UNCLASSIFIED

AD NUMBER

AD894188

NEW LIMITATION CHANGE

TO
Approved for public release, distribution unlimited

FROM
Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; AUG 1971. Other requests shall be referred to Air Force Flight Dynamics Laboratory, ATTN: FY, Wright-Patterson AFB, OH 45433.

AUTHORITY

AFFDL ltr dtd 2 May 1979

THIS PAGE IS UNCLASSIFIED
PREDICTION OF UNSTEADY AERODYNAMIC LOADINGS OF NON-PLANAR WINGS AND WING-TAIL CONFIGURATIONS IN SUPersonic FLOW

Part II: Computer Program Description

GORDON D. KRAMER
GEORGE E. KEYLON
THE BOEING COMPANY
COMMERCIAL AIRPLANE GROUP

TECHNICAL REPORT AFFDL-TR-71-108, PART II

MARCH 1972

Distribution limited to U.S. Government agencies only; test and evaluation; statement applied August 1971. Other requests for this document must be referred to AF Flight Dynamics Laboratory (FY), Wright-Patterson AFB, Ohio 45433.

AIR FORCE FLIGHT DYNAMICS LABORATORY
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO
NOTICE

When Government drawings, specifications, or other data, are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AIR FORCE: 30-3-72/80
PREDICTION OF UNSTEADY AERODYNAMIC LOADINGS OF NON-PLANAR WINGS AND WING-TAIL CONFIGURATIONS IN SUPersonic FLOW

Part II Computer Program Description

GORDON D. KRAMER
GEORGE E. KEYLON
FOREWORD

Part II of this report was prepared by Boeing Computer Services, Inc., in conjunction with The Boeing Company, Commercial Airplane Group, Renton, Washington for the Aerospace Dynamics Branch, Vehicle Dynamics Division, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, under Air Force Contract AF 33615-70-C-1126, "Unsteady Aerodynamics and Flutter of Wing Horizontal Tail Configurations in Supersonic Flow". This work was conducted under Project No. 1370, "Dynamic Problems in Military Flight Vehicles" and Task No. 137003, "Prevention of Dynamic Aeroelastic Instabilities in Advanced Military Aircraft". Part II of this report is intended to describe in detail the computer program developed under the above contract, and is subsidiary to Part I, Theoretical Development, Program Usage, and Application.

Mr. William S. Rowe of The Boeing Company served as supervisor of the work covered under this contract. The co-authors of this part of the report worked from engineering analyses prepared and documented in Part I by Dr. J. M. II, C. J. Borland and J. R. Hogley. Mr. Lawrence J. Huttsell of the Aerospace Dynamics Branch (FYS) was Project Engineer. Appreciation is expressed to Mr. H. Huffman and Mr. R. Hirst for graphical work done in this report.

Manuscript was released by the authors in August, 1971 for publication as an AFFDL report.

This report has been reviewed and approved.

WALTER MYKYTOW
Assistant for Research & Technology
Vehicle Dynamics Division
Air Force Flight Dynamics Laboratory
ABSTRACT

The Mach box technique has been extended to include wing and tail with
dihedral angles and vertical separation. A digital computer program, written
in FORTRAN, is presented. The program provides for up to nine sweep angles of
the leading and trailing edges of each surface. First order piston theory
thickness correction is available as an option, and two refinement procedures
are provided, subdivision with averaging and velocity potential smoothing.
For a maximum of twenty oscillatory mode shapes the program calculates
normal washes, velocity potentials, lifts, pressures and generalized forces
matrices. If only one surface is being analyzed, sampling of wake up-wash,
side-wash and longitudinal wash is available.
# TABLE OF CONTENTS:

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II COMPUTER PROGRAM DESCRIPTIONS</td>
<td></td>
</tr>
<tr>
<td>1. General Description</td>
<td>2</td>
</tr>
<tr>
<td>2. General Purpose Subroutines</td>
<td>4</td>
</tr>
<tr>
<td>3. Main Control Program</td>
<td>9</td>
</tr>
<tr>
<td>4. Data Input Processor</td>
<td>10</td>
</tr>
<tr>
<td>5. Geometry Processor</td>
<td>11</td>
</tr>
<tr>
<td>6. Modal Data Processor</td>
<td>27</td>
</tr>
<tr>
<td>7. Aerodynamic Influence Coefficients Section</td>
<td>34</td>
</tr>
<tr>
<td>8. Normal-washes and Velocity Potentials</td>
<td>44</td>
</tr>
<tr>
<td>9. Velocity Potential Smoothing Sections</td>
<td>58</td>
</tr>
<tr>
<td>10. Generalized Air Forces Section</td>
<td>61</td>
</tr>
<tr>
<td>11. Common Block Organization</td>
<td>64</td>
</tr>
<tr>
<td>12. Array Storage</td>
<td>71</td>
</tr>
<tr>
<td>13. Internal Scratch Files</td>
<td>75</td>
</tr>
<tr>
<td>14. Output Files</td>
<td>81</td>
</tr>
<tr>
<td>15. Implementation and Debugging</td>
<td>84</td>
</tr>
<tr>
<td>III PROGRAM FLOWCHARTS</td>
<td></td>
</tr>
<tr>
<td>CONTROL</td>
<td>86</td>
</tr>
<tr>
<td>DATAPP</td>
<td>87</td>
</tr>
<tr>
<td>GEOMBX</td>
<td>88</td>
</tr>
<tr>
<td>MODES</td>
<td>89</td>
</tr>
<tr>
<td>VICMAIN</td>
<td>90</td>
</tr>
<tr>
<td>NWLPT</td>
<td>91</td>
</tr>
<tr>
<td>SMTH</td>
<td>92</td>
</tr>
<tr>
<td>CHORDF</td>
<td>93</td>
</tr>
<tr>
<td>FORCES</td>
<td>94</td>
</tr>
<tr>
<td>REFERENCES</td>
<td></td>
</tr>
<tr>
<td>APPENDIX A: Sample Input and Output Data</td>
<td>A1</td>
</tr>
<tr>
<td>APPENDIX B: Program Listings</td>
<td>B1</td>
</tr>
</tbody>
</table>
## LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>xxii</td>
</tr>
<tr>
<td>2.</td>
<td>xxiv</td>
</tr>
<tr>
<td>3.</td>
<td>3</td>
</tr>
<tr>
<td>4.</td>
<td>15</td>
</tr>
<tr>
<td>5.</td>
<td>18</td>
</tr>
<tr>
<td>6.</td>
<td>21</td>
</tr>
<tr>
<td>7.</td>
<td>21</td>
</tr>
<tr>
<td>8.</td>
<td>35</td>
</tr>
<tr>
<td>9.</td>
<td>35</td>
</tr>
<tr>
<td>10.</td>
<td>36</td>
</tr>
<tr>
<td>11.</td>
<td>50</td>
</tr>
<tr>
<td>12.</td>
<td>53</td>
</tr>
<tr>
<td>13.</td>
<td>54</td>
</tr>
<tr>
<td>14.</td>
<td>71</td>
</tr>
<tr>
<td>15.</td>
<td>71</td>
</tr>
<tr>
<td>16.</td>
<td>72</td>
</tr>
<tr>
<td>17.</td>
<td>72</td>
</tr>
<tr>
<td>18.</td>
<td>73</td>
</tr>
<tr>
<td>19.</td>
<td>73</td>
</tr>
<tr>
<td>20.</td>
<td>74</td>
</tr>
<tr>
<td>21.</td>
<td>82</td>
</tr>
<tr>
<td>22.</td>
<td>83</td>
</tr>
<tr>
<td>23.</td>
<td>A1</td>
</tr>
</tbody>
</table>

*Note: Figure 18 is 'AIC Integration, Full Box' on page 35, Figure 19 is 'AIC Integration, Edge Boxes' on page 35, Figure 20 is 'AIC Integration, Apex Boxes' on page 36, Figure 21 is 'Tip Chord Trailing Edge Velocity Potential Calculation' on page 50, Figure 22 is 'Subdivided "Effective Area"' on page 53, Figure 23 is 'Unsubdivided Boxes outside the "Effective Area"' on page 54,*
<table>
<thead>
<tr>
<th>Mathematical Symbol</th>
<th>FORTRAN Symbol</th>
<th>Dimension</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>length/time</td>
<td>Speed of Sound = U/M</td>
</tr>
<tr>
<td>A(n,m)</td>
<td>A</td>
<td>area</td>
<td>Integration area of box n,m</td>
</tr>
<tr>
<td>b₁</td>
<td>B1</td>
<td>length</td>
<td>Chordwise dimension of Mach box.</td>
</tr>
<tr>
<td>b₁S</td>
<td>B1S</td>
<td>length</td>
<td>b₁/N_S = chordwise dimension of a subdivided box.</td>
</tr>
<tr>
<td>Cᵥᵣ</td>
<td>C</td>
<td>non-dimensional</td>
<td>Velocity potential spatial aerodynamic influence coefficient (AIC).</td>
</tr>
</tbody>
</table>
| Cᵥᵣᵩ      | CKRN  | non-dimensional| Spatial AIC giving velocity potential at a point on surface "xy" due to constant outward normal wash over a box on surface "ab"; possible values for superscripts are: x or a \{R-right \{U-wing  \\ L-left \{T-tail  \\  
| Cᵥᵣᵩ       | CKRN  | non-dimensional| Velocity potential planar AIC                                            |
| Cᵥᵣᵩᵩ      | SKRN  | non-dimensional| Planar AIC defined for subdivided sending boxes.                         |
| ΔCᵣᵩᵩⁿ,m  | DELCP  | 1/length       | Pressure coefficient difference at box n,m for the j'th mode (program output)  |
| lᵩ,c         | --              | length         | Local reference chord                                                  |
### NOMENCLATURE

<table>
<thead>
<tr>
<th>Mathematical Symbol</th>
<th>FORTRAN Symbol</th>
<th>Dimension</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_{i,j}^m$</td>
<td>--</td>
<td>1/length</td>
<td>Local lift coefficient per unit span for the $j$th mode</td>
</tr>
<tr>
<td>$c_{m,j}^m$</td>
<td>SECWM</td>
<td>1/(length)$^2$</td>
<td>Local moment coefficient per unit span for the $j$th mode</td>
</tr>
<tr>
<td>$c_{rW}$</td>
<td>--</td>
<td>length</td>
<td>Wing root section chord length.</td>
</tr>
<tr>
<td>$c_{rT}$</td>
<td>--</td>
<td>length</td>
<td>Tail root section chord length.</td>
</tr>
<tr>
<td>$D$</td>
<td>--</td>
<td>1/time</td>
<td>Substantial derivative; $D = \frac{\partial}{\partial t} + U \frac{\partial}{\partial x}$</td>
</tr>
<tr>
<td>$f_j(x,y)$</td>
<td>DEFS(1,L)</td>
<td>non-dimensional</td>
<td>$j$th mode shape deflection at location ($X,Y$).</td>
</tr>
<tr>
<td>$f_{ij}$</td>
<td>--</td>
<td>non-dimensional</td>
<td>Deflection of $i$th lumped mass in mode $j$</td>
</tr>
<tr>
<td>$\frac{\partial f_j(x,y)}{\partial x}$</td>
<td>DEFS(2,L)</td>
<td>1/length</td>
<td>Slope of $j$th mode shape function.</td>
</tr>
<tr>
<td>$\tilde{f}_{j}$</td>
<td>--</td>
<td>time</td>
<td>Scaled modal displacement at box $n,m$</td>
</tr>
<tr>
<td>$J$</td>
<td>--</td>
<td>1/length</td>
<td>$j$th mode shape deflection / $a$</td>
</tr>
<tr>
<td>$i$</td>
<td>--</td>
<td>mass x length$^2$</td>
<td>Moment of inertia about the elastic axis of the $i$th lumped mass</td>
</tr>
<tr>
<td>$f_{ij}$</td>
<td>--</td>
<td>force/length</td>
<td>Generalized stiffness</td>
</tr>
<tr>
<td>$s$</td>
<td>XKS</td>
<td>Non-dimensional</td>
<td>Reduced frequency based on leading planform semi-span, $k_m = \frac{\omega s}{U}$</td>
</tr>
<tr>
<td>Mathematical Symbol</td>
<td>FORTRAN Symbol</td>
<td>Dimension</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------</td>
<td>-----------</td>
<td>------------</td>
</tr>
<tr>
<td>$k_1$</td>
<td>$\Xi KI, Kl$</td>
<td>non-</td>
<td>Reduced frequency based on the chordwise dimension of the Mach box</td>
</tr>
<tr>
<td></td>
<td></td>
<td>dimensional</td>
<td>$k_1 = \frac{\omega b_1}{U}$</td>
</tr>
<tr>
<td>$F_1$</td>
<td>$K1BAR$</td>
<td>non-</td>
<td>$k_1 \frac{h^2}{\beta^2}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>dimensional</td>
<td></td>
</tr>
<tr>
<td>$L$</td>
<td>$CAPL$</td>
<td>non-</td>
<td>Vertical separation of the center lines of the 2 surfaces, positive upward measuring from the wing to the tail.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>dimensional</td>
<td></td>
</tr>
<tr>
<td>$L_{n,m}^j$</td>
<td>--</td>
<td>force/length</td>
<td>Lift on box $n,m$ for the $j$th mode</td>
</tr>
<tr>
<td>$L_{m}^j$</td>
<td>--</td>
<td>force/length</td>
<td>Lift on the $m$th chordwise strip of boxes for the $j$th mode.</td>
</tr>
<tr>
<td>$L_{j}$</td>
<td>--</td>
<td>force/length</td>
<td>Lift on a complete half-surface or half-airplane for the $j$th mode.</td>
</tr>
<tr>
<td>$L_{n,m}^j$</td>
<td>--</td>
<td>force/length</td>
<td>Amplitude of box lift $L_{n,m}^j$</td>
</tr>
<tr>
<td>$L_{m}^j$</td>
<td>--</td>
<td>force/length</td>
<td>Amplitude of section lift $L_{m}^j$</td>
</tr>
<tr>
<td>$L_{j}$</td>
<td>--</td>
<td>force/length</td>
<td>Amplitude of total lift $L_{j}$</td>
</tr>
<tr>
<td>$L_{n,m}^{j}^{BXL1FT}$</td>
<td>--</td>
<td>non-</td>
<td>Non-dimensional amplitude of box lift (program output)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>dimensional</td>
<td>$L_{j}^{n,m} = L_{j}^{n,m} \text{int} \left( \frac{b_1}{\beta} \right) L_{j}^{n,m} \text{sec}$</td>
</tr>
</tbody>
</table>

viii
## NOMENCLATURE

<table>
<thead>
<tr>
<th>Mathematical Symbol</th>
<th>FORTRAN Symbol</th>
<th>Dimension</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>L&lt;sub&gt;j&lt;/sub&gt;</td>
<td>SLIFT</td>
<td>non-dimensional</td>
<td>nondimensional amplitude of section lift (program output)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( n \cdot L_j - L_j e^{-i\omega t} \equiv \frac{b_1}{\beta} L_j e^{-i\omega t} )</td>
</tr>
<tr>
<td>L&lt;sub&gt;j&lt;/sub&gt;</td>
<td>TLIFT</td>
<td>non-dimensional</td>
<td>Nondimensional amplitude of total lift (program output)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( L_j - L_j e^{-i\omega t} \equiv \frac{b_1}{\beta} L_j e^{-i\omega t} )</td>
</tr>
<tr>
<td>l</td>
<td>EL</td>
<td>non-dimensional</td>
<td>The c coordinate location of a pulse receiving point, i.e., the perpendicular distance from the sending plane to receiving point.</td>
</tr>
<tr>
<td>M</td>
<td>XMACH</td>
<td>non-dimensional</td>
<td>Mach number</td>
</tr>
<tr>
<td>m&lt;sub&gt;jj&lt;/sub&gt;</td>
<td>mass</td>
<td></td>
<td>Generalized mass for the j&lt;sup&gt;th&lt;/sup&gt; mode</td>
</tr>
<tr>
<td>m&lt;sub&gt;i&lt;/sub&gt;</td>
<td>mass</td>
<td></td>
<td>i&lt;sup&gt;th&lt;/sup&gt; lumped mass</td>
</tr>
<tr>
<td>n&lt;sub&gt;x&lt;/sub&gt;, m&lt;sub&gt;y&lt;/sub&gt;</td>
<td>ENRUS</td>
<td>non-dimensional</td>
<td>Normal wash at box n,m on surface &quot;xyz&quot; due to local source strength, where possible subscript values are:</td>
</tr>
</tbody>
</table>
|                     | ENRLS<sub>x</sub>, ENL<sub>y</sub>, ENSUBD |                        | \[
|                     |                |                        | x = \{R-right \ L-left \} \ y = \{U-upper \ L-lower \} \ z = \{W-wing \ T-tail \} |
|                     |                |                        | e.g. \( N_{n,m} \) means normal wash on the right upper wing at box n,m   |

ix
## NOMENCLATURE

<table>
<thead>
<tr>
<th>Mathematical Symbol</th>
<th>FORTRAN Dimension Definition</th>
<th>Dimension</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>ENRULW</td>
<td>non-</td>
<td>Normal wash at box (n,m) on surface &quot;xyz&quot; due to remote source strengths on surface &quot;abc&quot;, where possible subscript values for (a,b,) and (c) are the same as for (x,y,) and (z,) respectively, defined above; e.g. (\hat{A}_{n,m}) means normal RUT LLW wash at box (n,m) on the right upper tail due to source strengths on the left lower wing.</td>
</tr>
<tr>
<td>N</td>
<td>ENRLLL</td>
<td>non-</td>
<td>No. of &quot;sub-boxes&quot; (chordwise and spanwise) to be used in the odd integer subdivision improvement technique.</td>
</tr>
<tr>
<td>n</td>
<td>--</td>
<td>non-</td>
<td>the (n) coordinate location of a pulse receiving point.</td>
</tr>
<tr>
<td>(n_c, m_c, l_c)</td>
<td>---</td>
<td>---</td>
<td>Sending Surface Coordinate System</td>
</tr>
<tr>
<td>(\bar{n}_c, \bar{m}_c, \bar{l}_c)</td>
<td>---</td>
<td>---</td>
<td>Receiving Point Coordinate System</td>
</tr>
<tr>
<td>(p, p(x,y,t))</td>
<td>---</td>
<td>force/area</td>
<td>local static pressure</td>
</tr>
<tr>
<td>(P_w)</td>
<td>---</td>
<td>force/area</td>
<td>Free stream static pressure</td>
</tr>
<tr>
<td>(\Delta p(x,y,t))</td>
<td>---</td>
<td>force/area</td>
<td>pressure difference between upper and lower surfaces at point ((x,y)) at time (t) (\Delta p(x,y,t) = p(x,y,t)^{\text{upper}} - p(x,y,t)^{\text{lower}})</td>
</tr>
</tbody>
</table>
### NOMENCLATURE

<table>
<thead>
<tr>
<th>Mathematical Symbol</th>
<th>FORTRAN Symbol</th>
<th>Dimension</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta \bar{p}(x,y) )</td>
<td>--</td>
<td>force/area</td>
<td>Amplitude of pressure difference: ( \Delta p(x,y,t) = \Delta \bar{p}(x,y) e^{i \omega t} )</td>
</tr>
<tr>
<td>( q_{ij} )</td>
<td>--</td>
<td>force/length</td>
<td>Generalized force due to the deformation in the ( i )th elastic mode and loading for the ( j )th modal deflections</td>
</tr>
<tr>
<td>( \bar{q}_{ij} )</td>
<td>--</td>
<td>force/length</td>
<td>Amplitude of generalized force</td>
</tr>
<tr>
<td>( \bar{Q}_{ij} )</td>
<td>GENAF</td>
<td>non-dimensional</td>
<td>Non-dimensional generalized force (program output); ( Q_{ij} = \bar{Q}<em>{ij} e^{i \omega t} = \frac{b_i}{\sqrt{\beta}} \bar{Q}</em>{ij} e^{i \omega t} )</td>
</tr>
<tr>
<td>( \bar{Q}_{ij} )</td>
<td>QAGARD</td>
<td>( 1/(\text{length})^2 )</td>
<td>Generalized force in the AGARD notation</td>
</tr>
<tr>
<td>( q_{ij}, q_{ij}'' )</td>
<td>GENAFC</td>
<td>( 1/(\text{length})^2 )</td>
<td>Real and imaginary parts of ( \bar{Q}_{ij} ) in the AGARD definition (program output)</td>
</tr>
<tr>
<td>( q )</td>
<td>--</td>
<td>force/area</td>
<td>dynamic pressure</td>
</tr>
<tr>
<td>( q_j(t) )</td>
<td>--</td>
<td>length</td>
<td>Generalized coordinate relating physical deflection to ( j )th modal deflections: ( Z(x,y,t) = \sum_j f_j(x,y) q_j(t) )</td>
</tr>
<tr>
<td>( q_j )</td>
<td>--</td>
<td>length</td>
<td>Amplitude of ( j )th generalized coordinate</td>
</tr>
<tr>
<td>( s )</td>
<td>S</td>
<td>length</td>
<td>Wing semi-span.</td>
</tr>
</tbody>
</table>
**NOMENCLATURE**

<table>
<thead>
<tr>
<th>Mathematical Symbol</th>
<th>FORTRAN Symbol</th>
<th>Dimension</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>--</td>
<td>length²</td>
<td>Area of integration. Bounded by edge of planform plus diaphragm and lying inside the forward Mach cone of the receiving point.</td>
</tr>
<tr>
<td>t</td>
<td>--</td>
<td>time</td>
<td>Time</td>
</tr>
<tr>
<td>U</td>
<td>--</td>
<td>length/time</td>
<td>Free stream velocity.</td>
</tr>
<tr>
<td>( { } )</td>
<td></td>
<td>non-</td>
<td>Velocity spatial AIC for perturbation velocity parallel to the free stream.</td>
</tr>
<tr>
<td>( { } )</td>
<td>( { } )</td>
<td>length/time</td>
<td>Perturbation velocity in the stream-wise direction, positive downstream.</td>
</tr>
<tr>
<td>( { } )</td>
<td>( { } )</td>
<td>non-</td>
<td>Velocity spatial aerodynamic influence coefficient (AIC) for velocity normal to the free stream and parallel to the sending surface.</td>
</tr>
<tr>
<td>( { } )</td>
<td>( { } )</td>
<td>non-</td>
<td>Spatial AIC giving velocity normal to the free stream and parallel to surface &quot;ab&quot; at a point on surface &quot;xy&quot; or in the flowfield (FF), due to constant normal wash over a box on surface &quot;ab&quot;</td>
</tr>
<tr>
<td>( { } )</td>
<td>( { } )</td>
<td>non-</td>
<td>Perturbation velocity in the span-wise direction, positive right (looking upstream).</td>
</tr>
<tr>
<td>( { } )</td>
<td>( { } )</td>
<td>non-</td>
<td>Velocity spatial aerodynamic influence coefficient (AIC) for velocity norm⁻¹ to the sending plane.</td>
</tr>
</tbody>
</table>
## NOMENCLATURE

<table>
<thead>
<tr>
<th>Mathematical Symbol</th>
<th>FORTRAN Symbol</th>
<th>Dimension</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>((xy))</td>
<td>(W)</td>
<td>non-</td>
<td>Spatial AIC giving velocity normal to surface &quot;ab&quot; at a point on surface &quot;xy&quot; or in the flow field (FF) due to a constant normal wash over a box on surface &quot;ab&quot;.</td>
</tr>
<tr>
<td>(a_{ab})</td>
<td></td>
<td>dimensional</td>
<td></td>
</tr>
<tr>
<td>(f_{ab})</td>
<td></td>
<td>length/</td>
<td>Perturbation velocity in the vertical direction, positive upward.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>time</td>
<td></td>
</tr>
<tr>
<td>(x, y, z)</td>
<td></td>
<td>length</td>
<td>Reference (global) coordinate system, X positive aft, Y positive right, Z positive upward.</td>
</tr>
<tr>
<td>Mathematical Symbol</td>
<td>FORTRAN Symbol</td>
<td>Dimension</td>
<td>Definition</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------</td>
<td>-----------</td>
<td>------------</td>
</tr>
<tr>
<td>$X_w, Y_w, Z_w$</td>
<td></td>
<td></td>
<td>Wing local coordinate system.</td>
</tr>
<tr>
<td>$X_T, Y_T, Z_T$</td>
<td></td>
<td></td>
<td>Tail local coordinate system</td>
</tr>
<tr>
<td>$X_{WLE}$</td>
<td>$XWLE$</td>
<td>length</td>
<td>The location of a leading edge definition point of the wing planform, measured along the $X_w$ coordinate.</td>
</tr>
<tr>
<td>$X_{TLE}$</td>
<td>$XTLE$</td>
<td>length</td>
<td>Same as above for the tail planform, measured along the $X_T$ coordinate.</td>
</tr>
<tr>
<td>$X_{WTE}$</td>
<td>$XWTE$</td>
<td>length</td>
<td>The location of a trailing edge definition point of the wing planform.</td>
</tr>
<tr>
<td>$X_{TTE}$</td>
<td>$XTTE$</td>
<td>length</td>
<td>Same as above for the tail, measured along the $X_T$ coordinate.</td>
</tr>
<tr>
<td>${X_F}$</td>
<td>$XEDGE$</td>
<td>length</td>
<td>Location of the edge center of a Mach box used for the placement of the box pattern, measured along the $X_w$ coordinate.</td>
</tr>
<tr>
<td>${X_C}$</td>
<td>$XCENTR$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>${X_{CW}}$</td>
<td></td>
<td></td>
<td>The location of the most upstream row of boxes on the wing measured along the $X_w$ co-ordinate.</td>
</tr>
<tr>
<td>${X_{CT}}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mathematical Symbol</td>
<td>FORTRAN Symbol</td>
<td>Dimension</td>
<td>Definition</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------</td>
<td>-----------</td>
<td>------------</td>
</tr>
<tr>
<td>YWL(E)</td>
<td>YWLE</td>
<td>length</td>
<td>The location of a leading edge definition point of the wing tail planform measured along coordinate ( Y_W ).</td>
</tr>
<tr>
<td>YTI(E)</td>
<td>YTLE</td>
<td>length</td>
<td>[ \text{definition point of the wing tail planform measured along coordinate } Y_T ].</td>
</tr>
<tr>
<td>( \bar{y} )</td>
<td>YBAR</td>
<td>non-dimension</td>
<td>Offset of receiving chord from the nearest sending chord.</td>
</tr>
<tr>
<td>(( z_u ))</td>
<td></td>
<td>length</td>
<td>{ Upper } surface deflection at ( (x,y) ) as a function of time ( { Lower } )</td>
</tr>
<tr>
<td>(( z_L ))</td>
<td></td>
<td>length</td>
<td>mean surface deflection: ( Z_m(x,y,t) = f_j(x,y)e^{i\omega t} )</td>
</tr>
<tr>
<td>( z_T(x,y) )</td>
<td></td>
<td>length</td>
<td>Local thickness at ( (x,y) )</td>
</tr>
<tr>
<td>( \frac{\partial z_T}{\partial x} (x,y) )</td>
<td>TSLFN</td>
<td>non-dimension</td>
<td>Local thickness slope at ( (x,y) )</td>
</tr>
<tr>
<td>( \bar{z}_T )</td>
<td></td>
<td>non-dimension</td>
<td>Thickness slope piston theory correction; ( \bar{z}_T = 1 + \frac{Y^{0.5} \partial z_T}{2} )</td>
</tr>
<tr>
<td>( n,m )</td>
<td>ALPHA</td>
<td>non-dimension</td>
<td>Edge box area ratio for box((n,m)).</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>BETA</td>
<td>non-dimension</td>
<td>( \sqrt{\alpha^2 - 1} )</td>
</tr>
<tr>
<td>Mathematical Symbol</td>
<td>FORTRAN Symbol</td>
<td>Dimension</td>
<td>Definition</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------</td>
<td>-----------</td>
<td>------------</td>
</tr>
<tr>
<td>( b_1 / \beta )</td>
<td>BIBETA</td>
<td>length</td>
<td>Spanwise dimension of the Mach box.</td>
</tr>
<tr>
<td>( b_1 )</td>
<td>BIBTAS</td>
<td>length</td>
<td>Spanwise dimension of a subdivided Mach box.</td>
</tr>
<tr>
<td>( \gamma )</td>
<td>GAMMA</td>
<td>non-dimensional</td>
<td>Ratio of specific heats, ( \gamma = 1.4 )</td>
</tr>
<tr>
<td>( \zeta )</td>
<td>ZETA</td>
<td>non-dimensional</td>
<td>Dummy variable in the ( Z_w ) or ( Z_T ) coordinate.</td>
</tr>
<tr>
<td>( \bar{\eta} )</td>
<td>ETABAR</td>
<td>non-dimensional</td>
<td>Dummy variable of integration in the ( \bar{m}_c ) coordinate.</td>
</tr>
<tr>
<td>( \theta )</td>
<td>THETA</td>
<td>radians</td>
<td>( \sin^{-1} \frac{\bar{\eta}}{\ell} = \sin^{-1} \frac{\eta}{\sqrt{\ell^2 - \lambda^2}} )</td>
</tr>
<tr>
<td>( \theta_{ij} )</td>
<td>--</td>
<td>radians/length</td>
<td>Torsion of ( i )th lumped mass in mode ( j ).</td>
</tr>
<tr>
<td>( \bar{\lambda} )</td>
<td>--</td>
<td>non-dimensional</td>
<td>( \bar{\lambda}_c ) coordinate location of a pulse sending box.</td>
</tr>
<tr>
<td>( \mu )</td>
<td>MU,YMU</td>
<td>non-dimensional</td>
<td>( \mu_c ) coordinate location of a pulse sending box.</td>
</tr>
<tr>
<td>( \bar{\mu} )</td>
<td>YMUBAR</td>
<td>non-dimensional</td>
<td>( \bar{\mu}_c ) coordinate location of a pulse sending box.</td>
</tr>
</tbody>
</table>
## NOMENCLATURE

<table>
<thead>
<tr>
<th>Mathematical Symbol</th>
<th>FORTRAN Symbol</th>
<th>Dimension</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V )</td>
<td>NU,XNU</td>
<td>non-dimensional</td>
<td>( \eta ) coordinate location of a pulse sending box.</td>
</tr>
<tr>
<td>( \bar{V} )</td>
<td>XNUBAR</td>
<td>non-dimensional</td>
<td>( \eta ) coordinate location of a pulse sending box.</td>
</tr>
<tr>
<td>( \xi )</td>
<td>XI</td>
<td>non-dimensional</td>
<td>Dummy variable of integration in the ( X ) or ( X_T ) coordinate.</td>
</tr>
<tr>
<td>( \bar{\xi} )</td>
<td>XIBAR</td>
<td>non-dimensional</td>
<td>Dummy variable of integration in the ( \eta ) coordinate.</td>
</tr>
<tr>
<td>( \rho, \rho_\infty )</td>
<td>--</td>
<td>Mass/Volume</td>
<td>Free stream density.</td>
</tr>
<tr>
<td>( \tau )</td>
<td>TAU</td>
<td>non-dimensional</td>
<td>( \sqrt{\frac{x^2}{\xi^2} - \lambda^2} )</td>
</tr>
<tr>
<td>( \varnothing(x,y,t) )</td>
<td>--</td>
<td>length(^2)/time</td>
<td>Disturbance velocity potential at point ((x,y)) and time (t), defined so that ( \varnothing ) is velocity, ( \frac{\partial \varnothing}{\partial x_i} ) positive in positive ( x_1 ) direction, where ( x_1 = X, Y, ) or ( Z )</td>
</tr>
<tr>
<td>( \Delta \varnothing(x,y,t) )</td>
<td>--</td>
<td>length(^2)/time</td>
<td>Disturbance velocity potential difference between the top and bottom side of the surface at point ((x,y)) and time (t): ( \Delta \varnothing(x,y,t) = \varnothing_{\text{upper}} - \varnothing_{\text{lower}} )</td>
</tr>
<tr>
<td>Mathematical Symbol</td>
<td>FORTRAN Symbol</td>
<td>Dimension</td>
<td>Definition</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------</td>
<td>----------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>( \Delta \bar{\phi}_j(x,y) )</td>
<td>--</td>
<td>length / time</td>
<td>Amplitude of velocity potential difference at point ((x,y)) or for box (n,m) for the (j^{th}) mode</td>
</tr>
<tr>
<td>( \Delta \tilde{\phi}_j(x,y) )</td>
<td>DELPHI non-dimensional</td>
<td></td>
<td>Non-dimensional velocity potential difference due to the unit (j^{th}) generalized coordinate (program output);</td>
</tr>
<tr>
<td>( \Delta \tilde{\phi}_j^{n,m} )</td>
<td></td>
<td></td>
<td>( \Delta \tilde{\phi}_j^{n,m} = \Delta \theta_j^{n,m} e^{i\omega t} = \frac{U}{b_j} \left( \frac{b_j}{\gamma} \right)^{n,m} e^{i\omega t} )</td>
</tr>
<tr>
<td>( \Delta \theta_j^{m \text{TE}} )</td>
<td>TVP non-dimensional</td>
<td></td>
<td>Trailing edge velocity potential difference</td>
</tr>
<tr>
<td>( \psi_w )</td>
<td>PSIW ( \text{degrees} )</td>
<td></td>
<td>Dihedral angle of (\text{wing}), radians, positive upwards from horizontal.</td>
</tr>
<tr>
<td>( \psi_T )</td>
<td>PSIT ( \text{degrees} )</td>
<td></td>
<td>Dihedral angle of (\text{tail}), radians, positive upwards from horizontal.</td>
</tr>
<tr>
<td>( \omega )</td>
<td>--</td>
<td>radians / time</td>
<td>Circular frequency</td>
</tr>
<tr>
<td>( \omega_j )</td>
<td>--</td>
<td>radians / time</td>
<td>Circular frequency of mode (j)</td>
</tr>
</tbody>
</table>

Superscripts

\( (n,m) \)
\( (\nu,\mu) \) Box location

Subscripts

L Lower limit of Integration; Left-hand surface; Lower surface
R Right-hand Surface
S Subdivided
T Tail
U Upper limit of Integration; Upper Surface
W Wing
FF Flowfield
GLOSSARY OF TERMINOLOGY

**Aftmost Box** - Each chord on each planform and diaphragm combination has one such box. It is the aftmost box on that chord for which AIC arrays must be calculated and may be on the planform or diaphragm.

**AIC** - Aerodynamic Influence Coefficient

**Area Ratio** - On-planform fraction of a box which is cut by the planform boundary.

**Apex Box** - The box on the sending surface which encloses the apex of the Mach hyperbola associated with the receiving box.

**Box Grid** - Non-dimensionalized geometric array of boxes whose extent is determined by the geometric properties of the planforms. The term "grid" embraces the arrays on both surfaces.

**Control Point** - The location at which a receiving box is deemed to be influenced by other boxes. In general, the center of the receiving box.

**Effective Area** - A concept which relates entirely to the sub-division technique. It is composed of those boxes sufficiently close to the receiving box that their influence on it is large enough for the subdivision refinement to affect results significantly. The size has been arbitrarily set to include the $\frac{N_{\text{BOX}}}{N_{S}}$ rows immediately ahead of the receiving box.

**Global Co-ordinate System** - An overall reference system of co-ordinates. For example, the airplane co-ordinate system $X \sim$ aft, $Y \sim$ right, $Z \sim$ up. $Y = 0$ at centerline of airplane.

**Leading Edge Diaphragm** - All diaphragms on which $\phi = 0$.

**Local Co-ordinate System** - A co-ordinate system lying in the plane of the surface. $x \sim$ aft, $y \sim$ root to right tip. $y = 0$ at center line of airplane.

**Longitudinal Separation** - Streamwise distance between the trailing edge of the wing and the leading edge of the tail, measured along the centerline.
GLOSSARY OF TERMINOLOGY

Mach Asymptote - The asymptote of the Mach hyperbola.

Mach Hyperbola - The intersection of the sending plane and the forward Mach cone of the receiving point. Since this is always non-dimensionalized, it is a rectangular hyperbola.

Map - A condensed description of a large amount of data which can be used to locate any desired data element. A map of a banded sparse matrix might consist of two numbers per row, the first being the first non-zero column of that row and the second being the band width for that row. The matrix itself could then be stored as band elements only.

Normal Offset - The $l_c$ distance between the sending box and the receiving point.

Parallel Offset - The $m_c$ distance between the sending box center and the receiving point.

Partial Box - A sending box which is cut by the Mach hyperbola but which is neither an apex box nor an edge box.

Planar A.I.C - An A.I.C defined by the geometric relation between a sending box and receiving box which lies in the same plane. $C_{\mu_\alpha,0}$ only.

Receiving Box - In defining the relationship between two boxes the receiving box is the box which can be influenced by the other box.

Receiving Chord - Those receiving boxes which lie on the same chord. The receiving chord is significant in that all the boxes lying on it use AIC arrays which are a subset of those for the aftmost box lying on that chord.

Sending Box - In defining the relationship between two boxes, the sending box is the box which influences the other box (c.f. Receiving Box).

Spatial A.I.C - An A.I.C defined by the geometrical relationship between two boxes which do not lie in the same plane. $C_{\mu_\alpha,0}$, $V_{\mu_\alpha,0}$, $W_{\mu_\alpha,0}$. 

xx
GLOSSARY OF TERMINOLOGY

Sub-box - A member of the array of boxes formed when the grid of sending boxes is subdivided. Note it refers to the small box which is a fraction of the large box, and not to a large box which has been subdivided.

Tail - The downstream surface.

Vertical Separation - The vertical distance between the center lines of the two surfaces. Positive if the second surface is above the first.

Wake Diaphragm - That part of the diaphragm where $\Delta P = 0$ due to the influence of a surface.

Wing - Upstream Surface - (E.g. a Canard could be referred to as a wing);
Figure 1 Coordinate Systems For A Right Wing

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Transformation</th>
<th>Definition</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Global or Reference Coordinate System. X positive Aft.</td>
<td>Wing Local Coordinate System, used to define wing leading and trailing edges. $X_T, Y_T, Z_T$ are similarly defined for the tail local axes.</td>
<td>Length</td>
</tr>
<tr>
<td>Y</td>
<td>$Y = Y_w = \cos \phi_w + (Z - WLAX)\sin \phi_w$</td>
<td></td>
<td>Length</td>
</tr>
<tr>
<td>Z</td>
<td>$Z = (Z - WLAX)\cos \phi_w - Y\sin \phi_w$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$n_c$</td>
<td>$(X - X_c)/b_1 + 1$</td>
<td>Sending Surface Coordinate System used to define box grid. The $(n_c, m_c)$ plane lies within the plane of the sending surface, in this case the right wing.</td>
<td>Non-dimensional</td>
</tr>
<tr>
<td>$m_c$</td>
<td>$Y_w/(b_1 / \beta) + 1/2$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$l_c$</td>
<td>$Z_w/(b_1 / \beta)$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

xxii
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Transformation</th>
<th>Definition</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_c$</td>
<td>$-n_c-n$</td>
<td>Receiving Point Coordinate</td>
<td>non-dimen</td>
</tr>
<tr>
<td>$m_c$</td>
<td>$-m_c-m$</td>
<td>Syste... parallel to the n_c,m_c,l_c coordinates but opposite in sign and having their origin at the pulse receiving point (n,m,l in the n_c,m_c,l_c coordinates)</td>
<td>non-dimen</td>
</tr>
<tr>
<td>$l_c$</td>
<td>$-l_c-l$</td>
<td></td>
<td>sional</td>
</tr>
</tbody>
</table>
YTT
ZW
xwT
A
0

Fiue2 CocvdineftSystnForA
RightTail
SYMBOL
TRANSFORMATION
DEFINITION
DIMENSION

\[ x_T = TLAX \]
\[ Y_T = Yeos \theta_T + (Z-TLAX)sin \theta_T \]
\[ Z_T = (Z-TLAX)cos \theta_T - Yeos \theta_T \]

Tail Local Coordinate System used to define tail leading and trailing edges.

length

\[ n_c \]
\[ m_c \]
\[ f_c \]

\[ \frac{X_T + TLAX - (WLAX + X_C)}{b_1 + 1} \]
\[ Y_T / (b_1 / \beta) + 1/2 \]
\[ Z_T / (b_1 / \beta) \]

Sending Surface Coordinate System. In this case the right tail is shown as the sending surface.

non-dimensional

Figure 2 Coordinate Systems For A Right Tail

xxiv
SECTION I

INTRODUCTION

Part II of this report describes the computer program written according to the analysis of Part I. Part II refers implicitly to Part I, Section III, Computer Program Usage, and material covered there is not repeated here. The program computes generalized unsteady air forces on a wing or wing and tail in supersonic flow, given geometric details of the surfaces and the oscillatory mode shapes of the surfaces. The surfaces may be coplanar, may have dihedral angles, and may be separated vertically. The Mach box technique may be used "straight", or three refinements may be applied: 1) Subdivision of the Mach boxes to improve velocity potentials, 2) Least-squares smoothing of calculated velocity potentials to eliminate roughness due to box representation of surface edges, 3) Piston theory correction for airfoil thickness. The refinements may be applied in any combination. As intermediate results, normal-wash, velocity potentials and wake sampling of upwash, sideward and longitudinal washes may be printed, all at box center locations. The box lifts, pressure distribution, section lifts and total lift are also available for each mode.
1. GENERAL DESCRIPTION

The computer program consists of a main (0,0) overlay, one primary level overlay, and eight secondary level overlays (see fig. (3)). The main (0,0) overlay is a general purpose driver, and easily can be used to incorporate other compatible programs with this one as a complete flutter system. Its sole function is to set up buffers and any other system oriented parameters, then call the primary level overlay, CONTROL.

Overlay CONTROL contains the basic logic of the program. It first calls secondary overlay DATAPP, which reads and processes the basic card data necessary for execution of the program. The resulting parameters are stored in labelled common blocks, accessible to all other overlays.

If PRVGEOM = .FALSE., overlay GEOMBX is next called. The planform geometry is read and processed to yield a disk file IGEOSC containing all internally necessary geometric parameters.

If PRVMODE = .FALSE., overlay CONTROL next calls overlay MODES. This area processes the three forms of modal data and places the results, evaluated at box centers, on scratch file MODESC.

Overlay CONTROL next enters a loop on reduced frequency. Each pass through the loop first executes overlay VICMAIN, which computes (or reads from previously saved tapes) all AIC arrays needed at the current reduced frequency. Next overlay NWVPMBX is called, to compute normal-washes, velocity potentials, and optional sample washes. If SMOOTH = .TRUE. overlay SMTH is called to do a least-squares surface fit of the resulting arrays. If CRDFIT = .TRUE. overlay CHORDF is called to smooth the values a chord at a time. The final overlay, FORCES, then computes box lifts, section lifts and generalized forces for any smoothed values first, then for the unsmoothed values. The desired results are printed as they are computed. The loop on reduced frequencies terminates at this point.

Overlay CONTROL reads the termination card which causes a transfer back to the execution of DATAPP (Recycle), the call of another overlay (if available), RETURN to the main (0,0) overlay, or EXIT to control cards.

The following sections give a more detailed description of all of the overlay main programs, and the major subroutines called by each.
Figure 3  Program Overlay Structure
1. GENERAL PURPOSE SUBROUTINES

Fortran Callable COMPASS Function SHIFT

Author: G. E. Keylon

Purpose: To shift the contents of a word left or right a specified number of bits, identical to the Fortran Extended capability.

Method: The word and the number of bits to be shifted are stored in machine registers. The word is then left circular shifted the number specified. This causes the word to be shifted left circular if the number is positive and right with sign extension if the number is negative. The result is left in register X6 so that this routine must be used as a function subprogram.

Usage:

INTEGER SHIFT

IWORD = SHIFT (NWORD,N)

Input
NWORD - The word to be shifted
N - The number to shift the word

If N is positive shift left circular.
If N is negative shift right with sign extension.

Output
IWORD has the results of the shift on NWORD.
Fortran Subprogram WRTEMX

Author: G. E. Keylon

Purpose: To write a matrix on a tape or disk file.

Method: The matrix is placed row-wise into a buffer in labelled common RWBUFF with all of the unused areas of its array omitted. The buffer is then written onto the specified tape or disk file with the Fortran BUFFER OUT statement. A 16 word header record is written in the same manner before each matrix. The header record contains matrix size, name and optional parameters.

Usage: CALL WRTEMX (IOUTFL, MXWRIT, RANDOU, NFS, NMS, LS, NMR, LWS, K, ID, A, ITYPE, M, N, PARM, IRR)

Input
IOUTFL - Tape number or left-justified file name.
MXWRIT - Logical variable, not used.
RANDOU - .T. Random File (not used)
          .F. Sequential File
NFS - Number of files to space before writing
NMS - Number of matrices to space before writing
LS - Level number to space (not used)
NMR - Name or number in random index (not used)
LWS - Level number of this matrix (not used)
K - Row dimension of array A.
    if Ks matrix is already in /RWBUFF/
ID - Array containing matrix name.
A - Array containing matrix
ITYPE - TYPE of matrix (i.e., real, complex, integer, null, mixed)
M - Row dimension of matrix
N - Column dimension of matrix
PARM - 10-word parameter array

Output
IRR - Error return
    0, no error
    1, matrix spacing is negative
    2, File spacing is negative
    4, M*N dimensions greater than buffer size
    1500+1, encountered EOF after matrix I
    while skipping matrices.
Fortran Subprogram COMBUF

Author: G. E. Keylon

Purpose: To put a complex matrix into a buffer prior to writing on file.

Method: The matrix is placed row-wise into a buffer with all of the unused areas of the matrix omitted. All of the real parts are stored in the first part of the buffer then all of the imaginary parts are placed immediately following the real.

Usage: CALL COMBUF (A, K2, M, N, BUFF)

Input:
A - Array that contains matrix, typed complex
K2 - 2x (row dimension of A)
M - Number of rows in matrix (not array size)
N - Number of columns in matrix (not array size)

Output:
BUFF - Buffer that will contain matrix
Fortran Subprogram READMX

Author: G. E. Keylon

Purpose: To read a matrix from tape or disk file.

Method: A 16-word header record and a matrix record are read from the specified file with BUFFERIN statements. The 16-word header record contains matrix size, name and optional parameters. The matrix is then placed in a given array in correct Fortran storage.

Usage: CALL READMX(INFILE, MXREAD, RANDIN, NFS, NMS, LS, NMR, K, NID, ID, ITYPE, LRS, A, M, N, PARM, IRR)

Input:

INFILE - Tape number or left justified alphanumeric file name
MXREAD - Logical variable (not used)
RANDIN - .T. Random File (not used)
        .F. Sequential File
NFS - Number of files to space before reading
NMS - Number of matrices to space before reading
LS - Level number to space (not used)
NMR - Random name or number (not used)
K - Row dimension of array A
    If K<0 matrix will be left in /RWBUFF/
NID - Number of words available in ID array

In/Out:

ID - Identification array
ITYPE - Real, diagonal, null, mixed, complex

Output:

LRS - Level number of matrix read (not used)
A - Array containing matrix
M - Row dimension of matrix
N - Column dimension of matrix
PARM - Array of numerical parameters stored with the matrix in the 16 word header record
IRR - Error return
    0, no error
    1, matrix spacing is negative
    2, file spacing is negative
    4, matrix dimensions illegal
    5, M GT. K
1500 + I, encountered EOF after matrix I while skipping matrices.
Fortran Subprogram CBUFFR

Author: G. E. Keylon

Purpose: To move a complex matrix from a buffer to a Fortran array.

Method: The matrix assumed stored row-wise in the buffer with all of the real parts followed by all of the imaginary parts. The conversion leaves the matrix in the array in typical Fortran storage.

Usage: CALL CBUFFER(A,K2,M,N,BUFF)

Input:
- K2 - 2x (row dimension of array A)
- M - Number of rows in matrix (not array size)
- N - Number of columns in matrix (not array size)
- BUFF - Buffer that contains matrix

Output:
- A - Array that will contain matrix in complex storage
3. MAIN CONTROL PROGRAM

Fortran Program CONTROL

Author: G. E. Keylon, G. D. Kramer

Purpose: To control the flow of the program to the various lower level overlay section.

Method: The program has all of the labeled common blocks so that information can be passed from lower levels to this program which will determine the program flow.

Usage: The CONTROL Program is a main routine. It is the only primary overlay section in the program. It calls all of the lower level or secondary overlay sections. It is called from the initial or main overlay section as follows:

CALL OVERLAY(6HAFMBOX, 1, 0, 0)

Common Input and Output:

This program does not input or create common values. It is the means by which common values are passed between the secondary overlays of the program.
4. DATA INPUT PROCESSOR

Fortran Program DATAPP

Author: G. E. Keylon

Purpose: To read most of the input data and set flags and options for use throughout the program. It prints the title and options for each run.

Method: The title and all the input options are read in. The heading is printed. The options are read under a NAMELIST format and flags set to default options unless read in.

Usage: The DATAPP program is the main program of a secondary overlay of the Mach Box program. It is called as an overlay section as follows:

CALL OVERLAY(CHAPMBOX, 1, 1, 0)

All input and output is through labeled common blocks.

Common Input:
PREVEX
OMACH
DEFAULT

Common Output:

<table>
<thead>
<tr>
<th>TITLE</th>
<th>ERR</th>
<th>SYM</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRVGEOM</td>
<td>XKUAL</td>
<td>MYPEW</td>
</tr>
<tr>
<td>PRVMODE</td>
<td>OPLAIC</td>
<td>MYPEW</td>
</tr>
<tr>
<td>DIHW</td>
<td>OSPAIC</td>
<td>COPLAN</td>
</tr>
<tr>
<td>DIHT</td>
<td>WTGEOM</td>
<td>NSUBDV</td>
</tr>
<tr>
<td>XMACH</td>
<td>WTNFAF</td>
<td>NSURF</td>
</tr>
<tr>
<td>NKVALS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XKI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XKS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NT5</td>
<td>WTB</td>
<td>SMOOTH</td>
</tr>
<tr>
<td>NT5</td>
<td>PRBOX</td>
<td>NDEG</td>
</tr>
<tr>
<td>INTAPE</td>
<td>PRPAIC</td>
<td>DPCCPPR</td>
</tr>
<tr>
<td>INPS</td>
<td>PRSAIC</td>
<td></td>
</tr>
<tr>
<td>ISMPLW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NPLAIC</td>
<td>PRMDS</td>
<td>MODCPR</td>
</tr>
<tr>
<td>NSPAIC</td>
<td>PROEF</td>
<td></td>
</tr>
<tr>
<td>PRNW</td>
<td>AICCPPR</td>
<td></td>
</tr>
<tr>
<td>NOUTP</td>
<td>PRUW</td>
<td></td>
</tr>
<tr>
<td>JOUPSP</td>
<td>PRSW</td>
<td></td>
</tr>
<tr>
<td>PRVP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OSAIC</td>
<td>PREL</td>
<td></td>
</tr>
<tr>
<td>FRL</td>
<td>GACPPR</td>
<td></td>
</tr>
<tr>
<td>PRSL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRNFAF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRDCP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRGNAC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5. GEOMETRY PROCESSOR

Fortran Program GEOMBX

Author: G. D. Kramer

Purpose: To read geometric data from cards and compute all necessary geometric parameters.

Method: Cards F through L are read in this section. As they are read they are printed, then checked for inconsistent or missing data, with suitable diagnostics. The leading and trailing edge data is checked in EDGCHK, then transformed to non-dimensional coordinates. Planform and diaphragm box code patterns are determined in BXCDPF and BXCDI, and optionally printed by PWTAIC. The fractional on-planform portion of all boxes cut by a planform edge is determined by OMAREA, which in turn calls ALPHAC and NTRCEP. If spatial AIC's are necessitated by non-zero dihedral angles or vertical separation of wing and tail, integer arrays MUAIC are determined for each AIC set (C,W,V). These serve as a map, so that only those AIC values needed will be calculated. The MUAIC arrays are computed in PWWAIC and PWTAIC. All resulting arrays are written on scratch file IGEOSC.

Usage: The GEOMBX program is the main program of a secondary overlay. It is called by:

CALL OVERLAY (6HAFMBX, 1, 2, 0)

Common Input:
- OMACH
- TITLE
- PRVGEOM
- DXHW
- DXHT
- XMACH

Common Output:
- COPLAN
- XSUBDV
- NSUBDV
- NSUBDV2
- NSUBCN
- B1
- B1BETA
- B1S
- BLBETAS
- XSUBDV
- MYBBW
- MYBBT
- MYBSW
- MYBST
- MYBST
- MYBBSW
- MYBBST
- XCBW
- IXBT
- XCENTR
- TLAX
- CAPL
Arrays output on scratch file IGEOSC:

IBOXW - Wing box codes (Wing and tail if COPLAN = .TRUE.)
IBOXT - Tail box codes
FEXLOC - Leading edge locations at chord centers
TEXLOC - Trailing edge locations at chord centers
ALPHA - Fractional areas of boxes cut by a planform edge
IJALPH - Locations of cut boxes, of the form (1000*J+I)
KPT - Table of contents for the MUAIC arrays (and AIC's)
MUAIC - Pointer array indicating where contributing boxes will be found for one spatial AIC set.
Fortran Subroutine EDGCHK

Author: G. D. Kramer

Purpose: Given the leading or trailing edge values, to check for illegal combinations.

Method: Either a leading or trailing edge is checked for monotonic increasing y-values, starting at zero. The last trailing edge value is compared with the previous last value. A leading edge is checked for monotonically increasing x-values.

Usage: DIMENSION XEDGE(10), YEDGE(10)

CALL EDGCHK (XEDGE, YEDGE, NEDGE, IEDGE, IRR)

Input Parameters:
- XEDGE - Array of X-values for edge location points
- YEDGE - Array of Y-values for edge location points
- NEDGE - Number of points to check
- IEDGE = 1, leading edge
- = 2, trailing edge

Output Parameter:
- IRR = 0, Successful
- = 1, Non-monotonic y-values
- = 2, Non-monotonic x-values, leading edge only
- = 4, Y-values not starting at zero
- = 8, Tip y-values not agreeing
- Other, additive combination of above conditions
Fortran Subroutine BXCDPF

Author: G. D. Kramer

Purpose: To generate on-planform box codes for one surface, and store them in a compressed format.

Method: For each (subdivided) chord, the location of the leading edge and trailing edge \((\text{FEXLOC}, \text{TEXLOC})\) is determined. Codes for all boxes between those values are then set to 1 in subroutine $\text{NCODER}$. The remainder of the box code array is not changed.

Usage: The subroutine is called by:

\[
\text{CALL BXCDPF}(\text{XLE, YLE, NLE, XTE, YTE, NTE, LSROWS, IBOX})
\]

Input Parameters:

- \(\text{XLE}\) = locations of leading edge, measured in \(n_c, m_c, l_c\) system
- \(\text{YLE}\) = \(y\) locations of leading edge, measured in \(n_c, m_c, l_c\) system
- \(\text{XTE}\) = locations of trailing edge, measured in \(n_c, m_c, l_c\) system
- \(\text{YTE}\) = \(y\) locations of trailing edge, measured in \(n_c, m_c, l_c\) system
- \(\text{NLE}\) = Number of leading edge definition points.
- \(\text{NTE}\) = Number of trailing edge definition points.
- \(\text{LSROWS}\) = Maximum number of subdivided rows allowed.

Input/Output Parameters:

- \(\text{IXBW}\) = Input: 0 indicates the wing is to be done.
- \(\text{IXBT}\) = Output: Subdivided row of first unsubdivided box center on the surface.

Output Parameters:

- \(\text{IBOX}\) = Compressed box codes, 1 for on-planform boxes found, unchanged elsewhere. See Figure 4.

Output Common Parameters:

- \(\text{MXBS}_{W,T}\) = Maximum X (aft) extension of the subdivided wing and tail pattern
- \(\text{MYBS}_{W,T}\) = Maximum Y (outward) extension of the subdivided on-planform wing and tail pattern
- \(\text{MXB}_{W,T}\) = Maximum X extension of the unsubdivided wing and tail pattern
FIGURE 4 Output from EXCDFF, Called for a Wing (N_SUBDW=3)
MYB \{ \frac{W}{T} \} \text{ Maximum Y extension of the unsubdivided}
\begin{align*}
&\text{wing} \\
&\text{tail}
\end{align*}
\text{ pattern}
\begin{align*}
\text{FEXLOC} &= \{ \text{Leading} \} \\
\text{TEXLOC} &= \{ \text{Trailing} \}
\end{align*}
\text{ edge X location at (subdivided) chord centers.}
Fortran Subroutine BXCDI

Author: G. D. Kramer

Purpose: Given an array indicating a pattern of on-planform Mach boxes, to determine the associated off-planform diaphragm boxes.

Method: Leading edge diaphragm boxes are first determined, followed by wake diaphragm boxes. The tip diaphragm is then determined as a function of the tip chord. For the wing, an integer array is interrogated to determine whether additional wake areas (and tip diaphragm) are needed for wing-tail interference.

Usage: The subroutine is called by:
CALL BXCDI (IWAKE, ISROWS, LSCHDS, IBOX)

Input Parameters:
IWAKE: Array of locations on the wing for aft-most unsubdivided box in each chord affecting a tail surface. Not used for the tail surface, first element = 0
ISROWS: Maximum number of subdivided rows allowed
LSCHDS: Maximum number of subdivided chords allowed

Input/Output
IBOX: Array of subdivided box codes, previously set 1 at planform locations by subroutine BXCDPF. See figure 5.

Common Input
MXBBSW: Maximum X extension of the subdivided [wing] box pattern, including diaphragm [tail]
IXBST: X-location of the first subdivided tail row
MYBSW: Maximum Y extension of the subdivided planform [wing] pattern [tail]
MYBST: Maximum Y extension of the subdivided [wing] pattern, including diaphragm [tail]
NSUBDV: Number of subdivisions

Common Output
MYBBSW: Modified, if necessary
MYBST: Maximum Y extension of the unsubdivided [wing] pattern, including diaphragm [tail]
Figure 5 Output from BXCDI, called for a Coplanar Wing and Tail (NSUBDV = 3)
Fortran Subroutine PRNTBC

Author: G. D. Kramer, G. E. Keylon

Purpose: Print the array of box codes, either all values or only unsubdivided box-center values

Method: The compressed box code array is decompressed using subroutine DCODER, one row at a time, and printed. If unsubdivided codes have been requested, only the control point values are printed.

Usage: CALL PRNTBC (IBOX, LBXCD, IROW, MXB, MYB, SUBD)

IBOX - Box code array
LBXCD - Row size of box code array
IROW - First row to print
MXB - Last row to print
MYB - Number of chords to print
SUBD - .T., subdivided box codes desired
       .F., unsubdivided (control point) box codes desired
Fortran Subroutine PWWAIC

AUTHOR:  G. D. Kramer

PURPOSE:  Given the box pattern and dihedral angle of the surface, to determine a pointer array (MUAIC) for one chord on the right surface which indicates contributing regions (if any) of the left surface on the given chord.

METHOD:  The geometric relationship of the sending surface to the receiving chord is first determined. Then for all rows, from the last receiving box forward to the forward edge of the box pattern, any sending boxes on the left surface are indicated in the MUAIC array.

USAGE:  CALL PWWAIC(WING,IBOX,LBXCD,IWAKE,JCOL)

Input Parameters:
WING    .T., wing is being considered.
         .F., tail is being considered.
IBOX     Array of box codes (IBOXW or IBOXT).
LBXCD    Length of array IBOX.
IWAKE    Array of locations of aft-most box to be considered on the wing. Ignored if WING = .F.
JCOL     Chord being considered (receiving).

Common Input:
PSI       Dihedral angle
NSUBDV    Number of subdivisions
XSUBDV    Number of subdivisions, real
NSUBDV2   NSUBDV/2
IXBW      Location of first unsubdivided box center.

Output Parameters:
The computed results are returned via common block MUAICS. They are:
SURF       Logical indicator - true means a sending surface was encountered.
MUAIC(2,50) Unsubdivided row "map" of sending box locations, see Figure 7.
EL         Normal offset of receiving chord from sending surface.
YEAR       Parallel offset of receiving chord.
NROWS      Number of rows considered.
FIGURE 6. Possible Arrangement of Sending Boxes, Left Surface to Right
(Actual sending boxes shaded.)

FIGURE 7. MUAIC Array Generated by PWWAIC for Figure 4.
Fortran Subroutine PWTAIC

Author: G. L. Kramer

Purpose: Given the box patterns and dihedral angles of the two surfaces, to determine pointer arrays (MJAIC arrays) for the right wing and the left wing contributing regions to a desired tail chord.

Method: The geometric relationship of the sending surfaces to the receiving chord is first determined. Then for all rows, from the last receiving box forward to the forward edge of the sending vox patterns, any sending boxes are indicated in the MJAIC arrays.

Usage: The subroutine is called, after suitable setup, by:

CALL PWTAIC (IBOXW, LBXCDW, IROW, JCOL, CAPLL, YMUVSP)

Input Parameters:

IBOXW - Array of wing box codes
LBXCDW - Row dimension of IBOXW
IROW - Unsubdivided receiving row number
JCOL - Unsubdivided receiving chord number
CAPLL - Vertical Separation of sending center line receiving center line
YMUVSP - \( \chi \) contribution due to vertical separation
\[ = \text{CAPLL} \cdot \sin(\chi) \]

Input Common Variables

PSIDIF = \( \phi_r - \phi_w \)
PSIT
NSUBDV
FSIW
MIBBW
MIBSW
IXBW

Output Parameters

The computed results are returned via common block MJAICS. They are:

\[
\text{SURF} \begin{cases} \text{T}, & \text{Contributing boxes were found on the right wing} \\ \text{F}, & \text{No contributions were found} \end{cases}
\]

\[
\text{SURFL} \begin{cases} \text{T}, & \text{Contributing boxes were found on the left wing} \\ \text{F}, & \text{No contributions were found} \end{cases}
\]
\{MUAIC\} Map of contributing boxes on the \{right\} wing, see Figure 7.

\{EL\} The normal offset between the wing \{right\} plane and the receiving point

\{YBAR\} The parallel offset between the nearest chord center on the wing \{right\} box pattern and the receiving point

\{NROWS\} Number of rows covered by the MUAIC array for the \{right\} wing contributions
Fortran Subroutine GMAREA

AUTHOR: G. E. Keylon, G. D. Kramer

PURPOSE: To compute the fractional on-planform portion of all planform boxes which are cut by a planform edge.

METHOD: For each chord, the X coordinates of the left side intercept, right side intercept, and any kinks within the box width are determined by subroutine NTRCEP for each planform edge cut by the chord (wing and/or tail). Then for each planform box on the chord, the routine determines whether any edge cuts the box or causes a contribution to the box area. For any affected box, subroutine ALPHAC is called to compute the fractional area, which is then stored in array ALPH, and its location is stored in array IJALPH as (J * 512 + I). The fraction may be greater than one, since it includes the planform area of any chordwise adjacent box whose center is off planform.

USAGE: The routine is called by:

CALL GMAREA (IBOX, LBXCD, WING, ALPHA, IJALPH, NALPH)

Input Parameters:

IBOX  Box code array
LBXCD Size of box code array
WING  .T., Wing or coplanar case
       .F., Tail

Input Common Parameters:

<table>
<thead>
<tr>
<th>COPLAN</th>
<th>MXBT</th>
<th>NSURP</th>
<th>XWLE</th>
<th>XTIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEXLOC</td>
<td>MXBW</td>
<td>NWLE</td>
<td>YWLE</td>
<td>YTIME</td>
</tr>
<tr>
<td>TEXLOC</td>
<td>MYBT</td>
<td>NWTE</td>
<td>XWTE</td>
<td>XTTE</td>
</tr>
<tr>
<td>IXBT</td>
<td>MYBW</td>
<td>NTLE</td>
<td>YWTE</td>
<td>YTTE</td>
</tr>
<tr>
<td>IXBW</td>
<td>NSUBDV</td>
<td>NTTE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Output Parameters:

ALPHA  Array of area multipliers
IJALPH Array of corresponding IJ locations, as (J * 512 + I)
NALPH  Number of fractions calculated
Fortran Subroutine NTRCEP

AUTHOR: G. E. Keylon

PURPOSE: To compute the X coordinates of the intersections of a planform edge with the sides and center of a chord.

