF( LAB .EQ. LE ) GO TO 70
LINE 70  C  IF( LAB .EQ. LB ) GO TO 70
LINE 71  C  IF( ISP ) GO TO 70
LINE 72  C  INVALID LABEL CARD READ
LINE 73  C  WRITE( I6, 2000 )
LINE 74  C  WRITE( I6, 2018 )
LINE 75  C  WRITE( I6, 2017 ) LAB, ID, FD
LINE 76  C  GO TO 10
LINE 77  C  SPAN CARDS ARE OUT OF ORDER
LINE 78  C  WRITE( I6, 2026 ) ISP, YSPAN
LINE 79  ERROR = .TRUE.
LINE 80  C  SPAN CARD HAS BEEN READ, UPDATE SPAN COUNTER AND CHECK CARD
LINE 81  C  ISP = ISP + 1
LINE 82  C  NC = ID(2)
LINE 83  C  YSPAN = FD(1)
LINE 84  C  IF( ISP .EQ. 1 ) GO TO 21
LINE 85  C  IF( ISP .EQ. SPAN CARDS ARE OUT OF ORDER
LINE 86  C  WRITE( I6, 2000 )
LINE 87  C  WRITE( I6, 2026 ) ID(1), ISP
LINE 88  C  WRITE( I6, 2027 ) ISP, YSPAN
LINE 89  ERROR = .TRUE.
LINE 90  C  CHORD CARD NOT FOUND
LINE 91  C  WRITE( I6, 2026 ) ISP, YSPAN
LINE 92  C  GO TO 22
LINE 93  C  WRITE( I6, 2028 ) ISP
LINE 94  C  WRITE( I6, 2027 ) ISP, YSPAN
LINE 95  ERROR = .TRUE.
LINE 96  C  WRITE( I6, 2029 ) ISP
LINE 97  ERROR = .TRUE.
LINE 98  C  WRITE( I6, 2030 ) ISP
LINE 99  C  WRITE( I62027 ) ISP, YSPAN
LINE 100  C  WRITE( I6, 2028 ) ISP
LINE 101  NUMBER OF CHORD LINES MUST BE GREATER THAN OR EQUAL TO ZERO
LINE 102  C  WRITE( I6, 2029 ) ISP
LINE 103  ERROR = .TRUE.
LINE 104  C  WRITE( I6, 2030 ) ISP
LINE 105  C  WRITE( I6, 3000 ) ISP, NC, YSPAN
LINE 106  C  WRITE( I6, 3000 ) ISP, NC, YSPAN
LINE 107  C  WRITE( I6, 3000 ) ISP, NC, YSPAN
LINE 108  C  WRITE( I6, 3000 ) ISP, NC, YSPAN
LINE 109  C  READ 1ST *CHORD* CARD, MUST BE LABELED *CHORD*
LINE 110  C  READ( I5, 1030 ) LAB, ID, FD
LINE 111  C  IF( LAB .EQ. LC ) GO TO 24
LINE 112  C  CHORD CARD NOT FOUND
LINE 113  C  WRITE( I6, 2000 )
LINE 114  C  WRITE( I6, 2029 ) ISP
LINE 115  ERROR = .TRUE.
LINE 116  C  WRITE( I6, 2030 ) ISP
LINE 117  C  WRITE( I6, 3000 ) ISP, NC, YSPAN
LINE 118  C  WRITE( I6, 3000 ) ISP, NC, YSPAN

-6.28-
LINE 119 X2 = FD(2)
LINE 120 C WRITE( 16, 3002 ) IC, X1, X2
LINE 121 X2 = X2/BEETA
LINE 122 X1 = X1/BEETA
LINE 123 WRITE( ISP) = YSPAN/REFLEN
LINE 124 XLE( ISP) = 0.5*(X1+X2)/REFLEN
LINE 125 LOOP ON NUMBER OF CHORD CARDS 2, NC
LINE 126 C DO 30 IC = 2, NC
LINE 127 C READ(15, 1000) LAB, ID, X4, X3
LINE 128 C WRITE( 16, 3002 ) IC, X4, X3
LINE 129 C X3 = X3/BEETA
LINE 130 C X4 = X4/BEETA
LINE 131 C XMIN = AMIN1L( X4-X1, (X3-X2) )
LINE 132 C IF( XMIN .GT. 0.0 ) GO TO 25
LINE 133 C XMAX = AMAX1L( X4-X1, (X3-X2) )
LINE 134 C IF( XMIN .GT. 0.0 ) AND. XMAX .GT. 0.0 ) GO TO 25
LINE 135 C WRITE( 16, 2000 )
LINE 136 C WRITE( 16, 2031 ) IC, ISP
LINE 137 C ERROR = .TRUE.
LINE 138 C CONTINUE
LINE 139 C NJ = 6*NE
LINE 140 C XYZ(NJE1) = X1/REFLEN
LINE 141 C XYZ(NJE2) = X2/REFLEN
LINE 142 C XYZ(NJE3) = X3/REFLEN
LINE 143 C XYZ(NJE4) = X4/REFLEN
LINE 144 C Y1 = Y2
LINE 145 C NNCH( ISP) = NE & NETSV
LINE 146 C GO TO 10
LINE 147 C 50 CONTINUE
LINE 148 C FREQUENCY CARD READ
LINE 149 C FREQ = .TRUE.
LINE 150 C DO 55 I = 1, 4
LINE 151 C FREQ(I) = FD(I)
LINE 152 C NFREQ = ID(I)
LINE 153 C IF( VFREQ .GT. 4 ) READ(15, 1001) ( FREQ(I), I=5, NFREQ )
LINE 154 C WRITE( 16, 3006 ) NFREC, ( FREQ(I), I=1,NFREQ)
LINE 155 C GO TO 10
LINE 156 C 60 CONTINUE
LINE 157 C END CARD READ
LINE 158 C 30TH MODE AND FREQUENCY DATA MISSING
LINE 159 C WRITE( 16, 2000 )
LINE 160 C WRITE( 16, 2014 )
LINE 161 C WRITE( 16, 2015 )
LINE 162 C ERROR = .TRUE.
LINE 163 C GO TO 90
-6.29-
LINE 181 65 CONTINUE
LINE 182 IF( FREQ ) GO TO 90
LINE 183 WRITE( 16, 2000 )
LINE 184 WRITE( 16, 2015 )
LINE 185 ERROR = .TRUE.
LINE 186 GO TO 90
LINE 187 C
LINE 188 C 70 CONTINUE
LINE 189 C LABEL CARD EXPECTED BUT NOT RECEIVED
LINE 190 WRITE( 16, 2000 )
LINE 191 WRITE( 16, 2016 )
LINE 192 WRITE( 16, 2017 ) LAB, ID, FO
LINE 193 ERROR = .TRUE.
LINE 194 GO TO 10
LINE 195 C
LINE 196 80 CONTINUE
LINE 197 C GRID CARD READ
LINE 198 C
LINE 199 I = ID(2)
LINE 200 J = ID(1)
LINE 201 IF( I > NSE(J) ) GO TO 81
LINE 202 IF( J > NKing( J ) ) GO TO 82
LINE 203 IF( J .EQ. JW .AND. I .LE. IS ) GO TO 82
LINE 204 ICL(I,J) = -1
LINE 205 GO TO 90
LINE 206 C
LINE 207 81 CONTINUE
LINE 208 C GRID DATA SPECIFIED FOR SECTION THAT DOES NOT EXIST
LINE 209 WRITE( 16, 2000 )
LINE 210 WRITE( 16, 2021 ) J, I
LINE 211 ERROR = .TRUE.
LINE 212 GO TO 90
LINE 213 C
LINE 214 82 CONTINUE
LINE 215 C GRID DATA FOR NEXT GRID SECTION IS OUT OF ORDER
LINE 216 WRITE( 16, 2000 )
LINE 217 WRITE( 16, 2022 ) J, I
LINE 218 ERROR = .TRUE.
LINE 219 C
LINE 220 90 CONTINUE
LINE 221 C SECTION TO RETURN
LINE 222 C
LINE 223 NETSV = NETSV & NE
LINE 224 IF( ISP .EQ. NSP ) RETURN
LINE 225 C NUMBER OF SPAN CARDS READ DOES NOT AGREE WITH NO. SPANS (NSP)
LINE 226 WRITE( 16, 2030 ) IS, JW, NSP, ISP
LINE 227 RETURN
LINE 228 C
LINE 229 1000 FORMAT( A4, 2X, 9I2, 4E12.0 )
LINE 230 1001 FORMAT( 24X, 4E12.0 )
LINE 231 C
LINE 232 2030 FORMAT( /// IX, 129(1H*) //24H ERROR IN READING INPUT. )
LINE 233 2031 FORMAT( 37H MODE DATA IS MISSING OR OUT OF ORDER )
LINE 234 2032 FORMAT( 42H FREQUENCY DATA IS MISSING )R OUT OF ORDER )
LINE 235 2033 FORMAT( 42H A LABEL CARD IS EXPECTED BUT NOT PRESENT. )
LINE 236 2034 FORMAT( 6TH CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. / IX, )
LINE 237 2035 FORMAT( 2A4, 2X, 9I2, 4E12.0 )
LINE 238 2036 FORMAT( 32H AN INVAILDE LABEL CARD WAS READ. )
LINE 239 2037 FORMAT( 27H ERROR ON GRID LABEL CARD. )
LINE 240 2038 FORMAT( 27H GRID DATA PRESENT FOR KING, 12, 8H SECTION, 12 / )
LINE 241 2039 FORMAT( 153H THIS SECTION HAS NOT BEEN DEFINED FOR THE STRUCTURE. )
LINE 242 C
LINE 243  2022 FORMAT ( 19H GRID DATA FOR WING, I2, 8H SECTION, I2, 17H IS OUT OF ORDER, I2 )
LINE 244       1 ORDER, I2 ) 01202440
LINE 245  2026 FORMAT ( 43H SPAN CARDS ARE OUT OF ORDER. CARD FOR SPAN, I2, 12 )
LINE 246       1 28H WAS READ WHEN CARD FOR SPAN, I2, 13H WAS EXPECTED ) 01202460
LINE 247  2027 FORMAT ( 25H THE WIDTH OF SPAN NUMBER, I2, 7H EQUALS, E14.6 )
LINE 248       1 42H THE SPAN WIDTH MUST BE GREATER THAN ZERO. ) 01202480
LINE 249  2028 FORMAT ( 34H THE NUMBER OF CHORD LINES IN SPAN, I2, 7H EQUALS, I3 )
LINE 250       1 48H THE NUMBER OF CHORDS MUST BE GREATER THAN ZERO. ) 01202500
LINE 251  2029 FORMAT ( 20H CHORD CARD FOR SPAN, I3, 13H NOT FOUND )
LINE 252  2030 FORMAT ( 24H SECTION, I2, 8H OF WING, I2, 20H WAS DEFINED TO HAVE, I3 )
LINE 253       1 13, 7H SPANS,, I4, 33H SPAN CARDS WERE ACTUALLY READ. ) 01202530
LINE 254  2031 FORMAT ( 18H CHORD LINE NUMBER, I3, 8H OF SPAN, I3, 62H IS IDENTICAL TO, I2 )
LINE 255       1 SPANS, OR IS ABOVE THE PREVIOUS CHORD LINE, 2 ) 01202550
LINE 256  2 ) 01202560
LINE 257  3000 FORMAT ( 1H/24X, 19H GRID INPUT FOR WING, I2, 8H SECTION, I2 )
LINE 258       3000 FORMAT ( 24X, 4H SPAN, I3, 4H HAS, I3, 27H CHORD LINES AND A WIDTH )
LINE 259       3002 FORMAT ( 1F, E13.6 / 24X, 5H CHORD, 9X, 2H X1, 14X, 2H X2 )
LINE 260       1F, E13.6 / 24X, 5H CHORD, 9X, 2H X1, 14X, 2H X2 ) 01202600
LINE 261       3002 FORMAT ( 24X, I5, 5X, 2E16.6 ) 01202610
LINE 262  3006 FORMAT ( // // 23X, 21H NUMBER OF FREQUENCIES )
LINE 263       3006 FORMAT ( 1E5, I12 / 20X, 19 H LIST OF FREQUENCIES, 5F14.4 )
LINE 264       3006 FORMAT ( 1E5, I12 / 20X, 19 H LIST OF FREQUENCIES, 5F14.4 ) 01202640

/END READ

END OF JOB.

39.4 SEC. USED .011 HRS. CHARGED 49.974 HRS. REMAINING

-6.31-
SUBROUTINE INP1 ( I5, I6, TITLE, ITRUN, MACH, REFLEN, XP, XPIN, PLTR, ITAE, EARO, BEETA, LAB, ID, FD, ERROR )

REAL MACH, PLOT = .FALSE.

DIMENSION TITLE(L4), FD(4), ID(9)

COMMON / NEXTCS / IFLUSH

DATA LRUN, 4HRUN / 0

READ RUN CARD

READ( I5, 1000 ) LAB, 10, FD

IF( LAB .EQ. LRUN ) GO TO 2

WRITE( 16, 2000 )

WRITE( 16, 2023 )

IFLUSH = 1

2 CONTINUE

ITRUN = ID(1)

PLOT = .FALSE.

IF( ID(2) .EQ. 1 ) PLOT = .TRUE.

IF( lab .EQ. 0 ) ITAE = 1

MACH = FD(1)

BEETA = SQRT(MACH*MACH-1.0)

REFLEN = FD(2)

XPIN = FD(3)

EARO = FD(4)

IF( EARO .EQ. 0.0) EARO = 1.1

IF( MACH .GT. 1.0 ) GO TO 5

WRITE( 16, 2000 )

WRITE( 16, 2024 ) MACH

MACH = 2.0

ERROR = .TRUE.

5 CONTINUE

IF( REFLEN .EQ. 0.0 ) GO TO 6

WRITE( 16, 2000 )

WRITE( 16, 2025 )

ERROR = .TRUE.

6 CONTINUE

REFLEN = 1.

40 CONTINUE

IF( EARO .EQ. 0.0 ) GO TO 7

WRITE( 16, 2000 )

WRITE( 16, 2027 ) EARO

ERROR = .TRUE.

7 CONTINUE

XP = XPIN/(BEETA*REFLEN)

WRITE ( 16, 3007 ) MACH, REFLEN, XPIN

WRITE( 16, 3019 ) EARO

WRITE( 16, 3008 ) ITRUN

IF( ITRUN .EQ. 3 ) GO TO 88

WRITE( 16, 3009 )

GO TO 9

88 CONTINUE

END
LINE 57 WRITE (16, 3018)
LINE 58 GO TO 9
LINE 59 8 CONTINUE
LINE 60 WRITE (16, 3010)
LINE 61 9 CONTINUE
LINE 62 WRITE (16, 3011) ID12)
LINE 63 IF (PLOTR ) GO TO 11
LINE 64 WRITE (16,3012)
LINE 65 GO TO 12
LINE 66 11 CONTINUE
LINE 67 WRITE (16, 3013)
LINE 68 12 CONTINUE
LINE 69 WRITE (16,3015)
LINE 70 IF (IWTE .NE. 1) WRITE (16, 3016)
LINE 71 IF (IWTE .EQ. 1) RETURN
LINE 72 RETURN
LINE 73 1000 FORMAT (A49

LINE 74 2000 FORMAT(/// 1X, 1291IH) /24H ERROR IN READING INPUT. J
LINE 75 2023 FORMAT(3H RUN CARD IS MISSING OR OUT OF ORDER / 89H RUN CARD MUST BE IMMEDIATELY FOLLOW THE TITLE CARD AND BE THE SECOND CARD IN THE INPUT DECK / 18H JOB IS TERMINATED /

LINE 76 2100 FORMAT(/// 1X, 1291IH) /24H MACH NUMBER MUST BE GREATER THAN 1.0 / 24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA /
LINE 77 2025 FORMAT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. /116H REFERENCE LENGTH MUST BE INPUT TO CHECK THE REST OF THE DATA. /3 A NON-NEGATIVE FLOATING POINT NUMBER IN COLUMNS 39 THROUGH 48 OF THE MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 78 2026 FORMAT(/// 1X, 1291IH) /24H MACH NUMBER WAS READ AS, E13.6 /
LINE 79 2027 FORMAT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 80 1000 FORMAT(/// 1X, 1291IH) /24H MACH NUMBER MUST BE GREATER THAN 1.0 / 24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 81 1000 FORMAT(/// 1X, 1291IH) /24H MACH NUMBER MUST BE GREATER THAN 1.0 / 24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 82 1000 FORMAT(/// 1X, 1291IH) /24H MACH NUMBER MUST BE GREATER THAN 1.0 / 24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 83 2028 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 84 2029 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 85 2030 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 86 2031 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 87 2032 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 88 2033 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 89 2034 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 90 2035 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 91 2036 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 92 2037 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 93 2038 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 94 2039 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 95 2040 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 96 2041 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 97 2042 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 98 2043 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 99 2044 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 100 2045 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 101 2046 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 102 2047 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 103 2048 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 104 2049 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 105 2050 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 106 2051 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

LINE 107 2052 FormaT(/// 1X, 1291IH) /24H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA. 

END
CARD DECK 14

DISPLAY ... FILE

1  FUNCTION IRELE ( GT, GW, NET) = 01400010
2  COMPLEX GBAR, RI, RIS = 01400020
3  COMMON / CEPS / EPS = 01400030
4  COMPLEX GT(1), GW(1) = 01400040
5  IRELE = 1 = 01400050
6  ANET = NET = 01400060
7  RABS() = 0.0 = 01400070
8  DO 600 IEL = 1, NET = 01400080
9  GTABS = CABS(GT(IEL)) = 01400090
10  GABS() = CABS( GW(IEL)) = 01400100
11  IF (GABS() .EQ. 0.0 .AND. GTABS .EQ. 3.0) ANET = ANET - 1.0 = 01400110
12  IF (GABS() .EQ. 0.0 .AND. GTABS .EQ. 0.0) GO TO 600 = 01400120
13  RI = GT(IEL) - GW(IEL) = 01400130
14  RIS = RI * RI = 01400140
15  IF( GTABS .NE. 0.0) RIS = RIS / GT(IEL) = 01400150
16  600 RABS() = CABS( RIS) GRABS() = 01400160
17  ERROR = RABS() / ANET = 01400170
18  ERRDR = SQRT( ERROR) = 01400180
19  IF( ERRDR .GT. EPS ) RETURN = 01400190
20  IRELE = 0 = 01400200
21  RETURN = 01400210
22  END = 01400220
DISPLAY...FILE

DECK 15

LINE 1  SURROJUTE I TRATE (I,9,NWR)N,IP DIP DNWR, DNIK RAR RM 01500110
LINE 2  1 NET A IZ ZHES(N'W) 01500200
LINE 3  COMPLEX SN GT SH 01500030
LINE 4  COMPLEX GMESS 01500040
LINE 5  REAL XMR NW 01500050
LINE 6  INTEGER PRINT 01500060
LINE 7  DIMENSION DNET IZNET) 0I500070
LINE 8  DIMENSION X(MC,MCI) IPMC) DNMRC DNIKMC RARMC RMCMC) 01500080
LINE 9  DIMENSION ZMESS(200) 01500090
LINE 10  DIMENSION XW(200) 01501000
LINE 11  DIMENSION W(200) 01501100
LINE 12  COMMON /CEPS/ EPS 01501200
LINE 13  COMMON /ITAT/ ITMAX 01501300
LINE 14  COMMON /IGWESS/ IGMESS 01501400
LINE 15  COMMON /KII2/ KST11,KST12 01501500
LINE 16  COMMON /TAPE/ IN I1O 01501600
LINE 17  COMMON /VLCOM/ NMAX,PRINT NWBLDwR, NROW(20) 01501700
LINE 18  REWIND 110 01501800
LINE 19  NWING = NET 01501900
LINE 20  IF (NWR.IEQ.1) GO TO 700 01502000
LINE 21  IMAX=30 01502100
LINE 22  IF (ITMAX.NEQ .0) IMAX=ITMAX 01502200
LINE 23  SET CONVERGENCE CRITERIA - EPS 01502300
LINE 24  ALF1=0.9 01502400
LINE 25  ALF2=1.1 01502500
LINE 26  ALF=ALF1 01502600
LINE 27  DO 99 99 01502700
LINE 28  NWING = NET 01502800
LINE 29  IF (NWING .EQ. 1) GO TO 700 01502900
LINE 30  REWIND 110 01503000
LINE 31  READ (110) X(IP) 01503100
LINE 32  READ (110) X(IP) 01503200
LINE 33  READ (110) X(IP) 01503300
LINE 34  READ (110) X(IP) 01503400
LINE 35  READ (110) X(IP) 01503500
LINE 36  READ (110) X(IP) 01503600
LINE 37  READ (110) X(IP) 01503700
LINE 38  READ (110) X(IP) 01503800
LINE 39  READ (110) X(IP) 01503900
LINE 40  READ (110) X(IP) 01504000
LINE 41  READ (110) X(IP) 01504100
LINE 42  READ (110) X(IP) 01504200
LINE 43  READ (110) X(IP) 01504300
LINE 44  READ (110) X(IP) 01504400
LINE 45  READ (110) X(IP) 01504500
LINE 46  READ (110) X(IP) 01504600
LINE 47  READ (110) X(IP) 01504700
LINE 48  READ (110) X(IP) 01504800
LINE 49  READ (110) X(IP) 01504900
LINE 50  READ (110) X(IP) 01505000
LINE 51  READ (110) X(IP) 01505100
LINE 52  READ (110) X(IP) 01505200
LINE 53  READ (110) X(IP) 01505300
LINE 54  READ (110) X(IP) 01505400
LINE 55  READ (110) X(IP) 01505500
LINE 56  READ (110) X(IP) 01505600

-6.39-
LINE 57    150 CONTINUE
LINE 58    160 CONTINUE
LINE 59    165 CONTINUE
LINE 60    IF (N4L0K.EQ.1) GO TO 530
LINE 61    170 CONTINUE
LINE 62    REWIND 110
LINE 63    190 IF (LABSPRINT.LT.3) GO TO 200
LINE 64    WRITE (6,630) IT
LINE 65    WRITE (6,650) NWLKNG, (GW(J), J=1, NWLKNG)
LINE 66    200 CONTINUE
LINE 67    IF (ITEST.EQ.3.AND.ITNE.EQ.1) GO TO 530
LINE 68    IF (IT.EQ.IMAX) GO TO 540
LINE 69    JJ=0
LINE 70    IF (IT.EQ.IMAX) GO TO 540
LINE 71    JJ=0
LINE 72    IF (IT.EQ.IMAX) GO TO 540
LINE 73    REWIND 190
LINE 74    REWIND 110
LINE 75    JJ=0
LINE 76    IF (IT.EQ.IMAX) GO TO 540
LINE 77    JJ=0
LINE 78    IF (IT.EQ.IMAX) GO TO 540
LINE 79    DD 510 I=1,NWBL0K
LINE 80    NORW=NRW(JW1)
LINE 81    READ (110) D, IP
LINE 82    IF (LABSPRINT.LT.3) GO TO 340
LINE 83    WRITE (6,560) I,NROW
LINE 84    WRITE (6,560) NROW,0
LINE 85    WRITE (6,560) NROW,0
LINE 86    WRITE (6,560) NROW,0
LINE 87    WRITE (6,560) NROW,0
LINE 88    WRITE (6,560) NROW,0
LINE 89    WRITE (6,560) NROW,0
LINE 90    WRITE (6,560) NROW,0
LINE 91    WRITE (6,560) NROW,0
LINE 92    DNWR(J)=0.0
LINE 93    DNMRI(J)=0.0
LINE 94    IF (NWING.LE.NMAX) GO TO 470
LINE 95    IF (I.EQ.3) GO TO 530
LINE 96    IF (I.EQ.3) GO TO 530
LINE 97    IF (J.EQ.3) GO TO 530
LINE 98    IF (J.EQ.3) GO TO 530
LINE 99    IF (J.EQ.3) GO TO 530
LINE 100   IF (J.EQ.3) GO TO 530
LINE 101   IF (J.EQ.3) GO TO 530
LINE 102   IF (J.EQ.3) GO TO 530
LINE 103   DNWR(J)=DNWR(J) & A(M) = REAL (GW(K))
LINE 104   DNWR(J)=DNWR(J) & A(M) = A(M) & AIMAG (GW(K))
LINE 105   RNWR(J)=RNWR(J) & NW(J)
LINE 106   RNWR(J)=RNWR(J) & NW(J)
LINE 107   RNWR(J)=RNWR(J) & NW(J)
LINE 108   If (J.EQ.3) GO TO 530
LINE 109   C3000 FORMAT(4X, 27H J,J,DNWR(J),RNWR(J), RW(J) = / (6X, 214,6E12.4) )
LINE 110   471 CONTINUE
LINE 111   CALL SOLVE (NWQR,NMAX,0,RWR, IP )
LINE 112   CALL SOLVE (NRW,NMAX,0,RW, IP )
LINE 113   DO 500 M=1,NROW
LINE 114   GT(I,J)=CMPLX(RW(J), RNWR(J))
LINE 115   SAVE=CMPLX(RW(J), RNWR(J))
LINE 116   SWR(I,J)=CMPLX(RW(J), RNWR(J))
LINE 117   SWR(I,J)=CMPLX(RW(J), RNWR(J))
LINE 118   IF (IT.EQ.2) GO TO 450

