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>
<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>
</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.
Figure 1. Computer Program Flow
CALL RD MODE
SELECT FREQUENCY
CALL DIAG
SELECT MODE
GENERATE INPUT DOWN WASH
CALL ITRATE CALL PHIL
READ MODES - SPLINE TO AERO GRID
DECOMPOSE W MATRIX
SOLVE FOR POTENTIAL GRADIENT

CONVERGENCE ?
YES
Wake Effect ?
YES
DISPLAY OUTPUT
GO TO NEXT CASE
END OF JOB

Figure 1. Computer Program Flow (cont'd)
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.

PRECEDING PAGE BLANK NOT FILLED
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 \( \text{EPS} = .01 \). \( \text{EPS} \) 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. \( \text{PERC} \) 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</td>
<td>FGEN, MECH, MESHCL, WINTGR</td>
</tr>
<tr>
<td></td>
<td>BEETA</td>
<td>SQRT (MACH^2-1)</td>
<td></td>
</tr>
<tr>
<td>CEPS</td>
<td>EPS</td>
<td>Convergence criteria for</td>
<td>IRELE, ITRATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ITRATE, IRELE</td>
<td></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></td>
<td>EREØNE</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></td>
<td>EMIRRO</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></td>
<td>RR</td>
<td>Average hyperbolic radius of the endpoints of an influencing line with respect to a given receiving point</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RL</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></td>
<td>RTR</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></td>
<td>RTL</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 $FREQP(I)=FREQ(I)*MACH/BETA$</td>
<td></td>
</tr>
<tr>
<td>FRTEXT</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></td>
<td>KST12</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></td>
<td>X</td>
<td>$X$ coordinate of the point</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>$Y$ coordinate of the point</td>
<td></td>
</tr>
<tr>
<td></td>
<td>J1</td>
<td>Wing number that the point is on</td>
<td></td>
</tr>
<tr>
<td></td>
<td>J2</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, RT0I, WINTER</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, J15</td>
<td>File numbers set up by block data subprogram</td>
<td>MAIN, DIAG, ITRATE, LOOPW, TIMOUT, WRITEO</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>WTES</td>
<td>Array of wake effects for elements</td>
<td>MAIN, DIAG, LOOPW, Waket, WINTER</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>
<tr>
<td>NAME</td>
<td>VARIABLES</td>
<td>DEFINITIONS</td>
<td>SUBROUTINEREFERENCE</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>WVl</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>WWl</td>
<td>XO,YO,ZD</td>
<td>The X, Y and Z coordinates of a receiving point</td>
<td>LØFPW, 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>XYSCL</td>
<td>XX</td>
<td>X coordinate of a point</td>
<td>ZFDZ</td>
</tr>
<tr>
<td>YY</td>
<td></td>
<td>Y coordinate of a point</td>
<td></td>
</tr>
<tr>
<td>XO</td>
<td></td>
<td>Offset value in X direction currently taken to be 0.0.</td>
<td></td>
</tr>
<tr>
<td>YO</td>
<td></td>
<td>Offset value in Y direction, currently taken to be 0.0.</td>
<td></td>
</tr>
<tr>
<td>BREF</td>
<td></td>
<td>Scalar transformation value, currently set to 1.0</td>
<td></td>
</tr>
<tr>
<td>ZZZZ</td>
<td></td>
<td>Value of first mode input</td>
<td></td>
</tr>
<tr>
<td>ITEST</td>
<td></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>
</tbody>
</table>
### Subroutine Write-Ups

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAIN</td>
<td>Main calling program</td>
<td>5.10</td>
</tr>
<tr>
<td>BLOCK</td>
<td>Define COMMON constants</td>
<td>5.12</td>
</tr>
<tr>
<td>CABS</td>
<td>Compute absolute value of a complex number</td>
<td>5.14</td>
</tr>
<tr>
<td>DECOM</td>
<td>Decompose matrix for solution</td>
<td>5.15</td>
</tr>
<tr>
<td>DIAG</td>
<td>Determine matrix blocks</td>
<td>5.17</td>
</tr>
<tr>
<td>ECHO</td>
<td>Provide input deck echo print</td>
<td>5.19</td>
</tr>
<tr>
<td>EFPLOT</td>
<td>Calls LRC CALPLT plot routine to end plot tape</td>
<td>5.21</td>
</tr>
<tr>
<td>EONE</td>
<td>Generate element influence</td>
<td>5.22</td>
</tr>
<tr>
<td>FGEN</td>
<td>Generate frequency coefficients</td>
<td>5.23</td>
</tr>
<tr>
<td>FREQW</td>
<td>Determines coefficients for 1 freq.</td>
<td>5.25</td>
</tr>
<tr>
<td>GETTIM</td>
<td>Transforms CPU time to integer</td>
<td>5.26</td>
</tr>
<tr>
<td>GRIDIN</td>
<td>Reads grid input data</td>
<td>5.28</td>
</tr>
<tr>
<td>INP1</td>
<td>Reads and analyzes RUN card</td>
<td>5.30</td>
</tr>
<tr>
<td>IRELE</td>
<td>Test for convergence</td>
<td>5.34</td>
</tr>
<tr>
<td>ITRATE</td>
<td>Iteration equation solver</td>
<td>5.35</td>
</tr>
<tr>
<td>LOOPW</td>
<td>Evaluate W, Wake coefficients</td>
<td>5.36</td>
</tr>
<tr>
<td>MESH</td>
<td>Grid for section without control lines</td>
<td>5.40</td>
</tr>
<tr>
<td>MESHCL</td>
<td>Grid for section with control lines</td>
<td>5.42</td>
</tr>
<tr>
<td>MFUN</td>
<td>Generate modes by polynomial</td>
<td>5.44</td>
</tr>
<tr>
<td>MXERR</td>
<td>Prints error messages</td>
<td>5.45</td>
</tr>
<tr>
<td>OUTP1</td>
<td>Prints velocity potentials, pressures</td>
<td>5.46</td>
</tr>
<tr>
<td>OUTP2</td>
<td>Prints total lift and total pressures</td>
<td>5.47</td>
</tr>
<tr>
<td>OUTP3</td>
<td>Prints aerodynamic coefficients</td>
<td>5.48</td>
</tr>
<tr>
<td>PHIL</td>
<td>Generate velocity potential</td>
<td>5.49</td>
</tr>
<tr>
<td>PLOT</td>
<td>Calls LRC CALPLT and NFRAME plot routines</td>
<td>5.51</td>
</tr>
<tr>
<td>Name</td>
<td>Function</td>
<td>Page</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>PLOTGD</td>
<td>Plots aerodynamic grid</td>
<td>5.52</td>
</tr>
<tr>
<td>PTGRID</td>
<td>Prints aerodynamic grid</td>
<td>5.54</td>
</tr>
<tr>
<td>RDETA</td>
<td>Read ETA from files 14, 15</td>
<td>5.56</td>
</tr>
<tr>
<td>RDMODE</td>
<td>Read and generate mode data</td>
<td>5.57</td>
</tr>
<tr>
<td>READAB</td>
<td>Read spline coefficients</td>
<td>5.60</td>
</tr>
<tr>
<td>READXY</td>
<td>Read Structural Grid</td>
<td>5.62</td>
</tr>
<tr>
<td>RSTART</td>
<td>Reads and checks restart files</td>
<td>5.64</td>
</tr>
<tr>
<td>RTOI</td>
<td>Generates transformation matrices</td>
<td>5.66</td>
</tr>
<tr>
<td>SOLVE</td>
<td>Backsubstitution for solution</td>
<td>5.58</td>
</tr>
<tr>
<td>SONS</td>
<td>Finds influence of sections</td>
<td>5.70</td>
</tr>
<tr>
<td>SONSPT</td>
<td>Prints influence of sections</td>
<td>5.72</td>
</tr>
<tr>
<td>SYMBOL</td>
<td>Calls LRC plot routine NOTATE</td>
<td>5.73</td>
</tr>
<tr>
<td>TIMOUT</td>
<td>Time print on file J9</td>
<td>5.74</td>
</tr>
<tr>
<td>TMAX</td>
<td>Computes the maximum elements in a real array</td>
<td>5.76</td>
</tr>
<tr>
<td>TRANS</td>
<td>Transformation of coordinates</td>
<td>5.77</td>
</tr>
<tr>
<td>WAKET</td>
<td>Write wake effects on file T12</td>
<td>5.78</td>
</tr>
<tr>
<td>WINGCK</td>
<td>Check course points for compatibility</td>
<td>5.80</td>
</tr>
<tr>
<td>WINGIN</td>
<td>Read wing input data</td>
<td>5.82</td>
</tr>
<tr>
<td>WINTGR</td>
<td>Computes W coefficients for 2 elements</td>
<td>5.87</td>
</tr>
<tr>
<td>WRETA</td>
<td>Write ETA in file II3 and II4</td>
<td>5.89</td>
</tr>
<tr>
<td>WRITEO</td>
<td>Write W coefficients on file II1</td>
<td>5.90</td>
</tr>
<tr>
<td>Writz</td>
<td>Write W coefficients in file II0</td>
<td>5.91</td>
</tr>
<tr>
<td>WRITZW</td>
<td>Write Wake effects on file II2</td>
<td>5.92</td>
</tr>
<tr>
<td>WTEPNT</td>
<td>Generate wake effects</td>
<td>5.93</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  
PGEN  MXERR  EFPLOR  PTGRID  
INPL  OUTP1  GETTIM  RDMUDE  
MESH  OUTP2  GRIDIN  RSTART  
PHIL  OUTP3  ITRATE  & SONSPT  
TIMOUT  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
\[ \text{Il4} = 14 \]
\[ \text{J9} = 9 \]
\[ \text{Il5} = 15 \]

See Section 2 for file descriptions.

4. \text{VELCOM} \quad \text{NMAX} = \text{maximum solution block size} = 10
\quad \text{PRINT} = \text{print control} = 2

5. \text{Calling Sequence: None}
1. **Function Name**: CABSO

2. **Purpose**:  
Compute absolute value of complex number.

3. **Input Arguments**:  
A = complex number

4. **Output Argument**:  
CABSO = value of function

5. **Subroutine User**: IRELE

6. **Subroutine Used**: SQRT

7. **Calling Sequence**:  
CABSO = CABSO(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 (incor) stored on file I10
   AROW     off diagonal block stored on file I11
   IZ       non zero element numbers array
   IP       - decomposing information array
   CROW     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 DIAG (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 15, and placed on unit 111 and printed with format control on unit 16. If an end of file is encountered, the subroutine sets the variable KØNTRL equal to 1.

4. **Input Arguments**: None

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

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

   /ZFDZIØ/

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

7. **Error Returns**: None

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

9. **Input Tapes**: None
10. **Output Tapes:**
   I11 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 EREONE 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/
/WW1/

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:
   \[ \text{KAPPA} = \frac{K \times M}{\sqrt{M \times M - 1}} \]
   where \( M \) is the MACH NUMBER.

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

   The frequency coefficients are calculated from:
   \[ C_1 = 1.0 \]
   \[ C_J = -C_{N-1} \times \frac{(\text{KAPPA}^2)}{(2 \times N \times (2 \times N - 1))} \]
   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**:
   /BASIC/ and /FQ1/
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 I8. 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 I8

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  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 (A4,2X,9I2,4F12.0).
   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 REFLEN. 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.
   **PLTR** Logical variable set to .TRUE. if plotting is requested. Otherwise it is .FALSE.
BEETA  BEETA = $\text{SORT(MACH}\times\text{MACH}-1.0)$.

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

EAAS  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:

/NEXTCO/

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

7. ERROR RETURNS

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.
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(I5,I6,TITLE,ITRUN,MACH,REFLEN,XP,XPIN,PLOTR,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/
   /WML/

7. **Error Returns:** None

8. **Calling Sequence:**
   Call LOOPW(NET,NWING,XYZ,NSP,SYM,TRS,NNCH,NE,NSECT,ZSECT,ISON,IS6,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.
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)

NO

UPDATE ELEMENT COUNTERS AND SKIP TO NEXT INFLUENCING SECTION

Flow Chart of Subroutine LOOPW
A

TEST MAGNITUDE OF INFLUENCE AGAINST DIAGONAL SELF INFLUENCE

INSIGNIFICANT

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

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

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: TRANZ
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 MESH 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,DLSPAN,PMSPAN,
   XP,INDXI,PRES,IPW,I6,LINLMAX,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:**
   - **L6:** 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 (L6,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 exponentials 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 Bocks 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.
Subroutine Name: PLOTGD

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

Equations and Procedure:
Standard CALCOMP plotting techniques are used.

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

Output Arguments: None

Common Blocks Used: None

Error Returns: None

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

Input Tapes: None

Output Tapes: None

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
   - **N$ECT** Number of sections in each wing.
   - **N$P** 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.
   - **Z$ECT** Z coordinates of each section.
   - **REPLEN** Input reference length.
   - **BEETA** $BEETA=\sqrt{MACH^2-1.0}$. 
5. **Output Arguments:**
   Xcen    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
   - **NXECT** 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:
/MODEM/
/MODEF/
/XYSCAL/

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,NSEC,NE,XCEN,XYZ,BETA,REPLN,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 $Z31$. 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 README 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 $331$. ($331$ 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, I15.

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.
   - I15    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**
   
   /NEXTC2/ (see Error Returns)

7. **Error Returns**

   IFIUXH in COMMON/NEXTC2/ 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,16,1,15,1,FREQR,XYZ,AREA).

9. **Input Tapes:**

   Logical unit 16 and 115, the restart tapes. 115 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, SET 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 ISONS(K1,K2) is determined such that ISONS(K1,K2) = 1 if section K2 has influence of section K1, 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 CORNY
CORNZ the wing corner points

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,ISONS)

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:** XONPT

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

3. **Procedure:**
   XON (see Subroutine XON write-up) is printed in tabular form.

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

5. **Output Arguments:** None

6. **Common Blocks Used:** None

7. **Error Returns:** None

8. **Calling Sequence:**
   Call XONPT(IXON,NWING,NSECT,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**
   XP,YP  X and Y coordinates of the corner points of the section in the local reference system of the section
   ZC    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,XP,YP,ZC)

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
   - IL2 = output tape number
   - IF = frequency number
   - LRECLW = length of record on tape 18
   - K1 = number frequencies

4. **Output Arguments:**
   - IZ = non zero elements in record
   - AROW = elements selected for given frequency stored on tape IL2
   - WROW = record from tape 18 containing wake effects coefficients
   - NNCH = row numbers of trailing edge elements

5. **Calling Sequence:**
   Call WAKET (IL2,AROW,WROW,IF,LRECLW,NSPT,IL8,IZ,K1,NNCH,FRQ)

6. **Files:**
   - IL2 = 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
   - I15 = 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**: Input X, Y, and Z coordinates of the wing
   - **Corny**: Input X, Y, and Z coordinates of the wing
   - **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)
   - REFLEN  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 preceded by the statement ERROR IN READING INPUT.)
Messages a through f have to do with errors on the WING data card, and are proceeded 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/
   /RRLL/
   /WAKE/
   /WAKEUP/
   /WVL/
   /WWL/

-5.87-
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:**
   - I11 = tape on which records are written

7. **Subroutines Required:**
   WRITZ

8. **Subroutine User:** DIAG

9. **Common Blocks:**
   /TAPE/I9, I10, I11
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 an 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, DFX, 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:** RMD\O DE

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 Z\S31.

(Program Z\S31 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.
<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>CABSQ</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>READMEY</td>
<td>31</td>
</tr>
<tr>
<td>EFPLDT</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>INP1</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>WRTEO</td>
<td>46</td>
</tr>
<tr>
<td>OUTP2</td>
<td>22</td>
<td>WRITZ</td>
<td>47</td>
</tr>
<tr>
<td>OUTP3</td>
<td>23</td>
<td>WRITEW</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>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>NSP(1)</td>
<td>00100570</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>---------</td>
<td>-----------</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>S(1101)</td>
<td>XLE(1001)</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>SW(844)</td>
<td>1</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>SW(1201)</td>
<td>00100600</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>SYM(4)</td>
<td>00100610</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>TITLE(24)</td>
<td>00100620</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>TRS(3,4)</td>
<td>ZSEC(3,4)</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>META(1300,10)</td>
<td>00100640</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>W(130001)</td>
<td>00100650</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>W(36000)</td>
<td>00100660</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>XCN(300)</td>
<td>00100670</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>XYZ(1800L)</td>
<td>00100680</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>XYZF0Z(300,3)</td>
<td>00100690</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>SYM(4)</td>
<td>00100700</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>TITLE(24)</td>
<td>00100710</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>TRS(3,4)</td>
<td>ZSEC(3,4)</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>META(1300,10)</td>
<td>00100730</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>W(130001)</td>
<td>00100740</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>NSW(4)</td>
<td>00100750</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>NSW(10)</td>
<td>00100760</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>NSW(50)</td>
<td>00100770</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>NCONR(8)</td>
<td>00100780</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>NFRQ(12)</td>
<td>00100790</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>NESP(12)</td>
<td>00100800</td>
</tr>
<tr>
<td>LINE</td>
<td>DIMENSION</td>
<td>NELEM(40)</td>
<td>00100810</td>
</tr>
<tr>
<td>LINE</td>
<td>COMM/</td>
<td>NETICS /</td>
<td>IFLUSH</td>
</tr>
<tr>
<td>LINE</td>
<td>COMM/N</td>
<td>PARAM/</td>
<td>NET</td>
</tr>
<tr>
<td>LINE</td>
<td>COMM/</td>
<td>T/PEPS /</td>
<td>PERC</td>
</tr>
<tr>
<td>LINE</td>
<td>COMM/</td>
<td>TIME/</td>
<td>ITIME</td>
</tr>
<tr>
<td>LINE</td>
<td>COMM/</td>
<td>VELCOM /</td>
<td>NMAX, PRINT, N88LCK, N88RMT(20)</td>
</tr>
<tr>
<td>LINE</td>
<td>COMM/</td>
<td>WAVE /</td>
<td>ITIM, IGNG, EPSW, IGNG, NNCH, INTE</td>
</tr>
<tr>
<td>LINE</td>
<td>COMM/</td>
<td>WAKE/</td>
<td>HET(21), TETE, WAKE, WAKEZ</td>
</tr>
<tr>
<td>LINE</td>
<td>COMM/</td>
<td>ZFUND</td>
<td>J5, J6</td>
</tr>
<tr>
<td>LINE</td>
<td>COMM/</td>
<td>ZFUNNY</td>
<td>J1, J2, J3, J4</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(AROW(1), CAROW(1))</td>
<td>00100910</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(MSEC, ITIME(2))</td>
<td>00100920</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(CAROW(1))</td>
<td>00100930</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(MSEC, ITIME(2))</td>
<td>00100940</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(CAROW(1))</td>
<td>00100950</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(MSEC, ITIME(2))</td>
<td>00100960</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(CAROW(1))</td>
<td>00100970</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(MSEC, ITIME(2))</td>
<td>00100980</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(CAROW(1))</td>
<td>00100990</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(MSEC, ITIME(2))</td>
<td>00101000</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(CAROW(1))</td>
<td>00101010</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(MSEC, ITIME(2))</td>
<td>00101020</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(CAROW(1))</td>
<td>00101030</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(MSEC, ITIME(2))</td>
<td>00101040</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(CAROW(1))</td>
<td>00101050</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(MSEC, ITIME(2))</td>
<td>00101060</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(CAROW(1))</td>
<td>00101070</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(MSEC, ITIME(2))</td>
<td>00101080</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(CAROW(1))</td>
<td>00101090</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(MSEC, ITIME(2))</td>
<td>00101100</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(CAROW(1))</td>
<td>00101110</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(MSEC, ITIME(2))</td>
<td>00101120</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(CAROW(1))</td>
<td>00101130</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(MSEC, ITIME(2))</td>
<td>00101140</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(CAROW(1))</td>
<td>00101150</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(MSEC, ITIME(2))</td>
<td>00101160</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(CAROW(1))</td>
<td>00101170</td>
</tr>
<tr>
<td>LINE</td>
<td>EQUIVALENCE</td>
<td>(MSEC, ITIME(2))</td>
<td>00101180</td>
</tr>
</tbody>
</table>

ORIGINAL PAGE 2
OF POOR QUALITY
LINE 119 EQUIVALENCE ( SWORK(11), WWORK(11) )
LINE 120 EQUIVALENCE ( SWORK(12:11), SWORK(11) )
LINE 121 EQUIVALENCE ( SWORK(11), W(11) )
LINE 122 EQUIVALENCE ( SWORK(11:101), P(11) )
LINE 123 EQUIVALENCE ( SWORK(1171), PHI(1) )
LINE 124 EQUIVALENCE ( SWORK(471), DENX(1) )
LINE 125 EQUIVALENCE ( SWORK(771), DENX(1) )
LINE 126 EQUIVALENCE ( SWORK(1071), CXP(1) )
LINE 127 EQUIVALENCE ( SWORK(11371), W(1) )
LINE 128 EQUIVALENCE ( SWORK(671), ESWRK(1) )
LINE 129 EQUIVALENCE ( SWORK(1) ), DR(1) )
LINE 130 EQUIVALENCE ( SWORK(301), OI(1) )
LINE 131 EQUIVALENCE ( SWORK(601), IZ(1) )
LINE 132 EQUIVALENCE ( SWORK(901), ARGW(1) )
LINE 133 EQUIVALENCE ( SWORK(1201), ESWRK(1) )
LINE 134 EQUIVALENCE ( TPEW(9) )
LINE 135 EQUIVALENCE ( WORK(11), AR(1) )
LINE 136 EQUIVALENCE ( WORK(5), CEX )
LINE 137 EQUIVALENCE ( WORK(53), CFD )
LINE 138 EQUIVALENCE ( WORK(57), CID )
LINE 139 EQUIVALENCE ( WORKM(66), CECL )
LINE 140 EQUIVALENCE ( WORKM(76), CTRX )
LINE 141 EQUIVALENCE ( WORKM(126), CORNX )
LINE 142 EQUIVALENCE ( WORKM(158), CORNX )
LINE 143 EQUIVALENCE ( WORKM(190), CORNX )
LINE 144 EQUIVALENCE ( WORKM(222), CSMNS )
LINE 145 EQUIVALENCE ( WORKM(366), CSECT )
LINE 146 EQUIVALENCE ( WORKM(378), CSMSC )
LINE 147 EQUIVALENCE ( WORKM(401), XYZ(1) )
LINE 148 EQUIVALENCE ( WORKM(2201), WTEROW(1) )
LINE 149 EQUIVALENCE ( WORKM(4601), WETA(1) )
LINE 150 EQUIVALENCE ( WORKM(7601), WETA(1) )
LINE 151 EQUIVALENCE ( WORKM(10601), EWORKM(1) )
LINE 152 EQUIVALENCE ( LCMW(1), XYZFDZ(1) )
LINE 153 EQUIVALENCE ( LCMF(1), WETA(1) )
LINE 154 DATA 13/6/8
LINE 155 DATA 15, 16 / 11, 6 /
LINE 156 DATA LMODE / 4HMODE /
LINE 157 DATA TITLE / "" /
LINE 158 DATA LMAX / 62 /
LINE 159 DATA LWP / 6293185304 /
LINE 160 DATA XP / 0.0 /
LINE 161 C
LINE 162 J5 = 15
LINE 163 J6 = 16
LINE 164 PLOT = .FALSE.
LINE 165 4 CONTINUE (ERR = 0)
LINE 166 CALL ECHI (KONTRL)
LINE 167 IF(KJYTRL .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 SECTION TO READ AND WRITE INPUT DATA
LINE 174 C
LINE 175 C
LINE 176 C ERROR = .FALSE.
LINE 177 RREX = 0.0
LINE 178 NMAX = 1
LINE 179 NCVEL = 0
LINE 180 TEXC) = .FALSE.
LINE 181 C READ( 5, 1006 ) TITLE
LINE 182 C CONTINUE
LINE 183 C READ( 15, 1006, END = 10 ) LABEL, TITLE
LINE 184 C CDC READ( 15, 1006 ) LABEL, TITLE
LINE 185 C CDC IF( E(15) ) 10, 8
LINE 186 C CDC q CONTINUE
LINE 187 C IF( LABEL .NE. TITLE ) GO TO 6
LINE 188 C WRITE( 10, 2006 ) TITLE
LINE 189 C NHNG, NO. WINGS BETWEEN LAND 4
LINE 190 C
LINE 191 C CALL INPI( 15, 16, TITLE, ITRUN, MACH, REFLCN, XP, XPN, PLUTR, LTRUN )
LINE 192 C 1 LATE, TARO, BEEFA, LAB, ID, FD, ERR0R )
LINE 193 C IF( IFLUSH .NE. 0 ) GO TO 500
LINE 194 C
LINE 195 C CALL MING(V) 15, 16, NHNG, NSECT, AR, NSP, CORNX, CORNY, CURNZ, SYM, ICL, CL, ERR0R, BEEFA, IPW, NFR4, FR44
LINE 196 C 2 REFLCN, LAB, ID, FD )
LINE 197 C
LINE 198 C MAIN 158
LINE 199 C IF(NWINGS .LT. NWING ) GO TO 610
LINE 200 C GENERATE AERO GRID
LINE 201 C
LINE 202 C NSPT = 1
LINE 203 C IF(PLTRI) CALL PSEUDO
LINE 204 C IF( PLOT ) CALL PLOTS( IYBUFF, 8000 )
LINE 205 C LOOP ON WINGS J
LINE 206 C NWING = 0
LINE 207 C VS = NSECT(J)
LINE 208 C IF(NSECT(J) .LT. NS ) GO TO 620
LINE 209 C LOOP ON SECTIONS OF EACH WING I
LINE 210 C NSPT = 0
LINE 211 C CALL 4ESH( CORNX(JCN,J), CORNY(JCN,J), CURNZ(JCN,J), XYZ(NINDEX,J)
LINE 212 C ZSEC T(JCN,J), TRS(JCN,J), NNCPI(NSPT), SW(NSPT), XLE(NSPT)
LINE 213 C)
LINE 214 C NSPT = 1
LINE 215 C IF(INSECSTN,.LT. NS ) GO TO 620
LINE 216 C LOOP ON SECTIONS OF EACH WING I
LINE 217 C NSPT = 0
LINE 218 C IF(INSECSTN,.LT. NS ) GO TO 620
LINE 219 C NSPT = 1
LINE 220 C VS = NSECT(J)
LINE 221 C IF(INSECSTN,.LT. NS ) GO TO 620
LINE 222 C LOOP ON SECTIONS OF EACH WING I
LINE 223 C NSPT = 0
LINE 224 C JCN = 2*J - 1
LINE 225 C JCN = 2*J - 1
LINE 226 C JCN = 2*J - 1
LINE 227 C NSPT( 19J), NE(I,J), NETSV, FREQ )
LINE 228 C NWING, NSECT, ICL, BEEFA, LAB, ID, FD, 15, 16
LINE 229 C NWING, NSECT, ICL, BEEFA, LAB, ID, FD, 15, 16
LINE 230 C NWING, NSECT, ICL, BEEFA, LAB, ID, FD, 15, 16
LINE 231 C NWING, NSECT, ICL, BEEFA, LAB, ID, FD, 15, 16
LINE 232 C NWING, NSECT, ICL, BEEFA, LAB, ID, FD, 15, 16
LINE 233 C NWING, NSECT, ICL, BEEFA, LAB, ID, FD, 15, 16
LINE 234 C NWING, NSECT, ICL, BEEFA, LAB, ID, FD, 15, 16
LINE 235 C NWING, NSECT, ICL, BEEFA, LAB, ID, FD, 15, 16
LINE 236 C NWING, NSECT, ICL, BEEFA, LAB, ID, FD, 15, 16
LINE 237 C NWING, NSECT, ICL, BEEFA, LAB, ID, FD, 15, 16
LINE 238 C NWING, NSECT, ICL, BEEFA, LAB, ID, FD, 15, 16
LINE 239 C NWING, NSECT, ICL, BEEFA, LAB, ID, FD, 15, 16
LINE 240 C NWING, NSECT, ICL, BEEFA, LAB, ID, FD, 15, 16
LINE 241 C NWING, NSECT, ICL, BEEFA, LAB, ID, FD, 15, 16
LINE 242 C NWING, NSECT, ICL, BEEFA, LAB, ID, FD, 15, 16

-6.6-
LINE 243 2 VSPI(I,J), NEI(I,J), NETSV, ICLII(J), CL(II,J), EARU)
LINE 244 18 CONTINUE
LINE 245 C
LINE 246 VSPI = NSPV & NSP(I,J)
LINE 247 NSPT = NSPT & NSP(I,J)
LINE 248 NEW = NEW & NEI(J)
LINE 249 C END LOOP ON SECTIONS OF WING CONTINUE
LINE 250 19 CONTINUE
LINE 251 C
LINE 252outine = 6*NET & 1 JCN = NUMBER OF CONER POINTS TITLE = WING
LINE 253 C
LINE 254 JCN = 2*(NSEC(J)-1) & 4
LINE 255 IF(NSEC(J) > 650) GO TO 650
LINE 256 IF( > PLOT ) CALL PLOT(JNEI(II,J), XYZ(INDEX), NNCH(J), NSP(I,J), GO TO 6220
LINE 257 1 NSEC(J), TITLE, J, ZSEC(J,J), AR, MACH)
LINE 258 19 END LOOP ON WINGS
LINE 259 NET = NET & NEW
LINE 260 CONTINUE
LINE 261 C
LINE 262 20 CONTINUE
LINE 263 IF(NFRQ < NFREQ) GO TO 660
LINE 264 C IF( ) LOOP CALL ENDPLT
LINE 265 C
LINE 266 NSPT = NSPT - 1
LINE 267 IF(NSPANS.LT. NSEC) GO TO 640
LINE 268 WRITE ( 6, 1014 ) ( NNCH(I), I=L,NSPT )
LINE 269 WRITE ( 6, 1015 ) ( XLE(II), I=L,NSPT )
LINE 270 WRITE ( 6, 1016 ) ( I=1,NSPT )
LINE 271 C014 FORMAT (/// ( 1018),)
LINE 272 C015 FORMAT (/// ( 5E12.4 ) )
LINE 273 C GENERATE NWROW AND NWBLOK
LINE 274 C
LINE 275 NWBLJK=0
LINE 276 ISUM=NNCH(1)
LINE 277 DO 1220 18 =1,NSPT
LINE 278 ISUM= ISUM
LINE 279 IDIFF = NNCH (18) - NNCH(18-1)
LINE 280 ISUM = ISUM & IDIFF
LINE 281 WRITE ( 6, 1020 ) ( I=1, NSPT )
LINE 282 NWBLJK = NWBLOK &1
LINE 283 NWROW (NWBLJK) = ISUM1
LINE 284 ISUM = IDIFF
LINE 285 1020 CONTINUE
LINE 286 NWBLJK = NWBLOK &1
LINE 287 NWROW(NWBLOK) = ISUM
LINE 288 CALL PTOID(16, TITLE, MACH, NWING, NSEC, NSP, NNCH, XYZ, ZSEC,
LINE 289 1 REFLEN, BEETA, XCEN )
LINE 290 C
LINE 291 C CALL SONS TO DETERMINE ISONS(K1,K2)=1 IF SECTION K2 HAS
LINE 292 C INFLUENCE ON SECTION K1, OTHERWISE ISONS(K1,K2)=0
LINE 293 C CALL SONS( NWING, NSEC, CORX, CORY, CORZ, ISONS )
LINE 294 C
LINE 295 C CALL SONSPT( ISONS, NWING, NSEC, MACH, 16 )
LINE 296 25 CONTINUE
LINE 297 C
LINE 298 C
LINE 299 C
LINE 300 C
LINE 301 C
LINE 302 C
LINE 303 C
LINE 304 C