METHOD: The routine determines in which interval of the edge the desired point lies. The x-coordinate of the point is then obtained from the standard two point equation of a line. This is done for all three points.

USAGE: The routine is called by:

CALL NTRCEP (J, YEDG, XEDG, L1, C1, R1, NBK1, K1, IDEX)

Input Parameters:

J = Chord number
YEDG = Arrays of Y locations of the edge definition points
XEDG = Arrays of X locations of the edge definition points
IDEX = 1, leading edge
2, trailing edge

Common Input Values (from local common block/LAREA/)

LEFT = Y-location of the left side of the chord
RIGHT = Y-location of the right side of the chord

Output Parameters

L1 = X coordinate of left side intersection
C1 = X coordinate of center line intersection
R1 = X coordinate of right side intersection

NBK1 = Number of edge definition points encountered between the left and right sides of the chord.

K1 = 0 if no edge definition point lies between the left and right sides of the chord.
= The first (leftmost) edge definition point number lying within the chord.
Subroutine ALPHAC

AUTHOR: G. E. Keylon, G. D. Kramer

PURPOSE: To compute the on-planform area of a box which is partially off the planform or which must include area from neighboring off-planform box(es) cut by a planform edge.

METHOD: If the box is the first box on the chord, or the last box on the chord, the box is divided spanwise into a series of trapezoids (or triangles) determined by planform edge definition points occurring within the chord. The areas of these trapezoids are then added, yielding \( \alpha \).

If the box is an interior box which is cut by one or more planform edge segments, the area is first set to one, then the area of the off-planform corner(s) determined as trapezoids or triangles is subtracted.

USAGE: The routine is called by

\[
\text{CALL ALPHAC (X, XLED, YLED, XTED, YTED, L1, C1, R1, NBK1, K1, L2, C2, R2, NBK2, K2, AREA)}
\]

Input Parameters:

- \( X \) = X coordinate of box center
- \( XLED \) = Planform leading edge definition points
- \( YLED \) = Planform leading edge definition points
- \( XTED \) = Planform trailing edge definition points
- \( YTED \) = Planform trailing edge definition points
- \( L1 \) = Left chord edge intersections with the planform
- \( C1 \) = Center leading edge
- \( R1 \) = Right leading edge
- \( NBK1 \) = Number of planform leading edge definition points within the chord
- \( K1 \) = First leading edge definition point within chord
- \( L2 \)
- \( C2 \)
- \( R2 \)
- \( NBK2 \)
- \( K2 \) = Same as above for trailing edge

Output Value:

\( \text{AREA} = \text{The desired box area, } \alpha \)
6. MODAL DATA PROCESSOR

Fortran Program MODES

AUTHOR: G. E. Keylon

PURPOSE: To read the modal input data, compute it by a least squares surface fitting routine or evaluate a polynomial equation with coefficients supplied by input and store this information on a scratch file.

METHOD: The information needed to determine the mode shapes is read in. The planform information is read from a scratch file created in the geometry section. The program then computes or reads the modal data at control points, orders the data and writes the data on a scratch file for use in following sections.

Subroutine ROPER is used to compute row pointers for storing box center modal values row-wise. Modal input from tape is handled by TAPMOD. If modal input option 2 was specified, FITTER is called to compute the surface fit polynomial coefficients. PRECOF is called if the coefficients from option 1 or 2 are to be printed. The coefficients are saved on a scratch file for future cycles, and the polynomial is evaluated at box centers, with the results stored on scratch file MODESC.

The program also has an option to read an array of Thickness slope function values derived from "Piston Theory" calculations. These values are input to an equation that computes the thickness correction factor.

\[ \Xi(x, y) = 1 + \frac{\gamma + 1}{2} \cdot M \cdot \frac{\partial z_T}{\partial x} \]  

where \( \gamma \) is ratio of specific heats for a perfect gas (1.40)

\( M \) is Mach number.

\( \frac{\partial z_T}{\partial x} \) is the thickness slope function values.

\( \Xi(x, y) \) is the thickness correction factor.

Input Methods:

(1) Polynomial Coefficient Input

The degree of a surface polynomial equation and the coefficients are read in. The deflection is then computed by the following polynomial equation:
Deflection = \sum_{i=1}^{\text{# of degrees}} \left( a_{i,0} x^i y^0 + \ldots + a_{0,i} x^0 y^i \right) (2)

where, point \((X,Y)\) is the coordinates of a box center in the planform local coordinate system, and \(a\) is the array of polynomial coefficients read in. The slope is computed by taking the derivative of the deflection in the \(X\) direction.

\[
\text{Slope} = \sum_{i=0}^{\text{# of degrees}} d \left( a_{i,0} x^i y^0 + \ldots + a_{0,i} x^0 y^i \right) / dx
\]

These equations are used to compute the deflection and slopes for all the planform boxes. The array of modal values is stored on a scratch file for use in the velocity potential and generalized forces sections of the program.

(2) Interpolation

The degree of a surface polynomial equation, the number of locations where deflections are to be given and the locations and deflections are read in. The deflections are perpendicular to the surface and the \((X,Y)\) locations are input in the planform local coordinate system. The program uses this data to fit a surface polynomial expression in the least squares error approximation. The routine that performs the surface fit is subroutine FITTER. This routine sets up an upper triangular, augmented matrix that represents the set of simultaneous linear equations formed by taking the partial derivatives of each deviation equation squared and setting it to zero. It then solves the set of simultaneous linear equations by using the Choleski square root method given in Reference 1. The solution is an array of polynomial coefficients that are used to compute the modal values in the same manner as method (1).
Modal Values at Box Centers

The values of the deflections and slopes are read in from cards or tape and stored on a scratch file for use in the velocity potential and generalized forces sections of the program. The values are stored in order of boxes within chord, and chords within planform. The order is fore to aft boxes, center most to tip chord and wing before tail. For card input, each chord begins on a new card. All of the mode shape for the wing will be read followed by all of the mode shapes for the tail.

USAGE:

The MODES program is the main program of a secondary overlay of the Mach Box program. It is called as follows:

CALL OVERLAY (6HAFMBOX, 1, 3, 0)

Input:
Uses labeled common blocks:
/PROBLM/
/GEOMTY/
/GEOM2/
/FILES/
/INFOCOMM/
/TAPEIO/
/MODES/
/RWBUFF/

Uses the following files:
IGEOSC

Output:
Output is stored on file:
MODESC
Fortran Subprogram ROPER

AUTHOR: G. E. Keylon

PURPOSE: To compute the row pointers indicating location of planform boxes.

METHOD: The subprogram uses the column pointers and determines the row pointers. The subprogram will also calculate pointers for a tail surface with overlapped planform and store the pointers after the first planform pointers.

USAGE: CALL ROPER

General labeled common blocks used:

/GEOMTY/
/GEOM2/

LOCAL labeled common blocks used:

/INDEX/ IS(100), NOC(100), JS(50), JOC(50)

Common Input:

IS(J) - The $i$th index of the first planform box on chord $J$.

NOC(J) - The number of planform boxes on chord $J$.

Common Output:

JS(I) - The $j$th index of the first planform box on row $I$.

JOC(I) - The number of planform boxes on row $I$. 
Fortran Subprogram FITTER

AUTHOR: G. E. Keylon

PURPOSE: To fit a surface in the least squares sense through a set of data points.

METHOD: The fitter routine is passed a set of ordered triplets and the degree of polynomial to fit. It is also given a scale factor if needed to scale the data to prevent the occurrence of arithmetic overflow or underflow. The program can fit real or complex data. The system of simultaneous linear equations that must be solved for employs the Choleski square root method (see Ref. 1). If the polynomial exceeds the maximum capability in either X or Y direction that degree is held and the other direction is allowed to use the full degree.

USAGE: CALL FITTER (M, N, X, Y, Z, C, CN, IDIM)

Input:

M - degree of polynomial equation
N - number of data points to fit curve through
X - Array of X coordinates
Y - Array of Y coordinates
Z - Array of Z coordinates
CN - scale factor
IDIM - Indicator of real or complex function
   = 1, function to fit is real
   = 2, function to fit is complex

Output:

C - Output polynomial coefficient array.
Fortran Subprogram MODOUT

AUTHOR: G. E. Keylon

PURPOSE: To print the mode shapes in a manner that the user can readily determine Mach box values of deflections and slopes.

METHOD: The mode shapes are rearranged in a print array so that one row or part of a row will be printed at a time. If there are more than 15 chords on the planform the program prints information for 15 chords, for all rows, and then prints for the next 15 chords until all information has been printed. The values may be scaled before printing to allow values to be printed under F mode Fortran format control. The scaling factor will be indicated in the title.

USAGE: CALL MODOUT (DEFSL, JS, JOC, NROWS, NM, IOVLAP)

Input:

DEFSL - Array of mode shapes
DEFSL(I,1) = deflection
DEFSL(I,2) = slope
JS - Array of pointers to first planform box on each row
JOC - Array of counters for the number of planform boxes on each row.
NROWS - number of rows
NM - Mode shape number
IOVLAP - Number of boxes of overlap between planforms for non-coplanar surfaces.

Output:

None
Fortran Subprogram PRECOF

AUTHOR: G. E. Keylon

PURPOSE: To print the polynomial coefficients used in evaluating mode shapes.

METHOD: The coefficients are printed with each coefficient having over it the corresponding powers of X and Y labeled. All the coefficients for a total power will on one line (i.e., line 1 - 0 power, line 2 - first power, line 3 - second power etc.).

USAGE: 

CALL PRECOF(IDEG, A, IFR)

Input:

Labeled common block /FILES/
IDEG - Degree of polynomial equation
A   - Array of coefficients
IFR - Flag indicating how coefficients are obtained.
      = 1, read from cards
      = 2, computed by least squares surface fit.

Output:

None
7. AERODYNAMIC INFLUENCE COEFFICIENTS SECTION

Fortran Program VICMAIN

AUTHOR: G. E. Keylon

PURPOSE: To determine all aerodynamics influence coefficients (AIC's) that must be computed or retrieved for a specified reduced frequency.

METHOD: A parameter array is read from the geometry scratch file for each spatial AIC that is needed. The program then determines if an array already exists on permanent tape storage. If it exists the array is read in, expanded if necessary, and stored on scratch file IAICSC if spatial, or in blank common if planar. If calculation is necessary, subroutine KERNEL is called to control the actual computations. KERNEL in turn calls ROMBER to do the integrations of FUNCT and VFUNC.

USAGE: The VICMAIN program is the main program of a secondary overlay of the Mach Box program. It is called as follows:

CALL OVERLAY (6HAFMB%X 1, 4, 0)

Input:

Uses labeled common blocks

/KERN/
/KVAL/
/PROBLM/
/FILES/
/GEOMTY/
/IOCONT/
/ARRAYS/
/RWBUFF/
/TAPEIO/

Uses the following files

IGEOSC, OSPAIC (optional), OPLAIC (optional)

Output:

Output is stored on files:

NPLAIC, NSPAIC, IAICSC (all optional)
Fortran Subprogram KERNEL

AUTHOR: G. E. Keylon

PURPOSE: To determine the boxes to be integrated and the limits of integration prior to calling the integration routine.

METHOD: The program determines from a parameter array from the geometry scratch file, the intersection of the Mach cone with the planform boxes it is attempting to integrate. It determines what boxes on a row are to be integrated and breaks each box up into a set of integrable limits. It then passes the limits of integration to subroutine ROMBER for integration by the Romberg integration method described in Reference 2.

Box Patterns and Limits:

- Hyperbola: One region to integrate

FIGURE 8 AIC Integration, Full Box

- Limit 1
- Limit 2

- Limit 3
- Limit 2
- Limit 1

One region to integrate

Two regions to integrate

FIGURE 9 AIC Integration Edge Boxes

35
Box patterns and limits for boxes cut on the left side by the Mach hyperbola are computed in a like manner.

The functions integrated by subroutine ROMBER are those for the velocity potential aerodynamic influence coefficients ($C_{\mu \lambda}$), the upwash aerodynamic influence coefficients ($W_{\phi \lambda}$), and the sidewash aerodynamic influence coefficients ($V_{\phi \lambda}$).

The equations for $C_{\mu \lambda}$ and $W_{\phi \lambda}$ are:

$$C_{\mu \lambda} = \frac{1}{\pi} \int_{-\infty}^{\infty} e^{i k \bar{y}} \left\{ J_0 \left( \frac{k_1}{M} \sqrt{\bar{y}^2 - \xi^2} \right) \right.$$ 

$$\sin^{-1}\left( \frac{\bar{y} L}{\sqrt{\bar{y}^2 - \xi^2}} \right) + \sum_{r=1}^{\infty} \frac{(-1)^r}{r} J_2 r \left( \frac{k_1}{M} \sqrt{\bar{y}^2 - \xi^2} \right)$$

$$\left[ \sin(2r \sin^{-1}\left( \frac{\bar{y} L}{\sqrt{\bar{y}^2 - \xi^2}} \right)) - \sin(2r \sin^{-1}\left( \frac{\bar{y} L}{\sqrt{\bar{y}^2 - \xi^2}} \right)) \right] d \bar{y}$$

$$W_{\phi \lambda} = \frac{1}{\pi} \int_{-\infty}^{\infty} e^{i k \bar{y}} \left\{ J_0 \left( \frac{k_1}{M} \sqrt{\bar{y}^2 - \xi^2} \right) \right.$$ 

$$\sin^{-1}\left( \frac{\bar{y} L}{\sqrt{\bar{y}^2 - \xi^2}} \right) + \sum_{r=1}^{\infty} \frac{(-1)^r}{r} J_2 r \left( \frac{k_1}{M} \sqrt{\bar{y}^2 - \xi^2} \right)$$

$$\left[ \sin(2r \sin^{-1}\left( \frac{\bar{y} L}{\sqrt{\bar{y}^2 - \xi^2}} \right)) - \sin(2r \sin^{-1}\left( \frac{\bar{y} L}{\sqrt{\bar{y}^2 - \xi^2}} \right)) \right] d \bar{y}$$
\[
\begin{align*}
W_{\rho, \lambda} &= \frac{\mathcal{L}}{\pi} \left[ \int_{\xi_l}^{\xi_u} e^{-i k \xi} \left\{ J_0 \left( \frac{\bar{\eta}_{\lambda}}{\sqrt{\xi^2 - \ell^2}} \right) \sin^{-1} \left( \frac{\bar{\eta}_{\lambda}}{\sqrt{\xi^2 - \ell^2}} \right) - \right. \\
&\quad \left. \sin^{-1} \left( \frac{\bar{\eta}_{\lambda}}{\sqrt{\xi^2 - \ell^2}} \right) \right\} + \sum_{r=1}^{\infty} \frac{(-1)^r}{r} J_{2r} \left( \frac{\bar{k}_{\lambda}}{M} \sqrt{\frac{\xi^2 - \ell^2}{\xi^2 - \ell^2}} \right) \right] \\
&\quad \left[ \sin \left( 2r \sin^{-1} \left( \frac{\bar{\eta}_{\lambda}}{\sqrt{\xi^2 - \ell^2}} \right) \right) - \sin \left( 2r \sin^{-1} \left( \frac{\bar{\eta}_{\lambda}}{\sqrt{\xi^2 - \ell^2}} \right) \right) \right] \frac{1 + i k \xi}{\xi} \, d\xi \\
&\quad + \left\{ J_0 \left( \frac{\bar{k}_{\lambda}}{M} \sqrt{\frac{\xi^2 - \ell^2}{\xi^2 - \ell^2}} \right) \right. \\
&\quad \left. - \sin \left( 2r \sin^{-1} \left( \frac{\bar{\eta}_{\lambda}}{\sqrt{\xi^2 - \ell^2}} \right) \right) \right\} \left[ \frac{\xi}{\xi_l} \right] \\
&\quad + e^{-i k \xi} \frac{\pi}{\ell} \\
\end{align*}
\] (5)

The \( W_{\rho, \lambda} \) equations is valid for all types of boxes, which are shown in Figures 8, 9, and 10. The \( W_{\rho, \lambda} \) equation, as written, is valid for the region of an apex box shown in Figure 10, that has the hyperbola as a boundary on both sides. The last term is zero for regions that have the hyperbola as a boundary for one side and the box edge as the other side boundary. The last term and the terms evaluated at the integration limits are zero for full boxes or regions bounded on both sides by the box edges. The values of the integrand used in ROMBER are computed by subroutine FUNCT which also calls subroutines RANGE and BESSEL to evaluate the Bessel functions.

The function \( W_{\rho, \lambda} \) is considerably different and holds for all regions to be integrated. The evaluation of the integral is done by subroutine VFUNEC. The equation is:
For cases where $E_1 = 0$, the equation becomes

$$
V_{0M} = \frac{1}{\pi k_1} \left[ \frac{-i \kappa \tilde{y}}{\tilde{y}^2} \left( \sin \left( \frac{\kappa_1}{M} \sqrt{\frac{\tilde{y}^2 - \eta^2}{\tilde{y}^2 - \eta^2}} \right) \right) \right] \tilde{y}_{u} + \int_{\tilde{y}_L}^{\tilde{y}_u} \left[ e^{-i \kappa \tilde{y}} \frac{1 + i \kappa \tilde{y}}{\tilde{y}^2} \left( \sin \left( \frac{\kappa_1}{M} \sqrt{\frac{\tilde{y}^2 - \eta^2}{\tilde{y}^2 - \eta^2}} \right) - \sin \left( \frac{\kappa_1}{M} \sqrt{\frac{\tilde{y}^2 - \eta^2}{\tilde{y}^2 - \eta^2}} \right) \right) \right] d \tilde{y} \tag{6}
$$

For cases where $k_1 = 0$, the equation becomes

$$
V_{0M} = -\frac{1}{\pi} \left[ \cosh^{-1} \left( \frac{\tilde{y}}{\sqrt{\eta^2 + \lambda^2}} \right) - \cosh^{-1} \left( \frac{\tilde{y}}{\sqrt{\eta^2 + \lambda^2}} \right) \right] \tilde{y}_{u} \tag{7}
$$

USAGE: CALL KERNEL(XMACH,K1,ERR,C,W,V)

Input:
XMACH  Mach number
K1     Reduced frequency
ERR    Convergence criteria (relative not absolute)

Labeled Common Block /VICPAR/

Output:
C  Velocity potential aerodynamic influence coefficients.
W  Upwash aerodynamic influence coefficients.
V  Sidewash aerodynamic influence coefficients.
Fortran Subprogram ROMBER

AUTHOR: G. E. Keylon

PURPOSE: To integrate the aerodynamic influence coefficient functions.

METHOD: The program uses the Romberg integration technique (Ref. 2).
The technique is a modified trapezoidal area method with an extrapolation method added. For analytical cases the sidewash aerodynamic influence coefficient will be solved by an analytic equation, not by numerical approximation.

USAGE: CALL ROMBER (XILL, XILU, IUC, ERR, IFLAG, KLBAR, YMUBAR, EL, XMACH, C, W, V)

Input:

XILL - Lower limit of integration
XILU - Upper limit of integration
IUC - Flag indicating type of box or edge condition of interval to be integrated = 0, full box
     1, left side of box is edge of Mach hyperbola.
     2, right side of box is edge of Mach hyperbola.
     3, both sides of box are edges of Mach hyperbola.
ERR - Convergence criteria (relative, not absolute)
IFLAG - Indicator of real or imaginary parts
         = 0, real part
         = 1, imaginary part
KLBAR - Function of reduced frequency and Mach number, \( \frac{K M^2}{(M^2 - 1)} \)
YMUBAR - Parallel offset of pulse sending box.
EL - Normal offset of receiving point from sending plane.
XMACH - Mach number

Output:

C - Velocity potential aerodynamics influence coefficient, \( C \sqrt{\mu\lambda} \)
W - Upwash aerodynamic influence coefficient, \( W \sqrt{\mu\lambda} \)
V - Sidewash aerodynamic influence coefficient, \( V \sqrt{\mu\lambda} \)
Fortran Subprogram FUNCT

AUTHOR: G. E. Keylon

PURPOSE: To evaluate the velocity potential and upwash aerodynamic influence coefficient functions for a set of independent variables.

METHOD: An array XI of independent variables is passed to the program through the calling sequence. The program evaluates the function at each point first checking for boundary conditions where the function approaches a singularity. Routines to find the range of and value of Bessel functions are called in the evaluation of the function.

USAGE: CALL FUNCT (K, XI, FXIC, FXIW, IFLAG, KIBAR, EL, YMUBAR, IUC, XMACH, BESSY)

Input:

K  - Number of functions to evaluate
XI - Array of independent variables
IFLAG - Indicator of real or imaginary part:
      = 0, real part
      = 1, imaginary part
KIBAR - Function of reduced frequency and Mach number, $K \cdot \frac{t^*}{t}$
EL  - Normal offset of receiving point from sending plane.
YMUBAR - Parallel offset of pulse sending box.
IUC - Flag indicating type of box or edge condition of interval to be integrated.
XMACH - Mach number

Output:

FXIC  - Function values for Velocity Potential AIC.
FXIW  - Function values for Upwash AIC.
BESSY - Evaluation at end points for upwash AIC.
Fortran Subprogram BESSEL

AUTHOR:  G. E. Keylon

PURPOSE:  To evaluate the Bessel functions for a given argument over a range
          of orders.

METHOD:  The argument and range (# of terms or order) is passed to the the
          routine. The routine then calculates the required terms and places
          them in an array and returns.

USAGE:  CALL BESSEL (K12, AV, NA)

Input:

  K12  - The argument, a function of independent variable, Mach
       number and reduced frequency.

  NA  - Highest order of the Bessel function to be evaluated.

Output:

  AV  - Array containing the Bessel functions.
Fortran Subprogram RANGE

AUTHOR: G. E. Keylon

PURPOSE: To determine the range (or order) of a Bessel function with a given argument.

METHOD: An order, or equation for an order, is given for various increments of arguments. This routine determines which interval the argument is in and computes the order.

USAGE: CALL RANGE (K12, NA)

Input:

K12 - The argument, function of independent variable, Mach number and reduced frequency.

Output:

NA - Highest order of the Bessel function to be evaluated.
Fortran Subprogram VFUNC

AUTHOR: G. E. Keylon

PURPOSE: To evaluate the sidewash aerodynamic influence coefficient function for a set of independent variables.

METHOD: An array of independent variables is passed to the program through the calling sequence. The program evaluates the function at each point, first checking for boundary conditions where the function approaches a singularity.

USAGE: CALL VFUNC (K, XI, FXIV, IFLAG, KIBAR, EL, YMUBAR, INC, XMACH, IND, VT)

Input:

K - Number of values to calculate
XI - Array of independent variables
IFLAG - Flag indicating real or complex part
  = 0, real part
  = 1, imaginary part
KIBAR - Function of reduced frequency and Mach number, $K_2 M^2/(M^2 - 1)$
EL - Normal offset of receiving box above sending plane.
YMUBAR - Parallel offset of pulse sending box.
INC - Flag indicating type of box edge condition of interval to be integrated.
XMACH - Mach number
IND - Indicator to calculate VT terms
  = 0, do not calculate
  = 1, calculate

Output:

FXIV - Function values for sidewash AIC.
VT - Extra terms calculated at the limits of integration.
8. NORMAL-WASHES AND VELOCITY POTENTIALS

Fortran Program NWLPT

AUTHOR: G. D. Kramer

PURPOSE: To compute normal washes and associated velocity potentials for each oscillatory mode shape at box centers. Wake sampling of upwash, sidewash and longitudinal wash is also provided.

METHOD: The necessary box patterns and other geometric items are first read in from the scratch file IGEOSC. The mode shape and velocity potential pointer array IPNTRM is read from scratch file MODESC, and a pointer array for normal-washes, IPNTDW, is generated by subroutine POINTR. These pointer arrays serve to associate a box location in a sparsely filled rectangular array with the corresponding mode, velocity potential or normal wash value in a singly dimensioned, densely filled array.

A loop on mode shapes is entered next. The box center deflections and shapes are read from MODESC into array DEFSL. Subroutine VELPOT is called for the wing to compute $N_{RW}$, $N_{RW}$, and $\Delta \phi$ at box centers, and trailing edge $\Delta \phi$ values in array TVP. If a tail is being analyzed as well, the contributing wing normal-washes are determined and VELPOT is called again. Optional printing of $N_{RW}$, etc. and $\Delta \phi$ is done in routine PRINTR.

If sampling of wake washes is desired, subroutine SMPLW is called to compute and print these results.

The $\Delta \phi$ array VELPOT and the TVP array are written on scratch file IVPSC for each mode shape.

USAGE: The NWLPT program is the main program of a secondary overlay of the Mach box program. It is called as follows:

CALL OVERLAY (GRAFBX, 1, 5, 0)

Input: Uses labeled common blocks

/CONTRL/ /FILES/
/PROBLM/ /XCONT/
/GEOMTY/ /TAPEIO/
/GEOM2/ /MODES/
/KERN/ /ARRAYS/
/KVAL/ /SAMPLW/

Uses scratch files

IGEOSC
MODESC

44
Fortran Subroutine POINTR

AUTHOR: G. E. Keylon, G. D. Kramer

PURPOSE: To generate part or all of a pointer array which indexes another array of box associated values (modes, normal-washes, etc.) stored compactly, row-wise.

METHOD: The box codes are scanned to determine the first box of interest and the number of boxes of interest on each row. From this, the pointer array is generated such that IPNTR(1,i) = the location of the first box value for row i, and IPNTR (2, i) = the chord number of the first box value for row i.

USAGE: The routine is called by:

```fortran
DIMENSION IBOX (LEXCD, # chords/10), IPNTR (2, MXIR)
LOGICAL DIAPH, SUBD, WING
CALL POINTR (IX, MX, MYB, IOVLAP, SUBD, DIAPH, IBOX, LEXCD, MXIR, 
  IPOINT, IPNTIN, IPNTR)
```

Input Parameters:

- **IX** = First row of the box pattern for which the pointer array is desired.
- **MX** = Number of rows desired.
- **MYB** = Maximum row length
- **IOVLAP** = Number of rows to allow for overlap (tail only).
- **SUBD** = .T., a pointer array for subdivided boxes is desired = .F., only unsubdivided box information is desired.
- **DIAPH** = .T., boxes in diaphragm areas are to be included.
  = .F., only on-planform boxes are of interest though space may be left within a row if imbedded diaphragm areas occur.
- **IBOX** = Array of subdivided box codes generated in the geometry section.
- **LEXCD** = Length of box code array.
- **MXIR** = Length of IPNTR array, used to control end-around buildup of the array.
- **IPOINT** = Value to be used for first pointer; 1 if IX=1, else the next location available in the array "pointed to" for row IX.

In/out Parameters:

- **IPNTIN** = Location of next available cell in the IPNTR array. This will be incremented for each row processed until MXIR is reached, when it is reset to 1.

- **IPNTR (2, MXIR)** = The pointer array, see Method above.
FOURMAN Subroutine GETAIC

Author: G. D. Kramer

Purpose: To get the desired Aerodynamic Influence Coefficient (AIC) arrays from scratch file IAICSC.

Method: From the calling sequence, the location of the desired AIC array(s) is determined. If they are in core, the routine returns. If there are none, the error flag is set. Otherwise, the disk file is positioned, and the desired arrays are read into local common block AICS.

Usage: EL, YBAR, NROWS, MUAIC (2,50) are in a common block, MUAICS for output from GETAIC. NWWAIC, NTTAIC, NRWTAAIC, NLWTAIC and PAIC (4,50) are in a common block /PAICS/, for use by the routine.

CALL GETAIC (JUCENT, ITYPE, ICODE, IR)

Input Parameters:

JUCENT = receiving chord number
ITYPE = 1, 2, 3, 4 indicating wing-wing, tail-tail, right-wing-tail, or left-wing-tail AIC's desired
ICODE = 0, C,V,W desired
1, V,W desired
2, V desired

Common Input:

NWWK = Number of AIC arrays available
NTTK = wing-wing
NRWTK = tail-tail
NLWTK = right wing-tail
left wing-tail

PAIC (4,50) = Table of contents for the AIC's.
PAIC (I,J) indicates where the AIC's for the Ith form of influence (see above) on the Jth chord are located.

Output Parameters:

IR = D, Success
1, C not found
2, C and W not found
3, Nothing found

Common Output:

C = C
W = W
V = V
Fortran Subroutine VELPOT

AUTHOR: G. D. Kramer

PURPOSE: To compute normal wash and velocity potential values for one mode shape.

METHOD: This routine calculates the following equations:

\[
\frac{dP_j^{n,m}}{dt} = \left[ i \varphi_i f_j^{n,m} + b_i \frac{\partial f_j^{n,m}}{\partial x} \right]
\]  

(8)

(1) For the wing:

\[
N_{n,m}^{n,m} = \frac{dF_L}{dt} - \hat{N}_{n,m}^{n,m}, \quad N_{n,m}^{n,m} = -\frac{dF_L}{dt} - \hat{N}_{n,m}^{n,m}
\]  

(9)

where

\[
\hat{N}_{n,m}^{n,m} = \sum_{\text{left wing} + \text{diaphragm}} \left[ \cos 2\psi_w \frac{W_{DLX}^{(RLW)}}{D_{DLX}^{(RLW)}} - \sin 2\psi_w \frac{V_{DLX}^{(RLW)}}{D_{DLX}^{(RLW)}} \right] N_{n,m}^{n,m} \times \text{SYM}
\]  

(10)

and

\[
\hat{N}_{n,m}^{n,m} = -\sum_{\text{left wing} + \text{diaphragm}} \left[ \cos 2\psi_w \frac{W_{DLX}^{(RLW)}}{D_{DLX}^{(RLW)}} - \sin 2\psi_w \frac{V_{DLX}^{(RLW)}}{D_{DLX}^{(RLW)}} \right] N_{n,m}^{n,m} \times \text{SYM}
\]  

(11)

and SYM = \{ +1.0, symmetric \}

\{ -1.0, antisymmetric \}

(2) For leading edge or tip diaphragm boxes:

\[
(N_{n,s}^{n,m} - N_{n,L}^{n,m}) = \frac{1}{C_{ao}} \left[ -\sum_{\text{right wing} + \text{diaphragm}} C_{D_{DLX}} (N_{n,s}^{n,m} - N_{n,L}^{n,m}) \right]
\]

\[
+ \text{SYM} \sum_{\text{left wing} + \text{diaphragm}} C_{D_{DLX}} (N_{n,s}^{n,m} - N_{n,L}^{n,m})
\]  

(12)
(3) For wake diaphragm boxes:

\[
(N_{RUS}^{n,m} - N_{RLS}^{n,m}) = \frac{1}{C_{000}} \left[ \Delta \phi^m_j - \sum_{\text{right wing + diaphragm}} C_{j,k}^m (N_{RUS}^{n,m} - N_{RLS}^{n,m})^{\text{sym}} \right.
\]

\[
+ \text{SYM} \sum_{\text{left wing + diaphragm}} C_{j,k}^m (N_{RUS}^{n,m} - N_{RLS}^{n,m}) \right]
\]

where

\[
\Delta \phi^m_j = \Delta \phi^m_{\text{tie}} e^{-i k_j (x_{j} - x_{j,m})} \]

\[
(\hat{N}_{RUS}^{n,m} + \hat{N}_{RLS}^{n,m}) = - (\hat{N}_{RUS}^{n,m} + \hat{N}_{RLS}^{n,m}) \]

(4) For the tail:

\[
(N_{RUT}^{n,m} - N_{RLT}^{n,m})^{\text{sym}} = 2 \left( \frac{dN_{RUT}^{n,m}}{dt} + (\hat{N}_{RUT}^{n,m} - \hat{N}_{RLT}^{n,m}) \right)
\]

\[
- 2 \hat{N}_{RUT}^{n,m} - 2 \hat{N}_{RLT}^{n,m} \]

where \(\hat{N}_{RUT}^{n,m}\) and \(\hat{N}_{RLT}^{n,m}\) are computed as in Equation (11) and (12).

\[
\hat{N}_{RUT}^{n,m} \sum_{\text{wing + diaph.}} \left[ \cos (\psi_T - \psi_W) W_{\text{RW}}^{(R^T)} - \sin (\psi_T - \psi_W) V_{\text{RW}}^{(R^T)} \right] N_{RW}^{n,m}
\]

\[
(17)
\]

\[
\hat{N}_{RUT}^{n,m} = \text{SYM} \sum_{\text{left wing + diaph.}} \left[ \cos (\psi_T + \psi_W) W_{\text{LW}}^{(R^T)} - \sin (\psi_T + \psi_W) V_{\text{LW}}^{(R^T)} \right] N_{LW}^{n,m}
\]

\[
(18)
\]
Velocity potentials:

\[ \Delta \bar{\phi}_j^{nm} = \sum_{\text{right surface and diaphragm}} C_{\mu \nu 0} (N_{RUS}^{\nu \mu} - N_{RLS}^{\nu \mu}) + \sum_{\text{left surface and diaphragm}} C_{\mu \nu 1} (N_{RUS}^{\nu \mu} - N_{RLS}^{\nu \mu})_{\text{SYM}} \]

Because the equations involve summations over unknown values, the order of calculation is very critical. The routine computes normal washes and velocity potentials in parallel, one row at a time, inboard-most box first. If the subdivision option is on, each subdivided box must have a set of normal washes computed as well, using equations similar to those above.

For each box, the \( \hat{N} \) terms are first zeroed out. If spatial contribution is present, subroutine GETAIC is called to get the necessary AIC arrays from scratch file IAICSC, and the proper summation is computed over the forward Mach hyperbola. This is first done for \( \hat{N}_{RUS} \) and \( \hat{N}_{RLS} \) and stored in variables \( \text{ENRULU} \), \( \text{ENRLLL} \).

If the surface is a tail, a similar procedure of getting AIC arrays and computing the proper summation for \( \hat{N}_{RUT} \) and \( \hat{N}_{RUT} \) is followed. The results are stored in \( \text{ENRURW} \) and \( \text{ENRULW} \).

If the box being considered is a planform box, the normal wash values are next computed from Equations (9) or (16).

Function B is called to compute the planar \( \Delta \bar{\phi} \) contribution, except for the contribution of the box to itself. This is stored in variable \( \text{DELPH} \). If the box is on planform, the out-of-plane contribution is added, yielding Equation (19). If the box is on a diaphragm, \( \text{DELPH} \) is used in Equation (12) or (13) to eventually yield the normal wash values at the diaphragm box center.

Trailing edge velocity potentials, array TVP, are computed whenever a trailing edge box is encountered. The computation is normally linear extrapolation from the last two box center values. In the event there is only one box on the tip chord, a Mach ray extrapolation is first done, followed by chord-wise linear interpolation. See Figure 11.
Values at A and B are extrapolated to C. Then the values at C and D are interpolated to give a value at E, the desired trailing edge value.

FIGURE 11  Tip Chord Trailing Edge Velocity Potential Calculation

The subdivision option causes the following:

1. All row and column loops are on subdivided boxes.

2. Any necessary $\hat{N}$ terms are calculated once per control point, and stored in temporary arrays for use on all subdivided boxes within the unsubdivided box. $\hat{N}$ terms and spatial contribution of left surface to $A\bar{\Phi}$ are not calculated using subdivided values.

3. Function $B$ and $A\bar{\Phi}$ are not computed for on-planform subdivided boxes which do not contain a control point.

4. Function $B$, when called, applies two equations - one within the "effective area" of subdivision, and the other outside this area. It is within function $B$ that the subdivision refinement actually takes place.

5. Any unsubdivided box which has one or more off-planform subdivided boxes has its normal wash values computed as the average of all subdivided values within its bounds, i.e.

$$N_{RUS}^{n,m} = \left( \sum_{\text{all subdivided}} N_{RUS}^{s} \right) / NSUBDV$$
The subroutine is called by:

CALL VELPOT(IBOX, LBOXCD, PKERNL, SKERNL, WING, DIHS)

Input Parameters:

IBOX   Array of box codes for the surface.
LBOXCD Length of the box code array.
PKERNL Primary (unsubdivided) \( C_{0}^{0} \) array.
SKERNL Subdivided \( C_{0}^{0} \) array.
WING   .TRUE., the surface is a wing.
        .FALSE., the surface is a tail.
DIHS   .TRUE., any surface dihedral is to be accounted for.
        .FALSE., any surface dihedral may be ignored.

Input Common Variables:

Global common blocks used:

/"XCOMTY/  
"GEOM2/  
"MODES/  
"FILES/  
"CHECKPR/  
Blank Common for C

Local common values:

/MUAICS/YBAR Parallel offset
    EL Normal offset
    MUAIC(2,50) AIC pointer array determined in the geometry section.
    NROWS Number of rows defined for the AIC set.
/AICS/ XXVL Current value of \( K_{1} \)
    C C
    W W
    V V
/DELTAP/ \{TEXLOC\} \{leading\} edge X-locations at chord centers
        \{trailing\}
        IPNTRM Pointer array for modes and velocity potentials.
        DEFSL Mode shape array - equivalenced to velocity potential array.
        IOVLAP Measure of tail overlap of wing, box mode shapes.
/NWASHES/ IPNTDW Pointer array for normal wash values.
/BXCODES/ IBOXW Wing box codes
Output Common Variables:

/DELTAP/  DELPHI  $\Delta \bar{\phi}$ array  
           TVP  $\Delta \bar{\phi}_{TE}$ array

/NWASHES/  ENRUS  $N_{RUW}$ or $N_{RUT}$  
            ENRLS  $N_{RLW}$ or $N_{RLT}$  
            IOVLAPW  Measure of tail overlap of wing diaphragm,  
                      for normal washes.

/SNWASH/  IPNTSD  Pointer array for subdivided normal washes  
             ENSUBD  $N_{RUSS}$ and $N_{RLS}$  
             IPNTIN  
             IPNTOT  \{ End-around pointers for array IPNTSD \}  
             IPNTLS  

52
Fortran Function B

AUTHOR: G. D. Kramer

PURPOSE: Given the location of a Mach box, to compute the planar contribution of the rest of the surface to the velocity potential difference for the box.

METHOD: The routine has two sections, one for computing the subdivided contribution within the "effective area," and the other to compute the unsubdivided contribution from ahead of the "effective area". If the subdivision option is off, the second section is used for the full contribution.

In the first section, the summation performed is

$$\mathcal{B}_s = \sum_{\nu} \sum_{\mu} \frac{k_1/n_s}{\sqrt{\nu \mu}} \left( N_{RUS}^{(s)} - N_{RLS}^{(s)} \right)$$

where the summation limits are as shown in Figure 12.

FIGURE 12 Subdivided "Effective Area"

Always contributes to the summation
Contributes if the dihedral angle $\gamma = 0$
The second section starts up where the first leaves off, and computes the second summation in Equation (22)

\[ B = B_s + \sum \sum C_{\nu\mu} \left( N_{\text{KUS}} - N_{\text{RLS}} \right) \]  

where the summation limits are as shown in Figure 13 and the AIC array and normal wash values are now unsubdivided, computed at control points.

FIGURE 13. Unsubdivided Boxes Outside the "Effective Area"
USAGE: The function is called by:
DELPH = B(IROW, JCOL, PKERNL, SKERNL, IBOX, LBXCD, WING, DIH)

Input Parameters:
IROW, JCOL Location of receiving point, subdivided
PKERNL Primary (unsubdivided) \( \nu \mu \sigma \) array
SKERNL Subdivided \( \nu \mu \sigma \) array
IBOX Box code array
LBXCD Length of box/code array
WING .TRUE., the surface is the wing
      .FALSE., the surface is the tail
DIH .TRUE., left side is to be ignored
      .FALSE., Include left side.

Input Common Parameters:
See subroutine VELPOT. Both subdivided and unsubdivided values are used.

Output:
The function value, \( B \), in this case stored in DELPH, is the result of the summations described under METHOD.
Fortran Subroutine SMPLW

AUTHOR: G. D. Kramer

PURPOSE: To compute and print upwashes, sideward and longitudinal washes at arbitrary chord locations in the wake of a wing.

METHOD: This routine is called once for each sampling chord. For each box on the chord, the right wing contribution is summed as

\[
WSUM = \sum_{rt \text{ wing}} W_{Pr} \alpha \times N_{RW}^{\nu_\lambda}
\]

(23)

\[
VSUM = \sum_{rt \text{ wing}} V_{Pr} \alpha \times N_{RW}^{\nu_\mu}
\]

(24)

\[
PHISUM = \sum_{rt \text{ wing}} C_{Pr} \alpha \times N_{RW}^{\nu_\mu}
\]

(25)

where \(N_{RW}^{\nu_\lambda} = \begin{cases} N_{RUW} & \text{if the chord is above the wing,} \\ N_{RLW} & \text{if the chord is below the wing.} \end{cases} \)

These sums are then combined as:

\[
UW_R = \frac{U}{U} = \left(\frac{1}{b_1}\right) \times \left(\cos \Psi_w \times WSUM + \sin \Psi_w \times VSUM\right)
\]

(26)

\[
SW_R = \frac{V}{U} = \left(\frac{1}{b_1}\right) \times \left(\cos \Psi_w \times VSUM - \sin \Psi_w \times WSUM\right)
\]

(27)

\[
PHI_R = PHISUM
\]

The left wing contributing summations are identical to Equations (23), (24), and (25), with \(N_{RW}^{\nu_\lambda}\) replaced by \(N_{LW}^{\nu_\lambda}\).

The results are then combined by

\[
Uw_{\text{complete}} = UW_R + \left(\frac{1}{b_1}\right) \times \left(\cos \Psi_w \times WSUM - \sin \Psi_w \times VSUM\right) \times SYM
\]

\[
SW_{\text{complete}} = SW_R + \left(\frac{1}{b_1}\right) \times \left(\cos \Psi_w \times VSUM - \sin \Psi_w \times WSUM\right) \times SYM
\]

56
\[ \Phi_{\text{complete}} = \Phi_R + \Phi_{\text{SUM}} \times \text{SYM} \]  

(30)

The printed upwash and sidewash is given by Equations (28) and (29). For longitudinal wash, the \( \Phi \) values computed in Equation (30) are used in

\[ LW(I) = \frac{1}{2} \left[ \Phi(I+1) - \Phi(I-1) \right] \]  

(31)

**USAGE:**

The routine is called by:

```
CALL SMPLW(IBOX,LBXCD,JCHRD, JT, IFRST, ILAST)
```

**Input Parameters:**
- **IBOX** Array of wing box codes
- **LBXCD** Length of box code array
- **JCHRD** Sample wash chord number, with reference to the order specified in the card data
- **JT** The y-location of the chord
- **IFRST** Number of the first sample box desired
- **ILAST** Number of the last sample box desired
9. VELOCITY POTENTIAL SMOOTHING SECTIONS

FORTRAN Program SMTH

Author: G. E. Keylon

Purpose: To smooth the velocity potentials by using a least squares surface fitting technique.

Method: The velocity potentials are read in from a disk file and smoothed with a least squares fit by subroutine FITTER, previously described. The polynomial equation derived from the fit is then used to compute an array of velocity potentials at planform box centers.

Usage: The SMTH program is the main program of a secondary overlay of the Mach box program. It is called as follows:

CALL OVERLAY (6HAFMBOX, 1, 6, 0)

Input:

USES LABELLED COMMON BLOCKS
/ARRAYS/
/FILES/
/IOCONT/
/PROBLM/
/KVAL/
/GEOMETRY/
/GEOM2/
/TAPEIO/
/RWBUFF/

Uses the following files
MODESC, IGEOSC, IVPSC

Output:

Output is stored on file IWTFSC which is changed to IVPSC.
FORTRAN Program CRDFIT

Author: G. E. Keylon

Purpose: To smooth the velocity potentials by using a least squares curve fit along each chord.

Method: The velocity potentials are read in from a disk file. The values for each chord are then separated into an array. The values are then changed to the numerical slope between the midpoint average values. Subroutine CURVE is then called to fit a least squares polynomial curve though these slopes. The polynomial equation is then integrated at each box on the chord and the integral value becomes the velocity potential at that box.

Usage: The CRDFIT program is the main program of a secondary overlay of the Mach box program. It is called as follows:

CALL OVERLAY (6HAFMBOX, 1, 7, 0)

Input:

USES LABELED COMMON BLOCKS

/ARRAYS/
/FILES /
/IIOCONT/
/PROBLM/
/KVAL /
/GEOMTY/
/GEOM2 /
/TAPEIO/
/RNBUFF/

Uses the following files
MODESC, IGEOSC, IVFSC

Output:

Output is stored on file IWTFSC which is changed to IVFSC.
FORTRAN Subprogram CURVE

Author: G. E. Keylon

Purpose: To fit a curve in the least squares sense through a set of data points.

Method: The CURVE routine is passed a set of ordered complex pairs and the degree of polynomial to fit. The system of simultaneous linear equations is solved employing the Choleski square root method (see Ref. 1). If the polynomial degree exceeds the limits possible to fit the degree is reduced to a lower level.

Usage: CALL CURVE (M,N,X,Z,C)

Input:

M - degree of polynomial equation
N - number of data points to fit curve through
X - Array of X coordinates (independent variable)
Z - Array of Z coordinates (dependent variable, complex)

Output:

C - output polynomial coefficient array, complex
10. GENERALIZED AIR FORCES SECTION

Fortran Program FORCES

AUTHOR: G. E. Keylon, G. D. Kramer

PURPOSE: To calculate the boxlifts, section lifts, and generalized air forces for a problem.

METHOD: Planform information is first read from the geometry and modes scratch files. The outer-most loop on thickness slope functions is then entered. One set of thickness slope functions, defined at box centers by Equation (1), is read in from scratch file ITSLSC. Next a loop on mode shapes, used as weighting functions for the generalized forces calculations, is entered. One mode shape is read from scratch file MODESC.

The third loop entered is on velocity potentials. The $\bar{\phi}$ array is read into DELPHI and $\bar{\psi}$ into array TVP from scratch file IVPSC. The box pattern for each surface is then passed over, one row at a time. For each box the following values are computed:

$$ L_j^{nm} = \text{BXLIFT(IDC)} = \frac{2}{\beta} \left[ \Delta \bar{\phi}_j^{nm} - \Delta \bar{\psi}_j^{nm} + i \alpha^{nm} k_1 \Delta \bar{\phi}_j^{nm} \right] \bar{\psi}_j^{nm} $$

$$ \Delta C_{rj} = \text{DELCP(IDC)} = \bar{L}_j^{nm} / (\alpha^{nm} + b_1) $$

$$ \bar{Q}_{ij}^{nm} = \left[ \left( f_{TE}^{nm} + \Delta \bar{\phi}_{TE}^{nm} - f_{LE}^{nm} * \Delta \bar{\phi}_{LE}^{nm} \right) - b_1 \alpha^{nm} \frac{\partial}{\partial x} \Delta \bar{\phi}_{j}^{nm} \right] \bar{\psi}_j^{nm} $$

The $Q_{ij}^{nm}$ terms are summed as calculated, and stored as

$$ AFROW(JVP) = \sum_n \sum_m Q_{ij}^{nm} / \beta_2 $$

After all boxes have been processed, if boxlifts and section lifts are desired and this is the first mode shape, box lifts are printed, section lifts are computed and printed, and total lift is printed:

$$ L_j^m = \text{SLIFT(JCOL)} = \sum_n \bar{L}_j^{nm} $$

$$ \bar{C}_j = \text{TLLIFT = } \sum_m \bar{C}_j^{m} $$
After this has been done for all velocity potentials, one row of the final generalized air forces arrays is computed as:

\[
\begin{align*}
\overline{Q}_{ij} & = \text{GENAF}(ij) = \frac{2}{\beta} \text{AFROW}(jv \ p) \\
Q'_{ij} & = -\frac{b_i}{s_i^2} \beta \text{Re} \{\overline{\phi}_{ij}\} \\
Q''_{ij} & = -\frac{b_i^2}{s_i^2 k_1} \text{Im} \{\overline{\phi}_{ij}\}
\end{align*}
\]

(38)

(39)

(40)

The program does the above for all mode shapes, prints the results, optionally writes them on tape, then terminates. Printing is done in routines PRNTBL, PRNTSL, and PRNTAF.

For Equations (32) and (34) box leading and trailing edge values are needed. Several geometric conditions exist:

1. Box leading or trailing edge is internal to the planform:
   Linear interpolation is used,
   \[
   f_{\text{LE}}^{nm} = \frac{1}{2} \left( f_{n-1, m}^{n-1, m} + f_{n, m}^{n, m} \right)
   \]
   (41)

   \[
   \Delta \overline{\phi}_{\text{LE}}^{nm} = \frac{1}{2} \left( \Delta \overline{\phi}_{n-1, m}^{n-1, m} + \Delta \overline{\phi}_{n, m}^{n, m} \right)
   \]
   (42)

   and similarly for the box trailing edge.

2. Box is cut by the planform leading edge:
   \[
   f_{\text{LE}}^{nm} = f_{\text{LE}}^{n, m} - (X_n - X_{\text{LE}}) \frac{\partial f_{\text{LE}}^{nm}}{\partial X}
   \]
   \[\text{point-slope}\] (43)

   \[
   \Delta \overline{\phi}_{\text{LE}}^{nm} = \begin{cases} 
   0 & \text{for wing or spatial tail} \\
   \Delta \overline{\phi}_{\text{TE}}^{n} w_{\text{wing}}^{nm} e^{-i \left( X_{\text{LE}, \text{tail}}^{m} - X_{\text{TE}, \text{wing}}^{m} \right) k_1 / b_i} & \text{other} 
   \end{cases}
   \]
   (44)
Box is cut by the planform trailing edge

\[ \int_{TE}^{nm} \Phi = \int_{nm}^{nm} + (X_{TE} - X_n) \cdot \frac{\partial \int_{nm}^{nm} \Phi}{\partial X} \]  

\[ \Delta \Phi_{TE} \]

Computed planform trailing edge value, TVP, as described under subroutine VELPOT. This normally is a linear extrapolation using the forward adjacent box center and the current one for the two necessary \( \Phi \) values.

**USAGE:**

The FORCES program is the main program of a secondary level overlay of the Mach Box program. It is called as follows:

```plaintext
CALL OVERLAY(6HAFMBOX, 1, 7, 0)
```

**Input:**

Uses the following files:

- MODESC
- IPNTRM
- IVPSC
- ITGLSC

**Output:**

Printer and tape NOUTP (optional).
11. COMMON BLOCK ORGANIZATION

The basic geometric and program control parameters are stored in a set of labeled common blocks which are loaded with the primary level overlay and thus are available to any secondary overlay. Some of the variables come directly from card input values (see Part I, Section III of this report), and others are internally computed.

/CONTRL/ PREVEX, OMACH, TITLE(8), PRVGEOM, PRVMODE, DIHW, DIHT, DEFAULT
PREVEX Tested for code word in the data preprocessor --
link to determine whether defaults should be set or
prior status maintained (recycle)
OMACH Mach # from previous cycle, compared on recycle
0.
to determine whether planform geometry needs changing.
TITLE(8) One-line title for all printed headings blank
PRVGEOM .T. previous geometry is to be used this cycle .F.
New geometry is to be read
PRVMODE .T. previous modes are to be used this cycle .F.
new modes are to be read
DIHW .T. Tail dihedral is to be used computing
.wing influence on itself .F.
DIHT .T. Wing dihedral is to be used computing
.tail influence on itself .F.
.DEFAULT .T. All parameters on Card C are to be set
.to their default values .F.
.F. Do not set parameters to default.

/PROBLM/ XMACH, NMODES, NTSLOP, NKVALS, SMOOTH, NDEG, CRDFIT, EXAIC, SUBDV, PLYWOOD
XMACH = Mach number for current cycle no default
NMODES = Number of input modes to use no default
NTSLOP = Number of thickness slope functions to be used 0
NKVALS = Number of reduced frequencies to be used 0
SMOOTH = .T., Velocity potentials surface smoothing desired .F.
No velocity potential surface smoothing desired .F.
NDEG = Maximum order for smoothing polynomial 0, program will
determine
CRDFIT = .T., Chordwise velocity potential smoothing desired .F.
.No chordwise smoothing desired .F.
EXAIC = .T., Integration accuracy of 10^-4 desired .F.
Integration accuracy of 10^-2 desired .F.
SUBDV = .T., Subdivision is to be applied .F.
No subdivision is desired .F.
PLYWOOD = .T., Full box areas to be used in box lifts .F., Planform box areas to be used.
Default

/GEOMETRY/ COPLAN, NSUBDV, XSUBDV, NSUBD2, NSUBCN, NSURF,
       BI, B1BETA, B1S, B1ETAS, WLAX, WLAZ, PSIW, MXBW, MXBBW,
       MYBW, MYBBW, MXBSW, MYBBSW, IXBW, XCENTR

/GEOMETRY/ TLAX, TLAZ, PSIT, MXBT, MYBT, MYBBT, MXBST,
            MYBST, MYBBT, IXBT, IXBST, CAPL

COPLAN .T. PSIW = PSIT and CAPL = 0. One box array
       is used .F. The 2 surfaces are not coplanar, or only
       one surface is defined
NSUBDV Number of subdivided rows (columns) per box 1
XSUBDV = Float (NSUBDV) 1.0
NSUBD2 = NSUBDV/2 0
NSUBCN = NSUBD2 + 1 = center location of first chord 1
NSURF Number of surfaces, 1 or 2 1
BI Box length = BI* /8.
B1BETA BI/8, box width, = YWLE(NWLE)/MYBW
B1S Subdivided box length = BI/XSUBDV
B1ETAS Subdivided box width = B1BETA/XSUBDV
WLAX {Wing} local axis location, in global X co-
       ordinate 0.
TLAX {Tail} 0.
WLAX {Wing} local axis location, in global Z co-
       ordinate 0.
TLAX {Tail} 0.
PSIW {Wing} dihedral angle, input in degrees but
       immediately changed to radians. 0.
PSIT {Tail} 0.
MXBW Number of rows to aftmost portion of the{wing}
       measured in the n_c coordinate --
MXBT {Tail} --
MXBBW Number of rows to aftmost wing diaphragm
       box, n_c coordinate --
MYBW {Wing} number of chords on the{wing}
       , m_c coordinate {=NCHRDS} --
MYBT {Tail} --
MYBBW Number of {wing} chords, including tip
       diaphragm --
MYBBSW Subdivided MXB {W} count --
MXBST --
MYBSW}  Subdivided MYB \{W\}_T count  --
MYBST  Subdivided MYB \{T\}_T count  --
MYBBSW  Subdivided MYBB \{W\}_T count  --
MYBBST  Subdivided MYBB \{T\}_T count  --
IXBW}  Subdivided grid X-location of the first
        unsubdivided \{wing\} box center  --
IXBT}  Subdivided grid X-location of the first
        subdivided \{tail\} box  --
IXBST  Subdivided grid X-location of the first
        subdivided tail box  --
XCENTR  X location of the center of the first box
        on the wing  No default
        X location of the first box centerline on the wing
CAPL  Non-dimensionalized vertical distance between
        centerlines of the wing and tail  0.

/KERN/ ERR, MXSKRN, IPKERN, NPLKRN, NSPATK, NR\O\WEA
ERR  Integration accuracy in AIC calculations  .01
MXSKRN  Size of the subdivided AIC, array (number of rows)
IPKERN  Location in array SKERNL where PKERNL(l)
        would be if it were not overlaid by the
        subdivided \C\upho\ array.
NPLKRN  Size of the planar AIC array (number of rows)  --
NSPATK  Number of spatial AIC arrays necessary  0
NR\O\WEA  Number of rows for the subdivided effective area  --

/KVAL/ IKVAL, XKVAL(20), XKS (20)
IKVAL  Current k-value number being solved
XKVAL  Array of reduced frequencies, \(k_1\), based on
        box length, \(b_1\)
XKS  Array of reduced frequencies, \(k_s\), based on
        semispan, \(s\).

/FILES/ NT5, NT6, INTAPE, INFSP, NPLAIC, NSPAIC, NOUTP,
IOJFSP, MODESC, IVPSC, IGEOSC, IWTFS, IAICSC
NT5  Card file (INPUT)
NT6  Print file (OUTPUT)
INTAPE  Binary input tape number, If 0 or 5 card
        input will be used
INFSP  Initial file spacing on the input tape  0
NPLAIC Tape number for the \{planar\} AIC arrays  0
NSPAIC  0
NOUTP  Binary output tape number. If 0, none written  0
IOUFSP Initial file spacing on tape NOUTP  0

MODESC, IPSC  \{planar\} AIC arrays
IGEOSC, IWTSFC \{spatial\} AIC arrays
IAICSC

/IIOCONT/  OPLAIC, OSPAIC,
WTGEOM, WTAGAF, WTSL, WTBL, PRBOX, PRPAIC,
PRSAIC, PRMODS, PROOF, PRUW, PRSW, PRVP, PRBL,
PRDCP, PRGNAF, PRGNAC, PRSL, PRLW, PRNW

OPLAIC  .T., an old \{planar\} AIC tape is being used  .T.
OSPAIC  .F., a new \{spatial\} AIC tape is being used  .F.

WTGEOM Not used  .F.

WTAGAF  .T., Write generalized air forces on tape  .T.
WTSL .T., Write section lifts on tape  .F.
WTBL .T., Write box lifts on tape  .F.
PRPAIC .T., Print the box code pattern(s)  .F.
PRSAIC .T., Print the \{planar\} AIC arrays  .F.
PRMODS .T., Print modal deflections and slopes  .F.
PROOF .T., Print modal polynomial coefficients, if available  .F.
PRUW .T., for wake wash sampling, print \{upwashes side washes \}
PRSW .T., for wake wash sampling, print \{longitudinal washes \}
PRLW .F.

PRVP .T., Print velocity potential differences  .F.
PRBL .T., Print box lifts, \(E_j^{n,m}\)  .F.
PRDCP .T., Print change in pressure, \(\Delta C_j^{n,m}\)  .F.
PRGNAF .T., Print generalized airforces, \(\bar{Q}_{j,j}\)  .T.
PRGNAC .T., Print generalized aerodynamic coefficients, \(Q'\) and \(Q^*\)  .F.
PRCM .T. Print sectional generalized airforces, \(\bar{q}_j^{n,m}\)  .F.
Default

PRSL .T., Print section lifts, $L_j^m$ .F.

PRNW .T., Print normal washes, $H_{RUW}, N_{RUW}$ etc. .F.

/TAPEIO/ NFS, NMS, LS, NMR, ID(20), NID, ITYPE, LWS, LWS, M, N, PARM(10), IRR

DIMENSION IPARM(10)

EQUIVALENCE (PARM, IPARM)

NFS }
NMS }
File 
Matrix 
spacing 
0
0

LS }
NMR }
Not used

ID }
ID array for the matrix 
NID }
Number of words in the ID array on tape 

ITYPE }
Matrix type - MIXED, COMPLEX

LS }
LWS }
Not used

M }
N }
Matrix dimensions 
--
--

PARM }
Numerical parameters for the matrix 
--

IRR }
Error return 
-- 0--

/MODES/ SYM, SYMT, MTYPEW, MTYPEP

SYM }
1, Symmetric modes 
-1, Antisymmetric modes 
0, Left surface contribution will be ignored

SYMT }
As above, for a non-planar tail. Differs only for vertical tail

MTYPEW }
1, Polynomial coefficients will be read for the wing
2, Deflections at arbitrary locations will be read
3, Box center values will be read

MTYPEP }
1, Same as above for the tail
2, 
3, 

/ARRAYS/ KBXCDW, LBXCDW, LBOXC, KBXCDT, LBXCDT, KJALPH, LJALPH, KALPHA, KKERNL, LKERNL, KPNTRM, LPNTRM, KDEFSL, KELPHI, LMODES, KPNTS, LPNTS, KSDW, LSDW, KPNTDW, LPNTDW, KDW, LDW, KTV, LTVP

68
Locations and limits for arrays:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Array affected</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>KBXCDW</td>
<td>IBOX (LBXCD, LBOXC)</td>
<td></td>
</tr>
<tr>
<td>LBXCDW</td>
<td></td>
<td>Not used</td>
</tr>
<tr>
<td>LBOXC</td>
<td></td>
<td>150</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>KBXCDT</td>
<td>IBOXT (LBXCDT, LBOXC)</td>
<td></td>
</tr>
<tr>
<td>LBXCDT</td>
<td></td>
<td>Not used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>90</td>
</tr>
<tr>
<td>KJALPH</td>
<td>IJALPH (IJALPH)</td>
<td></td>
</tr>
<tr>
<td>LJALPH</td>
<td></td>
<td>Not used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>200</td>
</tr>
<tr>
<td>KALPHA</td>
<td>ALPHA (IJALPH)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Not used</td>
</tr>
<tr>
<td>KKERNL</td>
<td>SKERNL (LLKERNL), PKERNL</td>
<td></td>
</tr>
<tr>
<td>LKERNL</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1640</td>
</tr>
<tr>
<td>KPNTRM</td>
<td>IPNTRM (2, LPNTRM)</td>
<td></td>
</tr>
<tr>
<td>LPNTRM</td>
<td></td>
<td>Not used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
</tr>
<tr>
<td>KDEFSL</td>
<td>DEFSL (2, LMODES)</td>
<td>Not used</td>
</tr>
<tr>
<td>KELPHI</td>
<td>DELPHI (LMODES), complex</td>
<td></td>
</tr>
<tr>
<td>LMODES</td>
<td></td>
<td>Not used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>500</td>
</tr>
<tr>
<td>KPNTSD</td>
<td>IPNTSD (2, LPNTSD)</td>
<td></td>
</tr>
<tr>
<td>LPNTSD</td>
<td></td>
<td>Not used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
</tr>
<tr>
<td>KSDW</td>
<td>ENSUBD (2, LSDW)</td>
<td></td>
</tr>
<tr>
<td>LSDW</td>
<td></td>
<td>Not used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>600</td>
</tr>
<tr>
<td>KPNTDW</td>
<td>IPNTDW (2, LPNTDW)</td>
<td></td>
</tr>
<tr>
<td>LPNTDW</td>
<td></td>
<td>Not used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
</tr>
<tr>
<td>KDW</td>
<td>ENRUS (LDW), ERNLS (LDW)</td>
<td></td>
</tr>
<tr>
<td>LDW</td>
<td></td>
<td>Not used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1275</td>
</tr>
<tr>
<td>KTVP</td>
<td>TEP (LTVP), TEZLOC (LTVP),</td>
<td></td>
</tr>
<tr>
<td>LTVP</td>
<td>EXLOC (LTVP)</td>
<td>Not used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>250</td>
</tr>
</tbody>
</table>
**/WBUFF/ BFCODE, IBFCNT, BUFF (3280)**

- **BFCODE** = Code word 8
- **IBFCNT** = Size of buffer 3280
- **BUFF** = Buffer array for use by READMX and WRTEMX

**/SAMPLW/ ISMPLW, ICHORD(10), IBOXF(10), IBOXL(10), ZLOC(10)**

- **ISMPLW** = Number of chords specified for wash sampling 0
- **ICHORD** = Chord number for sampling
- **IBOXF** = First box on chord to be sampled
- **IBOXL** = Last box on chord to be sampled
- **ZLOC** = Z-location of sampling chord, transformed internally to correspond to wing coordinates 0.

**/PLANXY/ NWLE, NWTE, NTLE, NTTE, XWLE(10), YWLE(10), XWTE(10), YWTE(10), XTLE(10), YTLE(10), XTTE(0), YTTE(0)**

- **NWLE** = Number of wing leading edge definition points
- **NWTE** = Number of wing trailing edge definition points
- **NTLE** = Number of tail leading edge definition points
- **NTTE** = Number of tail trailing edge definition points
- **XWLE** = Wing leading edge definition points
- **XWTE** = Wing trailing edge definition points
- **XTLE** = Tail leading edge definition points
- **XTTE** = Tail trailing edge definition points

**COMMON/ CHECKPR/ DPPCPR, GEOCPR, AICCPR, NWSCPR, SMCPR, GAFCPR**

These variables are all typed logical. They control whether or not internal checkout print statements will be executed. They will be read from Card C, default .FALSE.

