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>Computer Program Flow</td>
</tr>
<tr>
<td>2</td>
<td>External File Structure</td>
</tr>
<tr>
<td>3</td>
<td>Program Size Limitations</td>
</tr>
<tr>
<td>4</td>
<td>Convergence Criteria</td>
</tr>
<tr>
<td>5</td>
<td>Subroutine Write-Ups</td>
</tr>
<tr>
<td>6</td>
<td>Source Program Listing</td>
</tr>
</tbody>
</table>
INTRODUCTION

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

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

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

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

Mesh Generation Phase

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

Solution Phase

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

The downwash is generated for each mode by using DIAG and ITRATE. Wake effects are iterated if present. The output is displayed and the program now cycles to the next case.
ENTRY

DEFINE ALL WORK SPACE
DEFINE ALL CONTROL VARIABLES
DEFINE LOGICAL I/O UNITS
READ AND PRINT TITLE

CALL INPI
CALL WING IN
CALL GRID IN
CALL PT GRID
CALL SONS

CALL LOOP W

TAPE 15
TAPE 8

INPUT
OUTPUT

TAPE 15 WAKE EFFECTS COEFFICIENTS
TAPE 8 DOWN WASH COEFFICIENTS

YES

RESTART?

GENERATE DOWN WASH COEFFICIENT MATRICES FOR ALL FREQUENCIES

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

CONVERGENCE?

YES

DISPLAY OUTPUT
GO TO NEXT CASE
END OF JOB

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

Figure 1. Computer Program Flow (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.
Section 4
CONVERGENCE CRITERIA

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

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

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

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

Subroutine Write-Ups

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

<table>
<thead>
<tr>
<th>NAME</th>
<th>VARIABLES</th>
<th>DEFINITIONS</th>
<th>SUBROUTINE/REFERENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASIC</td>
<td>MACH</td>
<td>Mach Number (\sqrt{\text{MACH}^2-1})</td>
<td>FGEN, MECH, MESHCL, WINTGR</td>
</tr>
<tr>
<td></td>
<td>BEETA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CEPS</td>
<td>EPS</td>
<td>Convergence criteria for ITRATE, IRELE</td>
<td>IRELE, ITRATE</td>
</tr>
<tr>
<td>EEW</td>
<td>XUP,YUP</td>
<td>X and Y coordinates of the upper end of the influencing line</td>
<td>WINTGR, WVINT</td>
</tr>
<tr>
<td></td>
<td>XLW,YLW</td>
<td>X and Y coordinates of the lower end of the influencing line</td>
<td></td>
</tr>
<tr>
<td>EREØNE</td>
<td></td>
<td>Logical variable set to .TRUE. if a line (in the right hand sense) has influence on a given receiving point</td>
<td></td>
</tr>
<tr>
<td>EMIRRO</td>
<td></td>
<td>Logical variable set equal to .TRUE. if the mirror image of the line has influence on a given receiving point</td>
<td></td>
</tr>
<tr>
<td>RR</td>
<td></td>
<td>Average hyperbolic radius of the endpoints of an influencing line with respect to a given receiving point</td>
<td></td>
</tr>
<tr>
<td>RL</td>
<td></td>
<td>Average hyperbolic radius of the mirror image of the endpoints of an influencing line with respect to a given receiving point</td>
<td></td>
</tr>
<tr>
<td>RTR</td>
<td></td>
<td>The product of the hyperbolic radii at the endpoints of an influencing line with respect to a given receiving point</td>
<td></td>
</tr>
<tr>
<td>RTL</td>
<td></td>
<td>The product of the hyperbolic radii at the endpoints of the mirror image of an influencing line with respect to a given receiving point</td>
<td></td>
</tr>
<tr>
<td>NAME</td>
<td>VARIABLES</td>
<td>DEFINITIONS</td>
<td>SUBROUTINE REFERENCE</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>FQ1</td>
<td>NFREQ</td>
<td>Number of frequencies that are input for the current run</td>
<td>MAIN,FGEN,FREQW,LOOPW,WAKET,WINTGR</td>
</tr>
<tr>
<td>CF</td>
<td></td>
<td>An array, a list of 30 frequency coefficients for each frequency</td>
<td></td>
</tr>
<tr>
<td>FREQ</td>
<td></td>
<td>A list of the frequencies input</td>
<td></td>
</tr>
<tr>
<td>FREQP</td>
<td></td>
<td>A list of modified frequencies ( FREQP(I) = FREQ(I) \times MACH/BEETA )</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,</td>
</tr>
<tr>
<td></td>
<td>KST12</td>
<td></td>
<td>WTEPHT</td>
</tr>
<tr>
<td>MCOM</td>
<td>CFTABLE</td>
<td>Table of equations coefficients for mode polynomial</td>
<td>RDMODE</td>
</tr>
<tr>
<td>MODEF</td>
<td>EM</td>
<td>Storage for modal values for a single point</td>
<td>MFUN, RDMODE</td>
</tr>
<tr>
<td></td>
<td>DM</td>
<td>Storage for values of the derivative of the modes for a single point</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td>( X ) coordinate of the point</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td></td>
<td>( Y ) coordinate of the point</td>
<td></td>
</tr>
<tr>
<td>J1</td>
<td></td>
<td>Wing number that the point is on</td>
<td></td>
</tr>
<tr>
<td>J2</td>
<td></td>
<td>Section of WING J1 that the point is on</td>
<td></td>
</tr>
<tr>
<td>NEXTCS</td>
<td>IFLUSH</td>
<td>Set equal to if an error is found that is serious enough to cause termination of run</td>
<td>MAIN, INPI, WINTGR</td>
</tr>
<tr>
<td>NAME</td>
<td>VARIABLES</td>
<td>DEFINITIONS</td>
<td>SUBROUTINE REFERENCE</td>
</tr>
<tr>
<td>----------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>PARAM</td>
<td>NWING</td>
<td>Total number elements in structure</td>
<td>FREQW</td>
</tr>
<tr>
<td>RRLL</td>
<td>TVW</td>
<td>Transformation matrix to change the velocity influence coefficients from the receiving point to the influencing point</td>
<td>LØØPW, RTØI, WINTGR</td>
</tr>
<tr>
<td>YRO, YLO</td>
<td></td>
<td>Y coordinate of the receiving element in the system of the influencing element</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>YLO is in the system of the mirror image of the influencing element</td>
<td></td>
</tr>
<tr>
<td>ZR</td>
<td></td>
<td>Z coordinate of the receiving element in the system of the influencing element</td>
<td></td>
</tr>
<tr>
<td>ZRZR</td>
<td></td>
<td>ZR*ZR</td>
<td></td>
</tr>
<tr>
<td>ZL</td>
<td></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>ZLZL</td>
<td></td>
<td>ZL*ZL</td>
<td></td>
</tr>
<tr>
<td>TAPE</td>
<td>I9, I10,</td>
<td>File numbers set up by block data subprogram</td>
<td>MAIN, DIAG, ITRATE,</td>
</tr>
<tr>
<td></td>
<td>I11, I12,</td>
<td></td>
<td>LØØPW, TIMOUT, WRITEØ</td>
</tr>
<tr>
<td></td>
<td>I13, I14,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>J9, J15</td>
<td></td>
<td></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>PRINT</td>
<td></td>
<td>Print control</td>
<td></td>
</tr>
<tr>
<td>NWBLOCK</td>
<td></td>
<td>Number blocks for ITRATE</td>
<td></td>
</tr>
<tr>
<td>NWROW</td>
<td></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, WINTGR</td>
</tr>
<tr>
<td>TREDGE</td>
<td></td>
<td>Trailing edge indicator</td>
<td></td>
</tr>
<tr>
<td>WAKE1</td>
<td></td>
<td>Wake indicator</td>
<td></td>
</tr>
<tr>
<td>WAKE</td>
<td></td>
<td>Wake indicator</td>
<td></td>
</tr>
<tr>
<td>WAKENZ</td>
<td></td>
<td>Wake indicator for non-zero term</td>
<td></td>
</tr>
</tbody>
</table>

-5.4-
<table>
<thead>
<tr>
<th>NAME</th>
<th>VARIABLES</th>
<th>DEFINITIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>WV1</td>
<td>NTERM</td>
<td>Number of terms taken in the finite series approximation of the velocity influence expression</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WINTGR, WVINT</td>
</tr>
<tr>
<td>V(30)</td>
<td></td>
<td>Array of sidewash contributions for up to thirty terms</td>
</tr>
<tr>
<td>W(30)</td>
<td></td>
<td>Array of upwash contributions for up to thirty terms</td>
</tr>
<tr>
<td>WW1</td>
<td>XO,Y0,ZD</td>
<td>The X, Y and Z coordinates of a receiving point</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LOOPFW, WVINT, WINTGR</td>
</tr>
<tr>
<td>ZDZD</td>
<td></td>
<td>ZD*ZD</td>
</tr>
<tr>
<td>SYMK</td>
<td></td>
<td>Symmetry code of the wing that the influencing element is in</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>
</tr>
<tr>
<td>XYSCAL</td>
<td>XX</td>
<td>X coordinate of a point</td>
</tr>
<tr>
<td></td>
<td>YY</td>
<td>Y coordinate of a point</td>
</tr>
<tr>
<td></td>
<td>XO</td>
<td>Offset value in X direction currently taken to be 0.0.</td>
</tr>
<tr>
<td></td>
<td>YO</td>
<td>Offset value in Y direction, currently taken to be 0.0.</td>
</tr>
<tr>
<td></td>
<td>BREF</td>
<td>Scalar transformation value, currently set to 1.0</td>
</tr>
<tr>
<td></td>
<td>ZZZZ</td>
<td>Value of first mode input</td>
</tr>
<tr>
<td></td>
<td>ITEST</td>
<td>Set equal to 1 if mode is a constant for the entire grid</td>
</tr>
<tr>
<td>ZFDZIO</td>
<td>I5, I6</td>
<td>Input unit numbers, Output unit numbers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAIN, READAB, READXY</td>
</tr>
<tr>
<td>NAME</td>
<td>VARIABLES</td>
<td>DEFINITIONS</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>ZFUNNY</td>
<td>N</td>
<td>Number points on structural grid for spline</td>
</tr>
<tr>
<td></td>
<td>IERF</td>
<td>Error indicator number</td>
</tr>
<tr>
<td></td>
<td>B1, B2, B3</td>
<td>Spline constants</td>
</tr>
<tr>
<td></td>
<td>NFUNMX</td>
<td>Maximum N value</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>CABSO</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>
</tbody>
</table>
### Subroutine Write-Ups (contd)

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<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 113 and 114</td>
<td>5.89</td>
</tr>
<tr>
<td>WRITEO</td>
<td>Write W coefficients on file 11</td>
<td>5.90</td>
</tr>
<tr>
<td>Writz</td>
<td>Write W coefficients in file 10</td>
<td>5.91</td>
</tr>
<tr>
<td>Writzw</td>
<td>Write Wake effects on file 12</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
   - EFPL0T
   - PTGRID
   - INP1
   - OUTP1
   - GETTIM
   - RDMUDE
   - MESH
   - OUTP2
   - GRIDIN
   - RSTART
   - PHIL
   - OUTP3
   - ITRATE
   - SONSPT
   - TIMEOUT
   - WINGIN
   - WTEPHT
   See Section 6 for the function of each one of these subroutines.

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

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

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

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

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

4. **Definitions:**
   1. **ITERAT** \( \text{ITMAX}=10 = \text{maximum no. iterations for solution} \)
   2. **MXSTOR**
      - \( \text{NWINGS} = \text{no. wings} = 4 \)
      - \( \text{NSECTN} = \text{no. sections} = 3 \)
      - \( \text{NMODES} = \text{no. nodes} = 10 \)
      - \( \text{NSPANS} = \text{no. spans} = 100 \)
      - \( \text{NCORNR} = \text{no. corner pts.} = 8 \)
      - \( \text{NFRQUN} = \text{no. frequencies} = 12 \)
      - \( \text{NEPSPN} = \text{no. elements/span} = 70 \)
      - \( \text{NMXFDZ} = \text{no. mode storage} = 300 \)
   3. **TAPE**
      - \( \text{I9} = 8 \)
      - \( \text{I10} = 10 \)
      - \( \text{I11} = 11 \)
      - \( \text{I12} = 12 \)
      - \( \text{I13} = 13 \)
II4 = 14
J9 = 9
II5 = 15

See Section 2 for file descriptions.

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

5. Calling Sequence: None
1. **Function Name:** CABSO

2. **Purpose:**
   Compute the 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 19
   - NNCH: row number of trailing edge elements
   - MC: maximum dimension of incore matrix
   - NET: number of elements in total system
   - FRQ: frequency value
   - NSPT: No. trailing edges

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

6. **Subroutine User:**
   - Main program

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

8. **Files Used:**
   - $I_9$ = (input) file containing $W$ coefficients, for all frequencies
   - $I_{10}$ = diagonal block, decomposed for 1 frequency (output)
   - $I_{11}$ = off diagonal terms (output)
   - $I_{15}$ = (input) wake coefficients
   - $I_{13}$ = wake effects for 1 frequency in compressed form
1. **Subroutine Name:** ECHO

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

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

4. **Input Arguments:** None

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

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

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

7. **Error Returns:** None

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

9. **Input Tapes:** None
10. Output Tapes:
   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:** EFPILOT

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 EFPILOT.

4. **Input Arguments:**
   A dummy argument, not used, but present to keep the call to the routine identical to the IBM-CALCOMP version of EFPILOT. (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 EFPILOT(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:
   \[
   KAPPA = \frac{K \times M}{\sqrt{M^2 - 1}}
   \]
   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:
   \[
   C_1 = 1.0 \\
   C_J = -C_{N^*} \left( \frac{KAPPA^{*2}}{(2*N)(2*N-1)} \right)
   \]
   for \( J > 1 \), where \( N = J - 1 \).
   Currently a maximum of 30 terms are taken \( (J=30) \) for each frequency. A check on the exponent is made to avoid an exponent underflow. A scalar, XUNDER, is defined to be -77.5 on the IBM machine and -292.0 on the CDC machine. This is close to the largest negative exponent allowed each machine.

4. **Input Arguments**: None

5. **Output Arguments**: None

6. **Common Blocks Used**:
   /BASIC/ and /FQL/
7. **Error Returns**: None
8. **Calling Sequence**: Call FGEN
9. **Input Tapes**: None
10. **Output Tapes**: None
11. **Scratch Tapes**: None
12. **Storage Required**: 263 words
13. **Subroutine User**: MAIN
14. **Subroutine Required**: None
15. **Remarks**: None
1. **Subroutine Name:** FREQW

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

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

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

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

6. **Subroutine User:** DIAG

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

2. **Purpose:**

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

3. **Equations and Procedure:**

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

4. **Input Arguments:**

   ITIME Integer array of length 8.

5. **Output Argument:**

   ITIME Integer array of length 8, time is put in second location.

6. **Common Blocks Used:** None

7. **Error Returns:** None

8. **Calling Sequence:**

   CALL GETTIM (ITIME)

9. **Input Tapes:** None

10. **Output Tapes:** None

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

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

14. **Subroutine User:** MAIN

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

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

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

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

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

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,9I2,4E12.0).
   
   a) A check is made to be sure that it is a RUN card.

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

   c) The second integer field contains the plot request code. If there is a 'l' 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.

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

ERROR $\text{SET TO .TRUE if error condition is encountered.}$

EAR$\phi$ Element aspect ratio

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

6. **Common Blocks Used:**

```
/NEXTCS/
```

IFLU$\phi$ 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 IFLU$\phi$ 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(15,16,TITLE,ITRUN,MACH,REFLEN,XP,XPIN,PLOTR,IWTE, EAR0,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/
   /W1/

7. **Error Returns:** None

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

9. **Input Tapes:** None

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

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

11. Scratch Tapes: None
12. Storage Required: 1110 words
13. Subroutines Required: RTOI and WINTRGR
14. Subroutine User: MAIN
15. Remarks: Domain of Effective Far Field Elements - Contribution to the velocity component at a receiving point, from far field elements (i.e. when \( \text{is large} \)) is small. Computational economy can be regained by truncating the domain of influence, when the order of the magnitude of the influence coefficient \( W_{ij} \) is less than say \( (E) \) of \( W_{ii} \). The parameter \( E \) is fixed in the program at .005 and called PERC. The programmer can update this quality by changing the variable PERC in the BLOCK DATA routine.
BEGIN
LOOP ON RECEIVING ELEMENTS

FIND INFLUENCE COEFFICIENTS FOR RECEIVING ELEMENT ON ITSELF

LOOP ON INFLUENCING WING AND SECTIONS

DOES SECTION INFLUENCE THE SECTION OF THE RECEIVING ELEMENT

NO
UPDATE ELEMENT COUNTERS AND SKIP TO NEXT INFLUENCING SECTION

YES
FIND TRANSFORMATION MATRIX FOR THIS SECTION

LOOP ON SPANS OF THIS SECTION

LOOP ON ELEMENTS OF SPAN FROM BOTTOM TO TOP

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

Flow Chart of Subroutine LOOPW
t=

\[ y \]

\[ A \]

TEST 4

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.
   - `NSP`: 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,IFTABLE,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,LINE,LMAX,NNCH,NE,TITLE,NSPT)

9. Input Tapes: None

10. Output Tapes: None

11. Scratch Tapes: None

12. Storage Required: 650 words

13. Subroutine User: MAIN

14. Subroutine Required: None

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

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

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

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

5. **Output Arguments:** None

6. **Common Blocks Used:** None

7. **Error Returns:** None

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

9. **Input Tapes:** None

10. **Output Tapes:** None

11. **Scratch Tapes:** None

12. **Storage Required:** 260 words

13. **Subroutine User:** MAIN

14. **Subroutine Required:** None

15. **Remarks:** None
1. Subroutine Name: OUTP3
2. Purpose:
Write the table of GENERALIZED AERODYNAMIC COEFFICIENT MATRIX
IN AGARD DEFINITION.
3. Procedure:
Information is printed out in tabular form.
4. Input Arguments:
   I6     Logical unit number of printer
   MACH   Mach number
   FREQ   Frequency
   Q      Aerodynamic coefficient matrix
   LINE   Current line
   LMAX   Total number of lines allowed on page
   TITLE  Print title information, 4 characters per word, 56 characters in total
5. Output Arguments: None
6. Common Blocks Used: None
7. Error Returns: None
8. Calling Sequence:
   Call OUTP3 (I6,MACH,GREQ,JMODE,Q,LINE,LMAX,TITLE)
9. Input Tapes: None
10. Output Tapes: None
11. Scratch Tapes: None
12. Storage Required: 430 words
13. Subroutine User: MAIN
14. Subroutine Required: None
15. Remarks: None
1. **Subroutine Name:** PHIL

2. **Purpose:** Integration of velocity potentials.

3. **Equations and Procedure:**
The velocity potentials are integrated separately for each span. The equation and method is given in detail in Reference #1. This routine also determines the convergence of the velocity potentials when wake effect is considered.

4. **Input Arguments:**
   - **XLE** X coordinate of the center of the leading edge line for the span.
   - **XCEN** X coordinates of the center of the elements of the span.
   - **DPDX** Functional values to be integrated (COMPLEX).
   - **NEPS** The number of elements per span.
   - **A** Complex constant used as exponent.
     - \( A = (0.0, K'M) \).
   - **CXP** Complex expotentials taken at the center of the element. \( CXP(I) = CEXP(0.0, XCENLI) * K'M) \).

5. **Output Arguments:**
   - **PHI** Velocity potential for the span.
   - **PHIW** This array is used as both input and output to test for wake convergence. On input it is the velocity potential of the last iteration. On output it is identical to PHI.

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

7. **Error Returns:** None
8. **Calling Sequence:**

Call PHIL(XLE, XCEN, DPDX, NEPS, PHI, A, CXP, PHIW)

9. **Input Tapes:** None

10. **Output Tapes:** None

11. **Scratch Tapes:** None

12. **Storage Required:** 914 words

13. **Subroutines Required:** None

14. **Subroutine User:** MAIN

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

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

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

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

5. **Output Arguments:** None

6. **Common 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.
1. **Subroutine Name**: PLOTGD

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

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

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

5. **Output Arguments**: None

6. **Common Blocks Used**: None

7. **Error Returns**: None

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

9. **Input Tapes**: None

10. **Output Tapes**: None

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

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

14. Subroutine User: MAIN

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

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

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

4. **Input Arguments:**
   - I6 Logical unit number of output device.
   - TITLE Array containing title information (56 characters, 14 words, 4 characters each).
   - MACH Mach number
   - NWING Number of wings
   - NZECT Number of sections in each wing.
   - NZIP Number of spans in section I of wing J-NZIP(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.
   - ZZECT Z coordinates of each section.
   - REFLEN Input reference length.
   - BEETA BEETA=SQR(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 (L6,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 IL3

3. **Input Arguments:**
   IL3 = Tape number
   NET = number of elements

4. **Output Arguments:**
   DETADX = array on tape IL3
   ETA = array on tape IL3

5. **Scratch Tapes:** IL3

6. **Subroutine User:**
   Main program

7. **Calling Sequence:**
   Call RDETA (IL3, DETADX, ETA, NET)
1. **Subroutine Name:** RDMODE

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

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

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

5. **Output Arguments:**

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

6. **Common Blocks Used:**

/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,NSECT,NE,XCEN,XYZ,EEETA,REPLEN,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 /ZPDZI0/

7. **Error Returns:** None

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

15. **Remarks:**

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

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

3. Equations and Procedure:
   This reads in the structural grid for the spline routine.
   This is a modified version of subroutine ZFUN. It reads
   in the number of points in the structural grid and the X
   and Y coordinates of the structural grid, exactly as it was
   punched from Program 831. (831 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. READZXY 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, 115.

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**

/NEXTCS/  (see Error Returns)

7. **Error Returns**

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

8. **Calling Sequence:**

Call RSTART(MACH,NET,NFREQ,FREQ,REFLEN,IWTE,I8,I15,I6,FREQR, XYZ,AREA).

9. **Input Tapes:**

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

10. **Output Tapes:**  None

11. **Scratch Tapes:**  None

12. **Storage Required:**  450 words

13. **Subroutines Required:**  None

14. **Subroutine User:**  MAIN

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

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

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

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

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

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

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

9. **Input Tapes:** None

10. **Output Tapes:** None

11. **Scratch Tapes:** None

12. **Storage Required:** 200 words

13. **Subroutines Required:** None

14. **Subroutine User:** LOOPW

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

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

3. **Equations and Procedure:**
   Back substitution based on the factored form of the coefficient matrix. See Algorithm 423, "Collected Algorithms from CACM", by Cleve Moler.

4. **Input Arguments:**
   - \( N \) Order of the Matrix \( A \)
   - \( NDIM \) First dimension of a declared in calling program. If \( A \) is singly subscripted in calling program, 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 $ISON(K1,K2)$ is determined such that $ISON(K1,K2)=1$ if section $K2$ has influence of section $K1$, $ISON(K1,K2)=0$ if section $K2$ has no influence on section $K1$.

4. **Input Arguments:**
- NWING  Number of wings
- NSECT  Number os 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,ISON)

9. **Input Tapes:**  None

10. **Output Tapes:**  None
11. **Scratch Tapes:** None
12. **Storage Required:** 430 words
13. **Subroutine User:** MAIN
14. **Subroutine Required:** None
15. **Remarks:** None
1. **Subroutine Name:** XONXPT

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

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

4. **Input Arguments:**
   IXONX See write up, (Equations and Procedure Section) for Subroutine IXONX.
   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 IXONXPT(IXONX, 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 I12 for 1 frequency

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

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

5. **Calling Sequence:**
   Call WAKET (I12,AROW,WROW,IF,LRECLW,NSPT,I8,IZ,KL,NNCH,FRQ)

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

7. **Subroutines Used:** WRITZW

8. **Subroutine User:** DIAG

9. **Calling Sequence:**
   Call DIAG (WROW,IF,LRECL,NNCH,MC,D,NET,AROW,IZ,IP,NSPT,FRQ,CWROW,CAROW)

-5.78-
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  
- CORNY  
- CORNZ  
- NWING  
- ICN  
- I6

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 preceeded by the statement ERROR IN READING INPUT.)
Messages a through f have to do with errors on the WING data card, and are preceded by the message:

WING DATA CARD IS INCORRECT.

a) DATA FOR WING - WAS READ WHEN DATA FOR WING - WAS EXPECTED BY THE PROGRAM.

(WING data cards must appear in consecutive order in the deck. Each WING data card is the first card of a WING data group.)

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

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

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

e) NUMBER OF CORNER POINTS, __ IS NOT CONSISTANT WITH THE NUMBER OF SECTION __ FOR WING __.

(A wing with 1 section may have only 3 or 4 corner points. A wing with 2 sections, 5 or 6 corner points. A wing with 3 sections, 7 or 8 corner points.)

f) SYMMETRY CODE FOR WING __ IS INPUT AS __.

SYMMETRY CODE MAY ONLY EQUAL -1, 0, OR 1.

g) "CORNER" DATA CARD IS MISSING FOR WING ____.

(The CORNER cards defining a wing must immediately follow the WING card for that wing.)

Messages h through l have to do with errors on the LINE data card.
They are preceded by the following message

ERROR ON CONTROL "LINE" DATA CARD FOR WING ____.

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

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

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

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

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

m) MODE DATA IS MISSING OR OUT OF ORDER.

n) FREQUENCY DATA IS MISSING OR OUT OF ORDER.

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

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

8. Calling Sequence:
Call WINGIN( I5, I6, NWING, NSECT, AR, NSP, CORNX, CORNY, CONRZ,
SYM, ICL, CL, ERROR, BETA, 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/
   /WW1/

-5.87-
7. Error Returns: None
8. Calling Sequence:
   Call WINTGR(X,Y,WR0W)
9. Input Tapes: None
10. Output Tapes: None
11. Scratch Tapes: None
12. Storage Required: 1630 words
13. Subroutine Required: WVINT
14. Subroutines User: L00PW
15. Remarks: None
1. **Subroutine Name**: WRETA

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

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

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

5. **Scratch Tapes**: I13

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

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

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

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

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

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

7. Subroutines Required:
- WRITZ

8. Subroutine User: DIAG

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

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

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

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

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

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

7. **Subroutine User:** WRITEO

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

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

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

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

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

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

7. **Subroutine User:** WAKET

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

2. Purpose:
Generate wake effect in computation of \( \frac{dn}{dt} \).

3. Equations and Procedures:
\[
\frac{dn}{dt} = \frac{dn}{dt} - \mathbf{W}_{TE} \phi
\]

a) \( IZ \) and \( \mathbf{W}_{TE} \) 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) \( \mathbf{W}_{TE} \) is post multiplied by \( \phi \) elements and subtracted
   from \( \mathbf{DEDT} \).
d) The result from step c) is stored back on \( \mathbf{DEDT} \).

Steps a), b), c), and d) are repeated for each element.