-6.36-
LINE 119  SAVE=+W(I) SAVE 01501190
LINE 120  ALF=ALF1 01501200
LINE 121  IF (SAVE.GE.0.1) ALF=ALF2 01501210
LINE 122  490 CONTINUE 01501220
LINE 123  G(I)=ALF*G(I) & (1.-ALF) * G(I) 01501230
LINE 124  C  ITEST = IMELE(G(I), G(I)). ITST) 01501240
LINE 125  IT=IT+1 01501250
LINE 126  500 CONTINUE 01501260
LINE 127  JJ=J+1&ROW(I) 01501270
LINE 128  510 CONTINUE 01501280
LINE 129  ITEST = IMELE(GT, Gw, NET) 01501290
LINE 130  GO TO 190 01501300
LINE 131  530 WRITE (6,580) IT, EPS 01501310
LINE 132  RETURN 01501320
LINE 133  540 WRITE (6,590) IMAX, EPS 01501330
LINE 134  C IF (LMETH.EQ.0) GO TO 550 01501340
LINE 135  C  JJN TL 4 UE 01501350
LINE 136  C GO TO 60 01501360
LINE 137  550 CONTINUE 01501370
LINE 138  WRITE (6,600) NWIN, GT(N), Gw(N) 01501380
LINE 139  WRITE (6,610) NWIN, (GT(N), Gw(N), N=1,NWIN) 01501390
LINE 140  RETURN 01501400
LINE 141  610 FORMAT(1HO,40HTHE SOLUTION AT THE PRESENT ITERATION IS) 01501410
LINE 142  RETURN 01501420
LINE 143  C  01501430
LINE 144  C  01501440
LINE 145  570 FORMAT (2X,10H(N), N=1,13/(1X,10F10.5)) 01501450
LINE 146  580 FORMAT (1HO,30HTHE ITERATION CONVERGED AFTER ,13,2X,35ITERATIONS) 01501460
LINE 147  1WITH A TEST CRITERION OF,F10.7) 01501470
LINE 148  590 FORMAT (1HO,37HTHE ITERATION DID NOT CONVERGE AFTER ,13,2X,35ITERATIONS) 01501480
LINE 149  4ATION WITH A TEST CRITERION OF,F10.7) 01501490
LINE 150  600 FORMAT (1HO,40HTHE SOLUTION AT THE PREVIOUS ITERATION IS) 01501500
LINE 151  610 FORMAT (1HO,40HTHE SOLUTION AT THE PRESENT ITERATION IS) 01501510
LINE 152  620 FORMAT (1HO,30HITERATION NUMBER,14) 01501520
LINE 153  630 FORMAT (2X,10H(N), N=1,13/(1X, 6E18.5)) 01501530
LINE 154  C  01501540
LINE 155  END 01501550

-6.37-
DISPL...FILE
DECK 16

LINE 1  SUBROUTINE LOOPW (NET, N WING, XYZ, NSP, SYM, TRS, NNCH, NE)
LINE 2  1 NSECT, ZSECT, IONS(16), TAPER, XCEN, AREA, WRROW, LRECL,
LINE 3  2 WTRERWLWTE, PERC, ERROR, I WTE, REFLEN,
LINE 4  C NET TOTAL NUMBER OF ELEMENTS
LINE 5  C NTERM NUMBER OF TERMS IN INFLUENCE FUNCTION
LINE 6  C N WING NUMBER OF WINGS IN STRUCTURE
LINE 7  C XYZ COORDINATE ARRAY OF ALL ELEMENTS
LINE 8  C NSP NSP(I,J), NUM SPANS IN SECTION I OF WING J
LINE 9  C SYM(J) SYMMETRY CODE FOR WING J
LINE 10  C TRS TRS(I,J,1) TRANSFORMATION MATRIX OF SECT. I OF WING J
LINE 11  C NNCH ELEMENT NUMBER AT END OF SPAN
LINE 12  C IES(J) NUMBER OF ELEMENTS IN SECTION I OF WING J
LINE 13  C ZSECT(I,J) COORDINATE OF ALL ELEMENTS OF SECTION I, WING J
LINE 14  C IONS(K1,K2) = 1 IF SECTION K2 HAS INFLU. ON REC. SECTION K1
LINE 15  C ISENC I6 LOGICAL UNIT NUMBER FOR OUTPUT (PRINTER)
LINE 16  C TAPE8 LOGICAL UNIT NUMBER OF TAPE TO CONTAIN FREQUENCY-
LINE 17  C PENDING TERMS
LINE 18  C LRECL LENGTH OF WRROW EQUAL TO NTERM#NET
LINE 19  C ERROR LOGICAL SET TO TRUE IF ERROR IS DETECTED
LINE 20  C XCEN X CENTER OF EACH ELEMENT (OUTPUT)
LINE 21  C XO X COORDINATE OF RECEIVING PT. (CENTER OF ELEMENT)
LINE 22  C YRO Y COORDINATE OF RECEIVING PT. (IN RIGHT SIDE)
LINE 23  C YLO Y COORDINATE OF RECEIVING PT. (IN LEFT SIDE)
LINE 24  C AREA AREA FOR EACH ELEMENT (OUTPUT)
LINE 25  C WRROW STORAGE FOR FREQUENCY TERMS FOR ONE RECEIVING ELEMENT
LINE 26  C FREQ TERMS FOR ALL INFLUENCING ELEMENTS
LINE 27  C LRECL LENGTH OF WRROW EQUAL TO NTERM#NET
LINE 28  C ERROR LOGICAL SET TO TRUE IF ERROR IS DETECTED
LINE 29  C COMPLEX WTRERWLWTE
LINE 30  C COMPLEX WTES
LINE 31  C REAL MACH
LINE 32  C INTEGER TAPE8
LINE 33  C LOGICAL BKWSP
LINE 34  C LOGICAL ERROR
LINE 35  C LOGICAL TREDGE
LINE 36  C LOGICAL WAKE, WAKE1, WAKENZ
LINE 37  C LOGICAL WAKE 2
LINE 38  C DIMENSION NSP(12,1), TRS(4,3,1), NSP(1,1)
LINE 39  C NNCH(1), SYM(1)
LINE 40  C NSECT(1), NE(1,1), ZSECT(1), XCEN(1), XYZ(1), NSP(1,1)
LINE 41  C AREA(1)
LINE 42  C COMMON / BASIC/MACH
LINE 43  C COMMON / FREQ / N PEREQ, CF(12,30), FREQ(12), FREQP(12), FTEST
LINE 44  C COMMON / RQLL / Y(4), YRO, YLO, ZR, LZR, ZL, LZZ, IRC, IN
LINE 45  C COMMON / TAPES / 15,110,111,112,113,13,14,15,16
LINE 46  C COMMON / WAKEUP, NTEST(12), TREDGE, WAKE1, WAKE, WAKENZ, WK
LINE 47  C COMMON / WRE / XO, YO, ZO, ZD, ZD, SYM, NINSID
LINE 48  C ------------------------------
LINE 49  C THE FREQUENCY-INDEPENDENT TERMS FOR ALL INFLUENCING ELEMENTS
LINE 50  C OF A GIVEN RECEIVING ELEMENT ARE STOPED IN WRROW
LINE 51  C COMPUTE 1 ROW FOR EVERY ELEMENT (CALLED RECEIVING ELEMENT)
LINE 52  C TERMS IS THE NO. TERMS USED TO COMPUTE AN ELEMENT OF W
LINE 53  C INDXI INDEX INTO XYZ FOR RECEIVING ELEMENT
LINE 54  C INDXI INDEX INTO XYZ FOR INFLUENCING ELEMENT
LINE 55  C IRC RECEIVING ELEMENT COUNTER
LINE 56  C IRC INFLUENCING ELEMENT COUNTER

-6.38-
K1 IS COUNTER FOR RECEIVING SECTION, K2 FOR INFLUENCING SECTION.

DO 100 J=1,NWING
NSR = NSECT(J)
SYM = SYM(J)
DO 95 I=1,NSR
K1 = K1 + 1
IRC = IRC + 1
ZCR = ZSECT(I,J)
LOOP ON THE NUMBER OF ELEMENTS IN SECTION I OF WING J

DO 90 IER = I,NER
WAKE2 = .FALSE.
IRC = IRC + 1
INDX = IRC * 6
IF (WTE.EQ.0) GO TO 50
DO 45 IE = 1,LRWTE
45 WTEROW(IE1) = (0.0, 0.0)
CONTINUE

IF (WTE .EQ. 0) GO TO 50
DO 45 IE = 1,LRWTE
45 WTEROW(IE1) = (0.0, 0.0)
CONTINUE

CALL RT01 ( TRS(I,J), TRS(I,J), YCR, ZCR, ZFO, ZLO )
ZL = ZL*ZCR
ZLZL = ZL*ZCR
ZREDGE = .FALSE.
NINSID = 0
IIN = IRC
INDX = 6*IRC - 5
ICOL = NFREQ*IRC - 1 & 1
SET UP CONSTANTS FOR INFLUENCING ELEMENTS

INDXI = 1
IN = 1
NSPI = 0
K2 = 1
ICOL = 1

LOOP ON INFLU. ELEMENT BY WING JJ, SECTION II (DO 85, JJ = 0)

DO 85 JJ = 1, NWING
SYMK = SYM(JJ)
NSI = NSECT(JJ)
DO 80 II = 1, NSI

DO 80 I = 1, NSI
NEI = NEI*NFREQ
WROW(ICOL) = 0.0
ICOL = ICOL + 1
55 CONTINUE

SECTION IS ASSUMED TO HAVE SOME ELEMENTS WITH NON-ZERO INFLUENCE. EACH ELEMENT MUST BE CONSIDERED.
LINE 181 C NSPI = NSPI + 1
LINE 183 NPFSMI = NCH(NSPI) - IIN
LINE 184 NEPS = NEPSMI + 1
LINE 185 C BUMP INXI AND ICOL TO LAST ELEMENT IN SPAN
LINE 187 IIN = IIN & NEPS
LINE 188 INDXI = INDXI & 6*NEPSMI
LINE 189 ICOL = ICOL & NFREQ*NEPSMI
LINE 190 BKWSP = .FALSE.
LINE 191 NINSID = 0
LINE 192 TREDGE = .TRUE.
LINE 193 C LOOP ON INFLU. ELEM. OF SPAN ISP FROM BOTTOM UP DO 70 IEI
LINE 196 DO 70 IEI = 19NEPS
LINE 197 C
LINE 198 IIN = IIN - 1
LINE 199 IF( BKWSP) GO TO 65
LINE 200 WAKE1 = .FALSE.
LINE 201 C
LINE 202 CALL WINTGR( XYZ(INDXI), XYZ(IEXI), WRUW(ICOL) )
LINE 203 C TREDGE = .FALSE.
LINE 204 C
LINE 205 C IF( VINSID .EQ. 0 ) GO TO 69
LINE 206 DO 56 IW=L,NFREQ
LINE 207 INDWK = INDWK & 1
LINE 208 WTER(W(INDWK) = WTER(IW) )
LINE 209 CONTINUE
LINE 210 C IF( VINSID .EQ. 1 ) GO TO 69
LINE 211 RD = ABS( WROW(ICOL) / WIT )
LINE 212 IF( RD .GT. PERC ) GO TO 62
LINE 213 BKWSP = .TRUE.
LINE 214 IW = ICOL - 1
LINE 215 DO 68 IF=1,NFREQ
LINE 216 IJ = IJ & 1
LINE 217 WROW(IJ) = 0.0
LINE 218 CONTINUE
LINE 219 DO 68 IJ = 1,NFREQ
LINE 220 WFSV(IF) = 0.0
LINE 221 CONTINUE
LINE 222 IF( VINSID .LE. 2 ) GO TO 69
LINE 223 IJ = ICOL & NFREQ
LINE 224 IF = 1,NFREQ
LINE 225 CONTINUE
LINE 226 CONTINUE
LINE 227 IF( VINSID .GT. 3 ) GO TO 63
LINE 228 IJ = ICOL & NFREQ - 1
LINE 229 GO TO 63
LINE 230 IF( VINSID .GT. 3 ) GO TO 63
LINE 231 C
LINE 232 CONTINUE
LINE 233 TEST = WROW(IJ)
LINE 234 GO TO 69
LINE 235 C CONTINUE
LINE 236 CONTINUE
LINE 237 TEST = TEST - WROW(IJ)
LINE 238 IF( TEST .LT. 0.0 ) TEST = TEST/WROW(IJ)
LINE 239 IF( TEST .GT. 0.0 ) GO TO 61
LINE 240 BKWSP = .TRUE.
LINE 241 IJ = ICOL - 1
LINE 242 DO 64 IF=1,NFREQ

-6.41-
01602430
LINE 243  IJ = IJ & 1
LINE 244  64 WFSV(IF) = WROW(IJ)
LINE 245  GO TO 69
LINE 246  C
LINE 247  65 CONTINUE
LINE 248  IJ = ICOL - 1
LINE 249  DO 67 IF=1,NFREQ
LINE 250  IJ = IJ & 1
LINE 251  67 WROW(IJ) = WFSV(IF)
LINE 252  C
LINE 253  65 CONTINUE
LINE 254  ICOL = ICOL - NFREQ
LINE 255  INDXI = INDXI - 6
LINE 256  C
LINE 257  CONTINUE ON ELEMENTS IN THE INFLUENCING SPAN
LINE 258  70 CONTINUE
LINE 259  C
LINE 260  IIN = IIN & NEPS
LINE 261  INDXI = INDXI & 6*NEPS & 6
LINE 262  ICOL = ICOL & NFREQ*NEPS & NFREQ
LINE 263  C
LINE 264  END LOOP ON INFLUENCING SPANS OF SECTION II OF WING JJ
LINE 265  75 CONTINUE
LINE 266  C
LINE 267  END LOOP ON INFLUENCING SECTION II OF WING JJ
LINE 268  80 CONTINUE
LINE 269  C
LINE 270  END LOOP ON INFLUENCING WING JJ
LINE 271  85 CONTINUE
LINE 272  C
LINE 273  CONTINUE ON RECEIVING ELEMENT IN SECTION I OF WING J
LINE 274  C WROW I ROW HAS BEEN COMPUTED FOR A GIVEN RECEIVING ELEMENT.
LINE 275  C WRITE WROW ON FILE AND LOOP TO NEXT RECEIVING ELEMENT
LINE 276  WRITE(TAPE8) IRC,WROW
LINE 277  C ZERO FOR WAKE ELEMENTS
LINE 278  WRITE(TAPE8) IRC,WROW
LINE 279  C WAKE = FALSE - NO WAKE ELEMENTS FOR ANY ELEMENT
LINE 280  C WAKE = TRUE - ELEMENT IS WAKED
LINE 281  IF (IWE .EQ.0) GO TO 90
LINE 282  IF (.NOT. WAKE) GO TO 90
LINE 283  WRITE( 15 ) IRC,WTEROW
LINE 284  KH = KW + 1
LINE 285  C IF(IWE.NE.0)WRITE(6,3001)IRC,IER,I,J,WTEROW
LINE 286  C WRITE(6,3001)IRC,IER,I,J,WTEROW
LINE 287  C IWE = IWE + 1
LINE 288  C IWE = IWE + 1
LINE 289  C WRITE(6,3001)IRC,IER,I,J,WTEROW
LINE 290  C END LOOP ON RECEIVING ELEMENT IN SECTION I OF WING J
LINE 291  C 90 CONTINUE
LINE 292  CONTINUE ON RECEIVING ELEMENT IN SECTION I OF WING J
LINE 293  95 CONTINUE
LINE 294  CONTINUE ON RECEIVING SECTION I OF WING J
LINE 295  C
LINE 296  C
LINE 297  C
LINE 298  C END LOOP ON RECEIVING WING J
LINE 299  100 CONTINUE
LINE 300  C
LINE 301  C
LINE 302  110 CONTINUE
LINE 303  C
LINE 304  C

-6.42-
LINE 305  IF(WAKE)   END FILE 115
LINE 306   I6  =6
LINE 307  IF(.NOT. WAKE) WRITE(16,1000)
LINE 308  IF(WAKE) WRITE(16,1010) KW
LINE 309  1000 FORMAT(1HO///16X,37HNO WAKE EFFECTS HAVE BEEN DETERMINED )
LINE 310  1010 FORMAT(1HO///16X,38H NO WAKE EFFECTS HAVE BEEN DETERMINED FOR )
LINE 311  1 9H ELEMENTS )
LINE 312  END FILE TAPE8
LINE 313   RETURN
LINE 314 C3001 FORMAT/// 18H RECEIVING ELEMENT, 15 , 5X, 6HNUMBER, 14 , 5X,
LINE 315 C 1 10HJF SECTION, 12 , 5X, 7HOF WING,12 // 11X, 10E13.4 )
LINE 316   END
SUBROUTINE MESH(CORNX,CORNY,CORNZ, XYZ, Z, TRS, NNCH, SLE, NSS, NE, NETSV, EACO)

PURPOSE
GENERATE MESH FOR 1 SECTION WITHOUT CONTROL LINES

INPUT
CORNX, CORNY, CORNZ X, Y, AND Z COORD. OF CORNER PTS 1, 2, 3, 4

YSP DESIRED NUMBER OF SPANS FOR SECTION (INPUT BY USER)

EACO ELEMENT ASPECT RATIO

OUTPUT
XYZ 4 X AND 2 Y LOCAL COORD. FOR EACH ELEMENT OF SECTION

Z Z COORDINATE FOR ALL ELEMENTS IN SECTION (FROM TRANS)

TRS 2 BY 2 TRANS MATRIX TO GC FROM LOCAL TO GLOBAL

NNCH ELEMENT NUMBER (W.R.T. TOTAL STRUCTURE) OF LAST ELEMENT IN SPAN.

XLE AVERAGE X COORDINATE OF LEADING EDGE PER SPAN

NE NUMBER OF ELEMENTS IN SECTION.

NETSV COUNTER OF TOTAL NUMBER OF ELEMENTS IN STRUCTURE

CORNX(l), CORNY(l), CORNZ(l), TRS(l), X(4), Y(4)

DIMENSION XLE(l), SW(l)

DIMENSION XYZ(L), NNCH(ZD)

COMM /DAMACH, BETA

EQUIVALENCE (X(l), X1), (X(2), X2), (X(3), X3), (X(4), X4)

EQUIVALENCE (Y(l), Y1), (Y(2), Y2), (Y(3), Y3), (Y(4), Y4)

DATA DTEST /* 3 */

C NOTE THAT ELEMENT IS NUMBERED CLOCKWISE WHILE SECTION IS COUNTERCLOCKWISE

C DX34 DISTANCE BETWEEN POINTS THREE AND FOUR

C DX12 DISTANCE BETWEEN POINTS 1 AND 2

C DS Y INCREMENT BETWEEN SPANS, (SPAN LENGTH)

C NC1, NC2 NUMBER OF COPD DIVISIONS IN AREA 1 AND 2

C DC1 CHORD LENGTH

C OS Y INCREMENT BETWEEN SPANS, (SPAN LENGTH)

C OXLE X INCREMENT A LONG LEADING EDGE

C OXTE X INCREMENT A LONG TRAILING EDGE

C TRANSFORM CORNER POINTS TO LOCAL SYSTEM, WORK IN LOCAL SYSTEM

C CALL TRANS (CORNX, CORNY, CORNZ, TRS, X, Y, Z)

C FIND CRATIO (CHORD TO SPAN RATI)

C ENTOR X DISTANCE BETWEEN POINTS THREE AND FOUR

C DS = (X3 - X1)*BETA

C CRATIO = (EACO*SQR(DS*DS + DC*DC)) / (DC*BETA)

C ASSUME CONSTANT SLOPE FOR ALL LINES

C DX34 = X4 - X3

01700010
01700020
01700030
01700040
01700050
01700060
01700070
01700080
01700090
01700100
01700110
01700120
01700130
01700140
01700150
01700160
01700170
01700180
01700190
01700200
01700210
01700220
01700230
01700240
01700250
01700260
01700270
01700280
01700290
01700300
01700310
01700320
01700330
01700340
01700350
01700360
01700370
01700380
01700390
01700400
01700410
01700420
01700430
01700440
01700450
01700460
01700470
01700480
01700490
01700500
01700510
01700520
01700530
01700540
01700550
01700560

-6.44-
LINE 57  
DX12 = X2 - X1  
LINE 58  
DS = (Y3-Y1)/NSP  
LINE 59  
DC = CRATI0*DS  
LINE 60  
DXLE = (X3-X1)/NSP  
LINE 61  
DXTE = (X4-X2)/NSP  
LINE 62  
COMPUTE NC1, DC1 FOR AREA 1  
LINE 63  
LINE 64  
NC1=1  
LINE 65  
DC1 = AMAX1(DX12, DX34)  
LINE 66  
IF( DC1 .LT. DC ) GO TO 11  
LINE 67  
NC1 = DC1/DC  
LINE 68  
DC1 = DC1/NC1  
LINE 69  
LINE 70  
LINE 71  
LINE 72  
LINE 73  
LINE 74  
LINE 75  
LINE 76  
LINE 77  
LINE 78  
LINE 79  
LINE 80  
LINE 81  
LINE 82  
LINE 83  
LINE 84  
LINE 85  
LINE 86  
LINE 87  
LINE 88  
LINE 89  
LINE 90  
LINE 91  
LINE 92  
LINE 93  
LINE 94  
LINE 95  
LINE 96  
LINE 97  
LINE 98  
LINE 99  
LINE 100  
LINE 101  
LINE 102  
LINE 103  
LINE 104  
LINE 105  
LINE 106  
LINE 107  
LINE 108  
LINE 109  
LINE 110  
LINE 111  
LINE 112  
LINE 113  
LINE 114  
LINE 115  
LINE 116  
LINE 117  
LINE 118  
GO TO 60

-6.45-
LINE 119 20 CONTINUE
LINE 120 C
LINE 121 60 CONTINUE
LINE 122 MCH(I) = NE & NETSV
LINE 123 C
LINE 124 100 CONTINUE
LINE 125 C
LINE 126 C UPDATE COUNTER OF TOTAL ELEMENTS FOR STRUCTURE
LINE 127 NETSV = NETSV & NE
LINE 128 RETURN
LINE 129 END
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, Z, TRS, NNCH, SW, XLE)

PURPOSE
GENERATE MESH FOR 1 SECTION WITH CONTROL LINES

INPUT

CORNX, CORNY, CORNZ  X, Y, AND Z COORD. OF CORNER PTS 1, 2, 3, 4 IN GLOBAL SYSTEM

NSP  DESIRED NUMBER OF SPANS FOR SECTION (INPUT BY USER)

ICL  NUMBER OF CONTROL LINES IN SECTION (1 OR 2)

CL(1), CL(2)  X COORD. OF FIRST CONTROL LINE

CL(3), CL(4)  X COORD. OF SECOND CONTROL LINE

EARO  ELEMENT ASPECT RATIO

OUTPUT

XYZ  4 X AND 2 Y LOCAL COORD. FOR EACH ELEMENT OF SECTION (FROM TRANS)

Z  Z COORDINATE FOR ALL ELEMENTS IN SECTION (FROM TRANS)