- **DPPCPR** = Data preprocessor check-print
- **GEOCPR** = Geometry check-print
- **AICCPR** = AIC section check-print
- **NWSCPR** = Normal wash & velocity potential check-print
- **SMCPR** = Velocity potential smoothing check-print
- **GAFCPR** = Generalized Airforces check-print

70
12. ARRAY STORAGE

In order to conserve storage, a number of arrays are used as pointers for sparse arrays. All unusual array usage is described below.

a. Arrays Generated in the Geometry Section

IBOXW - Subdivided box pattern
for first planform, or both if "coplanar"

IBOXT - Subdivided box pattern
for 2nd planform, if non-"coplanar"

FIGURE 14. Box Code Arrays

The box code arrays are packed twenty numbers to a word, so IBOXW(1,1) contains codes for box (1,1) through box (1,20), IBOXW(2,1) contains codes for box (2,1) through box (2,20), etc.

FEXLOC(I) = The location of the leading edge at chord I, normalized to BIS with 1.0 corresponding to the center of the 1st (subdivided) row.

TEXLOC(I) = Same for trailing edge.

IWAKE(I) = Aftmost subdivided wing wake box needed by the tail.

FIGURE 15. Leading and Trailing Edge Arrays
For each AIC array needed:

\[
\begin{array}{c|c}
\text{NROWS} & \\
\hline
1 & \\
2 & \\
\end{array}
\]

\[\text{MUAIC}(1,J) = \text{first box needed in row } J\]
\[\text{MUAIC}(2,J) = \text{last box needed in row } J\]

**FIGURE 17. AIC Array Pointers**

The four KPT-- arrays indicate the location on scratch file LAIC of the desired AIC array set. For example, KPTT(3) is the AIC set number (4 matrices per set) of the AIC's for the influence of the left tail on right tail chord 3.
b. Arrays generated in the Modes Section

**IPNTRM** Pointer array for planform boxes on a row.

<table>
<thead>
<tr>
<th>J</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>N</th>
<th>N+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPNTRM(1,J)</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>10</td>
<td>31</td>
<td>89</td>
</tr>
<tr>
<td>IPNTRM(2,J)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

Normally the row number for which the pointer value is being computed. If there are 2 surfaces that are noncoplanar, the value of J representing the first row of the second planform is MYBW+IOVLAP. IOVLAP is the number of rows on the tail planform that have same x coordinates as rows on the wing planform. If there are no rows with this condition IOVLAP is zero.

**IPNTRM(1,J)** The sequential count + 1 of all boxes, planform or wake region, that are on or between the first and last planform box of all rows forward of the one J represents.

**IPNTRM(2,J)** The chord number of the first planform box on the row represented by J.

**FIGURE 18. Row Pointers**

c. Arrays Generated in the AIC Section

The \( C_{p,\alpha} \), \( W_{p,\alpha} \) and \( V_{p,\alpha} \) arrays are stored in a one dimensional matrix. For planar AIC's the \( W_{p,\alpha} \) and \( V_{p,\alpha} \) are not computed and the \( C_{p,\alpha} \) array is calculated for 1/2 of the Mach cone since it will be symmetrical. If subdivision is applied then 2 planar arrays are calculated with the subdivided array overlaying part of the unsubdivided array.

\[
U = \begin{array}{cccccccc}
0 & 1 & 2 & 3 & 4 & \ldots & N_{PLKRN} & N_{PLKRN} + 1 & \ldots & N_{PLKRN} \\
\end{array}
\]

\[
\bar{U} = \begin{array}{cccccccc}
0 & 0 & 1 & 0 & -1 & 2 & -3 & 0 & -1 & 2 & 3 & 4 & \ldots \\
\end{array}
\]

**FIGURE 19. Planar AIC**

73
For unsubdivided cases the array ends when $\tilde{y} = \text{NPLKRN}$. For subdivided cases when $\tilde{y} = \text{NPLKRN}$ the array contains the subdivided AIC calculated at $k = k_i/\text{NSUBDV}$. $\tilde{y}$ then is reduced to $\frac{\text{NPLKRN}}{\text{NSUBDV}} + 1$ and is allowed to increase again until it reaches NPLKRN or the number of rows to cover the planform.

Because of the possible condition where the receiving point of a planform may not be in alignment with boxes on other planforms the spatial AIC's must be calculated on both sides.

<table>
<thead>
<tr>
<th>$\tilde{y}$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>...</th>
<th>NROWS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

FIGURE 20. Spatial AIC's
13. INTERNAL SCRATCH FILES

a. Matrix Format

All arrays written on disk or tape files are formatted as variable sized matrices. Each matrix consists of two logical records of binary information, the first one being a 16-word matrix identification record, and the second containing the contents of the matrix. The reading/writing of these matrices is done by subroutines READMX and WRTEMX. In the following tape maps, each matrix is a separate box.

Matrix Identification Record - 16\textsubscript{10} words

<table>
<thead>
<tr>
<th>Word</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>One-word ID label, an integer or label</td>
</tr>
<tr>
<td>2</td>
<td>M, the number of rows in the matrix</td>
</tr>
<tr>
<td>3</td>
<td>N, the number of columns in the matrix</td>
</tr>
<tr>
<td>4</td>
<td>Not used, = 0</td>
</tr>
<tr>
<td>5</td>
<td>Not used, = 0</td>
</tr>
<tr>
<td>6</td>
<td>Number of words in matrix record</td>
</tr>
<tr>
<td>7</td>
<td>$K_l$</td>
</tr>
<tr>
<td>8</td>
<td>Mach Number</td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>User parameters, array PARM</td>
</tr>
<tr>
<td>11</td>
<td>to</td>
</tr>
<tr>
<td>16</td>
<td></td>
</tr>
</tbody>
</table>

Matrix Record - variable length (word 6 above)

Ordered consecutively by row, left to right within each row.
b. **Geometry Scratch File IGEOSC**

This file is generated in the geometry processor and contains all large geometry arrays. The space after the two geometry files is used for temporary scratch during mode shape processing.

<table>
<thead>
<tr>
<th>Matrix Dimensions</th>
<th>Parameter Array</th>
</tr>
</thead>
<tbody>
<tr>
<td>M = MXDBW*NSUBDV</td>
<td>PARM(1) = 0.</td>
</tr>
<tr>
<td>N = (MYBBSW-1)/NBWRD+1</td>
<td>PARM(2) = XMACH</td>
</tr>
<tr>
<td>M = MXEBST-IXBST+1</td>
<td></td>
</tr>
<tr>
<td>N = (MYBBST-1)/NBWRD+1</td>
<td></td>
</tr>
<tr>
<td>M = 1</td>
<td></td>
</tr>
<tr>
<td>N = MYBSW+MYBST</td>
<td></td>
</tr>
<tr>
<td>M = 1</td>
<td></td>
</tr>
<tr>
<td>N = MYBSW+MYBST</td>
<td></td>
</tr>
<tr>
<td>M = 1, 2, 3 or 4</td>
<td></td>
</tr>
<tr>
<td>N = max. # of AIC's needed in the 4 categories</td>
<td></td>
</tr>
<tr>
<td>M = 2</td>
<td>PARM(4) = YBAR</td>
</tr>
<tr>
<td>N = NROWS</td>
<td>PARM(5) = EL</td>
</tr>
<tr>
<td>M = 2</td>
<td>IPARM(6) = 0; C,W,V needed</td>
</tr>
<tr>
<td>N = NROWS</td>
<td>1; W,V needed</td>
</tr>
<tr>
<td>M = 2</td>
<td>2; V needed</td>
</tr>
</tbody>
</table>

(File 2 is first built on IVPSC by GEOMBX, then copied to IGEOSC.)
c. Modes Scratch File MODESC

This file is generated in the modal data processor. The deflections and slopes are given at all box centers.

Matrix Dimensions

<table>
<thead>
<tr>
<th>IPNTRM</th>
<th>M = 2</th>
</tr>
</thead>
</table>
| N = NPWTR S = \( \text{IPARM}(3) = 10VIA! \)
| DEFSL | M = 2 |
| Mode 1 | N = IPNTRM(1, NPWTR) - 1 |
| DEFSL | M = 2 |
| Mode 2 | N = IPNTRM(1, NPWTR) - 1 |

...
d. **Thickness Slopes Scratch File ITSLS**

This file is equvalenced to IWTPSC, which is first used in GEOMBX for temporary scratch while building the MUAIC arrays. The thickness slope functions are then written on the file at the end of the modal data processor. If \( NTSLOP = 0 \), one matrix of ones will be written, corresponding to \( \frac{\partial Z}{\partial x} = 0 \).

| TSLFN No. 1 | \( M = 1 \) |
| TSLFN No. 2 | \( N = IPNRM(1,NPNTS) \) |
| TSLFN No. \( NTSLOP \) | EOF |
### e. Spatial AIC Scratch File IAICSC

This file is first used for internal scratch during calculation of polynomial coefficients by the modal data processor. In the AIC section it is written with all spatial AIC's needed for one reduced frequency, as determined in the geometry section. IAICSC is re-written for each new reduced frequency.

<table>
<thead>
<tr>
<th>Matrix Dimensions</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$MUAIC_1$</td>
<td>$M = 2$</td>
</tr>
<tr>
<td>$N = NROWS_1$</td>
<td>$PARM(1) = k_1$, $PARM(2) = XMACH$, $PARM(4) = YBAR_1$, $PARM(5) = EL_1$</td>
</tr>
<tr>
<td>$C_{p_{max}}$</td>
<td>$M = 2$</td>
</tr>
<tr>
<td>$N = (NROWS_1+1)(NROWS_1/2)$</td>
<td></td>
</tr>
<tr>
<td>$W_{p_{max}}$</td>
<td></td>
</tr>
<tr>
<td>$V_{p_{max}}$</td>
<td></td>
</tr>
<tr>
<td>$MUAIC_2$</td>
<td>$M = 2$</td>
</tr>
<tr>
<td>$N = NROWS_2$</td>
<td>$PARM(4) = YBAR_2$, $PARM(5) = EL_2$</td>
</tr>
<tr>
<td>$C_{p_{max}}$</td>
<td>$M = 2$</td>
</tr>
<tr>
<td>$N = (NROWS_2+1)(NROWS_2/2)$</td>
<td></td>
</tr>
<tr>
<td>$W_{p_{max}}$</td>
<td></td>
</tr>
<tr>
<td>$V_{p_{max}}$</td>
<td></td>
</tr>
</tbody>
</table>

### EOF
f. Velocity Potentials Scratch File IVPSC

This file is first used for internal scratch by the geometry processor while assembling MUAIC arrays. It is later used in the modal data processor as temporary storage for the wing mode shapes to be merged with the tail modes, and again for the same purpose when working with thickness slope functions. In the normal wash and velocity potentials section it is written with the $\phi$ and $\omega_e$ arrays for each mode.

<table>
<thead>
<tr>
<th>Matrix Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M = 2$</td>
</tr>
<tr>
<td>$N = \text{IPNTRM}(1,\text{NPNTRS})-1$</td>
</tr>
</tbody>
</table>

| $M = 2$ |
| $\text{MYBSW if wing only}$ |
| $\{\text{MYBSW+HYBST otherwise}$ |

<table>
<thead>
<tr>
<th>$\text{DELPHI}_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{TVP}_1$</td>
</tr>
<tr>
<td>$\text{DELPHI}_2$</td>
</tr>
<tr>
<td>$\text{TVP}_2$</td>
</tr>
<tr>
<td>.</td>
</tr>
<tr>
<td>.</td>
</tr>
<tr>
<td>.</td>
</tr>
<tr>
<td>$\text{DELPHI}_{\text{NMODES}}$</td>
</tr>
<tr>
<td>$\text{TVP}_{\text{NMODES}}$</td>
</tr>
</tbody>
</table>
14. OUTPUT FILES

The program generates three optional output files. Two of them, the AIC files, are designed for reuse with the program during subsequent executions. The program automatically searches these files and updates them with any new AIC's generated.

The optional final output file is designed to pass the generalized air-forces matrices on for flutter or dynamic loads analyses. It is written optionally in the forces section of the program.
 FIGURE 1. Tape Storage of AIC Arrays
<table>
<thead>
<tr>
<th>Matrix ID</th>
<th>Contents</th>
<th>Function</th>
<th>Parameter Array</th>
</tr>
</thead>
<tbody>
<tr>
<td>100001</td>
<td>$\text{BL}_\text{Model}$ (2:MXB*MYB)</td>
<td>Box lifts for each mode, frequency 1 (optional)</td>
<td>$k_1$, $b_1$, Mach throughout</td>
</tr>
<tr>
<td>100002</td>
<td>$\text{BL}_\text{Mode2}$ (2:MXB*MYB)</td>
<td></td>
<td>Values for smoothed $\bar{\Phi}$ (optional)</td>
</tr>
<tr>
<td>1000MS</td>
<td>$\text{BL}_\text{ModeMS}$ (2:MYB*MYB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>$\bar{\Phi}$ (MS x 2.MS)</td>
<td>Generalized airforces, frequency 1</td>
<td></td>
</tr>
<tr>
<td>100001</td>
<td>$\text{BL}_\text{Model}$</td>
<td>Box lifts for each mode, frequency 1 (optional)</td>
<td>Values for unsmoothed $\bar{\Phi}$</td>
</tr>
<tr>
<td>1000MS</td>
<td>$\text{BL}_\text{ModeMS}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>$\bar{\Phi}$ (MS x 2.MS)</td>
<td>Generalized airforces, frequency 1</td>
<td></td>
</tr>
<tr>
<td>200001</td>
<td>$\text{BL}_\text{Model}$</td>
<td>Box Lifts frequency 2</td>
<td>Values for smoothed $\bar{\Phi}$ (optional)</td>
</tr>
<tr>
<td>2000MS</td>
<td>$\text{BL}_\text{ModeMS}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$\bar{\Phi}$ (MS x 2.MS)</td>
<td>Generalized airforces, frequency 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NFREQ</td>
<td>$\bar{\Phi}$ (MS x 2.MS)</td>
<td>Generalized airforces, frequency NFREQ, unsmoothed $\bar{\Phi}$</td>
<td></td>
</tr>
</tbody>
</table>

**FIGURE 22. TAPE MAP OF FORCES OUTPUT TAPE**
15. IMPLEMENTATION AND DEBUGGING

a. Update

The source program is written and maintained using the CDC 6600 SCOPE operating system UPDATE feature. Under UPDATE, all coding is either part of a *COMDECK or a *DECK. A *COMDECK may be replicated many times throughout the other decks. This feature is used for all global labeled common blocks and for most local common blocks, to insure that all routines needing them have identical common statements. A few subroutines which are needed in more than one overlay are also set up as *COMDECKs. The names of the *COMDECKs and *DECKs correspond as closely as possible to their Fortran identifiers - program name, subroutine name or common block name.

b. Open-ended Features

The writers of the program feel that most potential users probably have unique system features which may be utilized to optimize the execution of the program beyond its release status. With this in mind, numerous "hooks" have been coded in to make other features easy to implement.

1. All references to disk or tape files are by name, rather than by number. All file names are together in one common block, /FILES/. The internal scratch files are defined in one DATA statement in the zero overlay DRIVER, and the input, output and AIC files are defined via card input data.

2. All reading and writing of internal and external scratch files is handled by subroutines READMX and WRTEMX. These routines have several calling parameters which are unused, but available if it is desired to make use of labeling, random I.O., or level numbers. Because READMX and WRTEMX use BUFFERIN and BUFFEROUT, all files may share a common buffer area, allowing for a considerable savings in storage requirements.

3. Subroutine FLUSH is always called when a fatal error is encountered. This routine may be written to make use of any system error recovery procedure available. The release version prints a comment, flushes the OUTPUT file, and terminates with a Mode 1 error.

4. Subroutine DTIME is called between each secondary overlay. The release version returns CP time only; however, provision is made for PP time if the implementing system has that capability.

c. Debugging

It is recommended that a new user first run one of the sample data cases, to familiarize himself with the program features and to insure
that the program gives correct answers at his installation.

In the event the program fails "hard" (mode error, time limit, etc.), standard use of listings, load maps and dumps will usually pinpoint the cause.

If the program executes but seems to give bad numbers, additional intermediate printout may be helpful. The variables in common block /CHECKPR/ are designed to control the printing of additional check values. Each variable controls printing from one secondary overlay, so only the suspected area need be printed. The check printouts provided are rudimentary, so for given problems additional printouts would probably have to be written, but if they are made conditional on the common variables, they can be left in for future needs. The CHECKPR variables are all read from Card C of the data, or may be set in an executable statement after the call to DATAPP.
Program CONTROL — Primary level overlay which controls the program flow

ENTRY

Call Overlay (6HAPNOX,1,1), the Data Preprocessor Section.

Mach No. for this cycle = Mach no. for previous cycle, and previous geometry desired?

True

False

Call Overlay (6HAPNOX,1,2), the Geometry Calculation Section.

100

Call Overlay (6HAPNOX,1,3), the Modal Calculation Section

Loop on reduced frequencies
IKVAL = 1, NKVALS

Call Overlay (6HAPNOX,1,4), the AIC Calculation Section

Call Overlay (6HAPNOX,1,5), the Normal-wash and Velocity Potential Calculation Section

None Velocity Potential
Smoothing?

Surface Fit

Call Overlay (6HAPNOX,1,6), for Least Squares Surface Fit Smoothing

Chord Fit

Call Overlay (6HAPNOX,1,7) for Chordwise Least Squares Fit Smoothing

700

Call Overlay (6HAPNOX,1,8), the Generalized Forces Section

None Smoothing?

Yes

Call Overlay (6HAPNOX,1,8) again for Unsmoothed Generalized Forces

IKVAL
800

Recycle

Termination

EXIT

86

Call another Overlay
Program DATAPP — Secondary overlay which initiates control
parameters as a function of defaults or card data.

ENTRY

no

Recycle yes

Initialize control parameters to default values.

OMACH = Previous Mach no.

no

Initial Pass? yes

300

Read cards A, B, and C

Default ? F,

Read card D, set corresponding control parameters

Initialize reduced frequency arrays to -1.0

Read card E, determine NKVALS

Print heading title and all option values

Set array limits

Print disk file usage

Print reduced frequency arrays

RETURN

87
Program GINFOK — Secondary level overlay which computes planform geometry.

**ENTRY**

Read planform geometry information from cards F through K, converting to non-dimensional coordinate system.

Call BXCDFF to determine the (subdivided) box code pattern for the on-planform wing in array IBOXW

- yes — Coplanar?
  - no — Call BXCDFF to determine the (subdivided) box code pattern.
  - yes — Read planform geometry information from cards F through K, converting to non-dimensional coordinate system.

- Call BXCDFF to add the (subdivided) on-planform tail box codes to array IBOXW
  - no — Sampling? yes
    - Determine aftmost wing diaphragm needed for sampling.
    - Call BXCDFF to determine (subdivided) planform tail box codes in array IBOXT

- Call BXCDI to determine diaphragm box codes for the tail in array IBOXT
  - yes — Call BXCDI to determine the (subdivided) box codes for diaphragm areas in array IBOXW
    - Write box code array(s), FEXLOC and TEXLOC arrays onto IGEOSC
    - Call GMAREA to determine on-planform fractional area, G, of unsubdivided boxes cut by a planform edge.
      - no — Call PWAIIC for each chord on a surface with dihedral to compute the MUAIC array for left surface contribution to right
        - yes — Write table of contents array and MUAIC arrays onto IGEOSC
      - yes — spatial AIC's needed?
        - no — Tail? yes
          - Call PWAIIC to compute the MUAIC array for wing contributions to each tail chord.

RETURN
Program MOUES—Secondary level overlay which reads mode shapes and thickness slope functions

ENTRY

Read geometry data from IGEOC

Compute column pointer array

Call subroutine ROPER to compute row pointer array

Store row pointer array on scratch file

\( IPASS = 1, 2 \)

\( IPASS = 2 \)

\( IPASS = 1 \)

\( NS = 1, NSURF \)

\( NM = 1, NMODES \)

Read mode shapes for 1st planform if calculating 2nd planform

Polynomial Coefficients

Mode Input

Box Center Input

Read polynomial coefficients

Least Squares Fit

Read modal input at box centers and reorder for storage

Read modal data and call subroutine FITTER to fit polynomial equation

Compute deflections and slopes at box centers

Write the modal data on intermediate file if 1st of 2 surfaces, on final file if only 1 surface or 2nd surface

Read thickness slope values or create a function of 1.0 and store on the scratch file

RETURN

89
Program VICMAIN — Secondary level overlay which calculates all Aerodynamic Influence Coefficients

ENTRY

NV = 1, NVCS (# of AIC's)

Planar Planar Spatial or Spatial?

Set MUAIC pointer array Read MUAIC pointer array from geometry scratch file

Read AIC array from tape yes AICS on tape?

no

no Call Subroutine KERNEL (Determines limits of integration for each box Stores Bessel functions for later use.)

yes

Print AIC arrays

Subroutine RANGE (Give max. order of Bessel function)

Subroutine BESSEL (Computes the Bessel functions for a row of boxes.)

Subroutine ROMBER (Integrates the AIC functions by using Romberg integration technique.)

Planar Planar Spatial or Spatial?

Store on permanent file if specified

Subroutine FUNCT (To compute the value of the integrand for velocity potential and upwash terms.)

Spatial

Subroutine BFUNC (Retrieves Bessel function from stored array.)

Subroutine VFUNC (To compute the value of the integrand for Sidewash terms)

Store on permanent file if specified.

Store on scratch file

NV

RETURN
Program NWVLT Secondary overlay which calculates normal washes and velocity potentials.

ENTRY

Read IBOXW codes from IGEOSC into array IBOXW (2nd read for out-of-plane tail)

Read FEXLC and TEXLOC arrays from IGEOSC

If: PSIW0 & DIHW-T.
or PSIT0 & DIHT-T.
or PSIDF0
or CAPL0
or ISMLW0
yes
no

Read spatial AIC table of contents from IGEOSC into array PAIC

Read IPNTR, the mode shape / pointer array from MODESC

Call POINTR to generate IPNTDW, the (unsubdivided) normal-wash pointer array.

IMODE = 1, NMODES

Read Deflections and Slopes from MODESC into array DEFSL.

Call VELPOT to compute N_RJ1, N_RJ1, N_DN1, and trailing edge N_DN1, TVP for the wing (and tail, if coplanar).

Print normal-washes, optionally

noncoplanar no
tail? yes

call VELPOT to compute N_RU1, N_DG, and TVP for the spatial tail

Print tail normal-washes, optionally

140

Write N_DG and TVP arrays onto IVFSC

Print the N_DG array, optionally

Is sampling of the wash desired? yes

JCHRD = 1, ISMLW

Call SMPLW, which computes and prints the downwash, sidewash and longitudinal wash for the desired chord.

500

RETURN
Program SMTH: Secondary overlay which smooths velocity potentials by fitting a least squares surface through them

ENTRY

Read row and column pointer arrays from mode scratch file

Read the geometry information from the geometry scratch file

NM = 1, NMODES

Read the velocity potentials

NS = 1, NSURF

Build arrays of coordinates (x and y) and corresponding velocity potentials (AVPS)

Call Subroutine FITTER to fit a least squares polynomial surface through the data points. Subroutine will return polynomial coefficients

Evaluate polynomial equations to obtain vel. pot. values at box centers.

Compute trailing edge vel. pot. by linear extrapolation

NS

Write the new velocity potentials on a scratch file.

NM

RETURN
Program CHORDF — Secondary overlay which smooths velocity potentials by fitting a least squares curve along each chord.

ENTRY

Read row and column pointer arrays from mode scratch file.

Read the Geometry information from the geometry scratch file.

$NM = 1, NMODES$

Read the velocity potentials

$J = 1, NCHRDS$

Build arrays of coordinate (x) and slope of vel. pot. for boxes on chord J.

Call Subroutine FITTER to fit a least squares polynomial curve through the data points. Subroutine will return coefficients.

Evaluate integral of polynomial to obtain vel. pot. values at box centers.

Compute trailing edge vel. pot. by linear extrapolation.

Write the new velocity potentials on a scratch file.

RETURN
Program FORCES  Secondary overlay which computes box lifts section lifts, and generalized air forces

ENTRY

Read Row pointers from mode scratch file

Read Geometry data from geometry scratch file

ITSLOP = 1,NTSLOP(# of thickness slopes)

Read Thickness slope functions

NM =1,NMODES (# of modes)

Read Modal values for mode shape NM

JVP = 1,NVPS (# of vel. pot. or mode shapes)

Read Velocity potentials, for mode JVP

Compute the box lifts for each planform box,

Compute the generalized air force for velocity potential case JVP

Write the generalized air forces on the output tape, print all requested values

RETURN
REFERENCES


APPENDIX A

SAMPLE INPUT AND OUTPUT DATA

A simple spatial configuration, shown in Figure 23, was chosen as a sample problem for the demonstration of the card data input and a selection of the printed output. The planform used is a pair of identical rectangular surfaces (wing and tail) with small horizontal and vertical separation.

![Figure 23 Sample Problem Configuration](image)

The configuration was analyzed at Mach 1.2 for a reduced frequency (based on semi-span) of .5. Only the wing surface was allowed to oscillate, in plunging motion for mode 1 and in pitch about the wing leading edge for mode 2. These two modes were input on cards as polynomials. Chordwise velocity potential smoothing was requested.

In the interest of space the printout was edited to give samples only. A few pages of one spatial AIC array and the planar AIC are included, as well as most of the computations for mode 2 (wing pitch). Since for this configuration the upper and lower surface normal wash differs only in sign, only the upper normal washes are included. The generalized force calculations at the end are for smoothed velocity potentials.
Card Input Data

AFMDOX 1 0
SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS
$CARD xmin=1.20$
$CARD sy=1.0, MTYPE=1, MTYPE=1, NSURF=2, WTGNAF=.T., WTDL=.T.,
PRGNAC=.T., PROD=.T.,
PLYMUP = .T.,
CRKFG = .T., KECC=4,
PRGNAF=.T., PRDL=.T., PRSL=.T., PRPAIC=.T., PRSAIC=.T., PRDCF=.T.,
PRDOX=.T., PRVF=.T., EXAI=.F., PRMK=.T., PRMODS=.T. $
$$CARD$
$CARD XXS(1) = .50$
$CARD TLX=1.20, TLZ=.40$
$CARD NCHRD=10, XEDGE=0.00$

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

$CARDM NAXCES=2$

2
1.0 0.0 0.0 0.0
2.0 1.0 0.0 0.0
1.0 0.0 0.0
1.0 0.0 0.0

WING-1
WING-2
TAIL-1
TAIL-2

A2
SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

THE FOLLOWING OPTIONS ARE REQUESTED -

MACH NUMBER = 1.200000
SIMMETRIC ANALYSIS
FLYWOOD OPTION IS USED, (FLYWOOD BOUNDARY DETERMINED BY DWX PATTERN.)
BASIC (UNSPLITTED) ANALYSIS WILL BE USED
ANALYSIS FOR 2 FLYWOODS
APPROXIMATE KERNELS WILL BE USED
VELOCITY POTENTIALS WILL BE SMOOTHED BY A LEAST SQUARES
POLYNOMIAL CHORDWISE FIT, OF ORDER 4.
(\(Q_0\) = PROGRAM DETERMINED.)
PRINT THE DWX PATTERN
PRINT MODE SHAPE POLYNOMIAL COEFFICIENTS, IF AVAILABLE
PRINT MODE SHAPES USED
PRINT THE FLANAR AIC ARRAYS USED
PRINT THE SPATIAL AIC ARRAYS USED
PRINT NORMAL WASHES
PRINT 1/2 VELOCITY POTENTIALS
PRINT THE DWX LIFTS
PRINT THE SECTION LIFTS
PRINT PRESSURE DIFFERENCES COEFFICIENTS
PRINT GENERALIZED AERODYNAMIC COEFFICIENTS
PRINT GENERALIZED AIR FORCES
WRITE BOX LIFTS ON TAPE
WRITE GENERALIZED AIR FORCES ON TAPE
MODAL INPUT FOR WING IS POLYNOMIAL COEFFICIENTS
MODAL INPUT FOR TAIL IS POLYNOMIAL COEFFICIENTS
DINERDAL WING INFLUENCE CALCULATED
DINERDAL TAIL INFLUENCE CALCULATED

THE FOLLOWING TAPE SETUP IS REQUESTED -

OLD AIC TAPE = 0
NEW AIC TAPE = 0
OLD SPATIAL AIC TAPE = 0
NEW SPATIAL AIC TAPE = 0
INPUT DATA TAPE = 0 SPACED 0 FILES;
OUTPUT TAPE = 1 SPACED 0 FILES;

THE FOLLOWING IS THE REDUCED FREQUENCY ARRAY BASED ON WING SEMI-SFAN

.35000

ENTERING PROGRAM GEOM CURRENT ELAPSED TIME IS CP = .224, FP = 37.757

--- GEOMETRIC PARAMETERS ---

CARDF -LOCAL AXES DEFINITION-
WING 0.000 0.000 0.000 DEGREES
TAIL 1.200 1.400 0.000 DEGREES

CARDG -BOX PATTERN DEFINITION-
NUMBER 10 0.0000.0000 0.0000

CARDM -PLANFORM DEFINITION POINT COUNTS-
LEADING EDGE TRAILING EDGE
WING 2 2
TAIL 2 2

CASE TO CASEL -PLANFORM DEFINITIONS-
WING L.E. 0.000 0.000
0.000 1.000
<table>
<thead>
<tr>
<th></th>
<th>1.000</th>
<th>0.000</th>
<th>1.000</th>
<th>1.000</th>
<th>0.000</th>
<th>0.000</th>
<th>1.000</th>
<th>1.000</th>
</tr>
</thead>
<tbody>
<tr>
<td>WING T.C.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAIL L.C.</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAIL T.C.</td>
<td>1.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

-BOX DIMENSIONS-  
D1 (LENGTH) = 6.63324958e-02  
D1/DETA (WIDTH) = 1.00000000e-01
SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

BOX CODE PATTERN
MACH 1.2000000

| CODE | 1 = PLANFORM BOX
| 2 = DIAPHRAGM BOX
| 3 = WAKE BOX |

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>27</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>28</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>29</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>31</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>32</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>33</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
SAMPLE CASE --- TWO AR:2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

BOX CODE PATTERN
MACH 1.2000000

<table>
<thead>
<tr>
<th>CODE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PLANFORM BOX</td>
</tr>
<tr>
<td>2</td>
<td>DIAFRAGM BOX</td>
</tr>
<tr>
<td>3</td>
<td>WAKE BOX</td>
</tr>
</tbody>
</table>

---

2 4 6 8 10 12 14 16 18 20 22 24 26

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>16</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>18</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>21</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>22</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>23</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>24</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>25</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>26</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>27</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>28</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>29</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

ENTERING PROGRAM MODES CURRENT ELAPSED TIME 15 CP = 1.894, FP = 40.123
SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

MODE SHAPE NUMBER 2
MACH NUMBER = 1.200000

MODAL POLYNOMIAL COEFFICIENTS

FROM CARD INPUT

CONSTANT
------------
0.

X #11 Y #00 X #22 Y #01
-------------------
1.00000E+00 0.

X #22 Y #00 X #22 Y #01 X #22 Y #02
-------------------
0. 0. 0.

MODAL POLYNOMIAL COEFFICIENTS

FROM CARD INPUT

CONSTANT
------------
0.

X #11 Y #00 X #22 Y #01
-------------------
0. 0.

DEFLECTIONS X 1.0E 1

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.332</td>
<td>.332</td>
<td>.332</td>
<td>.332</td>
<td>.332</td>
<td>.332</td>
<td>.332</td>
<td>.332</td>
<td>.332</td>
<td>.332</td>
</tr>
<tr>
<td>2</td>
<td>.995</td>
<td>.995</td>
<td>.995</td>
<td>.995</td>
<td>.995</td>
<td>.995</td>
<td>.995</td>
<td>.995</td>
<td>.995</td>
<td>.995</td>
</tr>
<tr>
<td>4</td>
<td>2.322</td>
<td>2.322</td>
<td>2.322</td>
<td>2.322</td>
<td>2.322</td>
<td>2.322</td>
<td>2.322</td>
<td>2.322</td>
<td>2.322</td>
<td>2.322</td>
</tr>
<tr>
<td>5</td>
<td>2.985</td>
<td>2.985</td>
<td>2.985</td>
<td>2.985</td>
<td>2.985</td>
<td>2.985</td>
<td>2.985</td>
<td>2.985</td>
<td>2.985</td>
<td>2.985</td>
</tr>
<tr>
<td>7</td>
<td>4.312</td>
<td>4.312</td>
<td>4.312</td>
<td>4.312</td>
<td>4.312</td>
<td>4.312</td>
<td>4.312</td>
<td>4.312</td>
<td>4.312</td>
<td>4.312</td>
</tr>
<tr>
<td>8</td>
<td>4.975</td>
<td>4.975</td>
<td>4.975</td>
<td>4.975</td>
<td>4.975</td>
<td>4.975</td>
<td>4.975</td>
<td>4.975</td>
<td>4.975</td>
<td>4.975</td>
</tr>
<tr>
<td>9</td>
<td>5.638</td>
<td>5.638</td>
<td>5.638</td>
<td>5.638</td>
<td>5.638</td>
<td>5.638</td>
<td>5.638</td>
<td>5.638</td>
<td>5.638</td>
<td>5.638</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>1</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>2</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>3</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>4</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>5</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>6</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>7</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>8</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>9</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>10</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>11</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>12</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>13</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>14</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>15</td>
<td>1.500</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>16</td>
<td>0.500</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

SLOPES X 1.0E 0
### Sample Case --- Two AR=2 Surfaces with Horizontal and Vertical Separations

**AIC Calculations**

\[
\text{MACH} = 1.20000 \quad \text{K1} = 0.0321662 \quad \text{ERR} = 1.000000 \times 10^{-2} \quad \text{EL} = 4.00 \quad \text{YBAR} = 0.00
\]

<table>
<thead>
<tr>
<th>W</th>
<th>VELOCITY POTENTIAL COEFFICIENT</th>
<th>( W )</th>
<th>UPWASH COEFFICIENT</th>
<th>( V )</th>
<th>SIDEWASH COEFFICIENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1 1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1 0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1 -1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2 2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2 0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2 -1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2 -2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3 3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3 2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3 1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3 -1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3 -2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3 -3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4 4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4 3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4 2</td>
<td>-3.47636676E-02</td>
<td>1.8022436E-02</td>
<td>1.9726808E-01</td>
<td>-1.0036316E+01</td>
<td>9.1953259E+02</td>
</tr>
<tr>
<td>4 1</td>
<td>-1.1776083E-01</td>
<td>5.8538846E-02</td>
<td>1.7205458E-01</td>
<td>-7.5633097E-02</td>
<td>4.3922895E-02</td>
</tr>
<tr>
<td>4 0</td>
<td>-1.59655064E-01</td>
<td>6.3298524E-02</td>
<td>1.5964821E-01</td>
<td>-6.5034828E-02</td>
<td>0.0</td>
</tr>
<tr>
<td>4 -1</td>
<td>-1.1776083E-01</td>
<td>5.8538846E-02</td>
<td>1.7205458E-01</td>
<td>-7.5633097E-02</td>
<td>4.3922895E-02</td>
</tr>
<tr>
<td>4 -2</td>
<td>-3.47636676E-02</td>
<td>1.8022436E-02</td>
<td>1.9726808E-01</td>
<td>-1.0036316E+01</td>
<td>9.1953259E+02</td>
</tr>
<tr>
<td>4 -3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4 -4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>5 5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>5 4</td>
<td>-1.26580111E-02</td>
<td>8.4561092E-03</td>
<td>7.5703692E-02</td>
<td>-4.9966828E-02</td>
<td>6.9668587E-02</td>
</tr>
<tr>
<td>5 3</td>
<td>-1.1608895E-01</td>
<td>7.2567877E-02</td>
<td>1.1064121E-01</td>
<td>-6.0228606E-02</td>
<td>8.3626078E-02</td>
</tr>
<tr>
<td>5 2</td>
<td>-1.45526451E-01</td>
<td>8.5703207E-02</td>
<td>9.8310750E-02</td>
<td>5.9366506E-02</td>
<td>-4.2522420E-02</td>
</tr>
<tr>
<td>5 1</td>
<td>-9.9653889E-02</td>
<td>5.9220433E-02</td>
<td>-6.8837551E-02</td>
<td>3.9775079E-02</td>
<td>-1.8045378E-02</td>
</tr>
<tr>
<td>5 0</td>
<td>-9.18644239E-02</td>
<td>5.4681286E-02</td>
<td>-5.2389549E-02</td>
<td>3.0481507E-02</td>
<td>0.0</td>
</tr>
<tr>
<td>5 -1</td>
<td>-9.9653889E-02</td>
<td>5.9220433E-02</td>
<td>-6.8837551E-02</td>
<td>3.9775079E-02</td>
<td>1.8045378E-02</td>
</tr>
</tbody>
</table>
SAMPLE CASE --- TWO AREZ SURF
ORIENTAL AND VERTICAL SEPARATIONS

AIC CALCULATIONS

<table>
<thead>
<tr>
<th>MU</th>
<th>NO</th>
<th>VELOCITY POTENTIAL COEFFICIENT</th>
<th>UWASH COEFFICIENT</th>
<th>SIDEWASH COEFFICIENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-4.99669441E-01</td>
<td>1.3561231E-02</td>
<td>0.</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-3.6376G888E-01</td>
<td>3.7265379E-02</td>
<td>0.</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>-3.62537716E-01</td>
<td>3.5177404E-02</td>
<td>0.</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>-1.58274582E-01</td>
<td>3.4105949E-02</td>
<td>0.</td>
</tr>
<tr>
<td>2</td>
<td>-1</td>
<td>-1.98775746E-01</td>
<td>4.2138443E-02</td>
<td>0.</td>
</tr>
<tr>
<td>2</td>
<td>-2</td>
<td>-2.06053425E-01</td>
<td>4.7517355E-02</td>
<td>0.</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-9.8370396E-02</td>
<td>3.2854373E-02</td>
<td>0.</td>
</tr>
<tr>
<td>3</td>
<td>-1</td>
<td>-1.0549007E-02</td>
<td>3.5183514E-02</td>
<td>0.</td>
</tr>
<tr>
<td>3</td>
<td>-2</td>
<td>-1.47785247E-02</td>
<td>4.6633720E-02</td>
<td>0.</td>
</tr>
<tr>
<td>3</td>
<td>-3</td>
<td>-1.62295376E-02</td>
<td>5.6626110E-02</td>
<td>0.</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>-6.8152526E-02</td>
<td>3.1385644E-02</td>
<td>0.</td>
</tr>
<tr>
<td>4</td>
<td>-1</td>
<td>-7.0829074E-02</td>
<td>3.2603888E-02</td>
<td>0.</td>
</tr>
<tr>
<td>4</td>
<td>-2</td>
<td>-6.8774972E-02</td>
<td>3.7215369E-02</td>
<td>0.</td>
</tr>
<tr>
<td>4</td>
<td>-3</td>
<td>-1.16703126E-01</td>
<td>5.3954733E-02</td>
<td>0.</td>
</tr>
<tr>
<td>4</td>
<td>-4</td>
<td>-1.3404642E-02</td>
<td>6.3842648E-02</td>
<td>0.</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>-4.9350001E-02</td>
<td>2.9614994E-02</td>
<td>0.</td>
</tr>
<tr>
<td>5</td>
<td>-1</td>
<td>-5.0632371E-02</td>
<td>3.0379093E-02</td>
<td>0.</td>
</tr>
<tr>
<td>5</td>
<td>-2</td>
<td>-5.5023520E-02</td>
<td>3.2992061E-02</td>
<td>0.</td>
</tr>
<tr>
<td>5</td>
<td>-3</td>
<td>-6.5067119E-02</td>
<td>3.8967149E-02</td>
<td>0.</td>
</tr>
<tr>
<td>5</td>
<td>-4</td>
<td>-9.8152209E-02</td>
<td>5.6190438E-02</td>
<td>0.</td>
</tr>
<tr>
<td>5</td>
<td>-5</td>
<td>-1.12751838E-01</td>
<td>6.9620709E-02</td>
<td>0.</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>-3.63509608E-02</td>
<td>2.7556287E-02</td>
<td>0.</td>
</tr>
<tr>
<td>6</td>
<td>-1</td>
<td>-3.70965078E-02</td>
<td>2.6055896E-02</td>
<td>0.</td>
</tr>
<tr>
<td>6</td>
<td>-2</td>
<td>-3.92999919E-02</td>
<td>2.9814568E-02</td>
<td>0.</td>
</tr>
<tr>
<td>6</td>
<td>-3</td>
<td>-4.39152690E-02</td>
<td>3.3294611E-02</td>
<td>0.</td>
</tr>
<tr>
<td>6</td>
<td>-4</td>
<td>-5.31898999E-02</td>
<td>4.0271417E-02</td>
<td>0.</td>
</tr>
<tr>
<td>6</td>
<td>-5</td>
<td>-8.1940610E-02</td>
<td>6.1487834E-02</td>
<td>0.</td>
</tr>
<tr>
<td>6</td>
<td>-6</td>
<td>-9.5238540E-02</td>
<td>7.4189025E-02</td>
<td>0.</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>-2.66803249E-02</td>
<td>7.5257261E-02</td>
<td>0.</td>
</tr>
<tr>
<td>7</td>
<td>-1</td>
<td>-2.7094405E-02</td>
<td>7.5643242E-02</td>
<td>0.</td>
</tr>
<tr>
<td>7</td>
<td>-2</td>
<td>-2.84124875E-02</td>
<td>2.6885957E-02</td>
<td>0.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>14</td>
<td>-14</td>
<td>-3.18510175E-03</td>
<td>7.79147557E-02</td>
<td>0.</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>2.396573885E-04</td>
<td>4.51992878E-03</td>
<td>0.</td>
</tr>
<tr>
<td>15</td>
<td>-1</td>
<td>2.43896544E-04</td>
<td>4.59273037E-03</td>
<td>0.</td>
</tr>
<tr>
<td>15</td>
<td>-2</td>
<td>2.56161477E-04</td>
<td>4.61246030E-03</td>
<td>0.</td>
</tr>
<tr>
<td>15</td>
<td>-3</td>
<td>2.77124615E-04</td>
<td>5.16876549E-03</td>
<td>0.</td>
</tr>
<tr>
<td>15</td>
<td>-4</td>
<td>3.07618702E-04</td>
<td>5.75829446E-03</td>
<td>0.</td>
</tr>
<tr>
<td>15</td>
<td>-5</td>
<td>3.48331100E-04</td>
<td>6.47935638E-03</td>
<td>0.</td>
</tr>
<tr>
<td>15</td>
<td>-6</td>
<td>4.02957664E-04</td>
<td>7.52142171E-03</td>
<td>0.</td>
</tr>
<tr>
<td>15</td>
<td>-7</td>
<td>4.72467459E-04</td>
<td>8.70813507E-03</td>
<td>0.</td>
</tr>
<tr>
<td>15</td>
<td>-8</td>
<td>5.61571139E-04</td>
<td>1.03179566E-02</td>
<td>0.</td>
</tr>
<tr>
<td>15</td>
<td>-9</td>
<td>6.76593686E-04</td>
<td>1.24666678E-02</td>
<td>0.</td>
</tr>
<tr>
<td>15</td>
<td>-10</td>
<td>8.27843435E-04</td>
<td>1.51689736E-02</td>
<td>0.</td>
</tr>
<tr>
<td>15</td>
<td>-11</td>
<td>1.03563264E-03</td>
<td>1.89613281E-02</td>
<td>0.</td>
</tr>
<tr>
<td>15</td>
<td>-12</td>
<td>1.33109416E-03</td>
<td>2.45299399E-02</td>
<td>0.</td>
</tr>
<tr>
<td>15</td>
<td>-13</td>
<td>1.81457923E-03</td>
<td>3.58406982E-02</td>
<td>0.</td>
</tr>
<tr>
<td>15</td>
<td>-14</td>
<td>2.90106485E-03</td>
<td>5.86996286E-02</td>
<td>0.</td>
</tr>
<tr>
<td>15</td>
<td>-15</td>
<td>5.11249095E-03</td>
<td>7.50310604E-02</td>
<td>0.</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>3.90947265E-04</td>
<td>2.44086971E-03</td>
<td>0.</td>
</tr>
<tr>
<td>16</td>
<td>-1</td>
<td>4.00826484E-04</td>
<td>2.50047643E-03</td>
<td>0.</td>
</tr>
<tr>
<td>16</td>
<td>-2</td>
<td>4.30859044E-04</td>
<td>2.68137753E-03</td>
<td>0.</td>
</tr>
<tr>
<td>16</td>
<td>-3</td>
<td>4.61964034E-04</td>
<td>2.96997947E-03</td>
<td>0.</td>
</tr>
<tr>
<td>16</td>
<td>-4</td>
<td>5.56137462E-04</td>
<td>3.4575361E-03</td>
<td>0.</td>
</tr>
<tr>
<td>16</td>
<td>-5</td>
<td>6.5617024E-04</td>
<td>4.04138196E-03</td>
<td>0.</td>
</tr>
<tr>
<td>16</td>
<td>-6</td>
<td>7.86172918E-04</td>
<td>4.82632687E-03</td>
<td>0.</td>
</tr>
<tr>
<td>16</td>
<td>-7</td>
<td>9.51981429E-04</td>
<td>5.82796599E-03</td>
<td>0.</td>
</tr>
<tr>
<td>16</td>
<td>-8</td>
<td>1.16102929E-03</td>
<td>7.09745339E-03</td>
<td>0.</td>
</tr>
<tr>
<td>16</td>
<td>-9</td>
<td>1.43858055E-03</td>
<td>8.71031398E-03</td>
<td>0.</td>
</tr>
<tr>
<td>16</td>
<td>-10</td>
<td>1.7739494E-03</td>
<td>1.0782972CE-02</td>
<td>0.</td>
</tr>
<tr>
<td>16</td>
<td>-11</td>
<td>2.21935250E-03</td>
<td>1.35065064E-02</td>
<td>0.</td>
</tr>
<tr>
<td>16</td>
<td>-12</td>
<td>2.63027404E-03</td>
<td>1.7226170CE-02</td>
<td>0.</td>
</tr>
<tr>
<td>16</td>
<td>-13</td>
<td>3.17636508E-03</td>
<td>2.26623895E-02</td>
<td>0.</td>
</tr>
<tr>
<td>16</td>
<td>-14</td>
<td>5.16722035E-03</td>
<td>3.17493266E-02</td>
<td>0.</td>
</tr>
<tr>
<td>16</td>
<td>-15</td>
<td>8.84566854E-03</td>
<td>5.87618732E-02</td>
<td>0.</td>
</tr>
<tr>
<td>16</td>
<td>-16</td>
<td>1.2669990E-02</td>
<td>7.1560999E-02</td>
<td>0.</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>1.50787753E-02</td>
<td>6.2164924E-02</td>
<td>0.</td>
</tr>
<tr>
<td>17</td>
<td>-1</td>
<td>1.7240093AE-04</td>
<td>6.70254190E-04</td>
<td>0.</td>
</tr>
<tr>
<td>17</td>
<td>-2</td>
<td>2.1368206E-04</td>
<td>8.17782706E-04</td>
<td>0.</td>
</tr>
<tr>
<td>17</td>
<td>-3</td>
<td>2.81984663E-04</td>
<td>1.0965996E-03</td>
<td>0.</td>
</tr>
<tr>
<td>17</td>
<td>-4</td>
<td>3.85675665E-04</td>
<td>1.43250087E-03</td>
<td>0.</td>
</tr>
<tr>
<td>17</td>
<td>-5</td>
<td>5.22318763E-04</td>
<td>1.9209429E-03</td>
<td>0.</td>
</tr>
<tr>
<td>17</td>
<td>-6</td>
<td>6.98969092E-04</td>
<td>2.55524234E-03</td>
<td>0.</td>
</tr>
<tr>
<td>17</td>
<td>-7</td>
<td>9.22720356E-04</td>
<td>3.33261876E-03</td>
<td>0.</td>
</tr>
<tr>
<td>17</td>
<td>-8</td>
<td>1.20537875E-03</td>
<td>4.35552728E-03</td>
<td>0.</td>
</tr>
<tr>
<td>17</td>
<td>-9</td>
<td>1.55490316E-03</td>
<td>5.61442762E-03</td>
<td>0.</td>
</tr>
<tr>
<td>ROW</td>
<td>CHORD 1 REAL</td>
<td>CHORD 1 IMAGINARY</td>
<td>CHORD 2 REAL</td>
<td>CHORD 2 IMAGINARY</td>
</tr>
<tr>
<td>-----</td>
<td>-------------</td>
<td>-------------------</td>
<td>-------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>1</td>
<td>6.3324993E+02</td>
<td>1.10000000E+00</td>
<td>6.3324993E+02</td>
<td>1.10000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>6.3324993E+02</td>
<td>1.10000000E+00</td>
<td>6.3324993E+02</td>
<td>1.10000000E+00</td>
</tr>
<tr>
<td>3</td>
<td>6.3324993E+02</td>
<td>1.10000000E+00</td>
<td>6.3324993E+02</td>
<td>1.10000000E+00</td>
</tr>
<tr>
<td>4</td>
<td>6.3324993E+02</td>
<td>1.10000000E+00</td>
<td>6.3324993E+02</td>
<td>1.10000000E+00</td>
</tr>
<tr>
<td>5</td>
<td>6.3324993E+02</td>
<td>1.10000000E+00</td>
<td>6.3324993E+02</td>
<td>1.10000000E+00</td>
</tr>
<tr>
<td>6</td>
<td>6.3324993E+02</td>
<td>1.10000000E+00</td>
<td>6.3324993E+02</td>
<td>1.10000000E+00</td>
</tr>
<tr>
<td>7</td>
<td>6.3324993E+02</td>
<td>1.10000000E+00</td>
<td>6.3324993E+02</td>
<td>1.10000000E+00</td>
</tr>
<tr>
<td>8</td>
<td>6.3324993E+02</td>
<td>1.10000000E+00</td>
<td>6.3324993E+02</td>
<td>1.10000000E+00</td>
</tr>
<tr>
<td>9</td>
<td>6.3324993E+02</td>
<td>1.10000000E+00</td>
<td>6.3324993E+02</td>
<td>1.10000000E+00</td>
</tr>
<tr>
<td>10</td>
<td>6.3324993E+02</td>
<td>1.10000000E+00</td>
<td>6.3324993E+02</td>
<td>1.10000000E+00</td>
</tr>
</tbody>
</table>

Sample Case: Two airfoil surfaces with horizontal and vertical separations.
### SAMPLE CASE --- TWO ARizz SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS

**WING UPPPER SURFACE NORMAL WASH**

(MACH 1.200  RED. FREQ. = .50000)

MODE SHAPE 2

<table>
<thead>
<tr>
<th>ROW</th>
<th>CHORD 13</th>
<th>CHORD 14</th>
<th>CHORD 15</th>
<th>CHORD 16</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>REAL</td>
<td>IMAGINARY</td>
<td>REAL</td>
<td>IMAGINARY</td>
</tr>
<tr>
<td>1</td>
<td>-2.44373466E-03</td>
<td>1.09582337E-03</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>2</td>
<td>-2.44373466E-03</td>
<td>1.09582337E-03</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>3</td>
<td>-2.44373466E-03</td>
<td>1.09582337E-03</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>4</td>
<td>-2.44373466E-03</td>
<td>1.09582337E-03</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>5</td>
<td>-2.44373466E-03</td>
<td>1.09582337E-03</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>6</td>
<td>-2.44373466E-03</td>
<td>1.09582337E-03</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>7</td>
<td>-2.44373466E-03</td>
<td>1.09582337E-03</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>8</td>
<td>-2.44373466E-03</td>
<td>1.09582337E-03</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>9</td>
<td>-2.44373466E-03</td>
<td>1.09582337E-03</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>10</td>
<td>-2.44373466E-03</td>
<td>1.09582337E-03</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>11</td>
<td>-2.44373466E-03</td>
<td>1.09582337E-03</td>
<td>0.</td>
<td>0.</td>
</tr>
</tbody>
</table>

**PAGE CONTINUED**
### Sample Case --- Two AR#2 Surfaces with Horizontal and Vertical Separations

#### Wing Upper Surface Normal Wash (MAC 1.200, RED. FREQ. .50000)

##### Mode Shape 2

<table>
<thead>
<tr>
<th>CHORD</th>
<th>REAL</th>
<th>IMAGINARY</th>
<th>REAL</th>
<th>IMAGINARY</th>
<th>REAL</th>
<th>IMAGINARY</th>
<th>REAL</th>
<th>IMAGINARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>-1.74717044E-04</td>
<td>4.41133459E-04</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>13</td>
<td>-3.82337016E-04</td>
<td>1.06704076E-03</td>
<td>-1.10360272E-04</td>
<td>4.00858990E-04</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>15</td>
<td>-7.10895020E-04</td>
<td>3.7512436E-03</td>
<td>-3.02229597E-04</td>
<td>1.62435034E-03</td>
<td>-1.07912734E-04</td>
<td>0.87692371E-04</td>
<td>-2.04470869E-03</td>
<td>3.2813976E-04</td>
</tr>
<tr>
<td>16</td>
<td>-8.36221800E-04</td>
<td>4.72553607E-03</td>
<td>-3.30185935E-04</td>
<td>2.31286034E-03</td>
<td>-1.07120906E-04</td>
<td>1.4606239E-03</td>
<td>-1.28916542E-03</td>
<td>1.9761850E-04</td>
</tr>
<tr>
<td>17</td>
<td>-9.53291720E-04</td>
<td>5.64259432E-03</td>
<td>-3.12425674E-04</td>
<td>2.97521932E-03</td>
<td>-1.74634961E-03</td>
<td>2.06706664E-03</td>
<td>2.60353521E-03</td>
<td>1.3082984E-03</td>
</tr>
<tr>
<td>18</td>
<td>-1.55209046E-04</td>
<td>3.69123467E-03</td>
<td>-1.38033583E-03</td>
<td>4.20660277E-03</td>
<td>1.36125357E-03</td>
<td>3.22023215E-03</td>
<td>2.56144046E-03</td>
<td>2.35952355E-03</td>
</tr>
<tr>
<td>20</td>
<td>-3.45287732E-04</td>
<td>1.47639909E-03</td>
<td>1.39643516E-03</td>
<td>5.25260686E-03</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
</tbody>
</table>

### CHORD 25

<table>
<thead>
<tr>
<th>REAL</th>
<th>IMAGINARY</th>
<th>REAL</th>
<th>IMAGINARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>2.1165252E-05</td>
<td>2.95121951E-04</td>
<td>G.</td>
</tr>
<tr>
<td>17</td>
<td>3.25330385E-05</td>
<td>7.14594427E-04</td>
<td>3.34829454E-05</td>
</tr>
<tr>
<td>18</td>
<td>1.82446667E-05</td>
<td>1.1646662E-03</td>
<td>0.</td>
</tr>
<tr>
<td>ROW</td>
<td>CHORD 1</td>
<td>CHORD 2</td>
<td>CHORD 3</td>
</tr>
<tr>
<td>-----</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>1</td>
<td>0.91137621E-02</td>
<td>-2.47824535E-02</td>
<td>0.04920791E-02</td>
</tr>
<tr>
<td>2</td>
<td>-2.46441094E-02</td>
<td>0.62604131E-02</td>
<td>-2.46804922E-02</td>
</tr>
<tr>
<td>3</td>
<td>-3.0589411E-02</td>
<td>6.0511666E-02</td>
<td>-3.13032001E-02</td>
</tr>
<tr>
<td>4</td>
<td>-2.0664497E-02</td>
<td>1.2060762E-02</td>
<td>-2.8653215E-02</td>
</tr>
<tr>
<td>5</td>
<td>-3.3929794E-02</td>
<td>1.2262512E-02</td>
<td>-3.40362409E-02</td>
</tr>
<tr>
<td>6</td>
<td>-3.11028235E-02</td>
<td>1.7725001E-02</td>
<td>-3.05313725E-02</td>
</tr>
<tr>
<td>7</td>
<td>-3.9330104E-02</td>
<td>1.7616909E-02</td>
<td>-3.33764304E-02</td>
</tr>
<tr>
<td>8</td>
<td>-3.1811544E-02</td>
<td>2.3066896E-02</td>
<td>-3.1235647E-02</td>
</tr>
<tr>
<td>9</td>
<td>-3.6796652E-02</td>
<td>2.1230089E-02</td>
<td>-3.33209339E-02</td>
</tr>
<tr>
<td>10</td>
<td>-3.1134092E-02</td>
<td>2.6089763E-02</td>
<td>-3.04566532E-02</td>
</tr>
<tr>
<td>11</td>
<td>-3.4938749E-02</td>
<td>2.8420947E-02</td>
<td>-3.42220704E-02</td>
</tr>
<tr>
<td>12</td>
<td>-2.9378414E-02</td>
<td>2.8505377E-02</td>
<td>-2.6873845E-02</td>
</tr>
<tr>
<td>13</td>
<td>-3.2822241E-02</td>
<td>2.8908517E-02</td>
<td>-2.3620260E-02</td>
</tr>
<tr>
<td>14</td>
<td>-2.7262424E-02</td>
<td>3.15949597E-02</td>
<td>-2.68237195E-02</td>
</tr>
<tr>
<td>15</td>
<td>-3.0525828E-02</td>
<td>3.16924532E-02</td>
<td>-3.0248866E-02</td>
</tr>
</tbody>
</table>

A27
SAMPLE CASE --- TWO ARIZ SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS
TAIL UPPER SURFACE NORMAL WASH
(MACH 1.200 RED. FREQ. .50000)

<table>
<thead>
<tr>
<th>ROW</th>
<th>CHORD 9</th>
<th>CHORD 10</th>
<th>CHORD 11</th>
<th>CHORD 12</th>
<th>PAGE CONTINUED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>REAL</td>
<td>IMAGINARY</td>
<td>REAL</td>
<td>IMAGINARY</td>
<td>REAL</td>
</tr>
<tr>
<td>30</td>
<td>-1.19607E-02</td>
<td>-2.09037E-03</td>
<td>-4.59307E-03</td>
<td>-5.29569E-03</td>
<td>4.09517E-02</td>
</tr>
<tr>
<td>31</td>
<td>-1.19439E-02</td>
<td>-2.75531E-03</td>
<td>-4.91897E-03</td>
<td>-2.60736E-03</td>
<td>3.69389E-02</td>
</tr>
<tr>
<td>32</td>
<td>-1.09709E-02</td>
<td>-3.67242E-03</td>
<td>-4.67608E-03</td>
<td>-1.96746E-03</td>
<td>3.39353E-02</td>
</tr>
<tr>
<td>33</td>
<td>-1.11407E-02</td>
<td>-4.42532E-03</td>
<td>-4.83870E-03</td>
<td>-1.39367E-03</td>
<td>0.00000E+00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ROW</th>
<th>CHORD 13</th>
<th>CHORD 14</th>
<th>CHORD 15</th>
<th>CHORD 16</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>REAL</td>
<td>IMAGINARY</td>
<td>REAL</td>
<td>IMAGINARY</td>
</tr>
<tr>
<td>30</td>
<td>1.18494E-02</td>
<td>6.89323E-04</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
</tr>
<tr>
<td>31</td>
<td>1.73626E-02</td>
<td>9.39932E-04</td>
<td>1.94029E-03</td>
<td>3.13609E-04</td>
</tr>
<tr>
<td>32</td>
<td>1.03353E-02</td>
<td>8.93392E-04</td>
<td>3.90095E-04</td>
<td>5.66509E-04</td>
</tr>
<tr>
<td>33</td>
<td>1.00972E-02</td>
<td>1.00972E-04</td>
<td>1.39367E-03</td>
<td>1.70718E-04</td>
</tr>
<tr>
<td>34</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
</tr>
<tr>
<td>35</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
</tr>
</tbody>
</table>

A 29

<table>
<thead>
<tr>
<th>ROW</th>
<th>CHORD 17</th>
<th>CHORD 18</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>REAL</td>
<td>IMAGINARY</td>
</tr>
<tr>
<td>30</td>
<td>3.84941E-02</td>
<td>2.83413E-05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ROW</th>
<th>CHORD 19</th>
<th>CHORD 20</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>REAL</td>
<td>IMAGINARY</td>
</tr>
<tr>
<td>30</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
</tr>
<tr>
<td>ROW</td>
<td>CHORD 1</td>
<td>CHORD 2</td>
</tr>
<tr>
<td>-----</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>2</td>
<td>1.98901532E-01</td>
<td>1.06032774E-02</td>
</tr>
<tr>
<td>4</td>
<td>4.52690913E-01</td>
<td>5.90664299E-02</td>
</tr>
<tr>
<td>7</td>
<td>7.92203056E-02</td>
<td>1.91133573E-01</td>
</tr>
<tr>
<td>10</td>
<td>1.05679945E-02</td>
<td>3.45581915E-02</td>
</tr>
<tr>
<td>11</td>
<td>1.24145399E+00</td>
<td>4.15341562E-01</td>
</tr>
<tr>
<td>12</td>
<td>1.16482998E+01</td>
<td>4.37471514E+00</td>
</tr>
<tr>
<td>13</td>
<td>1.15667399E+01</td>
<td>4.3490359E-02</td>
</tr>
<tr>
<td>14</td>
<td>1.22347731E+01</td>
<td>4.14683356E+00</td>
</tr>
<tr>
<td>15</td>
<td>1.24801529E+01</td>
<td>3.83210227E+00</td>
</tr>
</tbody>
</table>
SAMPLE CASE --- TWO AR=2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS
WING VELOCITY POTENTIALS
(MACH 1.200   RED. FREQ. * .50000 )
MODE SHAPE 2

<table>
<thead>
<tr>
<th>ROW</th>
<th>CHORD 9</th>
<th></th>
<th>CHORD 10</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>REAL</td>
<td>IMAGINARY</td>
<td>REAL</td>
<td>IMAGINARY</td>
</tr>
<tr>
<td>12</td>
<td>-6.21203999E-01</td>
<td>9.66617232E-02</td>
<td>-4.03949368E-01</td>
<td>5.07496370E-02</td>
</tr>
<tr>
<td>13</td>
<td>-6.33321893E-01</td>
<td>9.71598345E-02</td>
<td>-4.12613332E-01</td>
<td>5.04889462E-02</td>
</tr>
<tr>
<td>14</td>
<td>-6.47431767E-01</td>
<td>9.81715199E-02</td>
<td>-4.20176186E-01</td>
<td>4.99178886E-02</td>
</tr>
<tr>
<td>15</td>
<td>-6.60142073E-01</td>
<td>9.90824242E-02</td>
<td>-4.26709994E-01</td>
<td>4.58274281E-02</td>
</tr>
</tbody>
</table>
SAMPLE CASE --- TWO AIR SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS
TAIL VELOCITY POTENTIALS
(MACH 1.200   RED. FREQ. = .50000 )
MODE SHAPE 2

<table>
<thead>
<tr>
<th>ROW</th>
<th>CHORD 9 REAL</th>
<th>CHORD 9 IMAGINARY</th>
<th>CHORD 10 REAL</th>
<th>CHORD 10 IMAGINARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>1.50820579E-01</td>
<td>-1.60200176E-01</td>
<td>6.55962221E-02</td>
<td>-6.84275988E-02</td>
</tr>
<tr>
<td>31</td>
<td>1.40998919E-01</td>
<td>-1.84285674E-01</td>
<td>7.94188208E-02</td>
<td>-1.03457324E-01</td>
</tr>
<tr>
<td>32</td>
<td>1.28780370E-01</td>
<td>-2.05906326E-01</td>
<td>7.26917087E-02</td>
<td>-1.16966095E-01</td>
</tr>
<tr>
<td>33</td>
<td>1.17931394E-01</td>
<td>-2.24969247E-01</td>
<td>6.52551056E-02</td>
<td>-1.28700313E-01</td>
</tr>
</tbody>
</table>