4. Input Arguments:

\( \mathbf{DEDT} \) = \( \frac{dh}{dt} \) array
\( \text{NET} \) = number of elements
\( \Phi_{IW} \) = \( \phi \) array of length \( \text{NET} \)
\( I12 \) = tape containing \( \mathbf{W}_{TE} \) arrays
\( \mathbf{W}_{TE} \) = \( \mathbf{W}_{TE} \) work array - only nonzero terms

5. Output Arguments:

\( \mathbf{DEDT} \) = revised \( \frac{dn}{dt} \)

6. Error Return: None

7. Calling Sequence:

Call WTEPHT(\( \mathbf{DEDT}, \text{NET}, \Phi_{IW}, IZ, I12, \mathbf{W}_{TE} \))

8. Scratch Tapes:

\( I12 \) = tape containing \( \text{NET} \) records.
Each record is of form:
\( \text{KEL}, \text{NNZ}, IZ, \mathbf{W}_{TE} \).
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, DFDZ, X, Y, A)

9. **Input Tapes:** None

10. **Output Tapes:** None

11. **Scratch Tapes:** None
12. **Storage Required:** 151 words
13. **Subroutines Required:** None
14. **Subroutine User:** RDMODE
15. **Remarks:**

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

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

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Deck No.</th>
<th>Name</th>
<th>Deck No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAIN</td>
<td>1</td>
<td>PLOTGD</td>
<td>26</td>
</tr>
<tr>
<td>BLOCK</td>
<td>2</td>
<td>PTGRID</td>
<td>27</td>
</tr>
<tr>
<td>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>READXY</td>
<td>31</td>
</tr>
<tr>
<td>EFPLOT</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>WRITEO</td>
<td>46</td>
</tr>
<tr>
<td>OUTP2</td>
<td>22</td>
<td>WRTIZ</td>
<td>47</td>
</tr>
<tr>
<td>OUTP3</td>
<td>23</td>
<td>WRTIZW</td>
<td>48</td>
</tr>
<tr>
<td>PHIL</td>
<td>24</td>
<td>WTEPHT</td>
<td>49</td>
</tr>
<tr>
<td>PLOT</td>
<td>25</td>
<td>WVINT</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ZMDZ</td>
<td>51</td>
</tr>
</tbody>
</table>
DATA FILE DECK 01

LINE 1 C MAIN
LINE 2 C CONSTANT POTENTIAL GRADIENT, ME, UU
LINE 3 C CHANGE STORAGE TO 300, 70, 100 FROM 200, 43, 50
LINE 4 C COMPLEX CARG(100)
LINE 5 C COMPLEX GE, PRES(300)
LINE 6 C COMPLEX GFREQ
LINE 7 C COMPLEX CROW (1200)
LINE 8 C COMPLEX CXM
LINE 9 C COMPLEX CXP(300)
LINE 10 C COMPLEX DCOMP(901)
LINE 11 C COMPLEX JCOMP(10,10)
LINE 12 C COMPLEX PHIO(70)
LINE 13 C COMPLEX PHIA(300)
LINE 14 C COMPLEX SWORKC(1671)
LINE 15 C COMPLEX ESUKC(11)
LINE 16 C COMPLEX GT(300)
LINE 17 C COMPLEX Q(10,10)
LINE 18 C COMPLEX PHI(70)
LINE 19 C COMPLEX DCOMP(901)
LINE 20 C COMPLEX WM(10,10)
LINE 21 C COMPLEX JCOMP(10,10)
LINE 22 C COMPLEX SWORK(1200)
LINE 23 C COMPLEX WM(300)
LINE 24 C COMPLEX WM(300)
LINE 25 C REAL IBUFF
LINE 26 C REAL VACCH
LINE 27 C INTEGER PRINT
LINE 28 C INTEGER TAP
LINE 29 C LOGICAL ERROR
LINE 30 C LOGICAL PLOT
LINE 31 C LOGICAL PLOT
LINE 32 C LOGICAL TEKDO
LINE 33 C LOGICAL TREDGE
LINE 34 C LOGICAL WAKE, WAKE1, WAKENZ
LINE 35 C DIMENSION AR(4), ISONS(12,12)
LINE 36 C DIMENSION AREA(300)
LINE 37 C DIMENSION AROW(300)
LINE 38 C DIMENSION CWORK(4529)
LINE 39 C DIMENSION DCOMP(3901)
LINE 40 C DIMENSION D(60,60), IP(601), DNWR(501), DENTL(60), RNK(60), RNK(60)
LINE 41 C DIMENSION OR(300), UI(300), DPPX(300), DEDT(300)
LINE 42 C DIMENSION ECWORK(1)
LINE 43 C DIMENSION EDWKR(1)
LINE 44 C DIMENSION EMISC(1)
LINE 45 C DIMENSION ESMARK(1)
LINE 46 C DIMENSION ETA(300), OCMDX(300)
LINE 47 C DIMENSION ECMARK(1)
LINE 48 C DIMENSION FERROR(12)
LINE 49 C DIMENSION IBUFF(2000)
LINE 50 C DIMENSION ICL(3,4), CL(4,3,4)
LINE 51 C DIMENSION IO(9), FO(4)
LINE 52 C DIMENSION IP(4)
LINE 53 C DIMENSION IZ(300)
LINE 54 C DIMENSION FTIME(1)
LINE 55 C DIMENSION NE(3,4)
LINE 56 C DIMENSION NSEC(4), CORX(8,4), CORY(8,4), CORKR(8,4), NS(3,4)
LINE 57 DIMENSION NSP(4)
LINE 58 DIMENSION SW(100), XLE(100), WNC(100)
LINE 59 DIMENSION SW(100)
LINE 60 DIMENSION SW(120)
LINE 61 DIMENSION SY(4)
LINE 62 DIMENSION TITLE(24)
LINE 63 DIMENSION TS(3,4), ZSEC(3,4)
LINE 64 DIMENSION W(300,10), W(300,10)
LINE 65 DIMENSION WORK(1300,1)
LINE 66 DIMENSION W(300,3600)
LINE 67 DIMENSION XCGN(300)
LINE 68 DIMENSION XYZ(1600)
LINE 69 DIMENSION XZFDZ(300,3)
LINE 70 COMM / BASIC MACHINE, BEATA, KAPPA, IPRINT(15).
LINE 71 COMM / WORK / NPNIU(112,301), FREQ(121), FREO(121), FREO(121), FREO(121), FREST, NTMAX(100710)
LINE 72 COMM / TG / IGMESS
LINE 73 COMM / WORK / NANGS, NSEC, NMODES, NSPANS, NCORN, NFREQU, NEPSN
LINE 74 1, NSFDZ
LINE 75 COMM / MAXIMUM NANGS = 4
LINE 76 COMM / MAXIMUM NSCIONS = 10
LINE 77 COMM / MAXIMUM NMODES = 10
LINE 78 COMM / MAXIMUM NSPANS = 50
LINE 79 COMM / MAXIMUM NCORNER PTS = 8
LINE 80 COMM / MAXIMUM NFREQUENCIES = 12
LINE 81 COMM / MAXIMUM NELEMENTS / SPAN = 40
LINE 82 COMM / NEXIGS / IFUISH
LINE 83 COMM / PARAMS / NET
LINE 84 COMM / PEREPS / PERC
LINE 85 COMM / TAPE / I1,110,111,112,113,114,115
LINE 86 COMM / TIME / ITIME
LINE 87 COMM / VELCOM / NMAX, PRINT, NMAX, NWORK(10)
LINE 88 COMM / WAKE / ITIM, ITIM, ICNVGN, EPSW, JCNVGN, LRAW, LINT, I3TE, NKE, WAKE
LINE 89 COMM / WORK / HES(12), ITRES, WAKE, WAKE
LINE 90 COMM / ZFDZ(1)/ JS, JS
LINE 91 COMM / MFUNNY / NIERF, 81, 12, 83, NFUNMX
LINE 92 COMM / WORK(112) / CAROW(11)
LINE 93 COMM / WORK(112) / CAROW(11)
LINE 94 COMM / WORK(112) / CAROW(11)
LINE 95 COMM / WORK(112) / CAROW(11)
LINE 96 COMM / WORK(112) / CAROW(11)
LINE 97 COMM / WORK(112) / CAROW(11)
LINE 98 COMM / WORK(112) / CAROW(11)
LINE 99 COMM / WORK(112) / CAROW(11)
LINE 100 COMM / WORK(112) / CAROW(11)
LINE 101 COMM / WORK(112) / CAROW(11)
LINE 102 COMM / WORK(112) / CAROW(11)
LINE 103 COMM / WORK(112) / CAROW(11)
LINE 104 COMM / WORK(112) / CAROW(11)
LINE 105 COMM / WORK(112) / CAROW(11)
LINE 106 COMM / WORK(112) / CAROW(11)
LINE 107 COMM / WORK(112) / CAROW(11)
LINE 108 COMM / WORK(112) / CAROW(11)
LINE 109 COMM / WORK(112) / CAROW(11)
LINE 110 COMM / WORK(112) / CAROW(11)
LINE 111 COMM / WORK(112) / CAROW(11)
LINE 112 COMM / WORK(112) / CAROW(11)
LINE 113 COMM / WORK(112) / CAROW(11)
LINE 114 COMM / WORK(112) / CAROW(11)
LINE 115 COMM / WORK(112) / CAROW(11)
LINE 116 COMM / WORK(112) / CAROW(11)
LINE 117 COMM / WORK(112) / CAROW(11)
LINE 118 COMM / WORK(112) / CAROW(11)

ORIGINAL PAGE OF POOR QUALITY.
Line 119 EQUIVALENCE ( SWORK(1), WORKM(1))
Line 120 EQUIVALENCE ( SWORK(12), SWORKC(1))
Line 121 EQUIVALENCE ( SWORK(1), OR(1))
Line 122 EQUIVALENCE ( SWORKC(101), OR(1))
Line 123 EQUIVALENCE ( SWORKC(171), PHI(1))
Line 124 EQUIVALENCE ( SWORKC(471), DEN(1))
Line 125 EQUIVALENCE ( SWORK(771), OR(1))
Line 126 EQUIVALENCE ( SWORKC(1071), CXP(1))
Line 127 EQUIVALENCE ( SWORKC(1371), OR(1))
Line 128 EQUIVALENCE ( SWORKC(4671), ESURKC(1))
Line 129 EQUIVALENCE ( SWORKR(1), DR(1))
Line 130 EQUIVALENCE ( SWORKR(301), OR(1))
Line 131 EQUIVALENCE ( SWORKR(601), IZ(1))
Line 132 EQUIVALENCE ( SWORKR(901), ARGW(1))
Line 133 EQUIVALENCE ( SWORKR(1201), ESUR(1))
Line 134 EQUIVALENCE ( TAPE(9), ESURK(1))
Line 135 EQUIVALENCE ( KMM(1), AR(1))
Line 136 EQUIVALENCE ( WKMM(5), UL(1))
Line 137 EQUIVALENCE ( WKMM(53), FD(1))
Line 138 EQUIVALENCE ( WKMM(57), ID(1))
Line 139 EQUIVALENCE ( WKMM(166), ICL(1))
Line 140 EQUIVALENCE ( WKMM(78), TRS(1))
Line 141 EQUIVALENCE ( WKMM(126), CORNX(1))
Line 142 EQUIVALENCE ( WKMM(158), CORNY(1))
Line 143 EQUIVALENCE ( WKMM(190), CORNZ(1))
Line 144 EQUIVALENCE ( WKMM(222), SNNS(1))
Line 145 EQUIVALENCE ( WKMM(366), SEC(1))
Line 146 EQUIVALENCE ( WKMM(378), EMSC(1))
Line 147 EQUIVALENCE ( WKMM(401), XYZ(1))
Line 148 EQUIVALENCE ( WKMM(2201), WTEROW(1))
Line 149 EQUIVALENCE ( WKMM(4601), WETA(1))
Line 150 EQUIVALENCE ( WKMM(7601), WET(1))
Line 151 EQUIVALENCE ( WKMM(11601), ESURK(1))
Line 152 EQUIVALENCE ( WKMM(11), XYZFDZ(1))
Line 153 EQUIVALENCE ( 13UPF(1), WETA(1))
Line 154 DATA 13/67/
Line 155 DATA 15, 16 / 11, 6 /
Line 156 DATA LMODE / 4MODE /
Line 157 DATA TITLE / 4TITLE /
Line 158 DATA LMAX / 62 /
Line 159 DATA TWIPL / 6, 283185308 /
Line 160 DATA XP / 0.0 /
Line 161 C
Line 162 J5 = 15
Line 163 J6 = 16
Line 164 4 PLOT = .FALSE.
Line 165 CONTINUE
Line 166 ERR = 0
Line 167 CALL ECHM (KONTRL)
Line 168 IF (KJNTVL .EQ. 1) GO TO 10
Line 169 PLOT = .FALSE.
Line 170 IFLUSH = 0
Line 171 NETSV = 0
Line 172 CALL GETTIM (TIME)
Line 173 C
Line 174 C
Line 175 C
Line 176 C
Line 177 C
Line 178 C
Line 179 C
Line 180 C

---6.5---
**Reproducibility of the Original Page is Poor**
LINE 243  2  VSP(I,J), NEI(I,J), NETSV, ICLII(J), CL(I,I,J), EARU  
LINE 244  18  CONTINUE  
LINE 245  C  
LINE 246  VSP = NSPV & NSP(I,J)  
LINE 247  NSPT = NSPT & NSP(I,J)  
LINE 248  NEW = NEW & NEI(I,J)  
LINE 249  C  END LOOP ON SECTIONS OF WING  
LINE 250  19  CONTINUE  
LINE 251  C  
LINE 252  I = 6*NET & 1  
LINE 253  C  JCN  = NUMBER OF CURER POINTS  
LINE 254  C  FINE THE WING  
LINE 255  IF( I(TURE )  
LINE 256  C  GO TO 650  
LINE 257  IF( PLOT ) CALL PLOToD( NEI(I,J), XYZ(NINDEX), NNCH(I,J), NSP(I,J)  
LINE 258  1  NSECT(J), TITLE, J, ZSECT(I,J), AR, MACH )  
LINE 259  C  WRITE( 6, 1014  
LINE 260  C  ) NNCH(1)  
LINE 261  C  NET = NET & NEW  
LINE 262  C  END LOOP ON WINGS  
LINE 263  20  CONTINUE  
LINE 264  C  IF(NPROM .LT. 4) GO TO 660  
LINE 265  C  IF( PLOT ) CALL ENDPLT  
LINE 266  C  NSPT = NSPT - 1  
LINE 267  C  IF(NSPANS .LT. NSPT) GO TO 640  
LINE 268  C  WRITE( 6, 1014  
LINE 269  C  ) NNCH(I), I = 1, NSPT  
LINE 270  C  WRITE( 6, 1015  
LINE 271  C  ) XLE(I), I = 1, NSPT  
LINE 272  C  WRITE( 6, 1016  
LINE 273  C  )  
LINE 274  C  NMAY) = NMAX  
LINE 275  NWBLJK = 0  
LINE 276  ISUM = NNCH(I)  
LINE 277  DO 1200  
LINE 278  18 = 2, NSPT  
LINE 279  ISUM = ISUM  
LINE 280  IDIFF = NNCH(I) - NNCH(I-1)  
LINE 281  ISUM = ISUM & IDIFF  
LINE 282  NWBLIK = NWBLOK & 1  
LINE 283  NWR04(NWBL0K) = ISUM  
LINE 284  ISUM = IDIFF  
LINE 285  CONTINUE  
LINE 286  1020  
LINE 287  NWBL0K = NWBL0K & 1  
LINE 288  CALL PTG10D(16, TITLE, MACH, NWING, NSECT, NSP, NNCH, XYZ, ZSECT,  
LINE 289  1  REFLEN, BETA, XCN )  
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  
LINE 293  C  CALL SONS( NWING, NSECT, CORNY, CORNX, ISONS )  
LINE 294  C  
LINE 295  C  CALL SONSPT( ISONS, NWING, NSECT, 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(16, 9001)
LINE 306 4001 FORMAT(14I10)
LINE 307 C -----------------------------0013050
LINE 308 CALL TIMOUT(MSEC, 4HREAD, 4H END, CPU, 4Houfl, 4HUT, 4HNEER, 4HATE, 4HGRID) 0013060
LINE 309 1 4H )
LINE 310 C -----------------------------0013070
LINE 311 CALL FGEN
LINE 312 LRCL = NET*NPREW 0013080
LINE 313 C -----------------------------0013090
LINE 314 CALL FGEN
LINE 315 C -----------------------------0013100
LINE 316 C IF THERE IS ONLY ONE FREQUENCY, AND IT = 0.0, SET IWT = J 0013110
LINE 317 IWT = 1 0013120
LINE 318 IF( VHFREQ EQ. 1 ) AND FREQ(1) EQ. 0.0 ) IWT = 0 0013130
LINE 319 IWT = NSPT*NPREW 0013140
LINE 320 IWTM = IWT 0013150
LINE 321 IF( IWTM EQ. 1 ) IWTM = 1 0013160
LINE 322 IF( ERROR ) GOTO 60 0013170
LINE 323 IF( ITPUN NE. 1 ) GO TO 60 0013180
LINE 324 C WRITE(6,3002) IWTM, IWT 0013190
LINE 325 C WRITE(16,1021) NTMAX, NCVFL, RKEX, FREEST 0013200
LINE 326 CALL LOUPM ( NET, NWING, XYZ, NSP, SYM, TRS, NNCH, NE) 0013210
LINE 327 1 NSECT, 1 SONS, 16, TAPPB, XCN, AREA, WRO, LREL, 0013220
LINE 328 2 WTKUW, LRWTE, PERC, ERROR, IWT, REFLEN, 0013230
LINE 329 C -----------------------------0013240
LINE 330 CALL GETTIM ( ITIME ) 0013250
LINE 331 CALL TIMOUT(MSEC, 4HREAD, 4H END, CPU, 4Houfl, 4HUT, 4HNEER, 4HATE, 4HGRID) 0013260
LINE 332 1 4HITM 0013270
LINE 333 C -----------------------------0013280
LINE 334 C IF( TEXT ) WRITE(16,1021) NTMAX, NCVFL, RKEX, FREEST 0013290
LINE 335 C WRITE(16,1022) NTMAX, RKEX 0013300
LINE 336 GO TO 110 0013310
LINE 337 C -----------------------------0013320
LINE 338 60 CONTINUE 0013330
LINE 339 C -----------------------------0013340
LINE 340 C IF( ITPUN NE. 3 ) GO TO 110 0013350
LINE 341 C CALL SUBROUTINE TO READ RESTART TAPE 0013360
LINE 342 C -----------------------------0013370
LINE 343 C READ MODE INPUT 0013380
LINE 344 C -----------------------------0013390
LINE 345 C CALL ROUTINE TO READ M NOTES IN UNNORMALIZED STRUCTURAL RF. SYST. 0013400
LINE 346 C -----------------------------0013410
LINE 347 C READ MODE INPUT 0013420
LINE 348 C -----------------------------0013430
LINE 349 C WRITE(16,2006) TITLE 0013440
LINE 350 C -----------------------------0013450
LINE 351 C WRITE(16,2006) TITLE 0013460
LINE 352 C -----------------------------0013470
LINE 353 C IF( ITPUN NE. 3 ) GO TO 110 0013480
LINE 354 C WRITE(16,2006) TITLE 0013490
LINE 355 C -----------------------------0013500
LINE 356 C CALL ROUTINE TO READ M NOTES IN UNNORMALIZED STRUCTURAL RF. SYST. 0013510
LINE 357 C -----------------------------0013520
LINE 358 C CALL ROUTINE TO READ M NOTES IN UNNORMALIZED STRUCTURAL RF. SYST. 0013530
LINE 359 C -----------------------------0013540
LINE 360 C CALL ROUTINE TO READ M NOTES IN UNNORMALIZED STRUCTURAL RF. SYST. 0013550
LINE 361 C -----------------------------0013560
LINE 362 C CALL ROUTINE TO READ M NOTES IN UNNORMALIZED STRUCTURAL RF. SYST. 0013570
LINE 363 C -----------------------------0013580
LINE 364 C CALL ROUTINE TO READ M NOTES IN UNNORMALIZED STRUCTURAL RF. SYST. 0013590
LINE 365 C -----------------------------0013600
LINE 366 1 4H )
LINE 367 C
LINE 368 C
LINE 369 IF( LTRUN .EQ. 3 ) GO TO 132
LINE 370 IF( LTRUN .EQ. 1 ) GO TO 132...
LINE 371 WRITE( 16, 3031 )
LINE 372 IF( I,TRUN .EQ. 1 ) WRITE( 16,3032 )
LINE 373 GO TO 4
LINE 374 132 CONTINUE
LINE 375 IF( NOT. ERROR! ) GO TO 135
LINE 376 WRITE(16,3031)
LINE 377 GO TO 4
LINE 378 135 CONTINUE
LINE 379 CALL MERA (NET,JMODE,OMG,OMQ,NET,13).
LINE 380 CALL MERA (NET,JMODE,OMG,OMQ,NET,14).
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
SET LINE .EQ. LMAX & 1 TO FORCE NEW PAGE WITH FREQ. HEADING
LINE 392 C
LINE 393 C
LINE 394 C
FRQ = FREQ(IF)
LINE 395 C
LINE 396 C
LINE 397 C
LINE 398 C
LINE 399 C
MC = 'X'
LINE 400 C
LINE 401 C
LINE 402 C
LINE 403 C
LINE 404 C
CALL TIMOUT(MSEC,4HREAD,4H AND,4H DEC,4HOMPO,4HOMPE,4H. 4A,
LINE 405 C
LINE 406, C
LINE 407 C
LINE 408 DO 169 J=1,JMODE
LINE 409 C
LINE 410 C
LINE 411 C
LINE 412 C
LINE 413 C
LINE 414 DO 169 I=1,JMODE
LINE 415 C
LINE 416 C
LINE 417 C
LINE 418 C
LINE 419 C
LINE 420 C
LINE 421 DO 283 JM = 1,JMODE
LINE 422 C
LINE 423 C
LINE 424 IF( JM .GT. 1 ) AND. LINE .NE. (LMAX+1) ) LINE=LMAX
LINE 425 C
LINE 426 IF(FRJ .EQ. 0.0) JCNVGW = NET
LINE 427 CALL 4DGA (13,OMAX,OMQ,NET)
LINE 428 DO 197 I=1,NET

---

LOOP ON FREQUENCIES

DO 300 IF=1,NFREQ

DO 391 IF .EQ. LMAX & 1 TO FORCE NEW PAGE WITH FREQ. HEADING

ZERO OUT GENERALIZED COEFFICIENTS (Q).

LOOP ON PRESSURE MODES

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

LOOP ON PRESSURE MODES

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

LOOP ON PRESSURE MODES

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

-6.9-
REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

LINE 429 C CALCULATE DET FOR ONE FREQUENCY AND ONE MODE
LINE 430 DET T(I)=CMPLX(DET A(I), FREQ(I)) + XP(I)
LINE 431 C DET T(I) = CMPLX( DET A(I), J(I)), FKJETA(I,J) )*XP(I)
LINE 432 T_HI W 1 I = 0.0
LINE 433 167 CONTINUE
LINE 434 169 CONTINUE
LINE 435 ICXV=0
LINE 436 JCNV=0
LINE 437 C WRITE(16, 2010) NDET
LINE 438 C WRITE(16, 3008) IF, FREQ(I), M
LINE 439 C
LINE 440 C
LINE 441 170 I = 1, NET
LINE 442 C WRITE(16, 2009) DET T(I)
LINE 443 DI(I) = DET T(I)
LINE 444 DI(I) = AIMAG DET T(I)
LINE 445 170 CONTINUE
LINE 446 C WRITE(16, 2010) NPDX
LINE 447 C WRITE(16, 2009) UDPOX(I)
LINE 448 DO 172 I = 1, NET
LINE 449 CABS1 = SQRT( OR (1) **2 + DIM. 1-* 2)
LINE 450 IF (CABSI.NE. 0.0) 30 TO 178
LINE 451 172 CONTINUE
LINE 452 DO 174 I = 1, NET
LINE 453 174 U010271 OULU451U
LINE 454 C WRITE (16,2027)
LINE 455 178 CONTINUE
LINE 456 178 CONTINUE
LINE 457 3027 FKMAT1 = (1.0, 0.0), 3BISOLUTION FOR DOWMASH USING ITERATION I
LINE 458 IF ( KTAKE(I) ) IGWES=0
LINE 459 IF (H TAKA(I) W=KZ I) IGWES=0
LINE 460 IF (IMAG ,EQ. 0) IGWES=0
LINE 461 IF (anceled ,EQ. 0) IGWES=0
LINE 462 CALL ITRATE(I,1,1R,DI,DPDX,GT MC,D,MP,DNW,DDW,RdR,RmI)
LINE 463 IF (wakenz ) I=0
LINE 464 181 CONTINUE
LINE 465 CALL GETTIM(I,TIME)
LINE 466 CALL TOUTMU(SEC,5HESL,4HE FO,4HR PC,4HTEN,4HDR,A,4HDR,A)
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 LOOP ON WINGS AND SPANS
LINE 472 C NSPI IS SPAN COUNTER
LINE 473 C
LINE 474 C
LINE 475 C VSP 1 = 0
LINE 476 NSPI = NSPI + 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 TD(J,JM) = TD(J,JM)
LINE 482 NSPW = NSPW + 1
LINE 483 C WRITE (16, 3007)
LINE 484 C LOOP ON SPANS IN WING
LINE 485 C DO 220 ISP = 1,NSPW
LINE 486 C
LINE 487 C
LINE 488 C NSPI = NSPI + 1
LINE 489 C WRITE (16, 3007)
LINE 490 C IF (NSPW = 1) NWPS = NSPW - 1

-6.10-
**CODE CONTINUES FROM PREVIOUS PAGE**

```plaintext
LINE 491 CALL PHIL(XLE(XSPI),XCEN(INDX),UPUX(INDX),NFPS,PHI,CXKM)
LINE 492 C
LINE 493 IF(I(N.EQ.0)) GO TO 182
LINE 494 IF(I(N.EQ.0)) GO TO 182
LINE 495 IF(I(N.EQ.0)) GO TO 182
LINE 496 IF(I(N.EQ.0)) GO TO 182
LINE 497 CONTINUE
LINE 498 C WRITE(16,3002)J,ISP
LINE 499 DLS = (0.0, 0.0)
LINE 500 PMS = (0.0, 0.0)
LINE 501 C LOOP ON NUMBER OF ELEMENTS IN SPAN ISP
LINE 502 DO 210 IE=1,NEPS
LINE 503 PRES(INDX) = (UPDX(INDX) + CPRES*PHI(IE)) / TWOPI
LINE 504 DLCA = PRES(INDX) * AREA(INDX)
LINE 505 C WRITE(16,3003)INDXR,PHI(IE),PRES(INDX),DLCA
LINE 506 DLS = DLS + DLCA
LINE 507 PMS = PMS + (XP - XCEN(INDXR)) * DLCA
LINE 508 INDXR = INDXR + 1
LINE 509 LOOP ON ELEMENT IE OF SPAN ISP
LINE 510 END LOOP ON ELEMENT IE OF SPAN ISP
LINE 511 CONTINUE
LINE 512 C END LOOP ON SPAN ISP OF WING J
LINE 513 WRITE(16,3005)TL(J,JM),TM(J,JM)
LINE 514 END LOOP ON WINGS J
LINE 515 END LOOP ON DISPLACEMENT MODE TO CALCULATE
LINE 516 DO 270 IM=1,JMODE
LINE 517 CALL ROE14(A,P1,14,DETA,ETA,NET)
LINE 518 LOOP OVER ALL ELEMENTS BY WINGS
LINE 519 END
```

---

Source: `CODE_PRINTOUT.txt`
LINE 615  GO TO 500
LINE 616  COC REPLACES
LINE 617  C 10 CONTINUE
LINE 618  10 IF (PLTR1) CALL PLOTS (111,11H, M10 )
LINE 619  IF (PLTR1) CALL EPLOT(1HEND )
LINE 620  WRITE (16,3328)
LINE 621  STOP
LINE 622  STOP
LINE 623  STOP
LINE 624  STOP
LINE 625  STOP
LINE 626  STOP
LINE 627  STOP
LINE 628  STOP
LINE 629  STOP
LINE 630  STOP
LINE 631  STOP
LINE 632  STOP
LINE 633  STOP
LINE 634  STOP
LINE 635  STOP
LINE 636  STOP
LINE 637  STOP
LINE 638  STOP
LINE 639  STOP
LINE 640  STOP
LINE 641  STOP
LINE 642  STOP
LINE 643  STOP
LINE 644  STOP
LINE 645  STOP
LINE 646  STOP
LINE 647  STOP
LINE 648  STOP
LINE 649  STOP
LINE 650  STOP
LINE 651  STOP
LINE 652  STOP
LINE 653  STOP
LINE 654  STOP
LINE 655  STOP
LINE 656  STOP
LINE 657  STOP
LINE 658  STOP
LINE 659  STOP
LINE 660  STOP
LINE 661  STOP
LINE 662  STOP
LINE 663  STOP
LINE 664  STOP
LINE 665  STOP
LINE 666  END

-6.13-
LINE 1  BLOCK DATA
LINE 2  INTEGER PRINT
LINE 3  COMM/NCEPS/ EPS
LINE 4  COMM/ITERAT/ ITMX
LINE 5  CJM/J/ MAXE/ N4XE
LINE 6  COMM/J/MXSTOR/ 4NINGS/NSEC T/NX/ASPAN,NKCNKJ,NEPSN
LINE 7  1, NXFDZ
LINE 8  COMM/ /PEREPS/ PERC
LINE 9  COMM/TAPE/ 15,110,111,112,113,114,19,115
LINE 10 COMM/VELCOM/ NMAX,PRINT, NWBLOK, NWROW(20)
LINE 11 C NWNINGS= 4
LINE 12 C NSEC T= 3
LINE 13 C NMODE = 10
LINE 14 C NSPAN= 100
LINE 15 C NCORVN = 8
LINE 16 C NFROJN = 12
LINE 17 C NEPSN = 70
LINE 18 DATA EPS/.01/
LINE 19 DATA ITMX/10/
LINE 20 DATA 19,110,111/8,10,11/
LINE 21 DATA 112/12/
LINE 22 DATA 113/13/
LINE 23 DATA 114/14/
LINE 24 DATA 115/15/
LINE 25 DATA J6/6/
LINE 26 DATA N4MX/60/
LINE 27 DATA NINGS,NSEC T,NMODES,ASPAN,NKCNKJ,NEPSN
LINE 28 1, NXFDZ
LINE 29 2/20,30,10,100, 8, 12, 70, 300/
LINE 30 DATA PERC/3,005/
LINE 31 DATA PRINT/2/
LINE 32 END
LINE 1 FUNCTION CABS3(A1)
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 SINGLELY SUBSCRIPTED IN CALLING PROGRAM, SET NDIM=N.
A - ON INPUT THE MATRIX TO BE FACTORED.

P - IP(K) K.LT.N CONTAINS THE ROW INTERCHANGE INFORMATION. IP(N) CONTAINS (-1)**(NUMBER OF INTERCHANGE) 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(1)*A(1,1)]* [A(2,2)]*..* [A(N,N)].
(4) THE ROW INTERCHANGE INFORMATION STORED IN IP IS NOT EASY TO INTERPRET. HOWEVER, IT IS USED PROPERLY BY 'SOLVE'.

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED.