-6.7-
LINE 305 WRITE(9, 9001)
LINE 306 9001 FORMAT(1H1, 5X, 7HSECTION, 30X, 16HCOPY SECUNJS USED, 5X, SHARED CPU, 11, 14)
LINE 307 C
LINE 308 CALL TIMOUT(MSEC, 4HREAD, 4H I NP, 4HIME, 4HEND 4H NLET, 4HATE, 4HRID)
LINE 309 1 4H )
LINE 310 C
LINE 311 C CALL FGEN
LINE 312 LRECL = NET*NPRE
LINE 313 C
LINE 314 C IF THERE IS ONLY ONE FREQUENCY, AND IT = .0, SET IWT= J
LINE 315 C
LINE 316 C LWRITE = 1
LINE 317 IF( VRFQ .EQ. 1 .AND. FREQ(I) .EQ. 0 ) IWT = 0
LINE 318 IF( ITR .NE. J ) LWRITE = NSPT, 'NFREO'
LINE 319 LWRITE = IWT
LINE 320 IF( LWRITE .EQ. 1) LTM = 1
LINE 321 IFL WRITE .EQ. 1) LTM = 1
LINE 322 IF( ITR .NE. J ) GO TO 60
LINE 323 IF( ITR .NE. J ) GO TO 60
LINE 324 C WRITE(6,3002) LWRITE, LTM
LINE 325 C
LINE 326 CALL LUOPM ( NET, NWING, XYZ, NSP, SYM, TRS, NCH, NE)
LINE 327 1 NSECT, ZSECT, ISONS, 16, TAPF8, XCEN, AREA, WROI, LRELL,
LINE 328 2 WTRWE, PERC, ERROR, LWRITE, REFLEN)
LINE 329 C
LINE 330 CALL GETTM ( ITIME )
LINE 331 CALL TIMOUT(MSEC, 4H I NP, 4HEN, 4HADING, 4HICL, 4HATE, 4HRID)
LINE 332 1 4H TR.)
LINE 333 C
LINE 334 C WRITE MODE INPUT
LINE 335 C
LINE 336 WRITE(16,2006) TITLE
LINE 337 IF ( LAB .NE. LMODE)
LINE 338 60 CONTINUE
LINE 339 IF ( LA8 .LT. JMODUE)
LINE 340 1 JMODUE, XYZ, AREA)
LINE 341 CALL FSTART( MACH, NET, NFREQ, FREQ, REFLEN, IWRITE, TAPEB, 115, 16,
LINE 342 1 FRED, XYZ, AREA)
LINE 343 C
LINE 344, 110 CONTINUE
LINE 345 C
LINE 346 C READ MODE INPUT
LINE 347 C MODE CARD HAS BEEN READ
LINE 348 C
LINE 349 C CALL ROUTINE TO READ MODES IN UNNORMALIZED STRUCTURAL REF. SYST.
LINE 350 C
LINE 351 C
LINE 352 WRITE(16, 2006) TITLE
LINE 353 IF ( LAB .NE. LMODE) 1 GO TO 132
LINE 354 C
LINE 355 CALL IOMODE( JMODE, NWING, NSECT, NE, XCEN, XYZ, BETA, REFLY,
LINE 356 1 NET, LAB, ID, ERROR, NFT, WDETA, 15, 15, XYZFDZ(1,1), XYZFDZ(1,2),
LINE 357 2 XYZFDZ(1,3), XMFDZ, IERR)
LINE 358 C
LINE 359 IF( IEM .NE. 0) GO TO 680
LINE 360 IF( I1MODE .LT. JMODE) GO TO 630
LINE 361 IF( JMODE .EQ. 0) GO TO 500
LINE 362 IF( IFLSH .EQ. 1) GO TO 500
LINE 363 C
LINE 364 C CALL GETTM ( ITIME )
LINE 365 CALL TIMOUT(MSEC, 4HREAD, 4H AND, 4H PRN, 4HCESS, 4H MOD, 4H IN, 4H INPUT,
LINE 366 1 4H )
I

LINE 367 C
LINE 368 C
LINE 369 IF( LTRUN .EQ. 3 ) GO TO 132
LINE 370 IF( LTRUN .EQ. 1 ) GO TO 126
LINE 371 WRITE( 16, 3031 )
LINE 372 IF( RDTR ) WRITE( 16, 3032 )
LINE 373 GO TO 4
LINE 374 132 CONTINUE
LINE 375 IF( NOT. ERROR ) GO TO 135
LINE 376 WRITE( J9,3033 )
LINE 377 GO TO 4
LINE 378 135 CONTINUE
LINE 379 CALL AETA (NET, JMODE, ETA, ETA, 113).
LINE 380 CALL AETA (NET, JMODE, ETA, ETA, 114).
LINE 381 C
LINE 382 C
LINE 383 C
LINE 384 C
LINE 385 C
LINE 386 C
LINE 387 C
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 CALL DIAG, WROW, IF, LREC, NNCH, MC, J, NET, ARROW, IZ, IP, NSPT, FRQ,
LINE 401 CALL GETTIM, ITIME
LINE 402 CALL TIMOUT, MSEC, 4HR, 4FAD, 4H AND, 4H DEC, 4HOMPO, 4HSEC, 4H. MAX
LINE 403 WRITE(J9,3025)
LINE 404 CALL TIMOUT(MSEC, 4HREAD, 4H AND, 4H DEC, 4HOMPO, 4HSEC, 4H. MAX)
LINE 405 1 4HTRIX
LINE 406 C
LINE 407 C
LINE 408 DD 189 I=1,NET
LINE 409 FRZ = XCM( I ) XKM
LINE 410 CXP( I ) = CMPLX( COS(FBL), SIN(FBL) )
LINE 411 168 CONTINUE
LINE 412 C
LINE 413 C
LINE 414 DD 163 I=1,JMODE
LINE 415 DO 165 J=1,JMODE
LINE 416 165 JT(J,J) = (0.0,0.0)
LINE 417 REWIND 113
LINE 418 C
LINE 419 C
LINE 420 C
LINE 421 DD 28) JM = 1, JMODE
LINE 422 C
LINE 423 C
LINE 424 IF( JMODE .LT. 16 .AND. NET .EQ. LMAX(J1) ) LINE=LMAX
LINE 425 IF(JMODE .EQ. J) LINE=LMAX
LINE 426 IF(FJ .EQ. 0.0) JCNVG = NET
LINE 427 CALL AETA ( 113, JMODE, ETA, NET)
LINE 428 DO 197 I=1,NET

--6.9--
IMPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

LINE 429 C CALCULATE DEDT FOR ONE FREQUENCY AND ONE MODE
LINE 430 C DEDT(I) = CMPLXR(ETA(I) + ETA(I)*F0 + ETA(I)*F0)^2 + LXP(I)
LINE 431 C DEDT(I) = CMPLXR(ETA(I) + ETA(I)*F0 + ETA(I)*F0)^2 + LXP(I)
LINE 432 PHI(I) = 0.0
LINE 433 167 CONTINUE
LINE 434 169 CONTINUE
LINE 435 ICNVG = 0
LINE 436 JCNV = 0
LINE 437 C WRITE(16, 2010) NDED
LINE 438 C WRITE(16, 2008) IF, FREQ(I)
LINE 439 C
LINE 440 C
LINE 441 C 170 I = 1, NET
LINE 442 C WRITE(16,2009) DEDT(I)
LINE 443 DI(I) = DEDT(I)
LINE 444 C(I) = AIMAGI(DEDT(I))
LINE 445 170 CONTINUE
LINE 446 C WRITE(16, 2010) NPDX
LINE 447 C WRITE(16, 2009) UPDX(I)
LINE 448 DO 172 I = NET
LINE 449 CABS1 = SQRT(OR(1)**2 + DIM(1)**2)
LINE 450 IF(CABS1.EQ.0.0) GO TO 178
LINE 451 C WRITE(16,173) I = NET
LINE 452 DO 174 I = 1, NET
LINE 453 CDPX(I) = 0.0
LINE 454 GO TO 181
LINE 455 178 CONTINUE
LINE 456 WRITE(16,3027)
LINE 457 3027 FORMAT(1X,38HISOLUTION FOR DOWNSHIFT USING ITERATION 1)
LINE 458 IF(I.EQ.0) IGWESS = 0
LINE 459 IFI(HY) IGWESS = 0
LINE 460 IFI(L) IGWESS = 0
LINE 461 IF(I.EQ.0) IGWESS = 0
LINE 462 CALL ITRATE(I,11,DR,DTR,DPDX,GT,MC,D,IP,DNHR,DNWI,RI,RI)
LINE 463 1 NET = ARX = DPDX,WW
LINE 464 181 CONTINUE
LINE 465 CALL GETTIM(I,TIME)
LINE 466 CALL TMOUT(MSEC,4HSOLV,4HR PC,4HTENT,4HGRAD,4HTENT)
LINE 467 1 4H
LINE 468 DO 180 I = 1, NET
LINE 469 DPDX(I) = DPDX(I)/CXP(I)
LINE 470 180 CONTINUE
LINE 471 C
LINE 472 C LOOP ON WINGS AND SPANS
LINE 473 C
LINE 474 C NSPI IS SPAN COUNTER
LINE 475 C VSP = 0
LINE 476 INDX = 1
LINE 477 DO 240 J = 1, NWING
LINE 478 C
LINE 479 TL(J, JM) = ( 0.0, 0.0)
LINE 480 TM(J, JM) = ( 0.0, 0.0)
LINE 481 INDX1 = INDX
LINE 482 NSPW = NSPW + 1
LINE 483 C WRITE(16,3007)
LINE 484 C
LINE 485 C LOOP ON SPANS IN WING
LINE 486 DO 220 ISP = 1, NSP
LINE 487 C
LINE 488 C NSPI = NSPI + 1
LINE 489 C VEPS = VEPS + (NSPI - INDX + 1)
LINE 490 IF(VEPS .LT. NEPS) GO TO 670

-6.10-
LINE 401 CALL PHI( XLE吁 SPI), XCEH(INDX), UPUX INDX), INFPS, PHI, CXKM. 00104910
LINE 402 I CP((1, SPI, PHI(INDX) 00104920
LINE 403 IF(1E; L: 0.0) GO TO 182 00104930
LINE 404 IF(1E: 0.0) GO TO 182 00104940
LINE 405 IF(NW. WAKE) GO TO 182 00104950
LINE 406 IF(NW. WAKENL) GO TO 182 00104960
LINE 407 CONTINUE 00104970
LINE 408 WRITE( 16, 3002 ) J, ISP 00104980
LINE 409 DLS = (0.0, 0.0) 00104990
LINE 500 PMS = (0.0, 0.0) 00105000
LINE 501 C LOOP ON NUMBER OF ELEMENTS IN SPAN ISP 00105010
LINE 502 C DO 210 IE = 1, NEPS 00105020
LINE 503 PKE. INDXR = (DPDX(INDX) & CPREW PHl(IE) )*TWOPI 00105030
LINE 504 DLCA = PRES(INDXR)*AREA(INDXR) 00105040
LINE 505 C WRITE(16, 3003) INDXR, PHI(IE), PRES(INDXR), DLS 00105050
LINE 506 CONTINUE 00105060
LINE 507 TL(J,JM) = TL(J, J) - PMS 00105070
LINE 508 TITLE = PHI(W, NSP1) 00105080
LINE 509 C END LOOP ON ELEMENT IE OF SPAN ISP 00105090
LINE 510 C END LOOP ON NUMBER OF ELEMENTS IN SPAN ISP 00105100
LINE 511 C CONTINUE 00105110
LINE 512 C C END LOOP ON SPAN ISP OF WING J 00105120
LINE 513 DO 220 IM = IpJMOO 00105130
LINE 514 C READ ETA FROM 114 00105140
LINE 515 CALL ROE(FA, [J4, DETAO X, ETA, NET) 00105150
LINE 516 C LOOP OVER ALL ELEMENTS OF WINGS 00105160
LINE 517 C END LOOP ON WINGS J 00105170
LINE 518 C END LOOP ON DISPLACEMENT MODE TO CALCULATE Q 00105180
LINE 519 C CONTINUE 00105190
LINE 520 C C LOOP ON DISPLACEMENT MODE TO CALCULATE Q 00105200
LINE 521 C CONTINUE 00105210
LINE 522 C C END LOOP ON WINGS J 00105220
LINE 523 C END LOOP ON WINGS J 00105230
LINE 524 IF(1E:EQ. 0.0) GO TO 240 00105240
LINE 525 IF (1E:TE.EQ. 0) GO TO 230 00105250
LINE 526 IF(NW. WAKE) GO TO 230 00105260
LINE 527 WRITE(16, 3040) ITH 00105270
LINE 528 CONTINUE 00105280
LINE 529 CALL JUPPI MACH, FRO, JM, J, NSECT(J), NSP1(J), SW, DLS. 00105290
LINE 530 PMPSPN(INSP1) = PMPSPN(INSP1) 00105300
LINE 531 CALL WTEPHT( DEUT, NET, PHIW, [1, [, [1, WTE) 00105310
LINE 532 C END LOOP ON DISPLACEMENT MODE TO CALCULATE Q 00105320
LINE 533 C CONTINUE 00105330
LINE 534 C CONTINUE 00105340
LINE 535 C CONTINUE 00105350
LINE 536 C CONTINUE 00105360
LINE 537 C CONTINUE 00105370
LINE 538 C CONTINUE 00105380
LINE 539 C CONTINUE 00105390
LINE 540 C CONTINUE 00105400
LINE 541 CALL WTEPHT( DEUT, NET, PHIW, 112, II2, MTE) 00105410
LINE 542 GO TO 169 00105420
LINE 543 C CONTINUE 00105430
LINE 544 C CONTINUE 00105440
LINE 545 C CONTINUE 00105450
LINE 546 C CONTINUE 00105460
LINE 547 C CONTINUE 00105470
LINE 548 C CONTINUE 00105480
LINE 549 C CONTINUE 00105490
LINE 550 C CONTINUE 00105500
LINE 551 C CONTINUE 00105510
LINE 552 C CONTINUE 00105520

-6.11-
LINE 553  INDEX = 0
LINE 554  DO 200 J=1,NWING.
LINE 555  SYMCJ = 2.0
LINE 556  SYME = 2.0
LINE 557  IF ( SYMCJ .EQ. 0.0 ) SYMCJ = 1.
LINE 558  NS = NSECT(J)
LINE 559  NEW = 0
LINE 560  DO 255 I=1,NS
LINE 561  LW = NEW & NE(J,NS)
LINE 562  DO 230 IE = 1,NEW
LINE 563  INDEX = INDEX & 1
LINE 564  DLGAJ = PKES(INDXR)*AREA(INDXR) & INV
LINE 565  QJM(J,M) = QJM(J,M) & ETA(INDXR)*DO
LINE 566  250 CONTINUE
LINE 567  260 CONTINUE
LINE 568  270 CONTINUE
LINE 569  C END LOOP ON MODE JM
LINE 570  280 CONTINUE
LINE 571  290 CONTINUE
LINE 572  C WRITE(16, 2010) NOUT
LINE 573  C QJM = QJM(J,J)
LINE 574  C DO 320 I=1,JMODE
LINE 575  320 WRITE(16, 3006) I, ( QJM(I,J), J=1,JMODE )
LINE 576  C END LOOP ON FREQUENCY IF
LINE 577  C LINE = LMAX
LINE 578  CALL OUTPY(I3, MACH, FRO, JMODE, Q, LINE, LMAX, TITLE )
LINE 579  CALL GETTIM ( TT )
LINE 580  CALL TIMOUT(MSEC, 4HSOLV, 4HE, P0, 4HR DO, 4H4NWA, 4HSH F, 4HOK D, 4HNE F)
LINE 581  300 CONTINUE
LINE 582  310 CONTINUE
LINE 583  320 WRITE(16, 3026)
LINE 584  IF ( IRUN .EQ. 3 ) GO TO 10
LINE 585  WRITE(16, 3030)
LINE 586  C 1 4REQ.
LINE 587  C 500 CONTINUE
LINE 588  C WRITE(16, 3030)
LINE 589  C 600 CONTINUE
LINE 590  C GO TO 4
LINE 591  C IERR = 610
LINE 592  CALL 4XERR(IERR, NWINGS, NWING, 16)
LINE 593  GO TO 500
LINE 594  C IERR = 620
LINE 595  CALL 4XERR(IERR, NSECTN, NS, 16)
LINE 596  GO TO 500
LINE 597  C IERR = 630
LINE 598  CALL 4XERR(IERR, NMODES, JMODE, 16)
LINE 599  GO TO 500
LINE 600  C IERR = 640
LINE 601  CALL 4XERR(IERR, NSPANS, NSPT, 16)
LINE 602  GO TO 500
LINE 603  C IERR = 650
LINE 604  CALL 4XERR(IERR, NCORN, JCN, 16)
LINE 605  GO TO 500
LINE 606  C IERR = 660
LINE 607  CALL 4XERR(IERR, NFRQUN, NFRED, 16)
LINE 608  GO TO 500
LINE 609  C IERR = 670
LINE 610  CALL 4XERR(IERR, NPSPN, NCPS, 16)
LINE 611  GO TO 500
LINE 612  C IERR = 680
LINE 613  CALL 4XERR(IERR, NFMDZ, 16)
LINE 614  -6.12-
LINE 615  GO TO 500  00106150
LINE 616  COC  REPLACES  00106160
LINE 617  C  10 CONTINUE  00106170
LINE 618  10 IF(PLUTRI) CALL PLOTT(11,UFF, WND0)  00106180
LINE 619  IF(PLUTRI) CALL EFPL0T(WEND)  00106190
LINE 620  WRITE(16,3320)  00106200
LINE 621  3028 FORMAT(1H0//////3X,10(1H*),5X,/* OF JOB,,5X,101H*))  00106210
LINE 622  STOP  00106220
LINE 623  C  00106230
LINE 624  1036 FORMAT(4A2,2X,14A4,10A1)  00106240
LINE 625  2006 FORMAT(1H1/,10X,14A4,10A1/)  00106250
LINE 626  3325 FORMAT(1H0)  00106260
LINE 627  3026 FORMAT(1H11)  00106270
LINE 628  3030 FORMAT(///,5X,10(1H*),5X,20HEND OF EXECUTION RUN,5X,101H*))  00106280
LINE 629  1 // 5X,101H*),5X,35H PROGRAM WILL NOW CYCLE TO NEXT CASE,5X,00106290
LINE 630  2 101H*))  00106300
LINE 631  3040 FORMAT(///,5X,331WAKE EFFECTS VELOCITY POTENTIALS / 8X)  00106310
LINE 632  1 17H ITERATION NUMBER:  112 )  00106320
LINE 633  C3004 FORMAT(/9H LIFT/DS =,IP2E12.4• 5X,11H MOMENT/DS =,IP2E12.4 /)  00106330
LINE 634  3002 FORMAT(/215)  00106340
LINE 635  C3003 FORMAT(1X,13,3(6X,IPZE12.4) 1)  00106350
LINE 636  3005 FORMAT(/,5X,4HTL =,IP2E12.5,5X,4HTM =,IP2E12.5 /)  00106360
LINE 637  C3007 FORMAT(/10H WING SPAN/11X,3HELE,13X,3HPRG,127X,4HPRLE,5X,4HLOAD0JU 00106370
LINE 638  3031 FORMAT(////5X,101H*),5X,101H*)  00106380
LINE 639  1 5X,101H*)  00106390
LINE 640  20H EXECUTE A COMPLETE RUN PJT A 'I' IN COLUMN 10 OF THE 'RUN' DATA CARD, 5X, 101H*)  00106400
LINE 641  3032 FORMAT(/5X,101H*),5X,70HIF ANOTHER PLOT IS NOT DESIRED, PUT 00106410
LINE 642  1 A 'J' IN COLUMN 10 OF THE 'RUN' DATA CARD, 5X, 101H*)  00106420
LINE 643  3033 FORMAT(///,5X,120(14*)///,5X,101H*),5X,40HJN TERMINATE DUE TO ERRORS GIVEN ABOVE, 5X, 5511H*)  00106430
LINE 644  13 ERRORS GIVEN ABOVE, 5X, 5511H*)  00106440
LINE 645  C1000 FORMAT(/A4,2X,912,4E12.0/)  00106450
LINE 646  C1001 FORMAT(24X,4E12.0/)  00106460
LINE 647  C1003 FORMAT(1X,'COL=/',13,5X,IPZE12.4/)  00106470
LINE 648  C1008 FORMAT(3E10.0/)  00106480
LINE 649  C1009 FORMAT(15//120F22.4/)  00106490
LINE 650  C1011 FORMAT(15X,A4/)  00106500
LINE 651  C1021 FORMAT(/,1X,120(14*)/5X,91MORE THAN,13,37 TERMS ARE NEEDED/)  00106510
LINE 652  C 1ED FOR CONVERGENCE FOR, 15, 19 PAIRS OF ELEMENTS, /5X, 00106520
LINE 653  C 2 THE MAXIMUM RADIUS*KAPPA =,E15.4, 14H, WITH KAPPA =,E15.6/)  00106530
LINE 654  C1022 FORMAT /5X,55 THE GREATEST NUMBER OF TERMS NEEDED FOR CONVERGENCE/)  00106540
LINE 655  C 1ENCE 15, 13 //5X,26H THE MAXIMUM RADIUS*KAPPA =,E15.4 //)  00106550
LINE 656  C2007 FORMAT(20X,11HMACH NUMBER, F22.4/)  00106560
LINE 657  C2038 FORMAT(/,23X,21HNUMBER OF FREQUENCIES/)  00106570
LINE 658  15, 12 // 20X,19H LIST OF FREQUENCIES, 5F14.4 / 19X,5F14.4/)  00106580
LINE 659  C2039 FORMAT(2E16.6/)  00106590
LINE 660  C2010 FORMAT(/,5X,24A/)  00106600
LINE 661  C3001 FORMAT(/15X,18H RECEIVING ELEMENT, 15, 5X, 0HNUMBER, 14, 5X/)  00106610
LINE 662  C 1 H0F SECTION, 12, 5X, 70H WING//12 / (1X, 10113.4/)  00106620
LINE 663  C3008 FORMAT(1X,40HNUM=,13, 406X,1P2E12.4//19X,406X,1P2E12.4/)  00106630
LINE 664  C3009 FORMAT(/1X, 3HFREQ, NO, 13, 3X, 5HFREQ =,F8.4, 5X, 3HMODE, 13/)  00106640
LINE 665  C3034 FORMAT(/1X,15H MODAL DATA FOR WING, 12, 8H SECTION, 12/)  00106650
LINE 666  END  00106660

-6.13-
<table>
<thead>
<tr>
<th>LINE</th>
<th>BLOCK DATA</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>LINE 2</td>
<td>INTEGER PRINT</td>
<td></td>
</tr>
<tr>
<td>LINE 3</td>
<td>COMM(N)/CEPS/ EPS</td>
<td></td>
</tr>
<tr>
<td>LINE 4</td>
<td>COMMJ / ITERAT / ITMAX</td>
<td></td>
</tr>
<tr>
<td>LINE 5</td>
<td>CJMMJ / MAXE / NMAX</td>
<td></td>
</tr>
<tr>
<td>LINE 6</td>
<td>COMMJ / MXSTSR / NINGS, NSECTION, NMODES, ASPANS, NCORN, NFKJN, NEPSN</td>
<td></td>
</tr>
<tr>
<td>LINE 7</td>
<td>1, MXFDZ</td>
<td></td>
</tr>
<tr>
<td>LINE 8</td>
<td>COMMJ / PEREPS / PERC</td>
<td></td>
</tr>
<tr>
<td>LINE 9</td>
<td>COMMJ / TAPE/ 19, 110, 111, 112, 113, 114, 19, 115</td>
<td></td>
</tr>
<tr>
<td>LINE 10</td>
<td>COMMJ / VELCOM / NMAX, PRINT, NWBLOCK, NWROW(20)</td>
<td></td>
</tr>
<tr>
<td>LINE 11</td>
<td>C NINGS = 4</td>
<td></td>
</tr>
<tr>
<td>LINE 12</td>
<td>C NSECTION = 3</td>
<td></td>
</tr>
<tr>
<td>LINE 13</td>
<td>C NMODES = 10</td>
<td></td>
</tr>
<tr>
<td>LINE 14</td>
<td>C NSPANS = 100</td>
<td></td>
</tr>
<tr>
<td>LINE 15</td>
<td>C NCORN = 8</td>
<td></td>
</tr>
<tr>
<td>LINE 16</td>
<td>C NFKJN = 12</td>
<td></td>
</tr>
<tr>
<td>LINE 17</td>
<td>C NEPSN = 70</td>
<td></td>
</tr>
<tr>
<td>LINE 18</td>
<td>DATA EPS / .01/</td>
<td></td>
</tr>
<tr>
<td>LINE 19</td>
<td>DATA ITMAX / 10/</td>
<td></td>
</tr>
<tr>
<td>LINE 20</td>
<td>DATA 19, 110, 111 / 8, 10, 11/</td>
<td></td>
</tr>
<tr>
<td>LINE 21</td>
<td>DATA 112/12/</td>
<td></td>
</tr>
<tr>
<td>LINE 22</td>
<td>DATA 113/13/</td>
<td></td>
</tr>
<tr>
<td>LINE 23</td>
<td>DATA 114/14/</td>
<td></td>
</tr>
<tr>
<td>LINE 24</td>
<td>DATA 115/15/</td>
<td></td>
</tr>
<tr>
<td>LINE 25</td>
<td>DATA J6/J9/</td>
<td></td>
</tr>
<tr>
<td>LINE 26</td>
<td>DATA NMAX/60/</td>
<td></td>
</tr>
<tr>
<td>LINE 27</td>
<td>DATA NMAX, NSPANS, NMODES, ASPANS, NCORN, NFKJN, NEPSN</td>
<td></td>
</tr>
<tr>
<td>LINE 28</td>
<td>1, MXFDZ</td>
<td></td>
</tr>
<tr>
<td>LINE 29</td>
<td>2 / 4, 3, 10, 100, 8, 12, 70, 300 /</td>
<td></td>
</tr>
<tr>
<td>LINE 30</td>
<td>DATA PERC / 3.005 /</td>
<td></td>
</tr>
<tr>
<td>LINE 31</td>
<td>DATA PRINT / 2/</td>
<td></td>
</tr>
<tr>
<td>LINE 32</td>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>

00200010
00200020
00200030
00200040
00200050
00200060
00200070
00200080
00200090
00200100
00200110
00200120
00200130
00200140
00200150
00200160
00200170
00200180
00200190
00200200
00200210
00200220
00200230
00200240
00200250
00200260
00200270
00200280
00200290
00200300
00200310
00200320

-6.14-
LINE 1    FUNCTION CABS(A)
LINE 2    COMPLEX A
LINE 3    AR = A
LINE 4    AI = AIMAG(A)
LINE 5    CABS3 = AR*2 + AI*2
LINE 6    CABS3 = SQRT(CABS3)
LINE 7    RETURN
LINE 8    END
SUBROUTINE DECOM (N,NDIM,A,IP)

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 SIMPLY SUBSCRIPTED 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 INTERPRET. HOWEVER, IT IS USED PROPERLY BY 'SOLVE'.