ENTERING PROGRAM CHORDF CURRENT ELAPSED TIME IS CP = 122.434; PP = 36.058
<table>
<thead>
<tr>
<th>Row</th>
<th>Chord 1</th>
<th>Chord 2</th>
<th>Chord 3</th>
<th>Chord 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Real</td>
<td>Imaginary</td>
<td>Real</td>
<td>Imaginary</td>
</tr>
<tr>
<td>1</td>
<td>-6.3184769E-02</td>
<td>-6.3184769E-02</td>
<td>-6.3184769E-02</td>
<td>-6.3184769E-02</td>
</tr>
<tr>
<td>2</td>
<td>-1.9761817E-01</td>
<td>-1.9761817E-01</td>
<td>-1.9761817E-01</td>
<td>-1.9761817E-01</td>
</tr>
<tr>
<td>3</td>
<td>5.3255639E-01</td>
<td>5.3255639E-01</td>
<td>5.3255639E-01</td>
<td>5.3255639E-01</td>
</tr>
<tr>
<td>4</td>
<td>-4.4803023E-01</td>
<td>-4.4803023E-01</td>
<td>-4.4803023E-01</td>
<td>-4.4803023E-01</td>
</tr>
<tr>
<td>5</td>
<td>-5.9616408E-01</td>
<td>-5.9616408E-01</td>
<td>-5.9616408E-01</td>
<td>-5.9616408E-01</td>
</tr>
<tr>
<td>6</td>
<td>-6.4408949E-01</td>
<td>-6.4408949E-01</td>
<td>-6.4408949E-01</td>
<td>-6.4408949E-01</td>
</tr>
<tr>
<td>7</td>
<td>-7.9172192E-01</td>
<td>-7.9172192E-01</td>
<td>-7.9172192E-01</td>
<td>-7.9172192E-01</td>
</tr>
<tr>
<td>8</td>
<td>-6.9033749E-01</td>
<td>-6.9033749E-01</td>
<td>-6.9033749E-01</td>
<td>-6.9033749E-01</td>
</tr>
<tr>
<td>10</td>
<td>-1.0515962E+00</td>
<td>-1.0515962E+00</td>
<td>-1.0515962E+00</td>
<td>-1.0515962E+00</td>
</tr>
<tr>
<td>11</td>
<td>-1.1259666E+00</td>
<td>-1.1259666E+00</td>
<td>-1.1259666E+00</td>
<td>-1.1259666E+00</td>
</tr>
<tr>
<td>12</td>
<td>-1.1863813E+00</td>
<td>-1.1863813E+00</td>
<td>-1.1863813E+00</td>
<td>-1.1863813E+00</td>
</tr>
<tr>
<td>13</td>
<td>-1.1929995E+00</td>
<td>-1.1929995E+00</td>
<td>-1.1929995E+00</td>
<td>-1.1929995E+00</td>
</tr>
<tr>
<td>14</td>
<td>-1.2331962E+00</td>
<td>-1.2331962E+00</td>
<td>-1.2331962E+00</td>
<td>-1.2331962E+00</td>
</tr>
<tr>
<td>15</td>
<td>-1.2444504E+00</td>
<td>-1.2444504E+00</td>
<td>-1.2444504E+00</td>
<td>-1.2444504E+00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Row</th>
<th>Chord 5</th>
<th>Chord 6</th>
<th>Chord 7</th>
<th>Chord 8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Real</td>
<td>Imaginary</td>
<td>Real</td>
<td>Imaginary</td>
</tr>
<tr>
<td>1</td>
<td>6.9025447E-02</td>
<td>6.9025447E-02</td>
<td>6.9025447E-02</td>
<td>6.9025447E-02</td>
</tr>
<tr>
<td>2</td>
<td>1.9021860E-01</td>
<td>1.9021860E-01</td>
<td>1.9021860E-01</td>
<td>1.9021860E-01</td>
</tr>
<tr>
<td>3</td>
<td>3.3160443E-01</td>
<td>3.3160443E-01</td>
<td>3.3160443E-01</td>
<td>3.3160443E-01</td>
</tr>
<tr>
<td>4</td>
<td>4.5962918E-01</td>
<td>4.5962918E-01</td>
<td>4.5962918E-01</td>
<td>4.5962918E-01</td>
</tr>
<tr>
<td>5</td>
<td>5.7610568E-01</td>
<td>5.7610568E-01</td>
<td>5.7610568E-01</td>
<td>5.7610568E-01</td>
</tr>
<tr>
<td>6</td>
<td>6.4170379E-01</td>
<td>6.4170379E-01</td>
<td>6.4170379E-01</td>
<td>6.4170379E-01</td>
</tr>
<tr>
<td>7</td>
<td>6.8467034E-01</td>
<td>6.8467034E-01</td>
<td>6.8467034E-01</td>
<td>6.8467034E-01</td>
</tr>
<tr>
<td>12</td>
<td>1.0285960E+00</td>
<td>1.0285960E+00</td>
<td>1.0285960E+00</td>
<td>1.0285960E+00</td>
</tr>
<tr>
<td>13</td>
<td>1.0299025E+00</td>
<td>1.0299025E+00</td>
<td>1.0299025E+00</td>
<td>1.0299025E+00</td>
</tr>
<tr>
<td>14</td>
<td>1.0299025E+00</td>
<td>1.0299025E+00</td>
<td>1.0299025E+00</td>
<td>1.0299025E+00</td>
</tr>
<tr>
<td>15</td>
<td>1.0299025E+00</td>
<td>1.0299025E+00</td>
<td>1.0299025E+00</td>
<td>1.0299025E+00</td>
</tr>
</tbody>
</table>
SAMPLE CASE --- TWO AREZ SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS
SMOOTHED WING VELOCITY POTENTIALS
(MACH 1.200 RED. FREQ. = .30000)
NODE SHAPE Z

<table>
<thead>
<tr>
<th>ROW</th>
<th>CHORD 9 REAL</th>
<th>CHORD 9 IMAGINARY</th>
<th>CHORD 10 REAL</th>
<th>CHORD 10 IMAGINARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>-6.1441010E-01</td>
<td>9.30535425E-02</td>
<td>-4.01612967E-01</td>
<td>5.04614257E-02</td>
</tr>
<tr>
<td>13</td>
<td>-6.2929244E-01</td>
<td>9.41080248E-02</td>
<td>-4.1030572E-01</td>
<td>5.01163937E-02</td>
</tr>
<tr>
<td>14</td>
<td>-6.4235339E-01</td>
<td>9.44866311E-02</td>
<td>-4.11010000E-01</td>
<td>4.85289414E-02</td>
</tr>
<tr>
<td>15</td>
<td>-6.5375402E-01</td>
<td>9.11906015E-02</td>
<td>-4.2310945E-01</td>
<td>4.59264449E-02</td>
</tr>
<tr>
<td>ROW</td>
<td>CHORD 1 REAL</td>
<td>CHORD 1 IMAGINARY</td>
<td>CHORD 2 REAL</td>
<td>CHORD 2 IMAGINARY</td>
</tr>
<tr>
<td>-----</td>
<td>--------------</td>
<td>------------------</td>
<td>--------------</td>
<td>------------------</td>
</tr>
<tr>
<td>19</td>
<td>2.57263780E+00</td>
<td>2.31088510E+00</td>
<td>5.91216175E+00</td>
<td>2.03040104E+00</td>
</tr>
<tr>
<td>20</td>
<td>1.25798157E+00</td>
<td>1.60348775E+00</td>
<td>1.36235601E+00</td>
<td>1.76307423E+00</td>
</tr>
<tr>
<td>21</td>
<td>2.01395101E+00</td>
<td>3.64117900E+00</td>
<td>8.01017365E+00</td>
<td>1.58353937E+00</td>
</tr>
<tr>
<td>22</td>
<td>2.49729536E+00</td>
<td>4.79061935E+00</td>
<td>2.57114895E+00</td>
<td>4.41446737E+00</td>
</tr>
<tr>
<td>23</td>
<td>3.06926959E+00</td>
<td>9.00211343E+00</td>
<td>3.06545565E+00</td>
<td>9.38770404E+00</td>
</tr>
<tr>
<td>24</td>
<td>3.53902130E+00</td>
<td>1.97445801E+00</td>
<td>3.49737711E+00</td>
<td>1.51733593E+00</td>
</tr>
<tr>
<td>25</td>
<td>3.91481459E+00</td>
<td>2.22615939E+00</td>
<td>3.62817078E+00</td>
<td>2.14519415E+00</td>
</tr>
<tr>
<td>26</td>
<td>4.19856076E+00</td>
<td>2.89107822E+00</td>
<td>4.14618298E+00</td>
<td>2.79230274E+00</td>
</tr>
<tr>
<td>27</td>
<td>4.43281665E+00</td>
<td>3.54181232E+00</td>
<td>4.34083333E+00</td>
<td>3.43154856E+00</td>
</tr>
<tr>
<td>28</td>
<td>4.82994647E+00</td>
<td>4.13733910E+00</td>
<td>4.42640966E+00</td>
<td>4.03952052E+00</td>
</tr>
<tr>
<td>29</td>
<td>4.49666718E+00</td>
<td>4.78183450E+00</td>
<td>4.39553477E+00</td>
<td>4.95727560E+00</td>
</tr>
<tr>
<td>30</td>
<td>4.32350069E+00</td>
<td>5.21147912E+00</td>
<td>4.24710021E+00</td>
<td>6.09135864E+00</td>
</tr>
<tr>
<td>31</td>
<td>4.07809389E+00</td>
<td>5.63020373E+00</td>
<td>3.99108200E+00</td>
<td>5.51466544E+00</td>
</tr>
<tr>
<td>32</td>
<td>3.73357676E+00</td>
<td>5.93777773E+00</td>
<td>3.65925526E+00</td>
<td>5.80683113E+00</td>
</tr>
<tr>
<td>33</td>
<td>3.59555519E+00</td>
<td>6.23612502E+00</td>
<td>3.28492738E+00</td>
<td>6.15479068E+00</td>
</tr>
</tbody>
</table>

**NOTE:** SHAPE 3
SAMPLE CASE --- TWO AX=Z SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS
WING BOX LIFTS
(MACH 1.200  RED.FREQ. = 0.30000 )
MODE SHAPE 2

<table>
<thead>
<tr>
<th>NO.</th>
<th>CHORD 9 REAL</th>
<th>CHORD 9 IMAGINARY</th>
<th>CHORD 10 REAL</th>
<th>CHORD 10 IMAGINARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>-5.74588492E-02</td>
<td>-5.76043018E-02</td>
<td>-3.41384900E-02</td>
<td>-3.92062991E-02</td>
</tr>
<tr>
<td>13</td>
<td>-3.1727248E-02</td>
<td>-4.49718898E-02</td>
<td>-2.82247455E-02</td>
<td>-4.39344020E-02</td>
</tr>
<tr>
<td>14</td>
<td>-4.63255066E-02</td>
<td>-7.18270061E-02</td>
<td>-2.43881753E-02</td>
<td>-4.80176656E-02</td>
</tr>
<tr>
<td>15</td>
<td>-4.39031367E-02</td>
<td>-7.32472235E-02</td>
<td>-2.30606948E-02</td>
<td>-5.01546887E-02</td>
</tr>
</tbody>
</table>

PAGE CONTINUED
SAMPLE CASE --- TWO AREA SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS
TAIL BOX LIFTS
(MACH 1.200 RED. FREQ. = .50000 )
MODE SHAPE 2

<table>
<thead>
<tr>
<th>ROW</th>
<th>CHORD 9</th>
<th>CHORD 10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>REAL</td>
<td>IMAGINARY</td>
</tr>
<tr>
<td>30</td>
<td>-1.129566E+02</td>
<td>-5.043263E+02</td>
</tr>
<tr>
<td>31</td>
<td>-1.196894E+02</td>
<td>-5.225059E+02</td>
</tr>
<tr>
<td>32</td>
<td>-1.230997E+02</td>
<td>-4.710339E+02</td>
</tr>
<tr>
<td>33</td>
<td>-1.193140E+02</td>
<td>-4.507596E+02</td>
</tr>
<tr>
<td>ROW</td>
<td>CHORD 1 REAL</td>
<td>CHORD 1 IMAGINARY</td>
</tr>
<tr>
<td>-----</td>
<td>--------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>1</td>
<td>-5.94231125E+00</td>
<td>2.04920801E-01</td>
</tr>
<tr>
<td>2</td>
<td>-1.89212067E+00</td>
<td>3.19371946E-01</td>
</tr>
<tr>
<td>3</td>
<td>-5.76628454E+00</td>
<td>6.19966979E-01</td>
</tr>
<tr>
<td>4</td>
<td>-5.63021345E+00</td>
<td>7.10695353E-01</td>
</tr>
<tr>
<td>5</td>
<td>-5.46077904E+00</td>
<td>1.05940634E+00</td>
</tr>
<tr>
<td>6</td>
<td>-5.26423263E+00</td>
<td>1.33653129E+00</td>
</tr>
<tr>
<td>7</td>
<td>-4.91167339E+00</td>
<td>1.43322471E+00</td>
</tr>
<tr>
<td>8</td>
<td>-4.59602812E+00</td>
<td>1.35415703E+00</td>
</tr>
<tr>
<td>9</td>
<td>-4.34111238E+00</td>
<td>1.01661132E+00</td>
</tr>
<tr>
<td>10</td>
<td>-3.62312184E+00</td>
<td>4.50049999E-01</td>
</tr>
<tr>
<td>11</td>
<td>-3.06854146E+00</td>
<td>3.04051413E-01</td>
</tr>
<tr>
<td>12</td>
<td>-2.94030326E+00</td>
<td>-1.17307548E+00</td>
</tr>
<tr>
<td>13</td>
<td>-2.97892695E+00</td>
<td>-2.05325415E+00</td>
</tr>
<tr>
<td>14</td>
<td>-1.76193146E+00</td>
<td>-2.61019807E+00</td>
</tr>
<tr>
<td>15</td>
<td>-1.62743118E+00</td>
<td>-3.16136368E+00</td>
</tr>
</tbody>
</table>

**SAMPLE CASE --- TWO AIRPLANE SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS**

**WING PRESSURE DIFFERENCE**

(MACH 1.200  REACT. FREQ. = .50000)

**NODE SHAPE 2**

---

**Note:** The table contains complex numbers for chord values, indicating steady-state behavior of an aerofoil profile.
SAMPLE CASE --- TWO AREZ SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS
WING PRESS. DIFFERENCE
(MACH 1.200   RED. FREQ. = .50000)
MODE SHAPE 2

<table>
<thead>
<tr>
<th>ROW</th>
<th>CHORD 9 REAL</th>
<th>CHORD 9 IMAGINARY</th>
<th>CHORD 10 REAL</th>
<th>CHORD 10 IMAGINARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>-6.40237400E-01</td>
<td>-8.71432634E-01</td>
<td>-5.14657102E-01</td>
<td>-5.91097198E-01</td>
</tr>
<tr>
<td>13</td>
<td>-7.79749415E-01</td>
<td>-9.79488093E-01</td>
<td>-4.25504049E-01</td>
<td>-6.62336031E-01</td>
</tr>
<tr>
<td>14</td>
<td>-6.94435918E-01</td>
<td>-1.07881775E+00</td>
<td>-3.57669577E-01</td>
<td>-7.23893546E-01</td>
</tr>
<tr>
<td>15</td>
<td>-4.55834462E-01</td>
<td>-1.13596298E+00</td>
<td>-3.47788765E-01</td>
<td>-7.36110352E-01</td>
</tr>
</tbody>
</table>

PAGE CONTINUED
<table>
<thead>
<tr>
<th>CHORD</th>
<th>REAL</th>
<th>IMAGINARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0001</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0010</td>
<td>2.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0100</td>
<td>3.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1000</td>
<td>4.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0002</td>
<td>-1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0101</td>
<td>-2.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1001</td>
<td>-3.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1100</td>
<td>-4.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**Sample Case:**

- **Two A=2 Surfaces with Horizontal and Vertical Separations**
- **Tail Press Difference (1 Mach = 1.0 Red.)**
- **Mode Shape:**

---

**Table:**

<table>
<thead>
<tr>
<th>CHORD</th>
<th>REAL</th>
<th>IMAGINARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0001</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0010</td>
<td>2.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0100</td>
<td>3.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1000</td>
<td>4.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0002</td>
<td>-1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0101</td>
<td>-2.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1001</td>
<td>-3.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1100</td>
<td>-4.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

---

**Diagram:**

- **A 51**
SAMPLE CASE --- TWO AR2 SURFACES WITH HORIZONTAL AND VERTICAL SEPARATIONS
TAIL PRESS. DIFFERENCE
(MACH 1.200 RED. FREQ. = .50000 )
MODE SHAPE 2
-----

CHORD 9

<table>
<thead>
<tr>
<th>ROW</th>
<th>REAL</th>
<th>IMAGINARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>-1.69699093E-01</td>
<td>-8.35718793E-01</td>
</tr>
<tr>
<td>31</td>
<td>-1.80439946E-01</td>
<td>-7.87707407E-01</td>
</tr>
<tr>
<td>32</td>
<td>-1.63973978E-01</td>
<td>-7.14110350E-01</td>
</tr>
<tr>
<td>33</td>
<td>-1.73463893E-01</td>
<td>-6.81603196E-01</td>
</tr>
</tbody>
</table>

CHORD 10

<table>
<thead>
<tr>
<th>ROW</th>
<th>REAL</th>
<th>IMAGINARY</th>
</tr>
</thead>
</table>

SECTION Lifts
(MACH 1.200 RED. FREQ. = .50000 )
MODE SHAPE 2
WING
-----

CHORD

<table>
<thead>
<tr>
<th>REAL</th>
<th>IMAGINARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4.16178096E+00</td>
<td>-7.82410686E-02</td>
</tr>
<tr>
<td>-3.02659830E+00</td>
<td>-3.08564912E+00</td>
</tr>
<tr>
<td>-2.06428380E+00</td>
<td>-4.33417750E+00</td>
</tr>
</tbody>
</table>

TOTAL Lift - 4.53

-3.21259904E+00 -2.59287395E+00

SECTION Lifts
TAIL
-----

CHORD

<table>
<thead>
<tr>
<th>REAL</th>
<th>IMAGINARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.34101091E+00</td>
<td>1.42763947E+00</td>
</tr>
<tr>
<td>1.11200019E+00</td>
<td>1.17471445E+00</td>
</tr>
<tr>
<td>4.33719974E-01</td>
<td>5.11652856E-01</td>
</tr>
</tbody>
</table>

TOTAL Lift - TAIL

PAGE CONTINUED
## Generalized Forces
(Mach 1.200  Red. Freq. = .90000)

<table>
<thead>
<tr>
<th>FUNCT VELOCITY POTENTIAL MODE 1</th>
<th>VELOCITY POTENTIAL MODE 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>REAL IMAGINARY</td>
<td>REAL IMAGINARY</td>
</tr>
<tr>
<td>1 -3.45284231E+00 -1.4924449E+01</td>
<td>-3.21293304E+01 -2.39207395E+00</td>
</tr>
<tr>
<td>2 -1.31903979E+00 -4.90711190E+00</td>
<td>-1.14917114E+01 -3.45896395E+00</td>
</tr>
</tbody>
</table>
GENERALIZED AERODYNAMIC COEFFICIENTS
(MACH 1.200  RED.FREQ. = .30000)
REAL PART

<table>
<thead>
<tr>
<th>FUNCT</th>
<th>VELOCITY POTENTIAL MODES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.45E9+231E-01 3.12E35904E+00</td>
</tr>
<tr>
<td>2</td>
<td>1.31502372E-01 1.14917114E+00</td>
</tr>
</tbody>
</table>
GENERALIZED AERODYNAMIC COEFFICIENTS
(MACH 1.200  RED.FREQ. = .50000)

---------------------------------------------

VELOCITY POTENTIAL MODES

<table>
<thead>
<tr>
<th>WT.</th>
<th>FUNCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

E 98457567C+00  5.16574791E-01
E 97422299C-01  6.91782786E-01

ENTERING PROGRAM FORCES CURRENT ELAPSED TIME IS  CP = 125.982, FP = 66.297

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

PROGRAM FORCES IS BEING RECALLED TO COMPUTE AIR FORCES WITHOUT SMOOTHING.

*******************************************************************************
APPENDIX B

PROGRAM LISTINGS
OVERLAY (AFMBCE, 0, 1)

PROGRAM DRIVER (INPUT, OUTPUT, TAPE3=INPUT, TAPE6=OUTPUT, TAPE1=1000).

TAPE2=1000, TAPE3=1000, MODESC=110, IVPSG=110, IGEOCS=110.

IWFSC=110, IAICSC=110)

C

THIS IS A DUMMY (0, 0) OVERLAY DRIVING PROGRAM

C

COMMON PKERNL (1640)

COMPLEX PKERNL

C

COMMON /FILES/ NT5, NT6, INTAPE, INFSP, NPLASIC, NSPAIC, NOUTP,

1 IOUFSP, MODESC, IVPSG, IGEOCS, IWFSC, IAICSC

COMMON /ARRAYS/ KBXCDM, LBXCDM, LBCWC, KBXCDT, LBXCDT, KJALRH, LJALRH,

1 KALPHA, KKERNL, UKERNL, KPNTNM, LPNTNM, KDEFSL, KELRH,

2 LMODES, KPNTSD, LPNTSD, KSDW, LSDK, KPNDW, LPNDW,

3 KMWM, LDWM, KTVP, LTVP

DATA ITPEI, MODESC, IVPSG, IGEOCS, IWFSC, IAICSC/

SLTAPEI, 6LMODESC, 6LIVPSG, 6LIGEOCS, 6LWFTSC, 6LIAICSC /

NT5 = 5

NT6 = 6

READ (5, 5) LINK, L3, L2

5 FORMAT (A6, 4X, 2110)

WRITE (6, 6) LINK, L1, L2

6 FORMAT (*1 PROGRAM BEGINS *, A6, 215)

CALL OVERLAY (LINK, L1, L2, 0)

WRITE (6, 7)

7 FORMAT (*O PROGRAM TERMINATES*)

CALL EXIT

END
OVERLAY (AFMBOX, 1, 0)
PROGRAM CONTROL

SUPERSONIC UNSTEADY AERODYNAMICS,
WING / HORIZONTAL TAIL, VERTICAL SEPARATION, DIHEDRAL

THIS PRIMARY OVERLAY CONTROLS THE CALLING OF THE COMPUTATIONAL
SECTIONS OF THE PROGRAM

THIS IS THE TOTAL COMMON FOR ALL THE OVERLAY STRUCTURE
COMMON / CONTROL/ PREVEX, ONACH, TITLE(8), PROVGEO, PRVMODE, DIHWT, DIHIT, 0002
1

1 DEFAULT

LOGICAL
PROVGEOM, PRVMODE, DIHWT, DIHIT, DEFAULT

1 LOGICAL

EXAIC, SUBDV, PLYWOOD

1 LOGICAL

SMOOTH, CRDFIT, EXAIC, SUBDV, PLYWOOD

1 COMMON

COPLAN, NSUBDV, XSUBDV, NSUBD2, NSUBCN, NSURF,

1 COMMON

BI, BIBETA, BIBIAS, MLAX, MLAZ, PRSM

2 MNSB, MNSB2, MNW, MNBW, MNSSW, MNSBW, MNSBSW

3 MIBW, MIBW2, MIW, MIBWT, MIBST, MIBST2

LOGICAL

COPLAN

1 COMMON

TLX, TLZ, PSIT, MBST, MYBT, MYBT2, MBST2, MYST

1 COMMON

MYBST, IXBT, IXBSTR, CARL

1 COMMON

KERN, ERR, NASKIN, IPKRN, NPL2K, NSPATK, NROWEA

1 COMMON

IKVAL, [KVAL(20), KKS(20)]

1 COMMON

NY5, NT6, INTAPE, IFSP, NPLAIC, NSPAIC, NNOIT,

1 COMMON

IGOFSP, MODES, IVISC, NGEOS, IVITSC, IAIISC

1 COMMON

I/OCONTROL/ OPLAIC, OSFAIC, WGEOM, MTGNAF, WTLW, WTLB, PRBOXK

1 COMMON

PREAIC, PSRAIC, PRMODS, PRCOEF, PRDM, PRSW, PRVP,

2 PRBL, PRDCP, PRGNAF, PRGANC, PRSL, PRLM, PRNM, FROM

1 COMMON

EBMTAIC, IRFMU, IPRMU, IFRMU

1 COMMON

LOGICAL

EBMTVALENCE (PRM, PRW)

1 COMMON

LOGICAL

OPLAIC, OSFAIC, WGEOM, MTGNAF, WTLW, WTLB, PRBOXK, PRPAIC,

1 COMMON

PREAIC, PSRAIC, PRMODS, PRCOEF, PRDM, PRSW, PRVP, PRBL, PRSL, PRGNAF,

2 PRDCP, PRGNAF, PRSL, PRLM, PRNM, FROM

1 COMMON

TOAPEX/ NS, NSL, LS, NR1, DR1(20), DIOTYPE, LS, LW5, M, N

1 COMMON

PARM(10), IR

1 DIMENSION IPARM(10)

1 DIMENSION IPARM(10)

1 DIMENSION IPARM(10)

1 COMMON

/MODES/ SYM, SYMT, MTYREW, MTYPET

1 COMMON

/MODES/ SYM, SYMT, MTYREW, MTYPET

1 COMMON

/ARRAYS/ KBXCDW, LBDXCDW, LBDRAC, KBXCDT, LBXCDT, KJALPH, LJALPH

1 COMMON

/ARRAYS/ KBXCDW, LBDXCDW, LBDRAC, KBXCDT, LBXCDT, KJALPH, LJALPH

1 COMMON

/KALPHA, KALPHN, KALEN, LPNTHM, KFNS, KHALEN, KLHPH, KLFPHH

1 COMMON

/KALPHA, KALPHN, KALEN, LPNTHM, KFNS, KHALEN, KLHPH, KLFPHH

1 COMMON

/LMOMOD, LMPHDW, LPMHDW, LSDW, LPSWE, LPMNDW, LPMNDW

1 COMMON

/LMOMOD, LMPHDW, LPMHDW, LSDW, LPSWE, LPMNDW, LPMNDW

3 COMMON

KLM, KDW, KTMV, LTV

3 COMMON

KLM, KDW, KTMV, LTV

1 COMMON

/SAMPLW/ ISMLPW, ICHORD(10), IBORF(10), IBORF(10), ZLOC(10)

1 COMMON

/SAMPLW/ ISMLPW, ICHORD(10), IBORF(10), IBORF(10), ZLOC(10)

1 COMMON

/NLME, NLTE, NLTE, XLME(10), XLTE(10), YLME(10), YLTE(10)

1 COMMON

/NLME, NLTE, NLTE, XLME(10), XLTE(10), YLME(10), YLTE(10)

2 COMMON

/XLTE(10), YLTE(10)

2 COMMON

/XLTE(10), YLTE(10)

1 COMMON

/CHCKPR/ DPPCPR, GEOPCR, NDCPR, AICCR, NABCPR, NABCPR, GACPFR

1 COMMON

/CHCKPR/ DPPCPR, GEOPCR, NDCPR, AICCR, NABCPR, NABCPR, GACPFR

1 COMMON

/BUF/ BDOC, IBFUND, BUNC(3280)

1 COMMON

/BUF/ BDOC, IBFUND, BUNC(3280)

1 DATA

BCFUND, IBFUND, 8BHUFFSIZE, 3280 /

1 DATA

BCFUND, IBFUND, 8BHUFFSIZE, 3280 /

1 DATA

TEVICY /ETAPH/ / DATA

1 DATA

TEVICY /ETAPH/ / DATA

1 DATA

PREVEX /IOHNEVER EXEC /

1 DATA

EXEC /IOHAFMBOX EXEC /

1 CALL

CONTINUE

1 CONTINUE

PROG = SHFATAPP

B3
CALL DTIME(CPTIME, PPTIME)
WRITE (INT6, 6005) PROG, CPTIME, PPTIME

%RITE WT.WOS) PROPCPTII.,PPTIME Cl'-ErRa. 00034

qj05
FRM4AT (UE3, Oxt*EWTERING PROCRAN
*AS,*
CURRENT ELAPSED TIME IS
CWirC
00035

* CP -- V3q*p
PP =*,F8.3
CamcRO
00036

CALL RELAY(TEV147,1,1,0)
IPARM(5) = NVALS
IF (CMACH .EQ. XMACH AND. PRIVATE) GO TO 100

CALL DTIME(CPTIME, PPTIME)
WRITE (W1T6,6005) PROG, CPTIME, PPTIME

100 CONTINUE
IF (NVAWS .LE. 0) GO TO 810

CALL DTIME(CPTIME, PPTIME)
WRITE (INT6, 6005) PROG, CPTIME, PPTIME
CALL OVERLAY(TEV147,1,3,0)

SPACE OUTPUT TAP IF DESIRED
IF (MOUTP .LE. 0) GO TO 200
IF (PREVEX .NE. EXEC) REWIND MOUTP
FILE SPACING A FUNCTION OF INSTALLATION CAPABILITIES

200 CONTINUE
LOOP ON NUMBER OF K1 VALUES THRU KERNELS, DOWNWASHES AND
AIR FORCES
DO 800 INVAL = 1,NKVALS

CALL KERNEL ROUTINES
PROG = OMAIC
CALL DTIME(CPTIME, PPTIME)
WRITE (INT6, 6005) PROG, CPTIME, PPTIME
CALL OVERLAY(TEV147,1,4,0)

PROG = OVELPOT
CALL DTIME(CPTIME, PPTIME)
WRITE (INT6, 6005) PROG, CPTIME, PPTIME
CALL DOWNWASH AND VELOCITY POTENTIAL ROUTINES.
CAL CALL OVERLAY(TEV147,1,5,0)

800 CONTINUE

IF (.NOT.SMOOTH) GO TO 800
IF (CRDFIT) GO TO 800

PROG = OSHMOOTH
CALL DTIME(CPTIME, PPTIME)
WRITE (INT6, 6005) PROG, CPTIME, PPTIME
CALL OVERLAY(TEV147,1,6,0)
GO TO 700
600 CONTINUE
   IF(.NOT.CRDFIT) GO TO 700
   PROG = 6HORDF
   CALL DTIME(CPTIME,PPTIME)
   WRITE (NT6,6003) PROG,CPTIME,PPTIME
   CALL OVERLAY(TEY147,1,7,0)
   CONTINUE

700 CONTINUE
   PROG = 6HORDF
   CALL DTIME(CPTIME,PPTIME)
   WRITE (NT6,6003) PROG,CPTIME,PPTIME
   CALL OVERLAY(TEY147,1,8,0)
   IF(.NOT.(SMOOTH.OR.CRDFIT)) GO TO 800
   NVPS = IACP
   IAIC = IVPSC
   IVPSC = NVPS
   CALL DTIME(CPTIME,PPTIME)
   WRITE (NT6,6003) PROG,CPTIME,PPTIME
   WRITE (NT6,6010)
   6010 FORMAT(10D,5X,90(1H*),//,6X,*PROGRAM FORCES IS BEING RECALLED TO CONTROL 00110
   1COMPUTE AIR FORCES WITHOUT SMOOTHING,4//6X,90(1H*))
   CALL OVERLAY(TEY147,1,8,9RECALL)
   CONTINUE

800 CONTINUE
   END OF LOOP ON REDUCED FREQUENCIES
   CONTINUE

810 CONTINUE
   IF (NOUTP .GT. 0) REWIND NOUTP
   IF (NPRA .EQ. 0) REWIND NPRA
   IF (NPRA .EQ. 0) REWIND NPRA
   IF (NPLA .GT. 0) REWIND NPLA
   CALL DTIME(CPTIME,PPTIME)
   WRITE (NT6,6006) CPTIME,PPTIME
   6006 FORMAT(10D,10X,*PROGRAM COMPLETED,6X,*CURRENT ELAPSED TIME IS *
   1*CP =*,FB,3,* PF =*,FB,3)
   READ(5,6005) LINK,L1,L2
   6005 FORMAT(16,4X,9,110)
   CONTINUE
   END

C DETERMINE IF ANOTHER CYCLE IS WANTED.
C IF L1 .EQ. -1, RECYCLE
   IF(L1.EQ.-1) GO TO 1
   IF(L1.EQ.-2) RETURN
   IF(L1.EQ.0) CALL EXIT
   IF(L1.EQ.D) CALL EXIT
   IF(L1.GT.0) CALL OVERLAY(LINK,L1,L2,0)

END
SUBROUTINE FLUSH(I)
C ROUTINE TO FORCE AN ERROR EXIT
DIMENSION MESSGE(I)
DATA MESSAGE /ICH PROGRAM F,ICHLSHED VIA,ICH MODE I , D /
DATA MT5 /6,OUTPUT/
WRITE (MT5,8000) (MESSAGE(I),I=1,3)
ENDFILE MT5
CALL REMARK(MESSAGE)
CALL FLUSHXXX
8000 FORMAT(5H### , 3A10, 4H ###)
END
SUBROUTINE RDINIT
COMMON /TAPEIO/ NFS, NHS, LS, NR, ID(20), NED, IYPE, LRS, LWB, M, N,
1 PARM(ID), IRR
DIMENSION IA(I), PARM(I), IRS
EQUIVALENCE (IA, IFS)
DO 10 I = 1,41
IA(I) = 0
10 CONTINUE
NED = 20
RETURN
END
SUBROUTINE DTIME(CPTIME, PPTIME)

    ROUTINE TO INTERROGATE THE SYSTEM CLOCKS

    PPTIME = 0
    CALL SECOND(CPTIME)
    RETURN

END

DTIME  00002
DTIME  00003
DTIME  00004
DTIME  00005
DTIME  00006
DTIME  00007
SUBROUTINE READMX(INFILE, NRAED, RAN DIN, NFS, NNS, LS, NMR, K, NID, READX 00002
               ID, ITYPE, LRS, A, M, N, PARM, IRP )
              READX 00003
C      ROUTINE TO READ A MATRIX ON TAPE OR DISK FILE.
C      THIS VERSION WILL WORK WITH SEQUENTIAL FILES ONLY.
C      SOME VARIABLES ARE PASSED FOR RANDOM OPERATION BUT
C      ARE NOT CURRENTLY USED.
C      INPUT -
C      INFILE - TAPE NUMBER OR LEFT ADJUSTED FILE NAME
C      NRAED - .T. SMART FORMAT (NOT USED)
C      .F. TELEDI FORMAT
C      RAN DIN - .T. RANDOM FILE (NOT USED)
C      .F. SEQUENTIAL FILE
C      NFS - NUMBER OF FILES TO SPACE
C      NNS - NUMBER OF MATRICES TO SPACE
C      LS - LEVEL NUMBER TO SPACE (NOT USED)
C      NMR - IDENTIFIER (NAME OR NUMBER) (NOT USED)
C      K - ROW DIMENSION OF ARRAY A
C      (IF K=0, MATRIX WILL BE LEFT IN /RWBUFF, IT WILL
C      BE STORED AS A ROW-WISE MATRIX, NOT AS A FORTRAN
C      COLUMN-WISE MATRIX, M-ROWS AND N-COLUMNS )
C      NID - NUMBER OF WORDS AVAILABLE IN ID ARRAY
C      OUTPUT -
C      ID - IDENTIFICATION ARRAY
C      ITYPE - REAL, DIAGNOS, NULL, MIXED, COMPLEX
C      LRS - LEVEL NUMBER OF MATRIX READ (NOT USED)
C      A - ARRAY CONTAINING MATRIX
C      M - ROW DIMENSION OF MATRIX
C      N - COLUMN DIMENSION OF MATRIX
C      PARM - ARRAY OF NUMERICAL PARAMETERS STORED WITH THE MATRIX
C      IRP -
C      0, NO ERROR
C      1, MATRIX SPACING IS NEGATIVE
C      2, FILE SPACING IS NEGATIVE
C      4, MATRIX DIMENSIONS ILLEGAL
C      5, M .GT. K
C      1500 + I, ENCONTERED ' ' AFTER MATRIX I WHILE
C      SKIPPING MATRICES.
C      DIMENSION ID(1), A(K,1), PARM(10), B(16)
C      COMMON/RWBUFF/ BFICODE, IBFCNT, BUFF(3280)
C      DIMENSION IBUFF(2500), I (Parm(16), I8(16)
C      EQUIVALENCE (BUFF, IBUFF), (A, IB)
C      LOGICAL NRAED, RAN DIN
C      IRP = 0
C      DO FILE SPACING
C      IF(NFS) 215,250,220
C      215 CONTINUE
C      IRP = 2
C      GO TO 1000

89
220 CONTINUE
DO 225 I=1,NFS
222 CONTINUE
BUFFER IN (INFILE,1) (BUFF(1),BUFF(IBFCNT))
221 CONTINUE
IF (UNIT,INFILE) 221,222,225
225 CONTINUE
230 CONTINUE
C
C DO MATRIX SPACING
C
IF (NNS) 235,250,240
239 CONTINUE
IRR = 1
GO TO 1000
240 CONTINUE
NHS = NHS + NHS
DO 245 I=1,NHS
BUFFER IN (INFILE,1) (BUFF(1),BUFF(IBFCNT))
241 CONTINUE
IF (UNIT,INFILE) 241,242,243
242 CONTINUE
GO TO 245
243 CONTINUE
IRR = 1500 + (I+1)/2
GO TO 1000
245 CONTINUE
250 CONTINUE
C
C READ B HEADER CARD
C
BUFFER IN (INFILE,1) (B(1),B(16))
300 CONTINUE
IF (UNIT,INFILE) 300,310,305
305 CONTINUE
IRR = 1500 + NHS +1
GO TO 1000
310 CONTINUE
C
C SET PARAMETERS AND SIZES
C
ISO) = IB(1)
M = IB(2)
N = IB(3)
MTN = IB(6)
DO 325 I=7,16
PARM(I-6) = B(I)
325 CONTINUE
C
C TEST FOR PROPER SIZES
C
IF (GTO.AND.M.GT.O.AND.MTN.LE.IBFCNT) GO TO 350
IRR = 4
GO TO 1000
350 CONTINUE
C
C READ THE ARRAY

B10
C
400 CONTINUE
BUFFER IN (INFILE,1) (BUFF(1),BUFF(MTH))
410 CONTINUE
IF (UNIT,INFILE) 410,420,415
415 CONTINUE
IRR = 1500 + MBS + 1
GO TO 1000
420 CONTINUE
C
IF K = 0 LEAVE THE MATRIX IN THE BUFF AREA AND EXIT
C
IF K.GT.0 TRANSFER BUFF TO ARRAY A
C
IF(K.LE.0) GO TO 1000
C
TRANSFORM BUFF TO ARRAY A
C
IF(TYPE.EQ.'HCOMPLEX) GO TO 475
IX = 0
DO 490 I=1,N
DO 490 J=1,N
IX = IX + 1
A(I,J) = BUFF(IX)
490 CONTINUE
GO TO 500
475 CONTINUE
K2 = K+K
CALL CBUFFR(A,K2,M,N,BUFF)
C
500 CONTINUE
C
1000 CONTINUE
RETURN
END
SUBROUTINE CBUFFR(A, K2, M, N, BUFF)
DIMENSION A(K2,1), BUFF(1)
C
C PUTS A COMPLEX ARRAY STORED IN BUFF INTO FORTRAN ARRAY A
C
IX = 0
IX2 = 1000
IN = M+1
DO 100 I=1, IN, 2
DO 100 J=1, N
IX = IX + 1
IX2 = IX2 + 1
A(I,J) = BUFF(IX)
A(I+1,J) = BUFF(IX2)
100 CONTINUE
RETURN
END
SUBROUTINE WRTENX(OUTFL, MKRIT, RANDOU, NFS, NMS, LS, NNR, LWS, WRTENX 00002
1 K; ID, A, ITYPE, N, N, PARM, IRR ) WRTENX 00003
WRTENX 00004
WRTENX 00005
WRTENX 00006
WRTENX 00007
WRTENX 00008
WRTENX 00009
WRTENX 00010
WRTENX 00011
WRTENX 00012
WRTENX 00013
WRTENX 00014
WRTENX 00015
WRTENX 00016
WRTENX 00017
WRTENX 00018
WRTENX 00019
WRTENX 00020
WRTENX 00021
WRTENX 00022
WRTENX 00023
WRTENX 00024
WRTENX 00025
WRTENX 00026
WRTENX 00027
WRTENX 00028
WRTENX 00029
WRTENX 00030
WRTENX 00031
WRTENX 00032
WRTENX 00033
WRTENX 00034
WRTENX 00035
WRTENX 00036
WRTENX 00037
WRTENX 00038
WRTENX 00039
WRTENX 00040
WRTENX 00041
WRTENX 00042
WRTENX 00043
WRTENX 00044
WRTENX 00045
WRTENX 00046
WRTENX 00047
WRTENX 00048
WRTENX 00049
WRTENX 00050
WRTENX 00051
WRTENX 00052
WRTENX 00053
WRTENX 00054
WRTENX 00055
WRTENX 00056
WRTENX 00057
WRTENX 00058

ROUTINE TO WRITE A MATRIX ON TAPE OR DISK FILE. WRTENX 00002
THIS VERSION WILL ONLY WORK WITH SEQUENTIAL FILES. WRTENX 00003
SOME VARIABLES ARE PASSED FOR RANDOM OPERATION BUT WRTENX 00004
ARE NOT CURRENTLY USED. WRTENX 00005

ICUTFL - TAPE NUMBER OR LEFT-JUSTIFIED FILE NAME WRTENX 00006
MKRIT - T. NAM FORMAT (NOT USED) WRTENX 00007
. F. TELQU FORMAT (NOT USED) WRTENX 00008
RANDOU - T. RANDOM FILE (NOT USED) WRTENX 00009
. F. SEQUENTIAL FILE (NOT USED) WRTENX 00010
NFS - NUMBER OF FILES TO SPACE (SEQ. ONLY) WRTENX 00011
NMS - NUMBER OF MATRICES TO SPACE WRTENX 00012
LS - LEVEL NUMBER TO SPACE (NOT USED) WRTENX 00013
NNR - IDENTIFIER (NAME OR NUMBER) (NOT USED) WRTENX 00014
LWS - LEVEL NUMBER OF THIS MATRIX (NOT USED) WRTENX 00015
N - ROW DIMENSION OF A WRTENX 00016
K - (IF 0, MATRIX IS ALREADY IN /BUFF/) WRTENX 00017
ID - ARRAY CONTAINING MATRIX NAME WRTENX 00018
A - ARRAY CONTAINING MATRIX WRTENX 00019
ITYPE - REAL,DIAG, NULL, MIXED, COMPLEX WRTENX 00020
N - ROW DIMENSION OF MATRIX WRTENX 00021
K - COLUMN DIMENSION OF MATRIX WRTENX 00022
PARM - 10 WORD PARAMETER ARRAY WRTENX 00023
IRR - ERROR RETURN WRTENX 00024
= 0, NO ERROR WRTENX 00025
1, MATRIX SPACING IS NEGATIVE WRTENX 00026
2, FILE SPACING IS NEGATIVE WRTENX 00027
3, NAM DIMENSIONS ARE .GT. IBGCNT WRTENX 00028
4, MATRIX IS ENCOUNTERED EOC AFTER MATRIX 1/2 WHILE WRTENX 00029
SKIPPING MATRICES. WRTENX 00030

DIMENSION ID(1), A(1), PARM(10), B(16) WRTENX 00031
DIMENSION IB(16) WRTENX 00032
EQUIVALENCE (B, IB) WRTENX 00033
LOGICAL MKRIT, RANDOU WRTENX 00034
COMMON /BUFF/ BFCODE, IBFNCNT, BUFF(3280) WRTENX 00035
DATA BFCODE, IBFNCNT /BFCNTSIZE, 3280 / WRTENX 00036

TEST FOR PROPER SIZE WRTENX 00037

IBIZ = 0
IF (ITYPE.EQ.3) IBIZ = ISIZ+ISIZ WRTENX 00038
IF (IBIZ.LE.IBGCNT) GO TO 205 WRTENX 00039
IRR = 4 WRTENX 00040
GO TO 1000 WRTENX 00041

205 CONTINUE WRTENX 00042
IF (K.LE.0) GO TO 300 WRTENX 00043
IF (K.GE.4) GO TO 210 WRTENX 00044
IRR = 5 WRTENX 00045
GO TO 1000 WRTENX 00046

B13
210 CONTINUE
C DO FILE SPACING
C IF(NFS) 215,230,220
215 CONTINUE
IRR = 2
GO TO 1000
220 CONTINUE
DO 225 I=1,NFS
222 CONTINUE
BUFFER IN (IOUTFL,1) (BUFF(1),BUFF(IBFCNT))
221 CONTINUE
IF(UNIT,IOUTFL) 221,222,225
225 CONTINUE
230 CONTINUE
C DO MATRIX SPACING
C IF(NMS) 235,250,240
235 CONTINUE
IRR = 1
GO TO 1000
240 CONTINUE
NMS = NMS + NMS
DO 245 I=1,NMS
BUFFER IN (IOUTFL,1) (BUFF(1),BUFF(IBFCNT))
241 CONTINUE
IF(UNIT,IOUTFL) 241,242,243
242 CONTINUE
GO TO 245
243 CONTINUE
IRR = 1500 + (I+1)/2
GO TO 1000
245 CONTINUE
250 CONTINUE
C CREATE B HEADER RECORD
C 300 CONTINUE
IB(1) = IDC(2)
IB(2) = N
IB(3) = N
B(4) = 0
IB(5) = 0
IB(6) = ISIZ
DO 325 I=7,16
B(I) = PARM(I-6)
325 CONTINUE
C IF(IX,LE,0) GO TO 400
C PUT ARRAY A INTO BUFFER
C IF((TYPE.EQ.,7)NCOMPLEX) GO TO 375
C NOT COMPLEX PUT INTO BUFFER.
C
IX = 0
DO 350 I=1,N
DO 350 J=1,N
IX = IX + 1
BUFF(IX) = A(I,J)
350 CONTINUE
GO TO 400
C
COMPLEX, CALL ROUTINE TO STORE INTO BUFFER.
C
375 CONTINUE
K2 = K+K
CALL COMBUF(A,K2,M,N,BUFF)
IX = 2***
C
400 CONTINUE
C
WRITE THE B HEADER RECORD AND THE BUFFER ARRAY RECORD
C
BUFFER CUT (ICUTFL,1) (B(I),B(16))
500 CONTINUE
IF (UNIT,ICUTFL) 500,510,510
510 CONTINUE
C
BUFFER CUT (ICUTFL,1) (BUFF(I),BUFF(IX))
520 CONTINUE
IF (UNIT,ICUTFL) 520,530,530
530 CONTINUE
C
1000 CONTINUE
RETURN
END
SUBROUTINE COBUF(A,K2,M,N,BUFF)
DIMENSION A(K2,I),BUFF(I)
C
C Puts complex array A into buffer BUFF
C
IX = 0
IX2 = MMN
MM = MMN-1
DO 100 I=1,MM,2
DO 100 J=1,N
IX = IX + 1
IX2 = IX2 + 1
BUFF(IX) = A(I,J)
BUFF(IX2) = A(I+1,J)
100 CONTINUE
RETURN
END
SET CONTROL PARAMETERS TO DEFAULT OPTIONS

DEFAULT = .FALSE.

100 CONTINUE

PRVUSE = .FALSE.
PRVMODE = .FALSE.
SYM = 1.0
WTPEW = 2
WTPEY = 2
NSURF = 1
DIHW = .TRUE.
DIHT = .TRUE.
ISIMPLW = 0
WGMAF = .TRUE.
WBL = .FALSE.
FPGMAF = .TRUE.
PRBL = .FALSE.
FRSL = .FALSE.
PRPAC = .FALSE.
PRSAIC = .FALSE.
PRCEOF = .FALSE.
PRMODS = .FALSE.
FRBOX = .FALSE.
PRDM = .FALSE.
PRSW = .FALSE.
PRDCP = .FALSE.
PRGMAC = .FALSE.
FRLW = .FALSE.
PRNM = .FALSE.
PRUM = .FALSE.
FRVP = .FALSE.
PRCM = .FALSE.
SUBDV = .FALSE.
EXAIC = .FALSE.
SMOOTH = .FALSE.
CRDFIT = .FALSE.
NDEG = 0
NRCREA = 0
PLYWOOD = .FALSE.
DPFCFR = .FALSE.
GEOCFR = .FALSE.
MODCFR = .FALSE.
AICCFR = .FALSE.
MBSCFR = .FALSE.
SMCPR = .FALSE.
CAFCPR = .FALSE.
PREVEX = EXEC

IF (DEFAULT) 400, 300

200 CONTINUE
OMACH = XMACH

300 CONTINUE
READ (MTS, 9005) TITLE

9005 FORMAT (BAI0)
READ (MTS, CARDB)
IF (XMACH.GT.1.0) GO TO 310
WRITE (MTS, 8005) XMACH
CALL FLUSH()
310 CONTINUE
    IF(XMACH.GE.1.2) GO TO 320
    WRITE (NT6,8010)
    GO TO 350
320 CONTINUE
    IF(XMACH.LE.3.0) GO TO 350
    WRITE (NT6,8015) XMACH
    CALL FLUSH(1)
340 CONTINUE
    WRITE (NT6,8020)
350 CONTINUE
8005 FORMAT(52H0** MACH NUMBER OF LESS THAN 1.0 CAN NOT BE USED.  
       1       14HMACH NUMBER = E15.6, 6H) 
8010 FORMAT(6CH0** WARNING -- MACH NUMBER LESS THAN 1.2 IS BEING USED.  
       1       3H) 
8015 FORMAT(43H0** MACH NUMBER GREATER THAN 5.0, XMACH = E15.6,  
       1       25H PROGRAM TERMINATED. 6H ) 
8020 FORMAT(64H0** WARNING -- MACH NUMBER GREATER THAN 3.0 IS BEING USE  
       16ED. 6H ) 
1ED. 6H READ (NTS,CARD) 
100 IF(SUBDV)500,510 
500 NSUBDV =3 
   GO TO 515 
510 NSUBDV =1 
515 CONTINUE
C
   IF(DEFAULT) 100,400 
C
400 CONTINUE
   QAIC = 0 
   NAIC = 0 
   CQAIC = 0 
   NSAIC = 0 
   INTAPE = 0 
   NCUTP = 1 
   INFSHEP = 0 
   ICURFSHEP = 0 
   READ (NTS,CARDD) 
C
   IF(QAIC.EQ.0) GO TO 520 
   NPLAIC = QAIC 
   OPLAIC = .TRUE. 
   GO TO 530 
520 CONTINUE
   NPLAIC = 0 
   OPLAIC = .FALSE. 
530 CONTINUE
   IF(NAIC.EQ.0) GO TO 540 
   NPLAIC = NAIC 
   OPLAIC = .FALSE. 
540 CONTINUE
C
Determine options of spatial kernels

B19
IF((ZAIC.ELZ.0) GO TO 560
M5PA1C = cVMIC
DATAPP 00152
M5PA1C = 0
QBAIC = .TRUE.
GO TO 570
560 CONTINUE
M5PA1C = 0
QBAIC = .FALSE.
570 CONTINUE
IF((ZAIC.ELZ.0) GO TO 560
M5PA1C = NSAIC
QBAIC = .FALSE.
580 CONTINUE
C
IF (NOUTP.NE.0) GO TO 660
IF(WGMNAF) WRITE (INT6,9041)
IF(WML ) WRITE (INT6,9042)
WGMNAF = .FALSE.
WML = .FALSE.
600 CONTINUE
C
CARD E
DO 610 I=1,20
W3(I) = -1.
W1(I) = -1.
610 CONTINUE
READ(INT5,CARD5)
DO 620 I=1,20
IF(W3(I).NE.-1.0 .OR. W1(I).NE.-1.0) GO TO 620
MVALUES = 1-1
GO TO 625
620 CONTINUE
MVALUES = 20
625 CONTINUE
C
WRITE (INT6,9520)
WRITE (INT6,9501)
WRITE (INT6,9551) TITLE
WRITE (INT6,9580) NAME
IF(DEFAULT) WRITE (INT6,9575)
IF(ISYM.EQ.1.0) WRITE (INT6,9552)
IF(ISYM.EQ.-1.0) WRITE (INT6,9553)
IF (PLYWOOD) WRITE (INT6,9554)
IF (PPPLAND.PLYWOOD) WRITE (INT6,9558)
IF(L.NOC.SYD) WRITE (INT6,9572)
IF(SUBYD) WRITE (INT6,9573)
IF (SUBYD .AND. NRMEA .NE. 0) WRITE (NT6,9546) NRMEA
IF(NSUBF.EQ.1) WRITE(NT6,9556)
IF(NSUBF.EQ.2) WRITE(NT6,9557)
IF(L.NOC.EXAIC) WRITE(NT6,9576)
IF(EXAIC) WRITE(NT6,9577)
IF(CRDFIT) SMOOTH = .FALSE.
IF(SMOOTH) WRITE (NT6,9581) NDEG
IF(CRDFIT) WRITE (NT6,9585) NDEG
IF (.NOT. (SMOOTH .OR. CRDFIT) .OR. NDEG .LE. 10) GO TO 630
NDEG = 10
WRITE (NT6,9043) NDEG
630 CONTINUE
IF(FRBOX) WRITE (NT6,9569)

B20
IF (PRCOEF) WRITE (NTS,9582)
IF (PRMODE) WRITE (NTS,9568)
IF (PRPAIC) WRITE (NTS,9585)
IF (PRSAIC) WRITE (NTS,9584)
IF (PRDM) WRITE (NTS,9570)
IF (PRSM) WRITE (NTS,9578)
IF (PRLM) WRITE (NTS,9544)
IF (PRAM) WRITE (NTS,9545)
IF (PRVP) WRITE (NTS,9571)
IF (PRBL) WRITE (NTS,9558)
IF (PRSL) WRITE (NTS,9566)
IF (PRCH) WRITE (NTS,9569)
IF (PRDCP) WRITE (NTS,9542)
IF (PRGNAC) WRITE (NTS,9543)
IF (PRGNAF) WRITE (NTS,9564)
IF (PRBL) WRITE (NTS,9562)
IF (PRGNAF) WRITE (NTS,9561)
IF (PRVECM) WRITE (NTS,9531)
IF (PRWODE) WRITE (NTS,9532)
IF (HYPEM,ED,1) WRITE (NTS,9533)
IF (HYPEM,ED,2) WRITE (NTS,9534)
IF (HYPEM,ED,3) WRITE (NTS,9535)
IF (NSURF,ED,1) GO TO 630
IF (HYPET,ED,1) WRITE (NTS,9536)
IF (HYPET,ED,2) WRITE (NTS,9537)
IF (HYPET,ED,3) WRITE (NTS,9538)
630 CONTINUE
IF (DIHM) WRITE (NTS,9539)
IF (NSURF,ED,1) GO TO 630
IF (DIHT) WRITE (NTS,9540)
630 CONTINUE
ERR = 0.01
IF (EXAIC) ERR = 0.0001
C
C THIS SET OF VARIABLES ARE DIMENSION SIZES FOR ARRAYS.
C THE NUMBER IS THE DIMENSION OF THE ARRAY.
C FOR DOUBLE DIMENSIONED ARRAYS IT IS THE LARGEST NUMBER.
C NOT THE PRODUCT OF THE TWO DIMENSIONS.

IIIERRL = 1
LIIERRL = 1640
LBXCDW = 190
LBXCDT = 90
LBXOC = 8
LJALPH = 200
LMTRM = 100
LMODES = 1000
LMTSBD = 30
LBDW = 800
LBDTDW = 100
LBDW = 1275
LTPV = 250
C
WRITE (NTS,9001)
WRITE (NTS,6002) QAIC,NAIC,QAIC,NSAIC,INTAPE,INFSP,NGUTP,LQFRSP
C
8001 FORMAT (INO/45X, 39H THE FOLLOWING TAPE SETUP IS REQUESTED - /)
FORMAT(51X,*OLD AIC TAPE =*,13,/*NEW AIC TAPE =*,13,/
1 51X,*OLD SPATIAL AIC TAPE =*,13,/*NEW SPATIAL AIC TAPE =*,13,
2 / 51X,*INPUT DATA TAPE =*,13,* SPACED*,13,* FILES,*
3 / 51X,*OUTPUT TAPE =*,13,* SPACED*,13,* FILES,* /*)

PRINT THE NWVAL OR WKS ARRAY.