TRS  2 X 2 TRANS MATRIX TO GC FROM LOCAL TO GLOBAL

NNCH  ELEMENT NUMBER (W.R.T. TOTAL STRUCTURE) OF LAST ELEMENT IN SECTION

NE  NUMBER OF ELEMENTS IN SECTION

SW  SPAN WIDTH

NCX  NUMBER OF CORNER DIVISIONS IN AREA 1, 2, AND 3

NCY  NUMBER OF CORNER DIVISIONS IN AREA 4

DXE  X INCREMENT ALONG LEADING EDGE

DXT  X INCREMENT ALONG TRAILING EDGE

DXC  X INCREMENT ALONG SECOND CONTROL LINE IF IT EXISTS

DXD  DISTANCE BETWEEN POINTS THREE AND FOUR

DX2  DISTANCE BETWEEN 1 AND 2 MINUS DX3

DATA TEST / .98 /

SECTION IS DIVIDED INTO 3 AREAS

AREA 1 BETWEEN LEADING EDGE AND 1ST CONTROL LINE

AREA 2 BETWEEN FIRST AND 2ND CONTROL LINES

AREA 3 BETWEEN CONTROL LINE AND TRAILING EDGE

AREA 2 DOES NOT EXIST IF THERE IS ONLY ONE CONTROL LINE

NOTE THAT ELEMENT IS NUMBERED CLOCKWISE WHILE SECTION IS COUNTERCLOCKWISE

CALL TRANS(CORNX, CORNY, CORNZ, TRS, X, Y, Z)
LINE 57 C FIND CRATIO (CHORD TO SPAN RATIO)
LINE 58 C
LINE 59 C
LINE 60 D = Y3 - Y1
LINE 61 DS = (X3 - X1)*BETA
LINE 62 CRATIO = (DC*SQRT( DS*DS + DC*DC ) / (DC*BETA))
LINE 63 C
LINE 64 C ASSUME CONSTANT SLOPE FOR ALL LINES
LINE 65 C
LINE 66 D = X4 - X3
LINE 67 DX2 = X2 - X1 - DX34
LINE 68 DS = (Y3 - Y1)/NSP
LINE 69 DC = CRATIO*DS
LINE 70 D = (X3 - X1)/NSP
LINE 71 D = (X4 - X2)/NSP
LINE 72 C FIND INFORMATION FOR AREA BETWEEN LEADING EDGE AND 1ST CL
LINE 73 C
LINE 74 C
LINE 75 CL1 = CL1(1)
LINE 76 CL2 = CL1(2)
LINE 77 D = (CL2 - CL1)/NSP
LINE 78 NCI = 1
LINE 79 DC1 = AMAX1( CL1 - X1, CL2 - X3 )
LINE 80 IF( DC1 .LT. DC ) GO TO 11
LINE 81 NCI = DC1/DC
LINE 82 DC1 = DC1/NCI
LINE 83 11 CONTINUE
LINE 84 C
LINE 85 C FIND INFO FOR AREA BETWEEN CONTROL LINES
LINE 86 C
LINE 87 NC2 = 0
LINE 88 IFI TCL .EQ. 1 | GO TO 14
LINE 89 C
LINE 90 CL3 = CL1(3)
LINE 91 CL4 = CL1(4)
LINE 92 D = (CL4 - CL3)/NSP
LINE 93 C FIND DC3 FOR AREA BETWEEN SECOND CONTROL LINE AND TRAILING EDGE
LINE 94 DC3 = AMAX1( X2 - CL3, X4 - CL4 )
LINE 95 NCI = 1
LINE 96 DC3 = AMAX1( CL3 - CL1, CL4 - CL2 )
LINE 97 IFI DC2 .LT. DC ) GO TO 16
LINE 98 NC2 = DC2/DC
LINE 99 DC2 = DC2/NC2
LINE 100 GO TO 16
LINE 101 C
LINE 102 C
LINE 103 14 CONTINUE
LINE 104 C FIND DC3 FOR AREA BETWEEN FIRST CONTROL LINE AND TRAILING EDGE
LINE 105 DC3 = AMAX1( X2 - CL1, X4 - CL2 )
LINE 106 C
LINE 107 C 16 CONTINUE
LINE 108 C
LINE 109 NCI = 1
LINE 110 IFI DC3 .LT. DC ) GO TO 18
LINE 111 NC3 = DC3/DC
LINE 112 DC3 = DC3/NC3
LINE 113 18 CONTINUE
LINE 114 C
LINE 115 C
LINE 116 C
LINE 117 C LOOP ON SPANS
LINE 118 C NE = 0

-6.48-
LINE 119 C       DO 100 I=I,NSP
LINE 120       100 I=I,NSP
LINE 121 C       FIND THE Y COORD. FOR THIS SPAN (AY AND BY)
LINE 122 C       BY = Y1 & I*DS
LINE 123       123
LINE 124 C       AY = BY - DS
LINE 125 C       FIND AX, BX, X COORD ON LEADING EDGE ON BOTH SIDES OF SPAN
LINE 126       126
LINE 127       127
LINE 128       128
LINE 129 C       SW(I) = DS
LINE 130 C       XLE(I) = 0.5*(AX+BX)
LINE 131       131
LINE 132       132
LINE 133 C       FIND CL1, BCL1 X COORD ON FIRST CONTROL LINE
LINE 134       134
LINE 135       135
LINE 136 C       AX = BX - DXLE
LINE 137       137
LINE 138       138
LINE 139 C       AX = BX - DXLE
LINE 140       140
LINE 141 C       DXM = DTEST*DC3
LINE 142       142
LINE 143       143
LINE 144 C       LOOP ON NC1 FOR AREA 1
LINE 145 C       DO 20 J1 = 1,NC1
LINE 146       146
LINE 147 C       NJ = 6*NE
LINE 148       148
LINE 149       149
LINE 150       150
LINE 151       151
LINE 152       152
LINE 153       153
LINE 154       154
LINE 155       155
LINE 156       156
LINE 157       157
LINE 158       158
LINE 159       159
LINE 160       160
LINE 161 C       CHECK TO SEE IF WE HAVE REACHED THE FIRST CONTROL LINE
LINE 162       162
LINE 163       163
LINE 164       164
LINE 165       165
LINE 166       166
LINE 167       167
LINE 168 C       CONTINUE
LINE 169       169
LINE 170 C       20 CONTINUE
LINE 171 C       25 CONTINUE
LINE 172 C       IF(ICL .EQ. 1) GO TO 40
LINE 173       173
LINE 174 C       SECTION FOR AREA 2
LINE 175       175
LINE 176 C       FIND ACL2, BCL2 X COORD ON SECOND CONTROL LINE
LINE 177 C       BCL2 = CL3 & I*DXC2
LINE 178       178
LINE 179       179
LINE 180 C       ACL2 = BCL2 - DXC2

-6.49-
LINE 181  TD2 = DTEST * DC2
LINE 182  TC2 = BCL2 - TD2
LINE 183  TD2 = ACL2 - TD2
LINE 184  C LOOP ON NC2 FOR AREA 2
LINE 185  DO 30 J1 = 1, NC2
LINE 186  C
LINE 187  C
LINE 188  NJ = 6 * NE
LINE 189  NE = NE + 1
LINE 190  XYZ(NJ+5) = AY
LINE 191  XYZ(NJ+6) = AY
LINE 192  XD = J1 * DC2
LINE 193  XC = BCL1 & XD
LINE 194  XD = ACL1 & XD
LINE 195  XA = XD - DC2
LINE 196  XB = XC - DC2
LINE 197  XYZ(NJ+1) = XA
LINE 198  XYZ(NJ+2) = XB
LINE 199  XYZ(NJ+3) = XC
LINE 200  XYZ(NJ+4) = XD
LINE 201  C CHECK TO SEE IF WE HAVE REACHED SECOND CONTROL LINE
LINE 202  IF(XC .GT. BCL2 .OR. XD .GT. ACL2) GO TO 29
LINE 203  GO TO 30
LINE 204  29 CONTINUE
LINE 205  XYZ(NJ+3) = BCL2
LINE 206  XYZ(NJ+4) = ACL2
LINE 207  GO TO 45
LINE 208  30 CONTINUE
LINE 209  GO TO 45
LINE 210  40 CONTINUE
LINE 211  40 CONTINUE
LINE 212  C THIS SECTION IS REACHED IFF THERE WAS ONLY 1 CONTROL LINE
LINE 213  ACL1, BCL1 ARE PUT INTO ACL2, BCL2 AND AREA 3 IS ALWAYS TREATED
LINE 214  AS THE PART BETWEEN 2ND CONTROL LINE AND LEADING EDGE
LINE 215  ACL2 = ACL1
LINE 216  BCL2 = BCL1
LINE 217  45 CONTINUE
LINE 218  LOOP ON NC3 FOR AREA 3
LINE 219  DO 50 J1 = 1, NC3
LINE 220  C
LINE 221  C
LINE 222  C
LINE 223  NJ = 6 * NE
LINE 224  NE = NE + 1
LINE 225  XYZ(NJ+5) = AY
LINE 226  XYZ(NJ+6) = AY
LINE 227  XYZ(NJ+7) = AY
LINE 228  XYZ(NJ+8) = AY
LINE 229  XD = J1 * DC3
LINE 230  XC = BCL2 & XD
LINE 231  XD = ACL2 & XD
LINE 232  XA = XD - DC3
LINE 233  XB = XC - DC3
LINE 234  XYZ(NJ+1) = XA
LINE 235  XYZ(NJ+2) = XB
LINE 236  XYZ(NJ+3) = XC
LINE 237  XYZ(NJ+4) = XD
LINE 238  C CHECK TO SEE IF WE REACHED TAILING EDGE (CX, DX)
LINE 239  IF(XC .GT. CX .OR. XD .GT. DX) GO TO 49
LINE 240  GO TO 50
LINE 241  49 CONTINUE
LINE 242  XYZ(NJ+3) = CX
LINE 243  
LINE 244  
LINE 245  
LINE 246  
LINE 247  
LINE 248  
LINE 249  
LINE 250  
LINE 251  
LINE 252  
LINE 253  
LINE 254  
LINE 255  

XYZ(V) = DX
GO TO 60
50 CONTINUE
C
60 CONTINUE
N(NCH(i)) = NE & NETSV
C
100 CONTINUE
C
UPDATE COUNTER OF TOTAL ELEMENTS FOR STRUCTURE
NETSV = NETSV & NE
RETURN
END
DISPLAY...FILE

DECK 19

LINE 1  SUBROUTINE MFUN ( JMODE) 01900010
LINE 2  COMMON /MODEF/ EM(10), DM(10), X, Y, J1, J2 01900020
LINE 3  DIMENSION CTABLE(6,20), IFTABL(2,20) 01900030
LINE 4  COMMON /MCOM/ CTABLE, IFTABL, NTABL, NEQF 01900040
LINE 5  DO 20 1H = 1, JMODE 01900050
LINE 6  EM(1H) = 0.0 01900060
LINE 7  DM(1H) = 0.0 01900070
LINE 8  DO 100 I = 1, NEQF 01900080
LINE 9   IFM = IFTABL (1, I) 01900090
LINE 10  IFM = IFTABL (2, I) 01900100
LINE 11  IF( IFM.EQ.J1) .AND. ( IFM.EQ.1H) GO TO 90 01900110
LINE 12  GO TO 100 01900120
LINE 13  90 CO = CTABLE (1, I) 01900130
LINE 14  CX = CTABLE (2, I) 01900140
LINE 15  CY = CTABLE (3, I) 01900150
LINE 16  CXY= CTABLE (4, I) 01900160
LINE 17  CX2= CTABLE (5, I) 01900170
LINE 18  CY2= CTABLE (6, I) 01900180
LINE 19  EM(IH) = CO & CX * X & CXY * Y & CX2 * X**2 & CY2 * Y**2 01900190
LINE 20  DM(IH) = CX & CXY * Y & 2.0 * CX2 * X 01900200
LINE 21  GO TO 200 01900210
LINE 22  100 CONTINUE 01900220
LINE 23  200 CONTINUE 01900230
LINE 24  RETURN 01900240
LINE 25  END 01900250
DISPLAY...FILE DECK 20

LINE 1  SUBRUTINE MERR(IERR, MXNO, NOIN,16) 02000010
LINE 2  WRITE (16,700) 02000020
LINE 3  IF(IERR .EQ. 610) WRITE(16,710) MXNC, NOIN 02000030
LINE 4  IF(IERR .EQ. 620) WRITE(16,720) MXNC, NOIN 02000040
LINE 5  IF(IERR .EQ. 630) WRITE(16,730) MXNC, NOIN 02000050
LINE 6  IF(IERR .EQ. 640) WRITE(16,740) MXNC, NOIN 02000060
LINE 7  IF(IERR .EQ. 650) WRITE(16,750) MXNC, NOIN 02000070
LINE 8  IF(IERR .EQ. 660) WRITE(16,760) MXNC, NOIN 02000080
LINE 9  IF(IERR .EQ. 670) WRITE(16,770) MXNC, NOIN 02000090
LINE 10  IF(IERR .EQ. 680) WRITE(16,780) MXNC, NOIN 02000100
LINE 11  700 FORMAT(1HO ///8X, 42HERROR IN INPUT MAXIMUM ALLOWED EXCEEDED) 02000110
LINE 12  710 FORMAT(1HO / 8X, 20HMAXIMUM NO. WINGS = 113, 6X, 4H NO.) 02000120
LINE 13  1 , 6H INPUT = 113 ) 02000130
LINE 14  720 FORMAT(1HO / 8X,30HMAXIMUM NO. SECTIONS / WING = 111, 6X, 4H NO.) 02000140
LINE 15  1 , 6H INPUT = 113 ) 02000150
LINE 16  730 FORMAT(1HO / 8X,20HMAXIMUM NO. MYPES = 113, 6X, 4H NO.) 02000160
LINE 17  1 , 6H INPUT = 113 ) 02000170
LINE 18  740 FORMAT(1HO / 8X,20HMAXIMUM NO. SPANS = 113, 6X, 4H NO.) 02000180
LINE 19  1 , 6H INPUT = 113 ) 02000190
LINE 20  750 FORMAT(1HO / 8X,20HMAXIMUM NO. CORNER POINTS = 111, 6X, 4H NO.) 02000200
LINE 21  1 , 6H INPUT = 113 ) 02000210
LINE 22  760 FORMAT(1HO / 8X,20HMAXIMUM NO. ELEMENTS / SPAN = 112, 6X, 4H NO.) 02000220
LINE 23  1 , 6H INPUT = 113 ) 02000230
LINE 24  770 FORMAT(1HO / 8X,30HMAXIMUM NO. POINTS ON SPLINE DATA = 113, 6X, 4H NO.) 02000240
LINE 25  1 , 6H INPUT = 113 ) 02000250
LINE 26  780 FORMAT(1HO / 8X,36HMAXIMUM NO. INPUTS OF SPLINE DATA = 113, 6X, 4H NO.) 02000260
LINE 27  1 4H NO., 6H INPUT = 113 ) 02000270
LINE 28  RETURN 02000280
LINE 29  END 02000290

/ END READ

END OF JOB.

39.6 SEC. USED .012 HRS. CHARGED 49.962 HRS. REMAINING
SUBROUTINE OUTPUT( MACH, FREQ, JMODEL, NWING, NS, NISP, SW, DLSPAN, PMSPAN, XP, INDXI, PRES, IPW, LMAX, NNCH, NE, TITLE, PHIW, NISP )

MACH   MACH NUMBER
FREQ   FREQUENCY
JMODEL  MODE NUMBER
NWING  WING NUMBER
NS   NUMBER OF SECTIONS IN NWING
NISP  NUMBER OF SPANS IN EACH SECTION OF NWING (ARRAY)
DLSPAN  LIFT/UNIT SPAN FOR EACH SPAN OF NWING (ARRAY)
PMSPAN  PITCHING MOMENT/UNIT SPAN FOR EACH SPAN (ARRAY)

LINE   XP
INDEX   ELEMENT NUMBER OF FIRST ELEMENT OF NWING
PRES   ELEMENT PRESSURES FOR NWING (ARRAY)
IPW   PRINT CONTROL FOR NWING
IPW=1 LIFT AND PRESSURE MOMENTS/UNIT SPAN TO BE PRINTED
IPW=2 ELEMENT PRESSURES TO BE PRINTED
IPW=3 BOTH OF THE ABOVE TO BE PRINTED
IPW=0 NONE OF THE ABOVE TO BE PRINTED
SW   SPAN WIDTH (ARRAY)
LINE   CURRENT LINE ON OUTPUT PAGE
LMAX  MAXIMUM LINE ON A PRINTED PAGE
NNCH  ARRAY CONTAINING ELEMENT NUMBER OF THE LAST ELEMENT OF NWING
NE   ARRAY CONTAINING THE NUMBER OF ELEMENTS IN EACH SECTION OF NWING

REAL MACH,
COMPLEX DLSPAN(1), PMSPAN(1), PRES(1)

DO 15 K SPCT = 1, NISP
15 CONTINUE

IF ( LMAX - LINE .GT. 14 ) GO TO 25
GO TO 35

WRITE (16, 2015) TITLE, FREQ
LINE = 6
WRITE LIFT AND PRESSURE MOMENT TABLE

-- ORIGINAL PAGE 2 OF POOR QUALITY --
LINE 57  30  CONTINUE
LINE 58  WRITE ( 16, 201 )
LINE 59  LINE = LINE & 12
LINE 60  35  CONTINUE
LINE 61  WRITE ( 16, 2001 ) JWIN
LINE 62  C
LINE 63  K = 1
LINE 64  ISPCT = KSPCT
LINE 65  WRITE ( 16, 2002 ) JMODE, FREQ, MACH
LINE 66  WRITE ( 16, 2003 ) XP
LINE 67  C
LINE 68  LOOP ON SECTIONS
LINE 69  DO 120 ISECT = 1, NS
LINE 70  NSPAN = NSP(ISECT)
LINE 71  LOOP ON SPANS OF SECTION
LINE 72  DO 115 ISPAN = 1, NSPAN
LINE 73  ISPCT = ISPCT & 1
LINE 74  K = K & 1
LINE 75  115 CONTINUE
LINE 76  LINE = LINE & NSPAN
LINE 77  120 CONTINUE
LINE 78  C
LINE 79  LINE = MOD(LINE, LMAX)
LINE 80  C
LINE 81  123 CONTINUE
LINE 82  IF ( IPW .EQ. 1 ) RETURN
LINE 83  C
LINE 84  WRITE ELEMENT PRESSURE TABLE
LINE 85  C
LINE 86  C
LINE 87  C
LINE 88  C
LINE 89  C
LINE 90  125 NEWJ = NEWJ & NE(I)
LINE 91  INDXR = INDXR & NEWJ - 1
LINE 92  C
LINE 93  IF ( LMAX .GT. 14 ) GO TO 130
LINE 94  WRITE ( 16, 2020 )
LINE 95  LINE = 7
LINE 96  GO TO 135
LINE 97  130 CONTINUE
LINE 98  WRITE ( 16, 201 )
LINE 99  LINE = LINE & 12
LINE 100  135 CONTINUE
LINE 101  WRITE ( 16, 2005 ) JWIN
LINE 102  C
LINE 103  WRITE ( 16, 2002 ) JMODE, FREQ, MACH
LINE 104  WRITE ( 16, 2009 )
LINE 105  INDXR = INDXR & 1
LINE 106  ISPCT = KSPCT
LINE 107  C
LINE 108  LOOP ON SECTIONS
LINE 109  DO 220 ISECT = 1, NS
LINE 110  NSPAN = NSP(ISECT)
LINE 111  LOOP ON SPANS OF SECTION
LINE 112  DO 215 ISPAN = 1, NSPAN
LINE 113  NEPS = NNCH(ISPCT) - INDXR & 1
LINE 114  WRITE ( 16, 2010 ) ISECT, ISPAN, PHIV(INDXR), PRES(INDXR)
LINE 115  INDXR = INDXR & 1
LINE 116  210 CONTINUE
LINE 117  ISPCT = ISPCT & 1
LINE 118  215 CONTINUE

LIN 119  220 CONTINUE
LIN 120  C        LINE = LINE & NEWJ
LIN 121  LINE = NE\$J/3 & LINE
LIN 122  IF( MOD(NEWJ,3) .GT. 0 ) LINE=LINE
LIN 123  LINE = MOD(LINE, LMAX )
LIN 124  C        RETURN
LIN 125  RETURN
LIN 126  2001 FORMAT( 46\&, 47H\&IFT AND PITCHING MOMENT PER UNIT SPAN FOR WING )
LIN 127  1NG, 1Z )
LIN 128  2002 FORMAT( 1H5, 5H, 11H\&ODE \&UMBER, 13 / 6X, 11H\&REQUENCY =, F7.4, / 02101280
LIN 129  1H, 22X, 13H\&MAC \&UMBER =, F7.4 )
LIN 130  2003 FORMAT( 5H, 3H \&CTION SPAN SPAN WIDTH, / 02101330
LIN 131  1H, 8X, 27H LIFT \&ER UNIT SPAN ----, / 02101380
LIN 132  12X, 38H\&ITCHING MOMENT PER UNIT SPAN ABOUT X=, E12.6 / )
LIN 133  2004 FORMAT( 5X, 217, 5X, E12.6, 2( 7X, E12.6, 3X, E12.6, 4X ) )
LIN 134  2005 FORMAT( 46X, 50HVELOCITY POTENTIALS AND ELEMENT \&RESSES FOR WING )
LIN 135  1H, 12 )
LIN 136  2007 FORMAT( 31H, 7\&LEMENT, 3X, 81H-1, 10H PRESSURE =, F12.6 )
LIN 137  2008 FORMAT( 31H, 6X, 17, 3X, E12.5, 3X, E12.6 )
LIN 138  2009 FORMAT( 6X, 27HSECTION SPAN ELEMENT, 11X )
LIN 139  1H, 27H VELOCITY POTENTIAL ----, 11X, 81H-1, 13H \&RESSURE )
LIN 140  12X, 2( 9H1H-1 )
LIN 141  2010 FORMAT( 6X, 217, 113, 2( 11X, E12.6, 3X, E12.6 ) )
LIN 142  2011 FORMAT( 1H1, 45X, 1444 // 46X, 11H\&REQUENCY =, F7.4//1X, 6512H 1 )
LIN 143  2012 FORMAT( 1H1 )
LIN 144  2021 FORMAT( // // )
LIN 145  END
LIN 146
-6.56-
DISPLAY FILE

LINE 1 SUBROUTINE OLTP2( I6,NWING,XP,TLM,JMODE,FREQ,MACH,LINE,LMAX ) 02200010
LINE 2 COMPLEX TL(4,10), TM(4,10) 02200020
LINE 3 REAL MACH 02200030
LINE 4 C TL TOTAL LIFT PER WING 02200040
LINE 5 C TM TOTAL MOMENT PER WING 02200050
LINE 6 C WRITE TABLE OF TOTAL LIFT AND PITCHING MOMENTS 02200060
LINE 7 C WRITE TABLE OF TOTAL LIFT AND PITCHING MOMENTS 02200070
LINE 8 C WRITE TABLE OF TOTAL LIFT AND PITCHING MOMENTS 02200080
LINE 9 C WRITE TABLE OF TOTAL LIFT AND PITCHING MOMENTS 02200090
LINE 10 IF( LMAX .GT. 14 ) GO TO 30 02200100
LINE 11 WRITE( I6, 2020 ) 02200110
LINE 12 WRITE( I6, 2020 ) 02200120
LINE 13 WRITE( I6, 2020 ) 02200130
LINE 14 WRITE( I6, 2020 ) 02200140
LINE 15 WRITE( I6, 2020 ) 02200150
LINE 16 WRITE( I6, 2020 ) 02200160
LINE 17 WRITE( I6, 2020 ) 02200170
LINE 18 WRITE( I6, 2020 ) 02200180
LINE 19 WRITE( I6, 2020 ) 02200190
LINE 20 WRITE( I6, 2020 ) 02200200
LINE 21 C DO 50 I=1,JMODE 02200210
LINE 22 WRITE( 16, 2020 ) (I,J, TL(J,I) , TM(J,I) , J=1,NWING) 02200220
LINE 23 C DO 50 I=1,JMODE 02200230
LINE 24 WRITE( 16, 2020 ) (I,J, TL(J,I) , TM(J,I) , J=1,NWING) 02200240
LINE 25 C LINE = LINE & NWING*JMODE 02200250
LINE 26 LINE = MOD( LINE, LMAX ) 02200260
LINE 27 C RETURN 02200270
LINE 28 C RETURN 02200280
LINE 29 C RETURN 02200290
LINE 30 2002 FORMAT( 6X, 11HFREQUENCY =, F7.4, 02200300
LINE 31 1 22X, 13HMACH NUMBER =, F7.4 / ) 02200310
LINE 32 2008 FORMAT( 4EX, 31HTOTAL LIFT AND PITCHING MOMENTS ) 02200320
LINE 33 2009 FORMAT( 15X,14.6X, 14, 5X, 2( 3X, E12.6 ), 8X, 2( 3X, E12.6 ) ) 02200330
LINE 34 2011 FORMAT( 15X, 4HMODE, 6X, 02200340
LINE 35 1 4HWING, 8X, 10(1H-), 6H LIFT, 1(1H-), 8X, 25HPITCHING MOMENT ABDU 02200350
LINE 36 14T X = ,E12.6 / ) 02200360
LINE 37 2020 FORMAT( 1H1 ) 02200370
LINE 38 2021 FORMAT( / / / / / ) 02200380
LINE 39 END 02200390