матриця триангуляцією за Гауссівським елімінацією. Виділене

ALGORITHM 423, 'COLLECTED ALGORITHMS FROM CACM', BY CLEVE MOLER

METHOD

MATRICE TRIANGULARIZATION BY GAUSSIAN ELIMINATION. SEE

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

DO 41 N = 1,NDIM

DO 41 K = 1,N

IF (ABS(A(I,K)) .GT. ABS(A(M,K))) M = I

41 CONTINUE

K(M) = M

A(M,K) = A(K,K)

T = A(M,K)

DO 51 I = M+1,N

A(M,I) = A(I,K)*T

51 CONTINUE

DO 52 I = 1,N

52 CONTINUE

DO 54 K = 1,N

54 CONTINUE

DO 55 K = 1,N

55 CONTINUE

DO 56 J = 1,N

56 CONTINUE

-6.16-
<table>
<thead>
<tr>
<th>LINE</th>
<th>Code</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>57</td>
<td>T = A(M,J)</td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>A(M,J) = A(K,J)</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>A(K,J) = T</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>IF (T .EQ. 0.) GO TO 4</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>DO 3 I = KP1,N</td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>IF (A(I,J) .EQ. 0.0) GO TO 21</td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>C = (A(I,J) + A(I,K) *T) / A(I,J)</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>C = A3(SIC)</td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>IF( C .LE. .0001) A(I,J)=0.0</td>
<td></td>
</tr>
<tr>
<td>66</td>
<td>IF( C .LE. .0001) GO TO 3</td>
<td></td>
</tr>
<tr>
<td>67</td>
<td>21 CONTINUE</td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>A(I,J) = A(I,J) + A(I,K) *T</td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>3 CONTINUE</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>4 CONTINUE</td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>5 IF (A(K,K) .EQ. 0.) IP(N) = 0</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>6 CONTINUE</td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>RETURN</td>
<td></td>
</tr>
<tr>
<td>74</td>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>
SUBROUTINE DIAG(ISROW,IF,LRECL,NNCH,MC, D, NET,AROW,IZ, IP, 00500010
LINE  2     NSPT,FREQ,AROW,CWROW  00500020
LINE  3     COMPLEX CAROW(1), CWROW(1)  00500030
LINE  4     COMPLEX IMES  00500040
LINE  5     INTEGER PRINT  00500050
LINE  6     LOGICAL TREDGE  00500060
LINE  7     LOGICAL WAKE,WAKE1,WKENZ  00500070
LINE  8     DIMENSION AROW(MC), IZ, IMC  00500080
LINE  9     DIMENSION MC,MC, IP,MC  00500090
LINE 10     DIMENSION NNCH(1)  00500100
LINE 11     DIMENSION WROW(LRECL)  00500110
LINE 12     COMMON /K1112/ KST11,KST12  00500120
LINE 13     COMMON /TAPE/ 19,110,111,112,113,114,19,19,195 00500130
LINE 14     COMMON /TAMCOM/ NMAX,IP,MC,PPW,IP,MC,PPP,PPW(20)  00500140
LINE 15     COMMON /WAKE/ ITM, ITM, ICNMGW, EPSW, ICNMGW, LRMTE, IMTE  00500150
LINE 16     COMMON /WAKEUP/ WTES(12), TREDGE,WAKE1,WAKE, WKENZ, WAKE  00500160
LINE 17     REWIND 111  00500170
LINE 18     REWIND 119  00500180
LINE 19     REWIND 110  00500190
LINE 20     IB = 19  00500200
LINE 21     IF (IWTE .EQ. 0) REWIND 112  00500210
LINE 22     IF (WAKE) REWIND 115  00500220
LINE 23     READ(19)  00500230
LINE 24     IF (WAKE) READ(115)  00500240
LINE 25     IBT = 0  00500250
LINE 26     K1 = 0  00500260
LINE 27     K = 0  00500270
LINE 28     WKENZ = .FALSE.  00500280
LINE 29     DO 40 18 = 1, NWBLOK  00500290
LINE 30     NROW = NROW(1)  00500300
LINE 31     DO 20 1 = 1, NROW  00500310
LINE 32     K = G1  00500320
LINE 33     CALL FREQW(IS,AROW,MROW,IF,LRECL)  00500330
LINE 34     DO 15 J = 1, NROW  00500340
LINE 35     M = J1BT  00500350
LINE 36     D(I,J) = AROW(M)  00500360
LINE 37     AROW(M) = 0.0  00500370
LINE 38    15 CONTINUE  00500380
LINE 39    IF (NWNBLOK .NE. 1)  00500390
LINE 40    CALL WRITED(K, IZ, WR, MC)  00500400
LINE 41    IF (IWTE .EQ. 0) GO TO 20  00500410
LINE 42    IF (K .EQ. 0) GO TO 20  00500420
LINE 43    IF (K .LT. WAKE) GO TO 20  00500430
LINE 44    IF (K .EQ. WAKE) GO TO 20  00500440
LINE 45    IF (IWE, .EQ. 0) GO TO 20  00500450
LINE 46    CALL MATEI I12, CAROW+CROW, IF, LRMTE, NSPT,  00500460
LINE 47    20 CONTINUE  00500470
LINE 48    IBT = IBT & NROW  00500480
LINE 49    CALL MATPR( D, NROW, NROW, "D MATRIX")  00500490
LINE 50    CALL DECOM(NROW,NMAX,0,IP)  00500500
LINE 51    WRITE (110) 0, IP  00500510
LINE 52    110 J0, IP  00500520
LINE 53    40 CONTINUE  00500530
LINE 54    ENO, FILE 110  00500540
LINE 55    END, FILE 111  00500550
LINE 56    REWIND 110  00500560
-6.18-
LINE 57   REWIND 111
LINE 58   IF(IWTE .NE. 0)  END FILE 112
LINE 59   IF(IWTE .NE. 0)  REWIND 112
LINE 60   RETURN
LINE 61   END
ORIGIN\]NAL Ma z1,,

C IS PLAY ...F ILE DECK 06
LINE 1
LINE 2
LINE 3
LINE 4
LINE 5
LINE 6
LINE 7
LINE 8
LINE 9
LINE 10
LINE 11
LINE 12
LINE 13
LINE 14
LINE 15
LINE 16
LINE 17
LINE 18
LINE 19
LINE 20
LINE 21
LINE 22
LINE 23
LINE 24
LINE 25
LINE 26
LINE 27
LINE 28
LINE 29
LINE 30
LINE 31
LINE 32
LINE 33
LINE 34
LINE 35
LINE 36
LINE 37
LINE 38
LINE 39
LINE 40

ORIGIN\]NAL Ma z1,,

C IS PLAY ...F ILE DECK 06
LINE 1
LINE 2
LINE 3
LINE 4
LINE 5
LINE 6
LINE 7
LINE 8
LINE 9
LINE 10
LINE 11
LINE 12
LINE 13
LINE 14
LINE 15
LINE 16
LINE 17
LINE 18
LINE 19
LINE 20
LINE 21
LINE 22
LINE 23
LINE 24
LINE 25
LINE 26
LINE 27
LINE 28
LINE 29
LINE 30
LINE 31
LINE 32
LINE 33
LINE 34
LINE 35
LINE 36
LINE 37
LINE 38
LINE 39
LINE 40

ORIGIN\]NAL Ma z1,,

C IS PLAY ...F ILE DECK 06
LINE 1
LINE 2
LINE 3
LINE 4
LINE 5
LINE 6
LINE 7
LINE 8
LINE 9
LINE 10
LINE 11
LINE 12
LINE 13
LINE 14
LINE 15
LINE 16
LINE 17
LINE 18
LINE 19
LINE 20
LINE 21
LINE 22
LINE 23
LINE 24
LINE 25
LINE 26
LINE 27
LINE 28
LINE 29
LINE 30
LINE 31
LINE 32
LINE 33
LINE 34
LINE 35
LINE 36
LINE 37
LINE 38
LINE 39
LINE 40

ORIGIN\]NAL Ma z1,,

C IS PLAY ...F ILE DECK 06
LINE 1
LINE 2
LINE 3
LINE 4
LINE 5
LINE 6
LINE 7
LINE 8
LINE 9
LINE 10
LINE 11
LINE 12
LINE 13
LINE 14
LINE 15
LINE 16
LINE 17
LINE 18
LINE 19
LINE 20
LINE 21
LINE 22
LINE 23
LINE 24
LINE 25
LINE 26
LINE 27
LINE 28
LINE 29
LINE 30
LINE 31
LINE 32
LINE 33
LINE 34
LINE 35
LINE 36
LINE 37
LINE 38
LINE 39
LINE 40

ORIGIN\]NAL Ma z1,,

C IS PLAY ...F ILE DECK 06
LINE 1
LINE 2
LINE 3
LINE 4
LINE 5
LINE 6
LINE 7
LINE 8
LINE 9
LINE 10
LINE 11
LINE 12
LINE 13
LINE 14
LINE 15
LINE 16
LINE 17
LINE 18
LINE 19
LINE 20
LINE 21
LINE 22
LINE 23
LINE 24
LINE 25
LINE 26
LINE 27
LINE 28
LINE 29
LINE 30
LINE 31
LINE 32
LINE 33
LINE 34
LINE 35
LINE 36
LINE 37
LINE 38
LINE 39
LINE 40

-6.20-
DISP...FILE       DECK 07

LINE 1           SUBROUTINE EFPIOT (A)  00700010
LINE 2           CALL CALPLT(C.0,0.0,999)  00700020
LINE 3           RETURN  00700030
LINE 4           END     00700040
SUBROUTINE EONE

LOGICAL EREONE, EMIRRO

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

COMMON / RRLL, VHW(4), YRO, YLO, ZR, ZZR, ZL, ZLLZ

COMMON / WAI, XI, YO, ZD, ZDZD, SYMK

TEST IF ELEMENT AND MIRROR IMAGE OF ELEMENT ARE IN MACH CONE

EREONE TRUE IF ELEMENT IS IN MACH CONE. IF EREONE=FALSE THEN EMIRRO=FALSE.

COORDINATES OF RECEIVING PT. (CENTER OF REC. ELE.)

1ST PT. ON LINE OF INFLUENCING ELE.

XUP, YUP 2ND PT. ON LINE OF INFLUENCING ELE.

SYMM SYMMETRY CODE (0.0 FOR NO SYMMETRY)

RR AVERAGE R OF RIGHT-HAND ELEMENT

RL AVERAGE R OF LEFT-HAND ELEMENT

NOTE THIS ROUTINE IS ALSO USED TO TEST TRAILING EDGE LINE OF ELEMENT WHEN X1, Y1 AND X2, Y2 ARE ON TRAILING EDGE AND SYM=0.

ERONE = .FALSE.

EMIRRO = .FALSE.

RK = 0.0

RL = 0.0

ZETA1 = XI - XLW

ZETA2 = XI - XUP

IF( ZETA1 .LE. 0.0 .AND. ZETA2 .LE. 0.0 ) RETURN

ETA1 = YRO - YLW

ETA2 = YRO - YUP

T1 = SQRT( ETA1*ETA1 + ZRZR )

T2 = SQRT( ETA2*ETA2 + ZRZR )

CONTINUE

R1 = ZETA1 - T1

R2 = ZETA2 - T2

IF( R1 .LE. 0.0 .AND. R2 .LE. 0.0 ) RETURN

R1 = R1*(ZETA1 - T1)

R2 = R2*(ZETA2 - T2)

IF( EREONE ) GO TO 90

EMIRRO = .TRUE.

RTR = R1*R2

RR = 0.5*( R1 + R2 )

IF( SYMK.EQ. 0.0 ) RETURN

ETA1 = YLO - YLW

ETA2 = YLO - YUP

T1 = SQRT( ETA1*ETA1 + ZLLZ )

T2 = SQRT( ETA2*ETA2 + ZLLZ )

GO TO 10

CONTINUE

EMIRRO = .TRUE.

RTL = R1*R2

IF( R1 .LE. 0.0 ) R1 = 0.0
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 FSLEN
LINE 2      REAL MACH
LINE 3 COMMON /BASIC/MACH, BEETA
LINE 4 COMMON /F01/NFREQ, CF(12, 30), FREQ(12), FREMP(12), FRTST, NTMX
LINE 5 C      NTMX THE MAXIMUM NUMBER OF TERMS USED IN WINGR, MINT
LINE 6 C      XUNDM MACHINE DEPENDENT NUMBER, LARGEST NEGATIVE EXPONENT
LINE 7 C      NTMX = 30
LINE 8 CDC XUNDM = -292.0
LINE 10 C WRITE (6, 1001) NFREQ, (FREQ(I), I=1, NFREQ)
LINE 11 C NTMX = 1
LINE 12 DO 5 I = 1, NFREQ
LINE 13 C WRITE (6, 1002) NFREQ, (FREQ(I), I=1, NFREQ), FRTST
LINE 15 C WRITE (6, 1000) NFREQ, (FREQ(I), I=1, NFREQ), FRTST
LINE 16 C NTMX = 1
LINE 18 DO 40 I = 1, NFREQ
LINE 20 XKK = FREQP(I)*FREQP(I)
LINE 21 IF (XKK .GT. 0.0) GO TO 15
LINE 22 DO 10 J = 2, NTMX
LINE 23 C F(I, J) = 0.0
LINE 25 CONTINUE
LINE 26 DO 30 J = 2, NTMX
LINE 27 N2 = N2 - 1
LINE 28 FACTR = XKK/FLOAT(N2*(N2-1))
LINE 30 A10 = ABS4 CF(I, N)
LINE 31 A10 = ALOG10(A10) & ALOG10(FACTOR)
LINE 32 IF (A10 .GE. XUNDM) GO TO 25
LINE 33 DO 20 J1 = 1, NTMX
LINE 34 C F(I, J1) = 0.0
LINE 35 NTMX = MAXO(NTMX, N)
LINE 36 DO 40 J1 = 1, NTMX
LINE 37 CONTINUE
LINE 38 CF(I, J) = -CF(I, N)*FACTOR
LINE 39 CONTINUE
LINE 40 NTMX = NTMX
LINE 41 C DO 40 CONTINUE
LINE 42 C DO 50 I = 1, NFREQ
LINE 44 C WRITE (6, 1001) I, (F(I, J), J=1, NTMX)
LINE 46 C WRITE (1001) //18, 5X, 5E18.5 / (13X, 5E18.5)
LINE 47 C CONTINUE
LINE 48 RETURN
LINE 49 END

-6.24-
DISPLAY FILE DECK 10

LINE 1 SUBROUTINE FREQ(I8,AROW,WROW *AF,LRECL) 01000010
LINE 2 DIMENSION WROW(LRECL), AROW(NWING) 01000020
LINE 3 COMMON /FQ1/ NFREQ 01000030
LINE 4 COMMON/PARAM/ NWING 01000040
LINE 5 EQUIVALENCE (NET,NWING) 01000050
LINE 6 READ(18) K1, WROW 01000060
LINE 7 K=NF 01000070
LINE 8 DO 100 J=1,NWING 01000080
LINE 9 AROW(J) = WROW(K) 01000090
LINE 10 K = K + NFREQ 01000100
LINE 11 100 CONTINUE 01000110
LINE 12 RETURN 01000120
LINE 13 END 01000130
LINE 1  SUBROUTINE GETTIM ( ITIME )  01100010
LINE 2  DIMENSION ITIME ( 8 )  01100020
LINE 3  CALL SECOND ( A )  01100030
LINE 4  ITIME ( 2 ) = A * 1000.0  01100040
LINE 5  RETURN  01100050
LINE 6  END  01100060
SUBROUTINE GRIDIN(CORNX,CORNY,CORNZ,XYZ,1,TRS,NNCH,SW,XLE,
LINE 1  NSP, NE, NETSV, REFLEN, NFREQ, FREQ,
LINE 2  NWIN, NSECT, ICL, ERROR, OEEA, LAB, ID, FD, IS, I6)
LINE 3  U1200010
LINE 4  01200020
LINE 5  01200030
LINE 6  01200040
LINE 7  01200045
LINE 8  01200050
LINE 9  01200060
LINE 10  01200070
LINE 11  01200080
LINE 12  01200090
LINE 13  01200100
LINE 14  01200110
LINE 15  01200120
LINE 16  01200130
LINE 17  01200140
LINE 18  01200150
LINE 19  01200160
LINE 20  01200170
LINE 21  01200180
LINE 22  01200190
LINE 23  01200200
LINE 24  01200210
LINE 25  01200220
LINE 26  01200230
LINE 27  01200240
LINE 28  01200250
LINE 29  01200260
LINE 30  01200270
LINE 31  01200280
LINE 32  01200290
LINE 33  01200300
LINE 34  01200310
LINE 35  01200320
LINE 36  01200330
LINE 37  01200340
LINE 38  01200350
LINE 39  01200360
LINE 40  01200370
LINE 41  01200380
LINE 42  01200390
LINE 43  01200400
LINE 44  01200410
LINE 45  01200420
LINE 46  01200430
LINE 47  01200440
LINE 48  01200450
LINE 49  01200460
LINE 50  01200470
LINE 51  01200480
LINE 52  01200490
LINE 53  01200500
LINE 54  01200510
LINE 55  01200520
LINE 56  01200530
LINE 57  01200540
LINE 58  01200550
LINE 59  01200560
LINE 60  01200570
LINE 61  01200580
LINE 62  01200590
LINE 63  01200600
LINE 64  01200610
LINE 65  01200620
LINE 66  01200630
LINE 67  01200640
LINE 68  01200650
LINE 69  01200660
LINE 70  01200670
LINE 71  01200680
LINE 72  01200690
LINE 73  01200700
LINE 74  01200710
LINE 75  01200720
LINE 76  01200730
LINE 77  01200740
LINE 78  01200750
LINE 79  01200760
LINE 80  01200770
LINE 81  01200780
LINE 82  01200790
LINE 83  01200800
LINE 84  01200810
LINE 85  01200820
LINE 86  01200830
LINE 87  01200840
LINE 88  01200850
LINE 89  01200860
LINE 90  01200870
LINE 91  01200880
LINE 92  01200890
LINE 93  01200900
LINE 94  01200910
LINE 95  01200920
LINE 96  01200930
LINE 97  01200940
LINE 98  01200950
LINE 99  01200960
LINE 100  01200970
LINE 101  01200980
LINE 102  01200990
LINE 103  01201000
LINE 104  01201010
LINE 105  01201020
LINE 106  01201030
LINE 107  01201040
LINE 108  01201050
LINE 109  01201060
LINE 110  01201070
LINE 111  01201080
LINE 112  01201090
LINE 113  01201100
LINE 114  01201110
LINE 115  01201120
LINE 116  01201130
LINE 117  01201140
LINE 118  01201150
LINE 119  01201160
LINE 120  01201170
LINE 121  01201180
LINE 122  01201190
LINE 123  01201200
LINE 124  01201210
LINE 125  01201220
LINE 126  01201230
LINE 127  01201240
LINE 128  01201250
LINE 129  01201260
LINE 130  01201270
LINE 131  01201280
LINE 132  01201290
LINE 133  01201300
LINE 134  01201310
LINE 135  01201320
LINE 136  01201330
LINE 137  01201340
LINE 138  01201350
LINE 139  01201360
LINE 140  01201370
LINE 141  01201380
LINE 142  01201390
LINE 143  01201400
LINE 144  01201410
LINE 145  01201420
LINE 146  01201430
LINE 147  01201440
LINE 148  01201450
LINE 149  01201460
LINE 150  01201470
LINE 151  01201480
LINE 152  01201490
LINE 153  01201500
LINE 154  01201510
LINE 155  01201520
LINE 156  01201530
LINE 157  01201540
LINE 158  01201550
LINE 159  01201560
LINE 57 \( Y_1 = Y(1) \bullet \text{REFLEN} \)
LINE 58 C \( \text{INITIALIZE ELEMENT AND SPAN COUNTERS (NE, AND ISP)} \)
LINE 59 \( \text{NE} = 0 \)
LINE 60 \( \text{ISP} = 0 \)
LINE 61 C \( \text{WRITE (16, 3000) JW, IS} \)
LINE 62 C \( \text{CONTINUE} \)
LINE 63 C \( \text{READ (15, 1000) LAB, ID, FD} \)
LINE 64 10 \( \text{CONTINUE} \)
LINE 65 C \( \text{IF (LAB .EQ. LS)} \) \( \text{GO TO 20} \)
LINE 66 C \( \text{IF (LAB .EQ. LG)} \) \( \text{GO TO 80} \)
LINE 67 C \( \text{IF (LAB .EQ. LF)} \) \( \text{GO TO 50} \)
LINE 68 C \( \text{IF (LAB .EQ. LM)} \) \( \text{GO TO 65} \)
LINE 69 C \( \text{IF (LAB .EQ. LE)} \) \( \text{GO TO 60} \)
LINE 70 C \( \text{IF (LAB .EQ. LB)} \) \( \text{GO TO 70} \)
LINE 71 C \( \text{READ (15, 1600)} \)
LINE 72 C \( \text{WRITE (16, 2000)} \)
LINE 73 C \( \text{WRITE (16, 2018)} \)
LINE 74 C \( \text{WRITE (16, 2026)} \)
LINE 75 C \( \text{WRITE (15, 1000)} \)
LINE 76 C \( \text{READ (15, 2000)} \)
LINE 77 C \( \text{READ (15, 2017)} \)
LINE 78 C \( \text{WRITE (16, 2017)} \)
LINE 79 C \( \text{WRITE (15, 2000)} \)
LINE 80 C \( \text{WRITE (16, 2026)} \)
LINE 81 20 \( \text{CONTINUE} \)
LINE 82 C \( \text{UPDATE SPAN COUNTER AND CHECK CARD} \)
LINE 83 C \( \text{ISP} = \text{ISP} \& \text{41} \)
LINE 84 C \( \text{NC} = \text{ID(2)} \)
LINE 85 C \( \text{YSPAN} = \text{FD(1)} \)
LINE 86 C \( \text{SPAN CARDS ARE OUT OF ORDER} \)
LINE 87 C \( \text{WRITE (15, 1000)} \)
LINE 88 C \( \text{WRITE (15, 2000)} \)
LINE 89 C \( \text{WRITE (16, 2027)} \)
LINE 90 C \( \text{WRITE (16, 2026)} \)
LINE 91 C \( \text{IF (LAB .EQ. LC)} \) \( \text{GO TO 24} \)
LINE 92 21 \( \text{CONTINUE} \)
LINE 93 C \( \text{WRITE (16, 2000)} \)
LINE 94 C \( \text{WRITE (16, 2027)} \)
LINE 95 C \( \text{WRITE (16, 2026)} \)
LINE 96 C \( \text{WRITE (16, 2028)} \)
LINE 97 C \( \text{ISP} = \text{ISP} \& \text{41} \)
LINE 98 22 \( \text{CONTINUE} \)
LINE 99 C \( \text{WRITE (16, 2000)} \)
LINE 100 C \( \text{WRITE (16, 2027)} \)
LINE 101 C \( \text{WRITE (16, 2028)} \)
LINE 102 C \( \text{WRITE (16, 2000)} \)
LINE 103 C \( \text{WRITE (16, 2027)} \)
LINE 104 23 \( \text{CONTINUE} \)
LINE 105 C \( \text{WRITE (16, 3001)} \)
LINE 106 C \( \text{WRITE (16, 3001)} \)
LINE 107 C \( \text{WRITE (16, 2000)} \)
LINE 108 C \( \text{WRITE (16, 2027)} \)
LINE 109 C \( \text{READ 1ST 'CHORD' CARD, MUST BE LABELED 'CHORD'} \)
LINE 110 C \( \text{READ (15, 1000) LAB, ID, FD} \)
LINE 111 C \( \text{IF (LAB .EQ. LC)} \) \( \text{GO TO 10} \)
LINE 112 C \( \text{CHORD CARD NOT FOUND} \)
LINE 113 C \( \text{WRITE (15, 2000)} \)
LINE 114 C \( \text{WRITE (15, 2028)} \)
LINE 115 C \( \text{WRITE (15, 2029)} \)
LINE 116 24 \( \text{CONTINUE} \)
LINE 117 C \( \text{ISP} = \text{ISP} \& \text{41} \)
LINE 118 C \( \text{ISP} = \text{ISP} \& \text{41} \)
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 SW(ISP) = YSPAN/REFLEN
LINE 124 XLE(ISP) = 0.5*(X1*X2)/REFLEN
LINE 125 C LOOP ON NUMBER OF CHORD CARDS 2,NC
LINE 126 C WRITE( 16, 3002 ) IC, X1, X2
LINE 127 C DO 30 IC = 2, NC
LINE 128 C WRITE( 16, 3002 ) IC, X1, X2
LINE 129 C X2 = X2/BEETA
LINE 130 C WRITE( 16, 3002 ) IC, X1, X2
LINE 131 C X1 = X1/BEETA
LINE 132 C WRITE( 16, 3002 ) IC, X1, X2
LINE 133 C X1 = X1/BEETA
LINE 134 C X2 = X2/BEETA
LINE 135 C SW(ISP) = YSPAN/REFLEN
LINE 136 C XLE(ISP) = 0.5*(X1*X2)/REFLEN
LINE 137 C LOOP ON NUMBER OF CHORD CARDS 2,NC
LINE 138 C WRITE( 16, 3002 ) IC, X1, X2
LINE 139 C X1 = X1/BEETA
LINE 140 C WRITE( 16, 3002 ) IC, X1, X2
LINE 141 C X2 = X2/BEETA
LINE 142 C WRITE( 16, 3002 ) IC, X1, X2
LINE 143 C X1 = X1/BEETA
LINE 144 C X2 = X2/BEETA
LINE 145 C X1 = X1/BEETA
LINE 146 C X2 = X2/BEETA
LINE 147 C X1 = X1/BEETA
LINE 148 C X2 = X2/BEETA
LINE 149 C X1 = X1/BEETA
LINE 150 C X2 = X2/BEETA
LINE 151 C X1 = X1/BEETA
LINE 152 C X2 = X2/BEETA
LINE 153 C X1 = X1/BEETA
LINE 154 C X2 = X2/BEETA
LINE 155 C X1 = X1/BEETA
LINE 156 C X2 = X2/BEETA
LINE 157 C X1 = X1/BEETA
LINE 158 C X2 = X2/BEETA
LINE 159 C X1 = X1/BEETA
LINE 160 C X2 = X2/BEETA
LINE 161 C X1 = X1/BEETA
LINE 162 C X2 = X2/BEETA
LINE 163 C X1 = X1/BEETA
LINE 164 C X2 = X2/BEETA
LINE 165 C X1 = X1/BEETA
LINE 166 C X2 = X2/BEETA
LINE 167 C X1 = X1/BEETA
LINE 168 C X2 = X2/BEETA
LINE 169 C X1 = X1/BEETA
LINE 170 C X2 = X2/BEETA
LINE 171 C X1 = X1/BEETA
LINE 172 C X2 = X2/BEETA
LINE 173 C X1 = X1/BEETA
LINE 174 C X2 = X2/BEETA
LINE 175 C X1 = X1/BEETA
LINE 176 C X2 = X2/BEETA
LINE 177 C X1 = X1/BEETA
LINE 178 C X2 = X2/BEETA
LINE 179 C X1 = X1/BEETA
LINE 180 C X2 = X2/BEETA

-6.29-
LINE 181 65 CONTINUE
LINE 182 IF( IFEQ ) GO TO 90
LINE 183 WRITE( 16, 2000 )
LINE 184 WRITE( 16, 2015 )
LINE 185 IF ERR0R = .TRUE. THEN
LINE 186 GO TO 90
LINE 187 C
LINE 188 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 IF ERR0R = .TRUE. THEN
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 .GT. NSE(J) ) GO TO 81
LINE 202 IF( J .GT. NW ) GO TO 81
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 C
LINE 212 ERROR = .TRUE.
LINE 213 GO TO 90
LINE 214 C
LINE 215 82 CONTINUE
LINE 216 C GRID DATA FOR NEXT GRID SECTION IS OUT OF ORDER
LINE 217 WRITE( 16, 2000 )
LINE 218 C
LINE 219 ERROR = .TRUE.
LINE 220 C
LINE 221 90 CONTINUE
LINE 222 C SECTION TO RETURN
LINE 223 C 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(// 1X, 129(1H4) )//24H ERROR IN READING INPUT. )
LINE 233 2031 FORMAT(37H MODE DATA IS MISSING OR OUT OF ORDER )
LINE 234 2035 FORMAT(42H FREQUENCY DATA IS MISSING )//OUT OF ORDER )
LINE 235 2036 FORMAT(42H A LABEL CARD IS EXPECTED BUT NOT PRESENT. )
LINE 236 2037 FORMAT(67H THIS SECTION HAS NOT BEEN DEFINED FOR THE STRUCTURE. )
LINE 237 1E PRINTED ON FOLLOWING LINE AND NEXT CARD WILL BE READ. / 1X,
LINE 238 2 A4, 2X, 9I2,4E12.0 )
LINE 239 2019 FORMAT(12H AN INVALED LABEL CARD WAS READ. )
LINE 240 2020 FORMAT( 27H ERROR ON GRID LABEL CARD. )
LINE 241 2021 FORMAT( 27H GRID DATA PRESENT FOR NW, IS, JW, NSP, ISP )
LINE 242 1 53H THIS SECTION HAS NOT BEEN DEFINED FOR THE STRUCTURE. )
2022 FORMAT (19H GRID DATA FOR WING, 12, 3H SECTION, 12, 17H IS OUT OF ORDER, 01202430
244 1 ORDER.) 01202440
245 2026 FORMAT (43H SPAN CARDS ARE OUT OF ORDER. CARD FOR SPAN, 12, 01202450
246 1 28H WAS READ WHEN CARD FOR SPAN, 12, 13H WAS EXPECTED ) 01202460
247 2027 FORMAT (25H THE WIDTH OF SPAN NUMBER, 12, 7H EQUALS, 64.6 / 01202470
248 1 42H THE SPAN WIDTH MUST BE GREATER THAN ZERO.) 01202480
249 2028 FORMAT (34H THE NUMBER OF CHORD LINES IN SPAN, 12, 7H EQUALS, 13 / 01202490
250 1 48H THE NUMBER OF CHORDS MUST BE GREATER THAN ZERO.) 01202500
251 2029 FORMAT (20H CHORD CARD FOR SPAN, 13, 13H NOT FOUND ) 01202510
252 2030 FORMAT (4H SECTION, 12, 8H OF WING, 12, 20H WAS DEFINED TO HAVE, 01202520
253 1 13, 7H SPANS,, 14, 33H 'SPAN' CARDS WERE ACTUALLY READ. ) 01202530
254 2031 FORMAT (18H CHORD LINE NUMBER, 13, 8H OF SPAN, 13, 01202540
255 1 62H IS IDENTICAL TO, CROSSES, OR IS ABOVE THE PREVIOUS CHORD LINE01202550
256 2 ) 01202560
257 C 01202570
258 3000 FORMAT (1H1/24X, 19H GRID INPUT FOR WING, 12, 8H SECTION, 12 ) 01202580
259 3001 FORMAT (24X, 4H SPAN, 13, 4H HAS, 13, 27H CHORD LINES AND A WIDTH01202590
260 1F, 13.6 / 24X, 5H CHORD, 9X, 2HX1, 14X, 2HX2 ) 01202600
261 3002 FORMAT (24X, 15, 5X, 2E16.6 ) 01202610
262 3006 FORMAT ( // // 23X, 21H NUMBER OF FREQUENCIES01202620
263 1ES, 112 / 20X, 19H LIST OF FREQUENCIES, 5F14.4 / 39X, 5F14.4 ) 01202630
264 END 01202640

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, 1)
PLTR, IWTE, EARO, BETTA, LAB, ID, FD, ERROR )
REAL MACH
LOGICAL ERROR, PLOTR
DIMENSION TITLE(14), FD(4), ID(9)
COMMON / NEXTCS / IFUSH
DATA LRUN, 4HRUN /
C READ RUN CARD
READ( I5, 1000 ) LAB, 10, FD
IF( LAB .EQ. LRUN ) GO TO 2
WRITE( 16, 2000 )
WRITE( 16, 2023 )
IFUSH = 1
2 CONTINUE
ITRUN = ID(1)
PLOTR = .FALSE.
IF( ID(2) .EQ. 1 ) PLOTR = .TRUE.
IWTE = 0
IF( ID(3) .EQ. 1 ) IWTE = 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 )
ERROR = .TRUE.
REFLEN = 1.0
5 CONTINUE
IF( REFLEN .NE. 0.0 ) GO TO 6
WRITE( 16, 2000 )
WRITE( 16, 2025 )
ERROR = .TRUE.
REFLEN = 1.0
6 CONTINUE
IF( EARO .GT. 0.0 ) GO TO 7
WRITE( 16, 2000 )
WRITE( 6, 2027 ) EARO
ERROR = .TRUE.
EARO = 1.0
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
**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 ) 1D(2)

**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( IATE .NE. 1 ) WRITE( 16, 3016 )

**LINE 71**
IF( IATE .EQ. 1 ) WRITE( 16, 3017 )

**LINE 72**
RETURN

**LINE 73**
1000 FORMAT( A49, 2X, 9I29, 4E12.0 )

**LINE 74**
2000 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 75**
2023 FORMAT ( 3FH 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**
1H MACH NUMBER WILL BE SET EQUAL TO 1.0 IN AN ATTEMPT TO CHECK THE REST OF THE DATA.