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

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

IP(N) = 1
DO 6 K = 1,N

IF (K .EQ. N) GO TO 5

KPL = KS1

DO 1 I = KPI,N

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

IF (M .EQ. K) IP(N) = -IP(N)

T = A(M,K)

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

A(K,K) = T

DO 2 I = KPI,N

2 A(I,K) = -A(I,K)/T

DO 4 J = KPI,N

4 A(I,J) = A(I,J) + A(I,K)*A(K,J)

5 CONTINUE

-6.16-
SUBROUTINE DIAGWROK,IF,LRECL,NNCH,MC, D, NET,AROW,IZ, IP

LINE 1  NSPT,FRQ,AROW,CHRWR
LINE 2  COMPLEX CAROW(1), CHRWR(1)
LINE 3  COMPLEX WTE
LINE 4  INTEGER PRINT
LINE 5  LOGICAL TREDGE
LINE 6  DIMENSION AROW(NET), IP(NET)
LINE 7  DIMENSION NNCH(I)
LINE 8  DIMENSION WROW(LRECL)
LINE 9  COMMON /K1112/ KST11,KST12
LINE 10  COMMON /TAPE/ I9, I10, I11, I12, I13, I14, J9, I15
LINE 11  COMMON /WAKE/ KAW, KAW(1), EPSW, JCNVGW, LRWTE, WTE
LINE 12  COMMON /WAKE UP/ WTE(12), TREDGE, KAW, KAW, KAWENZ, NWTE
LINE 13  READ(19)
LINE 14  READ(I15)
LINE 15  IBT = 0
LINE 16  K=0
LINE 17  DO 40 J = 1, NROW
LINE 18  K=K+1
LINE 19  CALL FREQW(IS,AROW,WROW,IF,LRECL)
LINE 20  DO 15 J=1, NROW
LINE 21  M= J+1
LINE 22  WRITE (110) D, IP
LINE 23  CONTINUE
LINE 24  END, FILE 110
LINE 25  REWIND 110
LINE 26  IF (IWTE .EQ. 0) READ(19)
LINE 27  IF (.NOT. WAKE) GO TO 20
LINE 28  IF (K .GT. KAW) GO TO 20
LINE 29  IF (F2.1.EQ.0.0) GO TO 20
LINE 30  CALL PAKET(I12, CAROW,CHRWR, IF,LRWTE, NSPT,
LINE 31  DO 15 J=1, NROW
LINE 32  CONTINUE
LINE 33  IBT = IBT + NROW
LINE 34  CALL MATPRINT(D, NROW, WRK, "D MATRIX")
LINE 35  WRITE (110) D, IP
LINE 36  CONTINUE
LINE 37  END, FILE 110
LINE 38  CONTINUE
LINE 39  IF (NROW .LE. 1) GO TO 40
LINE 40  CALL WRITED(K, AROW, IZ, WRK, NET)
LINE 41  IF (IWTE .EQ. 0) GO TO 20
LINE 42  IF (.NOT. WAKW) GO TO 20
LINE 43  IF (K .GT. KAW) GO TO 20
LINE 44  IF (K1 .GT. K) GO TO 20
LINE 45  IF (F2.1.EQ.0.0) GO TO 20
LINE 46  CALL WAKET(I12, CAROW,CHRWR, IF,LRWTE, NSPT,
LINE 47  DO 15 J=1, NROW
LINE 48  CONTINUE
LINE 49  IBT = IBT + NROW
LINE 50  CALL MATPRINT(D, NROW, WRK, "D MATRIX")
LINE 51  WRITE (110) D, IP
LINE 52  CONTINUE
LINE 53  END, FILE 110
LINE 54  END, FILE 111
LINE 55  END, FILE 110
LINE 56  END, FILE 110
LINE 57              RET   112
LINE 58              IF(IWTE .NE. 0)  ENO FILE 112
LINE 59              IF(IWTE .NE. 0)  REWIND 112
LINE 60              RETURN
LINE 61              END
00600010
00600020
00600030
00600040
00600050
00600060
00600070
00600080
00600090
00600100
00600110
00600120
00600130
00600140
00600150
00600160
00600170
00600180
00600190
00600200
00600210
00600220
00600230
00600240
00600250
00600260
00600270
00600280
00600290
00600300
00600310
00600320
00600330
00600340
00600350
00600360
00600370
00600380
00600390
00600400

-6.20-
<table>
<thead>
<tr>
<th>LINE</th>
<th>Statement</th>
<th>Description</th>
<th>LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUBROUTINE EFPLT(A)</td>
<td></td>
<td>00700010</td>
</tr>
<tr>
<td>2</td>
<td>CALL CALPLT(C.0,0.0,999)</td>
<td></td>
<td>00700020</td>
</tr>
<tr>
<td>3</td>
<td>RETURN</td>
<td></td>
<td>00700030</td>
</tr>
<tr>
<td>4</td>
<td>END</td>
<td></td>
<td>00700040</td>
</tr>
</tbody>
</table>
SUBROUTINE EONE

LOGICAL ERONE, EMIRRO
COMMON / EEW / XUP, YUP, XLW, YLW, ERONE, EMIRRO, RR, RL, RTR, RFL
COMMON / RRLL / TVW(4), YRO, YLO, ZR, ZRRZ, ZL, ZZLZ
COMMON / W11 / X0, Y0, Z0, ZDZD, SYMK

LINE 1 C
LINE 2 C
LINE 3 C
LINE 4 C
LINE 5 C
LINE 6 C
LINE 7 C
LINE 8 C
LINE 9 C
LINE 10 C
LINE 11 C
LINE 12 C
LINE 13 C
LINE 14 C
LINE 15 C
LINE 16 C
LINE 17 C
LINE 18 C
LINE 19 C
LINE 20 C
LINE 21 C
LINE 22 C
LINE 23 C
LINE 24 C
LINE 25 C
LINE 26 C
LINE 27 C
LINE 28 C
LINE 29 C
LINE 30 C
LINE 31 C
LINE 32 C
LINE 33 C
LINE 34 C
LINE 35 C
LINE 36 C
LINE 37 C
LINE 38 C
LINE 39 C
LINE 40 C
LINE 41 C
LINE 42 C
LINE 43 C
LINE 44 C
LINE 45 C
LINE 46 C
LINE 47 C
LINE 48 C
LINE 49 C
LINE 50 C
LINE 51 C
LINE 52 C
LINE 53 C
LINE 54 C
LINE 55 C
LINE 56 C

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
ZETAS = XQ - XW
ZETAS = XQ - XUP
IF( ZETAS .LE. 0.0 .AND. ZETAS .LE. 0.0 ) RETURN
ETA1 = YRO - YLW
ETA1 = YRO - YUP
T1 = SQRT( ETA1*ETA1 .G. ZRZL )
T2 = SQRT( ETA2*ETA2 .G. ZRZL )
CONTINUE
R1 = ZETAS - T1
R2 = ZETAS - T2
IF( R1 .LE. 0.0 .AND. R2 .LE. 0.0 ) RETURN
R1 = R1*(ZETAS .G. T1)
R2 = R2*(ZETAS .E. T2)
IF( ERONE ) GO TO 90
EMIRRO = .TRUE.
R TR = R1*R2
IF( R1 .LT. 0.0 ) R1 = 0.0
IF( R2 .LT. 0.0 ) R2 = 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 FGEN
LINE 2 COMMON /BASIC/MACH, BEETA
LINE 3 COMMON /FOI/NFREQ, CF(12,30), FREQ(12), FREQP(12), FRTST,NTMX
LINE 4 C NTMX = THE MAXIMUM NUMBER OF TERMS USED IN MINGR, MINT
LINE 5 C XUNDR = MACHINE DEPENDENT NUMBER, LARGEST NEGATIVE EXPONENT
LINE 6 C XUNDER = MACHINE DEPENDENT NUMBER, LARGEST NEGATIVE EXPONENT
LINE 7 C XUNDR = 292.0
LINE 8 CDC XUNDER = -77.5
LINE 10 C WRITE(6,1001) NFREQ, (FREQ(I), I=1,NFREQ)
LINE 11 C FRTST = -1.0E+65
LINE 12 DO 5 I = 1, NFREQ
LINE 13 C FREQ(12) = FREQ(I) * MACH / BEETA
LINE 14 C WRITE(6,1001) NFREQ, (FREQ(I), I=1,NFREQ), FRTST
LINE 15 C NTMX = 0
LINE 16 C NTMX = 1
LINE 17 C NTMX = 1
LINE 18 DO 40 I = 1, NFREQ
LINE 19 CF(I,I) = 1.0
LINE 20 XKK = FREQ(I)*FREQ(I)
LINE 21 IFI XKK = 0.0) GO TO 15
LINE 22 DO 10 J = 2, NTMX
LINE 23 C CF(I,J) = 0.0
LINE 24 GO TO 40
LINE 25 CONTINUE
LINE 26 DO 30 J = 2, NTMX
LINE 27 N2 = N2 - 1
LINE 28 FACTR = XKK / FLOAT(N2*(N2-1))
LINE 29 C CF(I,J) = ABS4 CF(I,N)
LINE 30 A10 = ALOG10 A10
LINE 31 A10 = ALOG10 A10
LINE 32 IFI XKK = 0.0) XUNDR = 0.0
LINE 33 DO 20 J = 2, NTMX
LINE 34 C CF(I,J) = 0.0
LINE 35 C NTMX = MAXO(NTMX, N)
LINE 36 GO TO 40
LINE 37 CONTINUE
LINE 38 CF(I,J) = - CF(I,N) * FACTOR
LINE 39 CONTINUE
LINE 40 NTMXX = NTMX
LINE 41 C CONTINUE
LINE 42 C CONTINUE
LINE 43 C CONTINUE
LINE 44 C CONTINUE
LINE 45 C WRITE(6,1001) I, (CF(I,J), J=1, NTMX)
LINE 46 C 1001 FORMAT(' // 18, 5X, SE18.5 // (13X, SE18.5)
LINE 47 C 50 CONTINUE
LINE 48 RETURN
LINE 49 END

-6.24-
<table>
<thead>
<tr>
<th>LINE</th>
<th>Statement</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUBROUTINE FREQ(I8,AROW,WROW,NFREQ,LRECL)</td>
<td>01000010</td>
</tr>
<tr>
<td>2</td>
<td>DIMENSION WROW(LRECL), AROW(NWING)</td>
<td>01000020</td>
</tr>
<tr>
<td>3</td>
<td>COMMON FQ1/NFREQ</td>
<td>01000030</td>
</tr>
<tr>
<td>4</td>
<td>COMMON/PARAM/NWING</td>
<td>01000040</td>
</tr>
<tr>
<td>5</td>
<td>EQUIVALENCE (NET,NWING)</td>
<td>01000050</td>
</tr>
<tr>
<td>6</td>
<td>READ(I8) K1, WROW</td>
<td>01000060</td>
</tr>
<tr>
<td>7</td>
<td>K = NF</td>
<td>01000070</td>
</tr>
<tr>
<td>8</td>
<td>DO 100 J = 1,NWING</td>
<td>01000080</td>
</tr>
<tr>
<td>9</td>
<td>AROW(J) = WROW(K)</td>
<td>01000090</td>
</tr>
<tr>
<td>10</td>
<td>K = K &amp; NFREQ</td>
<td>01000100</td>
</tr>
<tr>
<td>11</td>
<td>100 CONTINUE</td>
<td>01000110</td>
</tr>
<tr>
<td>12</td>
<td>RETURN</td>
<td>01000120</td>
</tr>
<tr>
<td>13</td>
<td>END</td>
<td>01000130</td>
</tr>
</tbody>
</table>
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

-6.26-
SUBROUTINE GRIDIN ( CORNX, CORNY, CORNZ, XYZ, Z, TRS, NNCH, SW, XLE, NSP, NE, NETSV, REFLEN, NFREQ, FREQ, NWS, NSECT, ICL, ERROR, ETA, LAB, ID, FD, IS, IF, ID))

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

INPUT

IS: INPUT UNIT (USUALLY CARD READER 5)

OUTPUT

CORNX, CORNY, CORNZ X, Y, AND Z COORD. OF CORNER POINTS

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

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

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

SW: SPAN WIDTH

XLE: AVERAGE X COORDINATE OF LEADING EDGE PER SPAN

NE: NUMBER OF ELEMENTS IN SECTION

ICL: ICL(I,J) CODE FOR SECTION I OF WING J

ERROR LOGICAL THAT COMES IN FALSE, MAY BE SET TRUE

ETA: SQRT(MACH*MACH-1) USED TO TRANSFORM CORNX AND CORNY

ERROR LOGICAL FOR TRANSFORM

LOGICAL ERROR

LOGICAL FREQ

DIMENSION CORNX(1), CORNY(1), CORNZ(1), TRS(1), XYZ(1), Y(1)

DIMENSION FREQ(10)

DIMENSION ID(1), FD(4)

DIMENSION XLE(1), SW(1)

DIMENSION XYZ(1), NNCH(40), ICL(394), NSECT(1)

DATA LB, LE, LC / 4H

DATA LG, LS, LM, LF / 4HGRID, 4HSPAN, 4HMUDE, 4HREQ /

DATA LG, LS, LM, LF / 4HGRID, 4HSPAN, 4HMUDE, 4HREQ /

TRANSFORM CORNER POINTS TO LOCAL SYSTEM, WORK IN LOCAL SYSTEM

CALL TRANSI CORNX, CORNY, CORNZ, TRS, X, Y, Z

*GRID* CARD MUST HAVE BEEN LAST CARD READ. (EITHER IN GRIDIN OR IN A PREVIOUS CALL TO THIS ROUTINE - SEE 80 CONTINUE)

FREQ = .FALSE.

IF W = IS(1)

IS = IS(2)
```
LINE 57    Y1 = Y(1) * REFLEN
LINE 58 C  INITIALIZE ELEMENT AND SPAN COUNTERS (NE AND ISP)
LINE 59    NE = 0
LINE 60    ISP = 0
LINE 61 C  WRITE( 16, 3000 ) JW, IS
LINE 62 C  10 CONTINUE
LINE 63 C  READ( IS, 1000 ) LAB, ID, FD
LINE 64 C  IF( LAB .EQ. LS ) GO TO 20
LINE 65 C  IF( LAB .EQ. LG ) GO TO 80
LINE 66 C  IF( LAB .EQ. LF ) GO TO 50
LINE 67 C  IF( LAB .EQ. LE ) GO TO 60
LINE 68 C  IF( LAB .EQ. LB ) GO TO 70
LINE 69 C  WRITE( 16, 2000 ) IE, ISP
LINE 70 C  WRITE( I6, 2018 ) ID(l), ISP
LINE 71 C  ERROR = .TRUE.
LINE 72 C  WRITE( I6, 2018 ) ISP, YSPAN
LINE 73 C  WRITE( 16, 2000 ) IE, ISP
LINE 74 C  WRITE( I6, 2027 ) ISP, YSPAN
LINE 75 C  WRITE( 16, 2000 ) IE, ISP
LINE 76 C  WRITE( I6, 2028 ) ISP
LINE 77 C  WRITE( 16, 2000 ) IE, ISP
LINE 78 C  WRITE( 16, 2000 ) IE, ISP
LINE 79 C  WRITE( 16, 2000 ) IE, ISP
LINE 80 C  WRITE( 16, 2000 ) IE, ISP
LINE 81 C  WRITE( 16, 2000 ) IE, ISP
LINE 82 C  WRITE( 16, 2000 ) IE, ISP
LINE 83 C  WRITE( 16, 2000 ) IE, ISP
LINE 84 C  WRITE( 16, 2000 ) IE, ISP
LINE 85 C  WRITE( 16, 2000 ) IE, ISP
LINE 86 C  WRITE( 16, 2000 ) IE, ISP
LINE 87 C  WRITE( 16, 2000 ) IE, ISP
LINE 88 C  WRITE( 16, 2000 ) IE, ISP
LINE 89 C  WRITE( 16, 2000 ) IE, ISP
LINE 90 C  WRITE( 16, 2000 ) IE, ISP
LINE 91 C  WRITE( 16, 2000 ) IE, ISP
LINE 92 C  WRITE( 16, 2000 ) IE, ISP
LINE 93 C  WRITE( 16, 2000 ) IE, ISP
LINE 94 C  WRITE( 16, 2000 ) IE, ISP
LINE 95 C  WRITE( 16, 2000 ) IE, ISP
LINE 96 C  WRITE( 16, 2000 ) IE, ISP
LINE 97 C  WRITE( 16, 2000 ) IE, ISP
LINE 98 C  WRITE( 16, 2000 ) IE, ISP
LINE 99 C  WRITE( 16, 2000 ) IE, ISP
LINE 100 C  WRITE( 16, 2000 ) IE, ISP
LINE 101 C  WRITE( 16, 2000 ) IE, ISP
LINE 102 C  WRITE( 16, 2000 ) IE, ISP
LINE 103 C  WRITE( 16, 2000 ) IE, ISP
LINE 104 C  WRITE( 16, 2000 ) IE, ISP
LINE 105 C  WRITE( 16, 2000 ) IE, ISP
LINE 106 C  WRITE( 16, 2000 ) IE, ISP
LINE 107 C  WRITE( 16, 2000 ) IE, ISP
LINE 108 C  WRITE( 16, 2000 ) IE, ISP
LINE 109 C  WRITE( 16, 2000 ) IE, ISP
LINE 110 C  WRITE( 16, 2000 ) IE, ISP
LINE 111 C  WRITE( 16, 2000 ) IE, ISP
LINE 112 C  WRITE( 16, 2000 ) IE, ISP
LINE 113 C  WRITE( 16, 2000 ) IE, ISP
LINE 114 C  WRITE( 16, 2000 ) IE, ISP
LINE 115 C  WRITE( 16, 2000 ) IE, ISP
LINE 116 C  WRITE( 16, 2000 ) IE, ISP
LINE 117 C  WRITE( 16, 2000 ) IE, ISP
LINE 118 C  WRITE( 16, 2000 ) IE, ISP
```
LINE 119   X2 = FD(I2)
LINE 120   C
LINE 121   WRITE(16, 3002) IC, X1, X2
LINE 122   X2 = X2/EEETA
LINE 123   X1 = X1/EEETA
LINE 124   SW(ISP) = YSPAN/REFLEN
LINE 125   XLE(ISP) = 0.5*X(16X21)/REFLEN
LINE 126   C
LINE 127   LOOP ON NUMBER OF CHORD CARDS 2, NC
LINE 128   DO 30 IC = 2, NC
LINE 129   C
LINE 130   READ(15, 1000) LAB, ID, X4, X3
LINE 131   C
LINE 132   WRITE(16, 3002) IC, X4, X3
LINE 133   X3 = X3/EEETA
LINE 134   X4 = X4/EEETA
LINE 135   C
LINE 136   XMIN = AMIN11(X4-X1), (X3-X2)
LINE 137   IF(XMIN .GT. 0.0) GO TO 25
LINE 138   XMAX = AMAX11(X4-X1), (X3-X2)
LINE 139   IF(XMIN .LE. 0.0) AND. XMAX .GT. 0.0) GO TO 25
LINE 140   WRITE(16, 2000)
LINE 141   WRITE(16, 2031) IC, ISP
LINE 142   ERROR = .TRUE.
LINE 143   25 CONTINUE
LINE 144   30 CONTINUE
LINE 145   NJ = 6*NE
LINE 146   NE = NE+1
LINE 147   XYZ(NJE1) = X1/REFLEN
LINE 148   XYZ(NJE2) = X2/REFLEN
LINE 149   XYZ(NJE3) = X3/REFLEN
LINE 150   XYZ(NJE4) = X4/REFLEN
LINE 151   XYZ(NJE5) = Y1/REFLEN
LINE 152   XYZ(NJE6) = Y2/REFLEN
LINE 153   X1 = X4
LINE 154   X2 = X3
LINE 155   30 CONTINUE
LINE 156   C
LINE 157   Y1 = Y2
LINE 158   NNCH(ISP) = NE & NETSV
LINE 159   C
LINE 160   GO TO 10
LINE 161   C
LINE 162   50 CONTINUE
LINE 163   C
LINE 164   FREQU. CARD READ
LINE 165   DO 55 I = 1, 4
LINE 166   55 FREQU(I) = FD(I1)
LINE 167   NFREQ = ID(I1)
LINE 168   IF(VREQ .GT. 4) READ(15, 1001) (FREQ(I), I=5, NFREQ)
LINE 169   WRITE(16, 3006) NFREQ, (FREQ(I), I=1,NFREQ)
LINE 170   GO TO 10
LINE 171   C
LINE 172   60 CONTINUE
LINE 173   C
LINE 174   END CARD READ
LINE 175   C
LINE 176   WRITE(16, 2000)
LINE 177   WRITE(16, 2041)
LINE 178   WRITE(16, 2015)
LINE 179   ERROR = .TRUE.
LINE 180   GO TO 90
-6.29-
LINE 181  65  CONTINUE
LINE 182  IF( FREQ ) GO TO 90
LINE 183  WRITE( 16, 2000 )
LINE 184  WRITE( 16, 2015 )
LINE 185  ERROR = .TRUE.
LINE 186  GO TO 90
LINE 187  C
LINE 188  70  CONTINUE
LINE 189  C  LABEL CARD EXPECTED BUT NOT RECEIVED
LINE 190  WRITE( 16, 2000 )
LINE 191  WRITE( 16, 2016 )
LINE 192  WRITE( 16, 2017 ) LAB, ID, FO
LINE 193  ERROR = .TRUE.
LINE 194  GO TO 10
LINE 195  C
LINE 196  80  CONTINUE
LINE 197  C  GRID CARD READ
LINE 198  C
LINE 199  I = ID(2)
LINE 200  J = ID(1)
LINE 201  IF( I .GT. NSECT(J) ) GO TO 81
LINE 202  IF( J .GT. NSECT(J) ) GO TO 81
LINE 203  IF( J .GT. NSECT(J) ) GO TO 82
LINE 204  IF( J .LT. JW .AND. J .LE. IS ) GO TO 82
LINE 205  ICL(I,J) = -1
LINE 206  GO TO 90
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, 2020 )
LINE 211  WRITE( 16, 2021 ) J, I
LINE 212  ERROR = .TRUE.
LINE 213  GO TO 90
LINE 214  82  CONTINUE
LINE 215  C  GRID DATA FOR NEXT GRID SECTION IS OUT OF ORDER
LINE 216  WRITE( 16, 2000 )
LINE 217  WRITE( 16, 2022 ) J, I
LINE 218  ERROR = .TRUE.
LINE 219  C
LINE 220  90  CONTINUE
LINE 221  C
LINE 222  C  SECTION TO RETURN
LINE 223  NETSV = NETSV & NE
LINE 224  IF( ISP .EQ. NSP ) RETURN
LINE 225  C  NUMBER OF SPAN CARDS READ DOES NOT AGREE WITH NO. SPANS (NSP)
LINE 226  WRITE( 16, 2030 ) IS, JW, NSP, ISP
LINE 227  RETURN
LINE 228  C
LINE 229  1000 FORMAT( A4, 2X, 9I2, 4E12.0 )
LINE 230  1001 FORMAT( 24X, 4E12.0 )
LINE 231  C
LINE 232  2030 FORMAT(/// IX, 129(1H*) )//24H ERROR IN READING INPUT.)
LINE 233  2040 FORMAT(37H MODE DATA IS MISSING OR OUT OF ORDER )
LINE 234  2050 FORMAT(42H FREQUENCY DATA IS MISSING )//OUT OF ORDER )
LINE 235  2060 FORMAT( 42H A LABEL CARD IS EXPECTED BUT NOT PRESENT. )
LINE 236  2070 FORMAT(/// IX, 129(1H*) )//24H ERROR IN READING INPUT.)
LINE 237  2080 FORMAT(/// IX, 129(1H*) )//24H ERROR IN READING INPUT.)
LINE 238  2090 FORMAT(/// IX, 129(1H*) )//24H ERROR IN READING INPUT.)
LINE 240  2010 FORMAT(/// IX, 129(1H*) )//24H ERROR IN READING INPUT.)
LINE 241  2020 FORMAT(/// IX, 129(1H*) )//24H ERROR IN READING INPUT.)
LINE 242  153H THIS SECTION HAS NOT BEENDEFINED FOR THE STRUCTURE. )
LINE 243  2022 FORMAT(19H GRID DATA FOR WING, I2, 3H SECTION, I2, 17H IS OUT OF ORDER.  ORDER, I)
LINE 244  1 ORDER, I
LINE 245  2026 FORMAT(19H SPAN CARDS ARE OUT OF ORDER. CARD FOR SPAN, I2, 17H WAS EXPECTED )
LINE 246  1 28H WAS READ WHEN CARD FOR SPAN, I2, 13H WAS EXPECTED )
LINE 247  2027 FORMAT(25H THE WIDTH OF SPAN NUMBER, I2, 7H EQUALS, E14.6 /
LINE 248  1 42H THE SPAN WIDTH MUST BE GREATER THAN ZERO. )
LINE 249  2028 FORMAT(34H THE NUMBER OF CHORD LINES IN SPAN, I2, 7H EQUALS, I3 /U12U249U
LINE 250  1 48H THE NUMBER OF CHORDS MUST BE GREATER THAN ZERO. )
LINE 251  2029 FORMAT(20H CHORD CARD FOR SPAN, I2, 13H NOT FOUND )
LINE 252  2030 FORMAT(4H SECTION, I2, 8H OF WING, I2, 20H WAS DEFINED TO HAVE, I2U252U
LINE 253  1 13, 7H SPANS., 14, 334 'SPAN' CARDS WERE ACTUALLY READ. )
LINE 254  2031 FORMAT(18H CHORD LINE NUMBER, I3, 8H OF SPAN, I3, 012U254U
LINE 255  1 62H IS IDENTICAL TO, CROSSES, OR IS ABOVE THE PREVIOUS CHORD LINE)
LINE 256  2 )
LINE 257  C
LINE 258  3000 FORMAT(1H/24X, 19H GRID INPUT FOR WING, I2, 8H SECTION, I2 )
LINE 259  3001 FORMAT(/ 24X, 4H SPAN, I3, 4H HAS, I3, 27H CHORD LINES AND A WIDTH 0U12U259U
LINE 260  1F, E13.6 / 24X, 5H CHORD, 9X, 24X1, 14X, 2H K2 )
LINE 261  3002 FORMAT(24X, I5, 5X, 2E16.6 )
LINE 262  3006 FORMAT( // / 23X, 21H NUMBER OF FREQUENCIES )
LINE 263  1ES, 112 / 20X, 19H LIST OF FREQUENCIES, 5F14.4 / (39X, 5F14.4 )
LINE 264  END

/END READ

END OF JOB.

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

-6.31-
CISPLAY...FILE  DECK 13

LINE  1 SUBROUTINE INP1(I5, I6, TITLE, ITRUN, MACH, REFLEN, XP, XPIN, ITRUN, MACH, REFLEN, XP, XPIN, 01300010
LINE  2 1 PLTR, IWTE, EARO, BEETA, LAB, ID, FD, ERROR ) 01300020
LINE  3 REAL MACH 01300030
LINE  4 LOGICAL ERRUR, PLOTR 01300040
LINE  5 DIMENSION TITLE(14), FD(4), ID(9) 01300050
LINE  6 COMMON /NEXCXMLIFLUSH 01300060
LINE  7 DATA LRUN, 4HRUN 01300070
LINE  8 C EBAR ELEMENT ASPECT RATIO 01300080
LINE  9 C READ RUN CARD 01300090
LINE 10 C READ RUN CARD 01300100
LINE 11 C READ RUN CARD 01300110
LINE 12 READ(I5, 1000) LAB, 10, FD 01300120
LINE 13 C READ RUN CARD 01300130
LINE 14 IF(LAB .EQ. LRUN) GO TO 2 01300140
LINE 15 WRITE( 16, 2000 ) 01300150
LINE 16 WRITE( 16, 2023 ) 01300160
LINE 17 IF(FLUSH = 1 01300170
LINE 18 2 CONTINUE 01300180
LINE 19 ITRUN = ID(1) 01300190
LINE 20 PLOTR = .FALSE. 01300200
LINE 21 IF(I(2) .EQ. 1) PLOTR = .TRUE. 01300210
LINE 22 IWTE = 0 01300220
LINE 23 IF(I(3) .EQ. 1) IWTE = 1 01300230
LINE 24 MACH = FD(1) 01300240
LINE 25 BEETA = SQRT(MACH*MACH-1.0) 01300250
LINE 26 REFLEN = FD(2) 01300260
LINE 27 XPIN = FD(3) 01300270
LINE 28 EARO = FD(4) 01300280
LINE 29 IF(EARO .EQ. 0.0) EARO = 1.1 01300290
LINE 30 IF(MACH .GT. 1.0) GO TO 5 01300300
LINE 31 WRITE( 16, 2000 ) 01300310
LINE 32 WRITE( 16, 2025 ) 01300320
LINE 33 MACH = 2.0 01300330
LINE 34 ERROR = .TRUE. 01300340
LINE 35 5 CONTINUE 01300350
LINE 36 IF(REFLEN .NE. 0.0) GO TO 6 01300360
LINE 37 WRITE( 16, 2000 ) 01300370
LINE 38 WRITE( 16, 2025 ) 01300380
LINE 39 ERROR = .TRUE. 01300390
LINE 40 REFLEN = 1.0 01300400
LINE 41 6 CONTINUE 01300410
LINE 42 IF(EARO .GT. 0.0) GO TO 7 01300420
LINE 43 WRITE( 16, 2000 ) 01300430
LINE 44 WRITE( 16, 2027 ) EARO 01300440
LINE 45 ERROR = .TRUE. 01300450
LINE 46 EARO = 1.0 01300460
LINE 47 7 CONTINUE 01300470
LINE 48 XP = XPIN/(BEETA*REFLEN) 01300480
LINE 49 WRITE( 16, 3007 ) MACH, REFLEN, XPIN 01300490
LINE 50 WRITE( 16, 3019 ) EARO 01300500
LINE 51 WRITE( 16, 3008 ) ITRUN 01300510
LINE 52 IF(ITRUN .EQ. 3) GO TO 88 01300520
LINE 53 IF(ITRUN .NE. 1) GO TO 8 01300530
LINE 54 WRITE( 16, 3009 ) 01300540
LINE 55 8 CONTINUE 01300550
LINE 56 88 CONTINUE 01300560

-6.32-
L INE 57  WRITE( 16, 3018 )
LINE 58  GO TO 9
LINE 59  8  CONTINUE
LINE 60  WRITE(16, 3010 )
LINE 61  9  CONTINUE
LINE 62  WRITE( 16, 3011 ) ID12)
LINE 63  IF( PLOTR ) GO TO 11
LINE 64  WRITE( 16,3012 )
LINE 65  GO TO 12
LINE 66  11  CONTINUE
LINE 67  WRITE( 16, 3013 )
LINE 68  12  CONTINUE
LINE 69  WRITE( 16, 3015 )
LINE 70  IF( IATE .NE. 1 ) WRITE( 16, 3016 )
LINE 71  IF( IATE .EQ. 1 )
LINE 72  RETURN
LINE 73  1000 FORMAT( 44, 2X, 9I2, 4E12.0 )
LINE 74  2000 FORMAT(///, 9X, 129H1M, //24H ERROR IN READING INPUT. )
LINE 75  2023 FORMAT( 3H RUN CARD IS MISSING OR OUT OF ORDER )
LINE 76  1T IMMEDIATELY FOLLOW THE TITLE CARD AND BE THE SECOND CARD IN THE 2INPUT DECK / 1H JOB IS TERMINATED )
LINE 77  2020 FORMAT( 3H MACH NUMBER MUST BE GREATER THAN 1.0 / 24H MACH NUMBER WILL BE SET EQUAL TO 1.0 INC )
LINE 78  2026 FORMAT( 9H OFF-DIAGONAL PERCENTAGE, IF ENTERED, MUST BE A POSITIVE NUMBER. )
LINE 79  2027 FORMAT( 6H ELEMENT ASPECT RATIO, IF ENTERED, MUST BE A POSITIVE NUMBER. )
LINE 80  2028 FORMAT( 20X, 13H RUN TYPE CODE, 120 )
LINE 81  3007 FORMAT( 20X, 11H MACH NUMBER, F22.5 / 20X, 16H REFERENCE LENGTH, F13.5 / )
LINE 82  1F17.5 // 20X, 20H PITCHING MOMENT AXIS, F13.5 )
LINE 83  3008 FORMAT( / 20X, 13H RUN TYPE CODE, 120 )
LINE 84  3009 FORMAT( 20X, 22H COMPLETE RUN ATTEMPT )
LINE 85  3010 FORMAT( 20X, 11H CHECK RUN )
LINE 86  3011 FORMAT( / 20X, 17H PLOT REQUEST CODE, 116 )
LINE 87  3012 FORMAT( 20X, 24H NO PLOTS ARE REQUESTED )
LINE 88  3013 FORMAT( 20X, 21H PLOTS ARE REQUESTED )
LINE 89  3014 FORMAT( / 20X, 23H OFF-DIAGONAL PERCENTAGE, F10.5 )
LINE 90  3015 FORMAT( / 20X, 16H WAKE EFFECT CODE, 117 )
LINE 91  3016 FORMAT( 20X, 3H WAKE EFFECT IS NOT CONSIDERED )
LINE 92  3017 FORMAT( 20X, 2H WAKE EFFECT IS CONSIDERED )
LINE 93  3018 FORMAT( 20X, 13H (RESTART RUN) )
LINE 94  3019 FORMAT( / 20X, 20H ELEMENT ASPECT RATIO, F13.5 )
LINE 95  3020 FORMAT( / 20X, 20H ELEMENT ASPECT RATIO, F13.5 )
LINE 96  3021 FORMAT( / 20X, 20H ELEMENT ASPECT RATIO, F13.5 )
LINE 97  3022 FORMAT( / 20X, 20H ELEMENT ASPECT RATIO, F13.5 )
LINE 98  3023 FORMAT( / 20X, 20H ELEMENT ASPECT RATIO, F13.5 )
LINE 99  3024 FORMAT( / 20X, 20H ELEMENT ASPECT RATIO, F13.5 )
LINE 100  3025 FORMAT( / 20X, 20H ELEMENT ASPECT RATIO, F13.5 )
LINE 101  3026 FORMAT( / 20X, 20H ELEMENT ASPECT RATIO, F13.5 )
LINE 102  3027 FORMAT( / 20X, 20H ELEMENT ASPECT RATIO, F13.5 )
LINE 103  3028 FORMAT( / 20X, 20H ELEMENT ASPECT RATIO, F13.5 )
LINE 104  3029 FORMAT( / 20X, 20H ELEMENT ASPECT RATIO, F13.5 )
LINE 105  3030 FORMAT( / 20X, 20H ELEMENT ASPECT RATIO, F13.5 )
LINE 106  3031 FORMAT( / 20X, 20H ELEMENT ASPECT RATIO, F13.5 )
LINE 107  END