-6.57-
DISPLAY...FILE

DECK 23

LINE 1  SUBROUTINE OUTP3 ( 16, MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE ) 02300010
LINE 2  COMPLEX Q(10,10) 02300020
LINE 3  COMPLEX ROW(3) 02300030
LINE 4  REAL MACH 02300040
LINE 5  DIMENSION NST(3) 02300050
LINE 6  DIMENSION TITLE(14) 02300060
LINE 7  C WRITE TABLE PRINT OF GENERALIZED AERODYNAMIC COEFFICIENT MATRIX 02300070
LINE 8  C 02300080
LINE 9  C IF( LINE .NE. LMAX CL ) GO TO 25 02300090
LINE 10 WRITE ( 16, 2015 ) TITLE, FREQ 02300100
LINE 11 LINE = 6 02300110
LINE 12 GO TO 30 02300120
LINE 13 25 CONTINUE 02300130
LINE 14 IF( LMAX - LINE .GE. 13 ) JMODE ) GO TO 30 02300140
LINE 15 WRITE ( 16, 2020 ) 02300150
LINE 16 LINE = 8 02300160
LINE 17 GO TO 35 02300170
LINE 18 30 CONTINUE 02300180
LINE 19 LINE = LINE & 13 02300190
LINE 20 GO TO 35 02300200
LINE 21 35 CONTINUE 02300210
LINE 22 WRITE ( 16, 2010 ) FREQ, JMODE, MACH 02300220
LINE 23 NCL = 3 02300230
LINE 24 DO 100 J = 1, JMODE, 3 02300240
LINE 25 KK = JMODE - J & 1 02300250
LINE 26 IF( KK .LT. 3 ) NCL = KK 02300260
LINE 27 DO 50 K = 1, NCL 02300270
LINE 28 KO = K & J - 1 02300280
LINE 29 ROW( K ) = 0(I, KO) 02300290
LINE 30 WRITE ( 16, 2012 ) I, ( ROW( K ), K = 1, NCL ) 02300300
LINE 31 WRITE ( 16, 2022 ) 02300310
LINE 32 WRITE ( 16, 2023 ) NST(1) 02300320
LINE 33 WRITE ( 16, 2024 ) NST(2) 02300330
LINE 34 WRITE ( 16, 2025 ) NST(3) 02300340
LINE 35 WRITE ( 16, 2026 ) 02300350
LINE 36 WRITE ( 16, 2027 ) 02300360
LINE 37 WRITE ( 16, 2028 ) 02300370
LINE 38 WRITE ( 16, 2029 ) 02300380
LINE 39 WRITE ( 16, 2030 ) 02300390
LINE 40 WRITE ( 16, 2031 ) 02300400
LINE 41 WRITE ( 16, 2032 ) 02300410
LINE 42 WRITE ( 16, 2033 ) 02300420
LINE 43 WRITE ( 16, 2034 ) 02300430
LINE 44 WRITE ( 16, 2035 ) 02300440
LINE 45 WRITE ( 16, 2036 ) 02300450
LINE 46 WRITE ( 16, 2037 ) 02300460
LINE 47 WRITE ( 16, 2038 ) 02300470
LINE 48 WRITE ( 16, 2039 ) 02300480
LINE 49 WRITE ( 16, 2040 ) 02300490
LINE 50 WRITE ( 16, 2041 ) 02300500
LINE 51 WRITE ( 16, 2042 ) 02300510
LINE 52 WRITE ( 16, 2043 ) 02300520
LINE 53 WRITE ( 16, 2044 ) 02300530
LINE 54 RETURN 02300540
LINE 55 FORMAT ( 10X, 11HFREQUENCY =, F7.4, 18X, 02300550
LINE 56 1 56MGENERALIZED AERODYNAMIC COEFFICIENTS IN AGARD DEFINITION / 02300560

-6.58-
LINE 57
2 10X, 13, 6H MODES, 27X, 13H MACH NUMBER =, F7.4

LINE 58
2011 FORMAT( / 10X, 12H DISPLACEMENT / 13X, 4H MODE, 8X,
LINE 59
1 18H ---- PRESSURE MODE, 13, 5H ---- )

LINE 60
2012 FORMAT( 15X, 13, 3( 8X, F12.6, 2X, E12.6 )

LINE 61
2015 FORMAT( 1H1, 45X, 14A4 // 46X, 11H FREQUENCY =, F7.4/1X, 6512H L
LINE 62
2020 FORMAT( 1H1 )

LINE 63
2021 FORMAT( )

LINE 64
2022 FORMAT( )

LINE 65
2023 FORMAT( 1HE, 63X, 18H ---- PRESSURE MODE, 13, 5H ---- )

LINE 66
2024 FORMAT( 1HE, 97X, 18H ---- PRESSURE MODE, 13, 5H ---- )

LINE 67
END
SUBROUTINE PHI (XLE, XCN, DPDX, NEPS, PHI, A, CXP, PHIW )

COMPLEX CL, CX, XDIFS

COMPLEX EXI, EAXI, AXI, AXIPI, SUM

COMPLEX PHI(1), DPDX(1), A, CXP(1)

COMPLEX PHIW(1)

LOGICAL WAKEI, WAKE, WAKENZ, TREDGE

DIMENSION XCN(1)

COMMON/WAKEUP/, WAKENZ(12), TREDGE, WAKEI, WAKE, WAKENZ

DATA C1 / (1.0, 0.0) /

IF ( EPS.EQ. 1 ) GO TO 40

XIP1 = CMPLX( XCN(2), 0.0 )

CX = CMPLX( (XCN(1) - XCN(2) ), 0.0 )

B1 = ( DPDX(1) - DPDX(2) ) / CX

B0 = DPDX(2) - B1*XIP1

IF ( CABS(A) .LT. 1.0E-20 ) GO TO 23

XI = CMPLX( XLE, 0.0 )

XIP1 = CMPLX( XCN(1), 0.0 )

AA = A * A

AXI = A*XI

AXIPI = AXIPI

EAXI = CXP( AXI )

EAXIPI = CXP( AXI )

EX = C1 / EAXIPI

SUM = (B1/AA)* (AXIPI - C1)*EAXIPI - (AXI - C1)*EAXI

1 = (B0/A)* (EAXIPI - EAXI )

PHI(1) = SUM*EX
LINE 57  10 CONTINUE
LINE 58  C
LINE 59  GO TO 52
LINE 60    20 CONTINUE
LINE 61  CX = CMPLXI ( XLE - XCEN(I1) ), 0.0 )
LINE 62  XDIFS = CMPLXI 0.5*(XLE+XLE-XCEN(I1)*XCEN(I1)), 0.0 )
LINE 63  SUM = 80*CX & B1*XDIFS
LINE 64  PHI(I1) = SUM
LINE 65  DO 30 IP1=2,NEPS
LINE 66    I = IP1 - 1
LINE 67  CX = CMPLXI ( XCEN(I1)-XCEN(IP1) ), 0.0 )
LINE 68  B1 = ( DPDX(IP1) - DPDX(IP1) ) / CX
LINE 69  B0 = DPDX(IP1) - B1*IP1
LINE 70  XDIFS = CMPLXI 0.5*(XCEN(I1)*XCEN(I1)-XCEN(IP1)*XCEN(IP1)), 0.0 )
LINE 71  SUM = SUM & B0*CX & B1*XDIFS
LINE 72  PHI(IP1) = SUM
LINE 73    30 CONTINUE
LINE 74  GO TO 52
LINE 75  C
LINE 76    40 CONTINUE
LINE 77  C
LINE 78  C SECTION FOR ONLY ONE ELEMENT PER SPAN
LINE 79  C
LINE 80  CX = CMPLXI ( XCEN(1)-XLE, 0.0 )
LINE 81  IF( CABSA(1) .LT. 1.E-20 ) GO TO 50
LINE 82  PHI(1) = ( DPDX(1)/A )*( C1 - CEXP(-A*CX ) )
LINE 83  GO TO 52
LINE 84    50 PHI(1) = DPDX(1)*CX
LINE 85    52 ICNVGW = 0
LINE 86    55 IF( ICNVGW =0 ) ICNVGW =0
LINE 87    58 ICNVGW = 1 MEANS AT LEAST ONE PHI HAS NOT CONVERGED FOR SPAN
LINE 88    60 ICNVGW=0 MEANS CONVERGENCE HAS OCCURRED FOR WAKE ELEMENTS
LINE 89    62 ICNVGW = 0 MEANS CONVERGENCE HAS OCCURRED FOR WAKE ELEMENTS
LINE 90    64 ITM = WAKE CONVG. ITERATION COUNTER INITIALLY SET =0
LINE 91    66 ITM = 0 IF WAKE ELEMENT EFFECT IS NOT DESIRED
LINE 92    68 ITM = 1 IF WAKE ELEMENT EFFECT IS DESIRED
LINE 93    70 IF( ITM .EQ. 0 ) GO TO 58
LINE 94    72 ICNVGW = IRELE( PHI, PHIW, NEPS)
LINE 95    74 IF( ICNVGW .EQ.0 ) JCNVGW = JCNVGW & NEPS
LINE 96    76 CONTINUE
LINE 97    78 DD 60 IP1=1,NEPS
LINE 98    80 IF( ITM .EQ.0 ) GO TO 60
LINE 99    82 IF( ITM .EQ.0 ) GO TO 60
LINE 100    84 ICNVGW = IRELE( PHI(IP1), PHIW(IP1), ICNVGW )
LINE 101    86 IF( ICNVGW .EQ.0 ) JCNVGW = JCNVGW & 1
LINE 102    88 PHIW(IP1) = PHI(IP1)
LINE 103    90 RETURN
LINE 104    92 END
DISPLAY...FILE

DECK 25

LINE 1  SUBROUTINE PLOT(X,Y,IPEN)
LINE 2   IF( IPEN .EQ. -23) GO TO 100
LINE 3   IF( IPEN .EQ. -23) IPEN = -3
LINE 4   CALL CALPLOT(X,Y,IPEN)
LINE 5   RETURN
LINE 6  100 SX = X & 2.0
LINE 7   CALL INFRAME(SX,Y)
LINE 8   RETURN
LINE 9   END

02500010
02500020
02500030
02500040
02500050
02500060
02500070
02500080
02500090
SUBROUTINE PLOTGD ( NE, XYZ, NNCH, NSP, NSECT, TITLE, JWING, MACH )

REAL MACH

DIMENSION NE(11, ZSECT(1) 9NSP(1)

DIMENSION PH(7), PV(7)

DIMENSION XYZ(1), NNCH(1), TITLE(1)

DATA FGDX, FBDY / 15.0, 10.0 /

DATA HT, HTW, HHT / .07, .05202, .035 /

DATA NETSV / 1 /

IFS J WING .EQ. 1 NETSV = 1

NEPWT = 0 KINDEX = 1 KFIRST = 1 KLAST = 0

DO 100 IS=1,NSECT

NEPS = NE(IS) K = KINDEX

XMAX = XYZ(K1) XMIN = XMAX

YMAX = XYZ(K4) YMIN = YMAX

DO 10 I=1,NEPS

XMIN = AMIN1(XMIN, XYZ(K)) XMAX = AMAX1(XMIN, XYZ(K))

K = K 6 I. DO 6 J=1,2

YMIN = AMIN1(YMIN, XYZ(K)) YMAX = AMAX1(YMAX, XYZ(K))

K = KG1 CONTINUE

FGDX = MAXIMUM X PLOT SIZE

FGDY = MAXIMUM Y PLOT SIZE

SFH = (XMAX—XMIN) /FGDX

SFV = (YMAX—YMIN)/FGDY

SF = AMAX1(SFH, SFV)

TMXX = (XMAX—XMIN)/SF

TMYY = (YMAX—YMIN)/SF

PH(6) = XMIN

PH(7) = SF

PV(6) = YMIN

PV(7) = SF

WRITE(6, 1002) XMAX, XMIN, YMAX, YMIN,

1 SFH, SFV, TMXX, TMYY, SF

DRAW LINE TO MARK ORIGINAL ORIGIN, LIFT PEN, AND REORIGIN

CALL PLOT( 0.25, 0.0, 2 )

CALL PLOT( 0.75, 0.0, 3 )

CALL PLOT( 2.0, 0.25, 23 )
LINE 57 C SECTION TO PUT TITLE AND HEADING ON PLOT  02600570
LINE 58 C  02600580
LINE 59 C  02600590
LINE 60 C Y = 0.25  02600600
LINE 61 DO 12 I=1,14  02600610
LINE 62 CALL SYMBOL ( -1.5, Y, 0.21, TITLE(I), 90.0, 4 )  02600620
LINE 63  02600630
LINE 64 C  02600640
LINE 65 CALL SYMBOL ( -1.15, 0.25, 0.21, 14HASPECT RATIO =, 90.0, 14 )  02600650
LINE 66 CALL NUMBER ( -1.15, 2.98, 0.21, AR, 90.0, 3 )  02600660
LINE 67 CALL SYMBOL ( -1.15, 4.7, 0.21, 13MACH NUMBER =, 90.0, 13 )  02600670
LINE 68 CALL NUMBER ( -1.15, 7.10, 0.21, MACH, 90.0, 3 )  02600680
LINE 69 CALL SYMBOL ( -0.80, 0.25, 0.21, 14SECTION NUMBER, 90.0, 14 )  02600690
LINE 70 FPN = FLOAT(I)  02600700
LINE 71 CALL NUMBER ( -0.80, 2.98, 0.21, FPN, 90.0, -1 )  02600710
LINE 72 CALL SYMBOL ( -0.80, 3.52, 0.21, THOSE WING, 90.0, 7 )  02600720
LINE 73 FPN = FLOAT(JWING)  02600730
LINE 74 CALL NUMBER ( -0.80, 4.97, 0.21, FPN, 90.0, -1 )  02600740
LINE 75 CALL SYMBOL ( -0.80, 6.33, 0.21, 3HZ =, 90.0, 3 )  02600750
LINE 76 FPN = ZSECTION(I)  02600760
LINE 77 CALL NUMBER ( -0.80, 7.06, 0.21, FPN, 90.0, 3 )  02600770
LINE 78 C  02600780
LINE 79 C SECTION TO PLOT ELEMENTS  02600790
LINE 80 C  02600800
LINE 81 K = KINDEX  02600810
LINE 82 DO 20 I=1,NEPS  02600820
LINE 83 DO 19 J=1,4  02600830
LINE 84 PH(J) = XYZ(K)  02600840
LINE 85 IF (J .GT. 2) GO TO 19  02600850
LINE 86 PV(J) = XYZ(K+4)  02600860
LINE 87  02600870
LINE 88  02600880
LINE 89  02600890
LINE 90  02600900
LINE 91  02600910
LINE 92 PV(5) = PV(1)  02600920
LINE 93 C WRITE(6,1003) I, (PH(I,J),I,J=1,5), (PV(I,J),I,J=1,5)  02600930
LINE 94 CALL LINE( PH, PV, 5, 1, 0, 0, 0)  02600940
LINE 95 20 CONTINUE  02600950
LINE 96 C  02600960
LINE 97 C SECTION TO LABEL ELEMENTS ON THE TRAILING EDGE  02600970
LINE 98 C  02600980
LINE 99 KLAST = NSP(IS) & KLAST  02600990
LINE 100 DO 30 I=KFIRST,KLAST  02600100
LINE 101 C NETSV IS THE NUMBER OF ELEMENTS IN THE PREVIOUS WINGS PLUS 1  02600101
LINE 102 J = 6*(NNCH(I)-NETSV) & I  02600102
LINE 103 SFH = 0.25*( XYZ(J) & XYZ(J+1) & XYZ(J+2) & XYZ(J+3) )  02600103
LINE 104 SFV = 0.53*( XYZ(J+4) & XYZ(J+5) )  02600104
LINE 105 SFH = (SFH-XMIN)/SF & HH  02600105
LINE 106 SFV = (SFV-YMIN)/SF & HH  02600106
LINE 107 C (SFH, SFV) ARE COORDINATES OF CENTER OF ELEMENT  02600107
LINE 108 C TD CENTER CHARACTER, MOVE OFF CENTER BY (HTW,HHT)  02600108
LINE 109 GP = FLOAT(I) ANCH(I) )  02600109
LINE 110 CALL NUMBER ( SFH, SFV, HT, GP, 90.0, -1 )  02600110
LINE 111 30 CONTINUE  02600111
LINE 112 KFIRST = KFIRST & NSP(IS)  06.120
LINE 113 C  026001130
LINE 114 C SECTION TO REORIGIN FOR NEXT PLOT AND RUMP BLOCK ADDRESS BY  026001140
LINE 115 C  026001150
LINE 116 C  026001160
LINE 117 SX = TMXX & 3.0  026001170
LINE 118 CALL PLOT(SX, -.25, -23)  026001180

-6.64-
LINE 119 C UPDATE COUNTER OF ELEMENTS IN WING
LINE 120 C NEPWT = NEPWT + NEPS
LINE 121 KINDEX = KINDEX + 6*NEPS
LINE 122 C END LOOP ON SECTIONS
LINE 123 C UPDATE ELEMENT COUNTER
LINE 124 C NETSV IS THE NUMBER OF ELEMENTS IN THE PREVIOUS WINGS PLUS 1
LINE 125 NETSV = NETSV + NEPWT
LINE 126 C RETURN
LINE 127 C FORMAT( 15, 2(13X, 5E14.4) )
LINE 128 C END
LINE 1  SUBROUTINE PTGRID(I6, TITLE, MACH, WING, NSECT, NSP, NNCH, XYZ, U2700010
LINE 2  1 I, ZSECT, REFLEN, BEETA, XCEN) U2700020
LINE 3  REAL MACH U2700030
LINE 4  DIMENSION NNCH(1), NSP(3,1) U2700040
LINE 5  DIMENSION TITLE(14), NSECT(1), XYZ(11), ZSECT(3,4) U2700050
LINE 6  DIMENSION XCEN(1) U2700060
LINE 7  DIMENSION XY(8) U2700070
LINE 8  BL = BEETA*REFLEN U2700080
LINE 9  WRITE(16,3000) TITLE, MACH U2700090
LINE 10  WRITE(16,3001) BL U2700100
LINE 11  KSPAN = 0 U2700110
LINE 12  K = 0 U2700120
LINE 13  N1 = 1 U2700130
LINE 14  DO 100 J=1,NWING U2700140
LINE 15  NS = NSECT(J) U2700150
LINE 16  DO 95 J=1,NS U2700160
LINE 17  WRITE(16,3001) J, ZSECT(I,J) U2700170
LINE 18  NSPAN = NSP(I,J) U2700180
LINE 19  DO 90 ISPAN =1,NSPAN U2700190
LINE 20  KSPAN = KSPAN + 1 U2700200
LINE 21  N1 = N1 + 6 U2700210
LINE 22  80 CONTINUE U2700220
LINE 23  85 CONTINUE U2700230
LINE 24  90 CONTINUE U2700240
LINE 25  95 CONTINUE U2700250
LINE 26  100 CONTINUE U2700260
LINE 27  RETURN U2700270
LINE 28  3000 FORMAT(1H1/35X,14A4//35X,32HAERODYNAMIC GRID FOR MACH NJ.MBI:R, U2700280
LINE 29  F8.4 / ) U2700290
LINE 30  3001 FORMAT( / 6H WING, 12, 8H SECTION, 12 / 5H L =*2X,E11.5/, U2700300
LINE 31  46 1 14H SPAN ELEMENT, 4X, 2HX1, 12X, 2HX2, 12X, 2HX3, 12X, 2HX4, U2700310
LINE 32  47 2 12X, 2HY1, 12X, 2HY2, 13X, 2HXC, 12X, 2HYC / ) U2700320
LINE 33  48 3002 FORMAT(I5, 19, 1X, 6I 3X, E11.5, 4X, E11.5, 3X, E11.5) U2700330
LINE 34  3003 FORMAT( / 35X, 32HSTRUCTURAL REFERENCE SYSU2700340
LINE 35  50 ITEM JA / 35X, 32HRROTATED STRUCTURAL REFERENCE SYSTEM / ) U2700350
LINE 51  END U2700360

-6.66-
<table>
<thead>
<tr>
<th>LINE</th>
<th>CODE</th>
<th>TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUBROUTINE ROIETA (I13, DETADX, ETA, NET)</td>
<td>02800010</td>
</tr>
<tr>
<td>2</td>
<td>DIMENSION DETADX(NET), ETA(NET)</td>
<td>02800020</td>
</tr>
<tr>
<td>3</td>
<td>READ (I13)</td>
<td>02800030</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>02800040</td>
</tr>
<tr>
<td>5</td>
<td>RETURN</td>
<td>02800050</td>
</tr>
<tr>
<td>6</td>
<td>END</td>
<td>02800060</td>
</tr>
</tbody>
</table>