**LINE 77**
12 Format(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 78**
2024 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 79**
2025 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 80**
2026 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 81**
2027 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 82**
2028 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 83**
2029 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 84**
2030 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 85**
2031 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 86**
2032 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 87**
2033 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 88**
2034 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 89**
2035 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 90**
2036 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 91**
2037 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 92**
2038 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 93**
2039 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 94**
2040 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 95**
2041 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 96**
2042 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 97**
2043 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 98**
2044 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 99**
2045 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 100**
2046 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 101**
2047 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 102**
2048 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 103**
2049 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 104**
2050 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 105**
2051 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 106**
2052 FORMAT(///, 1X, 1291HN, //24H ERROR IN READING INPUT. )

**LINE 107**
END

**-6.33-**
DISPLAY...FILE

DECK 14

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

LINE 1    SUBROUTINE RATE (1,9, NWI)   COMPLEX, D, IP, DNWR, DNWI, R, W, I, NMAX, EPS.
LINE 2    1 NET, A, 12, GNWI(9, NWI)
LINE 3    COMPLEX Gn, GT, sh
LINE 4    COMPLEX Gw, MS
LINE 5    REAL N, NWI
LINE 6    INTEGER PRINT
LINE 7    DIMENSION 0(MC, MC), IP(MC), DNWR(MC), DNWI(MC), R(MC), W(MC)
LINE 8    DIMENSION wvss(200)
LINE 9    DIMENSION wvss(200), NWI(NW), Gv(20), GT(20)
LINE 10   DIMENSION Wv(200)
LINE 11   COMMON /CEPSA/ EPS
LINE 12   COMMON /ITG/ IT
LINE 13   COMMON /KST/ KSTI
LINE 14   COMMON /NMASS/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 15   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 16   COMMON /NMASS/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 17   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 18   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 19   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 20   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 21   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 22   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 23   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 24   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 25   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 26   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 27   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 28   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 29   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 30   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 31   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 32   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 33   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 34   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 35   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 36   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 37   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 38   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 39   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 40   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 41   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 42   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 43   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 44   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 45   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 46   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 47   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 48   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 49   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 50   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 51   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 52   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 53   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 54   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 55   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)
LINE 56   COMMON /VELCOM/ NMAX, PRINT, NWBLOK, NWROW(20)

-6.39-
LINE  57 150 CONTINUE
LINE  58 100 CONTINUE
LINE  59 166 CONTINUE
LINE  60  IF(NMA<.LQK.EQ.1) GO TO 530
LINE  61 172 CONTINUE
LINE  62  REWIND 110
LINE  63 190 IF (LABSPRINT .LT. 3) GO TO 200
LINE  64  C WRITE (6,630) IT
LINE  65  C WRITE (6,650) NROW,(GW(N),N=1,NWNG)
LINE  66 200 CONTINUE
LINE  67  IF (ITEST.EQ.0 .AND. IT.NE.1) GO TO 530
LINE  68  IF (IT.EQ.IMAX) GO TO 540
LINE  69 330 JJ = 0
LINE  70  IT = IT+1
LINE  71  IT = 1
LINE  72  ITEST = 0
LINE  73  REWIND 110
LINE  74  REWIND 110
LINE  75  JJ = 0
LINE  76  IT = 1
LINE  77  L = 0
LINE  78  L = 0
LINE  79  DO 510 I = 1,NWBLNK
LINE  80  NROW = NROW+1
LINE  81  READ (110, D, IP
LINE  82  C IF (LABSPRINT .LT. 3) GO TO 340
LINE  83  C WRITE (6,564) I, NROW
LINE  84  C WRITE (6,566) NROW,0
LINE  85  C566 FORMAT (1HO, 7HTRATE, 7H1ROW = , 113, / (1X,1UF10.5))
LINE  86  C WRITE (6,562) NROW,IP
LINE  87  C340 CONTINUE
LINE  88  C562 FORMAT (1HO, 7HTRATE, 7HROW = , 113, / (1X,1UF10.5))
LINE  89  C564 FORMAT (1HO, 7HTRATE, 7HROW = , 113, 119, 5H RWS )
LINE  90  DO 471 J = 1,NROW
LINE  91  L = L+1
LINE  92  DNWR(J) = 0.0
LINE  93  DNWR(J) = 0.0
LINE  94  C IF (NWING.LE.NMAX) GO TO 470
LINE  95  IFILL .GT. KST(I) NNZ = 0
LINE  96  IFILL .LT. KST(I) GO TO 470
LINE  97  IFILL .LT. L )
LINE  98  READ (15) L1, NNZ, (IZ(M), M=1,NNZ), (AM(M), M=1,NNZ)
LINE  99  IFILL .LT. L ) NNZ = 0
LINE 100  DO 470 J = 1,NNZ
LINE 101  DO 460 M = 1,NNZ
LINE 102  K = IZ(M )
LINE 103  DNWR(J) = DNWR(J) & A(M) = REAL (GW(K)).
LINE 104  DNWR(J) = DNWR(J) & A(M) = AIMAG (GW(K)).
LINE 105 460 DNWR(J) = DNWR(J) & A(M) = REAL (GW(K)).
LINE 106  DNWR(J) = DNWR(J) & A(M) = AIMAG (GW(K)).
LINE 107 470 RWR(J) = RWR(J)-DNWR(J)
LINE 108  RWR(I) = NW1(J)+NW(J)
LINE 109  IF ((ABS(PRINT).LT. 3) GO TO 471
LINE 110 471 CONTINUE
LINE 111  CALL SOLVE ( NWQ,NMAX,O,RWR, IP )
LINE 112  CALL SOLVE (NROW,NMAX,O,RWH, IP )
LINE 113  DO 500 M = 1,NROW
LINE 114  GTH(I) = GW(I)
LINE 115  SAVE = 4W(I)
LINE 116  GWH(I) = CMPLX (RWR(M), RWH(M))
LINE 117  WRH(I) = GWH(I)-GTH(I)
LINE 118  IF (IT.EQ.2) GO TO 450

-6.36-
LINE 119  SAVE=W(I),SAVE  
LINE 120  ALF=ALF1  
LINE 121  IF (SAVE.GE.0.) ALF=ALF2  
LINE 122  490 CONTINUE  
LINE 123  G(W(I))=ALF*G(W(I))-(1.-ALF)*G(T(I))  
LINE 124  C  ITEST = IRELE(G(T(I)),G(W(I)), IT*ST)  
LINE 125  IT=IG1  
LINE 126  500 CONTINUE  
LINE 127  JJ=JNWRGW((I))  
LINE 128  510 CONTINUE  
LINE 129  ITEST = IRELE(GT, GW,NET)  
LINE 130  GO TO 190  
LINE 131  530 WRITE (6,580) IT,EPS  
LINE 132  RETURN  
LINE 133  540 WRITE (6,590) IMAX,EPS  
LINE 134  C IF (IMETH.EQ.0) GO TO 550  
LINE 135  550 CONTINUE  
LINE 136  C GO TO 60  
LINE 137  550 CONTINUE  
LINE 138  WRITE (6,600)  
LINE 139  WRITE (6,650) NWING,(GT(N),N=1,NWING)  
LINE 140  560 WRITE (6,610).  
LINE 141  WRITE (6,650) NWING,(GW(N),N=1,NWING)  
LINE 142  RETURN  
LINE 143  C  
LINE 144  C  
LINE 145  570 FORMAT (2X,10Hw(N),N=1,13/(X,10F10.5)).  
LINE 146  580 FORMAT (1HO,30HTHE ITERATION CONVERGED AFTER ,13,2X,35HITERATIONS  
LINE 147  1WITH A TEST CRITERION OF,F10.7)  
LINE 148  590 FORMAT (1HO,37HTHE ITERATION DID NOT CONVERGE AFTER ,13,2X,35HITERATION IS)  
LINE 149  1ATION8 WITH A TEST CRITERION OF,F10.7)  
LINE 150  600 FORMAT (1HO,40HTHE SOLUTION AT THE PREVIOUS ITERATION IS)  
LINE 151  610 FORMAT (1HO,40HTHE SOLUTION AT THE PRESENT ITERATION IS)  
LINE 152  630 FORMAT (17HITERATION NUMBER,14)  
LINE 153  650 FORMAT (2X,10GW(N),N=1,13/(X,4X,6E18.5)).  
LINE 154  C  
LINE 155  END
SUBROUTINE LOOPW(NET, NWING, XYZ, NSP, SYM, TRS, NNCH, NE, 0160010
LINE 2 1 NSECT, Zsect, ISUNS, IE, TAPER, Xcen, AREA, WROW, LRECL, 0160020
LINE 3 2 WTEROW(LRWT, PERC, ERROR, IWE, REFLEW, 0160030
LINE 4 4 C NET TOTAL NUMBER OF ELEMENTS 0160040
LINE 5 5 C NTERM NUMBER OF TERMS IN INFLUENCE FUNCTION 0160050
LINE 6 6 C NWING NUMBER OF WINGS IN STRUCTURE 0160050
LINE 7 7 C XYZ COORDINATE ARRAY OF ALL ELEMENTS 0160060
LINE 8 8 C NSP NSP(I,J), NW. SPANS IN SECTION I OF WING J 0160070
LINE 9 9 C SYM(I,J) SYMMETRY CODE FOR WING J 0160080
LINE 10 10 C TRS TRS(1,1,1) TRANSFORMATION MATRIX OF SECT. I OF WING J 0160090
LINE 11 11 C NNCH NNCH ELEMENT NUMBER AT END OF SPAN 0160010
LINE 12 12 C NE(I,J) NUMBER OF ELEMENTS IN SECTION I OF WING J 0160010
LINE 13 13 C NSECT(5) NUMBER OF SECTIONS IN WING J 0160010
LINE 14 14 C ZSECT(I,J) Z COORDINATE OF ALL ELEMENTS OF SECTION I, WING J 0160010
LINE 15 15 C ISONS Boolean matrix to tell if one section has influence function 0160010
LINE 16 16 C ISONS(A1,2) = 1 if section A2 has influence on rec. section A1 0160010
LINE 17 17 C I6 IS THE UPPER UNIT NUMBER FOR OUTPUT (PRINTER) 0160010
LINE 18 18 C TAPE6 LOGICAL UNIT NUMBER FOR TAPE TO CONTAIN FREQUENCY-INDEPENDENT TERMS 0160010
LINE 19 19 C 0160010
LINE 20 20 C XCEN X CENTER OF EACH ELEMENT (OUTPUT) 0160010
LINE 21 21 C XO X COORDINATE OF RECEIVING PT. (CENTER OF ELEMENT) 0160010
LINE 22 22 C YRO Y COORDINATE OF RECEIVING PT. (IN RIGHT SIDE) 0160010
LINE 23 23 C YLO Y COORDINATE OF RECEIVING PT. (IN LEFT SIDE) 0160010
LINE 24 24 C AREA AREA FOR EACH ELEMENT (OUTPUT) 0160010
LINE 25 25 C WROW STORAGE FOR FREQUENCY TERMS FOR ONE RECEIVING ELEMENT 0160010
LINE 26 26 C LRECL LENGTH OF WROW EQUAL TO NTERM#NET 0160010
LINE 27 27 C ERROR LOGICAL SET TO TRUE IF ERROR IS DETECTED 0160010
LINE 28 28 C COMPLEX WTEROW(LRWT) 0160010
LINE 29 29 C REAL MACH 0160010
LINE 30 30 C COMPLEX WTES 0160010
LINE 31 31 C INTENGE TAPE8 0160010
LINE 32 32 C LOGICAL BKWSP 0160010
LINE 33 33 C LOGICAL ERROR 0160010
LINE 34 34 C LOGICAL TREDGE 0160010
LINE 35 35 C LOGICAL WAKE, WAKE1, WAKE2 0160010
LINE 36 36 C LOGICAL WAKE2 0160010
LINE 37 37 C DIMENSION IONS(12,1), TRS(4,3,1), NSP(3,1) 0160010
LINE 38 38 C DIMENSION NNCH(1), SYM(1) 0160010
LINE 39 39 C DIMENSION NSECT(1), NE(3,1), ZSECT(3,1), Xcen(1), XYZ(1), AREAI(1) 0160010
LINE 40 40 C DIMENSION WSVI(20) 0160010
LINE 41 41 C DIMENSION WROW(LRECL) 0160010
LINE 42 42 C COMMON/BASIC/MACH 0160010
LINE 43 43 C COMMON / FOI / NPROQ, CF(12,30), FRECQ(12), FREQP(12), FREST 0160010
LINE 44 44 C COMMON / RKL / TVM(4), YRO, YLO, ZR, ZLR, ZL, ZLL, IRC, IN 0160010
LINE 45 45 C COMMON / TAPE/ 15, 110, 111, 112, 113, 114, 115 0160010
LINE 46 46 C COMMON / WAKEUP/ WTES(12), TREDGE, WAKE1, WAKE2, WAKE2 0160010
LINE 47 47 C COMMON / W / XU, YO, ZO, ZO2, ZO20, SYM, NNCH 0160010
LINE 48 48 C COMMON / W/ XU, YO, ZO, ZO2, ZO20, SYM, NNCH 0160010
LINE 49 49 C COMMON / W / XU, YO, ZO, ZO2, ZO20, SYM, NNCH 0160010
LINE 50 50 C THE FREQUENCY-INDEPENDENT TERMS FOR ALL INFLUENCING ELEMENTS 0160010
LINE 51 51 C OF A GIVEN RECEIVING ELEMENT ARE STOPPED IN WROW 0160010
LINE 52 52 C COMPUTE 1 ROW FOR EVERY ELEMENT (CALLED RECEIVING ELEMENT) 0160010
LINE 53 53 C ND TERMS IS THE NO. TERMS USED TO COMPUTE AN ELEMENT OF W 0160010
LINE 54 54 C INDEX INDEX INTO XYZ FOR RECEIVING ELEMENT 0160010
LINE 55 55 C INDEX INDEX INTO XYZ FOR RECEIVING ELEMENT 0160010
LINE 56 56 C INDEX INDEX INTO XYZ FOR RECEIVING ELEMENT 0160010
LINE 57 57 C INDEX INDEX INTO XYZ FOR RECEIVING ELEMENT 0160010
LINE 58 58 C INDEX INDEX INTO XYZ FOR RECEIVING ELEMENT 0160010
LINE 59 59 C INDEX INDEX INTO XYZ FOR RECEIVING ELEMENT 0160010
-6.38-
K1 IS COUNTER FOR RECEIVING SECTION, K2 FOR INFLUENCING SECTION

IF(WTNE.EQ.0) WRITE(U1,160) TAPE(8) MACH,NET,REFLEN,IFREQ

IF(WTNE.EQ.0) WRITE(U1,161) MACH,NET,REFLEN,IFREQ(I),IFREQ(I)

WRITE(U1,162) .FALSE.

INDEX = 1

LOOP ON RECEIVING ELEMENTS BY WING J AND SECTION I OF WING J

DO 10 J = 1,NWING

NSR = NSECT(J)

SYMK = SYM(J)

DO 95 I = 1,NSR

K1 = K1 + 1

NE(I,J) = NE(I,J) + 1

ZSECT(I,J) = Z COORDINATE OF SECTION I,WING J

ZCR = ZSECT(I,J)

LOOP ON THE NUMBER OF ELEMENTS IN SECT. I OF WING J (RECIEV.)

DO 90 IE = I,NER

WAKE2 = .FALSE.

IRC = IRC + 1

FIND CENTER PT. (XO, YCR, ZCR) IN RECEIVING SECTION SYSTEM

XO = XCEN(IRC)

YCR = 0.5*(XYZ(INDXR(4)) + XYZ(INDXR(5)))

DSX = 0.5*(XYZ(INDXR(5)) - XYZ(INDXR(4)))

AREA(IRC) = DSX*(XYZ(INDXR(3)) - XYZ(INDXR(5)) - XYZ(INDXR(4)) - XYZ(INDXR(2)))

ZCR IS CONSTANT FOR ALL ELEMENTS IN SECTION I OF WING J

INDEX = INDEX + 1

IF(IWTE.EQ.0) GO TO 50

DO 45 IE = 1,LRWTE

WTEROW(IEI) = (0.0, 0.0)

CONTINUE

LOOP ON RECEIVING ELEMENTS BY WING J AND SECTION I OF WING J

DO 100 IE = I,NER

IF(IWTE.EQ.0) GO TO 50

CONTINUE
CALL WINTGRI XYZ(INDXII), XYZ(INDXI&), WROW(ICOL)  

SET UP CONSTANTS FOR INFLUENCING ELEMENTS

INDXII = 1
INN = 1
NSPI = 0
K2 = 0
ICOL = 1

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

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

NEI = NEII(JJ)
NSPS = NSPI(JJ)
K2 = K2 & 1
CHECK IF SECTION K2 HAS ZERO INFLUENCE ON SECTION K1
IF( ISONS(K1,K2) .NE. 0 ) GO TO 60

NTNEI = NEI*NFREQ
IEI = I,NTNEI
WROW(ICOL) = 0.0
ICOL = ICOL + 1

DO 80 CONTINUE

SECT. K2 HAS ZERO INFLUENCE ON SECT. K1

NEI NO. ELEMENTS IN SECTION II OF WING JJ
NSPS = NSPI(JJ)
ICOL = ICOL + 1

CONTINUE

SECTION IS ASSUMED TO HAVE SOME ELEMENTS WITH NON-ZERO INFLUENCE. EACH ELEMENT MUST BE CONSIDERED.

TRANSFORM CENTER PT. FROM RECEIVING SECTION SYSTEM TO INFLUENCING SECTION SYSTEM. X IS SAME IN BOTH SYSTEMS.

CALL RTOI (TRSII,II, JJ), TRSII,II, JJ, YCR, ZCR, ZRO, ZLU )
LINE 181 C NSPI = NSPI & 1
LINE 182 C NFPSM1 = NNCH(NSPI) - 1
LINE 183 C NEPS = NEPSM1 & 1
LINE 184 C BUMP INDI AND ICOL TO LAST ELEMENT IN SPAN
LINE 185 C IIN = IIN & NEPS
LINE 186 C INDXI = INDXI & 6*NEPSM1
LINE 187 C ICOL = ICOL & NFREQ*NEPSM1
LINE 188 C BKWSP = .FALSE.
LINE 189 C NINSID = 0
LINE 190 C TREdge = .TRUE.
LINE 191 C LOOP ON INFLU. ELEM. OF SPAN ISP FROM BOTTOM UP DO 70 IEI
LINE 192 C 2.
LINE 193 C DO 70 IEI = 19NEPS
LINE 194 C IF( BKWSP) GO TO 65
LINE 195 C WAKE1 = .FALSE.
LINE 196 C CALL WINTGR( XYZ(INDXI), XYZ(IINUXI C41., WRUW( ICOL))
LINE 197 C TR EDGE = .FALSE.
LINE 198 C IF( VINSID .EQ. 0 ) GO TO 69
LINE 199 C IF( NOT. WAKE1 ) GO TO 58
LINE 200 C WAKE2 = .TRUE.
LINE 201 C INDWK = NFREQ*(NSPI-1)
LINE 202 C DO 56 IW=L,NFREQ
LINE 203 C INDWK = INDWK & 1
LINE 204 C WTER(w(INDWK) = WTES(IW)
LINE 205 C CONTINUE
LINE 206 C IF( VINSID .EQ. 1 ) GO TO 69
LINE 207 C RD = ABS( WROW(icol) / WII )
LINE 208 C IF( RD > GT. PERC ) GO TO 62
LINE 209 C BKWSP = .TRUE.
LINE 210 C IJ = ICOL - 1
LINE 211 C DO 68 IF=1,NFREQ
LINE 212 C WROW(IJ) = 0.0
LINE 213 C WFSV(IF) = 0.0
LINE 214 C CONTINUE
LINE 215 C IF( VINSID .LE. 2 ) GO TO 69
LINE 216 C IJ = ICOL & NFREQ - 1
LINE 217 C IF( VINSID .GT. 3 ) GO TO 63
LINE 218 C TEST = WROW(IJ)
LINE 219 C GO TO 69
LINE 220 C IF( TEST = TEST - WROW(IJ) )
LINE 221 C IF( TEST .NE. 0.0 ) TEST = TEST/ WROW(IJ)
LINE 222 C IF( TEST .GT. .02 ) GO TO 61
LINE 223 C BKWSP = .TRUE.
LINE 224 C IJ = ICOL - 1
LINE 225 C DO 64 IF=1,NFREQ

-6.41-
LINE 243  IJ = IJ & 1  01602430
LINE 244  64 WFSV(IF) = WROW(IJ)  01602440
LINE 245   GO TO 69  01602450
LINE 246  C  01602460
LINE 247  65 CONTINUE  01602470
LINE 248   IJ = ICOL - 1  01602480
LINE 249   DO 67 IF = 1,NFREQ  01602490
LINE 250   IJ = IJ & 1  01602500
LINE 251  67 WROW(IJ) = WFSV(IF)  01602510
LINE 252  C  01602520
LINE 253  65 CONTINUE  01602530
LINE 254   ICOL = ICOL - NFREQ  01602540
LINE 255   INDI = INDI - 6  01602550
LINE 256  C  01602560
LINE 257  END LOOP ON ELEMENTS IN THE INFLUENCING SPAN  01602570
LINE 258  70 CONTINUE  01602580
LINE 259  C  01602590
LINE 260   IIN = IIN & NEPS  01602600
LINE 261   INDI = INDI & 6*NEPS & 6  01602610
LINE 262   ICOL = ICOL & NFREQ*NEPS & NFREQ.  01602620
LINE 263  C  01602630
LINE 264  END LOOP ON INFLUENCING SPANS OF SECTION II OF WING JJ  01602640
LINE 265  75 CONTINUE  01602650
LINE 266  C  01602660
LINE 267  END LOOP ON INFLUENCING SECTION II OF WING JJ  01602670
LINE 268  80 CONTINUE  01602680
LINE 269  C  01602690
LINE 270  END LOOP ON INFLUENCING WING JJ  01602700
LINE 271  85 CONTINUE  01602710
LINE 272  C  01602720
LINE 273  WROW 1 ROW HAS BEEN COMPUTED FOR A GIVEN RECEIVING ELEMENT.  01602730
LINE 274  WRITE WROW ON FILE AND LOOP TO NEXT RECEIVING ELEMENT  01602740
LINE 275  C  01602750
LINE 276  WRITE( TAPE8) IRC, WROW  01602760
LINE 277  C  01602770
LINE 278  WAKE = FALSE - NO WAKE ELEMENTS FOR ANY ELEMENT  01602780
LINE 279  C  01602790
LINE 280  IF (IWE .EQ.0) GO TO 90  01602800
LINE 281  IF( .NOT. WAKE2 ) GO TO 90  01602810
LINE 282  WRITE (115) IRC, WTEROW  01602820
LINE 283  KWAKE = IRC  01602830
LINE 284  KW = KW + 1  01602840
LINE 285  C  01602850
LINE 286  IF(IWE .NE.0) WRITE(6,3001) IRC, IER, I, J, WTEROW  01602860
LINE 287  C  01602870
LINE 288  WRITE( 16, 3001 ) IRC, IER, I, J, (WROW(IJ), IJ=IJ, ICOL )  01602880
LINE 289  C  01602890
LINE 290  ICOL = ICOL + 1  01602900
LINE 291  C  01602910
LINE 292  WRITE( 16,3001 ) IRC, IER, I, J, WROW  01602920
LINE 293  C  01602930
LINE 294  END LOOP ON RECEIVING ELEMENT IN SECTION I OF WING J  01602940
LINE 295  90 CONTINUE  01602950
LINE 296  C  01602960
LINE 297  END LOOP ON RECEIVING SECTION I OF WING J  01602970
LINE 298  C  01602980
LINE 299  END LOOP ON RECEIVING WING J  01602990
LINE 300  100 CONTINUE  01603000
LINE 301  C  01603010
LINE 302  110 CONTINUE  01603020
LINE 303  C  01603030
LINE 304  C  01603040

---6.42---
LINE 305  IF(WAKE)   END FILE 115
LINE 306   16 =6
LINE 307  IF(.NOT. WAKE) WRITE(16,1000)
LINE 308  IF(WAKE) WRITE (16,1010) KW
LINE 309  1000 FORMAT(1HO///16X,37H WAKE EFFECTS HAVE BEEN DETERMINED )
LINE 310  1010 FORMAT(1HO///16X,38H WAKE EFFECTS HAVE BEEN DETERMINED FOR 114 ,
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 // (1X, 10E13.4 ) )
LINE 316  END
DISPLAY...FILE

DECK 17

LINE  1  SUBROUTINE MESH(CORNX,CORNY,CORNZ, XYZ, Z, TRS, NNCH, S4, XLE, 01700010
LINE  2   1
LINE  3        NSP, NE, NETSV, EA0)
LINE  4   01700020
LINE  5   01700030
LINE  6   01700040
LINE  7   01700050
LINE  8   01700060
LINE  9   01700070
LINE 10   01700080
LINE 11   01700090
LINE 12   01700100
LINE 13   01700110
LINE 14   01700120
LINE 15   01700130
LINE 16   01700140
LINE 17   01700150
LINE 18   01700160
LINE 19   01700170
LINE 20   01700180
LINE 21   01700190
LINE 22   01700200
LINE 23   01700210
LINE 24   01700220
LINE 25   01700230
LINE 26   01700240
LINE 27   01700250
LINE 28   01700260
LINE 29   01700270
LINE 30   01700280
LINE 31   01700290
LINE 32   01700300
LINE 33   01700310
LINE 34   01700320
LINE 35   01700330
LINE 36   01700340
LINE 37   01700350
LINE 38   01700360
LINE 39   01700370
LINE 40   01700380
LINE 41   01700390
LINE 42   01700400
LINE 43   01700410
LINE 44   01700420
LINE 45   01700430
LINE 46   01700440
LINE 47   01700450
LINE 48   01700460
LINE 49   01700470
LINE 50   01700480
LINE 51   01700490
LINE 52   01700500
LINE 53   01700510
LINE 54   01700520
LINE 55   01700530
LINE 56   01700540

NOTE THAT ELEMENT IS NUMBERED CLOCKWISE WHILE SECT IS COUNTERCLOCKWISE.

ASSUME CONSTANT SLOPE FOR ALL LINES.