-6.33-
DISPLAY...FILE
DECK 14

LINE 1  FUNCTION IRELE ( GT, GW, NET)
LINE 2  COMPLEX GBAR, RI, RIS
LINE 3  COMMON / CEPS / EPS
LINE 4  COMPLEX GT(I), GW(I)
LINE 5  IRELE = 1
LINE 6  ANET = NET
LINE 7  RABSO = 0.0
LINE 8  DO 600 IEL=1,NET
LINE 9  GTABS = CABS(GT(IEL))
LINE 10 GABSW = CABS(GW(IEL))
LINE 11 IF (GABSW .EQ. 0.0 .AND. GTABS .EQ. 0.0) ANET = ANET - 1.0
LINE 12 IF (GABSW .EQ. 0.0 .AND. GTABS .EQ. 0.0) GO TO 600
LINE 13 RI = GT(IEL) - GW(IEL)
LINE 14 RIS = RI * RI
LINE 15 IF (GTABS .NE. 0.0) RIS = RIS / GT(IEL)
LINE 16 600 RABSO = CABS(RIS)GRABSO
LINE 17 ERROR = RABSO / ANET
LINE 18 ERROR = SQRT(ERROR)
LINE 19 IF (ERROR .GT. EPS) RETURN
LINE 20 IRELE = 0
LINE 21 RETURN
LINE 22 END

-6.34-
DISPLAY...FILE

DECK 15

LINE 1  SURROJUTE ITRATE (1,9, NWI,M,..,;X, ...D,IP, DNWR, DNWI, RK, RM 1, 0150001)
LINE 2  1 NET, A, IL, JG=ESS, NWI 01500023
LINE 3  COMPLEX Sh, GT, sh 01500030
LINE 4  COMPLEX GMESS 01500040
LINE 5  REAL km, NWI 01500050
LINE 6  INTEGER PRINT 01500060
LINE 7  DIMENSION 0(MC,MC), IP(MC),DNWR(MC),DNWI(MC), RK(MC), RM(MC) 01500070
LINE 8  DIMENSION wMESS(200) 01500080
LINE 9  DIMENSION NWR(200), NWI(200), GW(200), GT(200) 01500090
LINE 10  DIMENSION kM(200), km(200), GWR(233), GT(200) 01500100
LINE 11  DIMENSION kM(200) 01500110
LINE 12  COMMON /CEPS/ EPS 01500120
LINE 13  COMMON /ITG/ ITMAX 01500130
LINE 14  COMMON /ITG/ IGMESS 01500140
LINE 15  COMMON /KLI, /KST12, KST12 01500150
LINE 16  COMMON /TAPE/ IN, I00 01500160
LINE 17  COMMON /VELCOM / NMAX, PRINT, NWBLOK, NWROW (20), 01500170
LINE 18  REWIND 10 01500180
LINE 19  NWING = NET 01500190
LINE 20  IF (NWING .EQ. 1) GO TO 700 01500200
LINE 21  IMAX=10 01500210
LINE 22  IF (ITMAX .NE. 0) IMAX=ITMAX 01500220
LINE 23  SET CONVERGENCE CRITERIA — EPS 01500230
LINE 24  C 01500240
LINE 25  C 01500250
LINE 26  C 01500260
LINE 27  C 01500270
LINE 28  C 01500280
LINE 29  C 01500290
LINE 30  C 01500300
LINE 31  C 01500310
LINE 32  C 01500320
LINE 33  C 01500330
LINE 34  C 01500340
LINE 35  C 01500350
LINE 36  C 01500360
LINE 37  C 01500370
LINE 38  C 01500380
LINE 39  C 01500390
LINE 40  C 01500400
LINE 41  C 01500410
LINE 42  C 01500420
LINE 43  C 01500430
LINE 44  C 01500440
LINE 45  C 01500450
LINE 46  C 01500460
LINE 47  C 01500470
LINE 48  C 01500480
LINE 49  C 01500490
LINE 50  C 01500500
LINE 51  C 01500510
LINE 52  C 01500520
LINE 53  C 01500530
LINE 54  C 01500540
LINE 55  C 01500550
LINE 56  C 01500560
LINE 57 150 CONTINUE
LINE 58 160 CONTINUE
LINE 59 165 CONTINUE
LINE 60 IF (N4.LO.1, EQ.1) GO TO 530
LINE 61 172 CONTINUE
LINE 62 IF(W(N)) 110
LINE 63 190 IF (LABS(PRINT).LT.3) GO TO 200
LINE 64 C WRITE (6,630) IT
LINE 65 C WRITE (6,650) NWDLK
LINE 66 200 CONTINUE
LINE 67 IF (ITEST.EQ.1).AND.ITNE.1) GO TO 530
LINE 68 IF (IT.EQ.IMAX(1)) GO TO 540
LINE 69 330 JJ =0
LINE 70 ILD =IT&L
LINE 71 1 I = 1
LINE 72 ITTEST=O
LINE 73 REWIND 19
LINE 74 REWIND 110
LINE 75 JJ =0
LINE 76 IL = 1
LINE 77 L = 0
LINE 78 L1 = 0
LINE 79 DO 510 I =1,NWBLK
LINE 80 NW# =NW(JM)
LINE 81 READ (110) D, IP
LINE 82 C IF (LABS(PRINT).LT.3) GO TO 340
LINE 83 C WRITE (6,564) 1,NROW
LINE 84 C WRITE (6,566) NROW, D
LINE 85 C566 FORMAT(1HO,
LINE 86 C564 FORMAT(1110,
LINE 87 C340 CONTINUE
LINE 88 C562 FORMAT(IHO, 7HTRATE, 7H1ROW = , 13, / (1X,1UF10.5))
LINE 89 C564 FORMAT(IHO, 7HTRATE, 7H1ROW = , 13, / (1X,1UF10.5))
LINE 90 DO 471 J=1,NROW
LINE 91 L = L1
LINE 92 DNWR(J) =0.0
LINE 93 DNWI(J) =0.0
LINE 94 C IF (NWING.LE.NMAX) GO TO 470
LINE 95 IFILL.GT. KST11) NNZ =0
LINE 96 - IFILL.GT. KST11) GO TO 470
LINE 97 IFILL.LT. L )
LINE 98 1READ (15) 11, NNZ, (IZ(M),M=1,NNZ), (AM(M),M=1,NNZ)
LINE 99 IFILL.GT. L ) NNZ =0
LINE 100 IF(NNZ .EQ. 0) GO TO 470
LINE 101 DD 470 M=1,NNZ
LINE 102 K = IZ(M)
LINE 103 DNWR(J) = DNWR(J) & A(M) = REAL (GW(K)).
LINE 104 460 DNWI(J) = DNWI(J) & A(M) = AIMAG (GW(K)).
LINE 105 470 RWR(J) = NWI(J-JJJ)-DNWR(J)
LINE 106 RWI(J) = NWI(J-JJJ)-DNWI(J)
LINE 107 IF ((ABS(PRINT).LT.3) GO TO 471
LINE 108 C WRITE(6,3000) J, J1,ONW(J), NW(J), RW(J)
LINE 109 C3000 FORMAT(4X, 27H J, J1,ONW(J), NW(J), RW(J) = / (6X, 214,6B12.4) )
LINE 110 471 CONTINUE
LINE 111 CALL SOLVE ( NROW, NMAX, D, RWR, IP )
LINE 112 CALL SOLVE (NROW, NMAX, D, RWI, IP )
LINE 113 DD 500 M=1,NROW
LINE 114 GT11 =GW(I1)
LINE 115 SAVE =A(II)
LINE 116 GW11 = CMPLX (RWR(M), RWI(M))
LINE 117 GT11 =GW(I1)-GT11
LINE 118 IF(IT .EQ.2) GO TO 490

-6.36-
LINE 119  SAVE = Y(W(I)) # SAVE  
LINE 120  ALF = ALF1  
LINE 121  IF (SAVE.GE.0.1) ALF = ALF2  
LINE 122  490 CONTINUE  
LINE 123  G(W(I)) = ALF * G(W(I)) + (1.0 - ALF) * G(T(I))  
LINE 124  C  ITEST = IRELE(G(T(I)), G(W(I)), IT-ST)  
LINE 125  IT = IT+1  
LINE 126  500 CONTINUE  
LINE 127  JJ = JJ # NWROW(I)  
LINE 128  510 CONTINUE  
LINE 129  ITEST = IRELE(GT, G(W, 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  IMETH = 0  
LINE 135  C  GO TO 550  
LINE 136  C  GO TO 60  
LINE 137  550 CONTINUE  
LINE 138  WRITE (6,600)  
LINE 139  560 WRITE (6,610)  
LINE 140  RETURN  
LINE 141  600 WRITE (6,650) NWLN, (GT(N), N=1, NWLNG)  
LINE 142  610 WRITE (6,660) NWLN, (GW(N), N=1, NWLNG)  
LINE 143  C  END  
LINE 144  C  
LINE 145  570 FORMAT (2X,10H(N), N=1,13/1X,10F10.5)  
LINE 146  580 FORMAT (1HO,38H THE ITERATION CONVERGED AFTER ,13,2X,35HITERATIONS)  
LINE 147  1WITH A TEST CRITERION OF,1T10.7)  
LINE 148  590 FORMAT (1HO,37H THE ITERATION DID NOT CONVERGE AFTER ,13,2X,35HITERATIONS)  
LINE 149  1ATION S WITH A TEST CRITERION OF,1T10.7)  
LINE 150  600 FORMAT (1HO,40H THE SOLUTION AT THE PREVIOUS ITERATION IS)  
LINE 151  610 FORMAT (1HO,40H THE SOLUTION AT THE PRESENT ITERATION IS)  
LINE 152  630 FORMAT (1HO,40H THE SOLUTION AT THE PRESENT ITERATION IS)  
LINE 153  650 FORMAT (2X,10HGW(N), N=1,13/1X,6E18.5)  
LINE 154  C  
LINE 155  END
LINE 1 SUBROUTINE LOOPW(NET, NNSP, SYM, TRS, NNCH, NE, 01600010)
LINE 2 1 NSUB, ZSUB, ISUB, NSG T, TAPEB, XGNN, AREA, WROW, LRECL, 01600020
LINE 3 2 WATERW, PERC, ERROR, IWT, REFLEX, 01600030
LINE 4 C NET TOTAL NUMBER OF ELEMENTS 01600040
LINE 5 C NTERM NUMBER OF TERMS IN INFLUENCE FUNCTION 01600050
LINE 6 C NWING NUMBER OF WINGS IN STRUCTURE 01600060
LINE 7 C NSP NSP(I,J) = NU SPANS IN SECTION I OF WING J 01600070
LINE 8 C SYM(S) SYMMETRY CODE FOR WING J 01600080
LINE 9 C TRS TRS(I,J,1) TRANSFORMATION MATRIX OF SECT. I OF WING J 01600090
LINE 10 C NNCH ELEMENT NUMBER AT END OF SPAN 01600100
LINE 11 C NSP(J) NUMBER OF SPANS IN SECTION I OF WING J 01600110
LINE 12 C XYZ COORDINATE ARRAY OF ALL ELEMENTS 01600120
LINE 13 C NSP(J) NUMBER OF SECTIONS IN WING J 01600130
LINE 14 C ZSUB(J,J) Z COORDINATE OF ALL ELEMENTS OF SECTION I, WING J 01600140
LINE 15 C NSUBS(J,J) = 1 IF SECTION K2 HAS INFLU. ON REC. SECTION K1 01600150
LINE 16 C ISEX(J,J) = 0 IF SECTION K2 HAS INFLU. ON REC. SECTION K1 01600160
LINE 17 C ISEX(J,1) = 0 IF SECTION K2 HAS INFLU. ON REC. SECTION K1 01600170
LINE 18 C WATP LOGICAL UNIT NUMBER TO CONTAIN FREQUENCY-INDEP. 01600180
LINE 19 C WATERW WATP LOGICAL UNIT NUMBER TO CONTAIN FREQUENCY-INDEP. 01600190
LINE 20 C XGNN X CENTER OF EACH ELEMENT (OUTPUT) 01600200
LINE 21 C YO X COORDINATE OF RECEIVING PT. (CENTER OF ELEMENT) 01600210
LINE 22 C YO Y COORDINATE OF RECEIVING PT. (IN LEFT SIDE) 01600220
LINE 23 C YO Y COORDINATE OF RECEIVING PT. (IN LEFT SIDE) 01600230
LINE 24 C AREA AREA FOR EACH ELEMENT (OUTPUT) 01600240
LINE 25 C LRECL LENGTH OF WROW EQUAL TO NTERM#NET 01600250
LINE 26 C ERROR LOGICAL SET TO TRUE IF ERROR IS DETECTED 01600260
LINE 27 C COMPLEX WATERW LRECL 01600270
LINE 28 C COMPLEX WATERW LRECL 01600280
LINE 29 C COMPLEX WATERW LRECL 01600290
LINE 30 C REAL MACH 01600300
LINE 31 C INTEGER TAPEB 01600310
LINE 32 C LOGICAL BKWSP 01600320
LINE 33 C LOGICAL ERROR 01600330
LINE 34 C LOGICAL TREDGE 01600340
LINE 35 C LOGICAL WAKE, WAKE1, WAKE2 01600350
LINE 36 C LOGICAL WAKE2 01600360
LINE 37 C DIMENSION NSUB(12,1), TRS(4,3,1), NSP(3,1) 01600370
LINE 38 C DIMENSION NNCH(1), SYM(1) 01600380
LINE 39 C DIMENSION NSUB(1), NE(11,1), ZSUB(3,1), XGNN(1), XYZ(1), AREA(1) 01600390
LINE 40 C DIMENSION WSVI(20) 01600400
LINE 41 C DIMENSION NROW(LRECL) 01600410
LINE 42 C COMMON/BASIC/MACH 01600420
LINE 43 C COMMON /ტ\ / NASCAR, CF(12,30), FRECl2), FRECl3), FREST 01600430
LINE 44 C COMMON /rstrip /TV(4), YO, XGNN, YO, ZR, ZR, LR, LR, ZL, LR, ZL, ZL, IRC, IN 01600440
LINE 45 C COMMON /TAPE/ 15,110,111,112,113,114,115 01600450
LINE 46 C COMMON/WAKEUP/ TREC(12), TREDGE, WAKE1, WAKE2, WAKE, WAKE, WAKE, WAKE 01600460
LINE 47 C COMMON /WWI /XO, YO, ZD, ZD, SYM, NINSID 01600470
LINE 48 C COMMON /WWI /XO, YO, ZD, ZD, SYM, NINSID 01600480
LINE 49 C COMMON /WWI /XO, YO, ZD, ZD, SYM, NINSID 01600490
LINE 50 C THE FREQUENCY-INDEPENDENT TERMS FOR ALL INFLUENCING ELEMENTS 01600500
LINE 51 C OF A GIVEN RECEIVING ELEMENT ARE STOPPED IN WROW 01600510
LINE 52 C COMPUTE 1 ROW FOR EVERY ELEMENT (CALLED RECEIVING ELEMENT) 01600520
LINE 53 C IN TERMS IS THE ND TERMS USED TO COMPUTE AN ELEMENT OF W 01600530
LINE 54 C INDEX INDEX INTO XYZ FOR RECEIVING ELEMENT 01600540
LINE 55 C INDEX INDEX INTO XYZ FOR RECEIVING ELEMENT 01600550
LINE 56 C INDEX INDEX INTO XYZ FOR RECEIVING ELEMENT 01600560
K1 IS COUNTER FOR RECEIVING SECTION, K2 FOR INFLUENCING SECTION.

IF(IWTE .NE. 0) REWIN TAPES

WRITE (1, (TAPE8) MACH, NET, REFLEN, IN", ": NFREQ, (FREQ(I), I=1, NFREQ)

WRITE (115) MACH, NET, REFLEN, IWTE, NFREQ, (FREQ(I), I=1, NFREQ)

WRITE = .FALSE.

INDXR = 1

K1 = 0

KW = 0

KWAKE = 0

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

DO 100 J = 1, NWING

NSR = NSECT(J)

SYM = SYM(J)

DO 95 I = 1, NSR

K1 = K1 + 1

NER = NE(I, J)

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

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

IF (IWTE .EQ. 0) GO TO 50

DO 45 IE1 = 1, LRWTE

WRITE (11) (WTER(IE1), IE1 = 1, NRWTE)

CONTINUE

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

XO = XCE(NIRC)

YCR = .50 * (XYZ(INDXR4) + XYZ(INDXR5))

DSX = 0.5 * (XYZ(INDXR5) - XYZ(INDXR4))

AREA(IRC) = DSX * (XYZ(INDXR3) - XYZ(INDXR2)) * XYZ(INDXR1)

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

INDXR = INDXR + 6

IF (IWTE .EQ. 0) GO TO 50

DO 45 IE1 = 1, LRWTE

WRITE (11) (WTER(IE1), IE1 = 1, NRWTE)

CONTINUE

FIND DIAGONAL ELEMENT

CALL RTDI (TRS(I, J), TRS(I, J), YCR, ZCR, ZK0, ZL0)

ZK = ZK0 - ZCR

ZL = ZL0 - ZCR

ZKK = ZK * ZR

ZLL = ZL * ZL

TREDGE = .FALSE.

NINSID = 0

IN = IRC

INDXI = 6 * IRC - 5

ICOL = NFREQ*(IRC-1) + 1
CALL WINTGRI(XYZ(INDXI), XYZ(INDX1&4), WROW(ICOL))

SET UP CONSTANTS FOR INFLUENCING ELEMENTS

INDXI = 1
IN = 1
NSPS = 0
K2 = 0
ICOL = 1

LOOP ON INFLUENCING ELEMENT BY WING JJ, SECTION II (DO 85 JJ = 1,NWING)

DO 85 JJ = 1,NWING

SYMK = SYM(JJ)
NSI = NSECT(JJ)

DO 80 I I = 1, NSI

NEI = NO. ELEMENTS IN SECTION II OF WING JJ
NEI = NEI(JJ)
NSPS = NSP(JJ)

IF(1SONS(K1,K2) .NE. 0) GO TO 60

NTNEI = NEI*NEFREQ
IEI = I, NTNEI

WROW(ICOL) = 0.0
ICOL = ICOL + 1

55 CONTINUE

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

TRANSFORM CENTER PT. FROM RECEIVING SECTION SYSTEM TO INFLU-ENCING SECTION SYSTEM. X IS SAME IN BOTH SYSTEMS.

CALL RTOI (TRSI1,II, JJ, YCR, ZCR, ZRO, ZLO)

ZD = ZRO - ZSECT(II, JJ)
ZL = ZLO & ZSECT(II, JJ)
ZRZK = ZR*ZK
ZLZL = ZL*ZL

LOOP ON SPANS OF INFLUENCING SECTION II OF WING JJ DO 75 ISP = 1, NSPS

DO 75 ISP = 1, NSPS
LINE 181 C     NSPI = NSPI + 1
LINE 182 C     NFPSM1 = VNCH(NSPI) - IIN
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     NFPSMI = NNCH(NSPI) - IIN
LINE 187 C     NEPS = NFPSMI + 1
LINE 188 C     LOOP ON INFLU. ELEM. OF SPAN ISP FROM BOTTOM UP DO 70 IEI
LINE 189 C     IF( BKWSP) GO TO 65
LINE 200 C     WAKE1 = .FALSE.
LINE 201 C     DO 70 IEI = 1, NEPS
LINE 202 C     IIN = IIN - 1
LINE 203 C     CALL WINTGR( XYZ(INDXI), XYZ(INUXI), WRUW(icol) )
LINE 204 C     TREDGE = .FALSE.
LINE 205 C     IF( VINSID .EQ. 0 ) GO TO 69
LINE 206 C     IF( .NOT. WAKE1 ) GO TO 58
LINE 207 C     WAKE = .TRUE.
LINE 208 C     WAKE2 = .FALSE.
LINE 209 C     INOWK = NFREQ*(NSPI-1)
LINE 210 C     DO 56 IW=L,NFREQ
LINE 211 C     INDWK = INOWK + I
LINE 212 C     WTERW(INDWK) = WTERW(IW)
LINE 213 C     IF( .NOT. WAKE2 ) GO TO 58
LINE 214 C     IF( VINSID .EQ. 0 ) GO TO 69
LINE 215 C     TESTW = WROW(IJ)
LINE 216 C     56 CONTINUE
LINE 217 C     IF( VINSID .EQ. 1 ) GO TO 69
LINE 218 C     RD = ABS(WROW(IJ) / WIT )
LINE 219 C     IF( 40 .LT. PERC ) GO TO 62
LINE 220 C     BKWSP = .TRUE.
LINE 221 C     IJ = ICOL - 1
LINE 222 C     DO 68 IF = 1, NFREQ
LINE 223 C     WROW(IJ) = 0.0
LINE 224 C     WFSV( IF) = 0.0
LINE 225 C     58 CONTINUE
LINE 226 C     IF( VINSID .EQ. 2 ) GO TO 69
LINE 227 C     TESTW = WROW(IJ)
LINE 228 C     IF( .NE. 0.0 ) TESTW = TEST/WROW(IJ)
LINE 229 C     IJ = ICOL + NFREQ - 1
LINE 230 C     IF( VINSID .GT. 3 ) GO TO 63
LINE 231 C     TEST = WROW(IJ)
LINE 232 C     61 CONTINUE
LINE 233 C     TEST = TEST - WROW(IJ)
LINE 234 C     GO TO 69
LINE 235 C     63 CONTINUE
LINE 236 C     TEST = TEST + WROW(IJ)
LINE 237 C     IF( .NE. 0.0 ) TEST = TEST/WROW(IJ)
LINE 238 C     BKWSP = .TRUE.
LINE 239 C     IJ = ICOL - 1
LINE 240 C     GO TO 61
LINE 241 C     IF( .NE. 0.0 ) TEST = TEST/WROW(IJ)
LINE 242 C     GO TO 61
LINE 243     IJ = IJ & 1
LINE 244     64 WFSV(IF) = WROW(IJ)
LINE 245     GO TO 69
LINE 246     C
LINE 247     65 CONTINUE
LINE 248     IJ = ICOL - 1
LINE 249     DO 67 IF=1,NFREQ
LINE 250     IJ = IJ & 1
LINE 251     67 WROW(IJ) = WFSV(IF)
LINE 252     C
LINE 253     65 CONTINUE
LINE 254     ICOL = ICOL - NFREQ
LINE 255     INDEX = INDEX - 6
LINE 256     C
LINE 257     END LOOP ON ELEMENTS IN THE INFLUENCING SPAN
LINE 258     70 CONTINUE
LINE 259     C
LINE 260     IIN = IIN & NEPS
LINE 261     INDEX = INDEX & 6*NEPS & 6
LINE 262     ICOL = ICOL & NFREQ*NEPS & NFREQ
LINE 263     C
LINE 264     END LOOP ON INFLUENCING SPANS OF SECTION II OF WING JJ
LINE 265     75 CONTINUE
LINE 266     C
LINE 267     END LOOP ON INFLUENCING SECTION II OF WING JJ
LINE 268     80 CONTINUE
LINE 269     C
LINE 270     END LOOP ON INFLUENCING WING JJ
LINE 271     85 CONTINUE
LINE 272     C
LINE 273     WROW IF ROW HAS BEEN COMPUTED FOR A GIVEN RECEIVING ELEMENT.
LINE 274     WRITE WROW ON FILE AND LOOP TO NEXT RECEIVING ELEMENT
LINE 275     C
LINE 276     WRITE(TAPE8) IRC,WROW
LINE 277     C
LINE 278     ZERO FOR WAKE ELEMENTS
LINE 279     WAKE = FALSE - NO WAKE ELEMENTS FOR ANY ELEMENT
LINE 280     WAKE1 = TRUE - ELEMENT IS WAKED
LINE 281     IF (IWE.EQ.0) GO TO 90
LINE 282     IF (.NOT. WAKE2) GO TO 90
LINE 283     WRITE(115) IRC,WTEROW
LINE 284     KH = KW & 1
LINE 285     C
LINE 286     IF(IWE.NE.0)WRITE(6,3001) IRC,IER,I,J,WTEROW
LINE 287     C
LINE 288     WRITE(16,3001) IRC,IER,I,J,WROW(IJ),IJ=IJ1,ICOL
LINE 289     301 ICOL = ICOL & 1
LINE 290     C
LINE 291     WRITE(16,3001) IRC,IER,I,J,WROW
LINE 292     C
LINE 293     END LOOP ON RECEIVING ELEMENT IN SECTION I OF WING J
LINE 294     90 CONTINUE
LINE 295     CONTINUE
LINE 296     95 CONTINUE
LINE 297     C
LINE 298     END LOOP ON RECEIVING SECTION I OF WING J
LINE 299     100 CONTINUE
LINE 300     C
LINE 301     C
LINE 302     110 CONTINUE
LINE 303     C
LINE 304     C

-6.42-
LINE 305 IF(WAKE) END FILE 115
LINE 306  I6 =6
LINE 307 IF(.NOT. WAKE) WRITE(16,1000)
LINE 308 IF(WAKE) WRITE((16,1010) KW
LINE 309  1000 FORMAT(1HO///16X,37H WAKE EFFECTS HAVE BEEN DETERMINED )
LINE 310  1010 FORMAT(1HO///16X,38H WAKE EFFECTS HAVE BEEN DETERMINED FOR ,114,
LINE 311   1 5H 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

ORIGINAL PAGE IS
OF POOR QUALITY
SUBROUTINE MESH(CORNX,CORNY,CORNZ, XYZ, Z, TRS, NNCH, S4, XLE, NSP, NE, NETSV, EAO)
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 65    NC1=1
LINE 66    DC1 = AMAX1(DX12, DX34)
LINE 67    IF( DC1 .LT. DC ) GO TO 11
LINE 68    NC1 = DC1/DC
LINE 69    DC1 = DC1/NC1
LINE 70    11 CONTINUE
LINE 71    NE=0
LINE 72    LOOP ON SPANS
LINE 75    DO 100 I=1,NSP
LINE 76    NC1=1
LINE 77    LOOP ON NC1 FOR AREA 1
LINE 80    FIND THE Y COORD. FOR THIS SPAN (AY AND BY)
LINE 81    BY = Y1 + I*DS
LINE 82    AY = BY - DS
LINE 83    FIND AX,BX, X COORD ON LEADING EDGE ON BOTH SIDES OF SPAN
LINE 84    BX = X1 + I*DXLE
LINE 85    AX = BX - DXLE
LINE 86    SW(I) = DS
LINE 87    KLE(I) = 0.5*(AX+BX)
LINE 88    FIND CX,DX THE X COORD ON TRAILING EDGE
LINE 89    CX = X2 + I*DXTE
LINE 90    DX = CX - DXTE
LINE 91    DXM = DTEST*DC1
LINE 92    CAM = CX - DXM
LINE 93    DEX = DX - DXM
LINE 94    DXM = 0X - DXM
LINE 95    LOOP ON NC1 FOR AREA 1
LINE 97    DO 20 J1 = 1,NC1
LINE 98    NE=NE+1
LINE 99    XYZ(4JG1)=AY
LINE 100   NE=NE+1
LINE 101   XYZ(4JG6)=BY
LINE 102   XYZ(4JG6)=BY
LINE 103   XD = J1*OC1
LINE 104   XC = BX & XD
LINE 105   XD = AX & XD
LINE 106   XX = XD - OC1
LINE 107   XB = XC - DC1
LINE 108   XYZ(4JG1)=AX
LINE 109   XYZ(4JG6)=XB
LINE 110   XYZ(4JG6)=BX
LINE 111   XYZ(4JG6)=BX
LINE 112    CHECK TO SEE IF WE HAVE REACHED TRAILING EDGE (CX AND DX)
LINE 113    IF( XC .GT. CX .OR. XD .GT. DX ) GC TO 19
LINE 114    GO TO 20
LINE 115    19 CONTINUE
LINE 116    XYZ(4JG6)=CX
LINE 117    XYZ(4JG6)=DX
LINE 118    GO TO 60
LINE 119 20 CONTINUE
LINE 120 C
LINE 121 60 CONTINUE
LINE 122 NCH(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,X,Y,Z,TRS,NNCH,SW,XLE)

PURPOSE  GENERATE MESH FOR 1 SECTION WITH CONTROL LINES

INPUT

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

IN GLOBAL SYSTEM

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

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

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

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

EARO  ELEMENT ASPECT RATIO

OUTPUT

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

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

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

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

ELEM IN SPAN. (USED FOR LABELING PLOT)

SW  SPAN WIDTH

XLE  AVERAGE X COORDINATE ALONG LEADING EDGE

NE  NUMBER OF ELEMENTS IN SECTION.