IF(W@l(1).E.100) GOTO 700
WRITE(NT5,6003)
WRITE(NT5,6004) (WX(1),I=1,NWVALS)
GO TO 900

700 CONTINUE
IF(WX(1),EQ.-1.0) GOTO 800
WRITE(NT5,6005)
WRITE(NT5,6004) (WX(1),I=1,NWVALS)
GO TO 900

800 CONTINUE
WRITE(NT5,6006)

900 CONTINUE
WRITE(NT5,6006)

6003 FORMAT(1H0,29X, *THE FOLLOWING IS THE REDUCED FREQUENCY ARRAY BASE
1D ON BOX LENGTH* /)
6005 FORMAT(1H0,29X, *THE FOLLOWING IS THE REDUCED FREQUENCY ARRAY BASE
1D ON WING SEMI-SPAN* /)
6004 FORMAT(1H / (31X,6F11.5)
6006 FORMAT(49H0*** WARNING -- NO REDUCED FREQUENCIES SPECIFIED.
1 5IH PROGRAM WILL TERMINATE AFTER GEOMETRY SECTION *** )
C
C
C
1000 RETURN

9900 FORMAT(1H0,29X,5X, *FOR DEPARTMENT OF THE AIR FORCE*19X,1H0, /)
1 30X,1H0,10X,*AERODYNAMICAL SYSTEMS DIVISION* 17X,1H0, / DATAPP 00288
2 30X,1H0,10X,*AIR FORCE FLIGHT DYNAMICS LABORATORY*10X,1H0, / DATAPP 00289
3 30X,1H0,10X,*WRIGHT-PATTERSON AIR FORCE BASE* 15X,1H0, / DATAPP 00290
4 30X,1H0,56X,1H0, / DATAPP 00291
5 30X,1H0,5X,*BY THE BOEING COMPANY* 28X,1H0, / DATAPP 00292
6 30X,1H0,10X,*COMMERCIAL AIRLINE DIVISION* 18X,1H0, / DATAPP 00293
7 30X,1H0,10X,*SEATTLE, WASHINGTON* 27X,1H0, / DATAPP 00294
8 30X,1H0,56X,1H0, / 30X,58 (1HE), / DATAPP 00295

9041 FORMAT(1H0,29X,5X, *WARNING -- NO OUTPUT TAPE WAS REQUESTED FOR GENERAL
1 112ED FORCES. *** )
9042 FORMAT(1H0,29X,5X, *WARNING -- NO OUTPUT TAPE WAS REQUESTED FOR BOX LIF
1 112ED )
9043 FORMAT(1H0,29X,5X, *WARNING -- ORDER FOR VELOCITY POTENTIAL SMOOTHING
1 36H TOO LARGE. IT HAS BEEN REDUCED TO 12, 4H ***
9551 FORMAT(1H0,5X, *TITLE -13X,6A10,13X,1H0, TITLE /1H0/45X,
1 37H THE FOLLOWING OPTIONS ARE REQUESTED - / )
9531 FORMAT(1H0,5X, *GEOMETRY FROM PREVIOUS CYCLE* )
9532 FORMAT(1H0,5X, *CONE SHAPES FROM PREVIOUS CYCLE* )
9533 FORMAT(51X,**LOCAL INPUT FOR WING IS POLYNOMIAL COEFFICIENTS** )  DATAPP 00323
9534 FORMAT(51X,**LOCAL INPUT FOR WING IS ARBITRARY LOCATIONS FOR SURFAC** DATAPP 00324
1E Fitting **) DATAPP 00325
9535 FORMAT(51X,**LOCAL INPUT FOR WING IS BOX CENTER VALUES** )  DATAPP 00326
9536 FORMAT(51X,**LOCAL INPUT FOR TAIL IS POLYNOMIAL COEFFICIENTS** )  DATAPP 00327
9537 FORMAT(51X,**LOCAL INPUT FOR TAIL IS ARBITRARY LOCATIONS FOR SURFAC** DATAPP 00328
1E Fitting **) DATAPP 00329
9538 FORMAT(51X,**LOCAL INPUT FOR TAIL IS BOX CENTER VALUES** )  DATAPP 00330
9539 FORMAT(51X,**DIHEDRAL WING INFLUENCE CALCULATED** )  DATAPP 00331
9540 FORMAT(51X,**DIHEDRAL TAIL INFLUENCE CALCULATED** )  DATAPP 00332
9542 FORMAT(51X,**PRINT PRESSURE DIFFERENCE COEFFICIENTS** )  DATAPP 00333
9543 FORMAT(51X,**PRINT GENERALIZED AERODYNAMIC COEFFICIENTS** )  DATAPP 00334
9544 FORMAT(51X,**PRINT LONGITUDINAL WASHES ALONG SAMPLING CHORDS** )  DATAPP 00335
9545 FORMAT(51X,**PRINT NORMAL WASHES** )  DATAPP 00336
9546 FORMAT(51X,**EFFECTIVE SUBDIVIDED AREA OF R.,13.,# ROWS REQUESTED** )  DATAPP 00337
9542 FORMAT(51X,**SYMMETRIC ANALYSIS** )  DATAPP 00338
9553 FORMAT(51X,**ANTI-SYMMETRIC ANALYSIS** )  DATAPP 00339
9554 FORMAT(51X,**FLYWOOD OPTION IS USED. (PLANFORM BOUNDARY DETERMINED** DATAPP 00340
1 BY BOX PATTERN.) **) DATAPP 00341
9556 FORMAT(51X,**SINGLE PLANFORM ANALYSIS** )  DATAPP 00342
9557 FORMAT(51X,**ANALYSIS FOR 2 PLANFORMS** )  DATAPP 00343
9558 FORMAT(1HD,100(1HS)/// THE SPURCE GOCSE IS LOOSE 0 1H0, 100(1HS)**) DATAPP 00344
9561 FORMAT(51X,**WRITE GENERALIZED AIR FORCES ON TAPE** )  DATAPP 00346
9562 FORMAT(51X,**WRITE BOX LIFTS ON TAPE** )  DATAPP 00347
9564 FORMAT(51X,**PRINT GENERALIZED AIR FORCES** )  DATAPP 00348
9565 FORMAT(51X,**PRINT THE BOX LIFTS** )  DATAPP 00349
9566 FORMAT(51X,**PRINT THE SECTION LIFTS** )  DATAPP 00350
9567 FORMAT(51X,**SECTION MOMENTS WILL BE COMPUTED WITH MODE SHAPE ONE** BCSFRB 00006
1 51X,**ASSUMED FOR THE PITCH MODE,** ) BCSFRB 00007
9568 FORMAT(51X,**PRINT MODE SHAPE USED** )  DATAPP 00351
9569 FORMAT(51X,**PRINT THE BOX PATTERN** )  DATAPP 00352
9570 FORMAT(51X,**PRINT THE UPLIFTS ALONG SAMPLING CHORDS** )  DATAPP 00353
9571 FORMAT(51X,**PRINT THE VELOCITY POTENTIALS** )  DATAPP 00354
9572 FORMAT(51X,**PRINT THE VELOCITY POTENTIALS** )  DATAPP 00355
9573 FORMAT(51X,**SUBDIVISION WILL BE APPLIED** )  DATAPP 00356
9575 FORMAT(51X,**ALL PARAMETERS SET TO "FAULT VALUES"** )  DATAPP 00357
9576 FORMAT(51X,**APPROXIMATE KERNELS WILL BE USED** )  DATAPP 00358
9577 FORMAT(51X,**EXACT KERNELS WILL BE USED** )  DATAPP 00359
9578 FORMAT(51X,**PRINT THE SIDESLIPS ALONG SAMPLING CHORDS** )  DATAPP 00360
9580 FORMAT(51X,**MOM (NUMBER = **, FB.6** )  DATAPP 00361
9581 FORMAT(51X,**VELOCITY POTENTIALS WILL BE SMOOTHED BY A LEAST-SQUARES** DATAPP 00362
1 **EJS** / 61X,**POLYNOMIAL SURFACE FIT, OF ORDER**,12,1H,**/ DATAPP 00363
2 61X,**Q0 = PROGRAM DETERMINED,**) **; DATAPP 00364
9582 FORMAT(51X,**PRINT MODE SHAPE POLYNOMIAL COEFFICIENTS, IF AVAILAB** DATAPP 00365
1 **LE **) DATAPP 00366
9583 FORMAT(51X,**PRINT THE PLANAR AIC ARRAYS USED**) DATAPP 00367
9584 FORMAT(51X,**PRINT THE SPATIAL AIC ARRAYS USED**) DATAPP 00368
9585 FORMAT(51X,**VELOCITY POTENTIALS WILL BE SMOOTHED BY A LEAST SQUARES** DATAPP 00369
1 **EJS** / 61X,**POLYNOMIAL CHORDWISE FIT, OF ORDER**,12,1H,**/ DATAPP 00370
2 61X,**Q0 = PROGRAM DETERMINED,**) **; DATAPP 00371
END
PROGRAM GEOMX

C THIS OVERLAY READS ALL GEOMETRIC INFORMATION (CARDS G TO L, OR FROM TAPE) AND COMPUTES THE INTERNAL GEOMETRY NEEDED.
C ERRORS IN GEOMETRIC DEFINITIONS ARE CAUGHT.
C ALL GEOMETRY IS NON-DIMENSIONALIZED BY BOX WIDTH (LENGTH).
C BOX CODES ARE DEFINED -
C 0 = NOT USED
C 1 = ON-P-LANIFORM
C 2 = DIAPHRAGM
C 3 = NAKE

C ALPHA ARRAY, FRACTIONAL PART OF EDGE BOXES, IS COMPUTED.
C MAXIMUM PLANAR AIC ARRAY SIZE IS DETERMINED.
C FOR EACH CHORD REQUIRING A SPATIAL AIC ARRAY, DETERMINE
C WHICH AIC ARRAY TO USE (KPTW, KPTTW, KPTWLT).
C EL, THE VERTICAL DISTANCE SEPARATING THE SURFACES
C YBAR, THE HORIZONTAL OFFSET
C MUSIC ARRAY, A MAP OF NEEDED AIC VALUES

COMMON /CTRLY/ PREVEX, CMACH, TITLE(8), PRGVECH, PRVMODE, DIHM, DIHT, Default
1                                       Default
1                                       Default
COMMON /PRGWAY/ XMAC, NMODES, NTSLP, NKVAI, SMOOTH, NDEG, CRDFIT,
1                                      EXAI, SUBDY, PLYWOOD
COMMON /KVAL/ KVAL(20), YKS(20)
COMMON /GEOMY/ COPLAN, SUBDY, XSUBDY, NSUB2, NSUBCN, NSURF,
1                                      B1, BIBETA, B2, BIBTAS, WAX, WAZ, PSIW,
2                                      MXBW, MYBBW, MBW, MBBW, MBBSW, MYBBW,
3                                      IXBW, XCENTR
COMMON /COPLAN/ TLA, TLZ, PSIT, MBXT, MBYT, MBBT, MBST, MBST,
1                                      MBST, IXBT, IXBT, CARL
COMMON /KERN/ ERR, MSKER, PKERN, PULKN, NSPAK, NROMA
COMMON /FILES/ NT5, NT6, INTAPE, INFSP, NPLAI, NSPAIC, NOUTP,
1                                      IOUTSP, MODESC, IVPSF, IGEOS, IWFSC, IAIASC
COMMON /IOCONT/ OPLAI, OSPAI, WTGCOM, WTGNAF, WTSI, WTLB, PRBOX,
1                                      PRAIC, PRSAIC, PRMCD, PRCEOF, PRDM, PRSW, PRVP,
2                                      PRBL, PRDCP, PRGNAF, PRGAC, PRSL, PRML, PRMN, PRCH
EQUIVALENCE (PRUM, PRDW)
LOGICAL OPLAI, OSPAI, WTGCOM, WTGNAF, WTSI, WTLB, PRBOX, PRPAIC,
1                                      PRSAIC, PRMCD, PRCEOF, PRDM, PRSW, PRVP, PRBL, PRDCP, PRGNAF,
2                                      PRBL, PRDCP, PRGNAF, PRGAC, PRSL, PRML, PRMN, PRCH
COMMON /TAPEOF/ NFS, NNS, LS, NRID(20), NID, NTYP, LRS, LUN, N, N,
1                                      PARM(10), IIR
DIMENSION IPARM(10)
EQUIVALENCE (IPARM, IPARM)
COMMON /MODES/ SYM, SYMT, NTYPEW, NTTYPE:
COMMON /ARRAYS/ KBXCDM, LBXCDM, LBXCN, LBXCDT, KJALPH, LJALPH,
1                                      KALPHA, LKERN, LKERL, LPNTRM, LPNTM, KDEFSL, KELPH
2                                      LWODES, KPPNTD, LPNTSD, KSND, LSND, KPNTDW, LPNTDW,
3                                      KNW, LOW, KTVP, LTVP
COMMON /SAMPLW/ TSMPL, TICGRD(10), IBORX(10), IBORY(10), ZLOC(10)
COMMON /MUSIC/ YBAR, EL, MUSIC(2, 50), NROMS, SURF,
1                                      YBARL, ELR, MUSICL(2, 50), NROMSL, SURF, PSDIF
LOGICAL SURF, SURFL
COMMON /EDGES/ FEXLOC(250), TEXLOC(250), JDIAG

OVERLAY (AFMBOX.1,2)
COMMON /PLANKY/ XMLE,NXTE,MTLE,MTTE, XMLE(10),YMLE(10), PLANKY 00002
1 XMLE(10),YMLE(10), X'LE(10),Y'LE(10), PLANKY 00003
2 XTTE(10),YTTE(10), PLANKY 00004
LOGICAL XMWRITE,RANDOU,MREAD,RANDIN
COMMON /CHECKFR/ DPFCPR,GECCPR,MODCPR,ACCPR,NASCPR,SMCPPR,GAFCPR
LOGICAL DPFCPR, GECCPR, MODCPR, ACCPR, NASCPR, SMCPPR, GAFCPR
EQUIVALENCE (CHECKFR,GECCPR)
LOGICAL CHECKFR
C
C DIMENSION IBOW(150,8),IBOWT(90,8), GEOMX 00026
1 IMAKE(160), ICODE(160) GEOMX 00027
DIMENSION KPTNW(50),KPTTT(50),KPTLWT(50),KPTLWT(50) GEOMX 00028
DIMENSION ALPHA(200), JALPH(200) GEOMX 00029
DIMENSION KPT(4,50) GEOMX 00030
EQUIVALENCE (KPT,ALPHA) GEOMX 00031
DATA XMWRITE,RANDOU,MREAD,RANDIN /4..F. / FTNX1
DATA NBWD /2D/ Y'TNX1
DATA INIT,XINIT /377671l,3765432177777777777/ GEOMX 00032
DATA EPS /1.0E-4/ GEOMX 00033
C NAMELIST PARAMETERS FOR CARDS TO BE READ IN THIS SECTION
NAMELIST /CARDF/ WLAX,WLAZ,PSIW, TLAX,TLAZ, PSIT, CHECKFR
1 /CARDF/ NOHRS, XCENTR, XEDGE, IEDGE,IBOXF,IBOXL,ZLOC
C CARDS NMLE, NITE, NTLE, NTTE (415)
C CARDS I TO L (6E10.0)
C
5001 FORMAT(415) FTNX1 00001
5002 FORMAT(6E10.0) FTNX1 00002
C
I$CHDS = LBOXC * 20 GEOMX 00050
NSUBD = NSUBD GEOMX 00051
NSUBDZ = NSUBD*2 GEOMX 00052
NSUBCN = NSUBCN + 1 GEOMX 00053
HALFBC = XSUBD*2.0 GEOMX 00054
C
IS PREVIOUS GEOMETRY TO BE USED -
IF (.NOT. FUGEOG) GO TO 15 GEOMX 00057
C YES. HAS THE MACH NUMBER CHANGED -
IF (I$MACH .EQ. OMACH) GO TO 2000 GEOMX 00058
C YES. SKIP THE GEOMETRY READS, BUT RECO THE REST FOR THE
C NEW BOX PATTERN
BIO = 31 GEOMX 00073
XCENTRO = XCENTR GEOMX 00074
GO TO 272 GEOMX 00075
C READ CARDS F AND G
15 CONTINUE GEOMX 00077
WLAZ = 0. GEOMX 00078
WLAZ = 0. GEOMX 00079
PSIW = 0. GEOMX 00080
TLAX = 0. GEOMX 00081
TLAZ = 0. GEOMX 00082
PSIT = 0. GEOMX 00083
READ (MT5,CARDF) GEOMX 00084
WRITE (MT5,6010) WLAX,WLAZ,PSIW GEOMX 00085
IF (NSURF .EQ. 2) WRITE (NT6,6012) TLAX,TLAZ,PSIT
DEGREE = .01745329251943
C CONVIR DEGREES TO RADIANS
IVAL = 4*PSIW
IF (PSIW .GT. 45 .OR. PSIW .LT. -45.) GO TO 8030
PSIW = PSIW * DEGREE
IF (NSURF .EQ. 1) GO TO 30
IVAL = 4*PSIW
C SPECIAL CHECK FOR VERTICAL TAIL
IF (PSIT .EQ. 90 .AND. SYM .LE. 0) GO TO 25
C SYMT = SYMامية FOR THE TAIL, IDENTICAL TO
C THE WING EXCEPT FOR A VERTICAL TAIL
IVAL = 4*PSIT
CALL CLEAR
IVAL = 0
IVAL = 4*PSIT
C 30 CONTINUE
NCHRS = INIT
XCENTR = XINIT
XEDGE = XINIT
DO 30 I=1,10
ICORD(I) = INIT
IBOKF(I) = INIT
IBOKL(I) = INIT
30 ZLOC(I) = XINIT
READ(NT5,CARDG)
C CHECK AND PRINT PARAMETERS READ
WRITE (NT6,6015) NCHRS,XCENTR,XEDGE
C IVAL = 6*NCHRS
IF (NCHRS .EQ. INIT) GO TO 8010
IF (NCHRS .LE. 0 .OR. NCHRS .GE. LSCH/NSUBDV) GO TO 8015
MBW = NCHRS
IVAL = 12*XCENTR
IF (XCENTR .EQ. XINIT) GO TO 120
C USE XCENTR DIRECTLY, IGNORE XEDGE
IF (XEDGE .EQ. XINIT) WRITE (NT6,9010)
GO TO 125
C GET XCENTR FROM XEDGE
120 CONTINUE
IF (XEDGE .EQ. XINIT) GO TO 8020
125 CONTINUE
IF (ISFMLW .EQ. 0) GO TO 200
IF (NSURF .EQ. 2) GO TO 170
DO 150 I = 1,ISMFLW
IF (ICORD(I) .GT. MBW .OR. ICORD(I) .LE. 0) GO TO 130
IF (IBOKF(I) .EQ. INIT .OR. IBOKL(I) .EQ. INIT) GO TO 140
IF (IBOKF(I) .LT. 1 .OR. IBOKL(I) .LT. LBXEDV/NSUBDV) GO TO 130
IF (IBOKF(I) .LE. IBOKL(I) ) GO TO 140
130 WRITE (NT6,9020) I
ISFMLW = I - 1
GO TO 150

B26
CONTINUE

IF (ZLOC(I) .EQ. UINIT) ZLOC(I) = 0.
CONTINUE

WRITE (NTS,6017) ISHPLW, (ICHORD(I), ISBOXF(I), ISBOXL(I), ZLOC(I), I = 1, ISHPLW)
GO TO 200

CONTINUE

GEOex 00144
GEOex 00145
GEOex 00146
GEOex 00147
GEOex 00148
GEOex 00149
GEOex 00150
GEOex 00151
GEOex 00152
GEOex 00153
GEOex 00154
GEOex 00155
GEOex 00156
GEOex 00157
GEOex 00158
GEOex 00159
GEOex 00160
GEOex 00161
GEOex 00162
GEOex 00163
GEOex 00164
GEOex 00165
GEOex 00166
GEOex 00167
GEOex 00168
GEOex 00169
GEOex 00170
GEOex 00171
GEOex 00172
GEOex 00173
GEOex 00174
GEOex 00175
GEOex 00176
GEOex 00177
GEOex 00178
GEOex 00179
GEOex 00180
GEOex 00181
GEOex 00182
GEOex 00183
GEOex 00184
GEOex 00185
GEOex 00186
GEOex 00187
GEOex 00188
GEOex 00189
GEOex 00190
GEOex 00191
GEOex 00192
GEOex 00193
GEOex 00194
GEOex 00195
GEOex 00196
GEOex 00197
GEOex 00198
GEOex 00199
GEOex 00200

GEOex 00190
GEOex 00191
GEOex 00192
GEOex 00193
GEOex 00194
GEOex 00195
GEOex 00196
GEOex 00197
GEOex 00198
GEOex 00199
GEOex 00200

140 CONTINUE

C I F ( ZLOC ( I ) .EQ. XINIT ) ZLOC ( I ) = 0 . C
CONTINUE

130 CONTINUE

160 CONTINUE

WRITE (NTS,6017) ISHPLW, (ICHORD(I), ISBOXF(I), ISBOXL(I), ZLOC(I), I = 1, ISHPLW)

CONTINUE

GO TO 200

C SAMPLING OF SCAW washES ILEGAL IF TAI L DEFINED

170 CONTINUE

WRITE (NTS,6030) ISHPLW
ISHPLW = 0

200 CONTINUE

C OBTAIN THE LEADING AND TRAILING EDGE VALUES

C CARD INPUT OF PLANES IS REQUIRED

READ (NT5,5001) XALE,NATE,NTLE,NITE
WRITE(NT6,621) NLE,NTITE
GO TO (214,212),NSURF
212 WRITE(NT6,622) NITE,NITE
IVAL = 4HNTLE
IF (NITE .LT. 2 .OR. NITE .GT. 10) GO TO 8030
IVAL = 4HTAIL
IF (NTLE .LT. 2 .OR. NTLE .GT. 10) GO TO 8030
IVAL = 4HNTLE
IF (NITE .LT. 2 .OR. NITE .GT. 10) GO TO 8030
IVAL = 4HTAIL
C GET THE LEADING AND TRAILING EDGE VALUES
C CARD INPUT OF PLANES IS REQUIRED
READ (NT5,5002) (XWE(I),YWE(I),I=1,NWE)
WRITE (NT6,6030) IVAL, (XWE(I),YWE(I),I=1,NWE)
CALL EDGHK(XALE,YALE,NTLE,1,IRR)
IF (IRR .NE. 0) GO TO 8030
IVAL = 4HNTLE
READ (NT5,5002) (XTLE(I),YTEL(I),I=1,NTLE)
WRITE (NT6,6030) IVAL, (XTLE(I),YTEL(I),I=1,NTLE)
CALL EDGHK(XTLE,YTEL,NTLE,2,IRR)
IF (IRR .NE. 0) GO TO 8030
IVAL = 4HTAIL
READ (NT5,5002) (XTTE(I),YTTE(I),I=1,NITE)
WRITE (NT6,6030) IVAL, (XTTE(I),YTTE(I),I=1,NITE)
CALL EDGHK(XTTE,YTTE,NITE,2,IRR)
IF (IRR .NE. 0) GO TO 8030
GO TO 270

C CARDS I AND J - WING DEFINITION POINTS

WRITE (NT6,6029)
IVAL = 5HNG L.E.
READ (NT5,5002) (XMLE(I),YMLE(I),I=1,NMLE)
WRITE (NT6,6030) IVAL, (XMLE(I),YMLE(I),I=1,NMLE)
CALL EDGHK(XMLE,YMLE,NMLE,1,IRR)
IF (IRR .NE. 0) GO TO 8030
IVAL = 5HNTLE
READ (NT5,5002) (XMLE(I),YMLE(I),I=1,NMLE)
WRITE (NT6,6030) IVAL, (XMLE(I),YMLE(I),I=1,NMLE)
CALL EDGHK(XMLE,YMLE,NMLE,2,IRR)
IF (IRR .NE. 0) GO TO 8030
IVAL = 5HTAIL
READ (NT5,5002) (XMLE(I),YMLE(I),I=1,NMLE)
WRITE (NT6,6030) IVAL, (XMLE(I),YMLE(I),I=1,NMLE)
CALL EDGHK(XMLE,YMLE,NMLE,2,IRR)
IF (IRR .NE. 0) GO TO 8030
GO TO 270

C CARDS K AND L - TAIL DEFINITION POINTS

IVAL = 5HNTLE
READ (NT5,5002) (XTLE(I),YTEL(I),I=1,NTLE)
WRITE (NT6,6030) IVAL, (XTLE(I),YTEL(I),I=1,NTLE)
CALL EDGHK(XTLE,YTEL,NTLE,1,IRR)
IF (IRR .NE. 0) GO TO 8030
IVAL = 5HTAIL
READ (NT5,5002) (XTTE(I),YTTE(I),I=1,NITE)
WRITE (NT6,6030) IVAL, (XTTE(I),YTTE(I),I=1,NITE)
CALL EDGHK(XTTE,YTTE,NITE,2,IRR)
IF (IRR .NE. 0) GO TO 8030
GO TO 270

B27
C  PLAN FORM DEFINITIONS TO BE READ FROM TAPE

C

270 CONTINUE

BIBETA = YALE(NALE)/NYBW

272 CONTINUE

B1 = BIBETA * SORT(YMAX*XMAX-1.0)
BIBETA2 = BIBETA* 0.5
B12 = B1 + 0.5
IF (NSUBDV .NE. 1) GO TO 275
BIS = B1
BIBTAS = BIBETA
GO TO 280

275 BIS = B1/XSUBDV
BIBTAS = BIBETA/XSUBDV

280 CONTINUE

WRITE (INT6,0040) B1,BIBETA

C  SET THE WVAL ARRAY IF WKS WAS INPUT

C

IF(WKS(1),EQ.,-1.0) GO TO 295
DO 290 I=1,WVALS
WVAL(I) = WKS(I) * (B1/YALE(NALE))
290 CONTINUE

295 CONTINUE

C  DETERMINE THE GLOBAL COORDINATE LOCATION OF THE FIRST UN-

C  SUBDIVIDED PLAN FORM BOX CENTER, XCENTR

IVAL = 0*XCENTR
IF (XCENTR .LE. XINIT) XCENTR = XEDGE + B12
XEDGE = YALE(1) + (YALE(2)-YALE(1)) + BIBTAS / YALE(2)
IF (PRVGEOM) XEDGE = B10 + XCENTR +
10 B10 + (YALE(2) - YALE(1)) * 0.5 / (YALE(2) - .5)
IF (XCENTR-XEDGE) 310,330,350
310 DO 315 I = 1,51
XCENTR = XCENTR + B1
IF (XCENTR .GE. XEDGE) GO TO 330
315 CONTINUE

GO TO 300

320 DO 325 I = 1,51
IF (XCENTR-B1 .LT. XEDGE) GO TO 330
XCENTR = XCENTR - B1
325 CONTINUE

GO TO 300

330 CONTINUE

C  IN PREVIOUS GEOMETRY BEGIN USED -

C

C  YES. CONVERT X-COORDINATE VALUES TO NEW BOX LENGTH

PSIDIF = PSIT - PSIW
SLIDE = -B10 + XCENTR - XCENTR
DO 335 I = 1,NALE
335 XALE(I) = (B10+XALE(I) + SLIDE)/B1 + 1.0
DO 340 I = 1,NATE
340 XATE(I) = (B10+XATE(I) + SLIDE)/B1 + 1.0
IF (NSURF ,EQ. 1) GO TO 390
DO 345 I = 1,NATE
345 CONTINUE

R28
345 XITLE(I) = (BIOWXITLE(I) + SLIAXE)/BI + 1.0
DO 350 I = 1, NTTE
350 XITLE(I) = (BIOWXITLE(I) + SLIAXE)/BI + 1.0
GO TO 390
C
C CONVERSION OF THE GEOMETRIC INFORMATION TO THE NON-DIMENSIONAL
C
C
355 CONTINUE
DO 360 I = 1, NTTE
YMLE(I) = (YMLE(I) - XCENTR)/BI + 1.0
360 YMLE(I) = YMLE(I)/BIBETA + 0.5
DO 365 I = 1, NTTE
YME(I) = (YME(I) - XCENTR)/BI + 1.0
365 YME(I) = YME(I)/BIBETA + 0.5
GO TO (370, 375), NSURF
370 CAPL = 0.
PBST = 0.
PRIDIF = -PS1W
NBST = 0
WBBST = 0
IF (SIMPLW .EQ. 0) GO TO 390
C TRANSFORM ZLOC FOR THE SAMPLE WASH CHORDS TO A NON-DIMENSIONAL
C
C
380 CONTINUE
DO 372 I = 1, NSURF
ZLOC(I) = (ZLOC(I) - ULAZ) /RIBETA
372 CONTINUE
GO TO 390
375 XDIF = WLAX + XCENTR - TLAX
DO 380 I = 1, NTTE
XITLE(I) = (XITLE(I) - XDIF)/BI + 1.0
380 YITLE(I) = YITLE(I)/BIBETA + 0.5
DO 385 I = 1, NTTE
YME(I) = (YME(I) - XDIF)/BI + 1.0
385 YME(I) = YME(I)/BIBETA + 0.5
CAPL = (TLAZ - WLAX)/BIBETA
PRIDIF = PBST - PS1W
C
C CHECK FOR TAIL CROSSING WING
IF (PRIDIF) 386, 389, 387
386 IF (CAPL .LE. 0) GO TO 399
GO TO 388
387 IF (CAPL .GE. 0) GO TO 389
388 YCROSS = CAPL/(SIN(PS1W) - SIN(PS1T)) + .5
IF (YMLE(NMLE) - COS(PS1W) .LT. YCROSS) GO TO 390
IF (YMLE(KMLE) - COS(PS1T) .GE. YCROSS) GO TO 8080
GO TO 380
389 YCROSS = .5
C
C ZERO OUT THE BOX CODE ARRAYS
390 CONTINUE
DO 430 J = 1, LBXOC
430 IBXOC(J,J) = 0
DO 430 I = 1, LBXOC
430 IBXOC(I,J) = 0
GO TO 390
C
C
B29
过程:

1. 初始化子程序
   - 在直角坐标系中，以 origin 为原点，正方向为 x, y, z 轴。
   - 定义变量和数组，包括坐标、长度、角度、方向等。

2. 计算交叉点
   - 使用几何方法计算两点间或线段间的交点。

3. 计算距离
   - 使用距离公式计算两点或直线段间的距离。

4. 计算角度
   - 使用余弦定理或正弦定理计算角度。

5. 计算面积
   - 使用几何公式计算平面或表面的面积。

6. 计算体积
   - 使用几何公式计算三维空间的体积。

7. 计算面积和体积的综合
   - 结合面积和体积的概念，计算综合的几何量。

8. 应用实例
   - 提供具体的应用实例，说明如何在实际工程中应用这些计算方法。

B30
IXBT = XSUBDIV*(XMIN-1.0) + IXBW + 1
IF (AINT(XMIN) .EQ. XMIN) IXST = IXBT - 1
XMIN = XYLE(1) - (1.0 - YTLE(1)) * DELE
IXBT = XMIN
IF (FLOAT(IXBT) .EQ. XMIN) IXBT = IXST - 1
IXBT = NSUBDV * IXBT + IXBW

I)BST = LOCATION OF FIRST SUBDIVIDED TAIL BOX
IXST = LOCATION OF FIRST UNSUBDIVIDED TAIL BOX CENTER
ISUBT = 2 - IXBT
ISUBT = THE SUBSCRIPT FOR ARRAY IBOX WHICH WILL KEEP TAIL BOXES WITHIN THE BOUNDARIES OF IBOX
IF (FLCAT(IXBT) .EQ. XST) IXBT = IXST - 1
CEOMBX 00376
IXET = NSDV * IXST + IXBW

C 315 CONTINUE
C
C COPLAN = .F.
C
C DETERMINE THE BOX CODES FOR THE SECOND PLANIFORM
CALL BXCPF(XTLE,YTLE,XTLE,YTLE,XTLE,YTLE,LBXCDT,IBOKT(ISUBT,1))
C
C RETURNS - IBOX, CODES 2 AND 3 ADDED FOR DIAPHRAGM AND WAKE
C
WBST = NUMBER OF SUBDIVIDED CHORDS TO END OF TAIL
WST = NUMBER OF SUBDIVIDED CHORDS ON TAIL
NST = NUMBER OF SUBDIVIDED CHORDS, BOTH PLANIFORMS
WST = NUMBER OF SUBDIVIDED CHORDS ON TAIL
NST = NUMBER OF UNSUBDIVIDED CHORDS, BOTH PLANIFORMS
FEDLOC = LEADING EDGE LOCATIONS, BOTH PLANIFORMS
TEDLOC = TRAILING EDGE LOCATIONS

GET DIAPHRAGM VALUES FOR THE TAIL
WST = WBST
IF (.NOT. (CHECKFR)) GO TO 515
CALL PRNTBC(IBOKT(ISUBT,1),LBXCDT,IXBT,WXST,WSST,..)
II = WSST + 1
III = WBSW + WST
WRITE (NT6,7040) (FEXLOC(I), I = II,III)
WRITE (NT6,7045) (TEXLOC(I), I = II,III)

910 CONTINUE
INK = 0
CALL BXCDI ( INK, LBXCDT,LSCHD, IBOX(ISUBT,1))
C
C RETURNS - IBOX, CODES 2 AND 3 ADDED FOR DIAPHRAGM AND WAKE
C
WST = NUMBER OF SUBDIVIDED CHORDS, INCLUDING
C DIAPHRAGM FOR TAIL
WST = NUMBER OF UNSUBDIVIDED CHORDS

IF (.NOT. (FRBOX .OR. CHECKFR)) GO TO 520
CALL PRNTBC(IBOKT(ISUBT,1),LBXCDT,IXBT,WSST,WSST,..)
IF (NSUBDV .EQ. 1) GO TO 520
IFR = (IXBT - IXST) / NSUBDV + 1
CALL PRNTBC(IBOKT(ISUBT,1),LBXCDT,IFR, WSST,WSST,..)

520 CONTINUE
C
C THE FOLLOWING LOOP DETERMINES THE LOCUS OF MAXIMUM AFTWARD
C PROJECTIONS OF THE INTERSECTIONS OF THE TAIL MACH CONES
C WITH THE WING PLANE (EXTENDED), MACH CONES FOR UNSUBDIVIDED
C TAIL CHORDS ARE USED, BUT ALL ARITHMETIC IS IN THE SUBDIVIDED
C COORDINATE SYSTEM.
C
C LOOP ON TAIL CHORDS
B32
INAKE(JW) = MAX0(INAKE(JW), INTRST)
GO TO 650

640 CONTINUE
IF (INTRST .LE. JW-JDIAG) GO TO 660
MYBSSW = JW
INAKE(JW) = INTRST

650 CONTINUE
IF (NSUBDV .EQ. 0) GO TO 657
DO 655 I = 1, NSUBDV
INAKE(JW-I) = INAKE(JW) - 1
INAKE(JW+I) = INAKE(JW) - 1
655 CONTINUE

657 CONTINUE
JW = JW + NSUBDV
GO TO 660

C END OF LOOP ON WING CHORDS

C 660 CONTINUE
IF (ISHEPLW .NE. 0) GO TO 706

C 690 CONTINUE
JT = JT + NSUBDV
IF (JT .LE. MYBBST) GO TO 525
C END OF LOOP ON TAIL CHORDS, FROM 548#

C
685 CONTINUE
NBBSSW = NBBSW
DO 690 JW = NSUBCM, NBBSSW, NSUBDV
NBBSSW = MAX0(NBBSSW, INAKE(JW))
690 CONTINUE
NBBW = NBBSSW
IF (NSUBDV .GT. 1) NBBW = (NBBW-IWBW/NSUBDV + 1
IF (CHECKPR) WRITE(NF6,7010) (INAKE(I), I = 1, NBBSW)
GO TO 720
C
C THE TWO SURFACES ARE COPLANAR. ENTER THE SECOND PLANFORM
C INTO THE SAME BOX ARRAY

700 CONTINUE
CPPLAN = .T.
CALL BXCPFF(XTLE, YTLE, XTEE, YTEE, LBXCW, IB0XW)
NBBSSW = WBBST
IF (.NOT. CHECKPR) GO TO 720
CALL PRNTBC(IB0XW, LBXCW, IXBST, WBST, WYBST, , .T.)
II = NBBW + 1
III = NBBW + WBBST
WRITE(NF6,7040) (FNLLOC(I), I = 0, III)
WRITE(NF6,7045) (FNLLOC(I), I = 0, III)
GO TO 720

C NO TAIL IS DEFINED. IS DOWNWASH SAMPLING DESIRED-

709 CONTINUE
CPPLAN = .F.
IF (ISHEPLW .EQ. 0) GO TO 720
C BYPASS THE TAIL PLANFORM AND BOX CODE SETUP, AND LOOP ON SAMPLE
C CHORDS TO DEFINE WING WAKE REGION
C (DO 708 JOHRD = 1, ISHEPLW )
JOHRD = 1
704 CONTINUE
JOHRD = 1
704 CONTINUE
J0 = (JOHRD(JOHRD-1)) NSUBDV + NSUBCH

B33
IF (IB0XF(JCHRD) .EQ. INIT) IB0XF(JCHRD) = (TELOC(JT)-1)XBW /
1 NSUBDV + 1
2
IF (IB0X(JCHRD) .EQ. INIT) IB0X(JCHRD) = (TELOC(JT)-1)XBW /
1 NSUBDV + 1
2
YCT = JT - .5
ICT = (IB0X(JCHRD)-1) * NSUBDV + 1XBW
EL = COS(PSIW) * ZLOC(JCHRD) - SIN(PSIW) * YCT
CAPL = ZLOC(JCHRD)
GO TO 580
C THE LOGIC FOR A TAIL CHORD IS USED. AFTER THE WING WAKE
C BOUNDS ARE DETERMINED FOR THIS TAIL CHORD, CONTROL IS RETURNED
C TO THIS LOOP
706 CONTINUE
708 CONTINUE
JCHRD = JCHRD + 1
IF (JCHRD .LE. ISMPLW) GO TO 704
C END OF LOOP ON SAMPLE CHORDS
GO TO 685
C GET DIAPHRAGM BOXES CODES FOR THE WING
720 CONTINUE
CALL BXCDI (IWKX, LBXCDW, LSCHD, IBOW )
C RETURNS - IBOW, CODES 2 AND 3 ADDED FOR DIAPHRAGM REGIONS
C MBBBSW = NUMBER OF SUBDIVIDED CHORDS, INCLUDING DIAPHRAGM
C MBBBW = NUMBER OF UNSUBDIVIDED CHORDS
C PRINT BOX CODES
IF (.NOT. (FROB .OR. CHECKPR)) GO TO 725
CALL PRNTBC(IBOW, LBXCDW, 1, MBBBSW, MBBBW, .T.)
IF (NSUBDV .NE. 1) CALL PRNTBC(IBOW, LBXCDW, 1, MBBBSW-1)MBBSW)/NSUBDV + 1)
1 1, MBBBSW-1XBW/NSUBDV, MBBBW, .F. )
GO TO 070
C DETERMINE THE PLANAR AIC ARRAY SIZE
NFKRN = MAX(MBBBW, MBBST-1XBW/NSUBDV + 1)
C WRITE THE BOX CODE ARRAYS INTO THE GEOMETRY SCRATCH FILE
REMNDE IGE0SC
CALL RDINT
ITYPE = SHMINED
IVAL = SHIB0XW
PARM(1) = 0.
PARM(2) = MAX
N = MBBBSW
N = (MBBBSW-1)/NSUBDV + 1
K = LBXCDW
CALL WRTBX(I0SC, ITYPE, N, N, PARM, IRR)
1 IF (.NOT. .ME. .G) GO TO 8070
C IF (NSUBF .EQ. 1) GO TO 750
IF (COPLAN) GO TO 750
IVAL = SHIB0XW
N = MBBST - 1XBST + 1
N = (MBBST - 1)/NOWRD + 1
C
K = LBXCDT
CALL RTEMX(IGEOC, NWRIT, RNDOU, NFS, NMS, LS, NMR, LWS, K, ID)
1  IBXRT, ITYPE, M, N, PARN, IRR)
IF (IRR .NE. 0) GOTO 8070
C
CHECK FOR DIAPHRAGMS CROSSING VERTICALLY
IF (YCRoss .LE. .5) GOTO 730
IF (FLOAT(YMBBT)*COS(PSIT) .LT. YCRCSS) GOTO 730
IF (FLOAT(YMBBT)*COS(PSIT) .GE. YCRoss) GOTO 8080
C
WRITE THE LEADING AND TRAILING EDGE LOCATIONS ONTO SCRATCH
730 CONTINUE
M = 1
N = MBSW + MBST
K = 1
IVAL = 6HFLXOC
CALL RTEMX(IGEOC, NWRIT, RNDOU, NFS, NMS, LS, NMR, LWS, K, ID)
1  FLEX, ITYPE, M, N, PARN, IRR)
IF (IRR .NE. 0) GOTO 8070
IVAL = 6HFLXOC
CALL RTEMX(IGEOC, NWRIT, RNDOU, NFS, NMS, LS, NMR, LWS, K, ID)
1  TFLX, ITYPE, M, N, PARN, IRR)
IF (IRR .NE. 0) GOTO 8070
C
DETERMINE THE ON-PLANFORM FRACTIONAL PART OF ALL UNSUBDIVIDED
C BOXES CUT BY A PLANFORM EDGE
CALL GMAREA(1BOOM, LBXCDT, .T., ALPHA, IJALPH, NALPHA)
NALPHA = NALPHA
IF (NSURF .EQ. 1 .OR. COPLAN GO TO 740
CALL GMAREA(1BOOM(ISUBT, 1), LBXCDT, .F., ALPHA(NALPHA+1)),
1  IJALPH(NALPHA+1), NALPH)
NALPHA = NALPHA + NALPH
740 CONTINUE
IF (CHECKPR) WRITE(NL6,7030) (IJALPH/I: ALPHA(I)), I=1, NALPH )
C
WRITE THE AREA MULTIPLIERS
M = 1
N = NALPHA
K = 1
IPARM(3) = NALPHA
IVAL = 5HALPHA
CALL RTEMX(IGEOC, NWRIT, RNDOU, NFS, NMS, LS, NMR, LWS, K, ID)
1  ALPHA, ITYPE, M, N, PARN, IRR)
IF (IRR .NE. 0) GOTO 8070
IVAL = 5HALPHA
CALL RTEMX(IGEOC, NWRIT, RNDOU, NFS, NMS, LS, NMR, LWS, K, ID)
1  JALPH, ITYPE, M, N, PARN, IRR)
IF (IRR .NE. 0) GOTO 8070
C
DETERMINE THE SPATIAL AIC PARAMETERS
THE MAGIC ARRAYS ARE WRITTEN TEMPORARILY ON IVPS FOR EDITING
ONTO IVPS FOR AFTER THE KPT- ARRAYS ARE WRITTEN ON
IGEOC, ALL 8PATT ARRAYS ARE TRANSFERRED TO IGEOC
ISCR = NUMBER OF MICS TRANSFERRED TO IVPS
NSCR = TOTAL NUMBER OF MICS PRESENTLY ON IVPS
ISCR = 0
B35
*NBCR = 0
*NCRSR = 0
*NCRMUL = 0
IMAX(3) = 0
IMAX(4) = 0
REWIND IVFSC
REWIND IVPSC
N = 2
K = 2
NPSR = 0
MAK = 0
IF (SYM .EQ. 0) GO TO 904
IF (PSI.W .EQ. 0 .OR. .NOT. DIHM) GO TO 800
c
C START OF LOOP FOR WING-WING PARAMETERS ON RECEIVING CHORDS,
C UNSUBDIVIDED (DETERMINES SPATIAL MUIC VALUES)
DO 790 JCOL = 1, MBB
C
C CALL PMARC (.T., IBOM, LBXC, IMAKE, JCOL)
C COMPUTES MUIC ARRAY FOR THE LEFT SURFACE CONTRIBUTION TO
C CHORD JCOL OF THE RIGHT SURFACE
C SURF = .T. INDICATES SOME RIGHT SURFACE CONTRIBUTION WAS
C FOUND
IF (.NOT. SURF) GO TO 800
MAK = MAK + 1
NPSR = NPSR + 1
KPWH(NMAK) = NPSR
IVAL = 10*WING-WING
IF (CHCRX) WRITE(NT6,7020) IVAL, JCOL, YEL, EL, NROW6, 
S NUIC(1, 1), NROW6-I+1), MUIC(2, NROW6-I+1), I=1, NROW6)
C
C WRITE MUIC ARRAY ON THE SCRATCH FILE
N = NROW6
PAM(4) = YBAR
PAM(5) = EL
CALL WRITEX(IPVSC, NMAK, RANDI, NFS, NS, NS, SLS, LSS, K, ID, 
S NUIC, IYVE, M, N, PAM, IRR)
IF (IRR .NE. 0) GO TO 8075
NBCR = NCR + 1
C
790 CONTINUE
C END OF LOOP ON RECEIVING CHORDS FOR WING-WING PARAMETERS,
C
800 CONTINUE
C WTRK = 0
C IF (NSURF .NE. 2) GO TO 900
C IF (SYM .EQ. 0) GO TO 900
C IF (PSI.W .EQ. 0 .OR. .NOT. DIHM) GO TO 900
C IF (PSI.W .EQ. 0 .) REWIND IVPSC
C C START OF LOOP FOR TAIL-TAIL PARAMETERS ON RECEIVING CHORDS,
C UNSUBDIVIDED
DO 805 JCOL = 1, MBB
C
C CALL PMARC (.F., IBOM(ISUBT), LBXC, IMAKE, JCOL)
C COMPUTES MUIC ARRAY FOR THE CONTRIBUTION OF THE LEFT TAIL ON
C CHORD JCOL OF THE RIGHT TAIL SURFACE
C
100 CONTINUE

B36
C
IF (.NOT. SURF) GO TO 900
MTK = MTK + 1
C
IVAL = 10H TAIL-TAIL
C
IF THE WING AND TAIL HAVE THE SAME DIHEDRAL, MODIFY THE WING
C
MUSIC ARRAY WHERE NEEDED BY THE TAIL, AND USE IT
C
IF (MUSIC .EQ. 0.) GO TO 840
C
OTHERWISE, WRITE THE MUSICS FOUND INTO IVPS'C
NPATK = NPATK + 1
KPTITT(NMTK) = NPATK
IVAL = 10H TAIL-TAIL
IF (CHECKR) WRITE(N4S,T020) IVAL,JCCL,YBAR,EL, NCRS,(MUSIC(1, 1) NCRS-I+1), MUSIC(2, NCRS-I+1), I=1,NCRS)
C
WRITE MUSIC ARRAY ON SCRATCH FILE
N = NCRS
PARM(4) = YBAR
PARM(5) = EL
CALL WRTDX(IVPS'C, NCRS, RAND0, N0S, N45, LS, NMR, LWS, K, ID, 1 MUSIC, ITYPE, M, N, PARM, IRR)
IF (IRR .NE. 0) GO TO 8075
NSCR = NSCR + 1
C
GO TO 895
840 CONTINUE
IF (ISCR .GE. NSCR) GO TO 850
CALL RDISN
CALL READM(IVPS'C, NCRS, RAND0, N0S, N45, LS, NMR, LWS, K, ID, 1 ITYPE, LS5, NCRS, IRR)
IF (IRR .NE. 0) GO TO 8090
ISCR = ISCR + 1
C
MERGE THE TWO MUSIC ARRAYS
DO 845 I = 1,N
IF (I .GT. NCRS) GO TO 842
IF (MUSIC(IN,1) .EQ. 0.) GO TO 845
IF (MUSIC(IN,2) .EQ. 0.) GO TO 842
MUSIC(IN,1) = MIN(MUSIC(IN,1),MUSIC(IN,2,1))
MUSIC(IN,2) = MAX(MUSIC(IN,2,1),MUSIC(IN,2,1))
GO TO 845
842 MUSIC(IN,1) = MUSIC(IN,1)
MUSIC(IN,2) = MUSIC(IN,2)
845 CONTINUE
NCRS = MAX(MUSIC(NCRS,N))
KPTITT(NMTK) = ISCR
IF (CHECKR) WRITE(N4S,T020) IVAL,JCCL,YBAR,EL, NCRS, (MUSIC(1,1) NCRS-I+1), MUSIC(2, NCRS-I+1), I=1,NCRS)
GO TO 895
C
THERE WERE NO MATRICES TO BE MERGED
850 NPATK = NPATK + 1
KPTITT(NMTK) = NPATK
C
WRITE MERGED AICS ON TO 2ND SCRATCH FILE
855 CONTINUE
N = NCRS
CALL WRTDX(IVPS'C, NCRS, RAND0, N0S, N45, LS, NMR, LWS, K, ID, 1 MUSIC, ITYPE, M, N, PARM, IRR)
IF (IRR .NE. 0) GO TO 8110

B37
C
900 CONTINUE
C
END OF LOOP FOR TAIL-TAIL PARAMETERS, FROM 800*
C
900 CONTINUE
C
COMPLETE ANY COPY FROM FIRST TO SECOND SCRATCH FILE
IF (ISCR .EQ. 0) REWIND IVPSC
IF (ISCR .GE. NSCR) GO TO 904
II = ISCR + 1
DO 902 I = II, NSCR
CALL RDINIT
CALL READM(X(IVPSC, NWIT, RANDCU, NFS, NNS, LS, NMR, K, NID, ID, I TYPE, LRS, MUAIC, M, N, PARM, IRR)
IF (IRR .NE. 0) GO TO 8090
CALL WRTENM(1IVPSC, NWIT, RANDCU, NFS, NNS, LS, NMR, LWS, K, ID, I TYPE, LRS, MUAIC, M, N, PARM, IRR)
IF (IRR .NE. 0) GO TO 8110
904 CONTINUE
C
COMPUTE THE RIGHT AND LEFT WING INFLUENCE PARAMETERS ON THE
TAIL OR SAMPLE CHORDS
MVITK = 0
MVITK = 0
ISCR = 0
ISCR = 0
NSCR2 = 0
NSCR2 = 0
REWIND IVPSC
IPARM(6) = 1
INITIALIZE THE MUAIC ARRAYS
DO 908 J = 1, 50
MUAIC(1, 1) = I+I
MUAIC(2, 1) = 0
MUAIC(1, I) = I+I
MUAIC(2, I) = 0
908 CONTINUE
IF (CCPLAN) GO TO 1015
IF (NSURF .NE. 2) GO TO 1120
CAPL = CAPL
YWMSP = CAPL*SIN(PSIW)
JTCL = MYSW - NSUB2
START OF LOOP ON TAIL CHORDS, TO COMPUTE
WING - TAIL INFLUENCE PARAMETERS
( DO 1010 JCOL = 1, MYBT )
JCOL = 1
1010 CONTINUE
IF (JCOL .LE. MYBT) GO TO 910
IROW = (IBOT-IBOT+1)/NSUBDV + 1
GO TO 915
1010 CONTINUE
JCOL = JCOL + NSUBDV
IROW = (JCOL-JCOL-1)/NSUBDV + 1
IF (IROW .EQ. MYBT) GO TO 930
1120 CONTINUE
CALL DCCdur(IBOT(IRUBT, I), LBXCRT, IRow, JCOL, XSUBT, JCOL, F, ICODE)
II = 1
DO 917 J = IRow, MYBT
IF (ICODE(J) .NE. 0) GO TO 918
917 CONTINUE
B38
II = II + 1

DO 920 I = III, NMBT
IF (ICDNF(I) .EQ. 0) GO TO 925
II = II + 1

920 CONTINUE

925 IRON = IRON + II - 2

C ENTRY INTO THE LOOP FROM SAMPLE WASH LOOP, FROM 1120*

C CALL MAIC(I,BOAM,BLXCDW,IRON,JCOL, CACL, YMVSP)

C GETS THE MAIC AND MAICL ARRAYS FOR RIGHT AND LEFT CONTRI-

C BUTTONS TO THE TAIL
IF (SURF) GO TO 935
IF (.NOT. SURFL) GO TO 1015
GO TO 985

935 NEWTK = NEWTK + 1

C DETERMINE WHETHER WING AND TAIL ARE PARALLEL
IF (PSLFDF .EQ. 0 .AND. NSURF .EQ. 2) GO TO 940
NPATK = NPATK + 1
KPARL (NEWTK) = NPATK

C WRITE NEW MAIC ARRAY ON SCRATCH
IVAL = 10HR WING-TAIL
N = NROWB
FARM(4) = YBAR
PARM(3) = EL
CALL WRITE(INPSK, NOWRIT, RANDOU, NPS, NSN, LS, NNR, LGS, K, ID, MAIC, ITYPE, M, N, PARM, ISR)
IF (ISR .NE. 0) GO TO 8075
NBRZ2 = NBRZ2 + 1
IF (CHECK) WRITE(NMTS, NO) VAL,JCC,YBAR,EL, NROWB, (MAIC(1, 1) NROWB-I+1), MAIC(2, NROWB+1), I1, NROWB)

C REINITIALIZE THE RIGHT MAIC ARRAY
DO 937 I = 1, NROWB
MAIC (1, I) = 1+I
MAIC(2, I) = 0

937 CONTINUE

C GO TO 982

C OLD MAIC ARRAY HAS BEEN FOUND WHICH MATCHES

940 CONTINUE

C ALL RIGHT HAND MAICS ARE THE SAME, PARALLEL SURFACES
IF (JCOL .GT. 1) GO TO 945
NPATK = NPATK + 1
IPARL = HPPATK

945 CONTINUE
KPARL (NEWTK) = IPARL
NCH3 = MAMD(NROWB, NROWB)

C

980 CONTINUE
IVAL = 10HR WING-TAIL
IF (CHECK) WRITE(NMTS, JOM) VAL,JCOL,YBAR,EL, NROWB, (MAIC(1, 1) NROWB-I+1), MAIC(2, NROWB-I+1), I1, NROWB)

C

982 CONTINUE
IF (.NOT. SURFL) GO TO 1000

B39
985 CONTINUE
MLWK = MLWK + 1
C DETERMINE WHETHER BOTH SURFACES HAVE NO DIHEDRAL
IF (PSIW .EQ. 0 .AND. PSIDIF .EQ. 0 .AND. NSURF .EQ. 2) GO TO 990
NPARX = NPARX + 1
KPLWM(MLWK) = NPARX
C WRITE NEW MAIC ARRAY ON SCRATCH
IVAL = 10ML WING-TAIL
N = NROWL
PARM(4) = YBARL
PARM(5) = ELL
CALL WRTEM(IWPS,C,WROIT,RANDOU,NFS,NMS,LS,MR,LWS, K, ID,
1   MAICL, ITYPE, W, N, PARM, IRR)
IF (IRR .NE. 0) GO TO 8075
NGORZ = NGORZ + 1
IF (CHECK) WRITE(N5,7020) IVAL,JCOL,YBARL,ELL, NROWL,
1   (MAICL1,NROWL-I+1),MAICL2,NROWL-I+1, I=1,NROWL)
C REINITIALIZE THE LEFT MAIC ARRAY
KPLWM(MLWK) = IPARAL
WMAIC(1,I) = MAIC(1,I) = 0
WMAIC(2,I) = MAIC(2,I) = 0
987 CONTINUE
GO TO 1000
C OLD MAIC ARRAY HAS BEEN FOUND WHICH MATCHES
990 CONTINUE
C USE THE SAME ARRAY FOR RIGHT AND LEFT CONTRIBUTIONS
KPLWM(MLWK) = IPARAL
NROWL = MAXI(NROWL,NROWL)
C 995 CONTINUE
IVAL = 10ML WING-TAIL
IF (CHECK) WRITE(N5,7020) IVAL,JCOL,YBARL,ELL, NROWL,
1   (MAICL1,NROWL-I+1),MAICL2,NROWL-I+1, I=1,NROWL)
1000 CONTINUE
IF (ISMPW .NE. 0) GO TO 1150
1010 CONTINUE
JCOL = JCOL + 1
IF (JCOL .LE. MBVT) GO TO 909
C END OF LOOP ON CHORDS, FOR WING-TAIL PARAMETERS
C C PLACE ANY BUILT UP ARRAY DUE TO PARALLEL SURFACES ON SCRATCH
IF (PSIDIF .NE. 0 .OR. PSIW .NE. 0) GO TO 1014
C MERGE THE RIGHT AND LEFT ARRAYS
NROWX = MINI(NROWL,NROWL)
DO 1012 I = 1,NROWX
   MAIC(1,I) = MINI( MAIC(1,I),MAICL(1,I) )
   MAIC(2,I) = MAXI( MAIC(2,I),MAICL(2,I) )
1012 CONTINUE
1014 IF (PSIDIF .NE. 0) GO TO 1015
C WRITE THE ARRAY ONTO IWFTSC
IVAL = 10PARAL TAIL
PARM(4) = YBAR
PARM(5) = ELL
IPARM(6) = 2
N = MAXI(NROWL,NROWL)
CALL WRTEM(WFTSC, WROIT,RANDOU,NFS,NMS,LS,MR,LWS, K, ID,
1   MAIC, ITYPE, W, N, PARM, IRR)
I

F (IRR .NE. 0) GO TO 8110
NSCR = NSCR + 1
C PLACE THE KPT-- ARRAYS ON GEOMETRY SCRATCH AS ONE MATRIX.
1015 CONTINUE
IF (NPATK .EQ. 0) ENDFILE IGECS0
IF (NPATK .EQ. 0) GO TO 2000
IPARM(5) = NA4K
IPARM(4) = NTIX
IPARM(5) = NEWTK
IPARM(6) = NLWTX
M = 0
N = 0
IF (NA4K .EQ. 0) GO TO 1030
M = 1
N = NA4K
DO 1020 I = 1, NA4K
1020 KPT(I,1) = KPTM4(I)
1030 CONTINUE
IF (NEWTK .EQ. 0) GO TO 1050
M = M + 1
N = MAX0(N, NEWTK)
DO 1040 I = 1, NEWTK
1040 KPT(M, I) = KPTT0(I)
1050 CONTINUE
IF (NLWTX .EQ. 0) GO TO 1070
M = M + 1
N = MAX0(N, NLWTX)
DO 1080 I = 1, NLWTX
1080 KPT(M, I) = KPTLW(I)
1070 CONTINUE
IF (NLWTX .EQ. 0) GO TO 1090
M = M + 1
N = MAX0(N, NLWTX)
DO 1090 I = 1, NLWTX
1090 CONTINUE
K = 4
IVAL = 3HKT
CALL WRTCH(IGECS0, HXVIT, RANDOU, NBS, NMS, LS, NM, LW, K, ID, 1, KPT, ITYPE, M, N, PARM, IRR)
IF (IRR .NE. 0) GO TO 8070
END FILE IGECS0
C
C MOVE THE MIAUC ARRAYS TO THE GEOMETRY SCRATCH TAPE
RENEW IMFSC
RENEW IVPS0
IVAL = 6HIAUC
K = 2
IF (NBOC .NE. 0) GO TO 1096
DO 1094 I = 1, NBOC
CALL RDMIT
CALL READMK(IMFSC, HXVIT, RANDOU, NBS, NMS, LS, NM, LW, K, ID, 1, IYPE, LS, MIAUC, M, N, PARM, IRR)
IF (IRR .NE. 0) GO TO 8100
CALL WRTCH(IGECS0, HXVIT, RANDOU, NBS, NMS, LS, NM, LW, K, ID, 1, MIAUC, IYPE, M, N, PARM, IRR)
IF (IRR .NE. 0) GO TO 8070
GO TO 341
GO TO 00945
GO TO 00946
GO TO 00947
GO TO 00948
GO TO 00949
GO TO 00950
GO TO 00951
GO TO 00952
GO TO 00953
GO TO 00954
GO TO 00955
GO TO 00956
GO TO 00957
GO TO 00958
GO TO 00959
GO TO 00960
GO TO 00961
GO TO 00962
GO TO 00963
GO TO 00964
GO TO 00965
GO TO 00966
GO TO 00967
GO TO 00968
GO TO 00969
GO TO 00970
GO TO 00971
GO TO 00972
GO TO 00973
GO TO 00974
GO TO 00975
GO TO 00976
GO TO 00977
GO TO 00978
GO TO 00979
GO TO 00980
GO TO 00981
GO TO 00982
GO TO 00983
GO TO 00984
GO TO 00985
GO TO 00986
1094 CONTINUE
1096 IF (NSCR2 .LE. 0) GO TO 1110
   DO 1100 I = 1, NSCR2
      CALL READMX(IVPC, MXWRT, RANDC, NPS, NPS, LS, NSR, K, NID, ID,
      1      IYPE, LRS, MACIC, M,N, PARM, IRR)
   IF (ARR .NE. 0) GO TO 8090
      CALL WRDECX(GEOB, MXWRT, RANDC, NPS, NPS, LS, NSR, LWS, K, ID,
      1      MACIC, IYPE, M,N, PARM, IRR)
   IF (ARR .NE. 0) GO TO 8070
1100 CONTINUE
1110 CONTINUE
   END FILE IGEOSC
   GO TO 2000
C
   LOOP ON SAMPLE WASHER CHORDS (USED IF ISPLW .NE. 0) TO
   DETERMINE MACIC ARRAYS FOR RIGHT AND LEFT WING INFLUENCE
   ON EACH CHORD
1120 CONTINUE
   NWTH = 0
   NWTH = 0
   IF (ISPLW .EQ. 0) GO TO 1015
   C ( DO 1200 JCORD = 1, ISPLW )
      JCORD = 1
1130 CONTINUE
   JCOR = IGEOSC(JCORD)
   CJPL = IGEOSC(JCORD)
   YMVP = IGEOSC(PS1)
   IRON = IGEOSC(ORD)
   GO TO 930
C
   THE LOGIC FOR TAIL CHORD IS USED. AFTER THE MACIC ARRAYS
   ARE DETERMINED AND STORED, CONTROL IS RETURNED TO THIS LOOP.
1150 CONTINUE
1200 CONTINUE
   JCORD = JCORD + 1
   IF (JCORD < ISPLW) GO TO 1130
C
   END OF LOOP ON SAMPLE WASHER CHORDS
   GO TO 1015
2000 CONTINUE
   ENDFILE IGEOSC
   REINIT IGEOSC
   REINIT IWPSC
   RETURN
C
   OUTPUT FORMATS
   CARD F
   FORMAT(I0.15X,39X- - - GEOMETRIC PARAMETERS - - - /
      1      1ND.29HARDF-LOCAL AXIA Definition-,
      2      4X, 10X-LOCATION, 4X, 10X-LOCATION, 4X,
      2      NONHEAL ANGLE (PSI) / 27X,5HANG, F10.5,4X, F10.3, 8X,
      3      .2, 8X DEGREES )
   FORMAT(I0.15X,39X- - - BOX PATTERN DEFI INITION-3X,6XCHORDS,10X,
      1      4X,8XEDGE /37X, 13, 8X, F10.4, 6X,F10.4 )

842
0017 FORMAT(17X,13H-SAMPLE WASH,13, 8H CHORDS-,5X,8(HChord,6X,5HBOXF, GEOMEX 01040
  1 8H,5HBOX, 6X, 4H2LOC / (37X, 13.8X, 13, 7X, 13, 6X, F7.2 ) ) GEOMEX 01041
C  CARD N GEOMEX 01042
0021 FORMAT(1SH,CARD) - PLANFORM DEFINITION POINT COUNTS-, 5X, GEOMEX 01043
  1 12H,LEADING EDGE,4X,13H,TRAILING EDGE / 4X,4W(MEM, 17, 9X, 17 ) GEOMEX 01044
0022 FORMAT(42X,4HTAIL, 17, 9X, 17 ) GEOMEX 01045
C  CARDS 1 TO L GEOMEX 01046
0029 FORMAT(1SH,CARD) TO CARD - PLANFORM DEFINITIONS-, 9X,1H9X,1H9Y, GEOMEX 01047
  1 4X,12H(LOCAL AREA) ) GEOMEX 01048
0030 FORMAT(32X,A10, F9.3, F10.3/ (41X,2F10.3) ) GEOMEX 01049
0040 FORMAT(1H0,7X,3SH-BOX DIMENSIONS-, B1 (LENGTH ) =, E18.8,5X, GEOMEX 01050
  1 13H8/BETA (4EDTH) =, E18.8 ) GEOMEX 01051
C  CHECK PRINT FORMATS, USED ONLY WHEN CHECR = .T. GEOMEX 01052
0050 FORMAT(13SHDIMA) ARRAY - , 4013 / (15X,4D13) ) GEOMEX 01053
0060 FORMAT(1SHDIMA) ARRAY FOR,A10, 8H, CHORD I2, 8H, YBAR = F6.3, GEOMEX 01054
  1 7H, EL = F7.2 / 10X, 3ROW.13.214 / (16X,214 ) ) GEOMEX 01055
0070 FORMAT(1SHD IDIA) (= J4100 + I OCTAL) AND ALPHA ARRAYS, AS GEOMEX 01056
  1 1H STORED / 6(13X,1HJALPH ALFA ) / (6(5X,6, F7.4) ) ) GEOMEX 01057
0080 FORMAT(1SHD IDIA) (= J4100 + I OCTAL) AND ALPHA ARRAYS, AS GEOMEX 01058
  1 1H STORED / 6(13X,1HJALPH ALFA ) / (6(5X,6, F7.4) ) ) GEOMEX 01059
0040 FORMAT(21SH CHECK PRINT, FEOLOC / (1OF12.7) ) GEOMEX 01060
0045 FORMAT(1H9X14X, XEVLOC / (1OF12.7) ) GEOMEX 01061
C  DIAGNOSTIC FORMATS GEOMEX 01062
0100 FORMAT(53H000) WARNING - WEDGE AND XCENTR WERE BOTH SPECIFIED. GEOMEX 01063
  1 2H WEDGE WILL BE IGNORED ) GEOMEX 01064
0100 FORMAT(43H000) WARNING - SAMPLE WASH SPECIFICATION SET I2, 8H IS IN GEOMEX 01065
  1 5H ERROR. ONLY THE PRECEDING ONES WILL BE CALCULATED ) GEOMEX 01066
0100 FORMAT(53H000) WARNING - SAMPLING OF UPHUSHES CANNOT BE DONE IF GEOMEX 01067
  1 3H A TAIL HAS BEEN DEFINED. SIMPLEY =I3,16H WILL BE IGNORED GEOMEX 01068
  2 4H ) GEOMEX 01069
9110 FORMAT(23H000) ERROR - PARAMETER ,A6,23H WAS NOT SPECIFIED. IT GEOMEX 01070
  1 2H MUST ALWAYS BE GIVEN ) GEOMEX 01071
9120 FORMAT(33H000) ERROR - EITHER WEDGE OR XCENTR MUST BE SPECIFIED GEOMEX 01072
  1 4H ) GEOMEX 01073
9130 FORMAT(13H000) ERROR - ,A6,29H IS OUTSIDE ALLOWED RANGE ) GEOMEX 01074
9150 FORMAT(13H000) ERROR - ,A10,8H, CHORD I2, 8H, YBAR = F6.3, GEOMEX 01075
  1 13H, A COMBINATION OF- ) GEOMEX 01076
  2 1H8, 10X,25H, NON-MONOTONIC Y-VALUE GEOMEX 01077
  3 1H8, VALUE NON-ZERO, 10X,34H, TIP T.E. Y-VALUE DISAGREES WITH GEOMEX 01078
  4 1H8 TIP L.E. VALUE ) GEOMEX 01079
9160 FORMAT(15H000) ERROR - XCENTR NOT WITHIN 50 BOX LENGTHS (B1 =,, GEOMEX 01080
  1 135.0,25H) OF THE WING L.E. (.135.0, 3H ) ) GEOMEX 01081
9170 FORMAT(32H000) ERROR - WHILE WRITING ON GEOMETRY SCRATCH FILE A10, GEOMEX 01082
  1 15H, ERROR CODE = I4, 4H ) GEOMEX 01083
9178 FORMAT(14X,8HARRAY ,A6,19H, DIMENSIONED (I4, I4, I4, 11H) WAS BEING GEOMEX 01084
  1 8H WRITTEN ) GEOMEX 01085
9179 FORMAT(14X,23H THE MUSICA ARRAY FOR A10,15H, DIMENSIONED (I4, I4, I4, GEOMEX 01086
  1 15H) WAS BEING WRITTEN ) GEOMEX 01087
9180 FORMAT(52H000) ERROR - THE TAIL AND WING, OR THEIR DIAPHRAGMS, GEOMEX 01088
  1 15H CAN NOT BE ABOVE TO BELOW ) GEOMEX 01089
9190 FORMAT(54H000) ERROR - WHILE READING FROM GEOMETRY SCRATCH FILE GEOMEX 01090
  1 A10,15H, ERROR CODE = I4, 4H ) GEOMEX 01091
9190 FORMAT(14X,8HARRAY ,A6,19H, DIMENSIONED (I4, I4, I4, 11H) WAS BEING GEOMEX 01092
  1 8H READ ) GEOMEX 01093
C  ERRORS - ALL ERRORS CALL FLUSH GEOMEX 01094
0100 CONTINUE GEOMEX 01095

B43
WRITE (NT6,9110) IVAL
GO TO 8900
8015 CONTINUE
WRITE (NT6,9130) IVAL
GO TO 8900
8020 CONTINUE
WRITE (NT6,9120)
GO TO 8900
8030 CONTINUE
WRITE (NT6,9130) IVAL
GO TO 8900
8040 CONTINUE
GO TO 8900
8050 CONTINUE
WRITE (NT6,9150) IVAL, IRR
GO TO 8900
8060 CONTINUE
WRITE (NT6,9180) BI, XEDGEX
GO TO 8900
8070 CONTINUE
WRITE (NT6,9170) IGEOSC, IRR
WRITE (NT6,9172) IVAL, M,N
GO TO 8900
8075 CONTINUE
WRITE (NT6,9170) IVPSC, IRR
WRITE (NT6,9175) IVAL, M,N
GO TO 8900
8080 CONTINUE
WRITE (NT6,9180)
GO TO 8900
8090 CONTINUE
WRITE (NT6,9190) IVPSC, IRR
WRITE (NT6,9192) IVAL, M,N
GO TO 8900
8100 WRITE (NT6,9190) IWFSC, IRR
WRITE (NT5,9192) IVAL, M,N
GO TO 8900
8110 WRITE (NT6,9170) IWFSC, IRR
WRITE (NT6,9175) IVAL, M,N
8900 CALL FLUSH(1)
END
SUBROUTINE EDGCHK (XEDGE, YEDGE, NEDGE, IEDGE, IRR)