-6.44-
REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

LINE  57  \[ DX12 = X2 - X1 \]
LINE  58  \[ DS = (Y3-Y1)/NSP \]
LINE  59  \[ DC = CRAT10*DS \]
LINE  60  \[ DXLE = (X3-X1)/NSP \]
LINE  61  \[ DXTE = (X4-X2)/NSP \]
LINE  62  \[ COMPUTE NC1, DC1 FOR AREA 1 \]
LINE  63  \[ NC1=1 \]
LINE  64  \[ DC1 = AMAX1(DX12, DX34) \]
LINE  65  \[ IF( DC1 < DC ) GO TO 11 \]
LINE  66  \[ NC1 = DC1/DC \]
LINE  67  \[ DC1 = DC1/NC1 \]
LINE  68  \[ CONTINUE \]
LINE  69  \[ NE=0 \]
LINE  70  \[ LOOP ON SPANS \]
LINE  71  \[ DO 100 I=1,NSP \]
LINE  72  \[ FIND THE Y COORD. FOR THIS SPAN (AY AND BY) \]
LINE  73  \[ BY = Y1 + I*DS \]
LINE  74  \[ AX = X1 + I*DXLE \]
LINE  75  \[ SW(I) = DS \]
LINE  76  \[ XLE(I) = 0.5*(AX+BX) \]
LINE  77  \[ FIND CX,DX THE X COORD ON TRAILING EDGE \]
LINE  78  \[ CX = X2 + I*DXTE \]
LINE  79  \[ DX = CX - DXTE \]
LINE  80  \[ DXM = DTEST*DC1 \]
LINE  81  \[ CXM = CX - DXM \]
LINE  82  \[ DXM = DX - DXM \]
LINE  83  \[ LOOP ON NC1 FOR AREA 1 \]
LINE  84  \[ DO 20 J1 = 1,NC1 \]
LINE  85  \[ NJ = 6*NE \]
LINE  86  \[ NE=NE+1 \]
LINE  87  \[ XYZ(VJG5)=AV \]
LINE  88  \[ XYZ(VJG6)=BY \]
LINE  89  \[ XD = J1*OC1 \]
LINE  90  \[ XC = BX & XD \]
LINE  91  \[ XD = AX & XD \]
LINE  92  \[ XA = XD - DC1 \]
LINE  93  \[ XB = XC - DC1 \]
LINE  94  \[ XYZ(VJG7)=X \]
LINE  95  \[ XYZ(VJG8)=XB \]
LINE  96  \[ XYZ(VJG9)=XC \]
LINE  97  \[ XYZ(VJG10)=X \]
LINE  98  \[ CHECK TO SEE IF WE HAVE REACHED TRAILING EDGE (CX AND DX) \]
LINE  99  \[ IF( XC < GT, CX ) OR. XD < GT, DX ) GC TO 19 \]
LINE  100  \[ GO TO 20 \]
LINE  101  \[ CONTINUE \]
LINE  102  \[ XYZ(VJG11)=X \]
LINE  103  \[ XYZ(VJG12)=X \]
LINE  104  \[ XYZ(VJG13)=X \]
LINE  105  \[ XYZ(VJG14)=X \]
LINE  106  \[ XYZ(VJG15)=X \]
LINE  107  \[ XYZ(VJG16)=X \]
LINE  108  \[ XYZ(VJG17)=X \]
LINE  109  \[ XYZ(VJG18)=X \]
LINE  110  \[ XYZ(VJG19)=X \]
LINE  111  \[ XYZ(VJG20)=X \]
LINE  112  \[ IF( XA < GT, XA ) OR. XB < GT, XB ) GC TO 19 \]
LINE  113  \[ GO TO 20 \]
LINE  114  \[ CONTINUE \]
LINE  115  \[ XYZ(VJG21)=X \]
LINE  116  \[ XYZ(VJG22)=X \]
LINE  117  \[ XYZ(VJG23)=X \]
LINE  118  \[ XYZ(VJG24)=X \]
LINE 119    20 CONTINUE
LINE 120    C
LINE 121    60 CONTINUE
LINE 122    NETV(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, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 1
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 2
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 3
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 4
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 5
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 6
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 7
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 8
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 9
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 10
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 11
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 12
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 13
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 14
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 15
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 16
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 17
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 18
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 19
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 20
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 21
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 22
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 23
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 24
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 25
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 26
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 27
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 28
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 29
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 30
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 31
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 32
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 33
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 34
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 35
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 36
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 37
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 38
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 39
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 40
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 41
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 42
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 43
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 44
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 45
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 46
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 47
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 48
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 49
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 50
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 51
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 52
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 53
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 54
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 55
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)

LINE 56
SUBROUTINE MESHCL(CORNX, CORNY, CORNZ, XYZ, XLE, SW, NNCH, TRS, X, Y, Z)
LINE 57 C FIND CRATIO (CHORD TO SPAN RATION) 01800570
LINE 58 C 01800580
LINE 59 C 01800590
LINE 60 DC = Y3 - Y1 01800600
LINE 61 DS = (X3 - X1)*BEETA 01800610
LINE 62 CRATIO = 01800620
LINE 63 C / (DC*BEETA) 01800630
LINE 64 C A SSUME CONSTANT SLOPE FOR ALL LINES 01800640
LINE 65 C 01800650
LINE 66 DX34 = X4 - X3 01800660
LINE 67 DX22 = X2 - X1 - DX34 01800670
LINE 68 DS = (Y3-Y1)/NSP 01800680
LINE 69 DC = CRATIO*DS 01800690
LINE 70 DLX2 = (X3-X1)/NSP 01800700
LINE 71 DXL2 = (X4-X2)/NSP 01800710
LINE 72 C FIND INFORMATION FOR AREA BETWEEN LEADING EDGE AND 1ST CLINE 01800720
LINE 73 C 01800730
LINE 74 C 01800740
LINE 75 CL1 = CL(1) 01800750
LINE 76 CL2 = CL(2) 01800760
LINE 77 DXC1 = (CL2-CL1)/NSP 01800770
LINE 78 NC1 = 1 01800780
LINE 79 DC1 = AMAX1(CL1-X1,CL2-X3) 01800790
LINE 80 IF( DC1 .LT. DC ) GO TO 11 01800800
LINE 81 NC1 = DC1/DC 01800810
LINE 82 DC1 = DC1/NC1 01800820
LINE 83 11 CONTINUE 01800830
LINE 84 C 01800840
LINE 85 C 01800850
LINE 86 C 01800860
LINE 87 NC2 = 0 01800870
LINE 88 IF( ICL .EQ. 1 ) GO TO 14 01800880
LINE 89 C 01800890
LINE 90 CL3 = CL(3) 01800900
LINE 91 CL4 = CL(4) 01800910
LINE 92 DXC2 = (CL4-CL3)/NSP 01800920
LINE 93 C FIND DC3 FOR AREA BETWEEN SECOND CONTROL LINE AND TRAILING EDGE 01800930
LINE 94 DC3 = AMAX1(X2-CL3, X4-CL4) 01800940
LINE 95 NC3 = 1 01800950
LINE 96 DC2 = AMAX1(CL3-CL1, CL4-CL2) 01800960
LINE 97 IF( DC2 .LT. DC ) GO TO 16 01800970
LINE 98 NC2 = DC2/DC 01800980
LINE 99 DC2 = DC2/NC2 01800990
LINE 100 GO TO 16 01801000
LINE 101 C 01801010
LINE 102 C 01801020
LINE 103 14 CONTINUE 01801030
LINE 104 C FIND DC3 FOR AREA BETWEEN FIRST CONTROL LINE AND TRAILING EDGE 01801040
LINE 105 DC3 = AMAX1(X2-CL1, X4-CL2) 01801050
LINE 106 C 01801060
LINE 107 C 01801070
LINE 108 C 01801080
LINE 109 NC3 = 1 01801090
LINE 110 IF( DC3 .LT. DC ) GO TO 18 01801100
LINE 111 NC3 = DC3/DC 01801110
LINE 112 DC3 = DC3/NC3 01801120
LINE 113 18 CONTINUE 01801130
LINE 114 C 01801140
LINE 115 C 01801150
LINE 116 C 01801160
LINE 117 C LOOP ON SPANS 01801170
LINE 118 NE = 0 01801180

-6.48-
DO 100 I=1,NSP

LINE 122 C FIND THE Y COORD. FOR THIS SPAN (AY AND BY)

LINE 123 BY = Y1 & I*DS

LINE 124 AY = BY - DS

LINE 125 C FIND AX,BX, X COORD ON LEADING EDGE ON BOTH SIDES OF SPAN

LINE 126 BX = X1 & I*DXLE

LINE 127 AX = BX - DXLE

LINE 128 SW(I) = DS

LINE 129 XLE(I) = 0.5*(AXXBX)

LINE 130 C FIND CX,DX THE X COORD ON TRAILING EDGE

LINE 131 CX = X2 & I*JXTE

LINE 132 DX = CX - DXTE

LINE 133 C FIND ACL1, BCL1 X COORD ON FIRST CONTROL LINE

LINE 134 BCL1 = CL1 & I*DXC1

LINE 135 ACL1 = BCL1 - DXC1

LINE 136 C

LINE 137 TDI = DTEST*DC1

LINE 138 TC1 = BCL1 - TDI

LINE 139 TDI = ACL1 - TD1

LINE 140 C

LINE 141 DXM = DTEST*DC3

LINE 142 C

LINE 143 C LOOP ON NCI FOR AREA 1

LINE 144 C

LINE 145 C DO 20 J1 = 1,NC1

LINE 146 C

LINE 147 C

LINE 148 NJ = 6*NE

LINE 149 NE=NE+1

LINE 150 XYZ(NJ&5)=AY

LINE 151 XYZ(NJ&6)=BY

LINE 152 XD = J1*DC1

LINE 153 XC = BX & XD

LINE 154 XD = AX & XD

LINE 155 XA = XD - DC1

LINE 156 XB = XC - DC1

LINE 157 XYZ(NJ&7)=XAX

LINE 158 XYZ(NJ&8)=XAX

LINE 159 XYZ(NJ&9)=XC

LINE 160 XYZ(NJ&10)=XD

LINE 161 C CHECK TO SEE IF WE HAVE REACHED THE FIRST CONTROL LINE

LINE 162 IFI XC GT CL1 OR XD GT ACL1 GO TO 19

LINE 163 J1=20

LINE 164 CONTINUE

LINE 165 XYZ(NJ&13)=BCL1

LINE 166 XYZ(NJ&14)=ACL1

LINE 167 GO TO 25

LINE 168 C

LINE 169 20 CONTINUE

LINE 170 C

LINE 171 25 CONTINUE

LINE 172 C IFI ICL EQ. 1 I GO TO 40

LINE 173 C SECTION FOR AREA 2

LINE 174 C

LINE 175 C

LINE 176 C

LINE 177 C FIND ACL2, BCL2 X COORD ON SECOND CONTROL LINE

LINE 178 BCL2 = CL3 & I*DXC2

LINE 179 ACL2 = BCL2 - DXC2

LINE 180 C
<table>
<thead>
<tr>
<th>LINE</th>
<th>Instructions</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>181</td>
<td>TD2 = DTEST * DC2</td>
<td></td>
</tr>
<tr>
<td>182</td>
<td>TC2 = ACL2 - TD2</td>
<td></td>
</tr>
<tr>
<td>183</td>
<td>TD2 = ACL2 - TD2</td>
<td></td>
</tr>
<tr>
<td>184 C</td>
<td>LOOP ON NC2 FOR AREA 2</td>
<td></td>
</tr>
<tr>
<td>185 C</td>
<td>DO 30 J1 = 1, NC2</td>
<td></td>
</tr>
<tr>
<td>186 C</td>
<td>NJ = 6 * NE</td>
<td></td>
</tr>
<tr>
<td>187 C</td>
<td>NC = NE + 1</td>
<td></td>
</tr>
<tr>
<td>188 C</td>
<td>XYZ(J1, 5) = AY</td>
<td></td>
</tr>
<tr>
<td>189 C</td>
<td>XYZ(J1, 6) = AY</td>
<td></td>
</tr>
<tr>
<td>190 C</td>
<td>XC = BCL1 &amp; XD</td>
<td></td>
</tr>
<tr>
<td>191 C</td>
<td>XD = J1 * DC2</td>
<td></td>
</tr>
<tr>
<td>192 C</td>
<td>XA = JC - DC2</td>
<td></td>
</tr>
<tr>
<td>193 C</td>
<td>XB = JC - DC2</td>
<td></td>
</tr>
<tr>
<td>194 C</td>
<td>XYZ(J1, 11) = XA</td>
<td></td>
</tr>
<tr>
<td>195 C</td>
<td>XYZ(J1, 62) = XB</td>
<td></td>
</tr>
<tr>
<td>196 C</td>
<td>XYZ(J1, 63) = XC</td>
<td></td>
</tr>
<tr>
<td>197 C</td>
<td>XYZ(J1, 64) = XD</td>
<td></td>
</tr>
<tr>
<td>198 C</td>
<td>CHECK TO SEE IF WE HAVE REACHED SECOND CONTROL LINE</td>
<td></td>
</tr>
<tr>
<td>199 C</td>
<td>IF (XC .GT. ACL2 .OR. XD .GT. ACL2) GO TO 29</td>
<td></td>
</tr>
<tr>
<td>200 C</td>
<td>GO TO 30</td>
<td></td>
</tr>
<tr>
<td>201 C</td>
<td>XYZ(4J &amp; 5) = AX</td>
<td></td>
</tr>
<tr>
<td>202 C</td>
<td>LINE 203 29 CONTINUE</td>
<td></td>
</tr>
<tr>
<td>203 C</td>
<td>XYZ(J1, 63) = BCL2</td>
<td></td>
</tr>
<tr>
<td>204 C</td>
<td>LINE 207 30 CONTINUE</td>
<td></td>
</tr>
<tr>
<td>205 C</td>
<td>XYZ(J1, 64) = ACL2</td>
<td></td>
</tr>
<tr>
<td>206 C</td>
<td>GO TO 45</td>
<td></td>
</tr>
<tr>
<td>207 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>208 C</td>
<td>XYZ(4J &amp; 1) = AX</td>
<td></td>
</tr>
<tr>
<td>209 C</td>
<td>XYZ(4J &amp; 2) = AX</td>
<td></td>
</tr>
<tr>
<td>210 C</td>
<td>XYZ(4J &amp; 3) = AX</td>
<td></td>
</tr>
<tr>
<td>211 C</td>
<td>XYZ(4J &amp; 4) = AX</td>
<td></td>
</tr>
<tr>
<td>212 C</td>
<td>CHECK TO SEE IF WE HAVE REACHED SECOND CONTROL LINE</td>
<td></td>
</tr>
<tr>
<td>213 C</td>
<td>IF (XC .GT. ACL2 .OR. XD .GT. ACL2) GO TO 29</td>
<td></td>
</tr>
<tr>
<td>214 C</td>
<td>GO TO 30</td>
<td></td>
</tr>
<tr>
<td>215 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>216 C</td>
<td>XYZ(J1, 63) = BCL2</td>
<td></td>
</tr>
<tr>
<td>217 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>218 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>219 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>220 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>221 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>222 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>223 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>224 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>225 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>226 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>227 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>228 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>229 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>230 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>231 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>232 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>233 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>234 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>235 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>236 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>237 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>238 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>239 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>240 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>241 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
<tr>
<td>242 C</td>
<td>PLAN 45</td>
<td></td>
</tr>
</tbody>
</table>

-6.50-
LINE 243  X W J Y = 0 X  OR I G I N A L  P A G E  2 4 1
LINE 244  G O  T O  6 0  O F  P O O R  Q U A L I T Y
LINE 245  5 0  C O N T I N U E
LINE 246  C
LINE 247  6 0  C O N T I N U E
LINE 248  N N C M I = N E  &  N E T S V
LINE 249  C
LINE 250  1 0 0  C O N T I N U E
LINE 251  C
LINE 253  N E T S V = N E T S V  &  N E
LINE 254  R E T U R N
LINE 255  E N D
DISPLAY...FILE

DECK 19

LINE 1  SUBROUTINE MFUN ( JMODE)
LINE 2  COMMON / MODEF / EM(10), DM(10), X, Y, J1, J2
LINE 3  DIMENSION C T A B L E (6,20), IFTABL(2,20)
LINE 4  COMMON /MCOM/ C T A B L E, IFTABL, NTABL, NEQF
LINE 5  DO 200  IH = 1, JMODE
LINE 6  EM(IH) = 0.0
LINE 7  DM(IH) = 0.0
LINE 8  DO 100  I = 1, NEQF
LINE 9   IFW = IFTABL(1, I)
LINE 10  IFM = IFTABL(2, I)
LINE 11  IF( IFW.EQ. J1).AND. ( IFM.EQ. IH ) GO TO 90
LINE 12  GO TO 100
LINE 13  90  CO = C T A B L E( 1, I)
LINE 14  CX = C T A B L E( 2, I)
LINE 15  CY = C T A B L E( 3, I)
LINE 16  CXY = C T A B L E( 4, I)
LINE 17  CX2 = C T A B L E( 5, I)
LINE 18  CY2 = C T A B L E( 6, I)
LINE 19  EM(IH) = CO & CX & X & CY & Y & CXY & X & Y & CX2 & X**2 & CY2 & Y**2
LINE 20  DM(IH) = CX & CXY & Y & 2.0 & CX2 & X
LINE 21  GO TO 200
LINE 22  100 CONTINUE
LINE 23  200 CONTINUE
LINE 24  RETURN
LINE 25  END
DISPLAY...FILE

DECK 20

LINE 1  SUBROUTINE MXERR(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, 2OHMAXIMUM NO. WINGS = 113,6X, 4H NO. 02000120
LINE 13  1 ,9H INPUT = 113 ) 02000130
LINE 14  720 FORMAT(1HO / 8X,3OHMAXIMUM NO. SECTIONS / WING = 111, 6X, 4H NO. 02000140
LINE 15  1 ,9H INPUT = 113 ) 02000150
LINE 16  730 FORMAT(1HO / 8X,2OHMAXIMUM NO. MODES = 113,6X, 4H NO. 02000160
LINE 17  1 ,9H INPUT = 113 ) 02000170
LINE 18  740 FORMAT(1HO / 8X,2OHMAXIMUM NO. SPANS = 113,6X, 4H NO. 02000180
LINE 19  1 ,9H INPUT = 113 ) 02000190
LINE 20  750 FORMAT(1HO / 8X,28HMAXIMUM NO. CORNER POINTS = 111,6X, 4H NO. 02000200
LINE 21  1 ,9H INPUT = 113 ) 02000210
LINE 22  760 FORMAT(1HO / 8X,26HMAXIMUM NO. FREQUENCIES = 112,6X, 4H NO. 02000220
LINE 23  1 ,9H INPUT = 113 ) 02000230
LINE 24  770 FORMAT(1HO / 8X,2CHMAXIMUM NO. ELEMENTS / SPAN = 112,6X, 4H NO. 02000240
LINE 25  1 ,9H INPUT = 112 ) 02000250
LINE 26  780 FORMAT(1HO, 8X, 36HMAXIMUM NO. POINTS ON SPLINE DATA = 113,6X, 4H NO. 02000260
LINE 27  1 4H NO., 9H 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

-6.53-
CIS PLAY.

FILE DECK 21

LINE 1 SUBROUTINE OUTPI (MACH, FREQ, JMODE, JWING, NS, NSP, SW, DLSPAN, PMSPAN, XP, INDXI, PRES, IPW, 16, LINE, LMAX)

LINE 2 1

LINE 4 C MACH MACH NUMBER 02100010

LINE 5 C FREQ FREQUENCY 02100060

LINE 6 C JMODE MODE NUMBER 02100070

LINE 7 C JWING WING NUMBER 02100080

LINE 8 C NS NUMBER OF SECTIONS IN JWING 02100090

LINE 9 C NSP NUMBER OF SPANS IN EACH SECTION OF JWING (ARRAY) 02100100

LINE 10 C DLSPAN LIFT/UNIT SPAN FOR EACH SPAN OF JWING (ARRAY) 02100110

LINE 11 C PMSPAN PITCHING MOMENT/UNIT SPAN FOR EACH SPAN (ARRAY) 02100120

LINE 12 C XP

LINE 13 C INDXI ELEMENT NUMBER OF FIRST ELEMENT OF JWING 02100130

LINE 14 C PRES ELEMENT PRESSURES FOR JWING (ARRAY) 02100140

LINE 15 C IPW PRINT CONTROL FOR JWING 02100150

LINE 16 C IPW=1 LIFT AND PRESSURE MOMENTS/UNIT SPAN TO BE PRINTED 02100160

LINE 17 C IPW=2 ELEMENT PRESSURES TO BE PRINTED 02100170

LINE 18 C IPW=3 BOTH OF THE ABOVE TO BE PRINTED 02100180

LINE 19 C IPW=0 NONE OF THE ABOVE TO BE PRINTED 02100190

LINE 20 C XP

LINE 21 C SW SPAN WIDTH (ARRAY) 02100200

LINE 22 C LINE CURRENT LINE ON OUTPUT PAGE 02100210

LINE 23 C LMAX MAXIMUM LINE ON A PRINTED PAGE 02100220

LINE 24 C NNCH ARRAY CONTAINING ELEMENT NUMBER OF THE LAST ELEMENT OF JWING 02100230

LINE 25 C NE ARRAY CONTAINING THE NUMBER OF ELEMENTS IN EACH SECTION OF JWING. 02100240

LINE 26 C 02100250

LINE 27 C REAL MACH, 02100260

LINE 28 C COMPLEX DLSPAN(1), PMSPAN(1), PRES(1) 02100270

LINE 29 C 02100280

LINE 30 C DIMENSION NNCH(1), NE(1) 02100290

LINE 31 C DIMENSION NSP(1), SW(1) 02100300

LINE 32 C DIMENSION TITLE(14) 02100310

LINE 33 C 02100320

LINE 34 C DD 15 KSPCT = INDXI 02100330

LINE 35 C IF (MACH(KSPCT) .GE. MACH) GO TO 17 02100340

LINE 36 C IF (LMAX - LINE .GT. 14) GO TO 25 02100350

LINE 37 C WRITE (16, 2015) TITILE, FREQ 02100360

LINE 38 C 02100370

LINE 39 C 02100380

LINE 40 C CONTINUE 02100390

LINE 41 C 02100400

LINE 42 C IF (LINE .NE. LMAX) GO TO 25 02100410

LINE 43 C WRITE (16, 2015) TITILE, FREQ 02100420

LINE 44 C 02100430

LINE 45 C 02100440

LINE 46 C 02100450

LINE 47 C 02100460

LINE 48 C 02100470

LINE 49 C 02100480

LINE 50 C 02100490

LINE 51 C WRITE LIFT AND PRESSURE MOMENT TABLE 02100500

LINE 52 C 02100510

LINE 53 C IF (LINE .GT. 14) GO TO 30 02100520

LINE 54 C WRITE (16, 2020) 02100530

LINE 55 C 02100540

LINE 56 C GO TO 35 02100550

-6.54-
LINE 57  30 CONTINUE
LINE 58  WRITE(16,201) 
LINE 59  LINE = LINE & 12
LINE 60  35 CONTINUE
LINE 61  WRITE(16,2001) JWING
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  DO 120 ISEC=1,NS
LINE 69  NSPAN = NSP(ISEC)
LINE 70  C
LINE 71  DO 115 ISPA=1,NSPAN
LINE 72  WRITE(16,2004) ISEC,ISPAN,SW(ISPCT),DSPAN(ISPCT),PSSPAN(ISPCT)
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(16,2005) JWING
LINE 85  C
LINE 86  C
LINE 87  C
LINE 88  C
LINE 89  C
LINE 90  C
LINE 91  C
LINE 92  C
LINE 93  C
LINE 94  WRITE(16,2020)
LINE 95  C
LINE 96  GO TO 135
LINE 97  C
LINE 98  WRITE(16,2021)
LINE 99  C
LINE 100  135 CONTINUE
LINE 101  WRITE(16,2005) JWING
LINE 102  C
LINE 103  WRITE(16,2002) JMODE, FREQ, MACH
LINE 104  WRITE(16,2009)
LINE 105  INDEX = INDEX + 1
LINE 106  ISPCT = KSPCT
LINE 107  C
LINE 108  DO 220 ISEC=1,NS
LINE 109  NSPAN = NSP(ISEC)
LINE 110  C
LINE 111  DO 215 ISPA=1,NSPAN
LINE 112  NEPS = NNCH(ISPCT) - INDEX + 1
LINE 113  DO 210 IEL = 1,NEPS
LINE 114  WRITE(16,2010) ISEC,ISPAN,INDEX,PHIW(INDEX),PRE(INDEX)
LINE 115  INDEX = INDEX + 1
LINE 116  210 CONTINUE
LINE 117  ISPCT = ISPCT + 1
LINE 118  215 CONTINUE
LINE 119 220 CONTINUE
LINE 120 C
LINE 121   LINE = LINE & NEWJ
LINE 122   LINE = NEWJ/3 & LINE
LINE 123   IF( 40D(NEWJ,3) .GT. 0 ) LINE=LINE1
LINE 124   LINE = MOD(LINE, LMAX)
LINE 125 C
LINE 126 RETURN
LINE 127 2001 FORMAT(46X, 47HLIFT AND PITCHING MOMENT PER UNIT SPAN FOR WING )
LINE 128   ING, (12 )
LINE 129 2002 FORMAT(1HG, 5X, 11HMODE NUMBER, 13 / 6X, 11HFREQUENCY =, F7.4, 0 )
LINE 130   1 22X, 13HMACH NUMBER =, F7.4 / )
LINE 131 2003 FORMAT(5X, 31HSECTION SPAN SPAN WIDTH, 0 )
LINE 132   1 8X, 27H--- LIFT PER UNIT SPAN ----, 0
LINE 133 2004 FORMAT(1X, 7X, 38HPITCHING MOMENT PER UNIT SPAN ABOUT X=, E12.6 / )
LINE 134 2005 FORMAT(1X, 6X, 59HVELOCITY POTENTIALS AND ELEMENT PRESSURES FOR WING, 0 )
LINE 135 2006 FORMAT(12 )
LINE 136 2007 FORMAT(36X,3H ELEMENT, 3X, 8(1H-1, 10H PRESSURE, 9(1H-1 ) / )
LINE 137 2008 FORMAT(3I7, 3X, E12.5, 3X, E12.6 )
LINE 138 2009 FORMAT(36X, 27HSECTION SPAN ELEMENT, 11X, 0 )
LINE 139 2010 FORMAT(12 )
LINE 140 2011 FORMAT(36X, 27HVELOCITY POTENTIAL ----, 11X, 8(1H-1, 10H PRESSURE, 0 )
LINE 141 2012 FORMAT(12 )
LINE 142 2013 FORMAT(36X, 27H--- VELOCITY POTENTIAL ----, 11X, 8(1H-1, 10H PRESSURE, 0 )
LINE 143 2014 FORMAT(12 )
LINE 144 2015 FORMAT(// )
LINE 145 2016 FORMAT(1H1, 45X, 1444 // 46X, 11HFREQUENCY =, F7.4/1X, 65(2H EI )
LINE 146 2017 FORMAT(1X, 81H02101450
LINE 147 2018 FORMAT(1X, 12101460

-6.56-
SUBROUTINE OLTP2(I6,NWING,XP,TLM,JM0DE,FREO,MACH,LINE,LMAX)

COMPLEX TL(4,10), TM(4,10)

REAL MACH

TL TOTAL LIFT PER WING

TM TOTAL MOMENT PER WING

WRITE TABLE OF TOTAL LIFT AND PITCHING MOMENTS

IF( LMAX-LINE.GT.14 ) GO TO 30

LINE = 7

GO TO 35

WRITE(I6, 2021)

WRITE(I6, 2008)

WRITE(I6, 2002)

WRITE(I6, 2011)

XP

DD 50 I=1,JM0DE

WRITE(I6,2009) (I, J, TL(J,I), TM(J,I), J=1,NWING)

LINE = LINE & NWING*JM0DE

LINE = MOD( LINE, LMAX )

RETURN

2002 FORMAT(6X,11HFREQUENCY =, F7.4, 22X, 13HMACH NUMBER =, F7.4 /)

2008 FORMAT( 46X,31HTOTAL LIFT AND PITCHING MOMENTS )

2009 FORMAT((15X,4,6X, 1,4, 5X, 2(3X, E12.6), 8X, 2(3X, E12.6) ) )

2011 FORMAT( 15X, 4HM0DE, 6X, 4HWING, 8X, 10(1H),6H LIFT, 11(1H), 8X, 25HPITCHING MOMENT )

1UT X = ,E12.6 /)

2020 FORMAT( 1H1 )

2021 FORMAT( / / / / )

END
DISPLAY...FILE

LINE  1  SUBROUTINE OUTP3( I6, MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE )
LINE  2  COMPLEX Q(10,10)
LINE  3  COMPLEX ROW(3)
LINE  4  REAL MACH
LINE  5  DIMENSION NST(3)
LINE  6  DIMENSION TITLE(14)
LINE  7  C
LINE  8  C  WRITE TABLE PRINT OF GENERALIZED AERODYNAMIC COEFF. MATRIX
LINE  9  C
LINE 10  IF( LINE .NE. LMAX ) GO TO 25
LINE 11  WRITE( 16, 2015 ) TITLE, FREQ
LINE 12  LINE = 6
LINE 13  GO TO 30
LINE 14  C
LINE 15  25 CONTINUE
LINE 16  IF( LMAX - LINE .GE. 13 & JMODE ) GO TO 30
LINE 17  WRITE( 16, 2020 )
LINE 18  LINE = 8
LINE 19  GO TO 35
LINE 20  30 CONTINUE
LINE 21  LINE = LINE + 13
LINE 22  GO TO 35
LINE 23  35 CONTINUE
LINE 24  WRITE( 16, 2010 ) FREQ, JMODE, MACH
LINE 25  C
LINE 26  NCL = 3
LINE 27  DO 100 J=1, JMODE, 3
LINE 28  KK = JMODE - J + 1
LINE 29  IF( KK .LT. 3 ) NCL = KK
LINE 30  DO 50 K=1, NCL
LINE 31  KO = K + J - 1
LINE 32  50 NST(K) = KO
LINE 33  IF( J .LT. 1 ) GO TO 65
LINE 34  IF( LMAX - LINE .GE. 56 & JMODE ) GO TO 60
LINE 35  WRITE( 16, 2020 )
LINE 36  LINE = 6
LINE 37  GO TO 65
LINE 38  60 CONTINUE
LINE 39  WRITE( 16, 2022 )
LINE 40  LINE = LINE + 5
LINE 41  65 CONTINUE
LINE 42  WRITE( 16, 2011 ) NST(1)
LINE 43  IF( QCL .GE. 2 ) WRITE( 16, 2023 ) NST(2)
LINE 44  IF( QCL .GE. 3 ) WRITE( 16, 2024 ) NST(3)
LINE 45  WRITE( 16, 2022 )
LINE 46  DO 50 K=1, JMODE
LINE 47  DO 90 R=K+1, NCL
LINE 48  90 KO = K + J - 1
LINE 49  80 ROW(K) = Q(1,K0)
LINE 50  GO TO 90
LINE 51  WRITE( 16, 2012 ) I, ( ROW(K), K=1, NCL )
LINE 52  LINE = LINE + JMODE
LINE 53  100 CONTINUE
LINE 54  RETURN
LINE 55  2010 FORMAT( 10X, 11HFREQUENCY =, F7.4, 18X,
LINE 56  1 56HG GENERALIZED AERODYNAMIC COEFFICIENTS IN AGARD DEFINITION /
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 ---- P RESSURE MODE, 13, 5H ---- )
LINE 60  2012 FORMAT( 15X, 13, 3( 8X, E12.6, 2X, E12.6 ) )
LINE 61  2015 FORMAT( 1H1, 45X, 14A4 // 46X, 11H FREQUENCY =, F7.4//1X, 5( 1H 4 )
LINE 62  2020 FORMAT( 1H1 )
LINE 63  2021 FORMAT( // // )
LINE 64  2022 FORMAT( )
LINE 65  2023 FORMAT( 1H6, 63X, 18H ---- P RESSURE MODE, 13, 5H ---- )
LINE 66  2024 FORMAT( 1H6, 57X, 18H ---- P RESSURE MODE, 13, 5H ---- )
LINE 67  END