-6.67-
SUBROUTINE RMODE
   JMODE, N+1, NSECTION, NE, XCEN, XYZ, BEEA,
   REFLEN, NET, LAY, ID, ERR00K, ETA, DETAOX, 15, 16, 1X, 1Y, A, N+4, IERR)
   02900010
   L
gcnal ERROR
   02900020
   CETA AND DETAOX DIMENSIONED (NUMBER OF ELEMENTS, NO. MODE)
   02900040
   DIMENSION CFTABL(6,20), IFTABL(2,20)
   02900050
   DIMENSION ETAMNET, DETAOXNET
   02900060
   DIMENSION ID(1)
   02900070
   DIMENSION MODE(20)
   02900080
   DIMENSION MODEF4(16)
   02900090
   DIMENSION NCHECK(4,3)
   02900100
   DIMENSION NSECTION(1), NE(3), XCEN(1), XYZ(1)
   02900110
   DIMENSION NUMER(3), MODF(7), MODF2(13), MODF3(7)
   02900120
   COMM /MCDM/ CFTABL, IFTABL, NTABL, NEQF
   02900140
   COMMON /EMO/, /DMO/, X, Y, J1, J2
   02900150
   COMMON /XYSCAL/, XX, YY, XQ, YQ, DREF, NZ, ITEST
   02900160
   COMMON /ZFUNNY/, NIERF, B1, B2, B3, NFUNMX
   02900170
   DATA LBBLANK / 4H /
   02900190
   DATA LEND / 3HENI) / 02900200
   DATA LMW / 4HMW / 02900210
   DATA LMK / 4HRKG / 02900220
   DATA MODONE / 2H /
   02900230
   DATA MODEF1 / 3H(//, 1H1, 4H(24X, 4H(4XM, 4HMODE, 4H12, 3H(3)
   02900240
   DATA MODEF2 / 4H(10X, 4H 5, 4H(43X, 4HMNT, 1H1, 4H10X, 4H3H)
   02900250
   DATA MODF3 / 4H(11, 2H0, 1H1, 4H3X, 4H2E15, 4H61), 1H /
   02900260
   DATA MODEF4 / 4H(// 2, 4H0I, 1H1, 4H6H, 4HMODE, 2HS) /
   02900270
   DATA NUMBR / 120 /
   02900280
   DATA NUMBR / 1H1, 1H2, 1H3 /
   02900290
   X0 = 0.0
   02900300
   Y0 = 0.0
   02900310
   BREF = 1.0
   02900320
   JMODE = 0
   02900330
   WRITE(16, 1999) ID(1), ID(2), ID(3), ID(4)
   02900340
   1999 FORMAT (21X, TOTAL NUMBER M, )DES, 2IX, LI2/21X, MMODES /
   02900350
   132 HBED BY SPLINE OR POLYNOMIAL , 1X, 12I2/21X
   02900360
   213H RIGID MODES , 17X, 112 / 21X, 21HNUMBER OF POLYNOMIAL
   02900370
   310HEQUATIONS , 8X, 112 ///
   02900380
   JRGID = ID(1)
   02900400
   JTYPE = ID(4)
   02900420
   RMODE
   02900430
   I = I, NEQF
   02900500
   IF( LAB EQ. LRIG ) GO TO 215
   02900440
   JMODE = ID(2)
   02900450
   BREF = BEEA*REFLEN
   02900460
   HALFT = 0.5*REFLEN
   02900470
   IF( JMODE .GE. 10 ) MODF4(3) = NUMBR(2)
   02900480
   IF( JMODE .GE. 1 ) MODF4(6) = MODONE
   02900490
   IF( JTYPE .EQ. 0 ) GO TO 10
   02900500
   WRITE(16, 2001)
   02900520
   WRITE(16, 2009)
   02900530
   NEQF = JTYPE
   02900540
   WRITE(16, 400) NEQF
   02900550
   DO 500 I = 1, NEQF
   02900560
LINE 57  READ (15, 405) IEQ, IFTABL(1, IEQ), IFTABL(2, IEQ), 02900570
LINE 58  1 (CFTABL(J, IEQ), J=1, 6) 02900580
LINE 59  WRITE(16, 410) IEQ, (IFTABL(J, IEQ), J=1, 2) (CFTABL(J, IEQ), 02900590
LINE 60  1 J = 1, 6) 02900600
LINE 61  500 CONTINUE 02900610
LINE 62  C LOOP OVER ELEMENTS BY WIND SECTION 02900620
LINE 63  J = 5 02900630
LINE 64  I = 1 02900640
LINE 65  DO 630 J1=1, NWING 02900650
LINE 66  NS = NSECT(J1) 02900660
LINE 67  DO 395 J2=1, NS 02900670
LINE 68  IN1 = NE(J2, J1) 02900680
LINE 69  DO 390 K=1, IN1 02900690
LINE 70  X = KCEN(I)*BR 02900700
LINE 71  Y = (XYZ(J) + XYZ(J+1)) * HALFR 02900710
LINE 72  CALL MFUN(JMODE) 02900720
LINE 73  DO 380 IH=1, JMODE 02900730
LINE 74  ETA(IH) = EM(IH) 02900740
LINE 75  OETADX(IH) = DM(IH) 02900750
LINE 76  380 CONTINUE 02900760
LINE 77  J = J+6 02900770
LINE 78  600 CONTINUE 02900780
LINE 79  390 CONTINUE 02900790
LINE 80  395 CONTINUE 02900800
LINE 81  600 CONTINUE 02900810
LINE 82  IN1 = 1 02900820
LINE 83  IN2 = NET 02900830
LINE 84  GO TO 132 02900840
LINE 85  10 CONTINUE 02900850
LINE 86  C READ MOWING CARD 02900860
LINE 87  C 02900870
LINE 88  READ 15, 10001 LAB, NW, NI1, NI2, NI3 02900880
LINE 89  C 02900890
LINE 90  IF(LAB .EQ. LEND) GO TO 330 02900900
LINE 91  IF(LAB .EQ. LMW) GO TO 300 02900910
LINE 92  C 02900920
LINE 93  C CHECK TO SEE IF SECTION IS DEFINED 02900930
LINE 94  IF(NW .GT. NWING) GO TO 310 02900940
LINE 95  C 02900950
LINE 96  NSI = 3 02900960
LINE 97  C 02900970
LINE 98  NSI = 3 02900980
LINE 99  IF(NI3 .EQ. 0) NSI = 2 02900990
LINE 100  IF(NI2 .EQ. 0) NSI = 1 02901000
LINE 101  IF(NI1 .EQ. 0) GO TO 310 02901010
LINE 102  C 02901020
LINE 103  C IF 2 SECTIONS ARE INPUT, THEY MUST BE CONTINUOUS 02901030
LINE 104  IF(NSI .EQ. 2 .AND. IABS(NI2-NI1) .NE. 1) GO TO 320 02901040
LINE 105  C 02901050
LINE 106  C FIND FIRST AND LAST SECTION FOR THIS MODE GROUP 02901060
LINE 107  C 02901070
LINE 108  NSF = NI1 02901080
LINE 109  NSL = NI1 02901090
LINE 110  NSF = NI1 02901100
LINE 111  WRITE(16, 2001) 02901110
LINE 112  NSL = NI1 02901120
LINE 113  WRITE(16, 2002) NW, NI1 02901130
LINE 114  NSF = NSL 02901140
LINE 115  WRITE(16, 2002) NW, NI2 02901150
LINE 116  NSF = NSF 02901160
LINE 117  NSF = NSF 02901170
LINE 118  NSF = NSF 02901180
LINE 119          WRITE( 16, 2002 ) NW, N13
LINE 120          CONTINUE
LINE 121          C
LINE 122          C          MORE CHECK TO SEE IF SECTION IS DEFINED
LINE 123          IF( NSL .GT. NSECT(NW) ) GO TO 310
LINE 124          C
LINE 125          C          FIND FIRST AND LAST ELEMENT NUMBERS, IN1, IN2
LINE 126          C
LINE 127          IN1 = 1
LINE 128          IF( NW .EQ. 1 ) GO TO 30
LINE 129          C          FIND NUMBER OF ELEMENTS IN LOWER NUMBER WINGS
LINE 130          NW1 = NW - 1
LINE 131          DO 25 J=1,NW1
LINE 132          NS = NSECT(J)
LINE 133          DO 25 I=1,NS
LINE 134          25 IN1 = IN1 + NE(I,J)
LINE 135          C
LINE 136          30 CONTINUE
LINE 137          IF( NSF .EQ. 1 ) GO TO 40
LINE 138          C          FIND NUMBER OF ELEMENTS IN LOWER NUMBER SECTIONS OF WING NW
LINE 139          NS = NSF - 1
LINE 140          DO 35 I=1,NS
LINE 141          35 IN1 = IN1 + NE(I,NW)
LINE 142          C
LINE 143          40 CONTINUE
LINE 144          C          NOW THAT WE HAVE IN1, FIND LAST ELEMENT IN2
LINE 145          C
LINE 146          IN2 = IN1 + NE(NSF,NW) - 1
LINE 147          NCHECK(NSF,NW) = 1
LINE 148          IF( YSI .EQ. 1 ) GO TO 45
LINE 149          C
LINE 150          IN2 = IN2 + NE(NSF1,NW)
LINE 151          NCHECK(NSF1,NW) = 1
LINE 152          IF( YSI .EQ. 2 ) GO TO 45
LINE 153          IN2 = IN2 + NE(NSL,NW)
LINE 154          NCHECK(NSL,NW) = 1
LINE 155          45 CONTINUE
LINE 156          C
LINE 157          IN1 AND IN2 HAVE BEEN FOUND
LINE 158          C
LINE 159          WRITE( 16, MODEF4 ) JMODE
LINE 160          WRITE( 16, 2003 )
LINE 161          WRITE( 16, 2008 )
LINE 162          C
LINE 163          SECTION TO READ MODE DATA
LINE 164          C
LINE 165          C          READ MODE INPUT
LINE 166          C          MODE CARD HAS BEEN READ
LINE 167          C
LINE 168          C
LINE 169          NFUNMX = NMX
LINE 170          CALL READXY( XF, YF )
LINE 171          IERR = IERF
LINE 172          IF( IERR .NE. 0 ) RETURN
LINE 173          C
LINE 174          DO 130 J=1,JMODE
LINE 175          READ( 15, 1006 ) MODE
LINE 176          WRITE( 16, 1016 ) MODE
LINE 177          CALL READAB( A, A )
LINE 178          IF( ITEST .EQ. 0 ) GO TO 110
LINE 179          DO 135 I = IN1, IN2
LINE 180          ETA(I,J) = ZIZ

02901190
02901200
02901210
02901220
02901230
02901240
02901250
02901260
02901270
02901280
02901290
02901300
02901310
02901320
02901330
02901340
02901350
02901360
02901370
02901380
02901390
02901400
02901410
02901420
02901430
02901440
02901450
02901460
02901470
02901480
02901490
02901500
02901510
02901520
02901530
02901540
02901550
02901560
02901570
02901580
02901590
02901600
02901610
02901620
02901630
02901640
02901650
02901660
02901670
02901680
02901690
02901700
02901710
02901720
02901730
02901740
02901750
02901760
02901770
02901780
02901790
02901800

-6.70-
DETAOX(I,J) = 0.0
LINE 181 105 CONTINUE
LINE 182 GO TO 130
LINE 183 110 CONTINUE
LINE 185 C BUMP INDXR TO FIRST ELEMENT IN SECTION
LINE 186 INDXR = 6*IN1 - 1 & 1
LINE 187 DO 120 I = IN1, IN2
LINE 188 C NOTE X0, Y0 ARE COORD. OF CENTER OF ELEM IN UNNORMALIZED
LINE 189 C STRUCTURAL REFERENCE SYSTEM
LINE 190 XX = XR*Xcen(I)
LINE 191 YY = HALFR*(XY(I,INDX64) & XYZ(I,INDX65))
LINE 192 INDXR = INDXR & 6
LINE 193 C CALL ZFOZ( ETA(I,J), DETAOX(I,J), XI, XF, YF, A)
LINE 194 CALL ZFOZ( ETA(I,J), DETAOX(I,J), XI, XF, YF, A)
LINE 195 C 120 CONTINUE
LINE 196 120 CONTINUE
LINE 197 C 130 CONTINUE
LINE 198 130 CONTINUE
LINE 199 C 132 CONTINUE
LINE 200 132 CONTINUE
LINE 201 C 1 IPR = 1
LINE 202 WRITE( 16, 2004 ) IPR
LINE 203 IF(JTYPE .NE. 0) GO TO 135
LINE 204 WRITE( 16, 2004 ) IPR
LINE 205 WRITE( 16, 2005 ) N, N1
LINE 206 IF( NSI .GE. 2 ) WRITE( 16, 2002 ) N, N1,
LINE 207 IF( NSI .EQ. 3 ) WRITE( 16, 2002 ) N, N1,
LINE 208 135 CONTINUE
LINE 209 WRITE( 16, 2005 )
LINE 210 WRITE( 16, 2008 )
LINE 211 J1 = 1
LINE 212 IHS = JMODE/3
LINE 213 IF( 3*IHS .NE. JMODE ) IHS = IHS & 1
LINE 214 C 1 DO 138 IH = 1, IHS
LINE 215 138 CONTINUE
LINE 216 J2 = J1 + 2
LINE 217 IF( J2 .GT. JMODE ) J2 = JMODE
LINE 218 INUMB = J2 - J1 & 1
LINE 219 MODEF1(2) = NUMBR(INUMB)
LINE 220 MODEF2(5) = NUMBR(INUMB)
LINE 221 MODEF3(3) = NUMBR(INUMB)
LINE 222 IF( IH .GE. 2 ) WRITE( 16, 2010 )
LINE 223 WRITE( 16, MDF1(1) ) ( I, T = J1, J2)
LINE 224 WRITE( 16, MDF2(1) ) ( I, T = J1, J2)
LINE 225 WRITE( 16, MDF3(1) ) ( I, (ETA(I,J), DETAOX(I,J), J = J1, J2), I = IN1, IN2)
LINE 226 J1 = J1 + 1
LINE 227 138 CONTINUE
LINE 228 C 1 IN1 = IN2 & 1
LINE 229 IN2 = IN1 & 1
LINE 230 C 140 CONTINUE
LINE 231 140 CONTINUE
LINE 232 IF(JTYPE .NE. 0) GO TO 200
LINE 233 C 1 CHECK TO SEE IF ALL SECTIONS HAVE BEEN READ AND ACCOUNTED FOR
LINE 234 C
LINE 235 C 1 DO 150 J = 1, NS
LINE 236 NS = NSECT(J)
LINE 237 DO 150 J = 1, NS
LINE 238 DO 150 J = 1, NS
LINE 239 IF( VCHECK(I,J) .EQ. 0 ) GO TO 10
LINE 240 150 CONTINUE
LINE 241 C
LINE 242 C

-6.71-
LINE 243 C
LINE 244 C READ NEXT CARD, CHECK FOR RIGID MODE INPUT
LINE 245 200 READI 15, 1000 I LAB, JAGRID
LINE 246 210 CONTINUE
LINE 247 IFI LAB .EQ. LEND I RETURN
LINE 248 IFI LAB .NE. LGRID I GO TO 340
LINE 249 215 CONTINUE
LINE 250 C SECTION FOR RIGID MODES
LINE 251 C CHECK THAT RIGID MODE NUMBER IS IN ORDER.
LINE 252 JMODE = JMODE + 1
LINE 253 IFJ JAGRID .NE. JMODE I GO TO 350
LINE 254 C ZERO OUT ALL ETA AND DETADX FOR THAT MODE
LINE 255 DO 220 I=1,NET
LINE 256 ETA(I, JMODE) = 0.0
LINE 257 DETADX(I, JMODE) = 0.0
LINE 258 220 CONTINUE
LINE 259 WRITE( 16, 2006 )
LINE 260 WRITE( 16, 2005 )
LINE 261 WRITE( 16, 2008 )
LINE 262 WRITE( 16, 2007 )
LINE 263 MODF1(2) = NUMBR(1)
LINE 264 WRITE( 16, MODF1 ) JMODE
LINE 265 MODF2(5) = NUMBR(1)
LINE 266 MODF2(3) = NUMBR(1)
LINE 267 WRITE( 16, MODF2 )
LINE 268 CONTINUE
LINE 269 READI 15, 1001 I LAB, IEL, ETA(IEL, JMODE), DETADX( IEL, JMODE)
LINE 270 IFI LAB .NE. LBLANK I GO TO 240
LINE 271 WRITE( 16, MODF3 ) IEL, ETA(IEL, JMODE), DETADX( IEL, JMODE)
LINE 272 GO TO 230
LINE 273 240 CONTINUE
LINE 274 JAGRID = IEL / 100
LINE 275 GO TO 210
LINE 276 C ERROR SECTION
LINE 277 C
LINE 278 C
LINE 279 C
LINE 280 300 CONTINUE
LINE 281 C MDL INPUT CARD NOT FOUND
LINE 282 WRITE( 16, 3000 )
LINE 283 WRITE( 16, 3001 )
LINE 284 ERROR = .TRUE.
LINE 285 RETURN
LINE 286 310 CONTINUE
LINE 287 C SECTION NOT DEFINED
LINE 288 WRITE( 16, 3000 )
LINE 289 WRITE( 16, 3002 )
LINE 290 WRITE( 16, 2002 ) NW N1
LINE 291 IF( VIS GE 2 ) WRITE( 16, 2002 ) NW N1
LINE 292 IF( VIS EQ 3 ) WRITE( 16, 2002 ) NW N1
LINE 293 ERROR = .TRUE.
LINE 294 RETURN
LINE 295 320 CONTINUE
LINE 296 C SECTIONS ARE NOT ADJACENT
LINE 297 WRITE( 16, 3000 )
LINE 298 WRITE( 16, 3003 ) N1 N1
LINE 299 ERROR = .TRUE.
LINE 300 RETURN
LINE 301 330 CONTINUE
LINE 302 C MODES NOT PRESENT FOR ALL SECTIONS
LINE 303 WRITE( 16, 3000 )
LINE 304 WRITE( 16, 3004 )

-6.72-
LINE 305 3004 FORMAT(50H MODAL DATA NOT PRESENT FOR THE FOLLOWING SECTIONS )
LINE 306 DO 332 J=1,NWING
LINE 307 NS = NSECT(J)
LINE 308 NO = 332 * NS
LINE 309 IF( VCHECK(J,J) .EQ. 0 ) WRITE( 16, 2002 ) J, 1
LINE 310 332 CONTINUE
LINE 311 ERROR = .TRUE.
LINE 312 RETURN
LINE 313 C
LINE 314 340 CONTINUE
LINE 315 WRITE( 16, 3000 )
LINE 316 WRITE( 16, 3005 ) LAB
LINE 317 ERROR = .TRUE.
LINE 318 RETURN
LINE 319 C
LINE 320 350 CONTINUE
LINE 321 WRITE( 16, 3000 )
LINE 322 WRITE( 16, 3006 ) JMODE, JRIGID
LINE 323 ERROR = .TRUE.
LINE 324 RETURN
LINE 325 C
LINE 326 400 FORMAT( 120H/ 50X, 114, 25H MODE FUNCTION EQUATIONS /)
LINE 327 1 4X, SBEQUATION ,2X,4X, 4H WING, 2X,4X, 4H MOLEC, 4X, 20X, 36X,8H CONSTANT, 2X, 8X, 1HX, 5X, 8X,1HY, 5X, 02903270
LINE 328 4 6X, 2HXY, 6X ,
LINE 329 3X,8H SQUARE , 3X, 3X, BHY SQUARE
LINE 330 405 FORMAT( 6X, 312, 12X, 86X, 6 )
LINE 331 410 FORMAT( 1HO,8X,112,4X,4X,112, 4X,4X,12, 2X,(16E14.4))
LINE 332 1000 FORMAT( A4, 2X, 412 )
LINE 333 1001 FORMAT( 1H0,8X,1I2, 4X, 2OE12.0 )
LINE 334 1006 FORMAT( (6E14.4) )
LINE 335 1016 FORMAT( 1X, 20A4 )
LINE 336 2001 FORMAT( 1H1/ 24X,16HI NPUT MODAL DATA // )
LINE 337 2002 FORMAT( 20X, 4HO, 12, 10H SECTION, 12 )
LINE 338 2003 FORMAT( // 20X,10HCOEFF CHECK )
LINE 339 2004 FORMAT( 11, 19X, 20H GENERATED MODAL DATA // )
LINE 340 2005 FORMAT( // 20X,16HAERODYNAMIC GRID/ )
LINE 341 2006 FORMAT( // 20X, 21H RIGID BODY MODE INPUT )
LINE 342 2007 FORMAT( // 20X, 55H ONLY ELEMENTS WITH A NON-ZERO MODAL VALUE MUST )
LINE 343 2008 FORMAT( // 20X, 30H STRUCTURAL REFERENCE SYS )
LINE 344 2009 FORMAT( 20X, 35H RIGID MODE NUMBER IS N)T IN ORDER. /
LINE 345 2010 FORMAT( // 20X, 44H MODAL DATA INPUT IS GENERATED BY POLYNOMIAL , 
LINE 346 2011 FORMAT( // 20X, 11H DEFINED BY USER )
LINE 347 2012 FORMAT( 1H1 )
LINE 348 2013 FORMAT( 1H1 )
LINE 349 C
LINE 350 3000 FORMAT( // 1X, 12014*) // 29H ERROR IN READING MODE INPUT. )
LINE 351 3001 FORMAT( 46H * MOIVNG DATA CARD IS EXPECTED BUT NOT FOUND. / )
LINE 352 1 51H THIS CARD MUST PRECEDE EACH MODE DATA INPUT GROUP. /
LINE 353 210S9 THIS CARD MUST FOLLOW THE 'MODE' DATA CARD OR THE LAST DATA CARD
LINE 354 3ARD OF THE PRECEDING MODE DATA INPUT GROUP. /
LINE 355 3002 FORMAT( 76H " MOIVNG" DATA CARD SPECIFIES INPUT FOR A SECTION THAT 
LINE 356 14S NOT BEEN DEFINED. / 19H SECTIONS SPECIFIED )
LINE 357 3003 FORMAT( 6D * MOIVNG" DATA CARD SPECIFIES TWO SECTIONS THAT ARE NOT )
LINE 358 1ADJACENT. / 8H SECTION, 13,12H AND SECTION, 13 )
LINE 359 3004 FORMAT( // 90H A LABEL DATA CARD FOLLOWING FLEXIBLE MODE DATA INPUT )
LINE 360 1 IT IS EITHER OUT OF ORDER OR INCOMPLETE. / 68H THE FIRST FOUR CHAR 
LINE 361 2TEKS OF THE CARD FOLLOW WITHIN PARENTHESES. (, 44, 1H )
LINE 362 3006 FORMAT( // 39H RIGID MODE NUMBER IS NOT IN ORDER. / 24H MODE NUMBER )
LINE 363 1R EXPECTED IS, 14 / 21H MODE NUMBER INPUT IS, 14 )
LINE 364 C
LINE 365 END

-6.73-
DISPLAY FILE DECK 30

LINE 1 SUBROUTINE READAB(Z, A)
LINE 2 INTEGER HEADER(20)
LINE 3 DIMENSION Z(I), A(I)
LINE 4 COMMON /ZFDZIO / I5, I6
LINE 5 COMMON /ZFUNNY / N,IERF,B1,H2, B3, NFUNXM
LINE 6 COMMON /XYSCAL / XX, YY, XG, YC, BREF, ZZZ, ITEST
LINE 7 C
LINE 8 C
LINE 9 C INITIALIZATION ENTRY POINT (COEFFICIENTS).
LINE 10 READ( 15, 11 ) HEADER
LINE 11 WRITE( 16, 11 ) HEADER
LINE 12 READ( 15, 13 ) ( Z(I), I=1, N )
LINE 13 WRITE( 16, 12 ) ( Z(I), I=1, N )
LINE 14 ITEST = 0
LINE 15 ZZZ = Z(1)
LINE 16 DO 5 I=2, N
LINE 17 IF ( Z(I) .NE. ZZZ ) GO TO 7
LINE 18 5 CONTINUE
LINE 19 ITEST = 1
LINE 20 7 CONTINUE
LINE 21 READ( 15, 11 ) HEADER
LINE 22 WRITE( 16, 11 ) HEADER
LINE 23 READ( 15, 13 ) ( A(I), I=1, N )
LINE 24 WRITE( 16, 12 ) ( A(I), I=1, N )
LINE 25 READ( 15, 11 ) HEADER
LINE 26 WRITE( 16, 11 ) HEADER
LINE 27 READ( 15, 13 ) B1, B2, B3
LINE 28 WRITE( 16, 12 ) B1, B2, B3
LINE 29 RETURN
LINE 30 11 FORMAT(20A4)
LINE 31 12 FORMAT(1PE15.7)
LINE 32 13 FORMAT(5E15.7)
LINE 33 END
| LINE 1 | SUBROUTINE READXY(X,Y)                                      | 03100010 |
| LINE 2 | INTEGER HEADER(20)                                         | 03100020 |
| LINE 3 | DIMENSION X(1), Y(1)                                        | 03100030 |
| LINE 4 | COMMON /ZFDZO10 / JS, 16                                    | 03100040 |
| LINE 5 | COMMON /ZFUNNY/ NS, IERF, B1, B2, BS, NFUNMX               | 03100050 |
| LINE 6 | C                                                            | 03100060 |
| LINE 7 | C INITIALIZATION ENTRY POINT (ABSCISSAS).                   | 03100070 |
| LINE 8 | C USAGE CALL READ NXY                                       | 03100080 |
| LINE 9 | READ( 15, 10 ) N                                           | 03100090 |
| LINE 10| WRITE( 16, 10 ) N                                          | 03100100 |
| LINE 11| READ( 15, 11 ) HEADER                                       | 03100110 |
| LINE 12| WRITE( 16, 11 ) HEADER                                      | 03100120 |
| LINE 13| IERF = 0                                                   | 03100130 |
| LINE 14| IF (V.GT. NFUNMX) IERF = 680                               | 03100140 |
| LINE 15| IF (V.GT. NFUNMX) RETURN                                    | 03100150 |
| LINE 16| READ( 15, 13 ) ( X(I), I=1,N )                             | 03100160 |
| LINE 17| WRITE( 16, 12 ) ( X(I), I=1,N )                            | 03100170 |
| LINE 18| READ( 15, 11 ) HEADER                                       | 03100180 |
| LINE 19| WRITE( 16, 11 ) HEADER                                      | 03100190 |
| LINE 20| READ( 15, 13 ) ( Y(I), I=1,N )                             | 03100200 |
| LINE 21| WRITE( 16, 12 ) ( Y(I), I=1,N )                            | 03100210 |
| LINE 22| RETURN                                                     | 03100220 |
| LINE 23| 10 FORMAT( 5X, 3HN =,13 )                                  | 03100230 |
| LINE 24| 11 FORMAT(ZC441)                                           | 03100240 |
| LINE 25| 12 FORMAT( 1P5E15, 7)                                      | 03100250 |
| LINE 26| 13 FORMAT( 5E15, 7 )                                       | 03100260 |
| LINE 27| END                                                        | 03100270 |
SUBROUTINE RSTART ( MACH, NET, NREQ, FREQ, REFLN, IWTE, 18, 115)

REAL MACH, NETR

DIMENSION FREQ(11, FREQRI), AREA(1), XYZ(1)

COMMON /NEXTCS /I FLUSH

ERROR = .FALSE.