C CHECKS FOR MONOTONIC EDGE VALUES OF X FOR LEADING EDGES, AND
C Y FOR EITHER LEADING OR TRAILING. CHECKS Y FIRST AND LAST
C VALUES TO ENSURE DEFINITION FROM CENTERLINE TO TIP
C XEDGE = X-VALUES FOR ONE PLANFORM EDGE
C YEDGE = Y-VALUES
C NEDGE = NUMBER OF (XEDGE,YEDGE) SETS DEFINING THE PLAN-
C FORM EDGE
C IEDGE = 1 FOR LEADING EDGE, 2 FOR TRAILING EDGE
C IRR = ERROR RETURN, 0 SUCCESSFUL
C 1, NON-MONOTONIC Y-VALUES
C 2, NON-MONOTONIC X-VALUES, LEADING EDGE ONLY
C 4, FIRST Y-VALUE NON-ZERO
C 8, TIP Y-VALUE OF A T.E. DISAGREES WITH PREVIOUS
C L.E. TIP VALUE
C
C DIMENSION XEDGE(1), YEDGE(1)
C
C IRR = 0
C IF (YEDGE(1) .NE. 0.) IRR = 4
C DO 100 I = 2, NEDGE
C IF (YEDGE(I) .LT. YEDGE(I-1)) GO TO 130
C GO TO (50, 100), IEDGE
C 90 IF (XEDGE(I) .LT. XEDGE(I-1)) GO TO 200
C 100 CONTINUE
C GO TO 250
C 130 IRR = IRR + 1
C GO TO 250
C 200 IRR = IRR + 2
C 250 CONTINUE
C GO TO (300, 350), IEDGE
C 300 YTIP = YEDGE(NEDGE)
C GO TO 350
C 350 IF (YEDGE(NEDGE) .NE. YTIP) IRR = IRR + 8
C
C 500 RETURN
C
C END
SUBROUTINE EDGCHK(EDGE,YEDGE,NEdge,IEEE,IRR)
C
C CHECKS FOR MONOTONIC EDGE VALUES OF X FOR LEADING EDGES, AND
C Y FOR EITHER LEADING OR TRAILING. CHECKS Y FIRST AND LAST
C VALUES TO ENSURE DEFINITION FROM CENTERLINE TO TIP
C
C  XEDGE = X-VALUES FOR ONE PLANFORM EDGE
C  YEDGE = Y-VALUES
C  NEDGE = NUMBER OF (XEDGE,YEDGE) SETS DEFINING THE PLAN-
C    FORM EDGE
C  IEDGE = 1 FOR LEADING EDGE, 2 FOR TRAILING EDGE
C  IRR = ERROR RETURN; 0 SUCCESSFUL
C
C  1; NON-MONOTONIC X-VALUES
C  2; NON-MONOTONIC Y-VALUES, LEADING EDGE ONLY
C  4, FIRST Y-VALUE NON-ZERO
C  8, TIP Y-VALUE OF A T.E. DISAGREES WITH PREVIOUS
C
C L.E. TIP VALUE
C
C DIMENSION XEDGE(1),YEDGE(1)
C
C IRR = 0
C IF (YEDGE(1) .NE. 0.) IRR = 4
C DO 100 I = 2,NEdge
C IF (YEDGE(I) .LT. YEDGE(I-1)) GO TO 150
C  GO TO (50,100), IEDGE
C 90 IF (YEDGE(I) .LT. YEDGE(I-1)) GO TO 200
C 100 CONTINUE
C  GO TO 250
C 130 IRR = IRR + 1
C  GO TO 250
C 200 IRR = IRR + 2
C 250 CONTINUE
C  GO TO (300,350), IEDGE
C 300 YTip = YEDGE(HEdge)
C  GO TO 350
C 350 IF (YEDGE(HEdge) .NE. YTIP) IRR = IRR + 8
C
C 500 RETURN
C
C DO

B45
IB = IB + 1

1000 CONTINUE
GO TO 3000

C

PROGRAM WILL RETRIEVE NJ BOXES FROM CHORD J

1100 CONTINUE

JWB = (J-1)/NEWRD + 1
JB = (NEWRD - MOD(J,NEWRD) ) / 3
IF (JB .EQ. 0) JB = 0
IJMASK = SHIFT(MASK,JB)
NB = -JB
DO 2000 II = I,IEND,ISKIP
IJWORD = IB0K(II,JB)
IJCODE = IJWORD.AND.IJMASK
IJCODE(II) = SHIFT(IJCODE,NB)
IB = IB + 1
2000 CONTINUE

C

3000 CONTINUE
RETURN
END
SUBROUTINE NCODE(IBOX,LBOX, IA, JA, IL, ICODE
DIMENSION IBOX(LBOX,1)

ENCODES ONE INTEGER BOX CODE INTO THE PACKED CODE ARRAY ALONG A PORTION OF A CHORD, REPLACING PREVIOUS VALUES

IBOX - ARRAY OF BOX CODES, PACKED 20 PER WORD
LBOX - ROW DIMENSION OF BOX CODES ARRAY
IA - I-TH INDEX OF FIRST BOX CODE TO SET
JA - J-TH INDEX OF FIRST BOX CODE TO SET
IL - I-TH INDEX OF THE LAST BOX CODE TO SET
ICODE - CODE VALUE, 1, 2, OR 3, FOR THE

INTEGER SHIFT
DATA MASK / 77777777777777DB /
DATA NBWD /2D/
JSB = (JA-1)/NBWD + 1
JB = (NBWD - MOD(JA,NBWD) ) * 3
IF (JSB .EQ. 6D) JB = 0

JB = A LEFT SHIFT COUNT
ICODE = SHIFT(ICODE,JB)
IJMASK = SHIFT(MASK,JB)
DO 100 II = IA,IL
IJCODE = IJMASK .AND. IBOX(II,JSB)
100 IBOX(II,JSB) = IJCODE .OR. ICODE
RETURN
END
SUBROUTINE PRNTBC(IBOX, LBXCD, IFRST, ILAST, JLAST, SUBD)

DIMENSION IBOX(LBXCD,1), ICCCE(150)

LOGICAL SUBD

PRINTS BOX CODES, SUBDIVIDED OR UNSUBDIVIDED

IBOX - COMPRESSED BOX CODE ARRAY

LBXCD - ROW DIMENSION OF BOX CODE ARRAY

IFIRST - FIRST ROW DESIRED TO PRINT

ILAST - LAST ROW DESIRED

JLAST - LAST COLUMN DESIRED (FIRST IS ALWAYS ONE)

SUBD - .T., SUBDIVIDED CODES DESIRED

.T., UNSDIVIDED CODES DESIRED

COMMON /GEOITY/COPLAN,NSUBDV,XSUBDV,NSUBEC,NSURF,

1 B1,BIBETA,BIS,BIBTAS,WAX,WAX1,PSIW,GPECKTY

2 N4W,IBIBW,IWBE,WBBM,WBSW,MIBBSW,MIBS,

3 IBXW, XCENTR

LOGICAL COPLAN

COMMON /CONTROL/PREVEX,OMACH,TITLE(8),PRVEOM,PRVWVE,DIHM,DIHT,

1 DEFAULT

LOGICAL PRVEOM,PRVWVE,DIHM,DIHT,DEFAULT

COMMON /PROBLM/XMACH,MODCES,NTSLOP,NKVALS,SUOOTH,NDEG,GRDFIT,

1 EXAI, SUBDV, PLYWOD

LOGICAL SMOOTH,GRDFIT,EXAI, SUBDV, PLYWOD

COMMON /FILES/ N75,N76,INTAPE,INFSF,NSPAIC,NSOUTP,

1 IOUSFP, MODESC, IVPSG, IGECSC, IWIFSC, IAICSC

DIMENSION BCD(4)

INTEGER BCD

DATA BCD/ 1H,1H1,1H2,1H3 /

DATA MDVRD/0 /

R001 FORMAT(1H1,5X,8A10/1H0)

R002 FORMAT(19X,2DHFOR SUBDIVIDED BOXES,25X, 8H1XW =,12.11H (SUBDIVID

1 SOMED ROW OF UNSUBDIVIDED CENTER) )

R005 FORMAT(22X,4HMACH,F11.7,56X,ICCODE = 1 PLANFORM BOX /

1 19X,20IH-),61X,42 = DIAPHRAGM BOX =/100X,43 = WAKE BOX *)

R010 FORMAT(10DH,4X,3114 / ( 9X,3014 )

R012 FORMAT(1H )

R020 FORMAT(1X,13,2X,6DA2 / (12X,6DA2)

WRITE (N76,6001) TITLE

IF (NSUBDV .EQ. 1) GO TO 100

IF (.NOT. SUBD) GO TO 100

WRITE (N76,6002) XWV

100 CONTINUE

WRITE (N76,6005) WMACH

WRITE (N76,6010) (1, I = 2,JLAST,2)

WRITE (N76,6012)

DO 250 IROW = IFRST, ILAST

CALL DCODE(BBOX,LBXCD, IRW-1, IRW,JLAST, SUBD, ICODE )

C

CHANGE INTEGER CODES TO ALPHANUMERIC

DO 200 IJ = 1, JLAST

IF (ICCODE(J) .EQ. 0) ICCODE(J) = BCD(1)

IF (ICCODE(J) .EQ. 1) ICCODE(J) = BCD(2)

IF (ICCODE(J) .EQ. 2) ICCODE(J) = BCD(3)

200 CONTINUE

B49
IF (ICODE(J).EQ.3) ICODE(J) = BCD(4)
200 CONTINUE
WRITE (NL6,6020) IRGW, (ICODE(J), J = 1, JLAST)
290 CONTINUE
RETURN
END
SUBROUTINE BXCPF(XLE,YLE,XTE,YTE,NE,LSR68,IBOX)

C GENERATES THE BOX CODES FOR THE ON-PLANFORM BOXES OF ONE
C SURFACE.
C
C INPUT PARAMETERS
C XLE = X-VALUES, LEADING EDGE, NON-DIMENSIONAL
C YLE = Y-VALUES, LEADING EDGE
C NE = NUMBER OF LEADING EDGE POINTS
C XTE = X-VALUES, TRAILING EDGE
C YTE = Y-VALUES, TRAILING EDGE
C NTE = NUMBER OF TRAILING EDGE POINTS
C LSR68 = MAXIMUM NUMBER OF SUBDIVIDED ROWS ALLOWED

C OUTPUT PARAMETERS
C IBOX = COMPRRESSED BOX CODES, SET I FOR PLANFORM BOXES,
C UNCHANGED ELSEWHERE

C COMMON /FILES/ NT5,NT8,INTEP,INFSP,MRIC,MRIC,NOUPT,
1 ICUPS,ICOSC,EICPS,IICPS,ICPSC,
COMMON /GEO2/ COPLAN,NUBDY,NSBDY,NSBDY,NSBCN,NUBDY,
1 B1,B1,B1,B11,B11,B11,VLAM,VLAM,PSW,
2 MBWN,MBWN,MBWN,MBWN,MBWN,MBWN,MBWN,MBWN,
3 IXBN,IXCENTR

LOGICAL COPLAN
COMMON /GEO2/ TLAX,TLAZ,PSW,MBST,MBST,MBST,MBST,MBST,
1 MBST,MBST,MBST,MBST,MBST,MBST,MBST,MBST,
COMMON /FILES/ PXLOC(250), PXLOC(250), J1AG

LOGICAL WING

DIMENSION XLE(1),YLE(1),XTE(1),YTE(1)
DIMENSION IBOX(LSR68,1)

C INPUT COMMON PARAMETERS -
C IMBT = SUBDIVIDED ROW OF FIRST UNSUBDIVIDED BOX ON TAIL
C NSBDY = (INTEGER) NUMBER OF SUBDIVISIONS
C NSBDY = (REAL) NSBDY
C NSBDY = NSBDY/2
C NSBDY = NSBDY/2 + 1, CENTER SUBDIVIDED BOX

C OUTPUT COMMON PARAMETERS -
C IMBN = 0, WING BEING DONE, CHANGED TO SUBDIVIDED ROW
C OF FIRST SUBDIVIDED BOX CENTER ON WING
C .NE. 0, TAIL BEING DONE, NOT CHANGED

C OUTPUT COMMON PARAMETERS -
C MBWN = NUMBER OF SUBDIVIDED ROWS TO AFT END OF (WING)
C MBST
C MBWN = NUMBER OF SUBDIVIDED CHORDS ON THE (TAIL)
C MBST
C MBWN = NUMBER OF UNSUBDIVIDED ROWS ON (WING)
C MBST
C MBWN = NUMBER OF UNSUBDIVIDED CHORDS ON (TAIL)
C MBST
C PELOC = ARRAY OF (LEADING) EDGE X-LOCATIONS, SUBDIVIDED
C TELOC (TRAILING)

C 851
\begin{verbatim}
WSBS = 0
DEL = 1.0/XSUBDV

C LOCATION OF FIRST CHORD
YNIN = .54*(1.0 + DEL)

C SLOPES OF FIRST LEADING AND TRAILING EDGE SEGMENTS
WREFLE = YLE(1)
YREFLE = YLE(1)
WREFTE = YTE(1)
YREFTE = YTE(1)
DELE = (YLE(2)-WREFLE) / (YLE(2)-YREFLE)
DELTE = (YTE(2)-WREFTE) / (YTE(2)-YREFTE)
ILE = 2
ITE = 2
IERR = 0

C WAS THIS CALL FOR WING OR TAIL -
IF (IXBW .BR. 0) GO TO 120
C SET UP COUNTERS FOR TAIL
SURF = 4HTAIL
WENG = .F.
MWT = IFIXYLE(NLE))
NSCHRD = MWT * NSUBDV
JXLOC = MWT * NSUBDV + 1
IXB = IXBST
LSRR = LSRO4S
IF (.NOT. COPLANE) LSRR = LSRR + IXBST - 1
GO TO 130
C SET UP COUNTERS FOR WING
120 CONTINUE
SURF = 4HWING
WENG = .T.
NSCHRD = MWT * NSUBDV
JXLOC = 1
MIN = WREFLE + (YMIN-YREFLE)*DELE
IXW = (1.-XMIN)*XSUBDV + 1
IPX = 1
LSRR = LSRO4S

C START LOOP ON SUBDIVIDED CHORDS
130 CONTINUE
YCHRD = YMIN
DO 390 JCHRD = 1,NSCHRD
C FIND LEADING EDGE OF THIS CHORD
140 CONTINUE
C IS THE CURRENT L.E. SEGMENT STILL GOOD -
IF (YCHRD - YLE(ILRE)) 180,170,160
C NO, ANOTHER SEGMENT IS NEEDED
150 CONTINUE
WREFLE = YLE(ILRE)
ILE = ILE + 1
C CHECK FOR EXCEEDING LIMIT
IF (ILE,NS, NLE) GO TO 710
C CHECK FOR EDGE SEGMENT PARALLEL TO (SKIP THE SEGMENT) OR
C CUTTING BACK TOWARD CENTER-LINE (ERROR)
IF (WREFLE - YLE(ILRE)) 160,150,730
C SEGMENT HAS POSITIVE SLOPE

B52
\end{verbatim}
180 \text{XREFLE} = \text{XLE}((\text{ILE} - 1))
\text{DELE} = (\text{XLE}((\text{ILE}) - \text{XREFLE}) / (\text{XLE}((\text{ILE}) - \text{XREFLE})
\text{GO TO 140}

C

C CHORD CENTER LIES ON SEGMENT ENDPOINT

170 CONTINUE
\text{PERLOC}((\text{JEXLOC}) = (\text{XLE}((\text{ILE}) - 1.0) * \text{XSUBDV} + \text{IXBW})
\text{GO TO 190}

C

C CHORD CENTER LIES WITHIN THE SEGMENT

180 CONTINUE
\text{PERLOC}((\text{JEXLOC}) = (\text{XREFLE} + \text{DELE}\times(\text{YCHORD} - \text{YREFLE}) - 1.0) * \text{XSUBDV}
1 + \text{IXBW}

C

190 CONTINUE
\text{PERLOC}((\text{JEXLOC}) = \text{IF}((\text{PERLOC}((\text{JEXLOC})) + .5
\text{ISTART} = \text{PERLOC}((\text{JEXLOC}) + 1

C

C THE FOLLOWING CODE FINDS THE TRAILING EDGE OF THIS CHORD IN

C THE SAME MANNER AS ABOVE.

240 CONTINUE
\text{IF} ((\text{YCHORD} - \text{YTE}(\text{ITE})) = 280, 270, 250

250 CONTINUE
\text{YREFTE} = \text{YTE}(\text{ITE})
\text{ITE} = \text{ITE} + 1
\text{IF} ((\text{ITE} \geq \text{MTE}) \text{GO TO 721})
\text{IF} ((\text{YREFTE} = \text{YTE}(\text{ITE})) = 260, 250, 740

280 \text{YREFTE} = \text{YTE}(\text{ITE} - 1)
\text{DELTE} = ((\text{XTE}(\text{ITE}) - \text{YREFTE}) / (\text{YTE}(\text{ITE}) - \text{YREFTE})
\text{GO TO 240}

270 CONTINUE
\text{YDELLOC}((\text{JEXLOC}) = (\text{XTE}(\text{ITE}) - 1.0) * \text{XSUBDV} + \text{IXBW}
\text{GO TO 290}

280 CONTINUE
\text{YDELLOC}((\text{JEXLOC}) = (\text{XREFLE} + \text{DELTE}(\text{YCHORD} - \text{YREFTE}) - 1.0) * \text{XSUBDV}
1 + \text{IXBW}

290 CONTINUE
\text{YDELLOC}((\text{JEXLOC}) = \text{IF}((\text{PERLOC}((\text{JEXLOC})) + .5
\text{IEND} = \text{PERLOC}((\text{JEXLOC})
\text{IF} ((\text{IEND} \geq \text{MTE}) \text{GO TO 770}

C

SET BOX CODES TO 1 FOR PLANFORM BOXES OF THIS CHORD

300 CONTINUE
\text{CALL NCCODE(\text{IBOX}, \text{LSXOW}, \text{ISTART}, \text{YCHORD}, \text{IEND}, 1)}

C

\text{XBOX} = \text{MAX}(\text{XBOX}, \text{IEND})
\text{JEXLOC} = \text{JEXLOC} + 1
\text{YCHORD} = \text{YCHORD} + \text{DEL}

330 \text{CONTINUE}

C END OF LOOP ON CHORDS

C

\text{IF} ((\text{AING}) \text{GO TO 390}
\text{XOBAT} = \text{XBOX}
\text{YOBAT} = \text{YOBAT} * \text{XOBAT} + 1
\text{WOBAT} = \text{NSCHORD}
\text{GO TO 370}

340 \text{NBOX} = \text{NBOX}

B53
MIW = (MWS-1)DSW/MSUBD + 1

MRSW = NSDIV

370 CONTINUE
IF (IERR .NE. 0) GO TO 750
C
RETURN
C
C ERROR DIAGNOSTICS
C
C LIMIT EXCEEDED
710 IERR = 1
EDGE = 8H LEADING
ISEC = ILE - 1
GO TO 750
720 IERR = 1
EDGE = 8HTAILING
ISEC = ITE - 1
GO TO 750
C BAD EDGE DEFINITION
730 IERR = 2
EDGE = 8H LEADING
ISEC = ILE - 1
GO TO 750
740 IERR = 2
EDGE = 8HTAILING
ISEC = ITE - 1
C
750 CONTINUE
WRITE(N6,7500) SURF, EDGE
7500 FORMAT(220H*** ERROR PROCESSING A5,10HGEOMETRY, ,A8, 8H EDGE ***)
GO TO (755,780,800) IERR
755 WRITE(N6,7550) ISEC
7550 FORMAT(5X, 8HSECTION ,12,24H IS BEYOND THOSE DEFINED )
GO TO 800
780 WRITE(N6,7600) ISEC
7600 FORMAT(5X, 8HSECTION ,12,38H OF THE EDGE DOUBLES BACK TOWARD THE
1 12H CENTER LINE )
GO TO 800
C PLANFORM EXCEEDS BOX PATTERN LIMIT
770 CONTINUE
.IERR = 3
EDGE = 8HTAILING
ISEC = ITE - 1
WRITE (N6,7700) ISEC,SURF,CHORD,LEN
7700 FORMAT(34H*** ERROR - SECTION ,12,24H IS THE TRAILING EDGE OF THE
1 A5,14H CAUSES CHORD 13,15H TO GO TO ROW 13,15H, WHICH EXCEEDS
2 14H THE LIMIT ***)
LEN = LSR46
GO BACK TO FINISH THE SURFACE, THEN PRINT PLANFORM AND FLUSH
GO TO 800
C PLANFORM EXCEEDS BOX PATTERN LIMIT
800 CALL 'PRINTB ((BOX,LSROWS, I5B,MSER, NSHDR, 1)),
800 Call 'FLUSH
C
END
SUBROUTINE BXCDI(IWKE,LSRO6,LSODS,IUCX)

DIMENSION IWAKE(1),IBOX(LSRO6,1)

C DETERMINES BOX CODES FOR DIAPHRAGM REGIONS
C
C IWAKE - ARRAY OF WAKE LIMITS, AS DICTATED BY A TAIL SURFACE
C LSRO6 - ROW DIMENSION OF THE BOX CODE ARRAY
C LSODS - MAXIMUM NUMBER OF BOX CODES ALLOWED PER ROW
C IBOX - BOX CODE ARRAY, COMPRESSED TO 20 CODES PER WORD
C
C COMMON /FILES/ NT5, NT6, INTAPE, INFSP, NPLAIIC, NPSAIC, NOUTP,
C 1 IOUTSP, MXESC, TVPSC, IGEOSC, IMTESC, IAILCSC
C COMMON /GEOMY/ COPLAN, NSUBDV, XSUBDV, NSUBCN, NSURF,
C 1 BI, BIBETA, BIS, BIBTA, WLA, WLA, PS1W,
C 2 MXBBW, MXBBW, MXBW, MYBBW, MYBBW, MYBBW,
C 3 IX5B, IXCEM
C LOGICAL COPLAN
C COMMON /GEOME/ TLAX, TLAZ, PS1T, MXBT, MYBT, MYBT, MXBT, MYBT,
C 1 MXBBT, MXBBT, MXBBT, CARS,
C COMMON /SAMFW/ ISAMFL, ISAMFW, IXCOD(IID), IBOXF(IID), IBOXL(IID), IBOXR(IID)
C COMMON INPUT VALUES-
C 1 MXBBW, MXBBW, IXBST, MXBT, MYB5W, MYB5W, MYB5W, MYB5W, NSUBDV
C ISAMFW
C COMMON OUTPUT VALUES-
C 1 MXBBW, MXBBW, MYB5W, MYB5W, MYB5W
C DIMENSION ICODE(IID)
C LOGICAL WING
C
C DETERMINE WHETHER THIS IS A WING OR TAIL
C IF (IWAKE(I) .EQ. 0) GO TO 80
C WING = .T.
C IXBS = 1
C IXBSI = 2
C MXBBW = MXBBW+JUBDV * IXW - NSUBCN
C IF (ICPLAN) MXBBW = MXBST
C JECLOC = 1
C MXB5W = MXB5W
C MXB5 = MYB5W
C GO TO 100
C
C THIS IS A TAIL SURFACE
C 80 WING = .F.
C IXBS = 1DST
C IXBSI = IXBS + 1
C MXBBW = MXBST
C JECLOC = MXBBW + 1
C MXB5W = MXB5W
C MXB5 = MYB5W
C 100 CONTINUE
C
C DETERMINE LEADING EDGE DIAPHRAGM
C DO 130 J = 2,IXBS
C CALL DECODER(IX5K,LSRO6, IXBS,J-1, IXBS,J-1, .T., ICODE(I))
C DO 120 J = IXBSI, IXBS,
C CALL DECODER(IX5K,LSRO6, J-1, J-1, .T., ICODE(I))
C ICODE(I) = ICODE(I)
C ICODE(2) = ICODE(2)

C

B55
CALL DCDER(IBO,LRSOWS, I+1,J-1, I+1,J-1, .T., ICODE(3))
IF (ICOD .NE. 0) GO TO 120
IF (ICODE(1) .EQ. 0 .OR. ICODE(3) .EQ. 0) GO TO 120
CALL DCDER(IBO,LRSOWS, I,J, J,J)
120 CONTINUE
130 CONTINUE

C END OF DOUBLE LOOP TO DETERMINE LEADING EDGE DIAPHRAGM AREAS
INK = 0
DO 180 J = 1,MYBS
IF (INK .NE. 0) J = IMAKE(J)
CALL DCDER(IBO,LRSOWS, IXBS,J, IXBS,J, .T., ICODE)
DO 170 I = IXBS,MYBS
ICODM = ICODE
CALL DCDER(IBO,LRSOWS, I,J, I,J, .T., ICODE)
IF (ICOD .NE. 1) GO TO 170
IF (ICODM .NE. 1 .AND. ICODEM .NE. 3) GO TO 170
IF (I .LE. INK) GO TO 180
IF (I .EQ. IXBS) GO TO 180

C THE BOX IS A CANDIDATE. SEARCH DIAGONALLY FOR POSSIBLE
C RECEIVING BOXES DOWNSTREAM.
JP = J
JM = J
IS = I+1
DO 190 II = IS,MYBS
IF (JM .GT. 1) JM = JM - 1
CALL DCDER(IBO,LRSOWS, II,JM, JM, .T., ICODE)
IF (ICOD .NE. 0) GO TO 190
IF (JP .GE. MYBS) GO TO 190
JP = JP + 1
CALL DCDER(IBO,LRSOWS, II,JP, II,JP, .T., ICODE)
IF (ICOD .NE. 0) GO TO 190
IF (.NOT. JM) GO TO 190
IF (II .LE. IMAKE(JM) ) GO TO 190
IF (II .LE. IMAKE(JP) ) GO TO 190
190 CONTINUE

C END OF LOOP ON DIAGONAL SEARCH
GO TO 170

C CONDITIONS HAVE BEEN FOUND FOR A VALID WAKE BOX
180 CONTINUE
CALL NCDER(IBO,LRSOWS, I,J, I,J)
ICOD = 3
170 CONTINUE

C END OF LOOP ON ROWS, AND
160 CONTINUE

C END OF LOOP ON CHORDS, FOR WAKE DIAPHRAGM, FROM 130
C DETERMINE THE TIP DIAPHRAGM REGION
LBB = 2
DO 300 J = 1,MYBS
C SEARCH FOR LAST NON-ZERO BOX CODE ON THE ROW, FROM LBB OUTWARD
CALL DCDER(IBO,LRSOWS, J,LBB, J,MYBS, .T., ICODE(LBB) )
300 CONTINUE

200 LBB = LBB+1
DO 2 J = LBB+1,MYBS
IF (ICOD(J) .EQ. 0) GO TO 220

B56
210 CONTINUE
  LBB = MYBBS
  GO TO 225
220 LBB = J - 1
C LBB = THE SUBSCRIPT OF THE LAST NON-ZERO BOX ON THE ROW
C TEST BOX DIRECTLY AHEAD FOR NON-ZERO VALUE
225 CONTINUE
  CALL DCORDER (IBOX, LSRQBS, I - 1, LBB, I - 1, LBB, ICOD)
  IF (ICOD .EQ. 0) GO TO 280
C SEARCH FOR A NON-ZERO CODE ON THE INBOARD DIAGONAL AFT
  J = LBB
  IPL = I + 1
  IF (IPL .GT. MYBBS) GO TO 310
  DO 250 II = IPL, MYBBS
  CALL DCORDER (IBOX, LSRQBS, II, J, II, J, ICOD)
  IF (ICOD .NE. 0) GO TO 255
  J = J - 1
250 CONTINUE
C NO DIAGONAL BOX WAS FOUND
  IF (NBURF .EQ. 1 .AND. IMPLW .EQ. 0) GO TO 310
  IF (.NOT. WENG) GO TO 310
C SEARCH BACK ALONG THE DIAGONAL FOR A DIAPHRAGM REGION
C CAUSED BY A TAIL SURFACE
  II = MYBBS + 1
  DO 240 III = IPL, MYBBS
  II = II - 1
  J = J + 1
  IF (ILK/J) .GE. II ) GO TO 250
240 CONTINUE
C NO DIAPHRAGM FOUND
  GO TO 310
C CONDITION FOUND REQUIRING DIAPHRAGM BOXES ON THE DIAGONAL.
250 JJ = J
  GO TO 260
255 JJ = J + 1
  II = II - 1
270 CONTINUE
C TEST FOR EXCEEDING BOX CODE ARRAY
  LBB = LBB + 1
  IF (LBB .GT. LSCHDS) GO TO 8500
  IF (SCHDS .LT. LBB) GO TO 210
C SET DIAGONAL ELEMENTS
  DO 270 J = JJ, LBB
  CALL DCORDER (IBOX, LSRQBS, II, J, II, 2)
  II = II - 1
270 CONTINUE
C ICOD (LBB) = 2
  MYBBS = MAXD (MYBBS, LBB)
  GO TO 260
C DETERMINE LAST NON-ZERO BOX ON NEXT ROW
280 CONTINUE
  MYBBS = MAXD (MYBBS, LBB)
  II = I - 1
  DO 290 K = II, I
    J = LBB - K + 1
  290 CONTINUE

B57
CALL DECODER (IBOX,LSROWS, II,J, II,J, .T., ICOB)
IF (ICOB .NE. 0) GO TO 295
290 CONTINUE
GO TO 300
295 LBB = J
300 CONTINUE
C END OF LOOP ON ROWS DETERMINING TIP DIAPHRAGM CODES, FROM 180°
C
310 CONTINUE
IF (MENG) GO TO 350
WBBST = MYBBS
WBBT = (MYBBS + NSUB21)/NSUBDV
GO TO 500
350 WBBBSW = MYBBS
WBBW = (MYBBS + NSUB21)/NSUBDV
IF (.NOT. COPLAN) GO TO 500
WBBST = WBBBSW
WBBT = WBBW
C
500 RETURN

550 RETE (INT6.9500)
9500 FORMAT (51X ERROR - TOO MANY CHORDS FOR BOX CODE ARRAY **** )
CALL PRINTBC (IBOX,LSROW,S,IXBS,MYBBS,MYBBS,.T.)
CALL FLUSH(1)
C END
SUBROUTINE GMAREA(IBOX,LBXCD,WING,ALPHA,IJALPH,NALPH)  
GMAREA 00002
GMAREA 00003
GMAREA 00004
GMAREA 00005
GMAREA 00006
GMAREA 00007
GMAREA 00008
GMAREA 00009
GMAREA 00010
GMAREA 00011
GMAREA 00012
GMAREA 00013
GMAREA 00014
GMAREA 00015
GMAREA 00016
GMAREA 00017
GMAREA 00018
GMAREA 00019
GMAREA 00020
GMAREA 00021
GMAREA 00022
GMAREA 00023
GMAREA 00024
GMAREA 00025
GMAREA 00026
GMAREA 00027
GMAREA 00028
GMAREA 00029
GMAREA 00030
GMAREA 00031
GMAREA 00032
GMAREA 00033
GMAREA 00034
GMAREA 00035
GMAREA 00036
GMAREA 00037
GMAREA 00038
GMAREA 00039
GMAREA 00040
GMAREA 00041
GMAREA 00042
GMAREA 00043
GMAREA 00044
GMAREA 00045
GMAREA 00046
GMAREA 00047
GMAREA 00048
GMAREA 00049
GMAREA 00050
GMAREA 00051

IBOX = ARRAY OF BOX TYPE CODES
LBOX = ROW DIMENSION OF BOX CODE ARRAY
WING = PLANFORM INDICATOR
ALPHA = ARRAY OF ALPHAS (NORMALIZED AREAS)
IJALPH = SUBSCRIPTS FOR IBOX ARRAY OF CELLS THAT HAVE ALPHAS
NALPH = NUMBER OF ALPHAS STORED

COMMON /GEOMY/ COPLAN,NSUBDY,NSUBBY,NSUBD2,NSUBCN,NSURF,
1 BI,BIBETA,BIS,BIBTAS,WLAX,WLAZ,PSIW,
2 MOW,MOWB,MYBN,MYBB,MYBSW,MYBBW,MYBSW,
3 NWD,EXENTR
LOGICAL COPLAN
COMMON /GE02/ TLX,TLZ,WBT,MBT,MYB,MBY,MYB,MYB,
1 WMBS,WMBT,MBYS,MBTY,
COMMON /PLANXY/ NLE,NTE,XLE,NTE,YLE(Id),YLE(Id),
1 YLE(Id),YLE(Id), XLE(Id),YLE(Id),
2 XLE(Id),YLE(Id)
COMMON /EDGES/ PXLOC(325), TXYLOC(325), JDIAG

COMMON PARAMETERS USED
WDB = LENGTH OF BOX PATTERN (X-DIRECTION)
WDB = MAXIMUM ON-PLANFORM SPAN (Y-DIRECTION)
CPOLAN = .T., SECOND SURFACE EXISTS FOR PLANFORM
WLLE = NUMBER OF POINTS DEFINING LEADING EDGE OF THE WING
WLTE = NUMBER OF POINTS DEFINING TRAILING EDGE OF THE WING
WNTLE = NUMBER OF POINTS DEFINING LEADING EDGE OF THE TAIL
WNTTE = NUMBER OF POINTS DEFINING TRAILING EDGE OF THE TAIL
XLE = X COORDINATE OF THE LEADING EDGE DEFINITION POINT
YLE = Y COORDINATE OF THE LEADING EDGE DEFINITION POINT
XME = X COORDINATE OF THE TRAILING EDGE DEFINITION POINT
YME = Y COORDINATE OF THE TRAILING EDGE DEFINITION POINT
XTE = X COORDINATE OF THE TRAILING EDGE DEFINITION POINT
YTE = Y COORDINATE OF THE TRAILING EDGE DEFINITION POINT
XLTE = X COORDINATE OF THE LEADING EDGE DEFINITION POINT
YLTE = Y COORDINATE OF THE LEADING EDGE DEFINITION POINT
XLTE = X COORDINATE OF THE TRAILING EDGE DEFINITION POINT
YTE = Y COORDINATE OF THE TRAILING EDGE DEFINITION POINT

COMMON /LAREA/ LEFT,RIGHT,ICODE
DIMENSION IBX(50)
DIMENSION ALPHA(1), IJALPH(1)
LOGICAL WING
REAL LINL,LI,N,L,LMT,LEFT
NALPH = 1
IF (WING) GO TO 5
WDB = MBD
INB = (IBX-1)XBD/NSUBBV + 1

B59
$M_3 = M_3 T$

$E + R = 0.04 A R E$

$M_4 = M_4 T$

IF (CPLAN) $M_3 = M_3 T$

$M_5 = M_5 T$

IF (CPLAN) $M_3 = M_3 T$

B CONTINUE

C

LOOP ON CHORDS

DO 90 J=1,M3B

YJ = J

RIGHT = YJ + 0.5

LEFT = YJ - 0.5

IF (.NOT. WING) GO TO 110

C

CALL MTRCEP TO DETERMINE LEFT, RIGHT AND CENTER LINE

C INTERCEPTS, AND THE BREAK POINTS OF EDGES OVER THIS

C CHORD.

C

IF (YJ .GT. YMLE(NMLE)) GO TO 10

CALL MTRCEP(J, YMLE, YMLE, LIM, CENL, RINL, NBK1, KINK1, 1)

CALL MTRCEP(J, YMLE, YMLE, LIM, CENL, RINL, NBK2, KINK2, 2)

10 IF (SURF .EQ. 1 .OR. .NOT. CPLAN) GO TO 20

C COMPUTE SLOPE AND INTERCEPTS FOR SECOND PLANFORM.

110 IF (YJ .GT. YTLE(NMLE)) GO TO 20

CALL MTRCEP(J, YTLE, XTLE, LIM, CEN4, RIN4, NBK4, KINK4, 2)

C SLOPE AND INTERVALS COMPUTED.

C

LOOP DOWN THE CHORD

CALL DOCORD(IBX, LBCD, IMB, X3B, J, M3B, J, .F., IBX)

II = 1

DO 85 I=1,X3B

XI = I

IF(IBX(II) .NE. 1) GO TO 80

BMLE = XI - 0.5

BMTE = XI + 0.5

IF(.NOT.WING) GO TO 40

IF (YJ .GT. YMLE(NMLE)) GO TO 40

IF (XI .GT. CEN3) GO TO 40

C BOX IS ON PLANFORM 1

ICODE = 1

C

ICODE = 1, 1ST L.E. BOX ON CHORD

C = 2, LAST T.E. BOX ON CHORD

C = 3, INTERNAL CUT BOX

IF(II .EQ. 1) GO TO 24

IF(IBX(II-1) .NE. 1) GO TO 24

ICODE = 2

IF (J .EQ. M3B) GO TO 24

IF (.NOT. WING) GO TO 40

IF (YJ .GT. YMLE(NMLE)) GO TO 40

IF (XI .GT. CEN2) GO TO 40

ICCC = 3

IU = 0

IB = 0

B60
IF(RINE.GT.BOXC10) IU=1
IF(RINE.LT.BOXC10) IB=1
IF(LINE.LT.BOXC10) IB=1
IF (LINE.EQ.0) GO TO 80

C BOX IS NOT ENTIRELY ON PLANFORM. COMPUTE AREA.
24 CONTINUE
CALL ALPHAC(XI, XILE, YILE, XITE, YITE,
LINE, CENE, RINE, NBK1, KINK1, LINE, CENE, RINE, NBK2, KINK2, ALPH (NALPH))
GO TO 75

C 40 CONTINUE
IF (NSURF.EQ.1) GO TO 80
C BOX IS ON PLANFORM 2.
ICODE =2
IF(I.EQ.iW) GO TO 44
ICODE = 1
IF(XI-I .LT.CEN) GO TO 44
ICODE =2
IF(IBX(I).LT.1) GO TO 44
ICODE =3
I=0
IB=0
IF(RINE.GT.BOX10) IU=1
IF(RINE.LT.BOX10) IB=1
IF(LINE.LT.BOX10) IB=1
IF (LINE.EQ.0) GO TO 80

C BOX IS NOT ENTIRELY ON PLANFORM. COMPUTE AREA.
44 CONTINUE
CALL ALPHAC(XI, XTLE, YTLE, XITE, YITE,
LINE, CENE, RINE, NBK3, KINK3, LINE, CENE, RINE, NBK4, KINK4, ALPH (NALPH))
GO TO 75

75 NALPH (NALPH) = J512 + 1
NALPH = NALPH + 1
80 CONTINUE
II = II + 1
85 CONTINUE
90 CONTINUE
NALPH = NALPH -1
RETURN
END
SUBROUTINE ALPHAC(X, XLED, YLED, XTED, YTED, L1, C1, R1, L2, C2, R2, NBK, AREA)
C
X = X COORDINATE OF CELL CENTER
L1 = X COORDINATE OF LEADING EDGE LEFT SIDE INTERSECTION
C1 = X COORDINATE OF LEADING EDGE CENTER LINE INTERSECTION
R1 = X COORDINATE OF LEADING EDGE RIGHT SIDE INTERSECTION
K1 = FLAG TO INDICATE LEADING EDGE KINK
L2 = X COORDINATE OF TRAILING EDGE LEFT SIDE INTERSECTION
C2 = X COORDINATE OF TRAILING EDGE ENTER LINE INTERSECT.
R2 = X COORDINATE OF TRAILING EDGE RIGHT SIDE INTERSECTION
K2 = FLAG TO INDICATE TRAILING EDGE KINK
AREA = AREA COMPUTED FOR THE CELL
C
COMMON /AREA / LEFT,RIGHT,ICODE
C
LEFT = Y COORDINATE OF LEFT SIDE OF CHORD
RIGHT = Y COORDINATE OF RIGHT SIDE OF CHORD
ICODE = 1, 1ST L.E. BOX ON CHORD
ICODE = 2, LAST T.E. BOX ON CHORD
ICODE = 3, INTERNAL CUT BOX
C
DIMENSION X(6), Y(6)
DIMENSION XLED(1), YLED(1), XTED(1), YTED(1)
REAL LEFT, L1, L2
EPS = 1.0E-04
BOSLE = X-0.5
BOTE = X + 0.5
XU = X - 1.0
XL = X + 1.0
AREA = 0.0
ISICE = 0
IF ICODE.EQ.3 GO TO 5000
IF (C1.GT.) AND (C2.LT.) GO TO 3000
1110 IF ICODE.EQ.1 GO TO 1000
1120 IF ICODE.EQ.2 GO TO 2000
GO TO 4000
C
C LEADING EDGE BOX
1000 CONTINUE
MTRAPS = NBK + 1
MTR = MTRAPS - 1
NAK = MTRAPS + 1
X(1) = L1
Y(1) = LEFT
X(NAK) = R1
Y(NAK) = RIGHT
IF (MTRAPS.EQ.1) GO TO 110
DO 100 NM=2,MTRAPS
KIDX = K1+NM-2
X(NM) = XLED(KIDX)
Y(NM) = YLED(KIDX)
100 CONTINUE
110 CONTINUE
DO 300 NM = 1,MTRAPS
IF (XC(NM) .GE. BOSLE) GO TO 300
A = BOSLE - XC(NM)
AY = YC(NM+1) - YC(NM)
B62
IF(DY.LT.EPS) GO TO 300
IF(XC(NX+1).GT.BOLE) GO TO 250
B = BOLE - XC(NX+1)
200 AREA = AREA + 0.5*(A+B)*DY
GO TO 300

C
EDGE CROSSES BOLE. COMPUTE INTERSECTION FOR DY
250 CONTINUE
B = 0.0
DX = XC(NX+1) - XC(NX)
S = DY/DX
DY = B*DX
GO TO 200
300 CONTINUE
IF (BOLE-BOLE.LT.EPS) GO TO 2000
IF(R2.LT.BOLE) ISLICE = 2
IF(L2.LT.BOLE) ISLICE = 1
IF(ISLICE.NE.0) GO TO 5000
GO TO 4000

C
TRAILING EDGE BOX
2000 CONTINUE
NTRAPS =NTRAPS + 1
NTH = NTRAPS- 1
MC = NTRAPS+ 1
X(1) = L2
YC(1) = LEFT
X(NC) =R2
YC(NC) =RIGHT
IF(NTRAPS.EQ.1) GO TO 2110
DO 2100 NA = 2,NTRAPS
KIDX = K2 + NA -2
XCN) = XTED(KIDX)
YC(NA) = YTED(KIDX)
2100 CONTINUE
2110 CONTINUE
DO 2300 NX=1,NTRAPS
IF(XC(NX).LT.BoLE) AND XC(NX+1).GT.BoLE) GO TO 2300
DY = YC(NX+1) - YC(NX)
IF(DY.LT.EPS) GO TO 2300
IF(XC(NX).LT.BoLE) OR Xc(NX+1).GT.BoLE) GO TO 2250

C
DOES NOT INTERSECT BOLE
A = XC(NX) - BOLE
B = XC(NX+1) - BOLE
2225 AREA = AREA + 0.5*(A+B)*DY
GO TO 2300

C
INTERSECTS BOLE
2250 CONTINUE
DX = XC(NX+1) -XC(NX)
S = DY/DX
A = BOLE - XC(NX)
YINT = YC(NX) + B*DX
IF( S.LT.0.0) GO TO 2275

C
SLOPE POSITIVE
A = 0
B = X(NX+1) - BOXLE
DY = YC(NX+1) - YINT
GO TO 2225
C
C SLOPE NEGATIVE
2275 CONTINUE
A = -A
B = 0
DY = YINT - YC(NX)
GO TO 2225
2300 CONTINUE
IF(R1.LT.BOLE) ISLICE =3
IF(ISLICE.NE.0) GO TO 5000
GO TO 4000
C
C CHORD HAS ONLY 1 BOX
3000 CONTINUE
BOXLE = R1
BOXTE = R1
GO TO 1000
C
C THE FOLLOWING IS THE CALCULATIONS FOR A BOX WITH FORE AND
C AFT BOXES ON PLANFORM. ONE OF 3 CORNER (L.L.,L.R., OR U.B.)
C IS CUT OFF.
9000 CONTINUE
IF(ISLICE.NE.0) GO TO 5005
AREA = (RIGHT-LEFT)
9005 CONTINUE
TA = 0.0
IF(ISLICE.EQ.3) GO TO 5020
IF(L2.LT.BOXTE) GO TO 5100
9010 IF(R2.LT.BOXTE) GO TO 5102
9020 CONTINUE
IF(ISLICE.GT.1.OR.ISLICE.LE.2) GO TO 5400
IF(R1.GT.BOLE) GO TO 5300
GO TO 5400
9100 ITAG = 1
I = 1
XC(I) = L2
YC(I) = LEFT
GO TO 5110
9102 ITAG = 2
I = 5
XC(I) = R2
YC(I) = RIGHT
9110 I = I+1
IF(NBK2.EQ.0) GO TO 5150
KIDX = K2 + 1 - 2
IF(I.EQ.2) KIDX = K2 + NBK2 + 1 - 2
IF(XYC(KIDX),GY,BOXTE) GO TO 5125
XC(I) = XYC(KIDX)
YC(I) = YTEC(KIDX)
GO TO 5110
5125 CONTINUE
IF(ITAG.EQ.2) KIDX=KIDX+1

C(1) = BOKTE

DX = XTED(KIDX) - XTED(KIDX-1)

DY = YTED(KIDX) - YTED(KIDX-1)

IF(ITAG.EQ.2) GO TO 5130

C

LOWER LEFT CORNER

DIST = BOKTE - XC(I-1)

YC(I) = YTED(KIDX-1) + DIST*(DY/DX)

GO TO 5160

C

LOWER RIGHT CORNER

5130 DIST = BOKTE - XC(I-1)

YC(I) = YC(I-1)+ DIST* (DY/DX)

GO TO 5160

5150 CONTINUE

XC(1) = BOKTE

DX = R2 - L2

DY = RIGHT - LEFT

IF(ITAG.EQ.2) GO TO 5155

DIST = BOKTE -L2

YC(I) = LEFT + DIST* (DY/DX)

GO TO 5160

5155 DIST = BOKTE -R2

YC(I) = RIGHT* DIST* (DY/DX)

5160 CONTINUE

NTRAPS = I-1

DO 5175 NX=1,NTRAPS

A = BOKTE - XC(NX)

B = BOKTE - XC(NX+1)

IF(A.GT.1.) A = 1.0

IF(B.GT.1.) B = 1.0

H = YC(NX+1) - YC(NX)

IF(ITAG.EQ.2) H = -H

TA = TA + 0.5*(A+B)*H

5175 CONTINUE

IF(ITAG.EQ.2) GO TO 5020

GO TO 5010

C

COMPUTE FOR UPPER RIGHT HAND CORNER

5300 CONTINUE

I = 1

XC(I) = R1

YC(I) = RIGHT

5310 I = I+1

IF(NBK1.EQ.0) GO TO 5350

KIDX = K1 + NBK1-1 - I +2

IF(XLED(KIDX).LT.BOKLE) GO TO 5325

XC(I) = XLED(KIDX)

YC(I) = YLED(KIDX)

GO TO 5310

5325 CONTINUE

XC(I) = BOKLE

DX = XLED(KIDX+1)-XLED(KIDX)

DY = YLED(KIDX+1)-YLED(KIDX)

DIST = BOKLE - RLED(KIDX)

YC(I) = YLED(KIDX) + DIST* (DY/DX)

B65
GO TO 5360
5350 CONTINUE
X(1) = Boole
DX = R1 - L1
DY = RIGHT - LEFT
DIST = Boole - L1
Y(1) = LEFT + DIST*(DY/DX)
5360 CONTINUE
NTRAPS = I-1
DO 5375 NX = 1, NTRAPS
A = X(NX) - Boole
B = X(NX+1) - Boole
IF(A.LT.1. AND. B.LT.1.) GO TO 5370
IF(A.GT.1. AND. B.GT.1.) GO TO 5365
C
C A.GT.BI AND B.LT.BI
DX = X(NX) - X(NX+1)
DY = YC(NX) - YC(NX+1)
IF(DY.LT.EPS) GO TO 5375
DIST = Boole - X(NX+1)
YINT = YC(NX+1) + DIST*(DY/DX)
TA = (YC(NX)-YINT) + TA
YC(NX) = YINT
A = 1.0
GO TO 5370
5365 CONTINUE
TA = TA + (YC(NX)-YC(NX+1))
GO TO 5375
5370 CONTINUE
H = YC(NX) - YC(NX+1)
TA = TA + 0.5*(A+B)*H
5375 CONTINUE
5400 AREA = AREA - TA
4000 CONTINUE
RETURN
END
SUBROUTINE NTRCEP(J, YEDG, XEDG, L1, CI, RI, NBX1, KI, IDEX)

C
C J = INDEX OF CHORD NUMBER
C NBX1 = NUMBER OF BREAK POINTS ON EDGE FOR THIS CHORD.
C L1 = X COORDINATE OF LEADING EDGE LEFT SIDE INTERSECTION
C CI = X COORDINATE OF LEADING EDGE CENTER LINE INTERSECTION
C RI = X COORDINATE OF LEADING EDGE RIGHT SIDE INTERSECTION
C KI = INDEX OF XEA AND YEA ARRAYS THAT DEFINE A KINK IF
C ONE EXISTS

COMMON /AREA / LEFT,RIGHT,ICODE
REAL LEFT,LI,L2
C LEFT = Y COORDINATE OF LEFT SIDE OF CHORD
C RIGHT = Y COORDINATE OF RIGHT SIDE OF CHORD
DIMENSION XEDG(1), YEDG(1)

YJ = J
EPS = 1.0E-04
K=2
1 IF(LEFT,LI,YEDG(K)-EPS) GO TO 2
K= K+1
GO TO 1
2 DX = XEDG(K) -XEDG(K-1)
DY = YEDG(K) - YEDG(K-1)
DIST = LEFT - YEDG(K-1)
LI = XEDG(K-1) + (DX/DY) * DIST
C
C FIND CENTER AND BEGIN COUNTING BREAKS
NBX1 = 1
KI = 0
3 IF(YJ .LT.YEDG(K) +EPS) GO TO 4
C
C KINKS BETWEEN LEFT AND CENTER LINE
IF(KI.EQ.0) K1 = K
NBX1 = NBX1 + 1
K = K + 1
GO TO 3
4 IF(IDEX.EQ.1) GO TO 40
IF(ABS(YEDG(K)-YJ) .GT.EPS) GO TO 40
IF(KI.EQ.0) K1 = K
NBX1 = NBX1 + 1
IF(YEDG(K+1)- YEDG(K),GT,EPS) GO TO 104
NBX1 = NBX1 + 1
K = K + 1
104 CONTINUE
C CI = XEDG(K)
GO TO 5
40 DX = XEDG(K) - XEDG(K-1)
DY = YEDG(K) - YEDG(K-1)
DIST = YJ - YEDG(K-1)
CI = XEDG(K-1) + (DX/DY) * DIST
C
C FIND K IN SAME MANNER AS CENTER
5 IF(RIGHT,LI,YEDG(K)+EPS) GO TO 6
C
C KINKS BETWEEN CENTER LINE AND RIGHT SIDE OF CHORD
IF(KI.EQ.0) K1 = K
NBX1 = NBX1 + 1
K = K + 1

B67
GO TO 5
6 DX = XEDG(K) - XEDG(K-1)
   DY = YEDG(K) - YEDG(K-1)
   DIST = RIGHT - YEEDG(K-1)
   R1 = XEDG(K-1) + (DX/DY) * DIST
RETURN
END
SUBROUTINE PA4IC(WING,IBOX,LBOX,IMAKE,JCCL)

COMPUTES THE POINTER ARRAY (MUAC) FOR THE SPATIAL AIC ARRAY
OF THE LEFT WING (TAIL) ON ONE CHORD OF THE RIGHT WING (TAIL)
WING = WING/TAIL INDICATOR
IBOX = BOX CODE ARRAY TO USE
IMAKE = ARRAY OF MAKE EDGE LOCATIONS FOR WING
JCCL = THE (UNSUBDIVIDED) CHORD NUMBER OF INTEREST
SURF = INDICATOR OF WHETHER ANY LEFT SURFACE IS INTER-
CEPTED BY THE MACH CONE FOR THIS CORD

COMMON /GEO4Y/ COPLAN,NSUBVX,KSUBV,NSUBCN,NSURF,
1 BI,BIBETA,BJS,BIBITAS,MLAX,MLAZ,PSIW,
2 WBBW,WOBBW,MYBW,MYBBW,MYBZW,MYBBZW,
3 IXBW,IXCENTR
LOGICAL COPLAN
COMMON /GEO2/ TLAX,TLAZ,PSIT,MYBT,MYBT,MYBTBT,MYBST,MYBST,
1 WBBST,IXBT,IXBST,CAPL
COMMON /MUAC/ YBAR,EL,MUACL(2,50),NRMB,SURF,
1 YBARL,ELL,MUACL(2,50),NRMSL,SURF,PSIDF
LOGICAL SURF,SURFL
COMMON /EDGES/ PEOLOC(250),TEXLOC(250),JDIAG

LOGICAL WING
DIMENSION IBOX(LBOX,8),ICODE(50),IMAKE(1)
DATA EPS / 1.D-E-4 /

IF (WING) GO TO 100
THE CALL IS FOR A TAIL CHORD
PB12 = PSIT + PSIT
IXB = (IXBT-IXBW)/NSUBV + 1
WBB = MYBT
IF (JCCL .LE. MYBT) GO TO 120
THE CHORD IS ON THE TIP DIAPHRAGM
IFIRST = IXB
NJ = MYBT-IFIRST+1
GO TO 130
THE CALL IS FOR A WING CHORD
100 CONTINUE
PB12 = PSIW + PSIW
IXB = 1
IF (COPLAN) GO TO 110
WBB = MYBW
IF (JCCL .GT. MYBT) GO TO 115
SUBV = JCCL*NSUBV - NSUB2
GO TO 125
THE CALL IS FOR A COPLANAR WING-TAIL
110 CONTINUE
WBB = MYBT
IF (JCCL .LE. MYBT) GO TO 120
THE CHORD IS ON THE TIP DIAPHRAGM
115 CONTINUE
IFIRST = 1
B69
NJ = N08
GO TO 150
C THE CHORD IS ON PLAN
120 CONTINUE
ISUB = MBMS + JCCL - MBDY - NSUBD2
C 125 CONTINUE
IFIRST = (16 + (15 + 1) * DM) / NSUBD2 + 1
NU = NU - IFIRST + 1
C
130 CONTINUE
CALL DDECOD (IBOX, LBOX, IFIRST, JCOL, MB8, JCOL, .F., ICODE)
IRON = IFIRST
DO 135 IA = 1, NU
IF (ICODE(i) .EQ. 0) GO TO 140
IRON = IRON + 1
135 CONTINUE
IF (IA .GE. NU) GO TO 155
DO 145 IA = IA, NU
IF (ICODE(i) .EQ. 0) GO TO 150
IRON = IRON + 1
145 CONTINUE
IRON = IROW - 1
150 CONTINUE
IROW = IRON - 1
155 CONTINUE
NROWS = IRON - 18 + 1
C COMPUTE HORIZONTAL AND VERTICAL OFFSETS
YMUSD = (JCOL - .5) * COS(PS12)
C = Y-OFFSET ON THE SENDING SURFACE OF THE PROJECTION OF THE
RECEIVING CHORD
JBAR = IFIX(YMUSD) + 1
C CHORD CONTAINING YMUSD
YBAR = YMUSD - JBAR + .5
C DISTANCE FROM NEAREST SENDING CHORD CENTER TO PROJECTION
OF THE RECEIVING CHORD, POSITIVE RIGHT.
CL = (1.0 - .5) * SIN(PS12)
C VERTICAL SEPARATION BETWEEN THE SENDING PLANE AND THE
RECEIVING CHORD
IF (YBAR) 160, 165, 170
160 JMIN = JBAR - 1
NBOXES = 2
GO TO 180
165 JMIN = JBAR
NBOXES = 1
GO TO 180
170 JMIN = JBAR
NBOXES = 2
C
180 CONTINUE
JL = 1
SURF = .F.
C
START OF LOOP ON ROWS, FORWARD FROM RECEIVING BOX CENTER, TO
DEFINE THE RECEIVE ARRAY

B70
DO 280 I = 1, NROWS
XI = I - .5 - ABS(DL)
IF (XI .LE. EPS) GO TO 280
IF (JMIN .GT. 0) GO TO 280
C CENTER LINE HAS BEEN CROSSED, THEREFORE THERE MAY BE CONTRIBUTION FROM THE LEFT MENG FOR THIS ROW
   JM = -JMIN + 1
   JMM = JM - 1
   CALL Dcoder(INBO, LBOX, IRW, IJMM, .F., ICOD)
DO 240 J = JM, JMM
   IF (ICOD(JMM) .NE. 0) GO TO 250
   JMM = JMM - 1
240 CONTINUE
   NROWS = I - 1
   GO TO 290
C CONTRIBUTING BOXES HAVE BEEN FOUND FOR THIS ROW
250 CONTINUE
   SURF = .T.
   JL = J
   IF (YBAR .GE. 0) GO TO 255
   NUAIC(1, I) = NBOXES - JM + 1
   NUAIC(2, I) = NBOXES - JL + 1
   GO TO 270
255 CONTINUE
   NUAIC(1, I) = JL
   NUAIC(2, I) = JM
   GO TO 270
C CENTER LINE HAS NOT BEEN CROSSED
260 NUAIC(1, I) = 0
   NUAIC(2, I) = 0
C
270 CONTINUE
   NBOXES = NBOXES + 2
   JMIN = JMIN - 1
   IRW = IRW - 1
280 CONTINUE
C END OF LOOP FORWARD ON ROWS, FROM 180°
C
290 CONTINUE
RETURN
END
IF(.NOT.MYGEOM) GO TO 50
WRITE (NTS,7005)
RETURN
C
50 CONTINUE
IF(MYTYPE.EQ.3) GO TO 75
IF(NSURF.EQ.2.AND.MYTYPE.EQ.3) GO TO 75
WRITE (NTS,7010)
GO TO 125
C
75 CONTINUE
WRITE (NTS,7015)
CALL FLUSH(1)
C
100 CONTINUE
C
NMODES = 0
NSLOP = 0
READ (NTS,CARD(1))
125 CONTINUE
READ IGEOBC
C
READ PEXLOC AND TEALOC ARRAY FROM GEOMETRY SCRATCH FILE
NAME =GEFSLOC
CALL RDINIT
ITYPE = SHMIXED
MBS = 1
IF(.NOT.COPLAN.AND.NSURF.EQ.2) MBS = 2
CALL READX(IGEOBC,NREAD,RANDIN,NFS,MBS,LS,NMR,1,NED,ITYPE,1
LRS,FEALOC,M,N,PARM,IRR)
IF(IRR.NE.0) GO TO 6010
C
NAME =GHTEXLOC
CALL RDINIT
ITYPE = SHMIXED
CALL READX(IGEOBC,NREAD,RANDIN,NFS,MBS,LS,NMR,1,NED,ITYPE,1
LRS,TEALOC,M,N,PARM,IRR)
IF(IRR.NE.0) GO TO 6010
C
C
ZERO OUT THE ROW AND COL *INTER
DO 150 I = 1,400
IS(I) = 0
150 CONTINUE
C
C
DETERMINE STARTING BOXES AND NUMBER OF BOXES PER CHORD.
IYB1 = (NSURF+1.)/2
DO 300 MB=1,NSURF
IF(NSURF.EQ.2) GO TO 200
NC = 1
MCH = MB/M
NCF= 0
ICN = MB - IYB1
1KB = 1MB
GO TO 225
C
B73
200 CONTINUE
NC = MCBW + 1
NCH = MCBW + NC
MCF = MCBW * NSUBDV
IFBT = (MCB-1)BVD/NSUBDV + 1

225 CONTINUE
IYB = IYB1 + MCF
DO 230 J = NC, NCH
IS(J) = FENLOC(IYB) + 1.0
ITEI = TENLOC(IYB)
IF(NSUBDV,SB,1) GO TO 240
IS(J) = (IS(J)-ICM)/NSUBDV + 1
ITEI = (ITEI-IYB)/NSUBDV + 1

240 CONTINUE
MOA(J) = ITEI-IS(J) + 1
IYB = IYB + NSUBDV

250 CONTINUE
CALL ROPER
C
C FIND OVERLAP OF 2 PLANFORMS IF THEY ARE NON-COPLANAR
IOLAP = 0
BNPNT = MCBW + 1
IF(NSURF,EB,1) GO TO 325
IF(COPLAN) GO TO 324
IF(IFBT,GT,MCB) GO TO 324
IOLAP = MCB - IFBT + 1
BNPNT = MCB + IOLAP + 1
GO TO 325

324 CONTINUE
BNPNT = MCB + 1

325 CONTINUE
C
C \text{COMPUTE POINTER ARRAY AND STORE ON MODES}
C
REWD, MODES
C
C
IIPMNTRM(1,1) = 1
IIPMNTRM(2,1) = JS(1)
DO 320 I = 2, BNPNTS
IIPMNTRM(1,1) = IIPMNTRM(1,1-1) + JOC(I-1)
IIPMNTRM(2,1) = JS(I)

320 CONTINUE
CALL RDINIT
IPARM(3) = IOLAP
ITYPE = 5MIXED
CALL \text{WRITE}(MODES, MCBWIT, RANDOU, NFS, NSL, NSR, LUS,2.1D, 0)
\text{IPMTRM,ITYPE,2,NNPRNPS,PARM,IRR)}
IF(IRR.NE.0) GO TO 6030
C
C \text{FIRST LOOP DETERMINES MODE SHAPES.}
C
C \text{SECOND LOOP DETERMINES THICKNESS SLOPES.}
C
C
DO 30 IPASS=1,2
IF(I .LT. EPS.EQ.2) GO TO 2100
C
C \text{COUNT NUMBER OF SURFACES}
C
DO 20JO JS = 1, NSURF
IF(NS.EQ.0) GO TO 330

B74
**FILE = IVPSC**

**REWIND IVPSC**

**GO TO 340**

**330 CONTINUE**

**IFILE = MODESC**

**340 CONTINUE**

C

**LOOP ON NUMBER OF MODES**

**DO 150 NM = 1,NMOTES**

C

**ZERO OUT THE DEFSL ARRAY**

**DO 350 I = 1,NMODES**

DEFSL(I,1) = 0.0

DEFSL(2,1) = 0.0

**350 CONTINUE**

C

**ZERO OUT THE COEFFICIENT ARRAY**

**DO 355 I = 1,21**

A(I) = 0.0

**355 CONTINUE**

C

**INPUT FIRST PLANFORM IF THERE IS A TAIL SECTION**

**IF(NB.EQ.1) GO TO 400**

**READ (IVPSC) DEFSL**

C

**400 CONTINUE**

**IF(NF.EQ.,MODE) GO TO 450**

**CALL RDIMET**

**IF(NB.EQ.1.AND.MM.EQ.1) NFS = 2**

MM = 0H CCEP.

**CALL READXX(GEOEC,MOREAD,RANDIN,NFS,NMB,LS,MM,1,NDID,ID,ITYPE,**

1

URS,A,M,N,PANM,IIR)

**IF(IIR.NE.0) GO TO 6010**

**NFS = 0**

C

**GO TO 551**

**450 CONTINUE**

**ITYPE = WTYPEW**

**IF(NB.EQ.2) ITYPE = WTYPE**

**GO TO (501,502,503),ITYPE**

C

**READ IN POLYNOMIAL COEFFICIENTS**

**501 CONTINUE**

**READ(NTS,8010) IDEG**

**IF (IDEG .LT. 0 .OR. IDEG .GT. 5) GO TO 6000**

**8010 FORMAT(215)**

**MD = IDEG + 1**

**DEG = MD**

**D2G = DEG.2**

**NC = DEG+D2G + DEG + EPS**

**READ(NTS,8015) (A(I),I=1,NC)**

**8015 FORMAT(TE10.0)**

**IFLAG = 1**

**GO TO 530**

C
READ IN DEFLECTIONS AT SELECTED LOCATIONS AND FIT A POLYNOMIAL OF DEGREE IDEF TO THE POINTS USING METHOD OF LEAST SQUARES.