NETSV  COUNTER OF TOTAL NUMBER OF ELEMENTS IN STRUCTURE

DIMENSION CL(1)

DIMENSION CORNX(1), CORNY(1), CORNZ(1), XLE(1), Y(1)

DIMENSION XLE(1), SW(1)

COMMON /BASIC/MACH,BEETA

EQUALVENCE (X(1), X1), (X(2), X2), (X(3), X3), (X(4), X4)

EQUIVALENCE (Y(1), Y1), (Y(2), Y2), (Y(3), Y3), (Y(4), Y4)

DATA OTEST / .98 /

SECTION IS DIVIDED INTO 3 AREAS

AREA 1 BETWEEN LEADING EDGE AND 1ST CONTROL LINE

AREA 2 BETWEEN FIRST AND 2ND CONTROL LINES

AREA 3 BETWEEN CONTROL LINE AND TRAILING EDGE

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

NOTE THAT ELEMENT IS NUMBERED CLOCKWISE WHILE SECTION IS COUNTED

DXLE  X INCREMENT A LONG LEADING EDGE

DXTE  X INCREMENT ALONG TRAILING EDGE

DXC1  X INCREMENT ALONG FIRST CONTROL LINE (LEADING C.LINE)

DXC2  X INCREMENT ALONG SECOND CONTROL LINE IF IT EXISTS

DX1,2,3  CHORD LENGTH IN AREAS 1, 2, AND 3

DX34  DISTANCE BETWEEN POINTS THREE AND FOUR

DX22  DISTANCE BETWEEN 1 AND 2 MINUS DX34

TRANSFORM CORNER POINTS TO LOCAL SYSTEM, WORK IN LOCAL SYSTEM

CALL TRANS(CORNX, CORNY, CORNZ, TRS, X, Y, Z)
LINE 57 C FIND CRATIO (CHORD TO SPAN RATIO) 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 = (E000*SQRT(DS*DS + DC*DC) / (DC*BEETA)) 01800620
LINE 63 C 01800630
LINE 64 C ASSUME CONSTANT SLOPE FOR ALL LINES 01800640
LINE 65 C 01800650
LINE 66 DX24 = X4 - X3 01800660
LINE 67 DX22 = X2 - X1 - DX24 01800670
LINE 68 DS = (Y3 - Y1) / NSP 01800680
LINE 69 DC = CRATIO*DS 01800690
LINE 70 DXLE = (X3 - X1) / NSP 01800700
LINE 71 DXTE = (X4 - X2) / NSP 01800710
LINE 72 C 01800720
LINE 73 C FIND INFORMATION FOR AREA BETWEEN LEADING EDGE AND 1ST CL 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 FIND INFO FOR AREA BETWEEN CONTROL LINES 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 01800930
LINE 94 DC3 = AMAX1( CL3-X2, CL4-X4 ) 01800940
LINE 95 NC2 = 1 01800950
LINE 96 DC2 = AMAX1( CL3-CL2, 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 01801040
LINE 105 DC3 = AMAX1( CL2-X1, CL4-X2 ) 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
LINE 119 C DO 100 I=1,NSP
LINE 120          018011190
LINE 121 C          01801200
LINE 122 C FIND THE Y COORD. FOR THIS SPAN (AY AND BY)          01801220
LINE 123 BY = Y1 + I*DS
LINE 124          01801240
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*(AX+BX)
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 TD1 = DTEST*DC1
LINE 138 TC1 = BCL1 - TD1
LINE 139 TD1 = ACL1 - TD1
LINE 140 C
LINE 141 DXM = DTEST*DC3
LINE 142 CXM = CX - DXM
LINE 143 DXM = DX - DXM
LINE 144 C LOOP ON NCI FOR AREA I
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)=XA
LINE 158 XYZ(NJ&8)=XB
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 IF(XC.GT.BCL1 .OR. XD .GT. ACL1) GO TO 19
LINE 163 GO TO 20
LINE 164 CONTINUE
LINE 165 XYZ(NJ&11)=BCL1
LINE 166 XYZ(NJ&12)=ACL1
LINE 167 GO TO 25
LINE 168 C
LINE 169 C 20 CONTINUE
LINE 170 C
LINE 171 C 25 CONTINUE
LINE 172 C IFI ICL .EQ. 1 I GO TO 40
LINE 173 GO TO 19
LINE 174 C SECTION FOR AREA 2
LINE 175 C
LINE 176 C
LINE 177 C FIND ACL2, BCL2 X COORD ON SECOND CONTROL LINE
LINE 178 BCL2 = CL2 + I*DXC2
LINE 179 ACL2 = BCL2 - DXC2
LINE 180 C
LINE  181  TD2 = DTEST+DC2
LINE  182  TC2 = ACL2 - TD2
LINE  183  TD2 = ACL2 - TD2
LINE  184  C LOOP ON NC2 FOR AREA 2
LINE  185  DO 30 J1 = 1,NC2
LINE  186  C
LINE  187  C
LINE  188  NJ = 6*NE
LINE  189  NE=NE+1
LINE  190  XYZ(J5)=A1
LINE  191  XYZ(J6)=A2
LINE  192  XD = J1*DC2
LINE  193  XC = BCL1 & XD
LINE  194  XD = ACL1 & XD
LINE  195  XA = XD - DC2
LINE  196  XB = XC - DC2
LINE  197  XYZ(J61)=XA
LINE  198  XYZ(J62)=XB
LINE  199  XYZ(J63)=XC
LINE  200  XYZ(J64)=XD
LINE  201  C CHECK TO SEE IF WE HAVE REACHED SECOND CONTROL LINE
LINE  202  IF (XC .GT. BCL2 .OR. XD .GT. ACL2) GO TO 29
LINE  203  GO TO 30
LINE  204  29 CONTINUE
LINE  205  XYZ(J63)=BCL2
LINE  206  XYZ(J64)=ACL2
LINE  207  GO TO 45
LINE  208  30 CONTINUE
LINE  209  GO TO 45
LINE  210  40 CONTINUE
LINE  211  40 CONTINUE
LINE  212  C THIS SECTION IS REACHED IFF THERE WAS ONLY 1 CONTROL LINE
LINE  213  C ACL1,BCL1 ARE PUT INTO ACL2,BCL2 AND AREA 3 IS ALWAYS TREATED
LINE  214  C AS THE PART BETWEEN 2ND CONTROL LINE AND LEADING EDGE
LINE  215  ACL2 = ACL1
LINE  216  BCL2 = BCL1
LINE  217  45 CONTINUE
LINE  218  C LOOP ON NC3 FOR AREA 3
LINE  219  C
LINE  220  DO 50 J1 = 1,NC3
LINE  221  C
LINE  222  C
LINE  223  NJ = 6*NE
LINE  224  NE=NE+1
LINE  225  XYZ(J5)=A1
LINE  226  XYZ(J6)=A2
LINE  227  XYZ(J61)=A2
LINE  228  XYZ(J62)=A1
LINE  229  XD = J1*DC3
LINE  230  XC = BCL2 & XD
LINE  231  XD = ACL2 & XD
LINE  232  XB = XC - DC3
LINE  233  XA = XD - DC3
LINE  234  XYZ(J61)=X1
LINE  235  XYZ(J62)=XB
LINE  236  XYZ(J63)=XC
LINE  237  XYZ(J64)=XD
LINE  238  C CHECK TO SEE IF WE REACHED TAILING Edge (CX,DX)
LINE  239  IF (XC .GT. CTX .OR. XD .GT. CDTX) GO TO 49
LINE  240  GO TO 50
LINE  241  49 CONTINUE
LINE  242  XYZ(J63)=CX

-6.50-
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>243</td>
<td>XYZ = 0x</td>
</tr>
<tr>
<td>244</td>
<td>GO TO 60</td>
</tr>
<tr>
<td>245</td>
<td>50 CONTINUE</td>
</tr>
<tr>
<td>246</td>
<td>C</td>
</tr>
<tr>
<td>247</td>
<td>60 CONTINUE</td>
</tr>
<tr>
<td>248</td>
<td>NNCH(I) = NE &amp; NETSV</td>
</tr>
<tr>
<td>249</td>
<td>C</td>
</tr>
<tr>
<td>250</td>
<td>100 CONTINUE</td>
</tr>
<tr>
<td>251</td>
<td>C</td>
</tr>
<tr>
<td>252</td>
<td>C UPDATE COUNTER OF TOTAL ELEMENTS FOR STRUCTURE</td>
</tr>
<tr>
<td>253</td>
<td>NETSV = NETSV &amp; NE</td>
</tr>
<tr>
<td>254</td>
<td>RETURN</td>
</tr>
<tr>
<td>255</td>
<td>END</td>
</tr>
</tbody>
</table>

END

-6.51-
DISPLAY...FILE

LINE 1 SUBROUTINE MFUN (JMODE) 01900010
LINE 2 COMMON /MODE1/, EM(10), DM(10), X, Y, J1, J2 01900020
LINE 3 DIMENSION CTABLE(6,20), IFTABL(2,20) 01900030
LINE 4 COMMON /COM/, CTABLE, IFTABL, NTABL, NEQF 01900040
LINE 5 DO 200 IM = 1, JMODE 01900050
LINE 6 EM(IM) = 0.0 01900060
LINE 7 DM(IM) = 0.0 01900070
LINE 8 DO 100 I = 1, NEQF 01900080
LINE 9 IFM = IFTABL(1, I) 01900090
LINE 10 IFM = IFTABL(2, I) 01900100
LINE 11 IF(EM.EQ.J1) .AND. (IFM.EQ.IM) GO TO 90 01900110
LINE 12 GO TO 100 01900120
LINE 13 90 CO = CTABLE(1, I) 01900130
LINE 14 CX = CTABLE(2, I) 01900140
LINE 15 CY = CTABLE(3, I) 01900150
LINE 16 CXY = CTABLE(4, I) 01900160
LINE 17 CX2 = CTABLE(5, I) 01900170
LINE 18 CY2 = CTABLE(6, I) 01900180
LINE 19 EM(IM) = CO & CX * X & CY * Y & CXY * X * Y & CX2 * X**2 & CY2 * Y**2 01900190
LINE 20 DM(IM) = CX & CXY * Y & 2.0 * CX2 * X 01900200
LINE 21 GO TO 200 01900210
LINE 22 100 CONTINUE 01900220
LINE 23 200 CONTINUE 01900230
LINE 24 RETURN 01900240
LINE 25 END 01900250

DECK 19

-6.52-
DISPLAY FILE

DECK 20

LINE 1  SUBROUTINE MXERR(IERR,MXNO,NOIN,IG)
LINE 2   WRITE (16,700) 02000010
LINE 3   IF(IERR .EQ. 610) WRITE(16,710) MXNC, NOIN 02000030
LINE 4           WRITE(16,720) MXNC, NOIN 02000040
LINE 5           WRITE(16,730) MXNC, NOIN 02000050
LINE 6   IF(IERR .EQ. 640) WRITE(16,740) MXNC, NOIN 02000060
LINE 7           WRITE(16,750) MXNC, NOIN 02000070
LINE 8           WRITE(16,760) MXNC, NOIN 02000080
LINE 9           WRITE(16,770) MXNC, NOIN 02000090
LINE 10  IF(IERR .EQ. 680) WRITE(16,780) MXNC, NOIN 02000100
LINE 11  700 FORMAT(1HO //8X, 42HERROR IN INPUT, MAXIMUM ALLOWED EXCEEDED) 02000110
LINE 12  710 FORMAT(1HO / 8X, 20HMAXIMUM NO. WINGS = 113,6X, 4H NO. 02000120
LINE 13   1, 9H INPUT = 113 ) 02000130
LINE 14  720 FORMAT(1HO / 8X, 30HMAXIMUM NO. SECTIONS / WING = 111, 6X, 4H NO. 02000140
LINE 15   1, 9H INPUT = 113 ) 02000150
LINE 16  730 FORMAT(1HO / 8X, 20HMAXIMUM NO. MODES = 113,6X, 4H NO. 02000160
LINE 17   1, 9H INPUT = 113 ) 02000170
LINE 18  740 FORMAT(1HO / 8X, 20HMAXIMUM NO. SPANS = 113,6X, 4H NO. 02000180
LINE 19   1, 9H INPUT = 113 ) 02000190
LINE 20  750 FORMAT(1HO / 8X, 20HMAXIMUM NO. CORNER POINTS = 111, 6X, 4H NO. 02000200
LINE 21   1, 9H INPUT = 113 ) 02000210
LINE 22  760 FORMAT(1HO / 8X, 20HMAXIMUM NO. FREQUENCIES / SPAN = 112, 6X, 4H NO. 02000220
LINE 23   1, 9H INPUT = 113 ) 02000230
LINE 24  770 FORMAT(1HO / 8X, 30HMAXIMUM NO. POINTS ON SPLINE DATA = 113, 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-
SUBLTINE OUTI1( MACH, FREQ, JMODE, JWING, NS, NSP, SW, DLSPAN, 02100010
PMSPan, XP, INDEX1, PRES, IPW, 16, LINE, LMAX, 02100020
1, NNCH, NE, TITLE, PHIW, NSPT )
02100030
02100040
02100050
02100060
MACH MACH NUMBER
02100070
02100080
FREQ FREQUENCY
02100090
02100100
02100110
02100120
JWING WING NUMBER
02100130
02100140
NS NUMBER OF SECTIONS IN JWING
02100150
02100160
02100170
02100180
02100190
02100200
02100210
02100220
02100230
02100240
NSP NUMBER OF SPANS IN EACH SECTION OF JWING (ARRAY)
02100250
DLSPAN LIFT/UNIT SPAN FOR EACH SPAN OF JWING (ARRAY)
02100260
02100270
PMSPan PITCHING MOMENT/UNIT SPAN FOR EACH SPAN (ARRAY)
02100280
02100290
02100300
02100310
02100320
02100330
02100340
02100350
02100360
02100370
02100380
02100390
02100400
02100410
02100420
02100430
02100440
02100450
02100460
02100470
02100480
02100490
02100500
02100510
02100520
02100530
02100540
02100550
02100560

real MACH,
line 31 complex
line 32 complex
line 33 dimension
line 34 dimension
line 35 dimension
line 36 c
line 37 do 15 kspct = 1,nspt
line 38 if ( nnch(kspct) .ge. index1 ) go to 17
line 39 15 continue
line 40 17 continue
line 41 c
line 42 if ( line .ne. lmax+1 ) go to 25
line 43 write (16, 205) title, freq
line 44 line = 6
line 45 c
line 46 25 continue
line 47 c
line 48 if ( ipw .eq. 2 ) go to 123
line 49 c
line 50 c
write lift and pressure moment table
line 51 c
line 52 c
line 53 if ( lmax - line .gt. 14 ) go to 30
line 54 write (16, 202) line = 7
goto 35

-6.54-
LINE 57  30 CONTINUE
LINE 58  WRITE(16, 201) 1 # JWING
LINE 59  LINE = LINE & 12
LINE 60  35 CONTINUE
LINE 61  WRITE(16, 201) 1 # JWING
LINE 62  C
LINE 63  K = 1
LINE 64  ISPCT = KSPCT
LINE 65  WRITE(16, 202) JMODE, FREQ, MACH
LINE 66  WRITE(16, 203) XP
LINE 67  C LOOP ON SECTIONS
LINE 68  DO 120 ISECT = 1, NS
LINE 69  NSPAN = NSP(ISECT)
LINE 70  C LOOP ON SPANS OF SECTION
LINE 71  DO 115 ISPAN = 1, NSPAN
LINE 72  WRITE(16, 204) ISECT, ISPAN, SW(ISPCT), DSLSPAN(ISPCT), PMSPAN(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 WRITE ELEMENT PRESSURE TABLE
LINE 84  C
LINE 85  C CALCULATE THE NUMBER OF ELEMENTS IN JWING 'NEWJ'
LINE 86  C
LINE 87  88 NEWJ = 0
LINE 88  DO 125 I = 1, NS
LINE 89  125 NEWJ = NEWJ + NE(I)
LINE 90  C
LINE 91  INDXR = INDX & NEWJ - 1
LINE 92  C IF( LMAX - LINE .GT. 14 ) GO TO 130
LINE 93  WRITE(16, 205)
LINE 94  LINE = 7
LINE 95  GO TO 135
LINE 96  C
LINE 97  130 CONTINUE
LINE 98  WRITE(16, 206)
LINE 99  LINE = LINE & 12
LINE 100  135 CONTINUE
LINE 101  WRITE(16, 207) 1 # JWING
LINE 102  C
LINE 103  WRITE(16, 208) JMODE, FREQ, MACH
LINE 104  WRITE(16, 209)
LINE 105  INDX = INDXR
LINE 106  ISPCT = KSPCT
LINE 107  C LOOP ON SECTIONS
LINE 108  DO 220 ISECT = 1, NS
LINE 109  NSPAN = NSP(ISECT)
LINE 110  C LOOP ON SPANS OF SECTION
LINE 111  DO 215 ISPAN = 1, NSPAN
LINE 112  NEPS = NNCH(ISPCT) - INDXR + 1
LINE 113  DO 210 IEL = 1, NEPS
LINE 114  WRITE(16, 209) ISECT, ISPAN, INDXR, PHIW(INDXR), PRES(INDXR)
LINE 115  INDXR = INDXR + 1
LINE 116  210 CONTINUE
LINE 117  ISPCT = ISPCT + 1
LINE 118  215 CONTINUE
LINE 119  CONTINUE
LINE 120  C
LINE 121  LINE = LINE & NEWJ
LINE 122  LINE = NEWJ/3 & LINE
LINE 123  IF ( MOD(NEWJ,3) .GT. 0 ) LINE=LINE+1
LINE 124  LINE = MOD LINE, LMAX)
LINE 125  C
LINE 126  RETURN
LINE 127  2001 FORMAT( 46X, 47H LIFT AND PITCHING MOMENT PER UNIT SPAN FOR WING)
LINE 128  1NG, 12 )
LINE 129  2002 FORMAT( 1H5, 5X, 11H MODE NUMBER, 13 / 6X, 11H FREQUENCY =, F7.4, / )
LINE 130  1 22X, 13H MAC NUMBER =, F7.4 /
LINE 131  2003 FORMAT( 5X, 31H SECTION SPAN SPAN WIDTH, 
LINE 132  1 8X, 27H LIFT PER UNIT SPAN ----, / )
LINE 133  2 7X, 33HPITCHING MOMENT PER UNIT SPAN ABOUT X=, E12.6 / )
LINE 134  2004 FORMAT( 15X, 5X, E12.6, 2( 7X, E12.6, 3X, E12.6, 4X ) ) )
LINE 135  2005 FORMAT( 46X, 50H VELOCITY POTENTIALS AND ELEMENT PRESSURES FOR WING)
LINE 136  1 12 )
LINE 137  2007 FORMAT( 36X,7HELEMENT,3X,8(1H-1,10H PRESSURE,9(1H-1 ) / )
LINE 138  2008 FORMAT( 31 6X, 17, 3X, E12.6, 3X, E12.6, 3X )
LINE 139  2009 FORMAT( 6X, 27H SECTION SPAN ELEMENT, 11X, 
LINE 140  1 27H VELOCITY POTENTIAL ----, 11X, 8(1H-1, 13H PRESSURE )
LINE 141  1 2( 9(1H-1 ) / )
LINE 142  2010 FORMAT( 6X, 217, 113, 2( 11X, E12.6, 3X, E12.6 ) )
LINE 143  2015 FORMAT( 1H4, 45X, 1444 // 46X, 11H FREQUENCY =, F7.4//1X, 65(2H ) )
LINE 144  2020 FORMAT( 1H1 )
LINE 145  2021 FORMAT( // // // // )
LINE 146  END

-6.56-
DISPLAY FILE DECK 22

LINE 1  SUBROUTINE OLTP2( I6, NWING, XP, TL, TM, JMODE, FREQ, MACH, LINE, LMAX ) 02200010
LINE 2  COMPLEX TL(4,10), TM(4,10) 02200020
LINE 3  REAL MACH 02200030
LINE 4  C   TL TOTAL LIFT PER WING 02200040
LINE 5  C   TM TOTAL MOMENT PER WING 02200050
LINE 6  C   WRITE TABLE OF TOTAL LIFT AND PITCHING MOMENTS 02200060
LINE 7  C   WRITE( I6, 2020 ) 02200070
LINE 8  C   LINE = 7 02200080
LINE 9  C   IF( LMAX - LINE .GT. 14 ) GO TO 30 02200090
LINE 10  WRITE( I6, 2021 ) 02200100
LINE 11  WRITE( I6, 2022 ) 02200110
LINE 12  WRITE( I6, 2023 ) 02200120
LINE 13  WRITE( I6, 2024 ) 02200130
LINE 14  WRITE( I6, 2025 ) 02200140
LINE 15  WRITE( I6, 2026 ) 02200150
LINE 16  WRITE( I6, 2027 ) 02200160
LINE 17  WRITE( I6, 2028 ) 02200170
LINE 18  WRITE( I6, 2029 ) 02200180
LINE 19  WRITE( I6, 2030 ) 02200190
LINE 20  WRITE( I6, 2031 ) 02200200
LINE 21  WRITE( I6, 2032 ) 02200210
LINE 22  WRITE( I6, 2033 ) 02200220
LINE 23  WRITE( I6, 2034 ) 02200230
LINE 24  WRITE( I6, 2035 ) 02200240
LINE 25  WRITE( I6, 2036 ) 02200250
LINE 26  WRITE( I6, 2037 ) 02200260
LINE 27  WRITE( I6, 2038 ) 02200270
LINE 28  WRITE( I6, 2039 ) 02200280
LINE 29  WRITE( I6, 2040 ) 02200290
LINE 30  WRITE( I6, 2041 ) 02200300
LINE 31  WRITE( I6, 2042 ) 02200310
LINE 32  WRITE( I6, 2043 ) 02200320
LINE 33  WRITE( I6, 2044 ) 02200330
LINE 34  WRITE( I6, 2045 ) 02200340
LINE 35  WRITE( I6, 2046 ) 02200350
LINE 36  WRITE( I6, 2047 ) 02200360
LINE 37  WRITE( I6, 2048 ) 02200370
LINE 38  WRITE( I6, 2049 ) 02200380
LINE 39  WRITE( I6, 2050 ) 02200390

-6.57-
DISPLAY...FILE

DECK 23

LINE 1     SUBROUTINE OUTP3 (16, MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE) 03300010
LINE 2     REAL MACH, Q, LINE, LMAX, TITLE 03300020
LINE 3     REAL X, Y, Z, C, D 03300030
LINE 4     REAL MACH, Q, LINE, LMAX, TITLE 03300040
LINE 5     REAL X, Y, Z, C, D 03300050
LINE 6     REAL X, Y, Z, C, D 03300060
LINE 7     WRITE (16, 2010) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300070
LINE 8     WRITE (16, 2020) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300080
LINE 9     WRITE (16, 2030) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300090
LINE 10    WRITE (16, 2040) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300100
LINE 11    WRITE (16, 2050) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300110
LINE 12    WRITE (16, 2060) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300120
LINE 13    WRITE (16, 2070) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300130
LINE 14    WRITE (16, 2080) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300140
LINE 15    WRITE (16, 2090) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300150
LINE 16    WRITE (16, 2100) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300160
LINE 17    WRITE (16, 2110) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300170
LINE 18    WRITE (16, 2120) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300180
LINE 19    WRITE (16, 2130) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300190
LINE 20    WRITE (16, 2140) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300200
LINE 21    WRITE (16, 2150) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300210
LINE 22    WRITE (16, 2160) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300220
LINE 23    WRITE (16, 2170) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300230
LINE 24    WRITE (16, 2180) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300240
LINE 25    WRITE (16, 2190) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300250
LINE 26    WRITE (16, 2200) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300260
LINE 27    WRITE (16, 2210) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300270
LINE 28    WRITE (16, 2220) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300280
LINE 29    WRITE (16, 2230) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300290
LINE 30    WRITE (16, 2240) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300300
LINE 31    WRITE (16, 2250) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300310
LINE 32    WRITE (16, 2260) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300320
LINE 33    WRITE (16, 2270) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300330
LINE 34    WRITE (16, 2280) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300340
LINE 35    WRITE (16, 2290) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300350
LINE 36    WRITE (16, 2300) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300360
LINE 37    WRITE (16, 2310) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300370
LINE 38    WRITE (16, 2320) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300380
LINE 39    WRITE (16, 2330) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300390
LINE 40    WRITE (16, 2340) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300400
LINE 41    WRITE (16, 2350) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300410
LINE 42    WRITE (16, 2360) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300420
LINE 43    WRITE (16, 2370) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300430
LINE 44    WRITE (16, 2380) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300440
LINE 45    WRITE (16, 2390) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300450
LINE 46    WRITE (16, 2400) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300460
LINE 47    WRITE (16, 2410) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300470
LINE 48    WRITE (16, 2420) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300480
LINE 49    WRITE (16, 2430) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300490
LINE 50    WRITE (16, 2440) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300500
LINE 51    WRITE (16, 2450) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300510
LINE 52    WRITE (16, 2460) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300520
LINE 53    WRITE (16, 2470) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300530
LINE 54    WRITE (16, 2480) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300540
LINE 55    WRITE (16, 2490) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300550
LINE 56    WRITE (16, 2500) MACH, FREQ, JMODE, Q, LINE, LMAX, TITLE 03300560

-6.58-
I.

LINE 57  \texttt{2 10X,13.6H MODES, 27X, 13HMACH NUMBER =, F7.4 )}  \texttt{02300570}
LINE 58  \texttt{2011 FORMAT( / 10X, 12+DISPLACEMENT / 13X, 4HMODE, 8X, \texttt{02300580}}
LINE 59  \texttt{1 18H---- PRESSURE MODE, 13, 5H ---- )}  \texttt{02300590}
LINE 60  \texttt{2012 FORMAT( 19X, 13, 3X, F12.6, 2X, F12.6 )}  \texttt{02300600}
LINE 61  \texttt{2015 FORMAT( 1m1, 45X, 14A4 // 46X,11HFREQUENCY =, F7.4//1X, 6512H LI}}  \texttt{02300610}
LINE 62  \texttt{2020 FORMAT( 1H1 )}  \texttt{02300620}
LINE 63  \texttt{2021 FORMAT( // // )}  \texttt{02300630}
LINE 64  \texttt{2022 FORMAT( )}  \texttt{02300640}
LINE 65  \texttt{2023 FORMAT( 1HE, 63X, 18H---- PRESSURE MODE, 13, 5H ---- )}  \texttt{02300650}
LINE 66  \texttt{2024 FORMAT( 1HE, 97X, 18H---- PRESSURE MODE, 13, 5H ---- )}  \texttt{02300660}
LINE 67  \texttt{END}  \texttt{02300670}
LINE 1  SUBROUTINE PHIL (XLE, XCEN, DPDX, NEPS, PHI, A, CXP, PHIW) 02400010
LINE 2  COMPLEX B1, BO, AA, XI, XIP1, EX 02400020
LINE 3  COMPLEX CI, C, XDIFS 02400030
LINE 4  COMPLEX EAXI, EAXIP1, AXI, AXIP1, SUM 02400040
LINE 5  COMPLEX PHI(1), DPDX(1), A, CXP(1) 02400050
LINE 6  COMPLEX PHIW(1) 02400060
LINE 7  LOGICAL WAKE1, WAKE, WAKENZ, TREDGE 02400070
LINE 8  DIMENSION XCN(1) 02400080
LINE 9  COMMON /WAKE/ ITW, ITH, ICNVGW, EPSW, JCNVGW 02400090
LINE 10 COMMON /WAKEUP/ ITWES(12), TREDGE, WAKE1, WAKE, WAKENZ 02400100
LINE 11 DATA C1 / (1.0, 0.0) / 02400110
LINE 12 IF (NEPS .EQ. 1) GO TO 40 02400120
LINE 13 XI = CMPLX(XLE, 0.0) 02400130
LINE 14 XIP1 = CMPLX(XCEN(2), 0.0) 02400140
LINE 15 CX = CMPLX((XCEN(1)-XCEN(2)), 0.0) 02400150
LINE 16 B1 = (DPDX(1) - DPDX(2))/CX 02400160
LINE 17 BO = DPDX(2) - B1*XIP1 02400170
LINE 18 IF (CABS(A) .LT. 1.0E-20) GO TO 23 02400180
LINE 19 XI = XIP1 02400190
LINE 20 DO 10 IP1=2, NEPS 02400200
LINE 21 XI = CMPLX(XCEN(IP1), 0.0) 02400210
LINE 22 CX = CMPLX((XCEN(I)-XCEN(IP1)), 0.0) 02400220
LINE 23 B1 = (DPDX(I) - DPDX(IP1))/CX 02400230
LINE 24 BO = DPDX(IP1) - B1*XIP1 02400240
LINE 25 XI = XIP1 02400250
LINE 26 AXI1 = AXI1 02400260
LINE 27 EAXI1 = CXP(AXI1) 02400270
LINE 28 AXIP1 = CXP(AXI1) 02400280
LINE 29 EX = CI/EAXI1 02400290
LINE 30 SUM = (B1/AA)*(EAXI1 - CI) 02400300
LINE 31 SUM1 = SUM*(BO/A)*(EAXI1 - EX) 02400310
LINE 32 PHI(1) = SUM*EX 02400320
LINE 33 XI = XIP1 02400330
LINE 34 CX = CMPLX((XCEN/IP1), 0.0) 02400340
LINE 35 B1 = (DPDX(I) - DPDX(IP1))/CX 02400350
LINE 36 BO = DPDX(IP1) - B1*XIP1 02400360
LINE 37 XI = XIP1 02400370
LINE 38 DO 10 IP1=2, NEPS 02400380
LINE 39 I = IP1 - 1 02400390
LINE 40 XIP1 = CMPLX((XCEN/IP1), 0.0) 02400400
LINE 41 CX = CMPLX((XCEN(I)-XCEN/IP1), 0.0) 02400410
LINE 42 B1 = (DPDX(I) - DPDX/IP1)/CX 02400420
LINE 43 BO = DPDX/IP1 - B1*XIP1 02400430
LINE 44 XI = XIP1 02400440
LINE 45 AXI = AXI1 02400450
LINE 46 EAXI = EAXI1 02400460
LINE 47 AXIP1 = AXIP1 02400470
LINE 48 EAXIP1 = EAXIP1 02400480
LINE 49 EX = CI/EAXIP1 02400490
LINE 50 SUM = SUM*(B1/AA)*(EAXIP1 - CI) 02400500
LINE 51 SUM1 = SUM*(BO/A)*(EAXIP1 - EX) 02400510
LINE 52 PHI/IP1) = SUM*EX 02400520
LINE 53 XI = XIP1 02400530
LINE 54 AXI = AXI1 02400540
LINE 55 EAXI = EAXI1 02400550
LINE 56 PHI/IP1) = SUM*EX 02400560
LINE 57  10 CONTINUE
LINE 58  C
LINE 59    GO TO 32
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 = B0*CX & B1*XDIFS
LINE 64    PHI(1) = SUM
LINE 65  DO 30 IP1=2,NEPS
LINE 66    I = IP1 - 1
LINE 67    CX = CMPLX(XCEN(II)-XCEN(IP1)), 0.0
LINE 68    BI = (DPDX(IP1)-DPDX(IP1))/CX
LINE 69    BO = DPOX(IP1) - BI*XP1
LINE 70    XDIFS = CMPLX(0.5*(XCEN(I11)*XCEN(IP11)-XCEN(IP1)*XCEN(IP1)), 0.0)
LINE 71    SUM = SUM & B0*CX & B1*XDIFS
LINE 72    PHI(IP11) = SUM
LINE 73  30 CONTINUE
LINE 74    GO TO 52
LINE 75  C
LINE 76  40 CONTINUE
LINE 77  C SECTION FOR ONLY ONE ELEMENT PER SPAN
LINE 78  C
LINE 79  C
LINE 80    CX = CMPLX(XCEN(1)-XLE), 0.0
LINE 81    IF(CABS(A1) .LT. 1.0E-20) GO TO 50
LINE 82    PHI(1) = (DPDX(1)/A)*(C1 - CEXP(1-A*CX))
LINE 83    GO TO 52
LINE 84  50 PHI(1) = DPOX(1)*CX
LINE 85  52 ICNVGW = 0
LINE 86  C INITIALIZE ICNVGW = 0
LINE 87  C ICNVGW = 1 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 ITW = 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(ITW .EQ. 0) GO TO 58
LINE 93    IF(.NOT. WAKE) GO TO 58
LINE 94    ICNVGW = IHERE(PHI, PHIW, NEPS)
LINE 95    IF(ICNVGW .EQ. 1) ICNVGW = ICNVGW & NEPS
LINE 96  58 CONTINUE
LINE 97  DD 60 IP1=1,NEPS
LINE 98    IF(ITW .EQ. 0) GO TO 60
LINE 99    IF(.NOT. WAKE) GO TO 60
LINE 100  ICNVGW = IHERE(PHI(IP1)), PHIW(IP1), ICNVGW
LINE 101  C ICNVGW .EQ. 0) ICNVGW = ICNVGW & 1
LINE 102  60 PHIW(IP1) = PHI(IP1)
LINE 103  62 RETURN
LINE 104  END
DISPLAY...FILE
DECK 25