IS = 18

CONTINUE

REWIND IS

READ(15) MACHR, NETR, RRFL, IWTER, NFREQR, (FREQR(I), I=1, NFREQR)

IF( MACHR .NE. 'MACH ' GO TO 90

IF( NETR .NE. NET ) GO TO 90

IF( NFREQR .NE. NFREQ ) GO TO 90

IF( REFLN .NE. RRFL ) GO TO 90

IF( IWTE .NE. 0 ) AND. IWTER .EQ. 0) GO TO 90

IF( FREQR(I) .NE. FREQ(I)) GO TO 90

CONTINUE

IF( IWTE .EQ. 0 ) GO TO 30

IF( IS .EQ. 115 ) GO TO 30

IS = 115

GO TO 10

CONTINUE

ERROR IN RESTART TAPE 

ERROR = .TRUE.

WRITE( 16, 2000 ) IS

WRITE( 16, 2001 ) MACH, MACHR, NET, NETR, REFLN, RRFL, IWTE, IWTER, NFREQ, NFREQR, (1, FREQRI), FREQRI, I=1, NFREQR)

IF( IWTE .EQ. D ) GO TO 99

IF( IS .EQ. 115 ) GO TO 99

IS = 115

GO TO 10

CONTINUE

IFLUSH = 1

RETURN

2000 FORMAT(1HI/ 60(2H)) //8BH ERROR IN READING RESTART TAPE ON UNIT

2001 FORMAT (41X, 11HCURRENT RUN, 12HRESTART TAPE )

2002 FORMAT( / 11X, 24HNUMBER, 11H7.5, 11H7.5)

2003 FORMAT( / 11X, 24HTOTAL NUMBER OF ELEMENTS, 11H7, 121)

2004 FORMAT( / 11X, 24HREFERENCE LENGTH, 11H7.5, 11H7.5)
SUBROUTINE RTOI( TR, TI, YCR, ZCR, ZRJ, ZLO )

C RTOI TRANSFORMS YCR AND ZCR FROM THE REceiving system TO THE INFLUENCING SYSTEM. YCR, ZCR ARE FIRST PUT IN REFERENCE SYSTEM BY PREMULTIPLYING BY TR. THIS IS Y,Z WHICH IS PUT IN INFLUENCING SYSTEM. YR, ZR ARE THEN PUT IN REFERENCE SYSTEM BY THE TRANSCES OF TI.

DIMENSION TR(1), TI(1)

COMMON / RLL / TVW(4), YR0, YL0, ZR, ZRR, ZL, ZLLL

Y = TR(1)*YCR & TR(3)*ZCR
Z = TR(2)*YCR & TR(4)*ZCR
YR0 = TI(1)*Y & TI(2)*Z
ZRO = TI(3)*Y & TI(4)*Z
YL0 = -TI(1)*Y & TI(2)*Z
ZL0 = TI(3)*Y - TI(4)*Z
TVW(1) = TR(3)*TI(1) & TR(4)*TI(2)
TVW(2) = TR(3)*TI(3) & TR(4)*TI(4)
TVW(3) = -TR(3)*TI(1) & TR(4)*TI(2)
TVW(4) = TR(3)*TI(3) - TR(4)*TI(4)

RETURN

END
SUBROUTINE SOLVE(IN,NDIM,A,B,IP)

PURPOSE

SOLUTION OF THE LINEAR SYSTEM OF EQUATIONS \( C \cdot X = B \).

USAGE

CALL SOLVE(N,NDIM,A,B,IP)

DESCRIPTION OF PARAMETERS

- **N** - ORDER OF THE MATRIX \( A \)
- **NDIM** - FIRST DIMENSION OF A DECLARED IN CALLING PROGRAM. SET NDIM=N.
- **A** - CONTAINS THE TRIANGULAR FACTORS OF THE MATRIX \( C \) (AS DETERMINED BY SUBROUTINE 'DECOM').
- **B** - ON INPUT, THE RHS VECTOR, ON OUTPUT, THE SOLUTION VECTOR.
- **IP** - VECTOR OF DIMENSION \( N \) CONTAINING ROW INTERCHANGE INFORMATION (AS DETERMINED BY SUBROUTINE 'DECOM').

REMARKS

'SOLVE' MUST BE USED IN CONJUNCTION WITH SUBROUTINE 'DECOM'.

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED.

METHOD

BACK SUBSTITUTION BASED ON THE FACTORED FORM OF THE COEFFICIENT MATRIX. SEE ALGORITHM 423, 'COLLECTED ALGORITHMS FROM CACM', BY CLEVE MOLER.

DIMENSION A(NDIM,1),B(1),IP(1)

IF (N .EQ. 1) GO TO 9

DO 1 K = 1, N

M = IP(K)

T = B(M)

B(M) = B(K)

B(K) = T

} DO 1 1 = KP, N

B(1) = B(1) - A(1,K)*T

B(1) = B(1)/A(1,1)

RETURN

END
SUBROUTINE SONS(NWIN, NSECT, CORNX, CORNY, CORNZ, ISONS)

DIMENSION NSECT(1), CORNX(8,4), CORNY(9941), CORNZ(8941)

ISONS = 1

DO 3 J = 1, NWIN
    NS1 = NSECT(J)
    DO 25 I = 1, NS1
        K1 = K1 + 1
        JCN1 = 2*I - 2
        LOOP ON SECTIONS WHICH MAY HAVE INFLUENCE (DO 20 AND DO 15)
        K2 = K2 + 1
        ISONS(K1, K2) = 1
        A SECTION ALWAYS INFLUENCES ITSELF AND ADJACENT SECTIONS
        IF (J .EQ. JJ .AND. IABS(I-II) .LE. 1) GO TO 15
        JCN2 = 2*II - 1

        LOOP ON BOTTOM CORNER POINTS OF RECEIVING SECTION (PTS 2, 4)
        DO 10 12 = 2, 4, 2
            IND1 = JCN1 + 12
            XC = CORNX(IND1, JJ)
            YC = CORNY(IND1, JJ)
            ZC = CORNZ(IND1, JJ)
            X2 = XC - X1
            X3 = XC - X2
            Y2 = YC - Y1
            Y3 = YC - Y2
            Z2 = ZC - Z1
            Z3 = ZC - Z2
            IF (Y1 .EQ. Y3) GO TO 8
            IF (Y1 .LE. Y3) GO TO 8
            YCY1 = YC - Y1
            Y3YC = Y3 - YC
            IF (YCY1 .LE. Y3YC) GO TO 8

        LOOP ON TOP CORNER Pts. OF INFLUENCING SECTION (PTS 1 AND 3)
        IND1 = JCN2 + 2
        K1 = CORNX(JCN2, JJ)
        X3 = CORNX(JCN2, JJ)
        IF (X3 .LT. X1 .AND. XC .LT. X3) GO TO 10
        Y1 = CORNY(JCN2, JJ)
        Y3 = CORNY(JCN2, JJ)
        Z1 = CORNZ(JCN2, JJ)
        Z3 = CORNZ(JCN2, JJ)
        IF (Y1 .EQ. Y3) GO TO 8
        IF (Y1 .LE. Y3) GO TO 8
        YCY1 = YC - Y1
        Y3YC = Y3 - YC
        IF (YCY1 .LE. Y3YC) GO TO 8

        IF (YCY1 = YCY1) GO TO 8
        IF (YCY1 .LE. Y3YC) GO TO 8

        IF (YC .BETWEEN Y1 AND Y3, CHECK INTERSECTION)
        A = YCY1/Y3
        X = X1 - 4*(X3-X1)
        Z = Z1 - A*(Z3-Z1)
        R = XC - X .LT. Z - ZC

    END

END

-6.80-
IF R GT 0, SECTION HAS INFLUENCE SO BRANCH OUT OF DO 10 LOOP

IF( R .GT. 0.0 ) GO TO 15

GO TO 10

CONTINUE

YC IS OUTSIDE OF Y1,Y3 CHECK BOTH POINTS

ZHI = XC - X1

ETA = YC - Y1

ZO = ZC - Z1

T = SQRT( ETA*ETA + ZO*ZO )

R = ZHI - T

IF R GT 0, SECTION HAS INFLUENCE SO BRANCH OUT OF DO 10 LOOP

CONTINUE

WE DID NOT BRANCH OUT OF DO 10 LOOP

SECTION K2 DOES NOT INFLUENCE SECTION K1

ISONS(K1,K2) = 0

END LOOP ON INFLUENCING SECTION (15) AND WING (20)

CONTINUE

END LOOP ON RECEIVING SECTION (25) AND WING (30)

CONTINUE

CONTINUE

RETURN

END
DISPLAY...FILE

DECK 36

LINE 1 SUBROUTINE SONSPT( ISONS, NWING, NSECT, MACH, I6 ) 03600010
LINE 2 C SONSPT PRINTS SECTION INFLUENCE TABLE 03600020
LINE 3 REAL MACH 03600030
LINE 4 DIMENSION IFT1( 9 ), IFT2( 9 ), IF(15) 03600040
LINE 5 DIMENSION (ISONS(12,12), NSECT(1), IW(15), IS(15) 03600050
LINE 6 C 03600060
LINE 7 DATA IFT1 / 4H(11H, 4H REC, 4HEIVI, 4HNG, , 1H, 4H(6H, 03600070
LINE 8 1 4H WIN, 4HG,12, 2H) / 03600080
LINE 9 DATA IF / 1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,2H10,2H11,2H12,2H13 03600090
LINE 10 1 2H14, 2H15 / 03600100
LINE 11 DATA IFT2 / 4H(11H, 4H SEC, 4HTION, 4HS, , 1H, 4H(6H, 03600110
LINE 12 1 4H SEC, 4H7,12, 2H) / 03600120
LINE 13 C 03600130
LINE 14 WRITE( 16, 1001 ) MACH 03600140
LINE 15 WRITE( 16, 1002 ) 03600150
LINE 16 K = 0 03600160
LINE 17 DO 10 J=1,NWING 03600170
LINE 18 NS = NSECT(J) 03600180
LINE 19 DO 5 I=1,NS 03600190
LINE 20 K = K + 1 03600200
LINE 21 IW(K) = J 03600210
LINE 22 IS(K) = I 03600220
LINE 23 5 CONTINUE 03600230
LINE 24 10 CONTINUE 03600240
LINE 25 IFT1(5) = IF(K) 03600250
LINE 26 IFT2(5) = IF(K) 03600260
LINE 27 WRITE( 16, IFT1 ) (IWI(I),I=1,K) 03600270
LINE 28 WRITE( 16, IFT2 ) (IS(I), I=1,K) 03600280
LINE 29 DO 20 I=1,K 03600290
LINE 30 WRITE( 16, 1003 ) IW(I), IS(I), ( ISONS(I,J), J=1,K ) 03600300
LINE 31 20 CONTINUE 03600310
LINE 32 1001 FORMAT( 1H1 / 4OX,4HTABLE OF INFLUENCING SECTIONS FOR MACH NUMBER 03600320
LINE 33 1, F8.4 ) 03600330
LINE 34 1002 FORMAT( 1X,50KEY 0 ALL ELEMENTS OF INFLUENCING SECTION HAVE 03600340
LINE 35 1, 50HERO INFLUENCE ON ALL ELEMENTS OF RECEIVING SECTION // 03600350
LINE 36 2 7X, 32H1 NON-ZERO INFLUENCE IS ASSUMED /// 03600360
LINE 37 3 13X, 20HINFLUENCING SECTIONS ) 03600370
LINE 38 1003 FORMAT( / 5H WING, 12 / 5H SECT, 12, 4X, 1518 ) 03600380
LINE 39 RETURN 03600390
LINE 40 END 03600400
LINE 1  SUBROUTINE SYMBOL(X, Y, SIZE, CHAR, ANG, NCHAR)
LINE 2  DIMENSION CHAR(3)
LINE 3  CALL NOTATE(X, Y, SIZE, CHAR, ANG, NCHAR)
LINE 4  RETURN
LINE 5  END
DISPLAY...FILE  

DECK 38

LINE 1  SUBROUTINE TIMOUT(MSEC,A,B,C,D,E,F,G,H)
LINE 2  COMMON /TAPE/ I9,I10,I11,I12,I13,I14,J9
LINE 3  DATA TI / 0.0 /
LINE 4  T = MSEC
LINE 5  T = T/1000.
LINE 6  TI = T - TI
LINE 7  WRITE(J9,3001)A,B,C,D,E,F,G,H,TI,T
LINE 8  TI = T
LINE 9  RETURN
LINE 10  3001 FORMAT(6X, 8.3, 5X, F8.3, 13X, F8.3, 1
LINE 11  END

-6.84-
CISPLAY...FILE   DECK 39

LINE  1 SUBROUTINE TMAX(NET, A, GBAR)  03900010
LINE  2 DIMENSION A(1)  03900020
LINE  3 DATA EPS / .0000001/  03900030
LINE  4 GBAR = .0  03900040
LINE  5 DO 480 IEL=1,NET  03900050
LINE  6 G = ABS(A(IEL))  03900060
LINE  7 IF(G.GT. GBAR ) GBAR = G  03900070
LINE  8 480 CONTINUE  03900080
LINE  9 IF(GBAR .EQ.0.0) GO TO 600  03900090
LINE 10 DO 580 IEL = 1,NET  03900100
LINE 11 G = EPS * GBAR  03900110
LINE 12 G = ABS( G)  03900120
LINE 13 AIEL = ABS (A(IEL) )  03900130
LINE 14 IF (G .GT. AIEL) A(IIEL) = .0  03900140
LINE 15 580 CONTINUE  03900150
LINE 16 600 RETURN  03900160
LINE 17 END  03900170

-6.85-
LINE 1  SUBROUTINE TRANS (X, Y, Z, T, XP, YP, ZC) 04000010
LINE 2  DIMENSION X(11), Y(11), Z(11), T(41), XP(1), YP(1) 04000020
LINE 3  T1X = X(4) - X(1) 04000030
LINE 4  T1Y = Y(4) - Y(1) 04000040
LINE 5  T1Z = Z(4) - Z(1) 04000050
LINE 6  T2X = X(3) - X(2) 04000060
LINE 7  T2Y = Y(3) - Y(2) 04000070
LINE 8  T2Z = Z(3) - Z(2) 04000080
LINE 9  XN = T1Y*T2Z - T1Z*T2Y 04000090
LINE 10  YN = T1Z*T2X - T1X*T2Z 04000100
LINE 11  ZN = T1X*T2Y - T1Y*T2X 04000110
LINE 12  XYZN = XN*XN + YN*YN + ZN*ZN 04000120
LINE 13  XYZN = SQRT(XYZN) 04000130
LINE 14  XN = XN/XYZN 04000140
LINE 15  YN = YN/XYZN 04000150
LINE 16  ZN = ZN/XYZN 04000160
LINE 17  C 04000170
LINE 18  T(1) = ZN 04000180
LINE 19  T(2) = -YN 04000190
LINE 20  T(3) = YN 04000200
LINE 21  T(4) = ZN 04000210
LINE 22  C 04000220
LINE 23  IT = 0 04000230
LINE 24  ZC = YN*Y(I) + ZN*Z(I) 04000240
LINE 25  ZX = ZC 04000250
LINE 26  IF (ABS(ZC) .LT. 1.E-10) ZX = 1.0 04000260
LINE 27  DD 10 I=1,4 04000270
LINE 28  XP(I) = X(I) 04000280
LINE 29  YP(I) = ZN*Y(I) - YN*Z(I) 04000290
LINE 30  C FOLLOWING CALCULATION FOR DEBUG 04000300
LINE 31  C Z1 = YN*Y(I) + ZN*Z(I) 04000310
LINE 32  C IF (ABS(Z1-ZC) /ZX .GT. 0.001) IT = 1 04000320
LINE 33  10 CONTINUE 04000330
LINE 34  C WRITE( 6,1001 ) ( X(I),XP(I), Y(I),YP(I), Z(I),Z1 , I=1,4) 04000340
LINE 35  C WRITE( 6,1002 ) T 04000350
LINE 36  C IF( IT .EQ. 1 ) GO TO 20 04000360
LINE 37  RETURN 04000370
LINE 38  C20 CONTINUE 04000380
LINE 39  C WRITE( 6,1003 ) 04000390
LINE 40  C1003 FORMAT( /X,120(1H+)//79H ERROR IN TRANS - ALL Z'S ARE NOT IDENTICAU4000400
LINE 41  C FOR A SECTION. PROCEED WITH CAUTION // 1X, 120(1H+) // // ) 04000410
LINE 42  C RETURN 04000420
LINE 43  C1001 FORMAT( 3(1X, 12.E14.4, 2X) ) 04000430
LINE 44  C1002 FORMAT( 3(1X, 12.E14.4) ) 04000440
LINE 45  END 04000450
SUBROUTINE waket(i12, arrow, wrow, if, lreclw, nspt, i2, k1, nnch)
    COMPLEX wtes
    COMPLEX arrow(nspt), wrow(lreclw)
    COMPLEX arrowj
    LOGICAL tredge, wake1, wake, wakenz
    DIMENSION iz(nspt)
    DIMENSION nnch(nspt)
    COMMON /FQ1/ nfrf
    COMMON /KST11/KST12
    COMMON/wakeup/wtes(12), tredge, wake1, wake, wakenz, wake
    IF(k1.eq.k) go to 60
    read(i8) k1, wrow
    if(k1.ne.k) return
    60 continue
    nzn = 0
    l = if
    if(nzn.ne.0) wakenz = .true.
    if(nzn.eq.0) return
    call writzw(i12, k1, nnz, iz, arrow)
    kst12 = k1
    return
    end
DISPFILE

DECK 42

LINE 1 SUBROUTINE WINGC(KCORNX,CORN Y,CORNZ, NWING, ICN, ERROR, l6 )
LINE 2 L O G I C A L E R R O R
LINE 3 D I M E N S I O N CORNX(l), CORNY(l), CORNZ(l)
LINE 4 C C H E C K O N V A L I D I T Y O F C O R N E R P O I N T S O F W I N G
LINE 5 ICNMI = ICN-1
LINE 6 D O 10 I = 1, ICNMI, 2
LINE 7 J = I & 1
LINE 8 IF ( CORNX( J ) . L T . CORNX(I) ) Go To 5
LINE 9 IF ( CORNY( J ) . N E . CORNY(I) ) Go To 5
LINE 10 IF ( CORNZ( J ) . N E . CORNZ(I) ) Go To 5
LINE 11 Go To 10
LINE 12 5 W R I T E ( 16, 1001 ) I, J, NW I N G
LINE 13 ERROR = . T R U E.
LINE 14 10 C O N T I N U E
LINE 15 R E T U R N
LINE 16 1001 F O R M A T ( // 120(1H*) /32H E R R O R I N D E F I N I N G C O R N E R P O I N T S, IZ,
LINE 17 1 H,3H1,3H OF W I N G, IZ / )
LINE 18 E N D

04200010
04200020
04200030
04200040
04200050
04200060
04200070
04200080
04200090
04200100
04200110
04200120
04200130
04200140
04200150
04200160
04200170
04200180
SUBROUTINE WINGINI(N5, 16, NWING, NSECT, AR, NSP, CORNX, CORNY, CORNZ, 04300010
CORNZ, SYM, ICL, CL, ERROR, BEETA, IPW, NREF, FREQ, 04200020
REFLEN, LAB, ID, FD) 04400030

THIS ROUTINE READS AND WRITES INPUT FOR WING DATA 04300040

INPUT 04300050
15 INPUT UNIT (USUALLY CARD READER 5) 04300060
16 OUTPUT UNIT (PRINTER 6) 04300070

BEETA SQRT(MACH*MACH-1) USED TO TRANSFORM CORNX AND CL 04300080
ERROR LOGICAL THAT COMES IN FALSE, MAY BE SET TRUE 04300090

OUTPUT 04300100
NWING NUMBER OF WING READ FOR STRUCTURE 04300110
NSECT ARRAY—NO. SECTIONS IN EACH WING 04300120
AR ARRAY—ASPECT RATIO OF EACH WING 04300130
NSP NSP(I, J) NO. SPANS FOR SECTION I OF WING J 04300140
CORNX, CORNY, CORNZ X, Y, AND Z COORD. OF CONTROL PTS. OF WING 04300150
SYM SYM(I) SYMMETRY CODE FOR WING J (A REAL MATRIX) 04300160
SYM = -1 FOR SYMMETRY, 0 FOR NO SYM. 04300170

ICL ICL(I, J) CODE FOR SECTION I OF WING J 04300180
ICL(I, J) = 1, OR 2 AUTO MESH WITH 1 OR 2 CONTROL LINES 04300190
X COORD OF CONTROL LINES FOR EACH SECTION 04300200
CL IS READ IN REF. SYSTEM, THEN DIVIDED BY BEETA 04300210

LAB, ID, FD INFORMATION ON LAST CARD READ 04300220

LOGICAL ERROR, CLINPT 04300230
LOGICAL FREQ 04300240

DIMENSION CORNX(16,41,4), CORNY(16,41,4), CORNZ(16,41,4), CL(4,3,4) 04300250
DIMENSION FREQ(1), IPW(1) 04300260
DIMENSION NSECT(1), AR(1), NSP(3,4), ICL(3,4), ID(9), FD(4) 04300270
DIMENSION SK1(4), SK2(4), SK3(4), SKODE(4) 04300280
DIMENSION SYM(4) 04300290

DATA L1, L2, L3, L4, L5, L6, L7, L8, L9, L10 04300300
DATA L11, L12, L13, L14, L15, L16, L17, L18, L19, L20 04300310
DATA L31, L32, L33, L34, L35, L36, L37, L38, L39, L40 04300330
DATA L41, L42, L43, L44, L45, L46, L47, L48, L49, L50 04300340
DATA L51, L52, L53, L54, L55, L56, L57, L58, L59, L60 04300350
DATA L51, L52, L53, L54, L55, L56, L57, L58, L59, L60 04300360
DATA L61, L62, L63, L64, L65, L66, L67, L68, L69, L70 04300370
DATA L71, L72, L73, L74, L75, L76, L77, L78, L79, L80 04300380
DATA L81, L82, L83, L84, L85, L86, L87, L88, L89, L90 04300390
DATA L91, L92, L93, L94, L95, L96, L97, L98, L99, L100 04300400
DATA L121, L122, L123, L124, L125, L126, L127, L128, L129, L130 04300430
DATA L141, L142, L143, L144, L145, L146, L147, L148, L149, L150 04300450
DATA L151, L152, L153, L154, L155, L156, L157, L158, L159, L160 04300460
DATA L171, L172, L173, L174, L175, L176, L177, L178, L179, L180 04300480
DATA L181, L182, L183, L184, L185, L186, L187, L188, L189, L190 04300490
DATA L201, L202, L203, L204, L205, L206, L207, L208, L209, L210 04300510
DATA L221, L222, L223, L224, L225, L226, L227, L228, L229, L230 04300530