902 CONTINUE
READ (INTS,0010) IDEF,NPTS
IF (IDEG .LT. 0 .OR. IDEF .GT. 5) GO TO 8005
IF (NPTS .GT. 100 .OR. NPTS .LT. 1) GO TO 8005
READ (INTS,8020) (XX(I),YY(I),ZZ(I),I=1,NPTS)
8020 FORMAT(6E10.0)
IDIM = 1
CN = 1.0
C
CN IS A SCALE FACTOR TO REDUCE THE MAGNITUDE OF THE NUMBERS
C IDIM IS A DIMENSION VARIABLE SET TO 1 TO INDICATE FIT IS BEING MADE ON REAL VALUES . IDIM = 2 FOR COMPLEX Z VALUES.
C
CALL FITTER(IDEG,NPTS,XX,YY,ZZ,CA,CN,IDIM)

IFLAG = 2
MODES 00217
MODES 00218
MODES 00219
MODES 00220
MODES 00221
MODES 00222
MODES 00223
MODES 00224
MODES 00225
MODES 00226
MODES 00227
MODES 00228
MODES 00229
MODES 00230
MODES 00231
MODES 00232
MODES 00233
MODES 00234
MODES 00235
MODES 00236
MODES 00237
MODES 00238
MODES 00239
MODES 00240
MODES 00241
MODES 00242
MODES 00243
MODES 00244
MODES 00245
MODES 00246
MODES 00247
MODES 00248
MODES 00249
MODES 00250
MODES 00251
MODES 00252
MODES 00253
MODES 00254
MODES 00255
MODES 00256
MODES 00257
MODES 00258
C
C STORE THE COEFFICIENTS ON THE THIRD FILE OF THE IGEOSC FILE.
C IF THE COEFFICIENTS ARE TO BE PRINTED THE ONES FOR THE FIRST SURFACE MUST BE STORED ON A SCRATCH FILE TEMPORARILY.
C
CALL RDINIT
IF(NS.EQ.1.AND.NS.EQ.1) NFS = 2
IPARM(3) = IDEF
IPARM(4) = IFLAG
ITYPE = SIMIXED
CALL WRTEX(IAGEOSC,MAXIT,RANDOI,NFS,NHS,NHR,LWS,1,IT,
A,ITYPE,1,NC,PARMIRR)
IF(IRR.NE.0) GO TO 6050
NFS = 0
C
C IF(NS.PROOF) GO TO 3550
IF(NS.EQ.2.OR.NSURF.EQ.1) GO TO 3550
IF(NH.EQ.1) REWIND IAICSC
C
CALL WRTEX(IAIICSC,MAXIT,RANDOI,NFS,NHS,NHR,LWS,1,IT,
A,ITYPE,1,NC,PARMIRR)
IF(IRR.NE.0) GO TO 6060
C
3550 CONTINUE
IF(NS.EQ.2.AND.NH.EQ.1) REWIND IAICSC
C
C EVALUATE THE POLYNOMIAL EQUATION FOR DEFLECTIONS.
C THE PARTIAL DERIVATIVE WITH RESPECT TO X TO GET SLOPES.
C
C 551 CONTINUE
IF(NH.NE.1) GO TO 560
IF(NS.EQ.2) GO TO 556
550 CONTINUE
IF(NS.EQ.1) GO TO 560
IF(NS.EQ.2) GO TO 556
C
CALCULATE X,Y COORDINATES FOR EVALUATION OF POLYNOMIAL

XI(1) = XCENTR
Y1(1) = 0.5*BETA
IF(NSURF.EQ.2) GO TO 552
MMAX = MAX(XSBW,MYBW)
GO TO 554
552 CONTINUE
MMAX = MAX(XSBT,MYBT,MYBW)
554 CONTINUE
DO 555 I = 2,MMAX
XI(I) = XI(I-1)+B1
Y1(I) = Y1(I-1)+B1*BTA
555 CONTINUE
GO TO 560
556 CONTINUE
XADJ = TLAX - XLAX
DO 557 I=1,MMAX
XI(I) = XI(I) - XADJ
557 CONTINUE
GO TO 560
580 CONTINUE
IF(NS.EQ.2) GO TO 580
IC = 0
ILIM = NWB
IBEG = 1
MCH = 0
GO TO 564
560 CONTINUE
IBEG = IFBT
ILIM = XSBT
MCH = MYBW
IC = 0
IUP = XSBW
IF(COPLAN) IUP = IFBT-1
DO 563 I=1,IUP
IC = IC + JOC(I)
563 CONTINUE
963 CONTINUE
964 CONTINUE
DO 575 IX=IBEG,ILIM
I = IX
IF(NS.EQ.2) I = IX + IOMAP
XP(I) =1.
DO 581 IP=2,IPDEG
961 XP(IP) = XP(IP-1) + XI(IX)
JI = J8(1)
J1 = JOC(I)+ J1 -1
DO 570 J=JI,JI
IC = IC +1
IB = IB + MCH(J+MCH) -1
IF(IX.LT.IB) GO TO 570
IF(IX.GT.I1) GO TO 570
YP(I) =1.
B77
DO 562 JP = 2, MDEG

562 YP(JP) = YP(JP-1) * YI(J)
D = A(I)
S = 0.0
IF (MDEG .LT. 2) GO TO 567
IA = 1
DO 565 L2 = 2, MDEG
DO 565 L3 = 1, L2
L4 = L2 - L3 + 1
IA = IA + 1
D = D + XP(L4) * YP(L3) * A(IA)
IF(L4 .EQ. 1) GO TO 565
L3 = L4 - 1
S = S + L5 * XP(L5) * YP(L3) * A(IA)
565 CONTINUE
567 CONTINUE

BCSMDA 00008
IA = 1

565 CONTINUE
567 CONTINUE
DEFSL(1, IC) = D
DEFSL(2, IC) = S

C READ IN DEFLECTIONS AND SLOPES AT BOX CENTERS
503 CONTINUE
IF(INTPROF .LT. 0.0 OR INTPROF .GT. 5.0) GO TO 700
C MODES ON TAPE, CALL SPECIAL ROUTINE TO HANDLE.
CALL TAPMCO(NS, DEFSL)
GO TO 900
700 CONTINUE
IF(NS .GE. 2) GO TO 720
C FIRST PLANForm
MCH = MYBW
NC = 1
GO TO 725
720 CONTINUE
NC = MYBW + 1
MCH = MYBW + WBT
C READ AND STORE DEFLECTIONS
725 CONTINUE

DO 750 J = NC, MCH
IST = IS(J)
NK = MOC(J) + IST - 1
JSUM = 0
ITROW = IST
IF(NOT, COPLAN, AND, NS .EQ. 2) ITROW = IST + IOVLAP
DO 760 I = 1, ITROW
750 JSUM = JSUM + JOC(I)
JSUM = JSUM - JOC(ITROW) + 1
READ (INT5, 9015) (DOS(I), I = 18, NK)
DO 750 I = IST, NK
IX = 1
IF(NOT, COPLAN, AND, NS .EQ. 2) IX = 1 + IOVLAP
ISUC = JSUM + J - JS(IX) - NC + 1
DEFSL(1, ISUB) = COS(I)

B78
JSUM = JSUM + JOC(I)
750 CONTINUE
C READ AND STORE SLOPES
DO 775 J=NC,NCH
IST = IS(J)
NK = NOC(J) + IST -1
JSUM = 0
ITCM = IST
IF(NOT.COPLAN.AND.NS.EQ.2) ITCM = ISTM + IOMAP
DO 770 I=1,ITCM
770 JSUM = JSUM + JOC(I)
JSUM = JSUM - JOC(ITCM + I)
READ(NI5,9015) (DCO(I),I=IST,NK)
DO 775 I=IST,NK
IX =I
IF(NOT.COPLAN.AND.NS.EQ.2) IX = I + IOMAP
JSUB = JSUM + J - JS(IX) - NC + 1
DEFSL(2,JSUB) = DCI(I)
JSUM = JSUM + JOC(IX)
775 CONTINUE
900 CONTINUE
C WRITE THE DEFSL ARRAY INTO MODESC FILE
IF(NB.EQ.NSURF) GO TO 925
WRITE (IFILE) DEFSL
GO TO 930
925 CONTINUE
CALL RDINT
ITYPE = 5*MIXED
N = NPMRM1(NPMR5)-1
CALL WRDMX(IFILE,NSMRT,KBCDU,NFS,NM,LJ,NB,LWS,2,1D,
1 3,
DEFSL,ITYPE,2,N,PARM,IRR)
IF(IRR.NE.0) GO TO 6020
930 CONTINUE
IF(NB.EQ.NSURF) GO TO 1500
:**(,NOT.PRMODES.AND..NOT. PROCEF) GO TO 1500
C PRINT MODES, COEFFICIENTS OR BOTH
C WRITE (NT5,9500) TITLE, NM, X0, RMAX
IF(NB.EQ.MSURF) GO TO 975
IF(NTYPE.EQ.3) GO TO 960
C PRINT COEFFICIENTS
C IF(NSURF.EQ.1) GO TO 980
CALL RDINT
ITYPE = 5*MIXED
CALL READMX(AICSC,MOREAD,RANDN1,NFS,NMS,LS,LMR1,1D,ITYPES,
1 LRB,XR,M,N,PARM,IRR)
IF(IRR.NE.0) GO TO 6070
C IFLG = IPARM(4)
 IDR1 = IPARM(3)
 CALL PROCEF(1D1,XR,IFLG)
CONTINUE
IF(NS, EQ, 1, AND, MTYPE, EQ, 3) GO TO 975
IF(NS, EQ, 2, AND, MTYPE, EQ, 3) GO TO 975
CALL PRECOF(Ideg, A, IFLAG)
C
975 CONTINUE
NK S = NEW MODES
IF(SURF, EQ, 2)
NRC6& = M*T + IOVLAP
END FILE IFILE
SFILE IFILE

1500 CONTINUE
END FILE IFILE
M = 1
N = 400
CALL WRDMS(IFILE, KMARIT, RANDOU, NFS, NHS, LS, NMR, LWS, 1, ID,
1 IS, ITYPE, M, N, PARM, IRR)
IF(IRR, NE, 0) GO TO 6040
END FILE IFILE
REWD IFILE
2000 CONTINUE
GO TO 3000
2100 CONTINUE
C DETERMINE THICKNESS SLOPES
NBV = IPNTRM(1, MPNTRS) - 1
REWD ITSLSC
IF(HTSLOP, NE, 0) GO TO 2225
C
WRITE ARRAY OF ONE'S
DO 2200 I = 1, NBV
DEFSL(1, I) = 1.0
2200 CONTINUE
C
CALL RDINIT
ITYPE = SMXRED
M = 1
N = NBV
CALL WRDMX(ITSLSC, KMARIT, RANDOU, NFS, NHS, LS, NMR, LWS, 2, ID,
1 DEFSL, ITYPE, M, N, PARM, IRR)
IF(IRR, NE, 0) GO TO 6080
C
END FILE ITSLSC
REWD ITSLSC
GO TO 3000
C
2225 CONTINUE
DO 2600 I = 1, NSURF
IF(NS, EQ, 2, AND, MTYPE, EQ, 3) GO TO 2230
IFILE = IVPSC
REWD IVPSC
GO TO 2240
2230 CONTINUE

B80
IFI = ITSLSCE
2240 CONTINUE
C DO 2700 NBL = 1, NTSLSC
C ZERO OUT THE ARRAY
DO 2250 I = 1, 500
DEFSL(I, I) = 0.0
2250 CONTINUE
C IF (NS.EQ.2) GO TO 2325
NDM = MBW
NC = 1
GO TO 2350
2325 CONTINUE
NC = MBW + 1
NDM = MBW + MBT
READ (IVPSC) DEFSL
2350 CONTINUE
C DO 2300 J = NC, NCH
IST = I(S(J))
NK = NOC(J) + IST - 1
JSUM = 0
ITROW = IST
IF (.NOT. COPLAN .AND. NS.EQ. 2) ITROW = IST + IOMLAP
DO 2400 I = 1, ITROW
2400 JSUM = JSUM + JOC(I)
JSUM = JSUM - JOC(ITROW) + 1
READ (MTS, 9015) (DOB(I), I = IST, NK)
DO 2300 I = IST, NK
IX = I
IF (.NOT. COPLAN .AND. NS.EQ. 2) IX = I + IOMLAP
ISUM = JSUM + J - JS(IX) - NC + 1
DEFSL(I, ISUM) = 1.0 + GANC * DOB(I)
JSUM = JSUM + JOC(IX)
2390 CONTINUE
C IF (NS.EQ. MBURF) GO TO 2550
WRITE (IVPSC) DEFSL
GO TO 2600
2550 CONTINUE
CALL RDINIT
ITYPE = SHMIXED
N = 1
N = MBW
CALL WRDEX (IFI, NWRT, RANDOU, NFS, NS, LS, NS, LS, 2, ID,
S, DEFSL, ITYPE, N, PARAM, IRR)
IF (IRR .NE. 0) GO TO 3000
C 2600 CONTINUE
C 2700 CONTINUE
END FILE IFILE
REWORK IFILE
2800 CONTINUE
3000 CONTINUE
 MCR 00485
 MCR 00486
 MCR 00487
 MCR 00488
 MCR 00489
 MCR 00490
 MCR 00491
 MCR 00492
 MCR 00493
 MCR 00494
 MCR 00495
 MCR 00496
 MCR 00497
 MCR 00498
 MCR 00499
 MCR 00500
 MCR 00501
 MCR 00502
 MCR 00503
 MCR 00504
 MCR 00505
 MCR 00506
 MCR 00507
 MCR 00508
 MCR 00509
 MCR 00510
 MCR 00511
 MCR 00512
 MCR 00513
 MCR 00514
 MCR 00515
 MCR 00516
 MCR 00517
 MCR 00518
 MCR 00519
 MCR 00520
 MCR 00521
 MCR 00522
 MCR 00523
 MCR 00524
 MCR 00525
 MCR 00526
 MCR 00527
 MCR 00528
 MCR 00529
 MCR 00530
 MCR 00531
 MCR 00532
 MCR 00533
 MCR 00534
 MCR 00535
 MCR 00536
 MCR 00537
 MCR 00538
 MCR 00539
 MCR 00540
 MCR 00541
C
9013 FORMAT(7E10.0)
   RETURN
C
C      INPUT DATA ERRORS
C
5000 WRITE (NT5,9000) IDES
   GO TO 6199
5005 WRITE (NT5,9000) IDES, NPTS
   GO TO 6199
C
C      AN ERROR FROM READING OR WRITING A MATRIX FROM TAPE OR
C      DISK FILE OCCURRED.- PRINT MESSAGES AND FLUSH
C
6010 CONTINUE
   WRITE (NT5,9010) IGEOSC, IRR
   WRITE (NT5,9011) NAME
   GO TO 6100
6020 CONTINUE
   WRITE (NT5,9020) MDESC, IRR
   WRITE (NT5,9021) NAME
   GO TO 6100
C
6030 CONTINUE
   WRITE (NT5,9020) MDESC, IRR
   WRITE (NT5,9022)
   GO TO 6100
6040 CONTINUE
   WRITE (NT5,9020) MDESC, IRR
   WRITE (NT5,9023)
   GO TO 6100
C
6050 CONTINUE
   WRITE (NT5,9050) IGEOSC, IRR
   WRITE (NT5,9051) NAME
   GO TO 6100
C
6060 WRITE (NT5,9050) IAICSC, IRR
   WRITE (NT5,9051) NAME
   GO TO 6100
C
6070 CONTINUE
   WRITE (NT5,9070) IAICSC, IRR
   WRITE (NT5,9071) NAME
   GO TO 6100
C
6080 CONTINUE
   WRITE (NT5,9080) ILSLSC, IRR
   WRITE (NT5,9081) NSL
C
6100 CONTINUE
   WRITE (NT5,9101) ID(1), ID(2)
   WRITE (NT5,9102) PVRM, IPRM
   WRITE (NT5,9103) NFS, NR3
   WRITE (NT5,9104) IYPE, M, N

B82
SUBROUTINE TAPMOD(NS, NM, DEFSL)
DIMENSION DEFSL(2,500), D(500), S(500)

C THIS PROGRAM WILL READ MODE SHAPES FROM TAPE IN THE SAME
C FORMAT AS PROGRAM TE0459. THE PROGRAM WILL BE REPLACED BY
C THE AIR FORCE FOR ITS USE AND BOEING MAY MODIFY OR REPLACE
C AS INPUT TAPES ARE MODIFIED OR REPLACED.
C
C NS = 1, FIRST PLANFORM
C = 2, SECOND PLANFORM
C NM = MODE SHAPE NUMBER
C DEFSL - ARRAY WHERE MODE SHAPES FOR ENTIRE PLANFORM(S) IS
C STORED.
C
COMMON /GEOMY/ COPLAN, NSUBDV, XSUBDV, NSUB2, NSUBCN, NSURF,
1 1 B1, BIBETA, B1B1AS, WLAX, WLAX, PS1W,
2 1 WKBW, WKBBW, MYBW, MYBBW, MYBSW, MYBSW, MYBBW,
1 1 XBW, XCNTR
LOGICAL COPLAN
COMMON /GEOM2/ TLAX, TLAX, PS1T, MBST, MBST, MBST, MBST,
1 1 MYST, MBST, MBST, MBST, CA1P
COMMON /FILES/ NS, NT6, INTAPE, INFSP, NPLAIC, NSPAIC, NOUTP,
1 1 IOFS, MODESC, IVPSC, IGEOSC, IWFSC, IAICSC
COMMON /TARETO/ NFS, NWS, LS, NNR, ID(120), NID, ITYPE, LWS, M, N,
1 1 PARM(10), IRR
DIMENSION IPARM(10)
EQUIVALENCE (IPARM, PARM)
COMMON /INDEX/ IS(100), IC(100), JS(100), JOC(100)
C
LOGICAL MOREAD, RANDIN
MOREAD = .FALSE.
RANDIN = .FALSE.
C
CALL RDINIT
IF(NS.NE.1) GO TO 710
IF(NS.EQ.1) REWIND INTAPE
NMS = 2
NFS = INFSP
710 CONTINUE
ITYPE = SHNIXED
CALL READMX(INTAPE, MOREAD, RANDIN, NFS, NMS, LS, NNR, 1, NID, ID, ITYPE,
1 1 LWS, B, M, N, PARM, IRR)
IF(IRR.NE.0) GO TO 6220
CALL RDINIT
ITYPE = SHNIXED
CALL READMX(INTAPE, MOREAD, RANDIN, NFS, NMS, LS, NNR, 1, NID, ID, ITYPE,
1 1 LWS, S, M, N, PARM, IRR)
IF(IRR.NE.0) GO TO 6220
C
IF(NS.EQ.2) GO TO 720
C
1 1 RST PLANFORM
NCH = MYBW
NC = 1
GO TO 725
C
SECOND PLANFORM
CONTINUE
NC = WYBW + 1
NCH = WYBW + WYBT
C
C STORE DEFLECTIONS AND SLOPES
CONTINUE
ITS = 0
DO 750 J = NC, NCH
IST = IS(J)
NK = NC(J) + IST - 1
JSUM = 0
DO 750 I = IST, NK
JSUM = JSUM + JOC(I)
JSUM = JSUM - JOC(IST) + 1
DO 750 I = IST, NK
ITS = ITS + 1
ISUB = JSUM + J - NC + 1 - JS(I)
DEFL(I, ISUB) = D(ITS)
DEFL(2, ISUB) = S(ITS)
JSUM = JSUM + JOC(I)
CONTINUE
C
RETURN
C
AN ERROR DURING READING A MATRIX FROM TAPE OR
DISK FILE OCCURRED. PRINT MESSAGES AND FLUSH
C
CONTINUE
9010 CONTINUE
9020 CONTINUE
WRITE (NT6, 9020) WODESC, IRR
WRITE (NT6, 9021) NM
CONTINUE
9100 CONTINUE
WRITE (NT6, 9101) ID(1), ID(2)
WRITE (NT6, 9102) PARM, PARM
WRITE (NT6, 9103) NFS, NMS
WRITE (NT6, 9104) ITYPE, M, N
WRITE (NT6, 9900)
C
CALL FLUSH(1)
C
9020 FORMAT(54H) ERROR - WHILE READING FROM THE MODE SCRATCH FILE
A10, 15H, ERROR CODE = [4, 4H ???)
9021 FORMAT(5X, 35H) ATTEMPT WAS MADE TO READ MODE SHAPE 13, //)
9101 FORMAT(5X, MX = 0, A10, 110)
9102 FORMAT(5X, MX, *PARAMETERS *10E11.3, /10X, *INTEGER, *(, 17, 9111)
9103 FORMAT(5X, *FILE SPACING = *15, MX, MATRIX SPACING = *15)
9104 FORMAT(5X, MX, MATRIX TYPE = *A10, * DIMENSIONED (*14.2H, X, 14.1H)
9000 FORMAT(40) ERROR OCCURRED IN MODES SECTION (SUBROUTINE TAPMOD)* FNYX
I
END
SUBROUTINE ROPER

C SUBROUTINE TO DETERMINE THE BOXES ON EACH ROW THAT SHOULD
HAVE NODE SHAPES.

C IS(J) - ROW INDEX OF FIRST PLANFORM BOX FOR CHORD J.
NOC(J) - NUMBER OF PLANFORM BOXES ON CHORD J
JS(J) - COL INDEX OF FIRST PLANFORM BOX FOR SPAN J.
JOC(J) - NUMBER OF BOXES BETWEEN FIRST AND LAST PLANFORM
BOX ON SPAN J.

C COMMON /GEOMTY/ COPLAN, NSUBDV, XSUBDV, NSUBCH, NSURF,
1 BI, BIBETA, BIS, BIBTAS, MLA, MLAZ, PSIW,
2 MXBW, MYBBW, MYBW, MWBBW, MWBW, MWBSW, MYBSW,
3 IXBW, XCENRT
LOGICAL COPLAN
COMMON /GEOM2/ TLAX, TLAZ, PSIT, MXBT, MYBT, MYBBT, MYBBST, 
1 MYBST, XBT, XBST, CARL
COMMON /INDEX/ IS(100), NOC(100), JS(100), JOC(100)
C IF(COPLAN) GO TO 100
MCH = MYBW
MOB = MBW
GO TO 200
100 CONTINUE
MCH = MYBW + MBT
MOB = MBT
GO TO 200
200 CONTINUE
DO 500 I = 1, MB
JS(I) = 0
JOC(I) = 0
JCUT = 0
DO 400 J = 1, MCH
IF(I.LT.IS(J)) GO TO 400
ILAST = IS(J) + NOC(J) -1
IF(I.GT.ILAST) GO TO 400
IF(JS(I).NE.0, GO TO 300
JS(I) = J
IF(J.GT.WBW) JS(I) = J - WBW
IF(JCUT.EQ.1) GO TO 400
JY = J
LS(JY,CE) = 1
300 CONTINUE
GO TO 400
500 CONTINUE
CALCULATE FOR SECOND PLANFORM. THIS IS ONLY USED FOR
NON-COPLANAR PLANFORMS.
C IF(INSURF.EQ.1) GO TO 1500

B86
IF(COLNAME)  GO TO 1500

NDH = MYBW + MYBT
JCH = MYBW + 1
IPBT = (IXBT-IXBW)/NSUBDV +1
IOMAP = 0
IF(IPBT.LE.XXW)   IOMAP = XXW - IPBT + 1
DO 1000 I=IPBT,XXW
IX = I + IOMAP
JS(IX) = 0
JCC(IX) = 0
DO 800 J=JCH,NCH
IF(IX.LT.IS(J)) GO TO 800
ILAST = JS(J) + NCC(J) -1
IF(IX.GT.ILAST) GO TO 800
IF(JS(IX).EQ.0) JS(IX) = J-MYBW
JV = J-MYBW
JCC(IX) = JV - JS(IX) + 1
800 CONTINUE
925 CONTINUE
1000 CONTINUE
1900 CONTINUE
RETURN
END
SUBROUTINE FITTER(M,N,X,Y,Z,C,CHM)

DIMENSION X(100), Y(100), Z(IDIM,100), C(IDIM,66)

DIMENSION VS(I0)

LOGICAL COMPLX

C N - DEGREE OF POLYNOMIAL EQUATION
C N - NUMBER OF DATA POINTS TO FIT CURVE THROUGH
C X - X COORDINATE OF DATA POINT
C Y - Y COORDINATE OF DATA POINT
C Z - Z COORDINATE OF DATA POINT
C C - OUTPUT COEFFICIENT ARRAY
C CN - SCALE FACTOR
C CN - SCALE FACTOR
C IDIM - INDICATOR OF REAL OR COMPLEX FUNCTION
C = 1, FUNCTION IS REAL
C = 2, FUNCTION IS COMPLEX
C IF COMPLEX SET DIMENSIONS OF FUNCTION AND COEFFICIENTS
C TO (IDIM * — — )
C
Determine number of coefficients
C
EPS = 1.0E-04
COMPLX = .FALSE.
IF(IDIM.EQ.2) COMPLX = TRUE.
C
SCALE DATA TO REDUCE MAGNITUDE OF MATRIX TERMS.
C SHOULD AVOID BOMBING DUE TO OVERFLOW CONDITIONS.
IF(CN.EQ.D) CN = 1.
IF(CN.EQ.1.) GO TO 15
DO 5 I = 1, N
  X(I) = X(I)/CN
  Y(I) = Y(I)/CN
5 CONTINUE

15 CONTINUE

MN = M + 1
M2 = MN/2.
MC = 10.0*M2 + M2 + EPS
IF(MC.LE.N) GO TO 25
M = M-1
GO TO 15

25 CONTINUE

MC = MC
C
Determine the maximum degree that can be computed in
each direction and set up order of solution.
C
NDV = 1
NDX = M
VS(I) = X(I)
DO 3 I = 1, N
  DO 5 J = 1, NDV
    IF(X(J).EQ.VS(I)) GO TO 25
3 CONTINUE

NDV = NDV + 1
VS(NDV) = X(I)
B28
IF(NCV-I.E.4) GO TO 65 FITTER 00059
95 CONTINUE
80 CONTINUE
MDX = MDV -1
65 CONTINUE
C
MDV = 1
MDY = M
VS(I) = Y(I)
DO 80 I=1,N
DO 70 J=1,NEV
IF(Y(I).EQ.VS(J)) GO TO 75
70 CONTINUE
MDV = MDV + 1
VS(NEV) = Y(I)
IF(NEV-1.EQ.N) GO TO 85
75 CONTINUE
80 CONTINUE
MDY = MDV - 1
65 CONTINUE
C
ITOF = NC +1
ITOF1 = ITOF
IF(COMPLEX) ITOF = ITOF + 1
C
ZERO OUT THE A ARRAY
C
DO 95 I=1,NC
C(I,1) = 0.0
IF(.NOT.COMPLEX) GO TO 90
C(2,1) = 0.0
90 CONTINUE
DO 95 J=1,ITOF
95 A(I,J) = 0.0
C
DETERMINE DEVIATION EQUATION AND SQUARE THE EQUATION
C
A1(I) =1.0
XP(I) =1.0
YP(I) =1.0
MM = M + 1
DO 200 K=1,N
DO 10 L=2,MM
XP(L) = XP(L-1)*X(K)
YP(L) = YP(L-1)*Y(K)
10 CONTINUE
C
I = 1
DO 40 LL=2,MM
DO 20 LL=2,L
IL= L - LL +1
IF(LL-1.GT.MDY) GO TO 30
IF(LL-1.GT.MDX) GO TO 20
I = I + 1
A1(I) = XP(LL)*YP(LL)
20 CONTINUE
30 CONTINUE
C
B89
CONTINUE
A[I+1] = Z[I,K]
IF(K.EQ.1) GO TO 45
NC = I
ITOT = NC + 1
ITOT1 = ITOT
IF(REALX) ITOT1 = ITOT1 + 1

45 CONTINUE
C
DO 1100 I=1,NC
DO 1100 J=I,ITOT
ASAV = AI(I)*AI(J)
A(I,J) = A(I,J) + ASAV
1100 CONTINUE
200 CONTINUE
C
C SQUARE ROOT METHOD
C INTERMEDIATE MATRIX
DO 1200 I=1,NC
IM = I+1
TMP = 0.0
IF(I.EQ.1) GO TO 1150
DO 1120 L=1,IM
1120 TMP = TMP + A(L,I)*A(L,J)
1150 CONTINUE
T = A(I,I) - TMP
IF(T.EQ.0.0) GO TO 4
A(I,I) = 0.0
GO TO 1200
4 CONTINUE
A(I,I) = SQRT(T)
IF(A(I,I).GT.EPS) GO TO 1155
A(I,ITOT) = 0.0
GO TO 1200
1155 CONTINUE
C
JS = I+1
DO 1180 J = JS,ITOT
TMP = 0.0
IF(J.EQ.1) GO TO 1175
DO 1180 L=1,IM
1180 TMP = TMP + A(L,I)*A(L,J)
1175 A(I,J) = (A(I,J) - TMP1/A(I,I)
1180 CONTINUE
1200 CONTINUE
C
C BACK SUBSTITUTE FOR COEFFICIENTS
DO 1400 K=1,NC
I = NC - K + 1
IPM = I
TMP = 0.0
TMP2 = 0.0
IF(A(I,I).GT.EPS) GO TO 1325
C(I,I) = 0.0
IF(REALX) C(I,1) = 0.0
1400 CONTINUE
C
GO TO 1400

1325 CONTINUE
IF(.NOT.COMPLX) GO TO 1375
DO 1350 L=1,N

THMP = THMP + A(I,L)*C(I,L)

IF(.NOT.COMPLX) GO TO 1375
THMP2 = THMP2 + A(I,L)*C(I,L)

1350 CONTINUE

1375 CONTINUE
C(I,1) = (A(I,1TOT) - THMP)/A(I,1)

IF(.NOT.COMPLX) GO TO 1400
CQ2,1) = (A(I,1TOT) - THMP2)/A(I,1)

1400 CONTINUE

C
C REORDER THE COEFFICIENTS IN CORRECT POWERS
C OF X AND Y.
C
IF(MAC,EQ.,NC) GO TO 1475

IZ = 1
I = 1
DO 1440 L=2,MM
DO 1420 LL=1,L
IL = L - LL + 1
I = I + 1
IF(LL-1.LE.MDY.AND.IL-1.LE.MDX) GO TO 1410

X(I) = 0.0
Y(I) = 0.0
GO TO 1420

1410 CONTINUE
IZ = IZ + 1
X(I) = C(I,IZ)
IF(COMPLX) Y(I) = C(2,IZ)

1420 CONTINUE

1440 CONTINUE

C
DO 1450 I=2,MAC
C(I,1) = X(I)
IF(COMPLX) C(2,1) = Y(I)

1450 CONTINUE

1475 CONTINUE

C
ELIMINATE THE SCALE FACTOR FROM THE COEFFICIENTS.
C
IF(NIC,EQ.,1.0) GO TO 1700
I=1
CP= 1.0/CN
DO 1600 L=2,MM
DO 1500 L2=1,L1
I = I+1
C(I,1) = C(I,1)*CP
CQ2,1) = C(2,1)*CP

1500 CONTINUE
CP= CP/CN

1600 CONTINUE

1700 CONTINUE

B91
C     THE C ARRAY NOW CONTAINS THE COEFFICIENTS.
C
RETURN
END
SUBROUTINE MODOUT (DEPSL, JS, JOC, NEGB, NM, IOVLAP)
DIMENSION DEPSL (2, 500), JS (100), JOC (100)
DIMENSION JNM (15), DS (50)
COMMON /GEOM/ COPLAN, NSUBV, XSUBV, NSUBD, NSUBC, NSURF,
1  B1, BIBETA, B1S, BIBETAS, MAX, LMAX, PSLW,
2  NSBM, MBBSW, MBSSW, MBSSW, MBSSW, MBSSW,
3  XBM, XENTR
LOGICAL COPLAN
COMMON /FILES/ NT5, NS5, INTAPE, INFSF, NPLAC, NPLAC, NCOUP,
1  IOUSP, MODESC, IVSBC, ISEGSC, IWEFSC, IAIASC
COMMON /PROBLM/ XMAC, NMODES, NTSLOP, NVALS, SMOOTH, NDEG, CRDFIT,
1  EXJAC, SUBDV, PLYWOOD
LOGICAL SMOOTH, CRDFIT, EXJAC, SUBDV, PLYWOOD
DIMENSION SLOMAT (5)
EQUIVALENCE (SLOMAT (2), IMAT)
DATA SLOMAT/ 10H (15, 3X), 10H, 0, 10HF, 15F7.3 / DATA IMAT, IMAT /10H
DATA IMAT, IMAT /10H 0, 10F -0/ IMAT = 10H 0
NBETS = 0
DO 10 I = 1, NROB
JL = JS(I)+JOC(I)-1
IF (NBETS LT JL) NBETS = JL
90 CONTINUE
NBETS = (NBETS-1)/15 + 1
DO 100 JN = 1, 2
C FIND LARGEST VALUE
C VALUE = 0.0
DO 100 JN = 1, 500
AV = ABS (DEPSL (NP, JL))
IF (AV > VALUE) VALUE = AV
100 CONTINUE
POW = 1
TSCALE = 10.
IF (VALUE GE 10.) TSCALE = 0.1
DO 110 NH = 1, 8
POW = POW * TSCALE
TTEN = VALUE * POW
IF (TTEN < VALUE) GO TO 110
IF (TTEN GE VALUE) GO TO 110
NPA = N
GO TO 115
110 CONTINUE
NPA = 0
115 CONTINUE
   IF (VALUE .LT. 1.00) GO TO 120
C             THE ARRAY MUST BE SCALED DOWN.
C
   NPX = -NPX
   IMAT = IMAT2
   GO TO 122
C             THE ARRAY MUST BE SCALED UP.
C
120 CONTINUE
   NPX = NPX
   IMAT = IMAT1
122 CONTINUE
   IMAT = IMAT + NPX
   IF (NPX.EQ.2) GO TO 124
   WRITE (NT6,9005) NPX
   GO TO 125
124 CONTINUE
   WRITE (NT6,9010) NPX
125 CONTINUE
   DO 900 JPS=1,NSETS
   JBASE = (JPS-1) + 15
   DO 190 JC=1,15
190 JPH(JC) = JC + JBASE
   JL=15
   IF (JPS.EQ.NSETS) JL = NSETS - 15*NSETS-1
   WRITE(NT6,9015) (JPH(J)) ,J=1,JL
   IOW = 0
   ITOF = 0
   DO 800 I=1,NROWS
     DO 200 J=1,50
200 DS(J) = 0.0
C             PUT THE VALUES INTO PRINT ARRAY
C
   J1 = JPS(I)
   IF (J1.EQ.0) GO TO 800
   JL = J1 + JOC(I) - 1
   DO 300 J=J1,JL
300 DS(J) = DEFSL(NP,ITOF)

C             PRINT THE ONES IN THIS SET
C
   JIP=(JPS-1) + 15 + 1
   JIL = JIP + 14
   IF (JL.LT.JIP) GO TO 800
   IF (J1.GT.JIL) GO TO 800
   IF (J1.GT.JL) JIL=JL
   M = 1
   IF (LAP.EQ.0) GO TO 350
   IF (L.EQ.W8B) GO TO 350
   M = I-IOV*LAP
   IF (IOV.NE.0) GO TO 350
   IOV = 1
WRITE(INT6,9015)
390 CONTINUE
WRITE(INT6,SLOMAT) M, (DS(J), J=1IP,JIL)
890 CONTINUE
990 CONTINUE
1000 CONTINUE
RETURN
9005 FORMAT(1HC,46X,*DEFLECTIONS X 1.0E *,12/, 47X,21(IH- ))
9010 FORMAT(1HG,//47X, * SLOPES X 1.0E *,12/, 47X,21(IH- ))
9015 FORMAT(1HG,5X,15(I4,3X))
9020 FORMAT(13,3X,15F7.3)
END
SUBROUTINE PRECOF(IODE,A,IPR)
COMMON /FILES / NT5,NT6,INTAPE,INFOSP,NPLAIC,NPAIC,NOUTP,
1 IQUFSP,MODESC,IVPS,IGESG,INTFSC,IAICSC

DIMENSION A(2),BLNK(7),BKN(6)

EQUIVALENCE (BLKN(2),BKN(1))

DIMENSION XP(7),YP(7)

DATA   BLNK / 7*1H /

C
C THIS SUBROUTINE PRINTS THE COEFFICIENTS USED IN THE
C POLYNOMIAL EQUATION USED IN CALCULATION OF MODE SHAPES
C
C IODE - DEGREE OF POLYNOMIAL EQUATION
C A - ARRAY OF COEFFICIENTS
C IPR - FLAG INDICATING HOW COEFFICIENTS ARE OBTAINED
C =1, READ FROM CARDS
C =2, FROM LEAST SQUARES SURFACE FIT
C
IF(IPR.EQ.1) WRITE(NF6,9055) A(1)
IF(IPR.EQ.2) WRITE(NF6,9065) A(1)

IF(IODE.EQ.0) GO TO 550

IODE = 2
DO 520 I=1,IODE
NCL = I+1
LDEX = IDEX + 1
DO 541 NXP=1,NCL
XP(NXP) = NCL-NXP
YP(NXP) = NXP-1
541 CONTINUE

WRITE(NF6,9060) (BLK(NXP),XP(NXP),YP(NXP),NXP=1,NCL)
WRITE(NF6,9061) (BKN(NXP),NXP=1,NCL)
WRITE(NF6,9062) (A(J),J=1DEX,LDEX)
IODE = LDEX + 1
520 CONTINUE

950 CONTINUE
950 CONTINUE

RETURN

9055 FORMAT(13X,"LOCAL POLYNOMIAL COEFFICIENTS",20X,"FROM CARD INPUT")
   1 /13X,29(1H-)/4G CONSTANT/2X,10.1H-)/E12.4 /
9060 FORMAT(1HD,6A1,4Hx = II,4HY = II,3X) 
9061 FORMAT(1H4,6A1,10X---------------,3X) 
9062 FORMAT(1E12.4,5E14.4)
9063 FORMAT(1H0,12X,LOCAL POLYNOMIAL COEFFICIENTS\13X,4BY LEAST SQUARE
1E3 SURFACE FIT/13X,29(1H-)/4G CONSTANT/2X,10(1H-)/ E12.4 )
END
ITYPE = 5IXEID VIC1AIN 00095

muARy = GATAB VIC10AIN 00096

CALL READN(1NSPAIC,NREAD,RANDIN,NFS,NMS,LS,MARK,500,NID,ID,ITYPE, V1CMAIN 00097

1 LRS, ATAB,M,H,PARM,IRR)

IF(IRR.NE.0) GO TO 6060

REMIND NSPAIC

C IF(MU.NE.1) GO TO 555

C WRITE (NT6,9215) NDEL +1

9215 FORMAT(IHI,14X,#SPATIAL AIC TAPE TABLE OF CONTENTS# /15X,34(IH-), 1 // 5X,#NO. = 4X,#MACH = 6X,#K1-VALUE = 6X,#ERROR# 4X,#SIZE# 5X, 2 #YBAR# 5X,#ZBAR# / )

NDEL = 0

DO 550 I=1,NKST

IF(A Mach(I),GT.0) GO TO 525

NDEL = NDEL +1

GO TO 550

525 WRITE (NT6,9020) I, AMACH(I),AKVAL(I),AERR(I),ISIZE(I),YBARS(I), 1 VERTS(I)

550 CONTINUE

WRITE (NT6,9025) NDEL

C SEARCH FOR MATRICES WITH CORRECT K-VALUE, MACH, ERROR, SIZE, 1 AND YBAR.

C 555 CONTINUE

IVAL = 0

NSIZ = 0

DO 600 I=1,NKST

IF(Abs(A Mach(I)-MACH),GT.1.0E-05) GO TO 600

IF(Abs(AKVAL(I)-K1 ),GT.1.0E-07) GO TO 600

IF(A Err(I),GT.5RR) GO TO 600

IF(Aabs(VertS(I)-El),GT.1.0E-04) GO TO 600

IF(AAbs(YBAR-YBARS(I)),GT.1.0E-04) GO TO 600

C THERE IS A GOOD MATRIX ON TAPE. DETERMINE IF SIZE IS ADEQUATE

C IF(IVAL.NE.0) GO TO 575

IF(NSIZ.GT.ISIZE(I)+10.AND.ERR.GT.AERR(I)) GO TO 600

IVAL = I

NSIZ = ISIZE(I)

575 CONTINUE

IF(NSIZE(I),GE.NROMS) GO TO 600

C THE SIZE IS NOT LARGE ENOUGH. SEE IF THIS IS LARGER THAN ANY 1 PREVIOUS MATRIX.

C IF(ISIZE(I),LE.NSIZ) GO TO 600

IF(NSIZ.GT.ISIZE(I)+10.AND.ERR.GT.AERR(I)) GO TO 600

IVAL = I

NSIZ = ISIZE(I)

600 CONTINUE

C DETERMINE IF THERE WAS A MATRIX ON TAPE THAT COULD BE ENLARGED

C IF(IVAL.EQ.0) GO TO 25
C CALCULATE 2 PLANAR AICS IF SUBDIVISION IS APPLIED.
NPK = 1
IF(NSUBDV.GT.1) NPK = 2
NVCS = NSPAIC + NPK
DO 100 NV = 1,NVCS
C SET K1 VALUE IF SUBDIVIDED AIC
C IF(NPK.EQ.2.AND.NV.EQ.NVCS) K1 = K1/FLOAT(NSUBDV)
C LNZ = UKERNEL
IF(NV.GT.NVCS.AND.NPK.EQ.2) LNZ = LSKERN
DO 100 I=1,LENZ
C(I) = (0.,0.)
W(I) = (0.,0.)
V(I) = (0.,0.)
100 CONTINUE
C 110 CONTINUE
IF(NV.GT.NVCS-NPK) GO TO 10
C READ MUSIC ARRAY FROM IGEOSC. FIRST FILE MUST BE SKIPPED
C PRIOR TO FIRST READ.
C CALL RDINIT
IF(NV.NE.1) GO TO 200
REWIND IGEOSC
NFS = 1
200 CONTINUE
ITYPE = SHAHED
WARRAY = SHAHUIC
CALL READX(IGEOSC, NRREAD, RANDIN, NFS, NMS, LS, NMR, 2, NID, ID, ITYPE, 1, URS, MUCS, M, NRS, FARM, IRR)
IF(IRR.NE.0) GO TO 6010
C YBAR = PARM(4)
EL = PARM(5)
NN = 1
C DETERMINE IF SPATIAL AICS ARE ON TAPE AND GET THEM
C IF POSSIBLE.
C MST = 0
IF(.NOT.OSPAIC) GO TO 25
C AICS ARE ON TAPE. GET TABLE OF CONTENTS.
REWIND NSPAIC
CALL RDINIT
NFS = 1
ITYPE = SHAHED
WARRAY = 6HTAB
CALL READX(NSPAIC, NRREAD, RANDIN, NFS, NMS, LS, NMR, 2, NID, ID, ITYPE, 1, URS, TAB, MST, N, FARM, IRR)
IF(IRR.NE.0) GO TO 6000
C CALL RDINIT
CCALCULATE 2 PLANAR AICS IF SUBDIVISION IS APPLIED.
NPK = 1
IF(NSUBDV.GT.1) NPK = 2
MC = NPATK + NPK
DO 100 NV = 1,NVCS

C SET K1 VALUE IF SUBDIVIDED AIC
IF(NP,.EQ.2.AND.NV,.EQ.NVCS) K1 = K1/FLOAT(NSUBDV)

C LENV = LKEEP
IF(NV,.EQ.NVCS.AND.NP,.EQ.2) LENV = LKEEP
DO 100 I=1,LENV
C(I) = (0.,0.,0.)
W(I) = (0.,0.,0.)
V(I) = (0.,0.,0.)
100 CONTINUE

C CONTINUE
IF(NV,.GT.NVCS-NP) GO TO 10

C READ MUAC ARRAY FROM IGEOGSC. FIRST FILE MUST BE SKIPPED
C PRIOR TO FIRST READ.
C
CALL RDINIT
IF(NV,.NE.1) GO TO 200
REWIND IGEOGSC
NF = 1
200 CONTINUE
ITYPE = SHMIXED
MARRAY = GMUAC
CALL READMX(IGEOGSC,MARRAY,RANGD,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,
1 LRS,MUAC,MUCS,M,PARM,IRR)
IF(IRR,.NE.0) GO TO 6010

C YBAR = PARM(4)
EL = PARM(5)
NN = 1

C DETERMINE IF SPATIAL AICS ARE ON TAPE AND GET THEM
C IF POSSIBLE.
C
NST = 0
IF(.NOT.OSPAIC) GO TO 25

C AICS ARE ON TAPE. GET TABLE OF CONTENTS.
REWIND NSPAIC
CALL REINIT
NF = 1
ITYPE = SHMIXED
MARRAY = GMUAC
CALL READMX(NSPAIC,MARRAY,RANGD,NFS,NMS,LS,NMR,500,NID,ID,ITYPE,
1 LRS, TAB,NST,H,M,PARM,IRR)
IF(IRR,.NE.0) GO TO 6060

C CALL REINIT
C THERE IS A MATRIX THAT CAN BE ENLARGED.
AMACH(IVAL) = -AMACH(IVAL)
WRITE (MT6,9030) IVAL, AERR(IVAL),ISIZE(IVAL),NROWS
C
C SPACE TO CORRECT ARRAY ON TAPE
CALL RDINIT
NMS = (IVAL-1)*4
ITYPE = SHMIXED
MARRY = GHMUTO
CALL READMX(NSPAIC,MREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,
  1 LRS,MUTO,W,M,FARM,IRR)
IF(Irr.NE.0) GO TO 6060
C
CALL RDINIT
ITYPE = SHMIXED
MARRY = GH W
CALL READMX(NSPAIC,MREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,
  1 LRS, W, M, N, FARM, IRR)
IF(IrR.NE.0) GO TO 6060
C
CALL RDINIT
ITYPE = SHMIXED
MARRY = GH V
CALL READMX(NSPAIC,MREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,
  1 LRS, V, W, M, N, FARM, IRR)
IF(Irr.NE.0) GO TO 6060
C
GO TO 25

600 CONTINUE
C THERE IS A GOOD MATRIX ON TAPE. READ THE TAPE, PRINT MESSAGE,
MAIL RESUME ON THIS ONE ANYWAY.
C
C SET NSIZ EQUAL TO NROWS SO THAT MATRIX WILL NOT BE WRITTEN
C ON TAPE
C
NSIZ = NROWS
CALL RDINIT
NMS = (I-1)*4
ITYPE = SHMIXED
MARRY = GHMUTO
CALL READMX(NSPAIC,MREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,
  1 LRS,MUTO,W,M,FARM,IRR)
IF(Irr.NE.0) GO TO 6060
C
CALL RDINIT
ITYPE = SHMIXED
MARRY = GH C
CALL READMX(NSPAIC,MREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,
I ILS,C,M,N, PARM,IRR)
1 IF(IRR.NE.0) GO TO 6060

CALL RDINIT
I TYPEx SMIXED
MARRAY = 6H W
CALL READWX(NSPAIC,MOREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,
1 LRS,WM,N, PARM,IRR)
1 IF(IRR.NE.0) GO TO 6060

CALL RDINIT
I TYPEx SMIXED
MARRAY = 6H V
CALL READWX(NSPAIC,MOREAD,RANDIN,NFS,NMS,LS,NMR,2,NID,ID,ITYPE,
1 LRS,WM,N, PARM,IRR)
1 IF(IRR.NE.0) GO TO 6060

GO TO 25

10 CONTINUE

YBAR = 0.0
EL = 0.0

Determine the size and locations of the planar AIC arrays.
The unsubdivided will be calculated first and stored in
the proper place in blank common.
The subdivided will be calculated second, overlaying some of
the unsubdivided numbers.

LSKERN = size of unsubdivided or subdivided array alone.
ISUB = number of rows on unsubdivided AIC that subdivided
AIC array will overlay
LTCBK = number of boxes of unsubdivided AIC that will be
overlaid
IPKERN = subscript of where first box of unsubdivided box
would be if it were not overlaid. This allows
program to reference unsubdivided array with proper
subscript.
MAXL = length of computed AIC array
XKA = NPLKRN
LUKERN = (XKA/2.) * (XKA + 1.) + .001
MSPKRN = NPLKRN
IST = 0
IF(NSUBDV.EQ.1) GO TO 340

IF(100V IF THE EFFECTIVE AREA WAS INPUT ON CARDC USE THAT
FIND IF THE PLANFORM LIMITS THE SIZE OF THE EFFECTIVE AREA.
IF(NRMEA.GT.20) NRMEA = 20
IF(NRMEA.NE.0) NMSKRN = NSUBDV * NRMEA

NBOKS = MYBSBW
NBOW = 1
NBORW = (MYBSBW-1)/2
300 CONTINUE
IF(NBOKS .GT. LSDKW) GO TO 323
NBOKES = NBOKES + NBORW
NBORW = NBORW - 2
IF(NBORW.LE.0) GO TO 330
NBROW = NBROW + 2
GO TO 300

325 CONTINUE
NBROW = NBROW - 1
IF (NBOKES-NBORW2 .GT. LSDW) NBROW = NBROW - 1
GO TO 335

330 CONTINUE
NBROW = NSKRKN

335 CONTINUE
XMA = NSKRKN
LSKERN = XMA*(XMA/2.) + (XMA/2.) + 0.001
ISUB = NSKRKN/NSUBDV
IF (NSROW .LT. NSKRKN) ISUB = NSROW/NSUBDV
SUB = ISUB
LTCKRN = SUB*SUB/2.) + (SUB/2.) + 0.001
IST = LSKERN - LTCKRN

340 CONTINUE
IPKERN = IST + 1
MAXL = IST + LUKERN
IF(MAXL.LE.LKERNL) GO TO 21
WRITE (NR6,9305) MAXL, LKERNL
9305 FORMAT(9SH#,# ERROR - THE SIZE OF THE AIC ARRAY FOR THIS PLANFORM
1 IS, IS, 15, 29H, THE MAXIMUM SIZE ALLOWED IS, 15, 5H, ## )
CALL FLUSH(1)

C

21 CONTINUE
C
IF(NP,-.EQ.2.AND.NV,.ED.NVCS-1) GO TO 22
NN = 1
NROWM = NSKRKN
GO TO 23

22 CONTINUE
NN = IPKERN
NROWM = NFUKRN

23 CONTINUE
DO 24 I=1,NROWM
MATAC(I,1) = I
MATAC(2,1) = I + I - 1
C

24 CONTINUE
C
C DETERMINE IF KERNELS EXIST ON TAPE AND GET THEM FROM TAPE
C IF POSSIBLE.
C
C
NKOT = 0
IF (.NOT.OPRALIC) GO TO 25
C
C KERNELS ARE ON TAPE, GET TABLE OF CONTENTS
REWIND: MPEARIC
CALL ADRMIT
NFS = 1
ITYPE = SHMIXED
NSKARY = 64TAR
CALL READMXPMAIC,MSREAD,RAINDR,NFS,NM,NMR,500,ND,ITYPE.
2 31X,1HC,34X,1HW,34X,1HV,.3X,2HNU,3X,2HNU,7X,5VELOCITY POTENTIAL VICTMAIN 00437
3COEFFICIENT* 10X,4UPWASH COEFFICIENT* 15X,5SIDEWASH COEFFICIENT* / VICTMAIN 00438
4 2 (3X,2H--),7X,32 (1H--),2X,32 (1H--),2X,32 (1H--) // VICTMAIN 00439

C

IF (.NOT. PRINT) GO TO 52
K=0
KN = 0
IF (NPX.EQ.2 .AND. NV.EQ.NVCS-1) KN = IPKERN -1

C

DO 50 I=1,NROWS
N= I-1
N= N/2
JS = 2*(I-1) +1
IF(YBAR.NE.0.D) JS = JS+1
IF(EL.EQ.0.D) JS = I
DO 50 J=1,JS
K=K+1
KN = KN + 1
N = I - J
IF(EL.EQ.0.D) N = 1 - J
IF(YBAR.LT.0.D) N = -N
WRITE (NITS,9210) M,N,C(KN),W(K),V(K)

9210 FORMAT(215,5X,6E17.8)
50 CONTINUE
GO TO 53

C

92 CONTINUE
RN = NROWS
K = KN* (RV2.) + (RV2.) + 1.DE-05
IF(EL.EQ.0.D) GO TO 53
K = K+K
IF(YBAR.EQ.0.D) K = K - NROWS
53 CONTINUE

C

IF (NSPATK .EQ. 0 .OR. NV .GT. NVCS-NPK) GO TO 55
IF(NV.EQ.1) REMIND IAICSC

C

WRITE THE SPATIAL AICS ON A SCRATCH FILE

C

CALL RDINIT
PARM(1) = K1
PARM(2) = VMACH
PARM(4) = YBAR
PARM(5) = EL
ITYPE = SHMLXED
NARRAY = SHMLXED
CALL WRTEMXX(IAICSC,NWRITE,RANDCU,NFS,NMS,LS,MR,LMS,2,ID,
$ 1 , H , ITYPE, 2, NROWS.PARM,IRR)
1F (IRR.NE.0) GO TO 6170

C

NARRAY = SH C
CALL WRTEMXX(IAICSC,NWRITE,RANDCU,NFS,NMS,LS,MR,LMS,2,ID,
$ 1, C, ITYPE, 2,K, PARM, IRR)
1F (IRR.NE.0) GO TO 6170

C

NARRAY = SH W

B105
IF(.NOT.PRINT) GO TO 52
K=0
KN = 0
IF(NPK.EQ.2.AND.MV.EQ.NVCS-1) KN = IKERN -1
DO 50 J=1,NEQMS
N= J-1
MN = M/2
JS = 2*(I-1) +1
IF(YBAR.NE.0.0) JS = JS+1
IF(EL.EQ.0.0) JS = 1
DO 50 J=1,JS
K=K+1
KN = KN + 1
N = I - J
IF(EL.EQ.0.0) N = 1 - J
IF(YBAR.EQ.0.0) N = -N
WRITE (NT5,9210) M,N,C(KN),W(K),V(K)
9210 FORMAT(2(I5,5X,9.8))
50 CONTINUE
GO TO 53
93 CONTINUE
RN = NEQMS
K = RN* (GK/2.) + (GR/2.) + 1.0E-05
IF(EL.EQ.0.0) GO TO 53
K = K+K
IF(YBAR.EQ.0.0) K = K - NEQMS
53 CONTINUE
C
IF(NB.PATK.EQ.0.OR.NY.GT.NVCS-NPK) GO TO 55
IF(NV.EQ.1) REWIND IAIASC
WRITE THE SPATIAL AICS ON A SCRATCH FILE
CALL RDINIT
PARM(1) = K1
PARM(2) = YAMIC
PARM(4) = YBAR
PARM(5) = EL
ITYPE = 5H MIXED
NARRAY = GM MATIC
CALL WATENX(IAIASC,NWRT,RANDOU,NPS,NMS,LS,LMN,LM6,2,1D, 
MUMIC,ITYPE,2,NEQMS,PARM,IRR)
IF(IRR.NE.0) GO TO 6170
NARRAY = GM C
CALL WATENX(IAIASC,NWRT,RANDOU,NPS,NMS,LS,LMN,LM6,2,1D, 
Nybar,ITYPE,2,K,PARM,IRR)
IF(IRR.NE.0) GO TO 6170
NARRAY = 5H W
CALL WTEMX(IACSC,MXRIT,RANDQ,NFS,NMS,LS,NMR,LSM,2,ID,1,
IF(IRR.NE.0) GO TO 6170

C
KWARY = 3H V
CALL WTEMX(IACSC,MXRIT,RANDQ,NFS,NMS,LS,NMR,LSM,2,ID,1,
IF(IRR.NE.0) GO TO 6170

C
IF(NYM.NE.NMCS-NP) GO TO 55
END FILE IACSC
REWIND IACSC

55 CONTINUE
IF(NPLAIC.EQ.0.AND.NSPAIC.EQ.0) GO TO 1000

C
WRITE THE KERNEL ON TAPE
IF(NYM.LE.NMCS-NP) GO TO 500

C
WRITE ON THE PLANAR KERNEL TAPE
IF(NPLAIC.EQ.0) GO TO 1000

C
DETERMINE IF A PREVIOUS MATRIX WAS ON TAPE.
C
IF NMOT = 0 IT IS A NEW TAPE AND THERE ARE NO OLD ONES
C
IF MSIZE IS LESS THAN NROWS A MATRIX WAS EXPANDED OR THERE
C
WAS NONE WITH CORRESPONDING PARAMETERS
C
IF(NMOT.EQ.0) GO TO 60
IF(NSIZE.NE.NROWS) GO TO 1000

C
80 CONTINUE
NMOT = NMOT + 1
AMACH(NMOT) = XMACH
AKVAL(NMOT) = Ki
AEER(NMOT) = ERR
ISIZE(NMOT) = NROWS
REWIND NPLAIC
CALL RDINIT
NMS = NMOT - 1
ITYPE = SHMIXED
KWARY = 8H C
CALL WTEMX(NPLAIC,KWARY,RANDQ,NFS,NMS,LS,NMR,LSM,2,ID,1,
C(INH), NZA, NY, PARAM, IRR)
IF(IRR.NE.0) GO TO 6150

C
END FILE NPLAIC
CALL RDINIT
ITYPE = SHMIXED
KWARY = 8H2 TAB
CALL WTEMX(NPLAIC,KWARY,RANDQ,NFS,NMS,LS,NMR,LSM,500,1D,1,
TAB, ITYPE, NMOT,3, PARAM, IRR)
IF(IRR.NE.0) GO TO 6150

C
WXARY = 8H MSIZE
CALL WTEMX(NPLAIC,KWARY,RANDQ,NFS,NMS,LS,NMR,LSM,500,1D,1,
MSIZE, ITYPE,1, NMOT, PARAM, IRR)
IF(IRR.NE.0) GO TO 6150
END FILE NPLAIC

B106
END FILE NPLAIC
FILE NPLAIC = .TRUE.
GO TO 1000
900 CONTINUE
C
C THIS AREA WILL WRITE SPATIAL KERNELS ON TAPE
C THIS TAPE WILL BE USED IN DOWNWASH AND VELOCITY POTENTIAL
C CALCULATIONS.
C IN THE DEVELOPMENT STAGE IT WILL BE ASCERTAINED IF A TAPE
C SHOULD BE SAVED FOR SUBSEQUENT RUNS.
C IF(NSPAIC.EQ.0) GO TO 1000
IF(NKST.NE.0) GO TO 910
C
C THERE WAS NO OLD KERNEL TAPE THEREFORE SKIP TO NV-NPR AND
C ADD TO TABLE OF CONTENTS.
C
NKST = NV
GO TO 920
910 CONTINUE
C
C IF NSIZ IS LESS THAN NROWS A MATRIX WAS EXPANDED OR THERE
C WAS NONE WITH CORRESPONDING PARAMETERS.
C IF(NSIZ.GE.NROWS) GO TO 1000
C
C THERE WAS AN OLD KERNEL TAPE THEREFORE SKIP TO NKST+1 AND
C ADD TO TABLE OF CONTENTS
C
NKST = NKST + 1
920 CONTINUE
AMACH(NKST) = XMAX
AKVAL(NKST) = K1
AERR(NKST) = ERR
YBAR(NKST) = YBAR
VERTS(NKST) = EL
ISIZE(NKST) = NROWS
REWIND NSPAIC
CALL RDIIVT
ITYPE = 5Hmixed
NMS = (NKST - 1) #4
MARRAY = 5HMUAC
CALL WRTDIN(NSPAIC,MAXRIT,RANDCN,NFS,NMS,LS,NPP,LWS,2,1D,
1
MUCAC,ITYPE,2,NROWS,PARM,IRR)
IF(IRR.NE.0) GO TO 6160
C
NMS = 0
ITYPE = 5Hmixed
MARRAY = 8H C
CALL WRTDIN(NSPAIC,MAXRIT,RANDCN,NFS,NMS,LS,NPP,LWS,2,1D,
1
C, ITYPE,2,K,PARM,IRR)
IF(IRR.NE.0) GO TO 6160
C
MARRAY = 8H W
CALL WRTDIN(NSPAIC,MAXRIT,RANDCN,NFS,NMS,LS,NPP,LWS,2,1D,
1
W, ITYPE,2,K,PARM,IRR)
IF(IRR.NE.0) GO TO 6160

B107
C  MXARY = 6H V
CALL WRTENX(NSPAIC, MWRITE, RANDON, NF, NS, LS, NR, LWS, IB, 
1         V, ITYPE, 2, K, PARM, IRR)
IF (IRR .NE. 0) GO TO 6160
C  END FILE NSPAIC
C  WRITE TABLE OF CONTENTS ARRAYS
ITYPE = SH MIXED
MXARY = 8H TAB
CALL WRTENX(NSPAIC, MWRITE, RANDON, NF, NS, LS, NR, LWS, 500, IB, 
1         TAB, ITYPE, NST, 3, PARM, IRR)
IF (IRR .NE. 0) GO TO 6160
C  MXARY = 8H TAB
CALL WRTENX(NSPAIC, MWRITE, RANDON, NF, NS, LS, NR, LWS, 500, IB, 
1         TAB, ITYPE, NST, 3, PARM, IRR)
IF (IRR .NE. 0) GO TO 6160
C  END FILE NSPAIC
END FILE NSPAIC
IF (NV, EQ, NVCS-NFR) QS PAIC = .TRUE.
1000 CONTINUE
RETURN
C 6010 CONTINUE
WRITE (NT6, 9010) IGEOSC, IRR
WRITE (NT6, 9011) MXARY
GO TO 6100
C 6020 CONTINUE
WRITE (NT6, 9050) NSPAIC, IRR
WRITE (NT6, 9011) MXARY
GO TO 6100
C 6030 CONTINUE
WRITE (NT6, 9080) NSPAIC, IRR
WRITE (NT6, 9011) MXARY
GO TO 6100
C 6150 CONTINUE
WRITE (NT6, 9150) NSPAIC, IRR
WRITE (NT6, 9151) MXARY
GO TO 6100
C 6160 CONTINUE
WRITE (NT6, 9160) NSPAIC, IRR
WRITE (NT6, 9151) MXARY
GO TO 6100
C 6170 CONTINUE
WRITE (NT6, 9180) IAICS, IRR
WRITE (NT6, 9151) MXARY
C 6180 CONTINUE
WRITE (NT6, 9101) ID(1), ID(2)
WRITE (MTS,9102) IPARM,IPARM
WRITE (MTS,9103) NKS,NKS
WRITE (MTS,9104) IYPE,IN
WRITE (MTS,9900)
C
   CALL FLUSH(1)
C
9005 FORMAT(60AIC ARRAY NO.*,13.*, GENERATED AT AN ACCURACY OF *F6.4.
   1 * IS BEING USED. *)
C
9010 FORMAT(53H50) ERROR - WHILE READING THE GEOMETRY SCRATCH FILE AIC
   1 3SH, ERROR CODE = (4,4H *** )
9011 FORMAT(5X,31H A4NT ATTEMPT WAS MADE TO READ THE 65,8H MATRIX.//)
9030 FORMAT(48H50) ERROR - WHILE READING THE PLANAR AIC FILE A10,
   1 1SH, ERROR CODE = (4,4H *** )
9060 FORMAT(47H50) ERROR - WHILE READING THE SPATIAL AIC FILE A10,
   1 1SH, ERROR CODE = (4,4H *** )
9150 FORMAT(47H50) ERROR - WHILE WRITING THE PLANAR AIC FILE A10,
   1 1SH, ERROR CODE = (4,4H *** )
9160 FORMAT(47H50) ERROR - WHILE WRITING THE SPATIAL AIC FILE A10,
   1 1SH, ERROR CODE = (4,4H *** )
C
9101 FORMAT(5X,5M4TRIX ID = *, A10, I10)
9102 FORMAT(5X,5PARMETERS *,10E1.3, /1Dx,**(INTEGER),*, 17, 9111 )
9103 FORMAT(5X,**FILE SPACING = *,13,** MATRIX SPACING = *,13 )
9104 FORMAT(5X,**M4TRIX TYPE =*,A10,**, DIMENSIONED (*14,3H X,14,1H )
C
9900 FORMAT(5O) ERROR OCCURRED IN AIC SECTION (VICMAIN). *)
END
SUBROUTINE KERNEL(XPACH,K,ERR,C,W,V)

COMMON /FILES/ MS,MSP,INTAPE,INF",NPAIC,NP2AC,NOUP,
1 T,CPSP,NODESC,IPVSC,IGEESC,ITFS,F;IAICSC
COMMON /VICPAR/ YBAR,EL,NRUNS,MAIC2,150)

DIMENSION XIL(4),IUC(3)

DIMENSION C(:),W(I),V(1)

DATA EPS/1.0E-4/

REAL Kt,KIBAR

ZERO (3.,0.)