LINE 1  SUBROUTINE PLOT(X,Y,IPEN)
LINE 2    IF( IPEN.EQ. -23) GO TO 100
LINE 3    IF( IPEN.EQ. 23) IPEN = -3
LINE 4    CALL CALPLT(X,Y,IPEN)
LINE 5    RETURN
LINE 6   100  SX = X + 2.0
LINE 7    CALL NFRA...
SUBROUTINE PLOTGD ( NE, XYZ, NNCH, NSP, NSECT, TITLE, JWING, ZSECT, AR, MACH )
DIMENSION NE(1), ZSECT(1), NSP(1)
DIMENSION PH(7), PV(7)
DIMENSION XYZ(1), NNCH(1), TITLE(1)
DATA FGDX, FGDY / 15.0, 10.0 /
DATA HT, HTW, HHT / .07, .05202, .035 /
DATA NETSV / 1 /
REAL MACH
DIMENSION NE(11), ZSECT(1), NSP(1)
DATA NE(1), NE(1), NE(1) / 1 /
IFS J WING .EQ. 1 NETSV = 1
NEPSWT = 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
XMIN = AMIN1(XMIN, XYZ(K))
XMAX = AMAX1(XMAX, XYZ(K))
K = K + 1
CONTINUE
DO 3 J = 1,4
XMIN = AMIN1(XMIN, XYZ(K))
XMAX = AMAX1(XMAX, XYZ(K))
K = K + 1
CONTINUE
K = K + 1
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 REDRAW
LINE 57 C SECTION TO PUT TITLE AND HEADING ON PLOT 02600570
LINE 58 C 02600580
LINE 59 C 02600590
LINE 60 C Y = 0.25 02600600
LINE 61 DO 12 I = 1, 14 02600610
LINE 62 CALL SYMBOL ( -1.5, Y, 0.21, TITLE(I), 90.0, 4 ) 02600620
LINE 63 12 Y = Y & 0.728 02600630
LINE 64 C 02600640
LINE 65 CALL SYMBOL ( -1.15, 2.25, 0.21, 14HASPECT RATIO =, 90.0, 14 ) 02600650
LINE 66 CALL NUMBER ( -1.15, 2.98, 2.21, AR, 90.0, 3 ) 02600660
LINE 67 CALL SYMBOL ( -1.15, 4.7, 2.21, 14MACH NUMBER =, 90.0, 13 ) 02600670
LINE 68 CALL NUMBER ( -1.5, 7.1, 2.21, MACH, 90.0, 3 ) 02600680
LINE 69 CALL SYMBOL ( -0.80, 2.25, 0.21, 14SECTION NUMBER, 90.0, 14 ) 02600690
LINE 70 FPN = FLLOAT(I) 02600700
LINE 71 CALL NUMBER ( -0.80, 2.98, 2.21, FPN, 90.0, -1 ) 02600710
LINE 72 CALL SYMBOL ( -0.80, 3.52, 2.21, 7HOF WING, 90.0, 7 ) 02600720
LINE 73 FPN = FLOAT(WNG) 02600730
LINE 74 CALL NUMBER ( -0.80, 4.97, 2.21, FPN, 90.0, -1 ) 02600740
LINE 75 CALL SYMBOL ( -0.80, 6.33, 2.21, 3HZ =, 90.0, 3 ) 02600750
LINE 76 FPN = ZSECT(I) 02600760
LINE 77 CALL NUMBER ( -0.80, 7.06, 2.21, FPN, 90.0, 3 ) 02600770
LINE 78 C 02600780
LINE 79 C SECTION TO PLOT ELEMENTS 02600790
LINE 80 C 02600800
LINE 81 K = KINDEX 02600810
LINE 82 DO 20 I = 1, NEPS 02600820
LINE 83 DO 19 J = 1, 4 02600830
LINE 84 PH(J) = XYZ(K) 02600840
LINE 85 IF ( J .GT. 2 ) GO TO 19 02600850
LINE 86 PV(I) = 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+2) 02600890
LINE 90 K = K & 2 02600900
LINE 91 PH(5) = PH(1) 02600910
LINE 92 PV(5) = PV(1) 02600920
LINE 93 C WRITE (6, 1003) I, ( PH(J), J=1,5), (PV(I), I=1,5) 02600930
LINE 94 CALL LINE ( PH, PV, 5, 1, 0, 0.01 02600940
LINE 95 20 CONTINUE 02600950
LINE 96 C 02600960
LINE 97 C SECTION TO LABEL ELEMENTS ON THE TRAILING EDGE 02600970
LINE 98 C 02600980
LINE 99 KLAST = NSP(I) & KLAST 02600990
LINE 100 DO 30 I = KFIRST, KLAST 02601000
LINE 101 C NETSV IS THE NUMBER OF ELEMENTS IN THE PREVIOUS WINGS PLUS 1 02601010
LINE 102 J = 6 * (NNCH(1) - NETSV) & 1 02601020
LINE 103 SFH = 0.25 * ( XYZ(J) & XYZ(J+61) & XYZ(J+62) & XYZ(J+63) ) 02601030
LINE 104 SFV = 0.5 * ( XYZ(J+64) & XYZ(J+65) ) 02601040
LINE 105 SFH = ( SFH - XMIN ) / SF - HHT 02601050
LINE 106 SFV = ( SFV - YMIN ) / SF - HTW 02601060
LINE 107 C (SFH, SFV) ARE COORDINATES OF CENTER OF ELEMENT 02601070
LINE 108 C TD CENTER CHARACTER, MOVE OFF CENTER BY (HTW, HTW) 02601080
LINE 109 GP = FLOAT(NNCH(I) ) 02601090
LINE 110 CALL NUMBER ( SFH, SFV, HT, GP, 90.0, -1 ) 02601100
LINE 111 30 CONTINUE 02601110
LINE 112 KFIRST = KFIRST & NSP(I) 02601120
LINE 113 C 02601130
LINE 114 C SECTION TO REDIGIN FOR NEXT PLOT AND RUMP BLOCK ADDRESS BY 02601140
LINE 115 C 02601150
LINE 116 SX = TMXX & 3.0 02601160
LINE 117 SX = AMAX1 ( SX, 17. ) 02601170
LINE 118 CALL PLOT( SX, -.25, -23 ) 02601180
LINE 119 C UPDATE COUNTER OF ELEMENTS IN WING
LINE 120 C NEPWT = NEPWT & NEPS
LINE 121 C KINDEX = KINDEX & 6*NEPS
LINE 122 C END LOOP ON SECTIONS
LINE 123 C 100 CONTINUE
LINE 124 C UPDATE ELEMENT COUNTER
LINE 125 C NETSV IS THE NUMBER OF ELEMENTS IN THE PREVIOUS WINGS PLUS 1
LINE 126 C NETSV = NETSV & NEPWT
LINE 127 C RETURN
LINE 128 C1002 FORMAT( // 'XMAX,XMIN', 2E14.4, 5X, 'YMAX,YMIN', ZE14.4 //
LINE 129 C3 FORMAT( // 'TMXX,TIMY', 2E14.4, 5X, 'SF', 1E14.4 //
LINE 130 C4 FORMAT( // 'PH FOLLOWED BY PV FOR EVERY ELEMENT' //)
LINE 131 C1003 FORMAT( // 15, 2I5, 5E14.4 )
LINE 132 C END
DISPLAY...FILE

DECK 27

LINE 1 SUBROUTINE PTGRID I6, TITLE, MACH, N'WING, NSECT, NSP, NNCH, XYZ, U270010
LINE 2 1 ZSECT, REFLEN, BEETA, XCEN J U2700120
LINE 3 REAL MACH U270030
LINE 4 DIMENSION NNCH(I), NSP(J), I U270040
LINE 5 DIMENSION TITLE(I), NSECT(I), XYZ(I), ZSECT(I), J U270050
LINE 6 DIMENSION XCEN(I) U270060
LINE 7 DIMENSION Xy(8) U270070
LINE 8 BL = BEETA*REFLEN U270080
LINE 9 WRITE (16,3000) TITLE, MACH U270090
LINE 10 WRITE(16,3001) KSPAN = 0 U270100
LINE 11 K = 0 U270110
LINE 12 N1 = 1 U270120
LINE 13 DO 10 J = 1, NWING U270130
LINE 14 NS = VSECT(J) U270140
LINE 15 DO 95 I = 1, NS U270150
LINE 16 WRITE (16,3002) J, I, ZSECT(I,J) U270160
LINE 17 NSPAN = NSP(I) U270170
LINE 18 DO 90 ISPAN = 1, NSPAN U270180
LINE 19 KSPAN = KSPAN + 1 U270190
LINE 20 NEP S = NNCH(KSPAN) - K U270200
LINE 21 DO 85 IE = 1, NEPS U270210
LINE 22 K = K + 1 U270220
LINE 23 IN = IN + 1 U270230
LINE 24 XY(I) = 0 U270240
LINE 25 IN = IN + 1 U270250
LINE 26 DO 80 JJ = 1, 4 U270260
LINE 27 XY(I,J) = XYZ(IN) + BL U270270
LINE 28 XY(I) = XY(I) + BL U270280
LINE 29 IN = IN + 1 U270290
LINE 30 80 CONTINUE U270300
LINE 31 XY(I) = XY(I)/4.0 U270310
LINE 32 XY(I) = XYZ(IN) + REFLEN U270320
LINE 33 XY(I) = XYZ(IN) + REFLEN U270330
LINE 34 XY(I) = 0.5*(XY(I) + XYZ(IN)) U270340
LINE 35 WRITE (16,3002) ISPAN, K, XY U270350
LINE 36 XCEN(K) = XY(I)/BL U270360
LINE 37 N1 = N1 + 6 U270370
LINE 38 85 CONTINUE U270380
LINE 39 90 CONTINUE U270390
LINE 40 95 CONTINUE U270400
LINE 41 100 CONTINUE U270410
LINE 42 RETURN U270420
LINE 43 3000 FORMAT(I11/35X,14A4/35X, 32HAERODYNAMIC GRID FOR MACH NJMBE, U270430
LINE 44 1 F8.4 / ) U270440
LINE 45 3001 FORMAT(I / 6H WING, 12, 8H SECTION, 12 / 5H L = *2X,E11.5/ U270450
LINE 46 1 14H SPAN ELEMENT, 4X, 2HX1, 12X, 2HX2, 12X, 2HK3, 12X, 2HX4, 2HX5, U270460
LINE 47 2 12X, 2HY1, 12X, 2HY2, 13X, 2HXC, 12X, 2HXC / ) U270470
LINE 48 3002 FORMAT(I5, 19, 1X, 61 3X, E11.5, 4X, E11.5, 3X, E11.5 ) U270480
LINE 49 3003 FORMAT(I ) 35X, 33STRUCTURAL REFERENCE SYSTEM / ) U270490
LINE 50 ITEM 0A / 35X, 35HRCTATED STRUCTURAL REFERENCE SYSTEM / ) U270500
LINE 51 END U270510

-6.66-
<table>
<thead>
<tr>
<th>LINE</th>
<th>Code</th>
<th>Description</th>
<th>Page 28 02800010</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUBROUTINE ROETA (113, DETADX, ETA, NET)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>DIMENSION DETADX(ETA), ETA(ETA)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>READ (113)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>JMI, NET1, DETADX, ETA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>RETURN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>END</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SUBROUTINE RMODEM JMODE, N, I JMG, NSECT, NE, Xcen, XYZ, BEETA, ETA, DETAX, DETAY, I, J, K, X, Y, Z, A, N4, IERR

LOGICAL ERROR

ETA AND DETAX DIMENSIONED (NUMBER OF ELEMENTS, NO. MODES)

DIMENSION CFTABL(N, M), IFTABL(N, M)

DIMENSION ETA(N, M), DETAX(N, M), DETAY(N, M)

DIMENSION ID(I), MODE(20), MODEF4(6)

DIMENSION NSECT(N, M), NE(I), XCEN(I), XYZ(I)

DIMENSION NUMBR(3), MODEF1(7), MODEF2(13), MODEF3(7)

DIMENSION XF(I), YF(I), A(I)

COMMON /MCOM/ CFTABL, IFTABL, NTABL, NEQF

COMMON /XSYCAL/ XX, YY, XG, YG, BREF, ZZZ, ITEST

COMMON /FUNNY/ N, IERF, B1, B2, B3, NFUNMX

DATA LBLANK / 4H /

DATA LEND / 3HENI) /
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(1, IEQ), J=1,2, (CFTABL(J, IEQ),  
LINE 60 1 J =1,6)  
LINE 61 500 CONTINUE  
LINE 62 C LOOP OVER ELEMENTS BY WINDING SECTION  
LINE 63 J = 5  
LINE 64 I = 1  
LINE 65 DO 630 J1=1,NWING  
LINE 66 NS = NSECT(J1)  
LINE 67 DO 395 J2=1,NS  
LINE 68 IN1 = NE(J2, J1)  
LINE 69 DO 390 K=1,IN1  
LINE 70 X = KCENT(I)*BR  
LINE 71 Y = ( XYZ(J) + XYZ(J+1) ) * HALFR  
LINE 72 CALL MFUN (JMODE)  
LINE 73 DO 380 IH=I,JMODE  
LINE 74 ETA(IH) = EM(IH)  
LINE 75 DETADX(IH) = DM(IH)  
LINE 76 380 CONTINUE  
LINE 77 I = I + 1  
LINE 78 J = J+1  
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, 10001 LAB, NW, NIL, NI2, NI3  
LINE 89 C  
LINE 90 IFLAB .EQ. LEND ) GO TO 330  
LINE 91 IFLAB .NE. LMW ) 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 IFLAB .EQ. 0 ) NSI = 2  
LINE 100 IFLAB .EQ. 0 ) NSI = 1  
LINE 101 IFLAB .EQ. 0 ) GO TO 310  
LINE 102 C IF 2 SECTIONS ARE INPUT, THEY MUST BE CONTINUOUS  
LINE 103 C  
LINE 104 IFLAB .EQ. 2 ) AND IABS(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 NSL = NI1  
LINE 110 WRITE(16, 2001)  
LINE 111 WRITE(16, 2302) NW, NI1  
LINE 112 IFLAB .EQ. 1 ) GO TO 20  
LINE 113 NSF = NI1( NSF, NI2 )  
LINE 114 NSL = MAX0( NSF, NI2 )  
LINE 115 WRITE(16, 2002) NW, NI2  
LINE 116 IFLAB .EQ. 2 ) GO TO 20  
LINE 117 NSF = MIN0( NSF, NI3 )  
LINE 118 NSL = MAX0( NSF, NI3 )  
LINE 119 6.69-
LINE 119 WRITE (16, 2002) NW, NI3
LINE 120 20 CONTINUE
LINE 121 C
LINE 122 C MORE CHECK TO SEE IF SECTION IS DEFINED
LINE 123 IF (NSL .GT. NSECT(NNW)) GO TO 310
LINE 124 C
LINE 125 C FIND FIRST AND LAST ELEMENT NUMBERS, IN1, IN2
LINE 126 C
LINE 127 IN1 = 1
LINE 128 IF (NW .EQ. 1) GO TO 30
LINE 129 C FIND NUMBER OF ELEMENTS IN LOWER NUMBER WINGS
LINE 130 NW1 = NW - 1
LINE 131 DO 25 J = 1, NW1
LINE 132 NS = NSECT(J)
LINE 133 DO 25 I = 1, NS
LINE 134 25 IN1 = IN1 + NE(I, J)
LINE 135 C
LINE 136 30 CONTINUE
LINE 137 IF (NSF = EQ. 1) GO TO 40
LINE 138 C FIND NUMBER OF ELEMENTS IN LOWER NUMBER SECTIONS OF WING NW
LINE 139 NS = NSF - 1
LINE 140 DO 35 I = 1, NS
LINE 141 35 IN1 = IN1 + NE(I, NW)
LINE 142 C
LINE 143 40 CONTINUE
LINE 144 C NOW THAT WE HAVE IN1, FIND LAST ELEMENT IN2
LINE 145 C
LINE 146 IN2 = IN1 + NE(NSF, NW) - 1
LINE 147 NCHECK(NE(NSF, NW)) = 1
LINE 148 IF (NSI .EQ. 1) GO TO 45
LINE 149 C
LINE 150 IN2 = IN2 + NE(NSF+1, NW)
LINE 151 NCHECK(NE(NSF+1, NW)) = 1
LINE 152 IF (NSI .EQ. 2) GO TO 45
LINE 153 IN2 = IN2 + NE(NSL, NW)
LINE 154 NCHECK(NE(NSL, NW)) = 1
LINE 155 45 CONTINUE
LINE 156 C
LINE 157 C IN1 AND IN2 HAVE BEEN FOUND
LINE 158 C
LINE 159 WRITE (16, MODEF4) JMODE
LINE 160 WRITE (16, 2003)
LINE 161 WRITE (16, 2008)
LINE 162 C
LINE 163 C SECTION TO READ MODE DATA
LINE 164 C
LINE 165 C
LINE 166 C READ MODE INPUT
LINE 167 C MODE CARD HAS BEEN READ
LINE 168 C
LINE 169 XFUNMX = NMX
LINE 170 CALL READXY(XF, YF)
LINE 171 IERR = IERF
LINE 172 IF (IERR .NE. 0) RETURN
LINE 173 C
LINE 174 DO 130 J = 1, JMODE
LINE 175 READ (15, 1006) MODE
LINE 176 WRITE (16, 1016) MODE
LINE 177 CALL READAB(A, A)
LINE 178 IF (IEST .EQ. 0) GO TO 110
LINE 179 DO 135 I = IN1, IN2
LINE 180 ETA(I, J) = ZIZ

-6.70-
LINE 181  DETADX(I,J1) = 0.0
LINE 182  105 CONTINUE
LINE 183  GO TO 130
LINE 184  110 CONTINUE
LINE 185  C
LINE 186  CONTINUE U29U18.10
LINE 187  DO 120 I = IN1, IN2
LINE 188  C
LINE 189  C
LINE 190  XX = 9R*XCEN(I)
LINE 191  YY = HALFR*(XYZ(INDXR64) & XYZ(INDXR5))
LINE 192  INDXR = INDXR & 6
LINE 193  C
LINE 194  CALL ZPDZ( ETA(I,J), DETADX(I,J1), XF, YF, A)
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( 16, 2004 ) IPR
LINE 204  IF(JTYPE .NE.0) GO TO 135
LINE 205  WRITE( 16, 2002 ) NW, NI1
LINE 206  IF( NSI .GE. 2 ) WRITE( 16, 2002 ) NW, NI2
LINE 207  IF( NSI .EQ. 3 ) WRITE( 16, 2002 ) NW, NI3
LINE 208  135 CONTINUE
LINE 209  WRITE( 16, 2005 )
LINE 210  WRITE( 16, 2008 )
LINE 211  J1 = 1
LINE 212  IHS = JMODE/3
LINE 213  IF( 3*IHS .NE. JMODE ) IHS = IHS & 1
LINE 214  C
LINE 215  DO 138 IH = 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) = NUMBR(INUMB)
LINE 220  MODEF2(5) = NUMBR(INUMB)
LINE 221  MODEF3(3) = NUMBR(INUMB)
LINE 222  IF( IH .GE. 2 ) WRITE( 16, 2010 )
LINE 223  WRITE( 16, MODEF1 ) ( I, I=J1, J2)
LINE 224  WRITE( 16, MODEF2 )
LINE 225  WRITE( 16, MODEF3 ) ( I, ETA(I,J), DETADX(I,J1), 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  IF(JTYPE .NE.0) GO TO 200
LINE 233  C
LINE 234  C
LINE 235  C
LINE 236  DO 150 J=1,NXING
LINE 237  NS = NSECT(J)
LINE 238  DO 150 I=1,NS
LINE 239  C
LINE 240  150 CONTINUE
LINE 241  C
LINE 242  C
LINE 243 C
LINE 244 C READ NEXT CARD, CHECK FOR RIGID MODE INPUT
LINE 245 200 READ( 15, 1000 ) LAB, JRAID
LINE 246 210 CONTINUE
LINE 247 IFI LAB .EQ. LENG IRETURN
LINE 248 IFI LAB .NE. LRIG I GO TO 340
LINE 249 CONTINUE
LINE 250 C SECTION FOR RIGID MODES
LINE 251 C CHECK THAT RIGID MODE NUMBER IS IN ORDER.
LINE 252 JMODE = JMODE + 1
LINE 253 IF( JRAID .NE. JMODE ) GO TO 350
LINE 254 C ZERO OUT ALL ETA AND DETAX FOR THAT MODE
LINE 255 DO 220 I =1, NET
LINE 256 ETA(I,JMODE) = 0.0
LINE 257 DETAX(I,JMODE) = 0.0
LINE 258 CONTINUE
LINE 259 WRITE( 16, 2004 )
LINE 260 WRITE( 16, 2005 )
LINE 261 WRITE( 16, 2006 )
LINE 262 WRITE( 16, 2007 )
LINE 263 MODEF1(2) = NUMBR(1)
LINE 264 WRITE( 16, MODEF1 ) JMODE
LINE 265 MODEF2(5) = NUMBR(1)
LINE 266 MODEF2(3) = NUMBR(1)
LINE 267 WRITE( 16, MODEF2 )
LINE 268 CONTINUE
LINE 269 READ( 15, 1001 ) LAB, IEL, ETA(IEL,JMODE), DETAX(IEL,JMODE)
LINE 270 IFI LAB .NE. LBLANK I GO TO 240
LINE 271 WRITE( 16, MODEF3 ) IEL, ETA(IEL,JMODE), DETAX(IEL,JMODE)
LINE 272 GO TO 230
LINE 273 CONTINUE
LINE 274 JRAID = IEL / 100
LINE 275 GO TO 210
LINE 276 C ERROR SECTION
LINE 277 C
LINE 278 C
LINE 279 C
LINE 280 300 CONTINUE
LINE 281 C MDWI INPUT CARD NOT FOUND
LINE 282 WRITE( 16, 3000 )
LINE 283 WRITE( 16, 3001 )
LINE 284 ERROR = .TRUE.
LINE 285 RETURN
LINE 286 310 CONTINUE
LINE 287 C SECTION IS NOT DEFINED
LINE 288 WRITE( 16, 3002 )
LINE 289 WRITE( 16, 3003 ) NW, NI1
LINE 290 WRITE( 16, 2002 ) NW, NI2
LINE 291 IF( VS1 .GE. 2 ) WRITE( 16, 2003 ) NW, NI2
LINE 292 IF( VS1 .EQ. 3 ) WRITE( 16, 2002 ) NW, NI3
LINE 293 ERROR = .TRUE.
LINE 294 RETURN
LINE 295 320 CONTINUE
LINE 296 C SECTIONS ARE NOT ADJACENT
LINE 297 WRITE( 16, 3004 )
LINE 298 WRITE( 16, 3005 ) NI1, NI2
LINE 299 ERROR = .TRUE.
LINE 300 RETURN
LINE 301 330 CONTINUE
LINE 302 C MODES NOT PRESENT FOR ALL SECTIONS
LINE 303 WRITE( 16, 3006 )
LINE 304 WRITE( 16, 3007 )
LINE 305 3004 FORMAT(50H MODAL DATA NOT PRESENT FOR THE FOLLOWING SECTIONS )
LINE 306 DO 332 J=1,NWING
LINE 307 45 = NSEC(J)
LINE 308 NO 332 =1,NS
LINE 309 IF( ICHECK(J,J) .EQ. 0 ) WRITE( 16, 2002 ) J, 1
LINE 310 332 CONTINUE
LINE 311 ERROR = .TRUE.
LINE 312 RETURN
LINE 313 C
LINE 314 340 CONTINUE
LINE 315 WRITE( 16, 3000 )
LINE 316 WRITE( 16, 3005 ) LAB
LINE 317 ERROR = .TRUE.
LINE 318 RETURN
LINE 319 C
LINE 320 350 CONTINUE
LINE 321 WRITE( 16, 3000 )
LINE 322 WRITE( 16, 3006 ) JMODE, JRIGID
LINE 323 ERROR = .TRUE.
LINE 324 RETURN
LINE 325 C
LINE 326 400 FORMAT( 1HO/ 50X, 114, 25H MODE FUNCTION EQUATIONS / )
LINE 327 1 4X, EQUATION 2X,4X, 4HWING, 2X,4X, 4HMODE,4X,20X,
LINE 328 2 12HCOEFFICIENTS /36X,8HCONSTANT , 2X, 8X, 1HX, 5X, 8X,1HY, 5X,0293270)
LINE 329 4 6X, 2HXY,6X ,
LINE 330 3X,8HX SQUARE , 3X, 3X,6HY SQUARE )
LINE 331 405 FORMAT( 6X, 312, 12X, 86E6.6 )
LINE 332 410 FORMAT( 1HO/ 8X,112,4X,4X,112, 4X,4X,12,2X,(6E14.4) )
LINE 333 1000 FORMAT( 44, 2X, 14, 14X, 2E12.0 )
LINE 334 1006 FORMAT( 20X, 20A4 )
LINE 335 1016 FORMAT( 1X, 20A4 )
LINE 336 2001 FORMAT( 1HO/ 24X,16HINPUT MODAL DATA // )
LINE 337 2002 FORMAT( 20X, 4HWING, 12, 10H SECTION, 12 )
LINE 338 2003 FORMAT( // 20X,10HCOEF CHECK )
LINE 339 2004 FORMAT( 11, 19X, 20HGENERATED MODAL DATA // )
LINE 340 2005 FORMAT( // 20X,16HAEKODYNAMIC GRID// )
LINE 341 2006 FORMAT( 1H1, // 20X, 21HPID ID BODY MODE INPUT )
LINE 342 2007 FORMAT( // 20X, 55HONLY ELEMENTS WITH A NON-ZERO MODAL VALUE MUST )
LINE 343 1BE INPUT )
LINE 344 2008 FORMAT( // 20X, 30H STRUCTURAL REFERENCE SYSTEM )
LINE 345 3000 FORMAT( // 1X, 120(1*) // 29H ERROR IN READING MODE INPUT. )
LINE 346 3001 FORMAT( // 1X, 120(1*) // 29H ERROR IN READING MODE INPUT. )
LINE 347 3002 FORMAT( 76H "MOWING" DATA CARD SPECIFIES INPUT FOR A SECTION THAT )
LINE 348 3003 FORMAT( 6H "MOWING" DATA CARD SPECIFIES TWO SECTIONS THAT ARE NOT )
LINE 349 1ADJACENT. / 8H SECTION, 13,12H AND SECTION, 13 )
LINE 350 3005 FORMAT( // 90H A LABEL DATA CARD FOLLOWING FLEXIBLE MODE DATA INPUT )
LINE 351 3006 FORMAT( // 39H RIGID MODE NUMBER IS NOT IN ORDER. / 24H MODE NUMBER )
LINE 352 1R EXPECTED IS, 14 / 21H MODE NUMBER INPUT IS, 14 )
LINE 353 END
DISPLAY...FILE

DECK 30

LINE 1  SUBROUTINE READAB(Z, A)
LINE 2  INTEGER HEADER(20)
LINE 3  DIMENSION Z(I),A(I)
LINE 4  COMMON /FDO20/ I5, I6
LINE 5  COMMON /DFUNNY/ NIERF, B1, B2, B3, NFUNMK
LINE 6  COMMON /XYSCAL/ XX, YY, XG, YC, BREF, ZZZ, ITEST
LINE 7  C
LINE 8  C
LINE 9  C INITIALIZATION ENTRY POINT (COEFFICIENTS).
LINE 10 READ( 15, 11) HEADER
LINE 11 WRITE( 16, 11) HEADER
LINE 12 READ( 15, 13) ( Z(I), I=1,N )
LINE 13 WRITE( 16, 12) ( Z(I), I=1,N )
LINE 14 ITEST = 0
LINE 15 ZZZ = Z(1)
LINE 16 DO 5 I=2, N
LINE 17 IF ( Z(I) .NE. ZZZ ) GO TO 7
LINE 18 5 CONTINUE
LINE 19 ITEST = 1
LINE 20 7 CONTINUE
LINE 21 READ( 15, 11) HEADER
LINE 22 WRITE( 16, 11) HEADER
LINE 23 READ( 15, 13) ( A(I), I=1,N )
LINE 24 WRITE( 16, 12) ( A(I), I=1,N )
LINE 25 READ( 15, 11) HEADER
LINE 26 WRITE( 16, 11) HEADER
LINE 27 READ( 15, 13) B1, B2, B3
LINE 28 WRITE( 16, 12) B1, B2, B3
LINE 29 RETURN
LINE 30 11 FORMAT(20A4)
LINE 31 12 FORMAT(3P15.7)
LINE 32 13 FORMAT(5E15.7)
LINE 33 END
SUBROUTINE READXY(X,Y)
LINE  2 INTEGER HEADER(20)
LINE  3 DIMENSION X(1), Y(1)
LINE  4 COMMON /ZFDZIO / 15, 16
LINE  5 COMMON /2FUNNY/ 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( 1P5E15.7)
LINE 26  13 FORMAT( 5E15.7)
LINE 27 END

-6.75-
SUBROUTINE RSTART ( MACH, NET, NFREQ, FREQ, REFLEN, IUTE, 18, 115)

REAL MACH, MACHR

DIMENSION FREQ(11, NFREQ), AREA(I), XYZ(1)

COMMON / NE XTC S / 1 FLUSH

ERROR = .FALSE.

IS = 10

CONTINUE

REWIND IS

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

IF( MACHR .NE. 'MACH 1 GO TO 90

IF( NETR .NE. NET ) GO TO 90

IF( NFREQR .NE. NFREQ ) GO TO 90

IF( REFLEN .NE. RRFL ) GO TO 90

IF( IUTE .NE. 0 .AND. IWTER .EQ. 0 1 GO TO 90

IF( FREQR(I) .NE. FREQ(11,I) GO TO 90

CONTINUE

IF( IWTER .EQ. 0 ) GO TO 30

IF( IS .EQ. 115 1 GO TO 30

IS = 115

GO TO 10

CONTINUE

ERROR = .TRUE.

WRITE( 16, 2000 ) IS

WRITE( 16, 2001 1

WRITE( 16, 2002 ) MACH, MACHR, NET, NETR, REFLEN, RRFL, IUTE,

WRITE( 16, 30 ) 1, FREQ, NFREQ, NFREQR, (I, FREQ(I), FREQRI, I=1,NFREQ)

IF( IUTE .EQ. 0 ) GO TO 99

IF( IS .EQ. 115 1 GO TO 99

IS = 115

GO TO 10

CONTINUE

IFLUSH = 1

RETURN

2000 FORMAT(1IH1/ 60(2H #) //38H ERROR IN READING RESTART TAPE ON UNIT

141 )

2001 FORMAT( //48H RESTART TAPE DID NOT CORRESPOND TO CURRENT RUN.

17 )

2002 FORMAT( //54H TOTAL NUMBER OF ELEMENTS, 117, 121

C

-6.76-
LINE 57  3 / 11X,  24H Wake Effect Code  117, 121  03200570
LINE 58  4 / 11X,  24H Number of Frequencies  117, 121  03200580
LINE 59  5 / ( 11X, 10FREQUENCY(.12, 111, 11X, E17.5, E21.5) )  03200540
LINE 60  END  03200600
DISPLAY...FILE  DECK 33

LINE 1 SUBROUTINE RTOI ( TR, TI, YCR, ZCR, ZRO, ZLO )
LINE 2 C RTOI TRANSFORMS YCR AND ZCR FROM THE RECEIVING SYSTEM TO THE INFLUENCING SYSTEM. YCR, ZCR ARE FIRST PUT IN REFERENCE SYSTEM BY PREMULTIPLYING BY TR. THIS IS Y, Z WHICH IS PUT IN INFLUENCING SYSTEM YC, ZC BY PREMULT. BY THE TRANSPOSE OF TI.
LINE 3 C
LINE 4 C DIMENSION TR(11), TI(11)
LINE 5 C COMMON / RRLL / TVW(4), YRO, YLO, ZR, ZRZ, ZL, ZLZL
LINE 6 C Y = TR(1)*YCR & TR(3)*ZCR
LINE 7 C Z = TR(2)*YCR & TR(4)*ZCR
LINE 8 C YRO = TI(1)*Y & TI(2)*Z
LINE 9 C ZRO = TI(3)*Y & TI(4)*Z
LINE 10 C YLO = -TI(1)*Y & TI(2)*Z
LINE 11 C ZLO = TI(3)*Y - TI(4)*Z
LINE 12 C TVW(1) = TR(3)*TI(1) & TR(4)*TI(2)
LINE 13 C TVW(2) = TR(3)*TI(3) & TR(4)*TI(4)
LINE 14 C TVW(3) = -TR(3)*TI(1) & TR(4)*TI(2)
LINE 15 C TVW(4) = TR(3)*TI(3) - TR(4)*TI(4)
LINE 16 C
LINE 17 C
LINE 18 C
LINE 19 C
LINE 20 C RETURN
LINE 21 C END

END OF JOB.