-6.59-
LINE 1 SUBROUTINE PHI(XLE, XCN, DPDX, NEPS, PHI, A, CXP, PHIW)
LINE 2 COMPLEX B1, BO, AA, XI, XIP1, EX
LINE 3 COMPLEX CI, CX, XDIFS
LINE 4 COMPLEX EAXI, EAXIP1, AXI, AXIP1, SUM
LINE 5 COMPLEX PHI(1), DPDX(1), A, CXP(1)
LINE 6 COMPLEX PHIW(1)
LINE 7 LOGICAL WAKEI, WAKE, WAKENZ, TREDGE
LINE 8 DIMENSION XCN(1)
LINE 9 COMMON /WAKE/ ITW, ITW, ICNVGW, EPSW, JCNVGW
LINE 10 COMMON /WAKEUP/ ITWES (12), TREDGE, WAKE, WAKE, WAKENZ
LINE 11 DATA C1/(1.0D0, 0.0D0)
LINE 12 C
LINE 13 IF (EPS.EQ.1) GO TO 40
LINE 14 C
LINE 15 XIP1 = CMPLX(XCN(2), 0.0D0)
LINE 16 CX = CMPLX(XCN(1) - XCN(2), 0.0D0)
LINE 17 B1 = (DPDX(1) - DPDX(2)) / CX
LINE 18 BO = DPDX(2) - B1*XIP1
LINE 19 IF (CABS(A).LT.1.0D-20) GO TO 23
LINE 20 C
LINE 21 XI = CMPLX(XLE, 0.0D0)
LINE 22 XIP1 = CMPLX(XCN(1), 0.0D0)
LINE 23 CX = CMPLX(XCN(1) - XCN(2), 0.0D0)
LINE 24 AA = A * A
LINE 25 AXI = A*XI
LINE 26 AXIP1 = A*XIPI
LINE 27 EAXI = CXP (AXI)
LINE 28 EAXIP1 = CXP (AXIP1)
LINE 29 EX = C1 / EAXIP1
LINE 30 C
LINE 31 SUM = (B1/AA) * (AXIP1-C1)*EAXIP1 - (AXI-C1)*EAXI
LINE 32 1 E = (B0/AA) * (EAXIP1 - EAXI)
LINE 33 C
LINE 34 PHI(1) = SUM*EX
LINE 35 C
LINE 36 C
LINE 37 XI = XIP1
LINE 38 BO 10 IP1=2,NEPS
LINE 39 I = IP1 - 1
LINE 40 XIP1 = CMPLX(XCN(IP1), 0.0D0)
LINE 41 CX = CMPLX(XCN(1) - XCN(IP1), 0.0D0)
LINE 42 B1 = (DPDX(1) - DPDX(IP1)) / CX
LINE 43 BO = DPDX(IP1) - B1*XIP1
LINE 44 C
LINE 45 XI = XIP1
LINE 46 AXI = AXIP1
LINE 47 EX = C1 / EAXIP1
LINE 48 AXIP1 = AXIPI
LINE 49 EAXIP1 = CXP/IP1)
LINE 50 EX = C1 / EAXIP1
LINE 51 C
LINE 52 SUM = SUM E(B1/AA)* (AXIP1-C1)*EAXIP1 - (AXI-C1)*EAXI
LINE 53 1 E = (B0/AA) * (EAXIP1 - EAXI)
LINE 54 C
LINE 55 PHI(IP1) = SUM*EX
LINE 56 C
LINE 57  
10 CONTINUE
LINE 58  C
LINE 59  GO TO 52
LINE 60  20 CONTINUE
LINE 61  CX = CMPLX( XLE-XCEN(1), 0, 0 )
LINE 62  XDIFS = CMPLX( 0.5*( XLE-XLE-XCEN(1)*XCEN(1), 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 = CMPLX( XCEN(I)-XCEN(IP1), 0, 0 )
LINE 68  B1 = ( DPDX(IP1) - DPDX(IP1) ) / CX
LINE 69  B0 = DPDX(IP1) - B1*XI
LINE 70  XDIFS = CMPLX( 0.5*( XCEN(I)*XCEN(I)-XCEN(IP1)*XCEN(IP1), 0.0 )
LINE 71  SUM = SUM & 80*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  SECTION FOR ONLY ONE ELEMENT PER SPAN
LINE 79  C
LINE 80  CX = CMPLX( XCEN(1)-XLE, 0, 0 )
LINE 81  IF( CABS(1) .LT. 1.E-20 ) GO TO 50
LINE 82  PHI(1) = ( DPDX(1)/A )*( C1 - CE XP(1-A*CX ) )
LINE 83  GO TO 52
LINE 84  50 PHI(1) = DPDX(1)*CX
LINE 85  52 ICNVGW = 0
LINE 86  52 ICNVGW = 0
LINE 87  C ICNVGW = 0 MEANS AT LEAST ONE PHI HAS NOT CONVERGED FOR SPAN
LINE 88  C ICNVGW = 0 MEANS CONVERGENCE HAS OCCURRED FOR WAKE ELEMENTS
LINE 89  C ITWM = WAKE CONVG. ITERATION COUNTER INITIALLY SET = 0
LINE 90  C ITWM = MAX. NO. ITERATIONS IF WAKE EFFECT IS DESIRED
LINE 91  C ITWM = 0 IF WAKE EFFECT IS NOT DESIRED
LINE 92  IF( ITWM .EQ. 0 ) GO TO 58
LINE 93  IF( ,NOT. WAKE ) GO TO 58
LINE 94  ICNVGW = IRELE( PHI, PHIW, NEPS)
LINE 95  IF( ICNVGW .EQ. 0 ) JCNVGW = JCNVGW & NEPS
LINE 96  58 CONTINUE
LINE 97  60 CONTINUE
LINE 98  C
LINE 99  C
LINE 100  C ICNVGW = IRELE( PHI(IP1), PHIW(IP1), ICNVGW )
LINE 101  C IF( ICNVGW .EQ. 0 ) JCNVGW = JCNVGW & 1
LINE 102  60 PHIW(IP1) = PHI(IP1)
LINE 103  62 RETURN
LINE 104  END
<table>
<thead>
<tr>
<th>LINE</th>
<th>Code</th>
<th>Description</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUBROUTINE PLOT(X,Y,IPEN)</td>
<td></td>
<td>02500010</td>
</tr>
<tr>
<td>2</td>
<td>IF( IPEN .EQ. -23) GO TO 100</td>
<td></td>
<td>02500020</td>
</tr>
<tr>
<td>3</td>
<td>IF( IPEN .EQ. -23) IPEN = -3</td>
<td></td>
<td>02500030</td>
</tr>
<tr>
<td>4</td>
<td>CALL CALPLT(X,Y,IPEN)</td>
<td></td>
<td>02500040</td>
</tr>
<tr>
<td>5</td>
<td>RETURN</td>
<td></td>
<td>02500050</td>
</tr>
<tr>
<td>6</td>
<td>100 SX = X &amp; 2.0</td>
<td></td>
<td>02500060</td>
</tr>
<tr>
<td>7</td>
<td>CALL INFRAME(SX,Y)</td>
<td></td>
<td>02500070</td>
</tr>
<tr>
<td>8</td>
<td>RETURN</td>
<td></td>
<td>02500080</td>
</tr>
<tr>
<td>9</td>
<td>END</td>
<td></td>
<td>02500090</td>
</tr>
</tbody>
</table>
SUBROUTINE PLOTGD ( NE, XYZ, NNCH, NSP, NSECT, TITLE, JWING, MACH )  

REAL MACH  

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

DIMENSION PH(7), PV(7)  

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

DATA FGDX, FGY, /
  15.0, .05202, .035  /  

DATA HT, HTW, HHT, /
  .07, .06202, .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(K)  

XMIN = XMAX  

YMAX = XYZ(K+4)  

YMIN = YMAX  

DO 10 I=1,NEPS  

10 J = 1,4  

XMIN = AMIN1(XMIN, XYZ(K))  

XMAX = AMAX1(XMAX, XYZ(K))  

K = K + 1  

DO 6 J=1,4  

YMIN = AMIN1(YMIN, XYZ(K))  

YMAX = AMAX1(YMAX, XYZ(K))  

K = K + 1  

CONTINUE  

FGDX = MAXIMUM X PLOT SIZE  

FGDY = MAXIMUM Y PLOT SIZE  

SFH = (XMAX-XMIN)/FGDX  

SFV = (YMAX-YMIN)/FGDY  

SF IS A SCALE FACTOR SUCH THAT 1 INCH = SF  

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 REDRIGIN  

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 Y = 0.25 02600600
LINE 61 DO 12 I=1,14 02600610
LINE 62 CALL SYMBOL (-1.5, Y, .21, TITLE(I), 90.0, 4) 02600620
LINE 63 12 Y = Y & 0.728 02600630
LINE 64 C  02600640
LINE 65 CALL SYMBOL (-1.15, .25, .21, 14HASPECT RATIO =, 90.0, 14) 02600650
LINE 66 CALL NUMBER (-1.15, 3.98, .21, AR, 90.0, 3) 02600660
LINE 67 CALL SYMBOL (-1.15, 4.7, .21, 13MACH NUMBER =, 90.0, 13) 02600670
LINE 68 CALL NUMBER (-1.15, 7.10, .21, MACH, 90.0, 3) 02600680
LINE 69 CALL SYMBOL (-0.80, .25, .21, 14SECTION NUMBER, 90.0, 14) 02600690
LINE 70 FPN = FLOAT(IS) 02600700
LINE 71 CALL NUMBER ( -0.80, 2.98, .21, FPN, 90.0, -1) 02600710
LINE 72 CALL SYMBOL ( -0.80, 3.52, .21, 7HOF WING, 90.0, 7) 02600720
LINE 73 FPN = FLOAT(JWING) 02600730
LINE 74 CALL NUMBER ( -0.80, 4.97, .21, FPN, 90.0, -1) 02600740
LINE 75 CALL SYMBOL ( -0.80, 6.33, .21, 3HZ =, 90.0, 3) 02600750
LINE 76 FPN = ZSECT(IS) 02600760
LINE 77 CALL NUMBER ( -0.80, 7.06, .21, FPN, 90.0, 3) 02600770
LINE 78 C  02600780
LINE 79 C  02600790
LINE 80 C  02600800
LINE 81 K = KINDEX 02600810
LINE 82 DD 20 I=1,NEPS 02600820
LINE 83 DD 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 19 K = K + 1 02600870
LINE 88 PV(3) = XYZ(K+1) 02600880
LINE 89 PV(4) = XYZ(K) 02600890
LINE 90 PH(5) = PH(1) 02600900
LINE 91 PV(5) = PV(1) 02600910
LINE 92 WRITE (6,1003) I, (PH(I),J=1,5), (PV(I),J=1,5) 02600920
LINE 93 CALL LINE( PH, PV, 5, 1, 0, 0, 0,0) 02600930
LINE 94 20 CONTINUE 02600940
LINE 95 C  02600950
LINE 96 C  02600960
LINE 97 C  02600970
LINE 98 C  02600980
LINE 99 KLAST = NSP(IS) & KLAST 02600990
LINE 100 DD 30 I=KFIRST*KLAST 02600000
LINE 101 C  02600100
LINE 102 NETSV IS THE NUMBER OF ELEMENTS IN THE PREVIOUS WINGS PLUS 1 02601010
LINE 103 J = 6*(NNCH(I)-NETSV) & 1 02601020
LINE 104 SFH = 0.25*( XYZ(J) & XYZ(J+1) & XYZ(J+2) & XYZ(J+3) ) 02601030
LINE 105 SFV = 0.50*( XYZ(J+4) & XYZ(J+5) ) 02601040
LINE 106 SFH = (SFH-XMIN)/SF & HHT 02601050
LINE 107 SFV = (SFV-YMIN)/SF - HTW 02601060
LINE 108 C  (SFH, SFV) ARE COORDINATES OF CENTER OF ELEMENT 02601070
LINE 109 GP = FLOAT(NNCH(I)) 02601080
LINE 110 CALL CENTER CHARACTER, MOVE OFF CENTER BY (HTW,HHT) 02601090
LINE 111 CALL NUMBER( SFH, SFV, HT, GP, 90.0, -1) 02601100
LINE 112 30 CONTINUE 02601100
LINE 113 KFIRST = KFIRST & NSP(IS) 02601110
LINE 114 C  02601120
LINE 115 C  02601130
LINE 116 C  02601140
LINE 117 C  02601150
LINE 118 CALL PLOT( SX, -.25, -.23) 02601160

-6.64-
LINE 119 C UPDATE COUNTER OF ELEMENTS IN WING
LINE 120 C NEPWT = NEPWT & NEPS
LINE 121 NEPWT = NEPWT & NEPS
LINE 122 KINDEX = KINDEX & 6*NEPS
LINE 123 C END LOOP ON SECTIONS
LINE 124 C CONTINUE
LINE 125 C CONTINUE
LINE 126 C UPDATE ELEMENT COUNTER
LINE 127 C NETSV IS THE NUMBER OF ELEMENTS IN THE PREVIOUS WINGS PLUS 1
LINE 128 NETSV = NETSV & NEPWT
LINE 129 C RETURN
LINE 130 C RETURN
LINE 131 C1002 FORMAT( // 'XMAX,XMIN', 2E14.4, 5X, 'YMAX,YMIN', 2E14.4 //
LINE 132 C 2 'SFH,SPV', 2E14.4 //
LINE 133 C 3 'TMXX,TMYY', 2E14.4, 5X, 'SF ', 1E14.4 //
LINE 134 C 4 'PH FOLLOWED BY PV FOR EVERY ELEMENT * /
LINE 135 C1003 FORMAT( // 15, 21/ 5X, 5E14.4 ) )
LINE 136 C END
LINE 1     SUBROUTINE PTGRID ( I6, TITLE, MACH, NING, NSECT, NSP, NNCH, XYZ) U2700010
LINE 2     1       ZSECT, REFLEN, BEETA, XCEN ) U2700020
LINE 3     REAL MACH U2700030
LINE 4     DIMENSION NNCH(1), NSP(1) U2700040
LINE 5     DIMENSION TITLE(14), NSECT(1), XYZ(11), ZSECT(3,4) U2700050
LINE 6     DIMENSION XCEN(11) U2700060
LINE 7     DIMENSION XY(8) U2700070
LINE 8     BL = BEETA*REFLEN U2700080
LINE 9     WRITE( 16,300 ) TITLE, MACH U2700090
LINE 10    WRITE( 16,300 ) ZSECT, REFLEN, BEETA, XCEN ) U2700100
LINE 11    KSPAN = 0 U2700110
LINE 12    K = 0 U2700120
LINE 13    N1 = 1 U2700130
LINE 14    DO 100 J=1,NING U2700140
LINE 15    NS = NSECT(J) U2700150
LINE 16    DO 95 J=1,NS U2700160
LINE 17    WRITE( 16,300 ) 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    NEPS = NNCH(KSPAN) - K U2700210
LINE 22    DO 85 IE = 1,NEPS U2700220
LINE 23    K = K & 1 U2700230
LINE 24    XY(7) = 0 U2700240
LINE 25    IN = N1 U2700250
LINE 26    DO 80 JJ = 1,4 U2700260
LINE 27    XY(7) = XY(7) + BL U2700270
LINE 28    XY(7) = XY(7) / BL U2700280
LINE 29    IN = IN + 1 U2700290
LINE 30    80 CONTINUE U2700300
LINE 31    XY(7) = XY(7) / 4.0 U2700310
LINE 32    XY(5) = XY(7) + BL U2700320
LINE 33    XY(6) = XY(7) + BL U2700330
LINE 34    XY(8) = 0.5*( XY(5) + XY(6) ) U2700340
LINE 35    WRITE( 16,300 ) ISPAN, K, XY U2700350
LINE 36    XCEN(K) = XY(7) / BL U2700360
LINE 37    N1 = N1 & 6 U2700370
LINE 38    85 CONTINUE U2700380
LINE 39    90 CONTINUE U2700390
LINE 40    95 CONTINUE U2700400
LINE 41    100 CONTINUE U2700410
LINE 42    RETURN U2700420
LINE 43    3000 FORMAT(1H1/35X,14A4/35X, 32H AERODYNAMIC GRID FOR MACH NMYBER, U2700430
LINE 44    1 F3.4 / ) U2700440
LINE 45    3001 FORMAT( / 6H WING, 12, 8H SECTION, 12 / 5H L = 2X, E11.5/ U2700450
LINE 46    1 14H SPAN ELEMENT, 4X, 2HX1, 12X, 2HX2, 12X, 2H3, 12X, 2HX4, 4X, 14H U2700460
LINE 47    2 12X, 2HY1, 12X, 2HY2, 13X, 2HC, 12X, 2HYC / ) U2700470
LINE 48    3002 FORMAT( I5, 19, 1X, 61 3X, E11.5, 3X, 11.5 ) U2700480
LINE 49    3003 FORMAT( / 35X, 32H STRUCTURAL REFERENCE SYSTEM / ) U2700490
LINE 50    ITEM J4 / 35X, 35H TATED STRUCTURAL REFERENCE SYSTEM / ) U2700500
LINE 51    END U2700510

-6.66-
### Display-File Deck 28

<table>
<thead>
<tr>
<th>LINE</th>
<th>Description</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUBROUTINE ROETA (I13, DETAX, ETA, NET)</td>
<td>02800010</td>
</tr>
<tr>
<td>2</td>
<td>DIMENSION DETAX(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>JMI, NET1, DETAX, ETA</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>
SUBROUTINE RMODE
C MODE, N4ING, NSECT, 
C, XEN, XYZ, BEETA, 
C
LINE 2 1 REFLEN, NET, LAY, ID, ERR0R, ETA, DETADX, 15, 16, XY, YF, A, N4X, I4ERR
C
LINE 3 LOGICAL ERROR
C
LINE 4 C ETA AND DETADX DIMENSIONED (NUMBER OF ELEMENTS, NO. MODES)
C
LINE 5 DIMENSION CFTABL(6, 20), IMPATB(2, 20)
LINE 6 DIMENSION ETA(1), DETADING(1)
LINE 7 DIMENSION ID(1)
LINE 8 DIMENSION MODE(20)
LINE 9 DIMENSION MODEF4(6)
LINE 10 DIMENSION NCHC(4, 3)
LINE 11 DIMENSION NSECT(11), NE(3), XEN(1), XYZ(1)
LINE 12 DIMENSION NUMR(3), MODEG(7), MODE2(13), MODE3(7)
LINE 13 DIMENSION XP(1), YP(1), A(1)
LINE 14 COMMON MCOK, C/CF, CFC, /, ITAB, NTAB, NETQ
LINE 15 COMMON / ORDER / EM(10), DM(10), X, Y, J1, J2
LINE 16 COMMON /XEVSCAL / XX, YY, XX, BEET A, ZIZ, TEST
LINE 17 COMMON /Z FunM/ N1, ERF, B1, B2, B3, IFUNMX
LINE 18 DATA LBLANK / 44H /
LINE 19 DATA LEND / 44H /
LINE 20 DATA LOM / 44HOM /
LINE 21 DATA LRIG / 44HRIGM /
LINE 22 DATA MODONE / 23 /
LINE 23 DATA MODE1 / 32H, 1H1, 4H (24X, 4H, 4H, 4H, 4H, 4H, 4H, 4H, 4H, 3H) /
LINE 24 DATA MODE2 / 4H (60H, 4H, 5H, 4H, 4HME, 4H, 4H, 4H, 4H, 4H)
LINE 25 DATA MODE3 / 4H (11, 2H1, 1H1, 4H (4X, 4H, 4H, 4H, 4H, 4H, 2H1)
LINE 26 DATA MODE4 / 4H (6, 2H1, 1H1, 4H, 6H, 4H, 4H, 4H, 4H, 2H1)
LINE 27 DATA MODE5 / 4H (4, 2H1, 1H1, 4H, 6H, 4H, 4H, 4H, 2H1)
LINE 28 DATA NUMBR / 160 /
LINE 29 DATA NUMR / 1H1, 1H2, 1H3 /
LINE 30 C
LINE 31 XG = 0.0
LINE 32 YG = 0.0
LINE 33 BREF = 1.0
LINE 34 JMODE = 0
LINE 35 WRITE(16, 1999) ID(1), ID(2), ID(3), ID(4)
LINE 36 IF( ID(1) .EQ. 0 ) RETURN
LINE 37 1999 FORMAT (51X, 51H TOTAL NUMBER MODES, 1H, 12X, 13H, 6HMOD),
LINE 38 9H TOT N NUMBR OF POLYNOMIAL, 1X, 6HMOD)
LINE 39 2 
LINE 40 13H RIGID MODES, 17X, 11Z / 21X, 21HNUMBER OF POLYNOMIAL
LINE 41 31H EQUATIONS, 8X, 112 //
LINE 42 JRGID = ID(1)
LINE 43 JTYPF = ID(4)
LINE 44 RMODE = 0
LINE 45 IF( LAB .EQ. 1 .OR. LRIG ) GO TO 215
LINE 46 JMODE = ID(2)
LINE 47 IF(1 = 1 .OR. LREFLEN ) RETURN
LINE 48 MODEF4(6) = NUMBR(2)
LINE 49 IF( JMODE .EQ. 1 ) MODEF4(6) = MODONE
LINE 50 C
LINE 51 IF( JTYPE .EQ. 2 ) GO TO 10
LINE 52 WRITE(16, 2001)
LINE 53 WRITE(16, 2002)
LINE 54 WRITE(16, 2003)
LINE 55 WRITE(16, 2004)
LINE 56 DO 500 I = 1, NEQF
LINE 57 DO 500 I = 1, NEQF
C
-6.68-
ORIGINAL PAGE IS
OF POOR QUALITY

```
LINE 57  READ (15, 405) IEQ, IFTABL(1, IEQ), IFTABL(2, IEQ)
LINE 58  1 (CFTABL(J, IEQ), J=1, 6)
LINE 59  WRITE(16, 410) IEQ, (IFTABL; IEQ), J=1,2, (CFTABL(J, IEQ),
LINE 60  1 J =1,6 )
LINE 61  500 CONTINUE
LINE 62  C  LOOP OVER ELEMENTS BY WITHIN SECTION
LINE 63  J = 5
LINE 64  I = 1
LINE 65  DO 630 J1=1, NWING
LINE 66  NS = NSECT(J1)
LINE 67  DO 695 J2=1, NS
LINE 68  IN1 = NE(J2, J1)
LINE 69  DO 700 K=1, IN1
LINE 70  X = XCEN(I)*BR
LINE 71  Y = (XYZ(J) + XYZ(J2) + HALFR
LINE 72  CALL MFUN(JMODE)
LINE 73  DO 390 IH=I,JMODE
LINE 74  ETA((,IH) = EM(IHI
LINE 75  OETADX(I, 1H) = DM(IHI
LINE 76  380 CONTINUE
LINE 77  I = I + 1
LINE 78  J = J + 6
LINE 79  390 CONTINUE
LINE 80  395 CONTINUE
LINE 81  600 CONTINUE
LINE 82  IN1 = 1
LINE 83  IN2 = NET
LINE 84  GO TO 132
LINE 85  10 CONTINUE
LINE 86  C  READ MOWING CARD
LINE 87  C
LINE 88  READ (15, 1000) LAB, NW, NIL, NI2, NI3
LINE 89  C
LINE 90  IF(LAB .EQ. LEND ) GO TO 330
LINE 91  IF(LAB .EQ. LMK ) GO TO 300
LINE 92  C
LINE 93  C  CHECK TO SEE IF SECTION IS DEFINED
LINE 94  IF( NW .GT. NWING ) GO TO 310
LINE 95  C
LINE 96  HOW MANY SECTIONS ARE DEFINED
LINE 97  C
LINE 98  NSI = 3
LINE 99  IF(NI3 .EQ. 0 ) NSI = 2
LINE 100  IF(NI2 .EQ. 0 ) NSI = 1
LINE 101  IF(NI1 .EQ. 0 ) GO TO 310
LINE 102  C  IF 2 SECTIONS ARE INPUT, THEY MUST BE CONTINUOUS
LINE 103  C
LINE 104  IF( NSI .EQ. 2 ) AND NABS(NI2-NI1).NE. 1 ) GO TO 320
LINE 105  C
LINE 106  C  FIND FIRST AND LAST SECTION FOR THIS MODE GROUP
LINE 107  C
LINE 108  NSF = NI1
LINE 109  NFL = NI1
LINE 110  WRITE( 16, 2001 )
LINE 111  WRITE( 16, 2302 ) NW, NIL
LINE 112  IF(VSI .EQ. 1 ) GO TO 20
LINE 113  NSF = NI1( NSF, NI2 )
LINE 114  NSL = MAX0( NSF, NIL )
LINE 115  WRITE( 16, 2002 ) NW, NI2
LINE 116  IF(VSI .EQ. 2 ) GO TO 20
LINE 117  NSF = MIN0( NSF, NI3 )
LINE 118  NSL = MAX0( NSF, NI3 )
```
LINE 119 WRITE( 16, 2002 ) NW, N13
LINE 120 20 CONTINUE
LINE 121 C
LINE 122 C MORE CHECK TO SEE IF SECTION IS DEFINED
LINE 123 IIF( 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 IIF( 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 IIF( 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 IIF( NSI .EQ. 1 ) GO TO 45
LINE 149 C
LINE 150 IN2 = IN2 + NE(NSF1,NW)
LINE 151 NCHECK(NSF1,NW) = 1
LINE 152 IIF( 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 IN1 AND IN2 HAVE BEEN FOUND
LINE 157 C
LINE 158 C WRITE( 16, MODEF4 ) JMODE
LINE 159 WRITE( 16, 2003 )
LINE 160 WRITE( 16, 2008 )
LINE 161 C
LINE 162 C SECTION TO READ MODE DATA
LINE 163 C------------------------------------------
LINE 164 C
LINE 165 C READ MODE INPUT
LINE 166 C MODE CARD HAS BEEN READ
LINE 167 C
LINE 168 C FUNMX = NMX
LINE 169 CALL READXY( XF, YF)
LINE 170 IERR = IERF
LINE 171 IIF( IERR .NE. 0) RETURN
LINE 172 C
LINE 173 C DO 130 J=1,JMODE
LINE 174 CALL READAB(A, A)
LINE 175 READ( 15, 1006 ) MODE
LINE 176 WRITE( 16, 1016 ) MODE
LINE 177 CALL READAB(A, A)
LINE 178 IIF( TEST .EQ. 0 ) GO TO 110
LINE 179 DO 135 I = IN1, IN2
LINE 180 ETA(I,J) = ZIZ

-6.70-
LINE 181  DETAOD(X(I,J)) = 0.0
LINE 182  105 CONTINUE
LINE 183  GO TO 130
LINE 184  110 CONTINUE
LINE 185  C BUMP INDEX 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 STRUCTURAL REFERENCE SYSTEM
LINE 189  C
LINE 190  XX = XRXCEN(I)
LINE 191  YY = HALFR*(XY1(INDXR)) & XYZ(INDXR))
LINE 192  INDXR = INDXR & 6
LINE 193  C CALL ZFDJX ETA(I,J), DETAODX(I,J), XI, XF, YF, A
LINE 194  120 CONTINUE
LINE 195  C
LINE 196  120 CONTINUE
LINE 197  C
LINE 198  130 CONTINUE
LINE 199  C
LINE 200  132 CONTINUE
LINE 201  C
LINE 202  IPR = 1
LINE 203  WRITE( I6, 204 ) IPR
LINE 204  IFJTYPE .NE.0) GO TO 135
LINE 205  WRITE( 16, 206 ) NW, N11
LINE 206  IF( NS1 .GE. 2 ) WRITE( 16, 207 ) NW, N12
LINE 207  IF( NS1 .EQ. 3 ) WRITE( 16, 208 ) NW, N13
LINE 208  135 CONTINUE
LINE 209  WRITE( 16, 209 ) NW, N12
LINE 210  WRITE( 16, 211 ) NW, N13
LINE 211  J1 = 1
LINE 212  IHS = JMODE/3
LINE 213  IF( 3*IHS .NE. JMODE ) IHS = IHS & 1
LINE 214  C
LINE 215  DO 138 I1 = 1, IHS
LINE 216  J2 = J1 & 2
LINE 217  IF( J2 .GT. JMODE ) J2 = JMODE
LINE 218  INUMB = J2 - J1 & 1
LINE 219  MODEF1(2) = NUMB(INUMB)
LINE 220  MODEF2(5) = NUMB(INUMB)
LINE 221  MODEF3(3) = NUMB(INUMB)
LINE 222  IF( N1 .GE. 2 ) WRITE( 16, 223 ) J1, I = J1, J2
LINE 223  WRITE(I, 16, MODEF1) ( I, I = J1, J2)
LINE 224  WRITE(I, 16, MODEF2) ( I, I = J1, J2)
LINE 225  WRITE( 16, MODEF3) ( I, (ETA(I,J), DETAODX(I,J), J = J1, J2), I = IN1, IN2)
LINE 226  J1 = J1 & 3
LINE 227  138 CONTINUE
LINE 228  C
LINE 229  IN1 = IN2 & 1
LINE 230  C
LINE 231  140 CONTINUE
LINE 232  IFJTYPE .NE.0 GO TO 200
LINE 233  C
LINE 234  C CHECK TO SEE IF ALL SECTIONS HAVE BEEN READ AND ACCOUNTED FOR
LINE 235  C
LINE 236  DO 150 J = 1, NKS1
LINE 237  NS = NSEC(T(J)
LINE 238  DO 150 I = 1, NS
LINE 239  C IF NCHECK(I,J) .EQ. 0 ) GO TO 10
LINE 240  150 CONTINUE
LINE 241  C
LINE 242  C

-6.71-
DISPLAY...FILE

LINE  1  SUBROUTINE READAB(Z, A)
LINE  2  INTEGER HEADER(20)
LINE  3  DIMENSION Z(1),A(1)
LINE  4  COMMON /ZFDZIO/ I5, I6
LINE  5  COMMON /ZFUNNY/ N,IERF,B1,B2,B3,NFUNMK
LINE  6  COMMON /XYSCAL/ XX,YY,XG,YG,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

-6.74-
DISPLAY...FILE DECK 31

LINE 1 SUBROUTINE READXY(X,Y)
LINE 2 INTEGER HEADER(20)
LINE 3 DIMENSION X(I), Y(I)
LINE 4 COMMON /ZFDO2O/ I5, I6
LINE 5 COMMON /ZFUNNY/ N,IERF,B1,B2,B3,NFUNMX
LINE 6 C
LINE 7 C INITIALIZATION ENTRY POINT (ABSCISSAS).
LINE 8 C USAGE CALL READ NXY
LINE 9 READ(15,10) N
LINE 10 WRITE(16,10) N
LINE 11 READ(15,11) HEADER
LINE 12 WRITE(16,11) HEADER
LINE 13 IERF = 0
LINE 14 IF (Y.GT. NFUNMX) IERF=680
LINE 15 IF (Y.GT. NFUNMX) RETURN
LINE 16 READ(15,13) (X(I), I=1,N)
LINE 17 WRITE(16,12) (X(I), I=1,N)
LINE 18 READ(15,11) HEADER
LINE 19 WRITE(16,11) HEADER
LINE 20 READ(15,13) (Y(I), I=1,N)
LINE 21 WRITE(16,12) (Y(I), I=1,N)
LINE 22 RETURN
LINE 23 10 FORMAT(5X,3HN = ,13)
LINE 24 11 FORMAT(2CA4)
LINE 25 12 FORMAT(1PE15.7)
LINE 26 13 FORMAT(5E15.7)
LINE 27 END