FRQ = .FALSE. 04300570
NHING = 0 04300580
BEERFL = BEETA*REFLEN 04300590

BDPREDUCIBILITY OF THE ORIGINAL PAGE IS POOR
LINE 57    IF( LAB .EQ. L5 ) GO TO 50
LINE 58    IF( LAB .EQ. LEND ) GO TO 60
LINE 59    IF( LAB .EQ. LB ) GO TO 70
LINE 60    INVALID LABEL CARD READ
LINE 61    WRITE( I6, 2000 1
LINE 62    WRITE( I6, 2018 1
LINE 63    WRITE( I6, 2017 ) LAB, ID, FD
LINE 64    ERROR = .TRUE.
LINE 65    GO TO 10
LINE 66    IF( LAB .EQ. LEND ) GO TO 60
LINE 67    IF( LAB .EQ. LB ) GO TO 70
LINE 68    INVALID LABEL CARD READ
LINE 69    WRITE( I6, 2000 1
LINE 70    WRITE( I6, 2018 1
LINE 71    WRITE( 16, 2017) LAB, ID, FD
LINE 72    ERROR = .TRUE.
LINE 73    CONTINUE
LINE 74    CONTINUE
LINE 75    IF( NWING .LE. 4 ) GO TO 21
LINE 76    IF( NWING .EQ. ID(1) ) GO TO 22
LINE 77    IF( ID(11 .LE. 4 ) GO TO 23
LINE 78    IF( NWING .EQ. ID(1) ) GO TO 22
LINE 79    IF( ID(11 .GT. 0 .AND. ID(11 .LE. 4 ) GO TO 23
LINE 80    IF( NWING .LE. 4 ) GO TO 21
LINE 81    IF( NWING .EQ. ID(1) ) GO TO 22
LINE 82    IF( NWING .EQ. ID(1) ) GO TO 22
LINE 83    IF( NWING .EQ. ID(1) ) GO TO 22
LINE 84    IF( NWING .EQ. ID(1) ) GO TO 22
LINE 85    IF( NWING .EQ. ID(1) ) GO TO 22
LINE 86    IF( NWING .EQ. ID(1) ) GO TO 22
LINE 87    IF( NWING .EQ. ID(1) ) GO TO 22
LINE 88    IF( NWING .EQ. ID(1) ) GO TO 22
LINE 89    IF( NWING .EQ. ID(1) ) GO TO 22
LINE 90    IF( NWING .EQ. ID(1) ) GO TO 22
LINE 91    IF( NWING .EQ. ID(1) ) GO TO 22
LINE 92    IF( NWING .EQ. ID(1) ) GO TO 22
LINE 93    IF( NWING .EQ. ID(1) ) GO TO 22
LINE 94    IF( NWING .EQ. ID(1) ) GO TO 22
LINE 95    IF( NWING .EQ. ID(1) ) GO TO 22
LINE 96    IF( NWING .EQ. ID(1) ) GO TO 22
LINE 97    IF( NWING .EQ. ID(1) ) GO TO 22
LINE 98    IF( NWING .EQ. ID(1) ) GO TO 22
LINE 99    IF( NWING .EQ. ID(1) ) GO TO 22
LINE 100   IF( NWING .EQ. ID(1) ) GO TO 22
LINE 101   IF( NWING .EQ. ID(1) ) GO TO 22
LINE 102   IF( NWING .EQ. ID(1) ) GO TO 22
LINE 103   IF( NWING .EQ. ID(1) ) GO TO 22
LINE 104   IF( NWING .EQ. ID(1) ) GO TO 22
LINE 105   IF( NWING .EQ. ID(1) ) GO TO 22
LINE 106   IF( NWING .EQ. ID(1) ) GO TO 22
LINE 107   IF( NWING .EQ. ID(1) ) GO TO 22
LINE 108   IF( NWING .EQ. ID(1) ) GO TO 22
LINE 109   IF( NWING .EQ. ID(1) ) GO TO 22
LINE 110   IF( NWING .EQ. ID(1) ) GO TO 22
LINE 111   IF( NWING .EQ. ID(1) ) GO TO 22
LINE 112   IF( NWING .EQ. ID(1) ) GO TO 22
LINE 113   IF( NWING .EQ. ID(1) ) GO TO 22
LINE 114   IF( NWING .EQ. ID(1) ) GO TO 22
LINE 115   IF( NWING .EQ. ID(1) ) GO TO 22
LINE 116   IF( NWING .EQ. ID(1) ) GO TO 22
LINE 117   IF( NWING .EQ. ID(1) ) GO TO 22
LINE 118   IF( NWING .EQ. ID(1) ) GO TO 22

-6.90-
LINE 119    IF( VST .LE. 3 ) NS = NST
LINE 120    25 CONTINUE
LINE 121    C
LINE 122    NSECT(JW) = NS
LINE 123    SYM(JW) = FLOAT(ISYM)
LINE 124    DD 26 I=1,NS
LINE 125    ICL(I,JW)=0
LINE 126    26 NSP(1,JW) = ID(165)
LINE 127    AR(JW) = FD(1)
LINE 128    C
LINE 129    CHECK ON SYMMETRY CODE
LINE 130    IF( IABS(ISYM) .LE. 1 ) GO TO 27
LINE 131    WRITE( I69, 2000 ) JW, NC, ISYM, SKODE(JW), NSECT(JW)
LINE 132    WRITE( I69, 2001 )
LINE 133    WRITE( I69, 2022 ) JW, ISYM
LINE 134    ERROR = .TRUE.
LINE 135    27 CONTINUE
LINE 136    C
LINE 137    IF( ISYM ) 28, 30, 32
LINE 138    28 CONTINUE
LINE 139    DD 29 I = 1,4
LINE 140    29 SKODE(I) = SK2(I)
LINE 141    GO TO 34
LINE 142    30 CONTINUE
LINE 143    DD 31 I = 1,4
LINE 144    31 SKODE(I) = SK3(I)
LINE 145    GO TO 34
LINE 146    32 CONTINUE
LINE 147    DD 33 I = 1,4
LINE 148    33 SKODE(I) = SK4(I)
LINE 149    C
LINE 150    34 CONTINUE
LINE 151    C
LINE 152    WRITE( I69, 3000 ) JW, NC, ISYM, SKODE, IPW(JW), NSECT(JW)
LINE 153    WRITE( I69, 3001 ) ( I, NSP(I,JW), I=1,NS )
LINE 154    WRITE( I69, 3002 ) AR(JW)
LINE 155    C
LINE 156    C
LINE 157    READ CORNER POINTS
LINE 158    C
LINE 159    DD 36 I=1,NC
LINE 160    READ( 15, 1000 ) LAB, ID, FD
LINE 161    IF( I .GT. 1 ) GO TO 35
LINE 162    IF( LAB .EQ. L3 ) GO TO 35
LINE 163    C
LINE 164    CORNER DATA CARD MISSING
LINE 165    WRITE( I69, 3003 ) JW
LINE 166    WRITE( I69, 3004 ) JW
LINE 167    ERROR = .TRUE.
LINE 168    C
LINE 169    C
LINE 170    CORN(Y1,JW) = FD(1)
LINE 171    CORN(Y1,JW) = FD(2)
LINE 172    CORN(Y1,JW) = FD(3)
LINE 173    36 CONTINUE
LINE 174    C
LINE 175    WRITE( I69, 3005 ) JW, ( I, CORN(X(I,JW), CORN(Y(I,JW)), CORN(Z(I,JW), I=1,NC)
LINE 176    IF( 2*(NC/2) .EQ. NC ) GO TO 37
LINE 177    CORN(X(NC1,JW)) = CORN(X(NC,JW)
LINE 178    CORN(Y(NC1,JW)) = CORN(Y(NC,JW)
LINE 179    CORN(Z(NC1,JW)) = CORN(Z(NC,JW)
LINE 180    37 CONTINUE
LINE 181 C CALL ROUTINE TO CHECK FOR PHYSICAL ERROR ON CORNER POINT
LINE 182 C CALL WINGCK(CORNX(1,JW),CORNY(1,JW),CORNZ(1,JW),JW,NL,ERROR,16)
LINE 183 DO 38 I=1,NC
LINE 184 38 CONTINUE
LINE 185 CALL WINGCK(CORNX(1,JW),CORNY(1,JW),CORNZ(1,JW),JW,NL,ERROR,16)
LINE 186 CONTINUE
LINE 187 GO TO 10
LINE 188 GO TO 10
LINE 189 40 CONTINUE
LINE 190 C SECTION TO PROCESS CONTROL LINE INFORMATION
LINE 191 IS = ID(1)
LINE 192 K = ICLI(IS,JW) & 1
LINE 193 ICLI(IS,JW) = K
LINE 194 IF( IS .LE. NSECT(JW) ) GO TO 41
LINE 195 WRITE( 16,2000 ) JW
LINE 196 WRITE( 16,2008 ) JW
LINE 197 WRITE( 16,2011 ) IS, JW, NSECT(JW)
LINE 198 ERROR = .TRUE.
LINE 199 41 CONTINUE
LINE 200 IF( K .LE. 2 ) GO TO 42
LINE 201 IF( K .EQ. ID(2) ) GO TO 43
LINE 202 IF( .NOT. CLINPT ) WRITE( 16,3002 ) JW
LINE 203 CLINPT = .TRUE.
LINE 204 WRITE( 16,3003 ) IS, K, FD(1), FD(2)
LINE 205 X1 = FD(1)/BEERFL
LINE 206 X2 = FD(2)/BEERFL
LINE 207 ICN = 2*IS - 1
LINE 208 IF( X1 .LT. CORNX(ICN,JW) ) GO TO 44
LINE 209 IF( X1 .GT. CORNX(ICN,JW) ) GO TO 44
LINE 210 IF( X2 .LT. CORNX(ICN,JW) ) GO TO 44
LINE 211 IF( X2 .GT. CORNX(ICN,JW) ) GO TO 44
LINE 212 GO TO 45
LINE 213 42 CONTINUE
LINE 214 43 CONTINUE
LINE 215 CONTROL LINE DATA OUT OF REFERENCE
LINE 216 WRITE( 16,2000 ) JW
LINE 217 WRITE( 16,2008 ) JW
LINE 218 WRITE( 16,2010 ) IS
LINE 219 ERROR = .TRUE.
LINE 220 44 CONTINUE
LINE 221 IF( .NOT. CLINPT ) WRITE( 16,3002 ) JW
LINE 222 CLINPT = .TRUE.
LINE 223 WRITE( 16,3003 ) IS, K, FD(1), FD(2)
LINE 224 X1 = FD(1)/BEERFL
LINE 225 X2 = FD(2)/BEERFL
LINE 226 ICN = 2*IS - 1
LINE 227 IF( X1 .LT. CORNX(ICN,JW) ) GO TO 44
LINE 228 IF( X1 .GT. CORNX(ICN,JW) ) GO TO 44
LINE 229 IF( X2 .LT. CORNX(ICN,JW) ) GO TO 44
LINE 230 IF( X2 .GT. CORNX(ICN,JW) ) GO TO 44
LINE 231 GO TO 45
LINE 232 45 CONTINUE
LINE 233 CONTROL LINE DEFINED OUTSIDE OF SECTION
LINE 234 WRITE( 16,2000 ) JW
LINE 235 WRITE( 16,2008 ) JW
LINE 236 WRITE( 16,2010 ) K, IS
LINE 237 ERROR = .TRUE.
LINE 238 46 CONTINUE
LINE 239 IF( K .EQ. 1 ) GO TO 46

-6.92-
LINE 243 IF( X1 .GE. CL(I1,IS, J1)) .AND. X2 .GE. CL(I2,IS, J2) GO TO 46
LINE 244 C SECOND CONTROL LINE DEFINED ABOVE FIRST
LINE 245 WRITE( 16, 2001 )
LINE 246 WRITE( 16, 2008 ) J
LINE 247 WRITE( 16, 2013 ) IS
LINE 248 ERROR = .TRUE.
LINE 249 CONTINUE
LINE 250 JCN = 2*K - 1
LINE 251 CL(JCN, IS, J1) = X1
LINE 252 CL(JCN, IS, J2) = X2
LINE 253 GO TO 10
LINE 254 C
LINE 255 50 CONTINUE
LINE 256 C FREQUENCY CARD READ
LINE 257 DO 55 I = 1, 4
LINE 258 55 FREQ(I) = FD(I)
LINE 259 NFREQ = ID(I)
LINE 260 IF( NFREQ .GT. 4 ) READ( 15, 1001 ) ( FREQ(I), I=5, NFREQ )
LINE 261 WRITE( 16, 2006 ) NFREQ, ( FREQ(I), I=1,NFREQ )
LINE 262 FREQR = .TRUE.
LINE 263 GO TO 10
LINE 264 C
LINE 265 60 CONTINUE
LINE 266 C END CARD READ
LINE 267 C MODE DATA IS MISSING
LINE 268 WRITE( 16, 2007 )
LINE 269 IF( .NOT. FREQR ) WRITE( 16, 2015 )
LINE 270 WRITE( 16, 2014 )
LINE 271 ERROR = .TRUE.
LINE 272 RETURN
LINE 273 C
LINE 274 65 CONTINUE
LINE 275 IF( FREQR ) RETURN
LINE 276 WRITE( 16, 2000 )
LINE 277 WRITE( 16, 2015 )
LINE 278 ERROR = .TRUE.
LINE 279 RETURN
LINE 280 C
LINE 281 70 CONTINUE
LINE 282 C LABEL CARD EXPECTED BUT NOT RECEIVED
LINE 283 WRITE( 16, 2002 )
LINE 284 WRITE( 16, 2016 )
LINE 285 WRITE( 16, 2017 ) LAB, ID, FD
LINE 286 ERROR = .TRUE.
LINE 287 WRITE( 16, 2020 )
LINE 288 C
LINE 289 80 CONTINUE
LINE 290 C GRID CARD READ
LINE 291 I = ID(2)
LINE 292 J = ID(1)
LINE 293 IF( I .GT. NSECT(J) ) GO TO 81
LINE 294 IF( J .GT. NHING ) GO TO 81
LINE 295 ICL(I,J) = -1
LINE 296 RETURN
LINE 297 C
LINE 298 81 CONTINUE
LINE 299 C GRID DATA SPECIFIED FOR SECTION THAT DOES NOT EXIST
LINE 300 WRITE( 16, 2000 )
LINE 301 WRITE( 16, 2020 )
LINE 302 WRITE( 16, 2021 ) J, I
LINE 303 ERROR = .TRUE.
LINE 304 C
LINE 305 1000 FORMAT (A4, 2X, 912, 4E12.0 ) 04303050
LINE 306 1001 FORMAT (2X, 4E12.0 ) 04303060
LINE 307 C 04303070
LINE 308 2030 FORMAT /// 1X, 1291H4: - ERROR IN READING INPUT. ) 04303080
LINE 309 2031 FORMAT(29H WING DATA CARD IS INCORRECT ) 04303090
LINE 310 2002 FORMAT (14H DATA FOR WING, 12, 28H WAS READ WHEN DATA FOR WING, 13, 4303100
LINE 311 1 28H WAS EXPECTED BY THE PROGRAM // ) 04303110
LINE 312 2003 FORMAT (92H WING NUMBER MUST BE GREATER THAN ZERO AND LESS THAN 0 ) 04303120
LINE 313 1 EQUAL TO 4, IT WAS READ AS, 12 /55H IT WILL BE TREATED AS WING 1 04303130
LINE 314 2FOR CHK CHECK PURPOSE ONLY / ) 04303140
LINE 315 2004 FORMAT (47H NO MORE THAN FOUR WING DATA CARDS MAY BE INPUT ) 04303150
LINE 316 2005 FORMAT (34H NUMBER OF SECTIONS INPUT FOR WING, 12, 3H IS, 12 / 04303160
LINE 317 1 47H THIS IS GREATER THAN THE MAXIMUM PER WING OF 3 ) 04303170
LINE 318 2006 FORMAT (24H NUMBER OF CORNER POINTS, 12, 19H IS NOT CONSISTANT , 04303180
LINE 319 1 27H THE NUMBER OF SECTIONS, 12, 9H FOR WING, 12 ) 04303190
LINE 320 2007 FORMAT(36H 'CORNER' DATA CARD MISSING FOR WING, 12 ) 04303200
LINE 321 2008 FORMAT (43H ERROR ON CONTROL 'LINE' DATA CARD FOR WING, 12 ) 04303210
LINE 322 2009 FORMAT(54H MORE THAN TWO CONTROL LINES ARE SPECIFIED FOR SECTION, 04303220
LINE 323 2010 FORMAT(49H CONTROL LINES ARE NOT INPUT IN CORNER FOR SECTION, 12, 12H THIS IS ABOVE THE FIRST ) 04303230
LINE 324 2011 FORMAT(44H CONTROL LINE 2 OF SECTION, 12, 12H IS ABOVE THE FIRST ) 04303240
LINE 325 2012 FORMAT(37H MODE DATA IS MISSING OR OUT OF ORDER ) 04303250
LINE 326 2013 FORMAT(39H FREQUENCY DATA IS MISSING OR OUT OF ORDER ) 04303260
LINE 327 2014 FORMAT(39H FREQUENCY DATA IS MISSING OR OUT OF ORDER ) 04303270
LINE 328 2015 FORMAT (38H CONTROL LINE IS SPECIFIED FOR SECTION, 12, 12H OF WING, 12 ) 04303280
LINE 329 2016 FORMAT (37H CONTROL LINE, 12H FOR SECTION, 12H IS NOT DEFINE ) 04303290
LINE 330 C 04303300
LINE 331 3000 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303310
LINE 332 3001 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303320
LINE 333 3002 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303330
LINE 334 C 04303340
LINE 335 3003 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303350
LINE 336 3004 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303360
LINE 337 3005 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303370
LINE 338 3006 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303380
LINE 339 3007 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303390
LINE 340 3008 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303400
LINE 341 3009 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303410
LINE 342 3010 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303420
LINE 343 C 04303430
LINE 344 3011 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303440
LINE 345 3012 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303450
LINE 346 3013 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303460
LINE 347 3014 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303470
LINE 348 3015 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303480
LINE 349 3016 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303490
LINE 350 3017 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303500
LINE 351 3018 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303510
LINE 352 3019 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303520
LINE 353 3020 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303530
LINE 354 3021 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303540
LINE 355 3022 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303550
LINE 356 3023 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303560
LINE 357 3024 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303570
LINE 358 3025 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303580
LINE 359 3026 FORMAT (1H67 CARD WILL BE PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. ) 04303590
LINE 360 END
OF

CIS PLAY ... F1L E 	 DECK 44

LINE 1 SUBROUTINE MINTGRK X, Y

LINE 2 COMPLEX X, Y

LINE 3 INTEGER GOBACK

LINE 4 LOGICAL EKEND, EMIRRO, TRAILR

LINE 5 LOGICAL ERROR

LINE 6 LOGICAL MIRROR

LINE 7 LOGICAL TEKRD

LINE 8 LOGICAL TRAINF

LINE 9 LOGICAL TREDGE, WAKE

LINE 10 DIMENSION CAR(12), CX(12)

LINE 11 DIMENSION RK(12,30)

LINE 12 DIMENSION SCA(30)

LINE 13 DIMENSION HEAD(20)

LINE 14 DIMENSION WAVE(20)

LINE 15 DIMENSION XI(11), YI(11), WROW(11)

LINE 16 DIMENSION MACH, KAPPA, IPRT(5)

LINE 17 COMM / EEW / XUP, YUP, XLE, YLE, EREUNE, EMIRRO, KR, RL, RTK, RKL

LINE 18 COMM / FREQ / CF(12,30), FREQ(12), FREQ(12), FREQ(12)

LINE 19 COMM / RLLL / TVW(4), YRU, YLO, ZR, ZRRZ, ZL, ZLZL, RKL, T1N

LINE 20 COMM / WAKE / Tm(1), Tm(1), ICV, WPS, ICVHV, LAMHT, INFE

LINE 21 COMM / WAKE / Tm(12), Tm(12), ICVHV, LAMHT, INFE

LINE 22 COMM / WAVE / RIV, RIV, YRU, YLO, ZR, ZRRZ, ZL, ZLZL, RKL, T1N

LINE 23 COMM / WAVE / X, Y, ZD, ZD, ZD, SYM, NINZT

LINE 24 COMM / ZPDZU / 15, 16

LINE 25 DATA KCONV / 0 / 0

LINE 26 DATA AKTEAM / 0.221, 0.645, 1.182, 1.007, 2.461, 3.14, 3.835, 4.541, 5.255, 5.974, 6.697, 7.42, 8.15, 8.88, 9.61, 10.31

LINE 27 C X COORD. OF POINTS OF INFLUENCING ELEMENT

LINE 28 C Y COORD. OF POINTS OF INFLUENCING ELEMENT

LINE 29 C X COORD. OF RECEIVING PT. CENTER OF ELEMENT

LINE 30 C Y COORD. OF RECEIVING PT. CENTER OF ELEMENT

LINE 31 C ZD DIFFERENCE BETWEEN Z'S OF INFL. AND RECEIVED ELEM.

LINE 32 C ZD IS THE SQUARE OF THE DIFFERENCE OF THE RECEIVED AND INFL. ELEMENTS

LINE 33 C TVW TRANSFORMATION TO COMBINE WAVE AND SIDEWASH

LINE 34 C VTERM NO. TERMS IN APPROXIMATION OF VELOCITY INTEGRALS

LINE 35 C SYM SYMMETRY CODE ( 0 NO SYM, 1 SYM )

LINE 36 C SYM SYMMETRY CODE ( 0 NO SYM, 1 SYM )

LINE 37 C SYM SYMMETRY CODE ( 0 NO SYM, 1 SYM )

LINE 38 C SYM SYMMETRY CODE ( 0 NO SYM, 1 SYM )

LINE 39 C SYM SYMMETRY CODE ( 0 NO SYM, 1 SYM )

LINE 40 C ERROR LOGICAL SET EQUAL TO TRUE IF AN ERROR IS DETECTED

LINE 41 C HROW AN ARRAY OF THE FREQUENCY INDEPENDENT TERMS OF THE VELOCITY INTEGRALS FOR A GIVEN MACH ELEM AND INFL ELEM.

LINE 42 C HROW AN ARRAY OF THE FREQUENCY INDEPENDENT TERMS OF THE VELOCITY INTEGRALS FOR A GIVEN MACH ELEM AND INFL ELEM.

LINE 43 C IFIRST TRUE IF ELEMENT IS THE FIRST ELEMENT OF INFL. SPAN

LINE 44 C IFIRST TRUE IF ELEMENT IS THE FIRST ELEMENT OF INFL. SPAN

LINE 45 C MAKE1 LOGICAL SET TO TRUE IF WAVE ELEM HAS INFLUENCE

LINE 46 C TREDGE LOGICAL INPUT AS TRUE IF INFL. ELEM. IS THE TRAILING ELEMENT

LINE 47 C TREDGE LOGICAL INPUT AS TRUE IF INFL. ELEM. IS THE TRAILING ELEMENT

LINE 48 C ATES/J COMPLEX ARRAY OF INFLUENCE OF WAVE EFFECT FOR .FREQUENCY J

LINE 49 C ATES/J COMPLEX ARRAY OF INFLUENCE OF WAVE EFFECT FOR .FREQUENCY J

LINE 50 C ATES/J COMPLEX ARRAY OF INFLUENCE OF WAVE EFFECT FOR .FREQUENCY J

LINE 51 C ATES/J COMPLEX ARRAY OF INFLUENCE OF WAVE EFFECT FOR .FREQUENCY J

LINE 52 C ATES/J COMPLEX ARRAY OF INFLUENCE OF WAVE EFFECT FOR .FREQUENCY J

LINE 53 C ATES/J COMPLEX ARRAY OF INFLUENCE OF WAVE EFFECT FOR .FREQUENCY J

LINE 54 C ATES/J COMPLEX ARRAY OF INFLUENCE OF WAVE EFFECT FOR .FREQUENCY J

LINE 55 C ATES/J COMPLEX ARRAY OF INFLUENCE OF WAVE EFFECT FOR .FREQUENCY J

LINE 56 C ATES/J COMPLEX ARRAY OF INFLUENCE OF WAVE EFFECT FOR .FREQUENCY J

-6.92-
BEGIN DOCUMENT

LINE 57 C TEST LEADING EDGE LINE
LINE 58 C YLW = Y(1)
LINE 59 C YUP = Y(2)
LINE 60 C XLW = X(1)
LINE 61 C XUP = X(2)
LINE 62 C CALL EODNE
LINE 63 C IF ELEMT IS NOT IN MAC4, CONUE.
LINE 64 C PROCEES ANYWAY IF IT IS A SELF ELEMENT (DIAGONAL TERM)
LINE 65 C IF ELEMNT OR ILN.EQ. IAC) GO TO 20
LINE 66 C RETURN
LINE 67 C CONTINUE
LINE 68 C ASSIGN 310 TO IWAKE1
LINE 69 C ASSIGN 160 TO IWAKE2
LINE 70 C NTERM = 1
LINE 71 C RRU = RR
LINE 72 C RLU = RL
LINE 73 C MIRROE = EMIRRO
LINE 74 C TRAINF = .FALSE.
LINE 75 C IWAKE1 = .FALSE.
LINE 76 C IF NOT. TRGUE ) GO TO 25
LINE 77 C X1W = X1(1)
LINE 78 C XUP = X(2)
LINE 79 C IF .NOT. EREONE (.TRUE.
LINE 80 C GO TO 40
LINE 81 C CONTINUE
LINE 82 C IF .NOT. EREONE ) GO TO 40
LINE 83 C TRAINF = .TRUE.
LINE 84 C IF ( LATE .NE. 0 ) AND. RTR .GT. 0 ) IWAKE1 = .TRUE.
LINE 85 C CONTINUE
LINE 86 C YLW = Y(1)
LINE 87 C YUP = Y(2)
LINE 88 C XLW = X(1)
LINE 89 C XUP = X(2)
LINE 90 C CALL EODNE
LINE 91 C CALL EODNE
LINE 92 C YLW = Y(1)
LINE 93 C YUP = Y(2)
LINE 94 C XLW = X(1)
LINE 95 C XUP = X(2)
LINE 96 C IF . NOT. EREONE ) GO TO 40
LINE 97 C TRAINF = .TRUE.
LINE 98 C IF ( LATE .NE. 0 ) AND. RTR .GT. 0 ) IWAKE1 = .TRUE.
LINE 99 C CONTINUE
LINE 100 C YLW = Y(1)
LINE 101 C YUP = Y(2)
LINE 102 C XLW = X(1)
LINE 103 C XUP = X(2)
LINE 104 C IF ( LATE .NE. 0 ) AND. RTR .GT. 0 ) IWAKE1 = .TRUE.
LINE 105 C CONTINUE
LINE 106 C DO 30 I=1,NFREQ
LINE 107 C WREDW1) = -WLEAD(I)
LINE 108 C CONTINUE
LINE 109 C DO 45 I=1,NFREQ
LINE 110 C WLEAD(I) = 0.0
LINE 111 C CONTINUE
LINE 112 C NINS1D = NINS1D + 1
LINE 113 C CONTINUE
LINE 114 C YLW = Y(1)
LINE 115 C Z0 = ZER
LINE 116 C Z12D = Z012R
LINE 117 C TV = TV(1)
LINE 118 C CONTINUE

END DOCUMENT
THE RIGHT HAND ELEMENT ONLY AT THIS TIME

ASSIGN 320 TO IMAKE1

ASSIGN 320 TO IMAKE2

YLW = Y(2)

YUP = Y(1)

XLW = X(3)

XUP = X(4)

YO = YK0

ZD = ZR

ZDZD = ZRZR

TV = TVN(1)

TW = TVN(2)

SYM = -1.0

MIRRA = EMIRRO

DO 315 J=1,NFREQ

WTES(J) = (0.0, 0.0).