33.2 SEC. USED .010 HRS. CHARGED 49.952 HRS. REMAINING

-G.78-
SUBROUTINE SOLVE(N,NDIM,A,B,IP)

PURPOSE

SOLUTION OF THE LINEAR SYSTEM OF EQUATIONS C*X = B.

USAGE

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

DESCRIPTION OF PARAMETERS

N - ORDER OF THE MATRIX A

NDIM - FIRST DIMENSION OF A DECLARED IN CALLING PROGRAM. 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, ON OUTPUT, THE SOLUTION VECTOR.

IP - VECTOR OF DIMENSION N CONTAINING ROW INTERCHANGE
INFORMATION (AS DETERMINED BY SUBROUTINE 'DECOM').

REMARKS

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

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED.

METHOD

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

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

IF (N .EQ. 1) GO TO 9

NMI = N - 1

DO 7 K = 1, NMI
  KP1 = K + 1
  M = IP(K)
  T = B(M)
  B(M) = B(K)
  B(K) = T
  DO 1 I = KP1,N
    B(I) = B(I) - A(I,K)*T
  DO 8 KB = 1, NMI
    KM1 = N - KB
    K = KM1 + 1
    B(K) = B(K) - A(K,K)
  T = -B(K)
  DO 8 I = 1, KM1
    B(I) = B(I) + A(I,K)*T
  DO 8 I = 1, NMI
    B(I) = B(I)/A(I,I)

RETURN

END
DECK 35

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

SUBROUTINE SONS DETERMINES BOOLEAN MATRIX ISONS.
ISONS(K1,K2) = 1 IF SECTION K2 HAS INFLUENCE ON REC. SECT. K1.
ISONS(K2) = 0 IF K2 DOES NOT INFLUENCE K1.
K1 IS RECEIVING SECTION NUMBER, K2 IS INFLUENCING SECTION NUMBER.

DIMENSION NSECT(1), CORNX(8,4), CORNY(994,12), CORNZ(8,994), ISUNS(12,12)

LJOP ON RECEIVING SECTIONS (DO 30 AND DO 25)
K1 = 0
DO 30 J = 1, NSECT
NS1 = NSECT(J)
DO 25 I = 1, NS1
K1 = K1 (J)
JCN1 = 2*I - 2
LOOP ON SECTIONS WHICH MAY HAVE INFLUENCE (DO 20 AND DO 15)
K2 = 0
DO 20 JJ = 1, NWIN
NS2 = NSECT(JJ)
DO 15 II = 1, NS2
K2 = K2 + 1
ISONS(K1,K2) = 1
A SECTION ALWAYS INFLUENCES ITSELF AND ADJACENT SECTIONS.
IF (J .EQ. JJ) .AND. IABS(I-II) .LE. 1 GO TO 15
JCN2 = 2*II - 1
LOOP ON BOTTOM CORNER POINTS OF RECEIVING SECTION (PTS 2, 4)
DO 10 12 = 2, 4, 2
IND1 = JCN1 & 12
X1 = CORNX(IND1, JJ)
Y1 = CORNY(IND1, JJ)
Z1 = CORNZ(IND1, JJ)
IF (X1 .LT. X3) .AND. X1 .LE. X3 GO TO 10
Y1 = CORNY(JCN2, JJ)
Z1 = CORNZ(JCN2, JJ)
IF (Y1 .LT. Y3) .AND. Y1 .LE. Y3 GO TO 8
YCY1 = YC - Y1
Y3YC = Y3 - YC
IF (YCY1 .EQ. Y3YC .LE. 0.0) GO TO 8
IF (YC1*Y3YC .LE. 0.0) GO TO 8
YC IS BETWEEN Y1 AND Y3, CHECK INTERSECTION
A = YCY1/(Y1-Y3)
X = X1 - A*(X3-X1)
Z = Z1 - A*(Z3-Z1)
R = XC - X & 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
LINE 79 10 CONTINUE
LINE 80 C WE DID NOT BRANCH OUT OF DO 10 LOOP
LINE 81 C SECTION K2 DOES NOT INFLUENCE SECTION K1
LINE 82 ISONSKI2 = 0
LINE 83 C
LINE 84 C END LOOP ON INFLUENCING SECTION (15) AND WING (20)
LINE 85 15 CONTINUE
LINE 86 20 CONTINUE
LINE 87 C
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

-6.81-
DISPLAY FILE DECK 36

LINE 1 C SUBROUTINE SONSPT ( ISONS, NWING, NSECT, MACH, 16 )
LINE 2 C SONSPT PRINTS SECTION INFLUENCE TABLE
LINE 3 REAL MACH
LINE 4 DIMENSION IFT1(9), IFT2(9), IF(15)
LINE 5 DIMENSION (ISON(12,12), NSECT(1), IW(15), IS(15))
LINE 6 C
LINE 7 DATA IFT1 / 4H(11H, 4H REC, 4HEIVI, 4HNG,, 1H, 4H(6H, )
LINE 8 1 4H WIN, 4HG,12, 2H( ) /
LINE 9 DATA IF / 1H1,1H1,1H1,1H1,4H5,1H6,1H7,1H8,2H10,2H11,2H12,2H13 /
LINE 10 1 2H14, 2H15 /
LINE 11 DATA IFT2 / 4H(11H, 4H SEC, 4HTION, 4H5,, 1H, 4H(6H, )
LINE 12 1 4H SEC, 4H7,12, 2H( ) /
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), I=1,K)
LINE 28 WRITE( 16, IFT2 ) (IS(I), I=1,K)
LINE 29 DO 20 I=1,K
LINE 30 WRITE( 16, 1003 ) IW(I), IS(I), ( ISON(1,J), J=1,K )
LINE 31 20 CONTINUE
LINE 32 1001 FORMAT( 1H1 / 40X, 45HTABLE OF INFLUENCING SECTIONS FOR MACH NUMBER )
LINE 33                                                                                       03600320
LINE 34 1002 FORMAT( //2X, 50HERG IN ALL ELEMENTS OF INFLUENCING SECTION HAVE 1H, 5H0 X )
LINE 35                                                                                       03600340
LINE 36 INF 2 7X, 3H1 NON-ZERO INFLUENCE IS ASSUMED //
LINE 37 3 13X, 2H0 INFLUENCING SECTIONS )
LINE 38 1003 FORMAT( / 5WING, 12 / 5H SEC, 12, 4X, 15I8 )
LINE 39 RETURN
LINE 40 END

-6.82-
<table>
<thead>
<tr>
<th>LINE</th>
<th>Statement</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUBROUTINE SYMBOL(X, Y, SIZE, CHAR, ANG, NCHAR)</td>
<td>03700010</td>
</tr>
<tr>
<td>2</td>
<td>DIMENSION CHAR(3)</td>
<td>03700020</td>
</tr>
<tr>
<td>3</td>
<td>CALL NOTATE(X, Y, SIZE, CHAR, ANG, NCHAR)</td>
<td>03700030</td>
</tr>
<tr>
<td>4</td>
<td>RETURN</td>
<td>03700040</td>
</tr>
<tr>
<td>5</td>
<td>END</td>
<td>03700050</td>
</tr>
</tbody>
</table>
DISPLAY...FILE

<table>
<thead>
<tr>
<th>LINE</th>
<th>Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUBROUTINE TIMOUT(MSEC,A,B,C,D,E,F,G,H)</td>
</tr>
<tr>
<td>2</td>
<td>COMMON /TAPE/ 19,110,111,112,113,114,J9</td>
</tr>
<tr>
<td>3</td>
<td>DATA TI / 0.0 /</td>
</tr>
<tr>
<td>4</td>
<td>T = MSEC</td>
</tr>
<tr>
<td>5</td>
<td>T = T/1000.</td>
</tr>
<tr>
<td>6</td>
<td>TI = T - TI</td>
</tr>
<tr>
<td>7</td>
<td>WRITE(J9,3001)A,B,C,D,E,F,G,H,TI,T</td>
</tr>
<tr>
<td>8</td>
<td>TI = T</td>
</tr>
<tr>
<td>9</td>
<td>RETURN</td>
</tr>
<tr>
<td>10</td>
<td>3001 FORMAT(6X,8.4,5X,F8.3,13X,F8.3)</td>
</tr>
<tr>
<td>11</td>
<td>END</td>
</tr>
</tbody>
</table>

-6.84-
LINE 1  SUBROUTINE TMAX(NET, A, GBAR) 03900010
LINE 2  DIMENSION A(1) 03900020
LINE 3  DATA EPS / .0000001/ 03900030
LINE 4  GBAR = 0.0 03900040
LINE 5  DO 480  IEL = NET 03900050
LINE 6  G = ABS(A(IEL)) 03900060
LINE 7  IF(G .GT. GBAR) GBAR = G 03900070
LINE 8  480 CONTINUE 03900080
LINE 9  IF(GBAR .EQ. 0.0) GO TO 600 03900090
LINE 10  DO 580  IEL = NET 03900100
LINE 11  G = EPS * GBAR 03900110
LINE 12  G = ABS( G) 03900120
LINE 13  AIEL = ABS(A(IEL)) 03900130
LINE 14  IF (G .GT. AIEL) A(IEL) = 0.0 03900140
LINE 15  580 CONTINUE 03900150
LINE 16  600 RETURN 03900160
LINE 17  END 03900170

-6.85-
SUBROUTINE TRANS(X, Y, Z, T, XP, YP, ZC)

DIMENSION X(11), Y(1), Z(1), T(4), XP(1), YP(1)

T1X = X(4) - X(1)
T1Y = Y(4) - Y(1)
T1Z = Z(4) - Z(1)
T2X = X(3) - X(2)
T2Y = Y(3) - Y(2)
T2Z = Z(3) - Z(2)

XN = T1Y*T2Z - T1Z*T2Y
YN = T1Z*T2X - T1X*T2Z
ZN = T1X*T2Y - T1Y*T2X

XYZN = SQRT(XYZN)

XN = XN/XYZN
YN = YN/XYZN
ZN = ZN/XYZN

IT = 0

IF (ABS(ZC) .LT. 1.E-10) ZX = 1.0

DO 10 IT = 1, 4

XP(I) = X(I)
YP(I) = YN*Y(I) - YN*Z(I)

C FOLLOWING CALCULATION FOR DEBUG

Z1 = YN*Y(I) - ZN*Z(I)
IF (ABS(ZC-Z1) /ZX .GT. 0.001) IT = 1

10 CONTINUE

WRITE( 6,1001 ) ( X(I), XP(I), Y(I), YP(I), Z(I), Z1 , I=1,4)

WRITE( 6,1002 ) T

IF ( IT .EQ. 1 ) GO TO 20

RETURN

C20 CONTINUE

WRITE( 6,1003 )

C1003 FORMAT(/1X,120(1H+)//79H ERROR IN TRANS - ALL Z'S ARE NOT IDENTICAL// 1X,120(1H+)/// )

C FOR A SECTION. PROCEED WITH CAUTION // 1X, 120(1H+) ///// )

RETURN

C1001 FORMAT( // 3(2E14.4,2X) )

C1002 FORMAT( // (2E14.4 ) )

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

LINE 2: 1,FQ,K)
LINE 3: COMPLEX WTCE
LINE 4: COMPLEX AROW(NSPT), WROW(LRECLW)
LINE 5: COMPLEX AROWJ
LINE 6: LOGICAL TREDGE, WAKE1, WAKE, WAKENZ
LINE 7: DIMENSION IZ(NSPT)
LINE 8: DIMENSION NNCH(NSPT)
LINE 9: COMMON /FQ1/ NFRQ
LINE 10: COMMON /K1112/ KST11, KST12
LINE 11: COMMON/WAKEUP/ WTES(12), TREDGE, WAKE1, WAKE, WAKENZ, WAKE
LINE 12: IF(K1.I.EQ.K) GO TO 60
LINE 13: READ(I8) K1,WROW
LINE 14: IF(K1.IE.K) RETURN
LINE 15: 60 CONTINUE
LINE 16: NNZ = 0
LINE 17: L = IF
LINE 18: DO 100 J = 1, NSPT
LINE 19: AROWJ = WROW(J)
LINE 20: AR = AROWJ
LINE 21: AI = AIMAG(AROWJ)
LINE 22: AC = SORT(AR **2 + AI **2)
LINE 23: IF (AC .EQ. 0.0) GO TO 100
LINE 24: NNZ = NNZ + 1
LINE 25: IZ(NNZ) = NNCH(J)
LINE 26: AROW(NNZ) = AROWJ
LINE 27: 100 L = LNFRQ
LINE 28: IF (NNZ .NE. 0) WAKENZ = .TRUE.
LINE 29: IF(NNZ.EQ.0) RETURN
LINE 30: CALL WRITZWI(112, K1, NNZ, IZ, AROW)
LINE 31: KST12 = K1
LINE 32: RETURN
LINE 33: END
SUBROUTINE WINGCK(CORNX,CORNY,CORNZ, NWING, ICN, ERROR, I6 )

LOGICAL ERROR

DIMENSION CORNX(1), CORNY(1), CORNZ(1)

C CHECK ON VALIDITY OF CORNER POINTS OF WING

ICNM1 = ICN-1

DO 10 I=1,ICNM1,2

J = I & 1

IF(CORNX(J) .LT. CORNX(I) ) GO TO 5

IF(CORNY(J) .NE. CORNY(I) ) GO TO 5

IF(CORNZ(J) .NE. CORNZ(I) ) GO TO 5

GO TO 10

WRITE( 1, 1001 ) I, J, NWING

ERROR = .TRUE.

CONTINUE

RETURN

1001 FORMAT( //120(1H4) /32H ERROR IN DEFINING CORNER POINTS, IZ, /12,8H OF WING, IZ /1 )
SUBROUTINE WINGIN(I5, 16, NWING, NSECT, AR, NSP, CORNX, CORNY, CORNZ, CURNX, CURNY, CURNZ, SYM, ICL, CL, ERROR, BEETA, IPW, NREF, FREQ, REFLEN, LAB, ID, FD)

THIS ROUTINE READS AND WRITES INPUT FOR WING DATA.

INPUT

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

BEETA SQRT(MACH*MACH-1) USED TO TRANSFORM CORNX AND CL

ERROR LOGICAL THAT COMES IN FALSE, MAY BE SET TRUE

OUTPUT

NWING NUMBER OF WING READ FOR STRUCTURE
NSECT ARRAY- NO. SECTIONS IN EACH WING
AR ARRAY- ASPECT RATIO OF EACH WING
NSP(I,J) NO. SPANS FOR SECTION I OF WING J
CORNX, CORNY, CORNZ X, Y, AND Z COORD. OF CONTROL PTS. OF WING
SYM SYMMETRY CODE FOR WING J (A REAL MATRIX)

SYM = FOR SYMMETRY, -1 FOR ANTI-SYM, 0 FOR NO SYM.

ICL(I,J) CODE FOR SECTION I OF WING J
ICL(I,J) = -1 SPECIFIC GRID INFORMATION IS INPUT
ICL(I,J) = 0 AUTOMATIC MESH - NO CONTROL LINES
ICL(I,J) = 1, OR 2 AUTO MESH WITH 1 OR 2 CONTROL L.

CL X COORD OF CONTROL LINES FOR EACH SECTION

LAB, ID, FD INFORMATION ON LAST CARD READ

LOGICAL ERROR, CLINPT
LOGICAL FREQR

DIMENSION CORNX(8,4), CORNY(8,4), CORNZ(8,4), CL(4,3,4)
DIMENSION FREQ(1), IPW(1)
DIMENSION NSECT(1), AR(1), NSP(3,4), ICL(3,4), ID(9), FD(4)
DIMENSION SK1(4), SK2(4), SK3(4), SKODE(4)
DIMENSION SYM(4)

DATA L1, L2, L3, L4, LB / 4HGRID, 4HWING, 4HCORN, 4HLINE, 4H
data l5, l6, lend / 4HFREQ, 4HMODE, 4HEND /
data sk1 / 4HSYM, 4HMTR, 4HIC1, 4H /
data sk2 / 4HANT, 4H-SY, 4HMET, 4HIC1 /
data sk3 / 4HNOD, 4H SYM, 4HMTR, 4HIC1 /

FREQ = FALSE.
NWING = 0
BEERFL = BEETA*REFLEN

********************************************************************

LINE 1 SUBROUTINE WINGIN(I5, 16, NWING, NSECT, AR, NSP, CORNX, CORNY, 04300010
LINE 2 1 CORNZ, SYM, ICL, CL, ERROR, BEETA, IPW, NREF, FREQ, 04300020
LINE 3 2 REFLEN, LAB, ID, FD ) 04300030
LINE 4 C 04300040
LINE 5 C THIS ROUTINE READS AND WRITES INPUT FOR WING DATA 04300050
LINE 6 C 04300060
LINE 7 C INPUT 04300070
LINE 8 C INTPUT UNIT (USUALLY CARD READER 5) 04300090
LINE 9 C OUTPUT UNIT (PRINTER 6) 04300100
LINE 10 C BEETA SQRT(MACH*MACH-1) USED TO TRANSFORM CORNX AND CL 04300110
LINE 11 C ERROR LOGICAL THAT COMES IN FALSE, MAY BE SET TRUE 04300120
LINE 12 C 04300130
LINE 13 C OUTPUT 04300140
LINE 14 C 04300150
LINE 15 C NWING NUMBER OF WING READ FOR STRUCTURE 04300160
LINE 16 C NSECT ARRAY- NO. SECTIONS IN EACH WING 04300170
LINE 17 C AR ARRAY- ASPECT RATIO OF EACH WING 04300180
LINE 18 C NSP(I,J) NO. SPANS FOR SECTION I OF WING J 04300190
LINE 20 C CORNX,CORNY,CORNZ X,Y,AND Z COORD. OF CONTROL PTS. OF WING 04300200
LINE 21 C SYM SYMMETRY CODE FOR WING J (A REAL MATRIX) 04300210
LINE 22 C SYM= FOR SYMMETRY, -1 FOR ANTI-SYM, 0 FOR NO SYM. 04300220
LINE 23 C ICL(I,J) CODE FOR SECTION I OF WING J 04300230
LINE 24 C ICL(I,J) = -1 SPECIFIC GRID INFORMATION IS INPUT 04300240
LINE 25 C ICL(I,J) = 0 AUTOMATIC MESH - NO CONTROL LINES 04300250
LINE 26 C ICL(I,J) = 1, OR 2 AUTO MESH WITH 1 OR 2 CONTROL L. 04300260
LINE 27 C CL X COORD OF CONTROL LINES FOR EACH SECTION 04300270
LINE 28 C CL IS READ IN REF. SYSTEM, THEN DIVIDED BY BEETA 04300280
LINE 29 C LAB, ID, FD INFORMATION ON LAST CARD READ 04300290
LINE 30 C 04300300
LINE 31 C LOGICAL ERROR, CLINPT 04300310
LINE 32 C LOGICAL FREQR 04300320
LINE 33 C DIMENSION CORNX(8,4), CORNY(8,4), CORNZ(8,4), CL(4,3,4) 04300330
LINE 34 C DIMENSION FREQ(1), IPW(1) 04300340
LINE 35 C DIMENSION NSECT(1), AR(1), NSP(3,4), ICL(3,4), ID(9), FD(4) 04300350
LINE 36 C DIMENSION SK1(4), SK2(4), SK3(4), SKODE(4) 04300360
LINE 37 C DIMENSION SYM(4) 04300370
LINE 38 C 04300380
LINE 39 C DATA L1, L2, L3, L4, LB / 4HGRID, 4HWING, 4HCORN, 4HLINE, 4H 04300390
LINE 40 C DATA L5, L6, LEND / 4HFREQ, 4HMODE, 4HEND / 04300400
LINE 41 C DATA SK1 / 4HSYM, 4HMTR, 4HIC1, 4H / 04300410
LINE 42 C DATA SK2 / 4HANT, 4H-SY, 4HMET, 4HIC1 / 04300420
LINE 43 C DATA SK3 / 4HNOD, 4HSY, 4HMTR, 4HIC1 / 04300430
LINE 44 C 04300440
LINE 45 C FREOR = FALSE.
LINE 46 C NWING = 0 04300450
LINE 47 C BEERFL = BEETA*REFLEN 04300460
LINE 48 C 04300470
LINE 49 C 04300490
LINE 50 C 04300500
LINE 51 10 READ(15,1000 ) LAB, ID, FD 04300510
LINE 52 C 04300520
LINE 53 IF( LAB .EQ. L2 ) GO TO 20 04300530
LINE 54 04300540
LINE 55 IF( LAB .EQ. L1 ) GO TO 80 04300550
LINE 56 IF( LAB .EQ. L6 ) GO TO 65 04300560

-6.89-
LINE 57 IF ( LAB .EQ. LS ) GO TO 50
LINE 58 IF ( LAB .EQ. LEND ) GO TO 60
LINE 59 IF ( LAB .EQ. LB ) GO TO 70
LINE 60 C INVALID LABEL CARD READ
LINE 61 C
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(1) = 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 ; WRITE ( I6, 2000 )
LINE 85 ; WRITE ( I6, 2018 )
LINE 86 ; WRITE ( I6, 2004 )
LINE 87 ; ERROR = . TRUE.
LINE 88 21 CONTINUE
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, 2001 )
LINE 93 ; WRITE ( I6, 2002 ) ID(1), NWING
LINE 94 ERROR = . TRUE.
LINE 95 22 CONTINUE
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, 2003 ) ID(1)
LINE 101 ERROR = . TRUE.
LINE 102 JW = 1
LINE 103 23 CONTINUE
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, 2001 )
LINE 108 ; WRITE ( I6, 2005 ) JW, NS
LINE 109 ERROR = . TRUE.
LINE 110 NS = 3
LINE 111 24 CONTINUE
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, 2001 )
LINE 117 ; WRITE ( I6, 2006 ) NC, NS, JW
LINE 118 ERROR = . TRUE.

-6.90-
04301190
LINE 119  IF( VST .LE. 3 ) NS = NST
04301200
LINE 120  25 CONTINUE
04301210
LINE 121  C
04301220
LINE 122  NSECT(JW) = NS
04301230
LINE 123  SYM(J) = FLOAT(ISYM)
04301240
LINE 124  DO 26 I=1,NS
04301250
LINE 125  ICL(I,JW)=0
04301260
LINE 126  26 NSP(I,JW) = ID(I65)
04301270
LINE 127  AR(JW) = FD(1)
04301280
LINE 128  C
04301290
LINE 129  IF( SNAP(ISYM) .LE. 1 ) GO TO 27
04301300
LINE 130  WRITE(1,2000)(1,NSP(I,JW),I=1,NS)
04301310
LINE 131  WRITE(1,2001)(1,NSP(I,JW),I=1,NS)
04301320
LINE 132  C
04301330
LINE 133  IF( SNAP(ISYM) .GT. 1 ) GO TO 28
04301340
LINE 134  WRITE(1,2000)(1,ISYM)
04301350
LINE 135  27 CONTINUE
04301360
LINE 136  C
04301370
LINE 137  WRITE(1,2000)(1,2000)(1,NSP(I,JW))
04301380
LINE 138  28 CONTINUE
04301390
LINE 139  GO TO 34
04301400
LINE 140  29 SKODE(I) = SK2(I)
04301410
LINE 141  GO TO 34
04301420
LINE 142  30 CONTINUE
04301430
LINE 143  GO TO 34
04301440
LINE 144  31 SKODE(I) = SK3(I)
04301450
LINE 145  GO TO 34
04301460
LINE 146  32 CONTINUE
04301470
LINE 147  GO TO 34
04301480
LINE 148  33 SKODE(I) = SK1(I)
04301490
LINE 149  34 CONTINUE
04301500
LINE 150  C
04301510
LINE 151  WRITE(1,3000)(1,ISYM)
04301520
LINE 152  WRITE(1,3001)(1,NSP(I,JW),I=1,NS)
04301530
LINE 153  WRITE(1,3002)(1,AR(JW))
04301540
LINE 154  C
04301550
LINE 155  WRITE(1,3003)(1,2000)(1,NSP(I,JW),I=1,NS)
04301560
LINE 156  C
04301570
LINE 157  C
04301580
LINE 158  READ CORNER POINTS
04301590
LINE 159  DD 36 I=1,NC
04301600
LINE 160  READ(15,1000) LAB, ID, FD
04301610
LINE 161  IF( I .GT. 1 ) GO TO 35
04301620
LINE 162  WRITE(1,3000)(1,AR(JW))
04301630
LINE 163  C
04301640
LINE 164  C
04301650
LINE 165  WRITE(1,3002)(1,AR(JW))
04301660
LINE 166  ERROR = .TRUE.
04301670
LINE 167  35 CONTINUE
04301680
LINE 168  CORNX(I,JW) = FD(I1)
04301690
LINE 169  CORNY(I,JW) = FD(I2)
04301700
LINE 170  CORNZ(I,JW) = FD(I3)
04301710
LINE 171  36 CONTINUE
04301720
LINE 172  C
04301730
LINE 173  WRITE(16,3004)(1,NC)
04301740
LINE 174  C
04301750
LINE 175  IF( 2*(NC/2) .EQ. NC ) GO TO 37
04301760
LINE 176  CORNX(1,JW) = CORNX(1,JW)
04301770
LINE 177  CORNY(1,JW) = CORNY(1,JW)
04301780
LINE 178  CORNZ(1,JW) = CORNZ(1,JW)
04301790
LINE 179  NC = NC+1
04301800
LINE 180  37 CONTINUE

-6.91-
CALL ROUTINE TO CHECK FOR PHYSICAL ERRORS AT CORNER POINT
CALL WINCK(CORNX(I,JW),CORNY(I,JW),CORNZ(I,JW),JW,NL,ERROR,16)
DO 38 I=1,NC
CORNX(I,JW) = CORNX(I,JW)/BEERFL
CORNY(I,JW) = CORNY(I,JW)/REFLEN
CORNZ(I,JW) = CORNZ(I,JW)/REFLEN
CONTINUE
GO TO 10
40 CONTINUE
SECTION TO PROCESS CONTROL LINE INFORMATION
IS = ID(1)
K = ICL(IS,JW) + 1
IF( IS .LE. NSECT(JW) ) GO TO 41
WRITE(16,2000) JW
WRITE(16,2008) IS, JW, NSECT(JW)
ERROR = .TRUE.
GO TO 10
41 CONTINUE
IF( K .LE. 2 ) GO TO 42
WRITE(16,2000) JW
WRITE(16,2008) IS
WRITE(16,2010) K
ERROR = .TRUE.
42 CONTINUE
IF( K .EQ. ID(2) ) GO TO 43
CONTROL LINE DATA OUT OF SEQUENCE
WRITE(16,2000) JW
WRITE(16,2008) IS
WRITE(16,2010) K
ERROR = .TRUE.
43 CONTINUE
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 .LE. CORNX(IGN,JW) ) GO TO 44
IF( X2 .LT. CORNX(IGN2,JW) ) GO TO 44
IF( X2 .LT. CORNX(IGN3,JW) ) GO TO 44
GO TO 45
44 CONTINUE
CONTROL LINE DEFINED OUTSIDE OF SECTION
WRITE(16,2000) JW
WRITE(16,2008) IS
WRITE(16,2010) K
ERROR = .TRUE.
45 CONTINUE
IF( K .EQ. 1 ) GO TO 46
IF( X1 .GE. CL(1,IS,JW) .AND. X2 .GE. CL(2,IS,JW) ) GO TO 46

SECOND CONTROL LINE DEFINED ABOVE FIRST

WRITE (16, 2006 )

WRITE (16, 2008 ) JW

WRITE (16, 2013 ) IS

ERROR = .TRUE.

CONTINUE

JCN = 2*K - 1

CL(JCN,IS,JW) = X1

CL(JCN+1,IS,JW) = X2

GO TO 10

SUCCESSFUL

CONTINUE

FREQUENCY CARD READ

DO 55 I = 1, 4

FREQ(I) = FD(I)

NFREQ = ID(I)

IF( NFREQ .GT. 4 ) READ( 15, 1001 ) ( FREQ(I), I=5, NFREQ )

WRITE (16, 3036 ) NFREQ, ( FREQ(I), I=1,NFREQ )

GO TO 10

SUCCESSFUL

CONTINUE

END CARD READ

MODE DATA IS MISSING

WRITE (16, 2000 )

WRITE (16, 2015 )

LAB, ID, FD

ERROR = .TRUE.

RETURN

CONTINUE

GRID DATA READ

I = ID(2)

J = ID(1)

IF( I .GT. NSECT(J) ) GO TO 81

IF( J .GT. NHING ) GO TO 81

ICL(I,J) = -1

RETURN

CONTINUE

GRID DATA SPECIFIED FOR SECTION THAT DOES NOT EXIST

WRITE (16, 2000 )

WRITE (16, 2020 )

WRITE(16, 2021 ) J, I

ERROR = .TRUE.