-6.75-
SUBROUTINE RSTART ( MACH, NET, NREQ, FREQ, REFLEN, IWT, 18, 115, 03200010
LINE 2 SUBROUTINE RSTART ( Mach, Net, NREQ, FREQ, REFLEN, IWT, 18, 115, 03200020
LINE 3 LOGICAL ERROR
LINE 4 REAL MACH, MACHR
LINE 5 DIMENSION FREQ(11), FREQR(II), AREA(II), XYZ(II)
LINE 6 COMMON / NEXTCS / IFLUSH
LINE 7 ERROR = .FALSE.
LINE 8 IS = 10
LINE 9 10 CONTINUE
LINE 10 REWIND IS
LINE 11 READ(15) MACH, NETR, RRFL, IWTER, NFREQR, (FREQR(I), I=1, NFREQR)
LINE 12 IF( MACH NE. 'MACH ' GO TO 90
LINE 13 IF( NETR NE. NET ) GO TO 90
LINE 14 IF( NFREQR NE. NFREQ ) GO TC 90
LINE 15 IF( REFLEN NE. RRFL ) GO TO 90
LINE 16 IF( IWT .NE. 0 ) AND. IWTER .EQ. 3 ) GO TO 90
LINE 17 DO 15 I=1, NFREQ
LINE 18 IF( FREQR(I) .NE. FREQ(I) ) GO TO 90
LINE 19 15 CONTINUE
LINE 20 IF( IWTER .EQ. 0 ) GO TO 30
LINE 21 IF( IS .EQ. 115 ) GO TO 30
LINE 22 IS = 115
LINE 23 GO TO 10
LINE 24 30 CONTINUE
LINE 25 IF( ERROR ) GO TO 99
LINE 26 C RESTART TAPE(SI HAVE BEEN READ AND FOUND TO BE CORRECT
LINE 27 C FIND AREA
LINE 28 INDXR = 1
LINE 29 DO 50 IRC = 1, NET
LINE 30 DSX = 0.51 (XYZ(INDXR&51-XYZ(INDXR&41)
LINE 31 AREA(IRC) = DSX* (XYZ(INDXR&3)-XYZ(INUXRG1)1
LINE 32 INDXR = INDXR E 6
LINE 33 50 CONTINUE
LINE 34 RETURN
LINE 35 90 CONTINUE
LINE 36 C ERROR IN RESTART TAPE
LINE 37 ERROR = ,TRUE.
LINE 38 WRITE( 16, 2000 ) IS
LINE 39 WRITE ( 16, 2001 )
LINE 40 WRITE ( 16, 2002 ) MACH, MACHR, NET, NETR, REFLEN, RRFL, IWT,
LINE 41 1 IWTER, NFREQ, NFREQR, ( I, FREQ(I), FREQR(I),I=1, NFREQ)
LINE 42 IF( IWTER .EQ. 0 ) GO TO 99
LINE 43 IF( IS .EQ. 115 ) GO TO 99
LINE 44 IS = 115
LINE 45 GO TO 10
LINE 46 99 CONTINUE
LINE 47 IFLUSH = 1
LINE 48 RETURN
LINE 49 2000 FORMAT(1HI/ 60(2H //3BH ERROR IN READING RESTART TAPE ON UNIT
LINE 50 141 )
LINE 51 2001 FORMAT( //4BH RESTART TAPE DID NOT CORRESPOND TO CURRENT RUN.
LINE 52 67H FOLLOWING IS A COMPARISON OF THE CURRENT RUN AND THE KE RST
LINE 53 2TAPE. // 4IX, 11HCURRENT RUN, 9X, 124RESTART TAPE )
LINE 54 2002 FORMAT( / 11X, 24HMACH NUMBER , E17.5, E21.5
LINE 55 1/11X, 24HTOTAL NUMBER OF ELEMENTS, 117, 121
LINE 56 2/11X, 24HREFERENCE LENGTH , E17.5, E21.5

-6.76-
24M: WAKE EFFECT CODE
24: NUMBER OF FREQUENCIES
24: FREQUENCY(12, 141, 11X, E17.5, E21.5)
LINE 1 SUBROUTINE RTOI ( TR, TI, YCR, ZCR, ZPJ, ZLO )
LINE 2 C RTOI TRANSFORMS YCR AND ZCR FROM THE RECEIVING SYSTEM I.J THE INFLUENCING SYSTEM. YCR, ZCR ARE FIRST PUT IN REFERENC SYSTEM BY PREMULTIPLYING BY TR. THIS IS Y, Z WHICH IS PUT IN INFLUENCING SYSTEM BY THE TRANSFORM OF TI
LINE 3 C INFLUENCING SYSTEM YC, ZC BY PREMULT. BY THE TRANSFORM OF TI
LINE 4 C DIMENSION TR(11), TI(11)
LINE 5 COMMON / RLLL / TVW(4), YRO, YLO, ZR, ZZR, ZL, ZZZL
LINE 6 Y = TR(1)*YCR & TR(3)*ZCR
LINE 7 Z = TR(2)*YCR & TR(4)*ZCR
LINE 8 YR0 = TI(1)*Y & TI(2)*Z
LINE 9 ZR0 = TI(3)*Y & TI(4)*Z
LINE 10 YLO = -TI(1)*Y & TI(2)*Z
LINE 11 ZL0 = TI(3)*Y - TI(4)*Z
LINE 12 TVW(1) = TR(3)*TI(1) & TR(4)*TI(2)
LINE 13 TVW(2) = TR(3)*TI(3) & TR(4)*TI(4)
LINE 14 TVW(3) = -TR(3)*TI(1) & TR(4)*TI(2)
LINE 15 TVW(4) = TR(3)*TI(3) - TR(4)*TI(4)
LINE 16 RETURN
LINE 17 ENO

END OF JOB.
33.2 SEC. USED .010 HRS. CHARGED 49.952 HRS. REMAINING

-G.78-
LINE 1        SUBROUTINE SOLVE IN,NDIM,A,B,IP
LINE 2        C
LINE 3        SUBROUTINE SOLVE
LINE 4        C
LINE 5        PURPOSE
LINE 6        C  SOLUTION OF THE LINEAR SYSTEM OF EQUATIONS CX = B.
LINE 7        C
LINE 8        USAGE
LINE 9        C CALL SOLVE(IN,NDIM,A,B,IP)
LINE 10       C
LINE 11       C DESCRIPTION OF PARAMETERS
LINE 12       N - ORDER OF THE MATRIX A
LINE 13       NDIM - FIRST DIMENSION OF A DECLARED IN CALLING PROGRAM. IF A IS
LINE 14       SIMPLY SUBSCRIPTED IN CALLING PROGRAM, SET NDIM = N.
LINE 15       A - CONTAINS THE TRIANGULAR FACTORS OF THE MATRIX C AS
LINE 16       DETERMINED BY SUBROUTINE 'DECM'.
LINE 17       B - ON INPUT, THE RHS VECTOR, ON OUTPUT, THE SOLUTION VECTOR.
LINE 18       IP - VECTOR OF DIMENSION N CONTAINING ROW INTERCHANGE
LINE 19       INFORMATION (AS DETERMINED BY SUBROUTINE 'DECM').
LINE 20       REMARKS
LINE 21       'SOLVE' MUST BE USED IN CONJUNCTION WITH SUBROUTINE 'DECM'.
LINE 22       SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED.
LINE 23       NINE
LINE 24       M TMOD
LINE 25       BACK SUBSTITUTION BASED ON THE FACTORED FORM OF THE COEFFICIENT
LINE 26       MATRIX. SEE ALGORITHM 423, 'COLLECTED ALGORITHMS FROM CACM',
LINE 27       BY CLEVE MOLER.
LINE 28       DIMENSION A(NDIM,1),B(1),IP(1)
LINE 29       IF (N .EQ. 1) GO TO 9
LINE 30       NM1 = N-1
LINE 31       DO 7 K = 1,NM1
LINE 32       KP1 = K+1
LINE 33       M = IP(K)
LINE 34       T = B(M)
LINE 35       B(M) = B(K)
LINE 36       B(K) = T
LINE 37       DO 8 I = KP1,N
LINE 38       7 B(I) = B(I) + A(I,K)*T
LINE 39       DO 8 KB = 1,NM1
LINE 40       KM1 = N - KB
LINE 41       K = KM1 + 1
LINE 42       B(K) = B(K)/A(K,K)
LINE 43       T = -B(K)
LINE 44       DO 8 I = 1,KM1
LINE 45       8 B(I) = B(I) + A(I,K)*T
LINE 46       DO 8 KB = 1,NM1
LINE 47       9 B(I) = B(I)/A(I,I)
LINE 48       RETURN
LINE 49       END

-6.79-
SUBROUTINE SONS(NWIN, NSECT, CORNX, CORNY, CORNZ, ISONS)

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

DO 30 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
  END

DO 10 12 = 2, 4, 2
  IND1 = JCN1 & 12
  XC = CORNX(IND1, J)
  YC = CORNY(IND1, J)
  ZC = CORNZ(IND1, J)
  LOOP ON TOP CORNER PTS. OF INFLUENCING SECTION (PTS 1 AND 3)

IF( XC .LT. X1 ) AND. X1 .GT. X3 GO TO 10
  IF( YC .LT. Y1 ) AND. Y1 .GT. Y3 GO TO 8
  IF( ZC .LT. Z1 ) AND. Z1 .GT. Z3 GO TO 14

A = YCY1/(Y1-Y3)
X = X1 - A*(X3-X1)
Z = Z1 - A*(Z3-Z1)

YCY1 = YC - Y1
Y3YC = Y3 - YC

IF( Y1 .EQ. Y3 ) GO TO 8
IF( YCY1#Y3YC .LE. 0 ) GO TO 8

R = XC - X AND Z - ZC
LINE 57 C IF R GT 0, SECTION HAS INFLUENCE SO BRANCH OUT OF DO 10 LOOP
LINE 58 IF(R .GT. 0.0) GO TO 15
LINE 59 GO TO 10
LINE 60 CONTINUE
LINE 61 C YC IS OUTSIDE OF Y1,Y3 CHECK BOTH POINTS
LINE 62 C
LINE 63 ZHI = XC - X1
LINE 64 ETA = YC - Y1
LINE 65 ZO = ZC - Z1
LINE 66 T = SQRT( ETA*ETA + ZO*ZO )
LINE 67 R = ZHI - T
LINE 68 C IF R GT 0, SECTION HAS INFLUENCE SO BRANCH OUT OF DO 10 LOOP
LINE 69 IF(R .GT. 0.0) GO TO 15
LINE 70 C
LINE 71 ZHI = XC - X3
LINE 72 ETA = YC - Y3
LINE 73 ZO = ZC - Z3
LINE 74 T = SQRT( ETA*ETA + ZO*ZO )
LINE 75 R = ZHI - T
LINE 76 C IF R GT 0, SECTION HAS INFLUENCE SO BRANCH OUT OF DO 10 LOOP
LINE 77 IF(R .GT. 0.0) GO TO 15
LINE 78 C WE DID NOT BRANCH OUT OF DO 10 LOOP
LINE 79 10 CONTINUE
LINE 80 C
LINE 81 C SECTION K2 DOES NOT INFLUENCE SECTION K1
LINE 82 KONS(K1*K2) = 0
LINE 83 C
LINE 84 C END LOOP ON INFLUENCING SECTION (15) AND WING (20)
LINE 85 15 CONTINUE
LINE 86 CONTINUE
LINE 87 20 CONTINUE
LINE 88 C END LOOP ON RECEIVING SECTION (25) AND WING (30)
LINE 89 25 CONTINUE
LINE 90 30 CONTINUE
LINE 91 C
LINE 92 RETURN
LINE 93 END
DISPLAY...FILE

LINE 1 C SUBROUTINE SONSPT ( ISONS, NWING, NSECT, MACH, I6 )
LINE 2 C SONSPT PRINTS SECTION INFLUENCE TABLE
LINE 3 REAL MACH
LINE 4 DIMENSION IFT1(9), IFT2(9), IF(15)
LINE 5 DIMENSION ISONS(12,12), NSECT(1), IW(15), IS(15)
LINE 6 C
LINE 7 DATA IFT1 / 4H(11H, 4H REC, 4H ETV, 4H NG, 1H, 4H(6H,
LINE 8 1 4H WIN, 4H GL, 2HJ) /
LINE 9 DATA IF / 1H1, 1H2, 1H3, 1H4, 1H5, 1H6, 1H7, 1H8, 2H10, 2H11, 2H12, 2H13 /
LINE 10 1 2H14, 2H15 /
LINE 11 DATA IFT2 / 4H(11H, 4H SEC, 4H TION, 4H, 1H, 4H(6H,
LINE 12 1 4H SEC, 4H T, 2HJ) /
LINE 13 C
LINE 14 WRITE( 16, 1001 ) MACH
LINE 15 WRITE( 16, 1002 )
LINE 16 K = 0
LINE 17 DO 10 J=1,NWING
LINE 18 NS = NSECT(J)
LINE 19 DO 5 I=1,NS
LINE 20 K = K + 1
LINE 21 IW(K) = J
LINE 22 IS(K) = I
LINE 23 5 CONTINUE
LINE 24 10 CONTINUE
LINE 25 IFT1(5) = IF(K)
LINE 26 IFT2(5) = IF(K)
LINE 27 WRITE( 16, IFT1 ) (IW(I), J=1,K)
LINE 28 WRITE( 16, IFT2 ) (IS(I), J=1,K)
LINE 29 DO 20 I=1,K
LINE 30 WRITE( 16, 1003 ) (IW(I), IS(I), ISONS(I,J), J=1,K)
LINE 31 20 CONTINUE
LINE 32 1001 FORMAT( 1H1 / 40X, 45HTABLE OF INFLUENCING SECTIONS FOR MACH NUMBER 03600320
LINE 33 1, FB.4 ) 03600330
LINE 34 1002 FORMAT( 2X, 50HEKEY 0 ALL ELEMENTS OF INFLUENCING SECTION HAVE 03600340
LINE 35 1, 50HERO INFLUENCE ON ALL ELEMENTS OF RECEIVING SECTION 03600350
LINE 36 2 7X, 32H NON-ZERO INFLUENCE IS ASSUMED /// 03600360
LINE 37 3 13X, 20H INFLUENCING SECTIONS ) 03600370
LINE 38 1003 FORMAT( / 5H WING, 12 / 5H SEC, 12, 4X, 15I8 ) 03600380
LINE 39 RETURN
LINE 40 END

03600010
03600020
03600030
03600040
03600050
03600060
03600070
03600080
03600090
03600100
03600110
03600120
03600130
03600140
03600150
03600160
03600170
03600180
03600190
03600200
03600210
03600220
03600230
03600240
03600250
03600260
03600270
03600280
03600290
03600300
03600310
03600320
03600330
03600340
03600350
03600360
03600370
03600380
03600390
03600400
<table>
<thead>
<tr>
<th>LINE</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUBROUTINE SYMBOL(X,Y,SIZE,CHAR,ANG,NCHAR)</td>
</tr>
<tr>
<td>2</td>
<td>DIMENSION CHAR(3)</td>
</tr>
<tr>
<td>3</td>
<td>CALL NOTATE(X,Y,SIZE,CHAR,ANG,NCHAR)</td>
</tr>
<tr>
<td>4</td>
<td>RETURN</td>
</tr>
<tr>
<td>5</td>
<td>END</td>
</tr>
</tbody>
</table>
SUBROUTINE TIMEOUT(MSEC, A, B, C, D, E, F, G, H)
LINE 2 COMMON /TAPE/ 19, 110, 111, 112, 113, 114, 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, 844, 5X, F8.3, 13X, F8.3)
LINE 11 END
SUBROUTINE TMAXNET, A, GBAR
DIMENSION A(I)
DATA EPS / .0000001/
GBAR = 0.0
DO 480 IEL = 1, NET
G = ABS(A(IEL))
IF(G.GT. GBAR) GBAR = G
480 CONTINUE
IF(GBAR .EQ. 0.0) GO TO 600
DO 580 IEL = 1, NET
G = EPS * GBAR
G = ABS(G)
AIEL = ABS(A(IEL))
IF(G .GT. AIEL) A(IEL) = 0.0
580 CONTINUE
600 RETURN
END
ORIGINAL PAGE IS OF POOR QUALITY

CISPLAY...FILE DECK 40

LINE 1 SUBROUTINE TRANS( X, Y, Z, T, XP, YP, ZC ) 04000010
LINE 2 DIMENSION X(1), Y(1), Z(1), T(4), XP(1), YP(1) 04000020
LINE 3 TX = X(4) - X(1) 04000030
LINE 4 TY = Y(4) - Y(1) 04000040
LINE 5 TZ = 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 = TY*T2Z - TZ*T2Y 04000090
LINE 10 YN = T2X*T2Y - TX*T2Z 04000100
LINE 11 ZN = TX*T2Y - TY*T2X 04000110
LINE 12 XYZN = XN*ZN & YN*YNN & 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(1) & ZN*Z(1) 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(ZC-Z1) /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 IDENTICA 04000400
LINE 41 C IL FOR A SECTION. PROCEED WITH CAUTION // 1X, 120(1H+)///// ) 04000410
LINE 42 C RETURN 04000420
LINE 43 C1001 FORMAT( // 3(Z-E14.4,2X) ) 04000430
LINE 44 C1002 FORMAT( // (E14.4 ) ) 04000440
LINE 45 END 04000450

-6.86-
SUBROUTINE WAKET(I12,AROW,WROW,IF,LRECLW,NSPT,18,IZ,K1,NNCH)

COMPLEX WTES
COMPLEX AROW(NSPT), WROW(LRECLW)
COMPLEX AROWJ
LOGICAL TREDGE,WAKE,WAKENZ
DIMENSION IZ(NSPT)
DIMENSION NNCH(NSPT)
COMMON /FO1/FREQ
COMMON /K1112/KST11,KST12
COMMON/WAKEUP/TTREDGE,WAKE,WAKENZ,KWAKE

IF(K1.EQ.K) GO TO 60
READ(I8) K1,WROW
IF(K1.NE.K) RETURN

NNZ = 0
L = IF
DO 100 J = 1,NSPT
AROW = WROW(L)
AR = AROWJ
AC = MOD(AR **2 + AI **2)
IF (AC .EQ. 0.0) GO TO 100
NNZ = NNZ .AND. 1
IZ(NNZ) = NNCH(J)
AROW(NNZ) = AROWJ

100 L=LENFREQ
IF (NNZ .NE. 0) WAKENZ = .TRUE.
IF(NNZ.EQ.0) RETURN
CALL WRITZW(I12,K1,NNZ,IZ,AROW)
KST12 =K1
RETURN
END
LINE 1  SUBROUTINE WINGCK(CORNX, CORNY, CORNZ, NWING, ICN, ERROR, I6) 04200010
LINE 2      LOGICAL ERROR                           04200020
LINE 3      DIMENSION CORNX(1), CORNY(1), CORNZ(1) 04200030
LINE 4      COMMON ICNMI, ICN, ERROR, I6           04200040
LINE 5      ICNMI = ICN - 1                         04200050
LINE 6      DO 10 I = 1, ICNMI + 2                  04200060
LINE 7      J = I & 1                             04200070
LINE 8      IF(CORNX(J) .LT. CORNX(I))  GO TO 5    04200080
LINE 9      IF(CORNY(J) .NE. CORNY(I))  GO TO 5    04200090
LINE 10     IF(CORNZ(J) .NE. CORNZ(I))  GO TO 5    04200100
LINE 11     GO TO 10                              04200110
LINE 12     5 WRITE (6, 1001) I, J, NWING          04200120
LINE 13     ERROR = .TRUE.                         04200130
LINE 14     10 CONTINUE                           04200140
LINE 15     RETURN                                04200150
LINE 16     1001 FORMAT( //120(1X) ) /32H ERROR IN DEFINING CORNER POINTS, I2, 04200160
LINE 17     1 3HAND, I2, 8H OF WING, I2 /     04200170
LINE 18     END                                   04200180
SUBROUTINE WINGINI15,16, NWING, NSECT, AR, NSP, CORNX, CORNY, ERROR, BEETA, IPW, Nfreq, FREQ, 04300010
1 C
2 CORNZ, SYM, ICL, CL, ERROR, BEETA, IPW, FKEU, 04300020
3 RELEN, LAB, ID, FD 04300030
4 NWING = 0 04300040
5 BEERFL = BEETA*RELEN 04300050
6------------------------------------------------------------------04300060
7 READ(15,1000) LAB, ID, FD 04300070
8 IF (LAB .EQ. L2) GO TO 20 04300080
9 IF (LAB .EQ. L4) GO TO 40 04300090
10 IF (LAB .EQ. L11) GO TO 80 04300100
11 IF (LAB .EQ. L6) GO TO 65 04300110
12 IF (LAB .EQ. L8) GO TO 80 04300120
13 IF (LAB .EQ. L11) GO TO 80 04300130
14 IF (LAB .EQ. L11) GO TO 80 04300140
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 C
LINE 61 C  INVALID LABEL CARD READ
LINE 62 WRITE( I6, 2000 )
LINE 63 WRITE( I6, 2018 )
LINE 64 WRITE( I6, 2017 ) LAB, ID, FD
LINE 65 ERROR = .TRUE.
LINE 66 GO TO 10
LINE 67 C
LINE 68 C
LINE 69 C
LINE 70 20 CONTINUE
LINE 71 C  'WING' CARD HAS BEEN READ.
LINE 72 C
LINE 73 CLINPT = .FALSE.
LINE 74 NWING = NWING + 1
LINE 75 C
LINE 76 JW = ID(1)
LINE 77 NC = ID(2)
LINE 78 ISYM = ID(3)
LINE 79 IPW(JW) = ID(4)
LINE 80 NS = ID(5)
LINE 81 C
LINE 82 IF( NWING .LE. 4 ) GO TO 21
LINE 83 C  MORE THAN 4 WING DATA CARDS READ
LINE 84 C
LINE 85 WRITE( I6, 2000 )
LINE 86 WRITE( I6, 2001 )
LINE 87 ERROR = .TRUE.
LINE 88 C
LINE 89 IF( NWING .EQ. ID(1) ) GO TO 22
LINE 90 C  WING DATA CARD OUT OF SEQUENCE
LINE 91 WRITE( I6, 2000 )
LINE 92 WRITE( I6, 2011 )
LINE 93 WRITE( I6, 2012 ) ID(1), NWING
LINE 94 ERROR = .TRUE.
LINE 95 C
LINE 96 IF( ID(1) .GT. 0 .AND. ID(1) .LE. 4 ) GO TO 23
LINE 97 C  WING NUMBER NOT BETWEEN 1 AND 4
LINE 98 WRITE( I6, 2000 )
LINE 99 WRITE( I6, 2001 )
LINE 100 WRITE( I6, 2013 ) ID(1)
LINE 101 ERROR = .TRUE.
LINE 102 JW = 1
LINE 103 C
LINE 104 IF( NS .LE. 3 ) GO TO 24
LINE 105 C  MORE THAN 3 SECTIONS INPUT FOR GIVEN WING
LINE 106 WRITE( I6, 2000 )
LINE 107 WRITE( I6, 2011 )
LINE 108 WRITE( I6, 2005 ) JW, NS
LINE 109 ERROR = .TRUE.
LINE 110 NS = 3
LINE 111 C
LINE 112 NST = (NC+1)/2 - 1
LINE 113 IF(NS .EQ. NST ) GO TO 25
LINE 114 C  NO CORNER PTS NOT CONSISTANT WITH NO. SECTIONS OF WING
LINE 115 WRITE( I6, 2000 )
LINE 116 WRITE( I6, 2011 )
LINE 117 WRITE( I6, 2006 ) NC, NS, JW
LINE 118 ERROR = .TRUE.

-6.90-
LINE 119 IF ( VST .LE. 3 ) NS = NST
LINE 120 25 CONTINUE
LINE 121 C NSECT(JW) = NS
LINE 122 LINE 123 SYM(JW) = FLOAT(ISYM)
LINE 124 DD 26 I = 1,NS
LINE 125 LINE 126 ICL(I,JW) = 0
LINE 127 NSP(I,JW) = ID(I65)
LINE 128 AR(JW) = FD(I1)
LINE 129 C CHECK ON SYMMETRY CODE
LINE 130 IF ( IABS(ISYM) .LE. 1 ) GO TO 27
LINE 131 WRITE(16,2000)
LINE 132 C 28 CONTINUE
LINE 133 WRITE(16,2001 J,W,ISYM)
LINE 134 ERROR = .TRUE.
LINE 135 27 CONTINUE
LINE 136 C 28 CONTINUE
LINE 137 WRITE(16,3000) JW, NC, ISYM, SKODE, IPW(JW), NSECT(JW)
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) = SK1(I)
LINE 149 C 34 CONTINUE
LINE 150 C 35 CONTINUE
LINE 151 WRITE(16,3001) JW, NC, ISYM, SKODE, IPW(JW), NSECT(JW)
LINE 152 WRITE(16,3000) JW, NC, ISYM, SKODE, IPW(JW), NSECT(JW)
LINE 153 WRITE(16,3001) JW, NC, ISYM, SKODE, IPW(JW), NSECT(JW)
LINE 154 WRITE(16,3000) JW, NC, ISYM, SKODE, IPW(JW), NSECT(JW)
LINE 155 C 36 CONTINUE
LINE 156 C READ CORNER POINTS
LINE 157 WRITE(16,3004) JW, NC, CORNX(I,JW), CORNY(I,JW), CORNZ(I,JW)
LINE 158 DD 36 I = 1,NC
LINE 159 READ(15,1000) LAB, ID, FD
LINE 160 READ(15,1000) LAB, ID, FD
LINE 161 IF ( I .GT. 1 ) GO TO 35
LINE 162 IF ( I .GT. 1 ) GO TO 35
LINE 163 C CORNER DATA CARD MISSING
LINE 164 WRITE(16,3000)
LINE 165 WRITE(16,3000)
LINE 166 ERROR = .TRUE.
LINE 167 35 CONTINUE
LINE 168 CORNX(I,JW) = FD(1)
LINE 169 CORNY(I,JW) = FD(2)
LINE 170 CORNZ(I,JW) = FD(3)
LINE 171 36 CONTINUE
LINE 172 C 37 CONTINUE
LINE 173 WRITE(16,3004) JW, NC, CORNX(I,JW), CORNY(I,JW), CORNZ(I,JW)
LINE 174 C IF ( 2*(NC/2) .EQ. NC ) GO TO 37
LINE 175 CORNX(NC+1,JW) = CORNX(NC,JW)
LINE 176 CORNY(NC+1,JW) = CORNY(NC,JW)
LINE 177 CORNZ(NC+1,JW) = CORNZ(NC,JW)
LINE 178 NC = NC+1
LINE 179 37 CONTINUE
LINE 180
CALL ROUTINE TO CHECK FOR PHYSICAL ERROR ON CORNER POINT

CALL WINGCK(CORNX(1, JW), CORNY(1, JW), CORNZ(1, JW), JW, NL, ERROR, 16)

DO 38 I = 1, NC

CORNX(1, JW) = CORNX(1, JW) / BEERFL
CORNY(1, JW) = CORNY(1, JW) / REFLN
CORNZ(1, JW) = CORNZ(1, JW) / REFLN

CONTINUE

GO TO 10

C PROCESS CONTROL LINE INFORMATION

IS = ID(1)
K = ICL(IS, JW) & 1
IF( IS .GT. NSECT(JW) ) GO TO 41
WRITE( 16, 2000 ) JW
WRITE( 16, 2001 ) IS, JW, NSECT(JW)
ERROR = .TRUE.
GO TO 10

IF( K .LE. 2 ) GO TO 42
MORE THAN TWO CONTROL LINES READ FOR SAME SECTION
WRITE( 16, 2000 ) JW
WRITE( 16, 2001 ) IS
ERROR = .TRUE.

IF( K .EQ. ID(2) ) GO TO 43
CONTROL LINE DATA OUT OF SEQUENCE
WRITE( 16, 2000 ) JW
WRITE( 16, 2001 ) IS
ERROR = .TRUE.

IF( .NOT. CLINPT ) WRITE( 16, 3002 ) JW
CLINPT = .TRUE.
WRITE( 16, 3003 ) IS, K, FD(1), FD(2)

X1 = FD(1) / BEERFL
X2 = FD(2) / BEERFL
IGN = 2 * IS - 1
IF( X1 .LT. CORNX(IGN, JW) ) GO TO 44
IF( X2 .LT. CORNX(IGN+2, JW) ) GO TO 44
IF( X2 .LT. CORNX(IGN+3, JW) ) GO TO 44
GO TO 45

CONTROL LINE DEFINED OUTSIDE OF SECTION
WRITE( 16, 3002 ) JW
WRITE( 16, 3003 ) IS, K, FD(1), FD(2)

CONTINUE

CONTROL LINE SPECIFIED FOR SECTION THAT DOES NOT EXIST
WRITE( 16, 2000 ) JW
WRITE( 16, 2001 ) IS, JW, NSECT(JW)
ERROR = .TRUE.

GO TO 10

IF( K .LE. 2 ) GO TO 42
MORE THAN TWO CONTROL LINES READ FOR SAME SECTION
WRITE( 16, 2000 ) JW
WRITE( 16, 2001 ) IS
ERROR = .TRUE.

IF( .NOT. CLINPT ) WRITE( 16, 3002 ) JW
CLINPT = .TRUE.
WRITE( 16, 3003 ) IS, K, FD(1), FD(2)

X1 = FD(1) / BEERFL
X2 = FD(2) / BEERFL
IGN = 2 * IS - 1
IF( X1 .LT. CORNX(IGN, JW) ) GO TO 44
IF( X2 .LT. CORNX(IGN+2, JW) ) GO TO 44
IF( X2 .LT. CORNX(IGN+3, JW) ) GO TO 44
GO TO 45

CONTROL LINE DEFINED OUTSIDE OF SECTION
WRITE( 16, 3002 ) JW
WRITE( 16, 3003 ) IS, K, FD(1), FD(2)

CONTINUE

CONTROL LINE SPECIFIED FOR SECTION THAT DOES NOT EXIST
WRITE( 16, 2000 ) JW
WRITE( 16, 2001 ) IS, JW, NSECT(JW)
ERROR = .TRUE.

GO TO 10

IF( K .LE. 2 ) GO TO 42
MORE THAN TWO CONTROL LINES READ FOR SAME SECTION
WRITE( 16, 2000 ) JW
WRITE( 16, 2001 ) IS
ERROR = .TRUE.

IF( .NOT. CLINPT ) WRITE( 16, 3002 ) JW
CLINPT = .TRUE.
WRITE( 16, 3003 ) IS, K, FD(1), FD(2)

X1 = FD(1) / BEERFL
X2 = FD(2) / BEERFL
IGN = 2 * IS - 1
IF( X1 .LT. CORNX(IGN, JW) ) GO TO 44
IF( X2 .LT. CORNX(IGN+2, JW) ) GO TO 44
IF( X2 .LT. CORNX(IGN+3, JW) ) GO TO 44
GO TO 45

CONTROL LINE DEFINED OUTSIDE OF SECTION
WRITE( 16, 3002 ) JW
WRITE( 16, 3003 ) IS, K, FD(1), FD(2)

CONTINUE

CONTROL LINE SPECIFIED FOR SECTION THAT DOES NOT EXIST
WRITE( 16, 2000 ) JW
WRITE( 16, 2001 ) IS, JW, NSECT(JW)
ERROR = .TRUE.