WSAVE(J) = WLEAD(J)

IF (FR TEST .EQ. 0.0) GO TO 145

RK = RR*FRTEST

ASSIGN 145 TO GJBACK

GO TO 600

Z1 = XO - XLW

Z2 = XU - XUP

E1 = YO - YLW

E2 = YO - YUP

Z3P = SQR(T E2 & ZDZO )

Z4P = SQR(T E1 & ZDZO )

ZTE = 0.5*(Z1GZ2)

ZLW = 0.5*(Z3PZ4P)

DIVIDE Z1S=ZU-ZL INTO A NUMBER OF DIVISIONS

ZDIS = ZTE*ZLh

DS = E2 - E1

NW = ZDIS/DS

IF (NW .EQ. 0.0) NW = 1

DELTAZ = ZDIS/NW

ZSMTE = -0.5*DELTALZ

IF (NW = 1) WE ARE DONE

IF ( NW .EQ. 1 ) GO TO 370
L I N E  3 0 5  3 7 5  W L E A D ( I )  =  0 . 0
L I N E  3 0 6  C  0 4 4 0 3 0 5 0
L I N E  3 0 7  3 1 7  X L W  =  X ( 3 )
L I N E  3 0 8  3 1 8  X U P  =  X ( 4 )
L I N E  3 0 9  3 1 9  Y L H  =  Y ( 2 )
L I N E  3 1 0  3 1 9  Y U P  =  Y ( 1 )
L I N E  3 1 1  3 1 0  Y O  =  Y L O
L I N E  3 1 2  3 1 2  Z D  =  Z L
L I N E  3 1 3  3 1 3  Z U Z D  =  Z L Z L
L I N E  3 1 4  3 1 4  T V  =  T V H ( 3 )
L I N E  3 1 5  3 1 5  T W  =  T V H ( 4 )
L I N E  3 1 6  3 1 6  S Y M  =  S Y M X
L I N E  3 1 7  3 1 7  I F ( F R T E S T . E Q . 0 . 0 )  G O  T O  1 3 5
L I N E  3 1 8  3 1 8  R K  =  X L F R T E S T
L I N E  3 1 9  3 1 9  A S S I G N  1 8 5  T O  G O B A C K
L I N E  3 2 0  3 2 0  G O  T O  6 0 0
L I N E  3 2 1  C
L I N E  3 2 2  3 8 0  C O N T I N U E
L I N E  3 2 3  C
L I N E  3 2 6  C  P U T  W L E A D  B A C K  T O  W S A V E  A N D  R E T R U N
L I N E  3 2 7  C
L I N E  3 2 8  3 2 8  D O  3 9 0  I =  1 , N F R E Q
L I N E  3 2 9  3 9 0  W L E A D ( I )  =  W S A V E ( I )
L I N E  3 3 0  C  R E T U R N
L I N E  3 3 1  C
L I N E  3 3 2  C
L I N E  3 3 3  C  0 4 4 0 3 3 3 0
L I N E  3 3 4  C  0 4 4 0 3 3 4 0
L I N E  3 3 5  6 0 0  C O N T I N U E
L I N E  3 3 7  C  I F ( K  . G T .  2 0 . 5 )  G O  T O  6 1 5
L I N E  3 3 8  D O  G I 6 1 0  N T E R M  =  6 , N T M X X
L I N E  3 4 0  6 1 0  C O N T I N U E
L I N E  3 4 1  6 1 5  C O N T I N U E
L I N E  3 4 3  N C V F L  =  N C V E L  &  1
L I N E  3 4 4  N T E R 4  =  N T M X X
L I N E  3 4 5  6 2 0  C O N T I N U E
L I N E  3 4 6  N T E R 4  =  M I N O F ( N T E R M , N T M X X )
L I N E  3 4 9  G O  T O  G O B A C K , ( 1 4 5 , 1 8 5 , 3 3 0 )
L I N E  3 5 0  C  0 4 4 0 3 4 9 0
L I N E  3 5 1  C
L I N E  3 5 2  C  0 4 4 0 3 5 1 0
L I N E  3 5 3  8 0 0  C O N T I N U E
L I N E  3 5 4  C
L I N E  3 5 5  D O  8 1 0  I =  1 , N F R E Q
L I N E  3 5 6  8 1 C  S K A ( I )  =  0 . 0
L I N E  3 5 7  C
L I N E  3 5 8  N T M 1  =  M A X O F ( N T E R M - 1 , 1 )
L I N E  3 5 9  D O  8 2 0  J =  1 , N T M 1
L I N E  3 6 0  S A V E  =  T V ( V J J ) &  T W ( W J J )
L I N E  3 6 1  D O  8 2 0  I =  1 , N F R E Q
L I N E  3 6 2  S K A ( I )  =  S K A ( I )  &  S A V E * C F ( I , J )
L I N E  3 6 3  8 2 0  C O N T I N U E
L I N E  3 6 4  C  I F ( N T E R M . E Q . 1 )  G O  T O  G O B A C K , ( 1 5 0 , 1 9 0 , 3 4 5 )
L I N E  3 6 6  C  0 4 4 0 3 6 6 0
L(NE 367 SAVE = TV$M(INTERM) & TV$M(INTERM))
LINE 368 TO 630 I=1,1NFREN
LINE 369 SKL = SKAI(I)
LINE 370 IF( KCONV .GT. 1000 ) GO TO 330
LINE 371 DIK = ABS( SKAI(I) - SKL )
LINE 372 RER = 1.0
LINE 373 IFLUSH = 1
LINE 374 IF( SKAI(I) .NE. 0.0 ) RERD = DIK/ABS(SKAI(I))
LINE 375 IF( RERD .LT. 0.35 ) GO TO 830
LINE 376 IF( RERD .LT. 0.0 AND. DIK .LT. .1E-4 ) GO TO 330
LINE 377 C WE HAVE NOT CONVERGED
LINE 378 IF( KCONV .EQ. 0 ) WRITE ( 16, 4009 )
LINE 379 WRITE ( 15, 4010 ) KCONV,IRC,ITIN,INTERM,FREU(I),SKAI(I),SKL,RERO
LINE 380 IF( KCONV .LE. 0.0 ) GO TO 830
LINE 382 C WRITE ( 16, 4011 )
LINE 383 C IFLUSH = 1
LINE 384 830 CONTINUE
LINE 385 C GO TO GOBACK, ( 150, 190, 340 )
LINE 386 C 4009 FORMAT ( IHL, 120(I*) ) // 10X, 85H THE VELOCITY INFLUENCE COMPUTATIONS DID NOT CONVERGE FOR AT LEAST ONE PAIR OF ELEMENTS // 15X,
LINE 387 4010 FORMAT ( IHL, 120(I*) ) // 10X, 85H CONVERGENCE TESTING HAS BEEN TERMINATED FOR THIS RUN. //
/ END REAC

END CF JOB.

43.1 SEC. USED . 012 HRS. CHARGED 49.906 HRS. REMAINING
DISPLAY...FILE DECK 45

LINE 1  SUBROUTINE WRETA( NET, JMODE, M, WETA, 113) 04500010
LINE 2  DIMENSION WETA(NET,10), WDETA(NET,10) 04500020
LINE 3  REWIND 113 04500030
LINE 4  DO 30 JM=1, JMODE 04500040
LINE 5  WRITE (113) JM, NET, (WDETA(I,JM), I=1,NET), (WETA(I,JM), I=1,NET) 04500050
LINE 6  30 CONTINUE 04500060
LINE 7  END FILE 113 04500070
LINE 8  REWIND 113 04500080
LINE 9  RETURN 04500090
LINE 10 END 0450100
SUBROUTINE WRITEO (K, AROW, IZ, ANEW, NET)

DIMENSION AROW(NET), IZ(NET), ANEW(NET)

COMMON /KST11,KST12, KST11

COMM /TAPE/ 19,110,111

DO 100 I = 1, NET

IF (AROW(I) .EQ. 0.0) GO TO 100

M = NNZ

IZ(M) = I

ANEW(M) = AROW(I)

100 CONTINUE

IF (NNZ .EQ. 0) RETURN

CALL WRITZ(111,K,NNZ,IZ,ANEW)

KST11 = K

RETURN

END
DISPLAY...FILE
DECK 47

LINE 1 SUPPLEMENT WRITZ(III), K, NNZ, IZ, ANEW
LINE 2 DIMENSION IZ(NNZ), ANEW(ANZ)
LINE 3 WRITE (III) K, NNZ, IZ, ANEW
LINE 4 RETURN
LINE 5 END

04700010
04700020
04700030
04700040
04700050
CISPLAY...FILE

DECK 48

LINE 1   SUBROUTINE WRITW (112, K, NNZ, IZ, ANEW) 04000010
LINE 2   DIMENSION IZ(NNZ) 04000020
LINE 3   COMPLEX ANEW(1NZ) 04000030
LINE 4   WRITE (112) K, NNZ, IZ, ANEW 04000040
LINE 5   200 CONTINUE 04000050
LINE 6   RETURN 04000060
LINE 7   END 04000070
CISPLAY...FILE

DECK 49

LINE 1   SUBROUTINE WTEPHT (DEDT, NET, PHIW, IZ, I12, WTE) 04900010
LINE 2   COMPLEX DEDT(1), PHIW(1), WTE(1) 04900020
LINE 3   DIMENSION IZ(1) 04900030
LINE 4   COMMON /K1112/ KST11, KST12 04900040
LINE 5   KEL = 0 04900050
LINE 6   DO 300 IEL = 1, NET 04900060
LINE 7   IF (IEL .GT. KST12) GO TO 300 04900070
LINE 8   IF (KEL .LT. IEL) 04900080
LINE 9   IREAD (I12) KEL, NNZ, (IZ(I), I=1,NNZ), (WTE(I), I=1,NNZ) 04900090
LINE 10  204 IF (KEL .NE. IEL) GO TO 300 04900100
LINE 11  DO 200 M = 1, NNZ 04900110
LINE 12  K = IZ(M) 04900120
LINE 13  I = IEL 04900130
LINE 14  DEDT(I) = DEDT(I) - WTE(M) * PHIW(K) 04900140
LINE 15  CONTINUE 04900150
LINE 16  300 CONTINUE 04900160
LINE 17  REWIND 112 04900170
LINE 18  RETURN 04900180
LINE 19  END 04900190
SUBROUTINE WWINY

XO, YO COORDINATES OF RECEIVING PT. (CENTER OF REC. ELE.)
XUP, YUP COORDINATES OF UPPER END OF LINE
XLW, YLW COORDINATES OF LOWER END OF LINE
ZDZD IS THE SQUARE OF THE DIFFERENCE OF THE REC AND INFL Z'S
ZD DIFFERENCE BETWEEN Z'S OF INFLU. AND RECEIV. ELEM.
V, W SIDE WASH AND UNPASH INTEGRALS
VTERM NUMBER OF TERMS IN APPROXIMATION OF V AND W

REAL L1
DIMENSION GETA(5), GCOF(5), XX(31), AG(3)
DIMENSION H(31)
COMMON / EEW / XUP, YUP, XLW, YLW
COMMON / W1 / VTERM, V(30)
COMMON / WV1 / XO, YO, ZO, ZDZD
DATA NGT / 5 /

LUP = XO — XUP
ZLW = XO — XLW
EUP = YO — YUP
ELW = YO — YLW
ZZUP = ZUP*ZUP
ZZLW = ZLW*ZLW
EEUP = EUP*EUP
EE-LW = ELW*ELW
SLOPE = (XUP — XLW)/(YUP — YLW)
ALPHA = ZLW — SLOPE*ELW
AA = ALPHA*ALPHA
TTUP = EEUP + ZDZD
TTLW = EE-LW + ZDZD
RR UP = ZZUP — TTUP
RRLW = ZZLW — TTLW
IF( RRRP .LT. 0.0 .OR. ZUP .LE. 0.0 ) RRUP = 0.0
IF( RRLW .LT. 0.0 .OR. ZLW .LE. 0.0 ) RRLW = 0.0
RRUP = SQRT(RRUP)
RRLW = SQRT(RRLW)
SS = SLOPE*SLOPE
TESTC = SS — 1.0
AMZ = AA .AND. SSZDZD
ACZ = AMZ — ZDZD
SIGNU = SIGN(P102, EUP)
SIGNL = SIGN(P102, ELW)
SZZ = SLOPE*ZDZD
SORC = ABS(TESTC)
IF( SORC .LT. 0.001 ) GO TO 20
SORC = SQRT(SORC)
SACZ = SQRT(ACZ)
IF( TESTC .LT. 0.0 ) GO TO 22
LINE 57 C  
LINE 58 FU = SIGNU 
LINE 59 IF(RUP .GT. 0.0) FU = -SIGNU (SLOPE*ZUP-EUP)/SQACZ 
LINE 60 FL = SIGNL 
LINE 61 IF( RLW .GT. 0.0) FL = -SIGNL (SLOPE*ZLW-ELW)/SQACZ 
LINE 62 FO = -(FL - FL)/SQRC 
LINE 63 GO TO 30 
LINE 64 20 CONTINUE 
LINE 65 C C = 0 
LINE 66 FO = (RUP - RLW) / (SLOPE*ALPHA) 
LINE 67 GO TO 30 
LINE 68 22 CONTINUE 
LINE 69 FU = SQRC*RUP & SLOPE*ZUP - EUP 
LINE 70 FL = SQRC*RLW & SLOPE*ZLW - ELW 
LINE 71 IF(RLW .EQ. 0.0) GO TO 26 
LINE 72 IF(RUP .EQ. 0.0) GO TO 24 
LINE 73 C C .GT. 0 NEITHER RUP NON RLW IS ZERO 
LINE 74 FO = (ALOG(FU/FL) / SQRC 
LINE 75 GO TO 30 
LINE 76 23 CONTINUE 
LINE 77 24 CONTINUE 
LINE 78 FO = (ALOG(SQACZ/FL)) / SQRC 
LINE 79 GO TO 30 
LINE 80 26 CONTINUE 
LINE 81 FO = (ALOG(SQACZ/FL)) / SQRC 
LINE 82 C FO HAS BEEN COMPUTED 
LINE 83 30 CONTINUE 
LINE 84 C FIND ALL (ALL IS CALLED L1 IN THE NOTES) 
LINE 85 ALU = 0.0 
LINE 86 ALL = ZUP - RUP 
LINE 87 IF(RUP .GT. 0.0) AND ALL .GT. 1E-5 
LINE 88 1 ALU = 0.5*ALOG((ZUP+RUP)/ALL) 
LINE 89 2 ALU = 0.0 
LINE 90 ALL = ZLW - RLW 
LINE 91 IF(ALW .EQ. 0.0) GO TO 34 
LINE 92 2 ALL = 0.5*ALOG((ZLW+RLW)/ALL) 
LINE 93 1 ALL = ALU - ALL 
LINE 94 C FIND S1 
LINE 95 S1 = 0.0 
LINE 96 IF(ZD .EQ. 0.0) GO TO 40 
LINE 97 FL = ALPHA*ELW - SZZ 
LINE 98 SL = SQRT(TTLW*ACZ) 
LINE 99 FU = ALPHA*EUP - SZZ 
LINE 100 IF(RUP .EQ. 0.0) GO TO 32 
LINE 101 SU = SQRT(TTUP*ACZ) 
LINE 102 IF( RLW .EQ. 0.0) GO TO 34 
LINE 103 C S1 FOR RUP .GT. 0, RL .GT. 3 AND ZD .NE. 0 
LINE 104 S1 = ARSIN(FU/SU) - ARSIN(FL/SL) 
LINE 105 GO TO 40 
LINE 106 32 CONTINUE 
LINE 107 S1 = SIGN(PIDZ, FU) - ARSIN(FL/SL) 
LINE 108 GO TO 40 
LINE 109 34 CONTINUE 
LINE 110 S1 = ARSIN(FU/SU) - SIGN(PIDZ, FL) 
LINE 111 40 CONTINUE 
LINE 112 C F1 AND F2 
LINE 113 C F1 = (ALPHA*ZD*S1/AMZ) & SZZ*AL1/AMZ 
LINE 114 C F2 = SLOPE*F1 - AL1 
LINE 115 C W(1) = RUP*EUP/TTUP - RLW*ELW/TTLW - TESTC*FO & SS*FL - SLOPE*F2 
LINE 116 40 CONTINUE 
LINE 117 C W(1) = RUP*EUP/TTUP - RLW*ELW/TTLW - TESTC*FO & SS*FL - SLOPE*F2
LINE 119 IF ( NTERM .LE. 1 ) GO TO 71
LINE 120 C
LINE 121 W(2) = 0.5*( (ZUP*ALPHA)*EUP - (ZLW*ALPHA)*ELW )
LINE 122 1 + 0.5*(ZUP*EUP - (ZLW*ELW) - 0.5*(AA-3)*TESTC*ZLZD)*FD
LINE 123 2 - (1.5*SZZ-ZW*AA)*F1 & 3.5*SZZ*F2
LINE 124 3
LINE 125 C
LINE 126 IF ( NTERM .LE. 2 ) GO TO 71
LINE 127 C
LINE 128 NHT = NTERM - 2
LINE 129 IWT = NTERM - 1
LINE 130 C
LINE 131 C FIND 5 PT. GAUSS-LEFENORE POINTS AND COEFFICIENTS (GETA,GCOF)
LINE 132 F1 = 0.5*(EUP - ELW)
LINE 133 F2 = 0.5*(ELW & EUP)
LINE 134 DO 43 IG=1,3
LINE 135 N = 6 - IG
LINE 136 F0 = F1**XX(IG)
LINE 137 GETA(IG) = F0 & F2
LINE 138 GETA(IG) = F0 & F2
LINE 139 GCOF(IG) = F1*AG(IG)
LINE 140 43 GCOF(N) = GCOF(IG)
LINE 141 C
LINE 142 DO 45 N=3,NTERM
LINE 143 V(N) = 0.0
LINE 144 W(N) = 0.0
LINE 145 45 W(N) = 0.0
LINE 146 C
LINE 147 C LOOP ON TERMS OF QUADRATURE
LINE 148 C
LINE 149 DO 70 IG=1,NHT
LINE 150 C
LINE 151 ETA = GETA(IG)
LINE 152 ZETA = SLOPE*ETA & ALPHA
LINE 153 IF( ZETA .LE. 0.0 ) GO TO 70
LINE 154 TTHH = ETA*ETA & ZLZD
LINE 155 RR = ZETA*ZETA - TTHH
LINE 156 IF ( RR .LE. 0.0 ) GO TO 70
LINE 157 GCOF = GCOF(IG)
LINE 158 RH = SORT( RR )
LINE 159 FL = ZETA - RH
LINE 160 ALH = 0.0
LINE 161 IF ( FL/ZETA ) .GT. 1.E-6 ALH = 0.5*ALOG1 ( ZETA/RH)/FL
LINE 162 C LOOP TO FIND NHT (=NTERM) H TERMS
LINE 163 H(1) = ZETA*ALH - RH
LINE 164 DO 50 N=1,NHT
LINE 165 NH = N61
LINE 166 N2 = 2*N
LINE 167 RH = RH*MR
LINE 168 H(NH) = 9*(N2*(N2+1)) - (N2-1)*TTHH*H(NH)/N2
LINE 169 50 CONTINUE
LINE 170 C
LINE 171 C FIND CONTRIBUTION OF UPWASH FOR A GIVEN PT. (FOR A GIVEN IG)
LINE 172 DO 60 N=2,1W
LINE 173 NW = N61
LINE 174 W(NW) = W(N) - GCOF(2*N-1)*H(N) - (2*(N-1)-1)*ZLZD*H(N-1)
LINE 175 60 CONTINUE
LINE 176 C
LINE 177 70 CONTINUE
LINE 178 C
LINE 179 C
LINE 180 71 CONTINUE

-6.109-
LINE 181 C
LINE 182 V(1) = 0.0
LINE 183 V(2) = 0.0
LINE 184 C
LINE 185 C
LINE 186 C IF ZD = 0, THERE IS NO SIDEWASH
LINE 187 IF( ZD .EQ. 0.0 ) RETURN
LINE 188 C
LINE 189 C SECTION FOR SIDEWASH
LINE 190 C
LINE 191 V(1) = V1 = ZD*( RUP/TTUP - RLW/TTLW )
LINE 192 IF( NTERM .LE. 1 ) GO TO 90
LINE 193 HU = ZUP*ALU - RUP
LINE 194 HL = ZLW*ALL - RLW
LINE 195 V(2) = V2 = ZD*( HU - HL )
LINE 196 IF( NTERM .EQ. 2 ) GO TO 90
LINE 197 RSU = RUP
LINE 198 RSL = RLW
LINE 199 C
LINE 200 DO 80 N=2,IWT
LINE 201 NV = N&1
LINE 202 NH = N-1
LINE 203 C
LINE 204 C NOTE THAT H IS COMPUTED FOR NH = N-1
LINE 205 IF( RUP .EQ. 0.0 ) GO TO 74
LINE 206 RSU = RSL*RRUP
LINE 207 HU = RSU/(N2*(N2-1)) - (N2-1)*TTUP*HU/N2
LINE 208 74 CONTINUE
LINE 209 IF( RLU .EQ. 0.0 ) GO TO 76
LINE 210 RSL = RSL*RLW
LINE 211 HL = RSL/(N2*(N2-1)) - (N2-1)*TTLW*HL/N2
LINE 212 76 CONTINUE
LINE 213 V(NVI) = - ZD*(2*H-1)*( HU - HL )
LINE 214 80 CONTINUE
LINE 215 C
LINE 216 90 CONTINUE
LINE 217 RETURN
LINE 218 END
CISPLAY...FILE	DECK 51

LINE 1
SUBROUTINE ZFDZI (ZFUN, DZDX, X, Y, A)

LINE 2
SUBROUTINE TO EVALUATE FUNCTION (ZFUN), AND DERIVATIVE (DZDX)

LINE 3
DIMENSION X(I), Y(I), A(I)

LINE 4
COMMON /ZFUNY/ N,IEHF, B1, R2, R3, AFUNMX

LINE 5
COMMON /XYSCL /XX, YY, XG, YG, BREF, ZZZ, ITEST

LINE 6
DATA EPS/L.E-78/

LINE 7
XP = XG*BREF*XX

LINE 8
YP = YG*BREF*YY

LINE 9
ZFUN = B1B2*XPMXI*YP

LINE 10
DZDX = B2

LINE 11
DO I = 1, N

LINE 12
XPMXI = XP - X(I)

LINE 13
R2 = XPMXI*XPMXI & (YP-Y(I))**2

LINE 14
R2EPS = R2 & EPS

LINE 15
XXLOG = ALOG (R2EPS)

LINE 16
ZFUN = ZFUN & A(I)*R2*XXLOG

LINE 17
DZDX = DZDX & 2.0*A(I)*XPMXI*(XXLOG & R2/R2EPS)

LINE 18
CONTINUE

LINE 19
RETURN

LINE 20
END

END OF JOB.

53.6 SEC. USED .015 HRS. CHARGED 49.937 HRS. REMAINING

-6.111-