RETURN

CONTINUE

-6.93-
1001 FORMAT (A4, 2X, 912, 4E12.0 )
1002 FORMAT (A4, 2X, 4E12.0 )
2000 FORMAT(/// 1X, 129(1H:), 1001 FORMAT(/// 2X, 2H, 912, 1003 FORMAT(/// 2H, 4E12.0 )
1003 FORMAT(/// 912, 4E12.0 )
1004 FORMAT(/// 912, 4E12.0 )
DISPLAY...FILE  DECK 44

LINE  1  SUBROUTINE WINTGR X, Y, Z, X, 04403000
LINE  2  ARRAY, N 04403001
LINE  3  REAL, DIMENSION (12) 04403002
LINE  4  REAL, DIMENSION (12) 04403003
LINE  5  REAL, DIMENSION (12) 04403004
LINE  6  REAL, DIMENSION (12) 04403005
LINE  7  REAL, DIMENSION (12) 04403006
LINE  8  REAL, DIMENSION (12) 04403007
LINE  9  REAL, DIMENSION (12) 04403008
LINE 10  REAL, DIMENSION (12) 04403009
LINE 11  REAL, DIMENSION (12) 04403010
LINE 12  REAL, DIMENSION (12) 04403011
LINE 13  REAL, DIMENSION (12) 04403012
LINE 14  REAL, DIMENSION (12) 04403013
LINE 15  REAL, DIMENSION (12) 04403014
LINE 16  REAL, DIMENSION (12) 04403015
LINE 17  REAL, DIMENSION (12) 04403016
LINE 18  REAL, DIMENSION (12) 04403017
LINE 19  REAL, DIMENSION (12) 04403018
LINE 20  REAL, DIMENSION (12) 04403019
LINE 21  REAL, DIMENSION (12) 04403020
LINE 22  REAL, DIMENSION (12) 04403021
LINE 23  REAL, DIMENSION (12) 04403022
LINE 24  REAL, DIMENSION (12) 04403023
LINE 25  REAL, DIMENSION (12) 04403024
LINE 26  REAL, DIMENSION (12) 04403025
LINE 27  REAL, DIMENSION (12) 04403026
LINE 28  REAL, DIMENSION (12) 04403027
LINE 29  REAL, DIMENSION (12) 04403028
LINE 30  REAL, DIMENSION (12) 04403029
LINE 31  REAL, DIMENSION (12) 04403030
LINE 32  REAL, DIMENSION (12) 04403031
LINE 33  REAL, DIMENSION (12) 04403032
LINE 34  REAL, DIMENSION (12) 04403033
LINE 35  REAL, DIMENSION (12) 04403034
LINE 36  REAL, DIMENSION (12) 04403035
LINE 37  REAL, DIMENSION (12) 04403036
LINE 38  REAL, DIMENSION (12) 04403037
LINE 39  REAL, DIMENSION (12) 04403038
LINE 40  REAL, DIMENSION (12) 04403039
LINE 41  REAL, DIMENSION (12) 04403040
LINE 42  REAL, DIMENSION (12) 04403041
LINE 43  REAL, DIMENSION (12) 04403042
LINE 44  REAL, DIMENSION (12) 04403043
LINE 45  REAL, DIMENSION (12) 04403044
LINE 46  REAL, DIMENSION (12) 04403045
LINE 47  REAL, DIMENSION (12) 04403046
LINE 48  REAL, DIMENSION (12) 04403047
LINE 49  REAL, DIMENSION (12) 04403048
LINE 50  REAL, DIMENSION (12) 04403049
LINE 51  REAL, DIMENSION (12) 04403050
LINE 52  REAL, DIMENSION (12) 04403051
LINE 53  REAL, DIMENSION (12) 04403052
LINE 54  REAL, DIMENSION (12) 04403053
LINE 55  REAL, DIMENSION (12) 04403054
LINE 56  REAL, DIMENSION (12) 04403055

ORIGINAL PAGE IS
OF POOR QUALITY
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 EONE
LINE 63 C IF( LNSID .EQ. 0 ) TO 40
LINE 64 C IF ELEMENT IS NOT IN MACH CONE, RETURN WITH ZERU IN WLEAD.
LINE 65 C PROCESS ANYWAY IF IT IS A SELF ELEMENT (DIAGONAL TERM)
LINE 66 C IF( LNSID .OR. IIN .EQ. 1 ) GO TO 20
LINE 67 IF LNSID .OR. IIN .EQ. 1) GO TO 20
LINE 68 C RETURN
LINE 69 C
LINE 70 C
LINE 71 20 CONTINUE
LINE 72 IF( LNSID .EQ. 1 .AND. IIN .EQ. 1 ) KCONW = 0
LINE 73 C
LINE 74 ASSIGN 310 TO I WAKE
LINE 75 ASSIGN 160 TO I WAKE2
LINE 76 C
LINE 77 NTTRM = 1
LINE 78 RRU = RR
LINE 79 RLU = RL
LINE 80 MIRROR = EMIRRO
LINE 81 TRAINF = .FALSE.
LINE 82 WAKE1 = .FALSE.
LINE 83 IF( NOT. INTRIGE ) GO TO 25
LINE 84 C TEST TRAILING EDGE LINE
LINE 85 C
LINE 86 YLW = Y(1)
LINE 87 YUP = Y(1)
LINE 88 XLW = X(1)
LINE 89 XUP = X(1)
LINE 90 C CALL EONE
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
LINE 98 IF( NOT. IERONE ) GO TO 40
LINE 99 TRAINF = .TRUE.
LINE 100 IF( LATE , I .EQ. 0 .AND. RTR .GT. 0. ) WAKE1 = .TRUE.
LINE 101 GO TO 40
LINE 102 C
LINE 103 25 CONTINUE
LINE 104 IF( NINSID .EQ. 0 ) GO TO 40
LINE 105 DD 30 I = 1, NNS
LINE 106 30 WREAD(I) = -WLEAD(I)
LINE 107 C
LINE 108 40 CONTINUE
LINE 109 DD 45 I = 1, NNS
LINE 110 45 WREAD(I) = 0.0
LINE 111 C
LINE 112 C
LINE 113 NINSID = NINSID + 1
LINE 114 C
LINE 115 C
LINE 116 C
LINE 117 Z12ZD = Z12CR
LINE 118 TV = TVM(1)
LINE 119  \( T_d = TVW(z) \)
LINE 120  C IF( \( FFTEST .EQ. 0.0 \) ) GO TO 145
LINE 121  C
LINE 122  C
LINE 123  C SECTION FOR NEAR FIELD FOR LEADING EDGE
LINE 124  C
LINE 125  RK = \( 4RTTEST \)
LINE 126  ASS13N 145 TO GOBACK
LINE 127  GO TO 600
LINE 128  145 CONTINUE
LINE 129  C
LINE 130  CALL WINT
LINE 131  C
LINE 132  ASS13N 150 TO GOBACK
LINE 133  GO TO 800
LINE 134  150 CONTINUE
LINE 135  DD 155 I=1,NFREQ
LINE 136  WLEA(I) = \( \text{WLEAD}(I) \) 
LINE 137  WROW(I) = \( \text{WROW}(I) \) 
LINE 138  155 CONTINUE
LINE 139  C
LINE 140  GO TO Make2, ( 160, 320 )
LINE 141  160 CONTINUE
LINE 142  C
LINE 143  C CHECK IF LEFT HAND ELEMENT (MIRROR IMAGE) EXISTS
LINE 144  C
LINE 145  IF( \( \text{NOT. MIRROR} \) ) GO TO 300
LINE 146  C SECTION FOR LEFT HAND SIDE OF LEADING EDGE
LINE 147  IF( \( FFTEST .EQ. 0.0 \) ) GO TO 185
LINE 148  C
LINE 149  C SECTION FOR NEAR FIELD, LEFT HAND SIDE LEADING EDGE
LINE 150  C
LINE 151  RK = \( 4LRTTEST \)
LINE 152  ASS13N 185 TO GOBACK
LINE 153  GO TO 600
LINE 154  185 CONTINUE
LINE 155  C
LINE 156  YO = YLU
LINE 157  ZD = ZL
LINE 158  ZDZD = ZLZL
LINE 159  TV = TVK(3)
LINE 160  TW = TVK(4)
LINE 161  C
LINE 162  CALL WINT
LINE 163  C
LINE 164  ASS13N 190 TO GOBACK
LINE 165  GO TO 800
LINE 166  190 CONTINUE
LINE 167  DD 195 I=1,NFRED
LINE 168  WLEA(I) = \( \text{WLEAD}(I) \) - \( \text{SYM} \times \text{SKA}(I) \)
LINE 169  WROW(I) = \( \text{WROW}(I) \) - \( \text{SYM} \times \text{SKA}(I) \)
LINE 170  195 CONTINUE
LINE 171  C
LINE 172  300 CONTINUE
LINE 173  C
LINE 174  IF( \( \text{NOT. TRAINF} \) ) RETURN
LINE 175  GO TO Make1, ( 310, 320 )
LINE 176  C
LINE 177  C
LINE 178  310 CONTINUE
LINE 179  C
LINE 180  C
LINE 181  \( \text{FIN} \\text{D} \text{\text{INFLUENCE} OF \text{TRAILING EDG}}E \text{ OF \text{TRAILING EDG}}E \text{ ELEMENT} \)

-6.97-
LINE 131 C (THE RIGHT HAND ELEMENT ONLY AT THIS TIME) 04401910
LINE 182 C 04401920
LINE 183 C 04401930
LINE 184 C 04401940
LINE 185 C 04401950
LINE 186 YLW = Y(2) 04401960
LINE 187 YUP = Y(1) 04401970
LINE 188 XLW = X(3) 04401980
LINE 189 XUP = X(4) 04401990
LINE 190 YO = YKO 04401990
LINE 191 ZD = ZR 04402000
LINE 192 ZDZD = ZRZR 04402010
LINE 193 TV = TVW(1) 04402020
LINE 194 TW = TVW(2) 04402030
LINE 195 SYM = -1.0 04402040
LINE 196 MIRRA = EMIRRO 04402050
LINE 197 C 04402060
LINE 198 DO 315 J=1,NFREQ 04402070
LINE 199 HTES(J) = (0.0, 0.0). 04402080
LINE 200 WSAVE(J) = WLEAD(J) 04402090
LINE 201 315 WLEAD(J) = 0.0 04402100
LINE 202 C 04402110
LINE 203 IF( FRTEST .EQ. 0.0 ) GO TO 145 04402120
LINE 204 RK = RR*FRTEST 04402130
LINE 205 ASSIGN 145 TO GJ64CK 04402140
LINE 206 GO TO 600 04402150
LINE 207 C 04402160
LINE 208 320 CONTINUE 04402170
LINE 209 C 04402180
LINE 210 IF( WDRT. = WAKE1 ) GO TO 370 04402190
LINE 211 C 04402200
LINE 212 C SECTION TO COMPUTE WAKE EFFECT 04402210
LINE 213 C THIS SECTION HANDLES BOTH RIGHT AND LEFT HAND ELEMENTS 04402220
LINE 214 C 04402230
LINE 215 Z1 = XO - XLW 04402240
LINE 216 Z2 = XU -XUP 04402250
LINE 217 E1 = YO - YLW 04402260
LINE 218 E2 = YO - YUP 04402270
LINE 219 Z3P = SQRT( E2*E2 & ZDZD ) 04402280
LINE 220 Z4P = SQRT( E1*E1 & ZDZD ) 04402290
LINE 221 ZTE = 0.5*(Z1GZ2) 04402300
LINE 222 ZLW = 0.5*( Z3PZ2P*1 04402310
LINE 223 C DIVIDE Z1S=ZU-ZL INTO A NUMBER OF DIVISIONS 04402320
LINE 224 ZDIS = ZTE/ ZLh 04402330
LINE 225 D5 = E2 - E1 04402340
LINE 226 NW = ZDIS/D5 04402350
LINE 227 IF( NW .LT. 0.0 ) NW = 1 04402360
LINE 228 IF( FRTEST .EQ. 0.0 ) NW = 1 04402370
LINE 229 DELTAZ = ZDIS/NW 04402380
LINE 230 ZSMZTE = -0.5*DELTALZ 04402390
LINE 231 C 04402400
LINE 232 C FIND WAKE EFFECT OF TRAILING EDGE LINE 04402410
LINE 233 C (NOTE WLEAD WAS COMPUTED IN OPPOSITE DIRECTION) 04402420
LINE 234 C 04402430
LINE 235 DDI 325 I=1,NFREQ 04402440
LINE 236 F82 = -( FREQ(I)/FETA )#ZSMZTE 04402450
LINE 237 CXRI(I) = COS( F82 ) 04402460
LINE 238 CXII(I) = SIN( F82 ) 04402470
LINE 239 HTES(I) = HTES(I) - CMPLX( WLEAD(I)*CXRI(I), WLEAD(I)*CXII(I) ) 04402480
LINE 240 325 CONTINUE 04402490
LINE 241 C IF NW=1 WE ARE DONE 04402500
LINE 242 IF( NW .LT. 1.0 ) GO TO 370 04402510

-6.98-
LINE 243 C  SECTION TO FIND WAKE EFFECT OF THE REST OF THE WAKE ELEMENTS
LINE 244 C  
LINE 245 C  
LINE 246 C  XLW = AVERAGE X
LINE 247 C  
LINE 248 C  
LINE 249 C  YLW = Y(I)
LINE 250 C  YUP = Y(I)
LINE 251 C  ELS = (Y-YLW)**2 + 2*Y
LINE 252 C  EUS = (Y-YUP)**2 + 2*Y
LINE 253 C  DO 360 LINE = 2, NW
LINE 254 C  
LINE 255 C  DO 227 I=1,NFREQ
LINE 256 C  
LINE 257 C  XXLEAD(I) = 0.0
LINE 258 C  
LINE 259 C  XLW = XLW & DELTA
LINE 260 C  XUP = XLW
LINE 261 C  ZETA = XO - XLW
LINE 262 C  ZSMZTE = ZSMZTE - DELTA
LINE 263 C  
LINE 264 C  IF (FRTEST .EQ. 0.0) GO TO 330
LINE 265 C  
LINE 266 C  
LINE 267 C  ASSIGN 330 TO GOBACK
LINE 268 C  GO TO 600
LINE 269 C  CONTINUE
LINE 270 C  CALL HINT
LINE 271 C  
LINE 272 C  
LINE 273 C  ASSIGN 340 TO GOBACK
LINE 274 C  GO TO 800
LINE 275 C  CONTINUE
LINE 276 C  DO 345 I=1,NFREQ
LINE 277 C  XXLEAD(I) = XXLEAD(I) - SYM*SKA(I)
LINE 278 C  
LINE 279 C  DO 350 J=1,NFREQ
LINE 280 C  XXLEAD(I)*ZSMZTE
LINE 281 C  CXPRI = CUS( FBZ )
LINE 282 C  
LINE 283 C  HZES(IJ) = HZES(IJ)
LINE 284 C  
LINE 285 C  CXRI(IJ) = CXRI
LINE 286 C  
LINE 287 C  CONTINUE
LINE 288 C  
LINE 289 C  
LINE 290 C  CONTINUE
LINE 291 C  
LINE 292 C  CONTINUE
LINE 293 C  
LINE 294 C  CHECK IF MIRROR IMAGE OF TRAILING EDGE LINE HAS INFLUENCE AND
LINE 295 C  HAS NOT YET BEEN CONSIDERED.
LINE 296 C  
LINE 297 C  IF (EMIRR .NOT. EMIRR) GO TO 380
LINE 298 C  
LINE 299 C  
LINE 300 C  
LINE 301 C  SET EMIRR TO FALSE, SO THAT WE DO NOT BRANCH HERE AGAIN
LINE 302 C  EMIRR = .FALSE.
LINE 303 C  
LINE 304 C  DO 375 I=1,NFREQ
LINE 305  375  WLEAD(I) = 0.0
LINE 306  C
LINE 307  XLW = X(3)
LINE 308  XUP = X(4)
LINE 309  YLW = Y(2)
LINE 310  YUP = Y(1)
LINE 311  YO = YLO
LINE 312  ZD = ZL
LINE 313  ZUZD = ZLZL
LINE 314  TV = TVH(3)
LINE 315  TW = TVH(4)
LINE 316  SYM = SYM
LINE 317  IF (FRTEST .EQ. 0.0) GO TO 185
LINE 318  RK = XL*FRTEST
LINE 319  ASSIGN 185 TO G08ACK
LINE 320  GO TO 600
LINE 321  C
LINE 322  380 CONTINUE
LINE 323  C
LINE 324  C WE ARE FINISHED CALCULATING WAKE EFFECT
LINE 325  C AND/OR INFLUENCE OF TRAILING EDGE OF A TRAILING EDGE ELEMENT
LINE 326  C PUT WLEAD BACK TO WSAVE AND RETURN
LINE 327  C
LINE 328  DO 390 I=1,NFRE
LINE 329  390 WLEAD(I) = WSAVE(I)
LINE 330  C
LINE 331  RETURN
LINE 332  C
LINE 333  380 CONTINUE
LINE 334  C
LINE 335  600 CONTINUE
LINE 336  C SECTION TO FIND NUMBER OF TERMS TAKEN IN SERIES
LINE 337  IFI (K .GT. 20.5) GO TO 615
LINE 338  DO 610 NTERM=6,NTMX
LINE 339  IFI (K .LE. RKTERM(NTERM)) GO TO 620
LINE 340  610 CONTINUE
LINE 341  615 CONTINUE
LINE 342  TEXCD = .TRUE.
LINE 343  NCVEL = NCVEL & 1
LINE 344  NTERM = NTERM
LINE 345  620 CONTINUE
LINE 346  NTERM = MINT(NTERM, NTERM XX)
LINE 347  IFI (K .GT. RKEX) RKEX = RK
LINE 348  IFI NTERM .GT. NTERM XX) NTERM = NTERM
LINE 349  GO TO G6ACK, (145, 185, 330)
LINE 350  C
LINE 351  C
LINE 352  C
LINE 353  800 CONTINUE
LINE 354  C
LINE 355  DO 810 I=1,NFREQ
LINE 356  810 SKE1(I) = 0.0
LINE 357  C
LINE 358  NTERM = MAX0( NTERM-L, L)
LINE 359  DO 820 j=1,NTM
LINE 360  SAVE = TVV(jj) & TVW(jj)
LINE 361  DO 820 j=1,NFREQ
LINE 362  SKE1(I) = SKE1(I) & SAVE*CFT(Ij)
LINE 363  820 CONTINUE
LINE 364  C
LINE 365  IFI NTERM .EQ. 1 ) GO TO G6ACK, (150, 190, 34)
LINE 367  SAVE = TVR(INTER) & TVR(INTER!
LINE 368  TO 830 I=1,1FREQ
LINE 369  SKL = SKAI(1)
LINE 370  IF( KCONV .GT. 1000 ) GO TO 330
LINE 371  DIK = ABS( SKAI(I) - SKL )
LINE 372  RER( = 1.0
LINE 373  IF( SKAI(I) .NE. 0.0 ) RERD = DIK/ABS(SKAI(I))
LINE 374  IF( RERD .LT. 0.35 ) GO TO 830
LINE 375  IF( SKAI(I) .EQ. 0.0 ) .AND. DIK .LT. .ELT ) GO TO 830
LINE 376  C WE HAVE NOT CONVERGED
LINE 377  IF( KCONV .EQ. 0 ) WRITE( 16, 4009 )
LINE 378  KCONV = KCONV + 1
LINE 379  WRITE( 15, 4010 ) KCONV, IFC, IIN, ITERN, FREU(I), SKAI(I), SKL, RERO
LINE 380  IF( KCONV .LE. 1000 ) GO TO 330
LINE 381  WRITE( 16, 4011 )
LINE 382  IF( FLUSH = 1 )
LINE 383  830 CONTINUE
LINE 384  GO TO GOBACK, ( 150, 190, 340 )
LINE 385  C
LINE 386  GO TO GOBACK, ( 150, 190, 340 )
LINE 387  C
LINE 388  4009 FORMAT( 1HL, 12D1, 1F1 ) // 10X, 85H THE VELOCITY INFLUENCE COMPUTATIONS
LINE 389  IN DID NOT CONVERGE FOR AT LEAST ONE PAIR OF ELEMENTS // 15X,
LINE 390  2 3HSEC, 5X, 4HIFL, 4X, 5HTERMS, 5X, 9H FREQUENCY, 8X, 5HFINAL,
LINE 391  3 12X, 8HPREVIOUS, 9X, 8HRELATIVE / 15X, 4HELEM, 4X, 4HELEM, 3IX,
LINE 392  4 5HVALUE, 12X, 5HVALUE, 12X, 5HERROR / 1.
LINE 393  4010 FORMAT( 11F1, 1F1 ) // 15X, 75H WARNING - AT LEAST 1000 VELOCITY
LINE 394  INFLUENCE COMPUTATION/DID NOT CONVERGE. / 15X,
LINE 395  4011 FORMAT( 11X, 12D1, 1F1 ) // 75H WARNING - AT LEAST 1000 VELOC
LINE 396  4015 FORMAT( 11X, 12D1, 1F1 ) // 75H WARNING - AT LEAST 1000 VELOC
LINE 397  END

/ ENDCF REAC

END CF JOB.

43.1 SEC. USED  .912 HRS. CHARGED  49.906 HRS. REMAINING
DISPLAY...FILE DECK 45

LINE 1  SUBROUTINE WETA ( NET, JMODE, \ldots, WETA, 113)  04500010
LINE 2  DIMENSION WETA (NET, 10), \ldots, WETA (NET, 10)  04500020
LINE 3  REWIND 113  04500030
LINE 4  \texttt{DO 30 JM=1, JMODE}  04500040
LINE 5  \texttt{WRITE (113) JM, NET, (WETA(I, JM), I=1, NET), (WETA(I, JM), I=1, NET)}  04500050
LINE 6  \texttt{30 CONTINUE}  04500060
LINE 7  \texttt{END FILE 113}  04500070
LINE 8  \texttt{REWIND 113}  04500080
LINE 9  \texttt{RETURN}  04500090
LINE 10 \texttt{END}  04500100
DISPLAY...FILE DECK 46

LINE 1  SUBROUTINE WRITEO (K,AROW,IZ,ANEW,NET)  
LINE 2  DIMENSION AROW(NET), IZ(NET), ANEW(NET)  
LINE 3  COMMON /X1112/ KST12,KST12  
LINE 4  COMMON /TAPE/ 19,110,111  
LINE 5  NNZ=0  
LINE 6  DO 100 I=1,NET  
LINE 7  IF (AROW(I) .EQ. 0.0) GO TO 100  
LINE 8  NNZ = NNZ+1  
LINE 9  M = NNZ  
LINE 10  IZ(M) = I  
LINE 11  ANEW(M) = AROW(I)  
LINE 12  100 CONTINUE  
LINE 13  IF (NNZ.EQ. 0) RETURN  
LINE 14  CALL WRITE1 111,K,NNZ,IZ,ANEW  
LINE 15  KST12 =K  
LINE 16  RETURN  
LINE 17  END
DISPLAY...FILE

LINE 1  SUBROUTINE WRITE(111, K, NNZ, IZ, ANEW)  04700010
LINE 2  DIMENSION 1Z(NNZ), ANEW(ANZ)        04700020
LINE 3  WRITE (111)  K, NNZ, IZ, ANEW               04700030
LINE 4  RETURN                                   04700040
LINE 5  END                                       04700050
CISPLAY...FILE

DECK 49

LINE 1  SUBROUTINE WTEPHT ( DEDT, NET, PHIW, IZ, I12, WTE)  04900010
LINE 2  COMPLEX DEDT(1), PHIW(1), WTE(1)  04900020
LINE 3  DIMENSION IZ(1)  04900030
LINE 4  COMMON /K1112/ KST11, KST12  04900040
LINE 5  KEL = 0  04900050
LINE 6  DO 300  IEL=1, NET  04900060
LINE 7  IF (IEL .GT. KST12) GO TO 300  04900070
LINE 8  IF (KEL .LT. IEL)  04900080
LINE 9  IREAD (I12) KEL, NNZ, (IZ(I1), I=1, NNZ), (WTE(I), I=1, NNZ)  04900090
LINE 10  204 IF (KEL .LE. IEL) GO TO 300  04900100
LINE 11  DD 200 M=1, NNZ  04900110
LINE 12  K= IZ(M)  04900120
LINE 13  I = IEL  04900130
LINE 14  DEDT(I) = DEDT(I) - WTE(M) * PHIW(K)  04900140
LINE 15  200 CONTINUE  04900150
LINE 16  300 CONTINUE  04900160
LINE 17  REWIND 112  04900170
LINE 18  RETURN  04900180
LINE 19  END  04900190

-6.106-
SUBROUTINE WVNY

COORDINATES OF RECEIVING PT. (CENTER OF REC. ELE.)
COORDINATES OF UPPER END OF LINE
ZDZD IS THE SQUARE OF THE DIFFERENCE OF THE REC AND INFL Z'S
DIFFERENCE BETWEEN Z'S OF INFL. AND RECEIV. ELEM.
SIDE WASH AND UPNASH INTEGRALS
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
COMMON / WV1 / NTERM, V(30)
COMMON / WW1 / XO, YO
DATA NGT / 5 /

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

TESTC = SS - 1.0
AMZ = AA - SS*ZDZD
ACZ = ACZ - ZDZD
SIGNU = SIGN(P102, EUP)
SIGNL = SIGN(P102, ELW)

F-0
SORC = ABS(TE STC 1)
IF(SORC .LT. 0.001) GO TO 22

IF( TESTC .GT. 0.0) 30 TO 22
LINE 57 C C .LT. 0 ( TESTC = C ) COMPUTE S2 05000570
LINE 58 FU = SIGNU 05000580
LINE 59 IF( RUP .GT. 0.0 ) FU = -.5 ( SLOPE*ZUP-EUP)/SQACZ 05000590
LINE 60 FL = SIGNL 05000600
LINE 61 IF( RLW .GT. 0.0 ) FL = -.5 ( SLOPE*ZLW-ELW)/SQACZ 05000610
LINE 62 FO = - ( FG - FL )/SQRC 05000620
LINE 63 GO TO 30 05000630
LINE 64 20 CONTINUE 05000640
LINE 65 C C = 0 05000650
LINE 66 FO = ( RUP - RLW ) / ( SLOPE*ALPHA ) 05000660
LINE 67 GO TO 30 05000670
LINE 68 22 CONTINUE 05000680
LINE 69 FU = SQRC*RUP & SLOPE*ZUP - EUP 05000690
LINE 70 FL = SQRC*RLW & SLOPE*ZLW - ELW 05000700
LINE 71 IF( RLW .EQ. 0.0 ) GO TO 26 05000710
LINE 72 IF( RUP .EQ. 0.0 ) GO TO 24 05000720
LINE 73 C C .GT. 0 NEITHER RUP NOR RLW IS ZERO 05000730
LINE 74 FO = ( ALOG(FU/FL) )/SQRC 05000740
LINE 75 GO TO 30 05000750
LINE 76 24 CONTINUE 05000760
LINE 77 26 CONTINUE 05000770
LINE 78 FO = ( ALOG( ABS(SQACZ/FL) ) )/SQRC 05000780
LINE 79 GO TO 35 05000790
LINE 80 26 CONTINUE 05000800
LINE 81 FO = ( ALOG( ABS(FU/SQACZ) ) )/SQRC 05000810
LINE 82 C FO HAS BEEN COMPUTED 05000820
LINE 83 30 CONTINUE 05000830
LINE 84 C FIND ALL ( ALL IS CALLED L1 IN THE NOTES ) 05000840
LINE 85 ALU = 0.0 05000850
LINE 86 ALL = ZUP - RUP 05000860
LINE 87 IF( RUP .GT. 0.0 ) ALU = 0.5*ALOG( ( ZUP+RUP)/ALL ) 05000870
LINE 88 1 ALL = 0.0 05000880
LINE 89 AL1 = ZLW - RLW 05000890
LINE 90 IF( RLW .GT. 0.0 ) ALL = 0.5*ALOG( ( ZLW+RLW)/ALL ) 05000900
LINE 91 2 ALL = 0.0 05000910
LINE 92 C AL1 = ALU - ALL 05000920
LINE 93 30 CONTINUE 05000930
LINE 94 C FIND S1 05000940
LINE 95 S1 = 0.0 05000950
LINE 96 IF( ZD .EQ. 0 ) GO TO 40 05000960
LINE 97 FL = ALPHA*ELW - SZZ 05000970
LINE 98 SL = SQRT( TTLW*ACZ ) 05000980
LINE 99 FU = ALPHA*EUP - SZZ 05000990
LINE 100 IF( RUP .EQ. 0 ) GO TO 32 05001000
LINE 101 SU = SQRT( TTLUP*ACZ ) 05001010
LINE 102 IF( RLW .EQ. 0 ) GO TO 34 05001020
LINE 103 31 S1 FOR RUP .GT. 0, RL .GT. 0, ZD .GT. 0 AND TD .NE. 0 05001030
LINE 104 S1 = ARSIN(FU/SU) - ARSIN(FL/SU) 05001040
LINE 105 GO TO 40 05001050
LINE 106 32 CONTINUE 05001060
LINE 107 S1 = SIGN( PID2, FU ) - ARSIN(FL/SU) 05001070
LINE 108 GO TO 40 05001080
LINE 109 34 CONTINUE 05001090
LINE 110 S1 = ARSIN(FU/SU) - SIGN( PID2, FL ) 05001100
LINE 111 40 CONTINUE 05001110
LINE 112 C 05001120
LINE 113 C F1 AND F2 05001130
LINE 114 C 05001140
LINE 115 F1 = ( ALPHA*ZD*S1/AMZ ) & SZZ+ALL/AMZ 05001150
LINE 116 F2 = SLOPE+F1 - AL1 05001160
LINE 117 C 05001170
LINE 118 W(1) = RUP*EUP/TTUP - RLW*ELW/TTLW - TESTC*F0 & SS*F1 - SLOPE+F2 05001180
IF (NTERM .LE. 1 ) GO TO 71

W(2) =
1 - 0.5*(ZUP*ALPHA)*EUP + (ZLW*ALPHA)*ELW*ALU
2 - RUP*EUP + RLW*ELW - 0.5*(AA-3.0)*TESTC*ZDZD*FD
3 - (1.5*SZZ-2.0*AA)*F1 + 3.5*SZZ*F2

IF (NTERM .LE. 2 ) GO TO 71

NHT = NTERM - 2
IWT = NTERM - 1

FIND 5 PT. GAUSS-LEFENORE POINTS AND COEFFICIENTS (GETA,GCOF)

F1 = 0.5*(EUP - ELW)
F2 = 0.5*(ELW + EUP)

DO 3 IG=193
N = 6 - IG
FO = F1*XX(IG)
GETA(1) = FO + F2
GETA(IG) = -FO + F2
GCOF(IG) = F1*AG(IG)
GCOF(N) = GCOF(IG)

LINE 140 43

N=3*NTERM
V(N) = 0.0
W(N) = 0.0

LOOP ON TERMS OF QUADRATURE

ETA = GETA(IG)
ZETA = SLOPE*ETA + ALPHA

IF( ZETA .LE. 0.0 ) GO TO 70
TTHH = ETA*ETA + ZDZD
RR = ETA - ZETA - TTHH
IF( RR .LE. 0.0 ) GO TO 70

COF = GCOF(IG)
RH = SORT( RR )
FL = ZETA - RH
ALH = 0.0
IF( (FL/ZETA) .GT. 1.0E-6 ) ALH = 0.5*ALOG(FL)/ZETA

H(1) = ZETA*ALH - RH

DO 50 N=1,NHT
N2 = N+1

H(N) = 9*H(N-1) - (N-1)*TTHH*H(N)/N2

CONTINUE

DO 60 N=2,1,NHT

N2 = N+1

H(N) = 9*H(N-1) - (N-1)*TTHH*H(N)/N2

CONTINUE

CONTINUE

CONTINUE

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

-6.110-
LINE 1  SUBROUTINE ZFDZI (ZFUN, DZDX, X, Y, A) 05100010
LINE 2  C SUBROUTINE TO EVALUATE FUNCTION (ZFUN) AND DERIVATIVE (DZDX) 05100020
LINE 3  DIMENSION X(I), Y(I), A(I) 05100030
LINE 4  COMMON /ZFUNY/ N, IEHF, B1, R2, R3, AFUNMX 05100040
LINE 5  COMMON /XSCL / XX, YY, XG, YG, BREF, ZZZ, ITEST 05100050
LINE 6  DATA EPS/1.E-78/ 05100060
LINE 7  XP = XG*BREF*XX 05100070
LINE 8  YP = YG*BREF*YY 05100080
LINE 9  ZFUN = B1*B2*XPMXI*YP 05100090
LINE 10  DZDX = R2 05100100
LINE 11  DO I =1,N 05100110
LINE 12  XPMXI = XP - X(I) 05100120
LINE 13  R2 = XPMXI*XPMXI & (YP-Y(I))**2 05100130
LINE 14  R2PEPS = R2 & EPS 05100140
LINE 15  XXLOG = ALOG(R2PEPS) 05100150
LINE 16  ZFUN = ZFUN & A(I)*R2 & XXLOG 05100160
LINE 17  DZDX = DZDX & 2.0*A(I)*XPMXI*(XXLOG & R2/R2PEPS) 05100170
LINE 18  CONTINUE 05100180
LINE 19  RETURN 05100190
LINE 20  END 05100200

END OF JOB.

53.6 SEC. USED 0.015 HRS. CHARGED 49.937 HRS. REMAINING