GO TO 10

IF( K .LE. 2 ) GO TO 42
MORE THAN TWO CONTROL LINES READ FOR SAME SECTION
WRITE( 16, 2000 ) JW
WRITE( 16, 2001 ) IS
ERROR = .TRUE.

IF( .NOT. CLINPT ) WRITE( 16, 3002 ) JW
CLINPT = .TRUE.
WRITE( 16, 3003 ) IS, K, FD(1), FD(2)

X1 = FD(1) / BEERFL
X2 = FD(2) / BEERFL
IGN = 2 * IS - 1
IF( X1 .LT. CORNX(IGN, JW) ) GO TO 44
IF( X2 .LT. CORNX(IGN+2, JW) ) GO TO 44
IF( X2 .LT. CORNX(IGN+3, JW) ) GO TO 44
GO TO 45

CONTROL LINE DEFINED OUTSIDE OF SECTION
WRITE( 16, 3002 ) JW
WRITE( 16, 3003 ) IS, K, FD(1), FD(2)

CONTINUE

CONTROL LINE SPECIFIED FOR SECTION THAT DOES NOT EXIST
WRITE( 16, 2000 ) JW
WRITE( 16, 2001 ) IS, JW, NSECT(JW)
ERROR = .TRUE.

GO TO 10
LINE 243 IF ( X1 .GE. CL(1,IS,JW) ) .AND. X2 .GE. CL(2,IS,JW) ) GO TO 46
LINE 246 WRITE ( 16, 2000 )
LINE 247 WRITE ( 16, 2008 ) JW
LINE 248 WRITE ( 16, 2013 ) IS
LINE 249 CONTINUE
LINE 250 JCN = 2*K - 1
LINE 251 CL(JCN,IS,JW) = X1
LINE 252 CL(JCN+1,IS,JW) = X2
LINE 253 GO TO 10
LINE 254 C SECOND CONTROL LINE DEFINED ABOVE FIRST
LINE 255 CONTINUE
LINE 256 WRITE ( 16, 2003 )
LINE 257 WRITE ( 16, 2009 )
LINE 258 IF ( X1 .GT. CL(1,IS,JW) ) READ ( 15, 1001 ) ( FREQ(I), I=5, NFREQ )
LINE 259 FREQ = ID(I)
LINE 260 WRITE ( 16, 3036 ) NFREQ, ( FREQ(I), I=1,NFREQ )
LINE 261 CONTINUE
LINE 262 FREQL = .TRUE.
LINE 263 GO TO 10
LINE 264 C FREQUENCY CARD READ
LINE 265 CONTINUE
LINE 266 WRITE ( 16, 2000 )
LINE 267 WRITE ( 16, 2004 )
LINE 268 IF ( I .GT. NSECT(J) ) GO TO 81
LINE 269 IF ( J .GT. NHING ) GO TO 81
LINE 270 ICL(I,J) = -1
LINE 271 CONTINUE
LINE 272 C GRID DATA SPECIFIED FOR SECTION THAT DOES NOT EXIST
LINE 273 WRITE ( 16, 2000 )
LINE 274 WRITE ( 16, 2015 )
LINE 275 IF ( I .GT. NSECT(J) ) GO TO 81
LINE 276 WRITE ( 16, 2000 )
LINE 277 WRITE ( 16, 2015 )
LINE 278 IF ( J .GT. NHING ) GO TO 81
LINE 279 WRITE ( 16, 2015 ) ID, FD
LINE 280 RETURN
LINE 281 C LABEL CARD EXPECTED BUT NOT RECEIVED
LINE 282 WRITE ( 16, 2000 )
LINE 283 WRITE ( 16, 2004 )
LINE 284 WRITE ( 16, 2015 )
LINE 285 IF ( I .GT. NSECT(J) ) GO TO 81
LINE 286 IF ( J .GT. NHING ) GO TO 81
LINE 287 RETURN
LINE 288 C GRID CARD READ
LINE 289 CONTINUE
LINE 290 C GRID DATA SPECIFIED FOR SECTION THAT DOES NOT EXIST
LINE 291 IF ( I .GT. NSECT(J) ) GO TO 81
LINE 292 RETURN
LINE 293 IF ( J .GT. NHING ) GO TO 81
LINE 294 RETURN
LINE 295 IF ( I .GT. NSECT(J) ) GO TO 81
LINE 296 RETURN
LINE 297 C GRID DATA SPECIFIED FOR SECTION THAT DOES NOT EXIST
LINE 298 C GRID CARD READ
LINE 299 WRITE ( 16, 2000 )
LINE 300 WRITE ( 16, 2004 )
LINE 301 WRITE ( 16, 2015 ) J, I
LINE 302 IF ( J .GT. NHING ) GO TO 81
LINE 303 RETURN
LINE 304 C
DISPLAY...FILE DECK 44

LINE 1 SUBROUTINE WINGRI X, Y, Z
LINE 2 COMPLEX X1, Y1, Z1
LINE 3 INTEGER GOBACK
LINE 4 LOGICAL EKDEONE, EMIRRO, THAILK
LINE 5 LOGICAL ERROR
LINE 6 LOGICAL MIRROR
LINE 7 LOGICAL TECOD
LINE 8 LOGICAL TRAINF
LINE 9 LOGICAL TREDGE, WAKE
LINE 10 DIMENSION CAR(12), CX(12)
LINE 11 DIMENSION RK(30)
LINE 12 DIMENSION RK(30)
LINE 13 DIMENSION W(30)
LINE 14 DIMENSION WAVE(20)
LINE 15 DIMENSION X(1), Y(1), WROW(1)
LINE 16 CALL BASIC/MACH, BEETA, KAPPA#1, PRNTL51
LINE 17 CALL / EEW / XUP, YUP, XLW, YLW, EREUNE, EMIRRO, RR, RL, RRR
LINE 18 CALL / FREQ / CF(12, 3), FREQ(12), FREQ(12), FREQ(12), FREQ(12), FREQ(12)
LINE 19 CALL / WAVE / WAVE(20)
LINE 20 CALL / WAVE / WAVE(20)
LINE 21 CALL / WAVE / WAVE(20)
LINE 22 CALL / WAVE / WAVE(20)
LINE 23 CALL / WAVE / WAVE(20)
LINE 24 CALL / WAVE / WAVE(20)
LINE 25 DATA KCONV: 0.0
LINE 26 DATA X1, Y1, Z1
LINE 27 1, 11.08, 11.81, 12.55, 13.28, 14.02, 14.76, 15.49, 16.23, 16.97
LINE 28 2, 17.71, 18.44, 19.18, 19.92, 20.6
LINE 29 3, 17.71, 18.44, 19.18, 19.92, 20.6
LINE 30 C X COORD. OF POINTS OF INFLUENCING ELEMENT
LINE 31 C Y COORD. OF POINTS OF INFLUENCING ELEMENT
LINE 32 C X COORD. OF REceiving PT. (CENTER OF ELEMENT)
LINE 33 C X COORD. OF REceiving PT. (IN RIGHT SIDE)
LINE 34 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 35 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 36 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 37 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 38 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 39 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 40 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 41 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 42 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 43 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 44 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 45 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 46 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 47 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 48 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 49 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 50 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 51 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 52 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 53 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 54 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 55 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 56 C X COORD. OF REceiving PT. (IN LEFT SIDE)
LINE 57 C TEST LEADING EDGE LINE
LINE 58 YLW = Y(1)
LINE 59 YUP = Y(2)
LINE 60 XLW = X(1)
LINE 61 XUP = X(2)
LINE 62 C CALL EODE
LINE 63 C
LINE 64 C IF ELEMENT IS NOT IN MAC4 CONE, RETURN WITH ZERU IN XRIN.
LINE 65 C PROCEED ANYWAY IF IT IS A SELF ELEMENT (DIAGONAL TERM)
LINE 66 C
LINE 67 IF( I.EQ. 0 ) GO TO 30
LINE 68 C RETURN
LINE 69 C
LINE 70 C
LINE 71 20 CONTINUE
LINE 72 I.EQ. 1 AND. IIN .EQ. 1 ) KCONV = 0
LINE 73 C
LINE 74 ASSIGN 310 TO IWAKE1
LINE 75 ASSIGN 160 TO IWAKE2
LINE 76 C
LINE 77 NTERM = 1
LINE 78 RRU = RR
LINE 79 RLU = RL
LINE 80 MIRRD = EMMKD
LINE 81 TREV = .FALSE.
LINE 82 WAKE1 = .FALSE.
LINE 83 IIF NOT. TRIE ) GO TO 25
LINE 84 C TEST TRAILING EDGE LINE
LINE 85 C
LINE 86 YLW = Y(2)
LINE 87 YUP = Y(1)
LINE 88 XLW = X(3)
LINE 89 XUP = X(4)
LINE 90 C CALL EODE
LINE 91 C
LINE 92 C
LINE 93 YLW = Y(1)
LINE 94 YUP = Y(2)
LINE 95 XLW = X(1)
LINE 96 XUP = X(2)
LINE 97 C IF NOT. EREDONE GO TO 40
LINE 98 C TRAVE = .TRUE.
LINE 99 C IF( LATE .NE. 0 AND. RTR .GT. 0 ) WAKE1 = .TRUE.
LINE 100 GO TO 40
LINE 101 C
LINE 102 C
LINE 103 25 CONTINUE
LINE 104 I( INSID .EQ. 0 ) GO TO 40
LINE 105 DD 30 I=1,INFREQ
LINE 106 30 WRD(I) = -WLEAD(I)
LINE 107 C
LINE 108 40 CONTINUE
LINE 109 DD 45 I=1,INFREQ
LINE 110 45 WLEAD(I) = 0.0
LINE 111 C
LINE 112 C
LINE 113 C NINSID = NINSID & 1
LINE 114 C
LINE 115 Y0 = YK0
LINE 116 Z0 = ZR
LINE 117 Z10 = Z0UR
LINE 118 TV = TV(1)

-6.96-
LINE 119  $T_d = TVW(12)$
LINE 120  C  IF ( FSTEST .EQ. 0.0 ) GO TO 145
LINE 121  C  DECIDE HOW MANY TERMS ARE NEEDED BASED ON RK
LINE 122  C  SECTION FOR NEAR FIELD, LEFT HAND SIDE LEADING EDGE
LINE 123  C  SECTION FOR NEAR FIELD, LEFT HAND SIDE OF LEADING EDGE
LINE 124  C  SECTION FOR NEAR FIELD, LEFT HAND SIDE OF LEADING EDGE
LINE 125  C  ASSIGN 145 TO GOBACK
LINE 126  C  GO TO 600
LINE 127  CALL WINT
LINE 128  CONTINUE
LINE 129  CALL WINT
LINE 130  CONTINUE
LINE 131  C  CHECK IF LEFT HAND ELEMENT (MIRROR IMAGE) EXISTS
LINE 132  C  ASSIGN 150 TO GOBACK
LINE 133  C  GO TO 800
LINE 134  CONTINUE
LINE 135  DD 155 I=1,NFREQ
LINE 136  WLE4(I) = WLEAD(I) * SKA(I)
LINE 137  WROW(I) = WROW(I) * SKA(I)
LINE 138  CONTINUE
LINE 139  C  GO TO MAKE2, (160,320)
LINE 140  CONTINUE
LINE 141  C  CHECK IF NOT MIRROR) GO TO 300
LINE 142  C  SECTION FOR LEFT HAND SIDE OF LEADING EDGE
LINE 143  C  SECTION FOR LEFT HAND SIDE OF LEADING EDGE
LINE 144  C  ASSIGN 185 TO GOBACK
LINE 145  C  GO TO 600
LINE 146  CONTINUE
LINE 147  IF ( FSTEST .EQ. 0.0 ) GO TO 185
LINE 148  C  DECIDE HOW MANY TERMS ARE NEEDED BASED ON RK
LINE 149  C  SECTION FOR NEAR FIELD, LEFT HAND SIDE LEADING EDGE
LINE 150  C  SECTION FOR NEAR FIELD, LEFT HAND SIDE LEADING EDGE
LINE 151  C  SECTION FOR NEAR FIELD, LEFT HAND SIDE LEADING EDGE
LINE 152  RK = £FRTEST
LINE 153  ASSIGN 185 TO GOBACK
LINE 154  GO TO 600
LINE 155  CONTINUE
LINE 156  Y = YLU
LINE 157  ZD = ZL
LINE 158  ZDZD = ZLZL
LINE 159  TV = TVK(3)
LINE 160  TW = TVK(4)
LINE 161  C  CALL WINT
LINE 162  CALL WINT
LINE 163  C  ASSIGN 190 TO GOBACK
LINE 164  C  ASSIGN 190 TO GOBACK
LINE 165  GO TO 800
LINE 166  CONTINUE
LINE 167  DD 195 I=1,NFREQ
LINE 168  WLEAD(I) = WLEAD(I) * SKA(I)
LINE 169  WROW(I) = WROW(I) * SKA(I)
LINE 170  CONTINUE
LINE 171  C  FIND INFLUENCE OF TRAILING EDGE OF TRAILING EDGE ELEMENT
LINE 172  CONTINUE
LINE 173  C  IF NOT TRAINF) RETURN
LINE 174  GO TO MAKE1, (310,320)
LINE 175  CONTINUE
LINE 176  C  FIND INFLUENCE OF TRAILING EDGE OF TRAILING EDGE ELEMENT
LINE 177  C  FIND INFLUENCE OF TRAILING EDGE OF TRAILING EDGE ELEMENT
LINE 178  CONTINUE
LINE 179  C  FIND INFLUENCE OF TRAILING EDGE OF TRAILING EDGE ELEMENT
LINE 180  C  FIND INFLUENCE OF TRAILING EDGE OF TRAILING EDGE ELEMENT

-6.97-
(THE RIGHT HAND ELEMENT ONLY AT THIS TIME)

LINE 131 C
LINE 132 C
LINE 133 C
LINE 134 C
LINE 135 C
LINE 136 C
LINE 137 C
LINE 138 C
LINE 139 C
LINE 140 C
LINE 141 C
LINE 142 C
LINE 143 C
LINE 144 C
LINE 145 C
LINE 146 C
LINE 147 C
LINE 148 C
LINE 149 C
LINE 150 C
LINE 151 C
LINE 152 C
LINE 153 C
LINE 154 C
LINE 155 C
LINE 156 C
LINE 157 C
LINE 158 C
LINE 159 C
LINE 160 C
LINE 161 C
LINE 162 C
LINE 163 C
LINE 164 C
LINE 165 C
LINE 166 C
LINE 167 C
LINE 168 C
LINE 169 C
LINE 170 C
LINE 171 C
LINE 172 C
LINE 173 C
LINE 174 C
LINE 175 C
LINE 176 C
LINE 177 C
LINE 178 C
LINE 179 C
LINE 180 C
LINE 181 C
LINE 182 C
LINE 183 C
LINE 184 C
LINE 185 C
LINE 186 C
LINE 187 C
LINE 188 C
LINE 189 C
LINE 190 C
LINE 191 C
LINE 192 C
LINE 193 C
LINE 194 C
LINE 195 C
LINE 196 C
LINE 197 C
LINE 198 C
LINE 199 C
LINE 200 C
LINE 201 315 C
LINE 202 C
LINE 203 C
LINE 204 C
LINE 205 C
LINE 206 C
LINE 207 C
LINE 208 320 C
LINE 209 C
LINE 210 C
LINE 211 C
LINE 212 C
LINE 213 C
LINE 214 C
LINE 215 C
LINE 216 C
LINE 217 C
LINE 218 C
LINE 219 C
LINE 220 C
LINE 221 C
LINE 222 C
LINE 223 C
LINE 224 C
LINE 225 C
LINE 226 C
LINE 227 C
LINE 228 C
LINE 229 C
LINE 230 C
LINE 231 C
LINE 232 C
LINE 233 C
LINE 234 C
LINE 235 C
LINE 236 C
LINE 237 C
LINE 238 C
LINE 239 C
LINE 240 325 C
LINE 241 C
LINE 242 C

THE RIGHT HAND ELEMENT ONLY AT THIS TIME

-6.98-
LINE 243 C SECTION TO FIND WAKE EFFECT OF THE REST OF THE WAKE ELEMS
LINE 244 C 
LINE 245 C Xlw = Average X
LINE 246 C XLW = 0.5*(XLW+YUP)
LINE 247 C YUP = Y(1)
LINE 248 C ELS = (Y0-0.5)*2
LINE 249 C EUS = (Y0-YUP)*2
LINE 250 C DO 360 LINE = 2, NW
LINE 251 C DO 227 I=1,NFREQ
LINE 252 C "WHEAD(I) = 0.0"
LINE 253 C XLW = XLW & DELTA Z
LINE 254 C XUP = XLW
LINE 255 C ZETA = XLW
LINE 256 C ZSZTE = ZSMZTE - DELTA Z
LINE 257 C IF(FREAT. EQ. 0.0) GO TO 330
LINE 258 C O.5*(SQRT(ZETA**2 - ELS) & SQRT(ZETA**2 - EUS)
LINE 259 C RK = RR*FRTEST
LINE 260 C ASSIGN 330 TO GOBACK
LINE 261 C GO TQ 360
LINE 262 C CONTINUE
LINE 263 C CALL HVINT
LINE 264 C ASSIGN 340 TO GOBACK
LINE 265 C GO TQ 800
LINE 266 C CONTINUE
LINE 267 C DO 345 I=1,NFREQ
LINE 268 C NLEAD(I) = NLEAD(I) - SKPA(I)
LINE 269 C DO 350 J=1,NFREQ
LINE 270 C FBZ = (FREE(IJ)/BEETA)*ZSZTE
LINE 271 C CXIPR = COS(FBZ)
LINE 272 C CXIPI = SIN(FBZ)
LINE 273 C WTEAS(J) = WTEAS(J)
LINE 274 C (WLEAD(IJ)*CXRIJ - CXIPR)*WLEAD(IJ)*CXRIJ - CXIPR)
LINE 275 C CXR(J) = CXIPR
LINE 276 C CXI(J) = CXIPI
LINE 277 C CONTINUE
LINE 278 C END LOOP ON LINE
LINE 279 C CHECK IF MIRROR IMAGE OF TRAILING EDGE LINE HAS INFLUENCE AND
LINE 280 C HAS NOT YET BEEN CONSIDERED.
LINE 281 C IF NOT, EMIRRI J GO TO 380
LINE 282 C IF(0.0) GO TO 380
LINE 283 C FIND INFLUENCE OF THE MIRROR IMAGE OF THE TRAILING EDGE.
LINE 284 C SET EMIRRI TO .FALSE. SO THAT WE DO NOT BRANCH HERE AGAIN
LINE 285 C EMIRRI = .FALSE.
LINE 286 C CONTINUE
LINE 287 C DO 370 I=1,NFREQ

-6.99-
END OF JOB.

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

LINE 1 SUBROUTINE WETA ( NET, JMODE, WETA, 113 )
LINE 2 DIMENSION WETA(NET,10), WDATA(NET,10)
LINE 3 REWIND 113
LINE 4 DO 30 JM = 1, JMODE
LINE 5 WRITE (113) JM, NET, (WDATA(I,JM), I=1,NET), (WETA(I,JM), I=1,NET)
LINE 6 30 CONTINUE
LINE 7 END FILE 113
LINE 8 REWIND 113
LINE 9 RETURN
LINE 10 END

-6.102-
DISPLAY...FILE

LINE 1  SUBROUTINE WRITED (K,AROW, IZ, ANEW, NET) 04600010
LINE 2  DIMENSION AROW(NET), IZ(NET), ANEW(NET) 04600020
LINE 3  COMMON /K1112/ KST11,KST12 04600030
LINE 4  COMM TY /TAPE/ 19,110,111 04600040
LINE 5  NNZ=0 04600050
LINE 6  DO 100 I=1,NET 04600060
LINE 7  IF (AROW(I) .EQ. 0.0) GO TO 100 04600070
LINE 8  NNZ = NNZ+1 04600080
LINE 9  M = NNZ 04600090
LINE 10  IZ(M) = I 04600100
LINE 11  ANEW(M) = AROW(I) 04600110
LINE 12  100 CONTINUE 04600120
LINE 13  IF(NNZ.EQ.0) RETURN 04600130
LINE 14  CALL WRITZI I11,K,NNZ,IZ,ANEW) 04600140
LINE 15  KST11 =K 04600150
LINE 16  RETURN 04600160
LINE 17  END 04600170

-6.103-
CISPLAY.FILE

DECK 48

LINE 1 SUBROUTINE WRITW (112, K, NNZ, L, ANEW)
LINE 2 DIMENSION IZ(NNZ)
LINE 3 COMPLEX ANEW(NNZ)
LINE 4 WRITE(112) K, NNZ, IZ, ANEW
LINE 5 200 CONTINUE
LINE 6 RETURN
LINE 7 END

-6.105-
SUBROUTINE WTEPHT (DEDT, NET, PHIW, IZ, I12, WTE)

COMPLEX DEDT(1), PHIW(1), WTE(1)

DIMENSION IZ(1)

COMMON /K1112/ KST11, KST12

KEL = 0

DO 300 IEL = 1, NET

IF (IEL .GT. KST12) GO TO 300

IF (KEL .LT. IEL) GO TO 300

DO 200 M = 1, NNZ

KEZ(M) = IZ(M)

I = IEL

DEDT(I) = DEDT(I) - WTE(M) * PHIW(K)

CONTINUE

300 CONTINUE

REWIND 112

RETURN

END
DECK 50

SUBROUTINE WVNY

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

Z D DIFFERENCE BETWEEN Z'S OF INFL. AND RECEIV. ELEMS.

V, W SIDE WASH AND UNPANSH INTEGRALS

TERM NUMBER OF TERMS IN APPROXIMATION OF V AND W

REAL L1

DIMENSION GETA(5), GCOF(5), XX(3), AG(3)

DIMENSION H(31)

COMMON / EEW / XUP, YUP, XLW, YLW

COMMON / H/ V(30)

COMMON / WV1 / NTERM, XO,YO, ZU, ZDZD

DATA NGT / 5 /

OA TA / P102 / 1.570796

XX(11, AG(1) XX(2) AG(2) XX(3), AG(3)

* 0.9061798459386639279786286782

* 0.53846931010568309103631442079

* 0.5588888888888888886888888888

** SET UP CONSTANTS

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 = EUEP & ZDZD

TTLW = EELW & ZDZD

RRUP = ZLZU - TTUP

RRlw = ZLZU - TTlw

IF ( RRRUP .LT. 0.0 .OR. ZUP. LE. 0.0 ) RRRUP = 0.0

IF ( RRrlw .LT. 0.0 .OR. ZLW. LE. 0.0 ) RRrlw = 0.0

RRUP = SQRT(RRRUP)

RRlpw = SQRT( RRrlw )

RUP = SQRTI RRRUP

RLW = SQRTI RRlw

SS = SLOPE*SLOPE

TESTC = SS - 1.0

AMZ = AA & SS * ZDZD

ACZ = AMZ - ZDZD

SIGNU = SIGN ( PI02, EUP )

SIGN = SIGN ( PI02, ELW )

SZZ = SLOPE * ZDZD

F0

SORC = ABS( TESTC)

SIGGC = SIGN( SORC)

SORC = SQRT( SORC)

SIGACZ = SIGN( ACZ)

IFI ( TESTC .GT. 0.0 ) GO TO 22
LINE 57 C C .LT. .0 ( TESTC = C ) COMPUTE S2
LINE 58 FU = -SIGNU
LINE 59 IF( RUP .GT. 0.0 ) FU = -1.0 ( SLOPE*ZUP-EUP/STACZ )
LINE 60 FL = -SIGNL
LINE 61 IF( RLW .GT. 0.0 ) FL = -1.0 ( SLOPE*ZLW-ELW/STACZ )
LINE 62 FO = -( FL - FL )/STAC
LINE 63 GO TO 30
LINE 64 20 CONTINUE
LINE 65 C C = 0
LINE 66 FO = ( RUP - RLW ) / ( SLOPE-LPHA )
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 NOR RLW IS ZERO
LINE 74 FO = ( ALOG(FU/FL) )/STAC
LINE 75 GO TO 30
LINE 76 C C .GT. 0 RUP .EQ. 0
LINE 77 24 CONTINUE
LINE 78 FO = ( ALOG(ABS(SACZ/FL1) ) )/STAC
LINE 79 GO TO 30
LINE 80 26 CONTINUE
LINE 81 FO = ( ALOG(ABS(FU/SACZ) ) )/STAC
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( (ZUPRUP)/ALL )
LINE 89 ALL = 0.0
LINE 90 AL1 = ZLW - RLW
LINE 91 IF( RLW .GT. 0.0 & AND. ALL .GT. .1E-5 )
LINE 92 1 ALL = 0.5*ALOG( (ZLWRLW)/ALL )
LINE 93 AL1 = 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*ELP - 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. 0 AND TO .NE. 0
LINE 104 S1 = ARSIN(FU/SU) - ARSIN(FL/SL)
LINE 105 GO TO 40
LINE 106 32 CONTINUE
LINE 107 S1 = SIGN( PID2, FU ) - ARSIN(FL/SL)
LINE 108 GO TO 40
LINE 109 34 CONTINUE
LINE 110 S1 = ARSIN(FU/SU) - SIGN( PID2, 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*F1 - SLOPE*F2
LINE 116 C

-6.108-
```
LINE 119 IF( NTERM .LE. 1 ) GO TO 71
LINE 120 C
LINE 121  W(2) =
LINE 122    1  -0.5*( (ZUP+ALPHA)*EUP-ALU - (ZLW+ALPHA)*ELW-ALL )
LINE 123    2  - RUP*EUP + RLU*ELW - 0.5*(AA-3.0)*TESTC*ZOD*FD
LINE 124    3  - (1.5*SZZ-2.0*AA)*F1 & 3.5*SZZ*F2
LINE 125 C
LINE 126 IF( NTERM .LE. 2 ) GO TO 71
LINE 127 C
LINE 128 NHT = NTERM - 2
LINE 129 IW = NTERM - 1
LINE 130 C
LINE 131 C  FIND 5 PT. GAUSS-LEFEBEU 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       FO = F1*X(IG)
LINE 137       GETA(4) = FO & F2
LINE 138       GETA(IG) = -FO & F2
LINE 139       GCOF(IG) = F1*AG(IG)
LINE 140 43  GCOF(N) = GCOF(IG)
LINE 141 C
LINE 142 C
LINE 143   DO 45  N=3,NTERM
LINE 144   V(N) = 0.0
LINE 145   45  W(N) = 0.0
LINE 146 C
LINE 147 C
LINE 148 C  LOOP ON TERMS OF QUADRATURE
LINE 149 C
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 & ZOD
LINE 155   RR = ZETA*ZETA & TTHH
LINE 156   IF( RR .LE. G-0 ) GO TO 70
LINE 157   COF = 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*ALOG( (ZETA&RM)/FL )
LINE 162 C
LINE 163   H(1) = ZETA*ALH - RH
LINE 164   DO 50  N=1,NHT
LINE 165       NH = N+1
LINE 166       N2 = 2*N
LINE 167       RH = RH*RH
LINE 168       H(NH) = 9.0/(N2*(N2+11)) - (N2-1)*TTHH*H(N)/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,IWT
LINE 173       NW = N+1
LINE 174   WN(W) = W(NW) - COF*(2*N-1)*H(N) - (2*(N-1)-1)*ZOD*H(N-1)
LINE 175 60  CONTINUE
LINE 176 C
LINE 177 70  CONTINUE
LINE 178 C
LINE 179 C
LINE 180 71  CONTINUE
```
LINE 181 C V(1) = 0.0 05001810
LINE 182 C V(2) = 0.0 05001820
LINE 183 C 05001830
LINE 184 C 05001840
LINE 185 C IF ZD = 0, THERE IS NO SIDEWASH 05001850
LINE 186 C IF( ZD .EQ. 0.0 ) RETURN 05001860
LINE 187 C 05001870
LINE 188 C 05001880
LINE 189 C SECTION FOR SIDEWASH 05001890
LINE 190 C 05001900
LINE 191 C V(1) = - ZD* ( RUP/TTUP - RLW/TTLW ) 05001910
LINE 192 C IF( NTERM .LE. 1 ) GO TO 90 05001920
LINE 193 C HU = ZUP*ALU - RUP 05001930
LINE 194 C HL = ZLW*ALL - RLW 05001940
LINE 195 C V(2) = - ZD* ( HU - HL ) 05001950
LINE 196 C IF( NTERM .LE. 2 ) GO TO 90 05001960
LINE 197 C RSU = RUP 05001970
LINE 198 C RSL = RLW 05001980
LINE 199 C 05001990
LINE 200 C DO 80 N=2,IWT 05002000
LINE 201 C NV = N61 05002010
LINE 202 C NH = N-1 05002020
LINE 203 C 05002030
LINE 204 C N2 = 2*NH 05002040
LINE 205 C NOTE THAT H IS COMPUTED FOR NH = N-1 05002050
LINE 206 C IF( RUP .EQ. 0.0 ) GO TO 74 05002060
LINE 207 C RSU = RSL*RRUP 05002070
LINE 208 C 05002080
LINE 209 C HU = RSU/(N2*(N2+1)) - (N2-1)*TTUP*HU/N2 05002090
LINE 210 C 05002100
LINE 211 C HL = RSL/(N2*(N2+1)) - (N2-1)*TTLW*HL/N2 05002110
LINE 212 C 05002120
LINE 213 C 05002130
LINE 214 C 05002140
LINE 215 C 05002150
LINE 216 C 05002160
LINE 217 C RETURN 05002170
LINE 218 C END 05002180

-6,110-
SUBROUTINE ZFDZI (ZFUN, DZDX, X, Y, A)

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

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

COMMON /ZFUNY/ N, IEHF, B1, R2, R3, A(FUNMX)

COMMON /XYSCL/ XX, YY, XG, YG, BREF, ZZ, ITEST

DATA EPS/1.E-78/

XP = X0BREF*XX
YP = Y0BREF*YY
ZFUN = B1*B2*XPMXI*YP
DZDX = B2
DO 1 I=1,N
XPMXI = XP - X(I)
R2 = XPMXI*XPMXI + (YP-Y(I))*2
R2PEPS = R2 / EPS
XXLOG = ALOG1 R2PEPS
ZFUN = ZFUN + A(I)*R2*XXLOG
DZDX = DZDX + 2.0*A(I)*XPMXI*XXLOG + R2/R2PEPS
1 CONTINUE
RETURN
END

END OF JOB.

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