TIVAH = XPACH*04

KIBAR = (KI*TI4AO-)/(T4AO-1.0)

IT = EL*CEL

IT=0

IF(YBAR.EQ.0.0.) NT= -1

DO 1000 ::I, mO6s

IF(EL.EQ.0.0) GO TO 50

IF(ABS(EL) .GT. VBARL) XLQ0 = ABS(EL)

XIB(I) = XIB(J-1) * XINC

A(J) = 0.0

DO 106 J=1,250

A(J) = 0.0

106 CONTINUE

B110
IWARN = 0
DO 108 J=1,5
TAU = SQRT(XIB(J)*XIB(J) - EL2)
TAUKM = ( KBAR + AM) * TAU
XIB(J) = TAU
CALL RANGE(TAUKM,N)
IF(IN.E.100) GO TO 107
WRITE (NTS,9005) N
9005 FORMAT(99HE) ** THE ARGUMENT FOR A BESSEL FUNCTION YIELDS AN ORDER
1 GREATER THAN 100. ORDER REDUCED TO 100. 
N = 100
IF(IWARN.EQ.1) GO TO 107
IWARN = 1
107 CONTINUE
CALL BESSEL(TAUKM,A(TJ),N)
108 CONTINUE
C
C THERE ARE BOXES ON THIS ROW. FIND LEFT MOST BOX AND PROCEED
ULeft = SQRT(VBARU*VBARU - EL2)
URIGHT = -ULeft
ILHALF = (NTP+1)/2
REM = ABS(VBAR) - 0.5
IL = ULEFT - REM
IL = ILHALF - IL
IR = REM - URIGHT + 1.0
IR = ILHALF + IR
IF(IL.LT.MUAIC(1,1)) GO TO 110
C
C HYPERBO.A IS LESS THAN ALLOWED, REDUCE LIMITS.
MUAIC(1,1) = IL
GO TO 120
C
C HYPERBOLA CROSSED A BOUNDARY, REDUCE CALCULATIONS.
110 IL = MUAIC(1,1)
120 CONTINUE
C
C TEST RIGHT SIDE
IF(IR.GT.MUAIC(2,1)) GO TO 130
C
C HYPERBO.A IS LESS THAN ALLOWED, REDUCE LIMITS.
MUAIC(2,1) = IR
GO TO 140
C
C HYPERBOLA CROSSES A BOUNDARY, REDUCE CALCULATIONS.
130 IR = MUAIC(2,1)
140 CONTINUE
C
C DETERMINE INTEGRALS FOR BOXES IL TO IR.
IF (IL.GT. IR) GO TO 950
DO 900 ID=IL,IR
U = IHAlF - ID
IU = U
ULeft = U + 0.5 + ABS(VBAR)
URIGHT = ULEFT - 1.0
YMUBAR = ULEFT - 0.5
9005 CONTINUE
C
CBARL = SQRT(ULEFT*ULEFT + EL2)
CBARR = SQRT(RIGHT*RIGHT + EL2)

C
IF(CBARL.LE.VBARL .AND. CBARR.LE.VBARL) GO TO 500
C
IF(1U) 300, 400, 200
C
BOX IS TO THE LEFT OF THE CENTER LINE OR APEX
200 CONTINUE
IF(CBARL.LT.VBARU) GO TO 220
C
EDGE BOX, ONLY HAS ONE SEGMENT TO INTEGRATE
NENT = 1
XIL(N) = CBARR
XIL(2) = VBARU
IUC(N) = 1
GO TO 700
C
DOUBLY CUT BOX, HAS TWO SEGMENTS TO INTEGRATE
220 NENT = 2
XIL(N) = VBARL
IF(CBARR.GT.VBARL) XIL(N) = CBARR
XIL(2) = CBARL
XIL(3) = VBARU
IUC(N) = 1
IUC(1) = 0
GO TO 700
C
BOX IS ON THE RIGHT OF THE CENTER LINE OR APEX
300 CONTINUE
IF(CBARR.LT.VBARU) GO TO 320
C
EDGE BOX, HAS ONLY ONE SEGMENT
NENT = 1
XIL(N) = CBARR
XIL(2) = VBARU
IUC(N) = 2
GO TO 700
C
DOUBLY CUT BOX, HAS TWO SEGMENTS
320 NENT = 2
XIL(N) = VBARL
IF(CBARR.GT.VBARL) XIL(N) = CBARR
XIL(2) = CBARL
XIL(3) = VBARU
IUC(N) = 1
IUC(1) = 0
GO TO 700
C
CENTER LINE OR APEX BOX
400 CONTINUE
IF(ABSL.LT.VBARL) GO TO 475
IF(CBARR.LT.VBARU .OR. CBARR.LT.VBARU) GO TO 420
C
ONLY BOX ON ROW, ONLY ONE SEGMENT TO INTEGRATE
NENT = 1
XIL(N) = ABS(EL)
C \( XIL(2) = VBARU \)
C \( IUC(1) = 3 \)
C \( \text{GO TO 700} \)

C 420 CONTINUE
C IF \( \text{CCHRLL} \lt \text{VBARU} \) AND \( \text{CCHRLL} \lt \text{VBARU} \) GO TO 440
C
C ONLY HAS 2 SEGMENTS
C \( \text{NMT} = 2 \)
C \( XIL(1) = \text{ABS(EL)} \)
C \( XIL(2) = \text{CBARR} \)
C \( XIL(3) = VBARU \)
C \( IUC(1) = 3 \)
C \( IUC(2) = 1 \)
C \( \text{GO TO 700} \)

C 440 CONTINUE
C UNLESS THE HYPERBOLA CENTER IS ON A BOX SIDE LINE,
C I.E. YBAR = 0.5, THEN THERE WILL BE 2 SEGMENTS.
C
C IF \( \text{ABS(YBAR)} \neq 0.5 \) GO TO 445
C \( XIL(1) = \text{ABS(EL)} \)
C \( XIL(2) = \text{CBARR} \)
C \( XIL(3) = VBARU \)
C \( IUC(1) = 1 \)
C \( IUC(2) = 0 \)
C \( \text{NMT} = 2 \)
C \( \text{GO TO 700} \)

C 445 CONTINUE
C WILL HAVE THREE SEGMENTS IF YBAR \( \neq \) ZERO
C \( XIL(1) = \text{ABS(EL)} \)
C \( XIL(2) = \text{CBARR} \)
C \( IUC(1) = 3 \)
C IF \( \text{ABS(YBAR)} \neq 0.0 \) GO TO 450
C
C TWO SEGMENTS
C \( \text{NMT} = 2 \)
C \( XIL(3) = VBARU \)
C \( IUC(2) = 0 \)
C \( \text{GO TO 700} \)

C THREE SEGMENTS
C \( \text{NMT} = 3 \)
C \( XIL(3) = \text{CBARR} \)
C \( IUC(2) = 1 \)
C \( XIL(4) = VBARU \)
C \( IUC(3) = 0 \)
C \( \text{GO TO 700} \)

C CENTER LINE BOX, BUT NOT APEX, HAS THREE SEGMENTS
C
C 450 CONTINUE
C IF \( \text{CBARR} \leq VBARL \) GO TO 220
C \( \text{NMT} = 3 \)
C \( XIL(1) = VBARL \)
C \( XIL(2) = \text{CBARR} \)
XIL(3) = CBARL
XIL(4) = VBARU
IUC(1) = 3
IUC(2) = 1
IUC(3) = 0
GO TO 700

C    FULL BOX, ONLY ONE SEGMENT TO INTEGRATE

500 CONTINUE
   NINT = 1
   XIL(1) = CBARL
   XIL(2) = VBARU
   IUC(1) = 0

C    LIMITS AND TYPES FOR ALL SEGMENTS ARE COMPLETED, INTEGRATE.

700 CONTINUE
   IDX = ID
   IF(EL.ED.0.0) IDX = ID - I + 1
   IX = IS + IDX - 1
   IF(C(IX).NE.0.0) GO TO 900
   DO 800 INT=1,NINT
      CSV = (0.,0.)
      WSV = (0.,0.)
      VSV = (0.,0.)
      IFLAG = 0
   C    CALL ROMBERG INTEGRATION FOR REAL PART
      CALL ROMBERG(XIL(INT),XIL(INTR+1),IUC(INT),ERR,IFLAG,KBAR,YBAR,
      EL,YMACH,CTM(1),VTM(1))
      EN1
   1   IF(EL.ED.0.0) GO TO 950
      IFLAG = 1
   C    CALL ROMBERG INTEGRATION FOR IMAGINARY PART
      CALL ROMBERG(XIL(INT),XIL(INTR+1),IUC(INT),ERR,IFLAG,KBAR,YBAR,
      EL,YMACH,CTM(2),VTM(2))
      EN2
   790 CONTINUE
      C(IX) = C(IX) + CSV
      W(IX) = W(IX) + WSV
      V(IX) = V(IX) + VSV
   800 CONTINUE

C    900 CONTINUE
      GO TO 1000
   C    950 CONTINUE
      MMUC(1,1) = 0
      MMUC(2,1) = 0
   C    1000 CONTINUE
      RETURN
   END
SUBROUTINE RCBMER(XILL, XILU, IUC, ERR, IFLAG, KIBAR, YMUBAR, EL, XMACH, ROICER 0002
1, C, W, V ) ROBER 0003
DIMENSION XI(512), FXIC(512), FXIW(512), FXIV(512) ROBER 0004
DIMENSION A(11,11), AW(11,11), AV(11,11), VT(2) ROBER 0005
REAL KIBAR ROBER 0006
PIE = 3.141592654 ROBER 0007

C XILL - XI LOWER LIMIT OF INTEGRATION ROBER 0009
C XILU - XI UPPER LIMIT OF INTEGRATION ROBER 0010
C IUC - FLAG INDICATING TYPE OF BOX OR EDGE CONDITION OF ROBER 0011
C INTERVAL TO BE INTEGRATED. ROBER 0012
C IUC = 0, FULL BOX ROBER 0013
C = 1, LEFT SIDE OF INTERVAL IS EDGE OF MACH HYP. ROBER 0014
C = 2, RIGHT SIDE OF INTERVAL IS EDGE OF MACH HYP. ROBER 0015
C = 3, BOTH SIDES OF INTERVAL IS EDGE OF MACH HYP. ROBER 0016
C ERR - CONVERGENCE TEST CRITERIA ROBER 0017
C IFLAG - INDICATOR OF REAL OR IMAGINARY PARTS ROBER 0018
C IFLAG = 0, REAL PART ROBER 0019
C = 1, IMAGINARY PART ROBER 0020
C KIBAR - FUNCTION OF REDUCED FREQUENCY, MACH NUMBER ROBER 0021
C YMUBAR - COORDINATE HORIZONTALLY OF PULSE SENDING BOX ROBER 0022
C EL - DISTANCE OF RECEIVING BOX ABOVE SENDING PLANE ROBER 0023
C XMACH - MACH NUMBER ROBER 0024
C C - C COEFFICIENT ROBER 0025
C V - V COEFFICIENT ROBER 0026
C W - W COEFFICIENT ROBER 0027
C
C CALCULATE INITIAL VALUES AT END POINTS ROBER 0028
C
C EL2 = EL*EL ROBER 0029
VW = -XMACH/(PI*PI*KIBAR) ROBER 0030
XI (1) = XILL ROBER 0031
XI (2) = XILU ROBER 0032
CALL FUNCT(2, XI, FXIC, FXIW, IFLAG, KIBAR, EL, YMUBAR, IUC, XMACH, BESSW) ROBER 0033
IF(EL.EQ.0.0) GO TO 101 ROBER 0034
IF(KIBAR.EQ.0.0) GO TO 101 ROBER 0035
IF(IUC.EQ.3) GO TO 101 ROBER 0036
CALL VFUNCT(2, XI, FXIV, IFLAG, KIBAR, EL, YMUBAR, IUC, XMACH, 1, VT) ROBER 0037
101 CONTINUE ROBER 0038
C
C TERMS = FXIC(1)/XI(1) ROBER 0039
C TERMS2 = FXIC(2)/XI(2) ROBER 0040
HINT = 0.5*(XILU-XILL) ROBER 0041
TC = HINT*FXIC(1)+FXIC(2) ROBER 0042
TV = HINT*FXIW(1)+FXIW(2) ROBER 0043
A(1,1) = TC ROBER 0044
AW(1,1) = TV ROBER 0045
IF(EL.EQ.0.0) GO TO 102 ROBER 0046
IF(KIBAR.EQ.0.0) GO TO 102 ROBER 0047
IF(IUC.EQ.3) GO TO 102 ROBER 0048
TV = HINT*FXIV(1)+FXIV(2) ROBER 0049
AV(1,1) = TV ROBER 0050
102 CONTINUE ROBER 0051
C
DO 30 I = 2, 11 ROBER 0052
1: = 2***(I-1) ROBER 0053
H = (XILU - XILL)/I ROBER 0054
30 CONTINUE ROBER 0055
C DETERMINE XI LOCATIONS TO EVALUATE FUNCTION AT

DO 3 J = 2, 1, 2
JJ = J/2
XI(JJ) = XILL + (J-1)*H
3 CONTINUE

CALL FUNCT(JJ, XI, FXIC, FXIW, IFLAG, KIBAR, EL, YMUBAR, IUC, XMACH, DUMMY)

IF(EL.EQ.0.0) GO TO 103
IF(KIBAR.EQ.0.0) GO TO 103
IF(IUC.EQ.3) GO TO 103
CALL VFUNC(JJ, XI, FXIV, IFLAG, KIBAR, EL, YMUBAR, IUC, XMACH, DUMMY)
103 CONTINUE

C DETERMINE TRAPEZOIDAL AREA WITH THE NEW FUNCTION EVALUATIONS

THNC = 0.0
THNW = 0.0
DO 5 J = 1, JJ
THNC = THNC + FXIC(J)
THNW = THNW + FXIW(J)
5 CONTINUE

TC = 0.5*THC + H*THNC
TW = 0.5*THW + H*THNW

C PUT THE NEW AREAS INTO THE ARRAY AND PERFORM EXTRAPOLATION

A(M,1) = TC
AW(M,1) = TW
IF(EL.EQ.0.0) GO TO 104
IF(KIBAR.EQ.0.0) GO TO 104
IF(IUC.EQ.3) GO TO 104
TV = 0.5*TV + H*THNC
AV(M,1) = TV
104 CONTINUE

DO 10 M = 2, N
A(M, M) = ((4***(N-1))**2*(M,N-1) - A(M-1,N-1)) / (4***(N-1) - 1)
AW(M, M) = ((4***(N-1))**2*(M,N-1) - AW(M-1,N-1)) / (4***(N-1) - 1)
10 CONTINUE

C DETERMINE IF THE TECHNIQUE HAS REACHED SUFFICIENT CONVERGENCE

C = A(M,M)
W = AW(M,M)
IF(EL.EQ.0.0) GO TO 105
IF(KIBAR.EQ.0.0) GO TO 105
IF(IUC.EQ.3) GO TO 105
V = AV(M,M)
105 CONTINUE

ERR = ABS(ERR*C)
MER = M - 1
UDIF = ABS(A(M,M,MM1) - A(MM1,MM1))
RDIF = ABS(A(M,M,MM1) - A(M,MM1))
DELS = 0.5 *(UDIF + RDIF)
IF (DELS .GE. RERR) GO TO 30

C
C HAS CONVERGED, TEST FOR W CONVERGENCE
IF (EL.EQ.0.0) GO TO 50
RERR = ABS (ERR*W)
UDIF = ABS (AV(M. MMI) - AV(MHI. MMI))
RDIF = ABS (AV(M. M) - AV(M. MMI))
DELS = 0.5 * (UDIF + RDIF)
IF (DELS.GT.RERR) GO TO 30

C
C HAS NOT CONVERGED MAKE ANOTHER LOOP.
30 CONTINUE
50 CONTINUE
C = -C/PI
IF (EL .LE. 0.0) GO TO 70
W = (EL/PI*E)**(1.0+TERM2-TERM1+BESSW)
IF (K1BAR.EQ.0.0) GO TO 65
IF (IUC.EQ.3) GO TO 70
V = W*(V + VT2) - VT(1))
GO TO 70

65 CONTINUE
ETAL = YMABAR - 0.5
IF (IUC.EQ.2) ETAL = - SQRT(XILU*XILU - EL2)
ETAU = YMABAR + 0.5
IF (IUC.EQ.1) ETAU = - SQRT(XILU*XILU - EL2)
XILUE = XILU * XILU
XILLE = XILU - XILU
ETAL2 = ETAL * ETAL
ETAL2 = ETAL + EL2
BE = ETAL2 + EL2
BU = ETAL2 - EL2
BS = XILUE - SU
BS = XILUE - SL
SU = EtaU + EL2
SU = EtaU - EL2
V1 = 0.0
V2 = 0.0
V3 = 0.0
W = 0.0
IF (S1.GT.0.0) V1 = ALG((XILU+SORT(S1))/SORT(SU))
IF (S2.GT.0.0) V2 = ALG((XILU+SORT(S2))/SORT(SL))
IF (S3.GT.0.0) V3 = ALG((XILU+SORT(S3))/SORT(SU))
IF (S4.GT.0.0) W4 = ALG((XILU+SORT(S4))/SORT(SL))
V = (-1.0/PIE)*(V1-V2-V3+W4)
70 CONTINUE
RETURN
END
SUBROUTINE FUNCT(K, XI, FXIC, FXIW, IFLAG, KIBAR, YMUBAR, IUC, IW, MACH, BESSY)

DIMENSION XI (512), FXIC (512), FXIW (512), A (50)

C K - NUMBER OF FUNCTIONS TO EVALUATE
C XI - VARIABLE OF INTEGRATION
C FXIC - FUNCTIONAL VALUE FOR C EQUATION
C FXIW - FUNCTIONAL VALUE FOR W EQUATION
C IFLAG - INDICATOR OF REAL OR IMAGINARY PARTS
C IFLAG = 0, REAL PART
C IFLAG = 1, IMAGINARY PART
C KIBAR - FUNCTION OF REDUCED FREQUENCY AND MACH NUMBER
C EL - DISTANCE OF RECEIVING BOX ABOVE SENDING PLANE
C YMUBAR - COORDINATE HORIZONTALLY OF PULSE SENDING BOX
C IUC - FLAG INDICATING TYPE OF BOX OR EDGE CONDITION OF
C INTERVAL TO BE INTEGRATED.
C MACH - MACH NUMBER
C BESSY - EVALUATION OF END POINTS FOR W COEFFICIENTS.

REAL KIBAR, PIE = 3.141592654, PIE2 = 1.570796327
EL2 = EL*EL
BESSY = 0.0

DO 1000 I=1,K

C SET UP CONSTANTS
TAU = SQRT(XI(1)*XI(1) - EL2)
TAUK = (KIBAR/MACH)*TAU
EP0W = KIBAR*X1(1)
IF(ABS(TAU).LT.1.0E-06) GO TO 25
THETAU = (YMUBAR + 0.5)/TAU
THETAL = (YMUBAR-0.5)/TAU
GO TO 50
25 CONTINUE
THETAL = 0.0
THETAU = 0.0
GO TO 50
50 CONTINUE

C IF(IFLAG.EQ.0) GO TO 100

C IMAGINARY PART
EXPN = -SINEP0W
EXPNW = (EP0W*COS(EPOW) - SIN(EPOW))/(XI(1)*XI(1))
GO TO 200

C REAL PART
100 EXPN = COS(EPOW)
EXPNW = (COS(EPOW) + EP0W*SIN(EPOW))/(XI(1)*XI(1))

C 200 CONTINUE
CALL BFUNC(TAUK,M,A,N)
IF(EL.EQ.0.0) GO TO 250
IF(1.6T.1) GO TO 250
IF(IUC.NE.3) GO TO 250
EXL = ABS(EL) - XI(1)

B118
IF(ABS(EXL).GT.1.0E-05) GO TO 250
BESSY = (EXP(PI/EXL))
250 CONTINUE
BESSX = A(I)
F**THX = 0.0
IF(IUC.EQ.0.OR.IUC.EQ.2) GO TO 300
C
C LEFT SIDE IS BOUNDARY CONDITION
C1 = PIE2
GO TO 400
C
300 CONTINUE
IF(ABS(THETAU).GE.1.0) GO TO 350
C1 = ASIN(THETAU)
GO TO 400
350 CONTINUE
C1 = SIGN(PIE2,THETAU)
C
400 CONTINUE
IF(IUC.LE.1) GO TO 500
C
C RIGHT SIDE IS BOUNDARY CONDITION
C2 = -PIE2
GO TO 600
C
500 CONTINUE
IF(ABS(THETAL).GE.1.0) GO TO 550
C2 = ASIN(THETAL)
GO TO 600
550 CONTINUE
C2 = SIGN(PIE2,THETAL)
600 CONTINUE
C
IF(IUC.EQ.3) GO TO 900
IF(N .EQ. 1) GO TO 900
C
SIGMAX = -1.0
R = 0
PSIGN = 1.0
M = (N+1)/2
DO 800 IR=2,M
R = R + 1
PSIGN = PSIGN * SIGMAX
P**R = PSIGN
B**TERM = B**TERM + P**R Holocaust (IR)*((SIN(2.04R+C1) - SIN(2.04R+C2))
DO 800 CONTINUE
C
800 CONTINUE
FXII(I) = EXPN * (BESSON(C1-C2) + B**TERM)
FXIW(I) = 0.0
IF(EL2.EQ.0.0) GO TO 1000
FXIW(I) = EXPN * (BESSON(C1-C2) + B**TERM)
C
1000 CONTINUE
RETURN
END
SUBROUTINE BESSEL(K12,A,NA)
DIMENSION A(1), AV(150)
REAL K12
C
K12 - FUNCTION OF X1 VALUE, MACH NUMBER AND REDUCED FREQUENCY
C
A = EVALUATION OF THE BESSEL FUNCTION
C
NA - ORDER OF THE BESSEL FUNCTION TO BE EVALUATED
C
ALPHA = 1.E-25
MT = NA + 1
BETA = .00008
IF(K12-BETA) 76,76,78
78 CONTINUE
IF(K12-ALPHA) 76,76,20
76 CONTINUE
(I2) = 0.
A(1) = 1.
GO TO 99
20 SUM = 0.
K = 1.5*K12 + 1.
NP = MAX(DK,MT)
I = NP+11
AV(I+2) = 0.0
AV(I+1) = ALPHA
30 AV(I) = AV(I+1) + K2/K12-AV(I+2)
IF (I-I) 40, 40, 50
50 IF (MOD(I,2)) 60, 70, 60
60 SUM = SUM + AV(I)
70 I = I+1
GO TO 50
40 C = 1./2. *SUM*AV(I))
I = 1
DO 90 II=1,NA,2
A(I) = AV(II) * C
I = I + 1
IF(I. LE.50) GO TO 99
90 CONTINUE
99 CONTINUE
RETURN
END
SUBROUTINE RANGE(K12, NA)
REAL K12

C CALCULATES THE RANGE ON THE VARIABLE N FOR SUBROUTINE BESSL
C
C K12 = FUNCTION OF X VALUE, MACH NUMBER AND REDUCED
C FREQUENCY
C NA = ORDER OF THE BESSEL FUNCTION TO BE EVALUATED
C
400 CONTINUE
C
1 IF(K12 -.01) 101, 98, 98
2 IF(K12 - 3.00) 102, 102, 99
3 IF(K12-19.00) 103, 103, 100
G0 TO 104
C
101 CONTINUE
NA=4
RETURN
102 CONTINUE
NA = 3.0*K12 + 7.0
RETURN
103 CONTINUE
NA = 2.0*K12 + 7.0
RETURN
104 CONTINUE
NA = (10.0/9.)*K12 + 29.
RETURN
END
SUBROUTINE VFUNCT(K,XI,FXIV,IFLAG,KIBAR,EL,YMU,IBC,XMACH,IND,VT)VFUNCT 00002
DIMENSION XI(256),FXIV(256),VT(2)VFUNCT 00003
REAL KIBARVFUNCT 00004

C THIS PROGRAM CALCULATES THE FUNCTION VALUES OF INTEGRATIONVFUNCT 00005
C FOR THE V COEFFICIENTS.VFUNCT 00006
CVFUNCT 00007
K - NUMBER OF VALUES TO CALCULATEVFUNCT 00008
X - VARIABLE ARRAY AT WHICH VALUES ARE CALCULATED.VFUNCT 00009
FXIV - FUNCTION VALUESVFUNCT 00100
IFLAG = FLAG INDICATING REAL OR COMPLEX PARTVFUNCT 00101
IFLAG = 0, REAL PARTVFUNCT 00102
= 1, IMAGINARY PARTVFUNCT 00103
KIBAR - FUNCTION OR REDUCED FREQUENCY AND MACH NUMBER.VFUNCT 00104
EL - DISTANCE OF RECEIVING BOX ABOVE SENDING PLANE.VFUNCT 00105
YMU - COORDINATE HORIZONTALLY OF PULSE SENDING PLANE.VFUNCT 00106
IBC - FLAG INDICATING TYPE OF BOX OR EDGE CONDITION OFVFUNCT 00107
INTERVAL TO BE INTEGRATED.VFUNCT 00108
XMACH - MACH NUMBERVFUNCT 00109
IND - INDICATOR TO CALCULATE VT TERMSVFUNCT 00110
= 0, DO NOT CALCULATEVFUNCT 00111
= 1, CALCULATEVFUNCT 00112
VT - EXTRA TERMS CALCULATE AT THE LIMITS OF INTEGRATIONVFUNCT 00113

EPS = 1.0E-04VFUNCT 00114
EL2 = EL+ELVFUNCT 00115
DO 500 I=1,KVFUNCT 00116

C CALCULATE CONSTANTSVFUNCT 00117
EPOW = KIBAR*XI(I)VFUNCT 00118
X12 = XI(I)/XI(I)VFUNCT 00119
FREQH = KIBAR/XMACHVFUNCT 00120
YMU = YMU + 0.5)VFUNCT 00121
YMU = YMU - 0.5)VFUNCT 00122

C IF(IFLAG.EQ.0) GO TO 100VFUNCT 00123

C IMAGINARY PARTVFUNCT 00124
EXPN = (EPOW*COS(EPOW-SIN(EPOW))/XI2VFUNCT 00125
IF(INC.EQ.1) EXPNY = -SIN(EPOW)VFUNCT 00126
GO TO 200VFUNCT 00127
C REAL PARTVFUNCT 00128
100 EXPN = (COS(EPOW+EPW+EPW-COS(EPOW))/XI2VFUNCT 00129
IF(INC.EQ.1) EXPNY = COS(EPOW)VFUNCT 00130
C
200 CONTINUEVFUNCT 00131
C1 = 0.0VFUNCT 00132
C2 = 0.0VFUNCT 00133
IF(INC.EQ.1) GO TO 300VFUNCT 00134
C CIR = XI2 - YMU2 - EL2VFUNCT 00135
IF(ABS(CIR).LT.EPS) GO TO 300VFUNCT 00136
C1 = SIN(FREQH*SQR(CIR))VFUNCT 00137
C300 CONTINUEVFUNCT 00138
IF(INC.EQ.2) GO TO 400VFUNCT 00139

B122
C
C2R = XI2 - YHUME - EL2
IF(ABS(C2R).LT.EPS) GO TO 400
C2 = SIN(FREAH=SOAT(C2R))

400 CONTINUE
FXIV(I) = EXPN * (CI-C2)
IF(IND.EQ.1) VT(I) = (EXPN/XI(I)) * (CI-C2)
500 CONTINUE
RETURN
END
SUBROUTINE BFUNC(X,AV,N)

C      X - ARGUMENT FOR THE BESSEL FUNCTION
C      JV - EVALUATION OF BESSEL FUNCTION
C      N - MAX ORDER OF BESSEL FUNCTION
C
C      COMMON /BESFUN/ XIB(5), A(50,5)
C      DIMENSION AV(I)
C
C      FIND INTERVAL X IS IN
C      IF(X.GE.XIB(1).AND.X.LE.XIB(5)) GO TO 50
C      EPS = 1.0E-04
C      I = 1
C      IF (ABS(X-XIB(I)) .LE. EPS*XIB(I)) GO TO 150
C      I = 5
C      IF (ABS(X-XIB(I)) .LE. EPS*XIB(I)) GO TO 150
C      WRITE (NT6,9005) X,XIB(1),XIB(5)
C      IF(NFAC(X,XIB(1))).EQ.0) GO TO 400
C
C      N = 1
C      CONTINUE
C      AV(N) = A(N,1)
C      IF(A(N+1,1).EQ.0) GO TO 400
C      N = N + 1
C      CONTINUE
C      N = 1
C      CONTINUE
C      AV(N) = A(N,1) + DXX * (A(N,1) - A(N,1-1))
C      IF(A(N+1,1).EQ.0) GO TO 400
C      N = N + 1
C      CONTINUE
C      RETURN
C END
OVERLAY(AFXabox,1,5)
PROGRAM NAVPHX

COMMON PERNL(1540)
COMMON /CONTROL/ PREX, OACH, TITLE(8), PRVGEOM, PRVMODE, DIHA, DIHT, CONTE(0002)
1 DEFAULT
COMMON /PROB/ XMACH, NOOES, NTSLOP, NKVALS, SMOOTH, NEIG, CREDIT, PROBLM(0002)
1 EXAI, SUBDV, PLYWOOD
COMMON /GEOM/ COPLAN, NSUBD, NSUBD2, NEUBON, NEURF, GEOHY(0002)
1 B1, B1BETA, B1BIS, W1, WLZ, PS1W, GEOHY(0003)
2 XBBW, XBBBN, MBW, MBBW, MBBSW, MBBSW, MBBSW, MBBSW, GEOHY(0004)
3 XBBW, XCENTR

LOGICAL COPLAN
COMMON /GEOM/ TLAX, TLAZ, PS1T, MBST, MBST, MBST, MBST, MBST, GEOHY(0002)
1 MBST, MBST, MBST, CARL
COMMON /KERN/ E1R, NISKRN, IPRKRN, NLARN, NLAPTK, NLGEA
KERN(0002)
1 IKVAL, IKVAL(20), KS(20)
COMMON /FILES/ NTS, NTST, INPAKE, INPSF, INPAIE, NPAIC, NOUTP, FILES(0002)
1 IQUFSP, IGEOSC, TWFSP, IAIASC
COMMON /ICONT/ OPLAC, OSPAIC, XGEOGM, W1TFAN, W1TSL, W1BL, FRBOX, ICONT(0002)
1 PRAI, PRAIC, PRAIR, PRAIF, PRAIF, PRAI, PRAI, PRAIC, PRAIC, PRAIC, PRAIC
2 PRAIC, PRAIC, PRAIC, PRAIC, PRAIC, PRAIC, PRAIC, PRAIC, PRAIC, PRAIC, PRAIC

COMMON /TAPES/ NF1, N15, NTI, NTI(20), NTD, ITYPE, LRL, LRL, LRL
TAPER(0002)

INTEGER PARM(10)
DIMENSION PARM(10)

COMMON /PAICS/ YBAR, EL, MAIC(2, 50), NOEG, SURF
PAICS(0002)

INTEGER PAIC

LOGICAL SURF, SURF
COMMON /MATERIALS/ YBAR, EL, MAIC(2, 50), NOEG, SURF, MAICS(0002)
1 YBAR, EL, MAIC(2, 50), NOEG, SURF, PSIDIF

COMMON /DELPHI/ LMDCAE, TV, LTVP, TERLOC(LTVP)
DELPHI(0002)

COMMON /DELPHI/ LMDCAE, TV, LTVP, TERLOC(LTVP)
DELPHI(0003)

1 IPRMTH(2, 100), NFIN, TOLMAP
B125
C

COMPLEX DELPHI, TVP

C

DEFSL(2,LMODES)

C

DIMENSION DEFSL(2,1000)

C

EQUIVALENCE (DELPHI(81), DEFSL)

C

ARRAYS DELPHI AND DEFSL ARE EQUIVALENCED TO GIVE A 2 ROW UN-OVERLAPPED SECTION

C

DIMENSION TVPX(500), XINITX(2)

C

COMPLEX XINIT

C

EQUIVALENC (TVP,TVPX), (XINIT,XINITX)

C

COMMON /NASHES/ IPTNDW(2,100), ENRLS(1275), ENRLS(1275), ICMLAP

C

COMMON ENRUS, ENRLS

C

COMMON /SNASH/ IPTNSTD(2,50), ENSUBD(2,600), IPTIN,(IPTOT,IPTLS)

C

SNASH 00002

C

COMMON IPTNSTD(LPTNSD), ENSUBD(2,LDSW)

C

COMPLEX ENSUBD

C

IBOWM(LB1XCM,LBOX), WHERE LBOX = LSCHDS/20

C

COMMON /IYCMES/ IBOWM(150,8)

C

IOWHM IS USED FOR BOTH WING AND TAIL BOX CODES

C

COMMON /LROT/ LROT

C

COMMON /CHECKFR/ DPPCR, GECPR, MODCR, AICCPR, NMSCP, SMCR, GAFPCR

C

LOGICAL DPPCR, GECPR, MODCR, AICCPR, NMSCP, SMCR, GAFPCR

C

LOGICAL CHECKFR

C

EQUIVALENCE (CHECKFR,NMSCP)

C

DIMENSION TITL(3)

C

DIMENSION FARMH(10), IPARMH(10)

C

EQUIVALENCE (FARMH, IPARMH)

C

LOGICAL LIMIT, RANDOU, WARD, MAPT

C

DATA TVPX / 5000 600000000000000377777778 /

C

DATA LIMIT,RANDOU, WARD,MAPT / .F.,...F.,F. /

C

DATA XINITX / 2* 37704000000000000000008 /

C

C

PSICIF = PSIT - PSIW

C

IPNXL = LPNXTD

C

LSCHM = LSHW

C

LROT = NCRY + NEUNCN

C

MYTP = (MYBSW + MYSW)

C

PARM(2) = B1

C

PARM(3) = XAMCH

C

IPARMH(4) = NMODES

C

C

REMWDE IGEOMC

C

REMWDE MODESC

C

REMWDE IVPSC

C

XXVAL = XXVAL(IKVAL)

C

C

READ GEOMETRIC INFORMATION

C

WING BOX CODES

C

NMODE = SHBOX CODES

C

CALL RDINIT

C

ITYPE = SH MIXED

C

K = LBOXCW

C

CALL READM (IGEOMC, WARD, .F., NFS, NMS, LS, NNR, K, NID.ID, ITYPE, NAVNAM)
LARS, IBOWM, M,N, FARM, IRR)
IF (IRR .NE. 0) Go to 910
MOBBS = M
IF (NSURF .EQ. 1 OR. COPLAN) Go to 20

C TAIL BOX CODES
MAURY = 10H TAIL CODES
CALL RDINIT
ITYPE = SHMIXED
CALL READMX (IGEOCC, MIRD, .F., NFS, NMS, LS, MNR, K, NID, ID, ITYPE, 1
LRS, IBOWM(MOBBS + 1), M,N, FARM, IRR)
IF (IRR .NE. 0) Go to 910
ISUBT = MOBBS + 2 - IXBST
C = SUBSCRIPT FOR IBOWM TO GET TAIL CODES

C 20 CONTINUE
MAURY = 10H FEXLOC
CALL RDINIT
ITYPE = SHMIXED
CALL READMX (IGEOCC, MIRD, .F., NFS, NMS, LS, MNR, 1, NID, ID, ITYPE, 1
LRS, FEXLOC, M.N. FARM, IRR)
IF (IRR .NE. 0) Go to 910

C MAURY = 10H TELLOC
CALL RDINIT
ITYPE = SHMIXED
CALL READMX (IGEOCC, MIRD, .F., NFS, NMS, LS, MNR, 1, NID, ID, ITYPE, 1
LRS, TELLOC, M.N. FARM, IRR)
IF (IRR .NE. 0) Go to 910
IF (N .NE. 1) Go to 930

C IF (PSIW .NE. 0 AND. DIMG) Go to 30
IF (SMFLW .NE. 0) Go to 30
IF (NSURF .EQ. 1) Go to 40
IF (CAPL .NE. O) Go to 30
IF (PSIDIF .NE. 0) Go to 30
IF (PSIT .NE. 0 AND. DINT) Go to 30
Go to 40

C READ THE ARRAY OF AIC TABLE .F. CONTENTS

C 30 CONTINUE
MAURY = 9HSPAT. TOC
CALL RDINIT
NMS = 2
K = 4
CALL READMX (IGEOCC, MIRD, .F., NFS, NMS, LS, MNR, K, NID, ID, ITYPE, 1
LRS, PAIG, M,N, FARM, IRR)
IF (IRR .NE. 0) Go to 910
DO 35 I = 1, 4
K(I) = IPARN(I+2)
35 CONTINUE

C 40 CONTINUE
REMIND IGEOCC

C GET POINTER ARRAY FOR NODES
CALL RDINIT
ITYPE = SHMIXED

B127
CALL READMX(MODESC, NRBD,,F., NFS,NMS,LS, NMR, 2, NIC,ID, ITYPE,
NRF, NMS, LS, NMR, 2, NIC, ID, ITYPE)
1 IF (IRR .NE. 0) GO TO 912
IF (M,NE. 2) GO TO 931
NPNTS = N
IOMAP = 1NPAM(3)
C IOMAP = NUMBER OF ROWS TO ALLOW FOR TAIL OVERLAP (TAIL ONLY)
C NPNTS = TOTAL NUMBER OF ROWS ON BOTH SURFACES, +1.
C (INCLUDES OVERLAP IF SPATIAL)
C
C SET UP POINTER ARRAY FOR UNSUBDIVIDED DOWNSHADES
IP = 1
MBB = MYBB
IF (COPLAN) MYBB = MAX(MYBB,MYBBT)
MYBB = MYBB+SUBDIV
MYBB = MYBB
IF (COPLAN) MYBB = MBB
IOMAP = 0
IF (INSURF .EQ. 1 OR COPLAN) GO TO 50
MBB = MBB
IXBT = (IXBT-IXBT)/SUBDIV + 1
IP = MYBB + 1
IPNT1 = IPNTD(1,IP)
C CALL P2MTR (IXBT, MYBT-IXBT+1, MYBB, ,F., ,T., IBOM,LBXCDW, LPNTD1,1,
1 IP, IPNTD1)
MBB = MBB
IF (COPLAN) MBB = MBBT
MYBB = MYBB
IOMAP = 0
IF (INSURF .EQ. 1 OR COPLAN) GO TO 50
MBB = MBB
IXBT = (IXBT-IXBT)/SUBDIV + 1
IP = MYBB + 1
IPNT1 = IPNTD(1,IP)
C CALL P2MTR (IXBT, MBBT-IXBT+1, MYBBT, ,F., ,T., IBOM(ISUBT,1),
1 LBXCDW, LPNTD1, IPNT, IP, IPNTD1)
IF (MYBB .GE. IXBT) IOMAP = MBB - IXBT + 1
5D CONTINUE
C DO LOOP ON MODE SHAPES
DO 80 I = 1,NMODE
C ZERO OUT THE DOWNSHADE AND VELOCITY POTENTIAL ARRAYS
LIM = IPNTD(1,IP) - 1
DO 80 1 = 1,LIM
ENRUS(1) = XINIT
ENRLS(1) = XINIT
80 CONTINUE
C READ IN MODE SHAPE
CALL RDSIM
MARRAY = IDOMODE SHAPE
ITYPE = 4REAL
CALL READMX(MODESC, NRBD,,F., NFS,NMS,LS, NMR, 2, MID, ID, ITYPE,
NRF, NMS, LS, NMR, 2, MID, ID, ITYPE)
1 IF (IRR .NE. 0) GO TO 912
IF (M,NE. 2) GO TO 931
B128
C COMPUTE DOWMASHES AND VELOCITY POTENTIALS FOR ONE MODE

CALL VELPO ( IBOM, LBCED, PKRENL, IPKERN, PKREN, .T., DMTN)

C OPTIONAL PRINT OF NORMAL DOWMASHES

IF (.NOT. CHECKPR .OR. PRMS) GO TO 90
IF (NSUBDY .EQ. 1) GO TO 87
IF (.NOT. CHECKPR) GO TO 87
TITL(1) = 10HEN SUBDIV
TITL(2) = 10HEDD, UPPER
TITL(3) = 10H, PARTIAL
IF (IPHIN .LT. IPHNT) GO TO 86
CALL PRINTR(TITL,IMODE,ENUSBD,2,1,IPHTIN-1,MYBSS,IPNTSD)

GO TO 87

86 CONTINUE

CALL PRINTR(TITL,IMODE,ENUSBD,2,IPHTOT,IPHTLS-1,MYBSS,IPNTSD)
CALL PRINTR(TITL,IMODE,ENUSBD,2,1,IPHTIN-1,MYBSS,IPNTSD)

87 CONTINUE

TITL(1) = 10HISING UPPER
TITL(2) = 10H SURFACE N
TITL(3) = 10HOMAL WASH
IF (COPLAN) TITL(1) = 10H WING/TA
IF (COPLAN) TITL(2) = 10HIL UPPER N
CALL PRINTR(TITL,IMODE,EXRUS,1,1,MYBSS,MYBSS,IPNTDW)
IF (.N, COPLAN TITL(1) = 10HISING LOWER
IF (.COPLAN TITL(2) = 10HIL LOWER N
CALL PRINTR(TITL,IMODE,EXRUS,1,1,MYBSS,MYBSS,IPNTDW

C

90 CONTINUE

IF (NSURF .EQ. 1 .OR. COPLAN) GO TO 130
C DETERMINE WHICH WING SURFACE CONTRIBUTES TO THE TAIL.

IF (CAPL .GT. O) GO TO 130
IF (CAPL .EQ. 0 .AND. PSIDIF .GT. O) GO TO 130
C THE LOWER WING SURFACE CONTRIBUTES TO THE TAIL

LIM = IPNTDW11,MXXBDW1) - 1
DO 120 I = 1,LIM
EXRUS(I) = EXRUS(I)

120 CONTINUE

C COMPUTE THE TAIL NORMAL DOWMASHES AND VELOCITY POTENTIALS

130 CONTINUE

CALL VELPO ( IBOM(ISURF,1),LBCED, PKRENL(IPKERN), PKREN, .T., DMTN)

C OPTIONAL PRINT OF NORMAL DOWMASHES

IF (.NOT. CHECKPR .OR. PRMS) GO TO 135
IF (NSUBDY .EQ. 1 .OR. NOT. CHECKPR) GO TO 135
TITL(1) = 10HEN SUBDIV
TITL(2) = 10HEDD, UPPER
TITL(3) = 10H, PARTIAL
IF (IPHIN .LT. IPHNT) GO TO 131
CALL PRINTR(TITL,IMODE,ENUSBD,2,1,IPHTIN-1,MYBSS,IPNTSD)

GO TO 133

131 CONTINUE

CALL PRINTR(TITL,IMODE,ENUSBD,2,IPHTOT,IPHTLS-1,MYBSS,IPNTSD)
CALL PRINTR(TITL,IMODE,ENUSBD,2,1,IPHTIN-1,MYBSS,IPNTSD)

133 CONTINUE

TITL(1) = 10HTAIL UPPER
CALL PRINTR(TITL,IMCCE,ENRLS,1,IXBUT,MBUT,MBBT,MMBT,1)
1 INTRD(1,IOVLP+1)
TITL(1) = T0NTAIL Lower
CALL PRINTR(TITL,IMCCE,ENRLS,1,IXBUT,MBUT,MBBT,MMBT,1)
INTRD(1,IOVLP+1)

135 CONTINUE M
C
C WRITE THE RESULTS ON SCRATCH FILE:
W
140 CONTINUE M
PARAM(1) = WXL
DO 210 I = 1,4
210 PARAM(I) = PARAM(I)
C
C XX WRITE DELTA PHIS, TEMPORARILY AS A REAL MATRIX, WRITE XX
I
\( \beta = 2 \)
I
\( \text{TYPE} = \text{ARREAL} \)
M = INTRD(1,MBET+IOVLP+1) - 1
CALL WRTENX(INTXC,ivec,..FS,LS,IVPSC,1,IPRM)
1 ITYPE, M,N,PARM, IRR)
IF (IRR .NE. 0) GO TO 920
C
M = 2
N = NTVP
CALL WRTENX(INTXC,ivec,..FS,LS,IVPSC,1,IPRM)
1 ITYPE, M,N,PARM, IRR)
IF (IRR .NE. 0) GO TO 230

C
IF (.NOT. PVF) GO TO 230
TITL(1) = WING
TITL(2) = IOVNC1\VCTY P
TITL(3) = IOVCONTENTS
M = NRVW
IF (.N. COPLAN) GO TO 220
TITL(1) = IOVING/ TAIL
M = INTRD - 1
220 CALL PRINTR(TITL,IMCCE,DELPHI,1,1,M,MB,IPNTRM)
1 IF (MSURF .EQ. 1 .OR. COPLAN) GO TO 230
TITL(1) = WING TAIL
CALL PRINTR(TITL,IMCCE,DELPHI,1,1,IXBUT,MBUT,MBBT,MMBT,1)
INTRD(1,IOVLP+1)

230 CONTINUE C
C ARE SAMPLE WASHES DESIRED -
C IF (MSURF .EQ. 2 .OR. ISMLWP .EQ. 0) GO TO 500
C YES, IS SAMPLE WASH PRINTOUT DESIRED
C IF (.N. IFRM .OR. IFRSW) GO TO 500
C LOOP ON CHORDS WHICH SAMPLE-WASH IS DESIRED
DO 500 JCRLD = 1,ISMLWP
JT = ICHRLD(JORD)
IFIRST = IBOK(JORD)
ILAST = IBOL(JORD)
CALL ISMLWP (ICRM,LBCRM,JORD,JT,IFIRST,ILAST)
500 CONTINUE C
C
500 CONTINUE C
C END OF LOOP ON MODE SHAPES, FROM STATEMENT 50*
C
RETURN
C DIAGNOSTICS - ALL CALL FLUSH
C
C READING FROM SCRATCH FILE
910 CONTINUE
WRITE (NT6,9100) IGCOSE
GO TO 930
912 CONTINUE
WRITE (NT6,9120) MODESC
GO TO 930
920 CONTINUE
C WRITING ON SCRATCH FILE
WRITE (NT6,9200) IVPSCE
GO TO 930
C INCORRECT DIMENSIONS READ
930 CONTINUE
I = 1
GO TO 932
931 I = 2
932 WRITE (NT6,9320) I
IF (WRD) GO TO 990
GO TO 962
C ERROR DETECTED READING A MATRIX
960 CONTINUE
WRITE (NT6,9600) IRR
GO TO 990
C ERROR DETECTED WRITING A MATRIX
952 CONTINUE
WRITE (NT6,9520) IRR
GO TO 990
C MATRIX DESCRIPTION
960 CONTINUE
WRITE (NT6,9600) ID(1),I=1,10,(ID(1),I=1,10)
WRITE (NT6,9622) PMAP,PARM
WRITE (NT6,9614) NMR,NMR,LAS,LWS
GO TO 964
962 WRITE (NT6,9620) ID(1),ID(2)
WRITE (NT6,9622) PMAP,PARM
WRITE (NT6,9624) NFS,NHS
964 WRITE (NT6,9640) ITYPE,M,H
WRITE (NT6,9630) MXARY
GO TO 990
C 990 CONTINUE
WRITE (NT6,9900)
C CALL FLUSH(1)
C
C DIAGNOSTIC FORMATS
9100 FORMAT(4H4H000 ERROR WHILE READING GEOMETRY SCRATCH FILE,A10,1H 4H000 )
9120 FORMAT(4H4H000 ERROR WHILE READING MOCES SCRATCH FILE,A10,1H 4H000)
9200 FORMAT(5H4H000 ERROR WHILE WRITING VELOCITY POTENTIAL SCRATCH

B131
1 6H FILE, A10, 4H ***
9300 FORMAT(1H0), 48H*** MATRIX READ ERROR. THE M DIMENSION SHOULD
1 4H BE .12, 4H ***)
9500 FORMAT(1EOH) *** ERROR CODE ,15, 28H WHILE READING THE FOLLOWING
1 11H MATRIX *** )
9520 FORMAT(1EOH) *** ERROR CODE ,15, 28H WHILE WRITING THE FOLLOWING
1 11H MATRIX *** )
9600 FORMAT( 5X,4MATRIX ID = *.10A10 / (20X,10A10) )
9614 FORMAT( 5X,22H MATRIX INDEX NAME ) = ,15,2H (A10,1H) /
1 5X,33H LEVEL NUMBER READ (OR WRITTEN) = 02,2H (.02,1H )
9620 FORMAT( 5X,4 MATRIX ID = *, A10, 110 )
9622 FORMAT( 5X,11H PARAMETERS, 10E11.3 /10X, 9H(INTEGER), 17,9111 )
9624 FORMAT( 5X,15H FILE SPACING = ,13, 19H, MATRIX SPACING = ,13 )
9630 FORMAT( 5X,A10,21H ARRAY WAS BEING USED )
9640 FORMAT( 5X,4 MATRIX TYPE = *,A10, *, DIMENSIONED (4I4,4X,14,X,4) )
9900 FORMAT(54H) ***ERROR OCCURRED DURING VELOCITY POTENTIAL CALCULAT
1 5HIONS *** )
C
END

B132
SUBROUTINE VELPOT(IXY, LBXCD, PKERN, SKERN, WING, DIMS)

C VELPOT IS CALLED ONCE FOR EACH MODE TO COMPUTE NORMAL WASHES AND VELOCITY POTENTIALS FOR THAT MODE.

C IXY = BOX CODES FOR THE SURFACE
C LBXCD = BOX CODE ARRAY ROW DIMENSION
C PKERN = ARRAY CONTAINING (NU, MJ, 0)
C SKERN = ARRAY CONTAINING SUBDIVIDED (NJ, MU, 0)
C WING = LOGICAL .T. FOR WING OR COPLANAR, .F. FOR TAIL
C DIMS = LOGICAL .T. TO INCLUDE DIMERIAL EFFECTS, .F. TO IGNORE, FOR WING/TAIL/FLAP.

C DIMENSION IXY(LBXCD,1), ICODE(1)

COMPLEX PKERN(1), SKERN(1)

LOGICAL WING, DIMS

C OUTPUTS -
C DELPHI = DELTA PHI (VELOCITY POTENTIAL) ARRAY

C COMMON PARAMETERS USED:
C NSUBD = NUMBER OF SUBDIVISIONS
C BI = BOX LENGTH
C BS = SUBDIVIDED BOX LENGTH

C COMMON /FILES/ NT5,NTS,NTAPE,INSF, NPXIC,NPBIC,NOUTP, FILES 00001
1 IQUSR,MODEIC,IVPSC, IGEORL, IWPSL, IAICSC FILES 00002
C COMMON /ARRAYS/ KBXCDW,KBXCDL,IXY, KBKXCT,KBKXCT, KJALRH, KJALRH, ARRAYS 00003
1 KALPH, KERL, KERL, KPHRM, LPHRM, DEPSL, KELPH, ARRAYS 00004
2 LMODE, KPMSTS, LPMSTS, KSMRL, LSMRL, LPMNTW, ARRAYS 00005
3 KDMW, KTVP, LTVP ARRAYS 00006
C COMMON /GEOMT/ COPLAN,NSUBD,XSUBD,NSUBC,NSURF, GEOMTY 00001
1 B1,BI,BETA,B1S,B1STAS,MLA,MLAZ,PSIT.
2 MSW,MXSW,MYSW,MXSW,MXSW,MYSW,MYSW,MYSW.
3 IXW, XCENTR GEOMTY 00002
C LOGICAL COPLAN GEOMTY 00003
C GEORL / TLAX,TLAT,PSIT,MYST,MYBT,MYBT,MYBT,MYST, GEORE 00004
1 MYBST,IXT,IXT,CAFL GEORE 00005
C COMMON /MODES/ SYM, SYM, SYM, SYM, SYM, MYPET MODEM 00006

C COMMON /AICS/ YVAR,EL,YMAIC(2,50), NCHSS, SURF,
1 YVARL,ELL,YMAICL(2,50), NCHSL, SURFL, PSIDIF
C LOGICAL SURF,SURFL AICS 00001
C COMMON /AICS/ XKL, C(1640), W(1640), V(1640) AICS 00002
C COMPLEX W, V AICS 00003
C DELPHI (LMODE3), TVP(TLP), TEMLOC (LTVI)
C COMMON /DELTA/ DELPHI (100), TVP (250), TEMLOC (250), PEMLOC (250), DELTA 00004
1 IPMTRM (2,100), IPNTRM, ICMAP DELTA 00005
C COMPLEX DELPHI, TVP DELTA 00006
C COMMON /DEPSL/ (2,LMODES)
C DIMENSION DEPSL (2,1000)
C EQUIVALENCE (DELPHI (1), DEPSL)
C EQUIVALENCE TO GIVE A 2 ROW UN-OVERLAPPED SECTION DEPSL 00007
C COMMON /NASH/ IPMTRM (2,100), EMRUS (1275), ENRUS (1275), ICMAP, NASHES 00008
C COMPLEX EMRUS, ENRUS, NASHES 00009
C COMMON /NASH/ IPMTRM (2,100), EMRUS (1275), ENRUS (1275), ICMAP, NASHES 000010
C EQUIVALENCE (DELPHI (1), DEPSL)
C EQUIVALENCE TO GIVE A 2 ROW UN-OVERLAPPED SECTION DEPSL 000011
C COMMON /NASHES/ IPMTRM (2,100), EMRUS (1275), ENRUS (1275), ICMAP, NASHES 000012
C COMPLEX EMRUS, ENRUS, NASHES 000013

B133
IPTSD (LPNTSD), ENSUBD (24.0SDV)

COMMON ENSUBD

IBOWA(LBOWA,LBOXA), WHERE LBOW = LSCHDS(20)

COMMON /BOWES/ IBOW(15.0,8)

IBOWA IS USED FOR BOTH WING AND TAIL BOX COES

COMMON /CHECKPR/ DPCCR, GECOCR, MODCCR, AICCPR, NMSCR, SMCCR, GAFCPR

LOGICAL DPCCR, GECOCR, MODCCR, AICCPR, NMSCR, SMCCR, GAFCPR

EQUIVALENCE (CHECKPR, NMSCPR)

LOGICAL CHECKPR

COMPLEX AZERO, DELPH, DELPHA, DELPHB, B, SDELPH, COEF

COMPLEX ENULU(50), ENULL(50), ENUWU(50), ENULW(50), ENULU(50), ENULW(50)

COMMON ENIF, ENSUM, ENSRUS, ENSRLS

COMPLEX ENSBD(2)

EQUIVALENCE (ENSUS, ENSDBD(1)), (ENSRLS, ENSBD(2))

COMMON ELSW(15,5), ELSL(50)

LOGICAL CRC1W, FR(3,1), CRC1E, FBXV, CRCV, SUBCFV

LOGICAL FULLBX(50)

INTEGER WETT, RWLX, LWR

DATA X14T(2) / 2*377C400000000000000011/

DATA XIT(2) / 2*377C400000000000000011/

IF (SWING) GO TO 80

IF (NYB) GO TO 80

IF (MNYB) GO TO 80

GO TO 90

CONTINUE
WBB = WBBW
WBB = (WBB-1) + NSUBDV + 1
WBB = MAX(MBB, WBBW)
GO TO 100
85 CONTINUE
WBB = MAX(MYBBW, WBBB)
WBB = MAX(MYBBB, WBBB)
90 CONTINUE
WBB = WBBT
WBB = WBBST
C
100 CONTINUE
PSIS2 = 2 * PSIS
PSISUM = PSIW + PSIT
CPIS2 = COS(PSIS2)
SPIS2 = SIN(PSIS2)
CPISUM = COS(PSISUM)
SPISUM = SIN(PSISUM)
IRFB = IXB - NSUBBZ
C = ROW OF FIRST SUBDIVIDED BOX IN THE FIRST ROW OF THE
C UNSUBDIVIDED BOXES
WBBST = (WBBST + NSUBDDW) * NSUBDV
IF (NSUBDV .EQ. 1) GO TO 110
SUBOFF = .F.
MYBBPI = WBB + 1
DO 105 I = 1, MYBBPI
ENRLU(I) = (0., 0.)
ENRLL(I) = (0., 0.)
ENRUR(I) = (0., 0.)
ENRLW(I) = (0., 0.)
FULLX(I) = .T.
105 CONTINUE
GO TO 120
110 SUBOFF = .T.
CROW = .T.
LROW = .T.
FROW = .T.
BROW = .T.
LROW = .T.
C
C LOOP ON ALL (SUBDIVIDED) ROWS OF THE SURFACE
C
120 CONTINUE
FLROW = FLOAT(IXBS) - 1.0
DO 1300 IRW = IXBS, MIRS
FLROW = FLROW + 1.0
C
C SET FLAGS FOR FIRST, CENTER AND LAST SUBDIVIDED ROW IN UN-
C SUBDIVIDED ROW
IF (SUBOFF) GO TO 270
CROW = .F.
IF (IRW - 1) 230, 220, 210
210 IF (MOD(IRW - 1), NSUBDV) .NE. 0) GO TO 240
220 CROW = .T.
GO TO 230
230 CROW = .F.
IF (IRW .EQ. 1BS OR. IRW .EQ. IRFB) CROW = .T.
GO TO 250

240 CROW = .F.,
1 = MOD(IROW+NSUBCV - 1X8, NSUBDV)
1F (1 - 1) 260, 245, 250

245 FROM = .T.,
250 LRCW = .F.,
1F (FROM .NE. MXBS) GO TO 270

280 LRCW = .T.,
C
C GET THE SUBDIVIDED ROW NUMBER FOR THE CENTER OF THE ASSOCIATED
C UNSUBDIVIDED BOX, ICENT

270 CONTINUE
C
IF (CROW) GO TO 340
IF (IFROM - IRFB) 310, 320, 330
C
NO FULL ROW

310 ICENT = 0
1KCENT = 0
NDCDCN = 0
CBOX = .F.,
LCBK = .F.,
GO TO 355

320 ICENT = IX8
GO TO 355

330 ICENT = ((ICENT - IRFB) # NSUBDV + 1X8
1F (ICENT .GT. MXBS) ICENT = ICENT - NSUBDV
GO TO 355

340 ICENT = IROW
C
C 350 CONTINUE
C
1KCENT = (ICENT - IRFB) # NSUBDV + 1
355 CONTINUE
JENLOC = 1
1F (.NOT. WING) JENLOC = MBBSW + 1
C
C LOOP ON ALL (SUBDIVIDED) CHORDS FOR THE SURFACE AND DIAPHRAGM

360 DO 1200 JCOL = 1, MBBSX
C
GAT THE CURRENT (SUBDIVIDED) BOX CODE
CALL DCPD3R (IBOX, LBOX, IROW, JCOL, IROW, JCOL, .T., NDCDCN)
1F (SUDBOFF) GO TO 480
C
C DO 1200 1200 JCOL = 1, MBBSX
C
GAT THE INFORMATION ABOUT POSITION WITHIN UNSUBDIVIDED BOX
C
C ICENT = I-LOCATION (SUBDIVIDED) OF THE CONTROL POINT
C JCENT = J-LOCATION (SUBDIVIDED) OF THE CONTROL POINT
C JUCENT = UNSUBDIVED I-LOCATION OF ASSOCIATED CONTROL POINT
C JUCENT = UNSUBDIVED J-LOCATION OF ASSOCIATED CONTROL POINT
C ICENT = UNSUBDIVED I-LOCATION OF THE NEAREST PLANFORM
C CONTROL POINT, IF THE SUBDIVIDED BOX IS ON-PLANFORM
C CBOX = .T., CURRENT BOX IS A CENTER BOX
C LBOX = .T., THIS IS THE LAST BOX ASSOCIATED WITH THE CONTRO.
C POINT
C NDCDCN = CODE FOR CONTROL POINT
C NDCDCN = CODE FOR THE CURRENT SUBDIVIDED BOX
C  IICENT = 1-LOCATION (SUBDIVIDED) FOR THE NEAREST PLANFORM
C CONTROL POINT
C
C  LBOX = .F.
C  JUCENT = (JCO - NSUBCV) / SUBDIV + 1.5
C  JCENT = NSUBCV + JUCENT - NSUBCV
C  IF (.NOT. CROW) GO TO 410
C  IF (JCENT .NE. JCOL) GO TO 410
C  CBOX = T.
C  NJDCEN = NCDBOX
C  JICENT = INOW
C  IF (NCDBOX) 450,1100,450
C   GET CENTER BOX CODE, NJDCEN
C
410 CBOX = .F.
C  NJDCEN = 0
C  IF (JCENT .LE. 0) GO TO 414
C  CALL DCORDER (IBOX,LBKD, JCENT, JCENT, JCENT, .T., NJDCEN)
C  JICENT = JCENT
C  GO TO 418
C
414 IICENT = IXC - NSUBCV
C
418 CONTINUE
C  IF (NCDBOX - 1) 420,424,450
C  SUBDIVIDED BOX IS NOT CONSIDERED (CODE = 0), IF CENTER CODE
C  IS ALSO ZERO, LOOP TO NEXT BOX. OTHERWISE, CHECK FOR LAST BOX
C  GO TO 418
C
420 IF (NCDCEN) 490,1100,450
C
424 CONTINUE
C  SUBDIVIDED BOX CODE = 1. CHECK WHETHER ITS CONTROL PT = 1
C
424 CONTINUE
C  IF (.NOT. CPLAN) GO TO 431
C  IF (JEXLOC(JEXLOC) .LT. FLRGM) JEXLOC = JEXLOC + HBYSW
C  IF (JEXLOC(JEXLOC) .GT. FLRGM) JEXLOC = JEXLOC - HBYSW
C  DETERMINE WHETHER SUBDIVIDED BOX IS ON SAME PLANFORM AS
C  IICENT (LOCATION OF NEAREST CONTROL POINT)
C  IF (JEXLOC .EQ. JCOL) GO TO 428
C  SUBDIVIDED BOX IS ON THE TAIL. IS IICENT ON THE WING -
C  IF (FLOAT(IICENT) .LE. TEXLOC(JCENT)) GO TO 432
C  NO. CHECK FOR IICENT OFF-PLANFORM.
C  GO TO 431
C
428 IF (FLOAT(IICENT) .GT. TEXLOC(JCENT)) GO TO 432
C
431 CONTINUE
C  CHECK CODE AT IICENT (NEAREST CONTROL POINT)
C  IF (NJDCEN .EQ. 1) GO TO 450
C
431 CONTINUE
C  SUBDIVIDED ON-PLANFORM BOX DOES NOT LIE WITHIN AN UNSUBDIVIDED
C  BOX MORE CONTROL POINT IS ON PLANFORM, SEARCH FOR AND AFT
C  FOR THE NEAREST CONTROL POINT ON THE SURFACE.
C
432 CONTINUE
C  IFCOM = IFIX(FEXLOC(JEXLOC)) + 1
C  ILCOM = TEXLOC(JEXLOC)
C  IMAX = 2 * NSUBCV
C  DO 430 I = NSUBCV,IMAX - SUBDIV
C  IICENT = IICENT +
C  IF (JICENT .NE. ICOM) GO TO 434
C  CALL DCORDER (IBOX,LBKD, IICENT, JCENT, IICENT, JCENT, .T., NJDCEN)
C  IF (NCDCEN .EQ. 1) GO TO 440

434 CONTINUE
   ICENT = ICENT-I
   IF (ICENT .LT. IFCON) GO TO 438
   CALL DCC&R(IBO,LBXCD,ICENT,JCENT,ICENT,JCENT,T,NCC)
   IF (NCC .EQ. 1) GO TO 440

438 CONTINUE
   C  NO CENTER BOX FOUND. A WARNING DIAGNOSTIC WILL BE PRINTED,
   C  THEN COMPUTATION WILL CONTINUE AT 420
   C  GO TO 3010.
   C  A BOX WAS FOUND
   440 CONTINUE
   C  THE ASSOCIATED CONTROL POINT HAS BEEN FOUND; GET THE UNSUB-
   C  DIVIDED SUBSCRIPT.
   450 CONTINUE
   IPCENT = (ICENT-1BWR)/NSUBDV + 1
   IF (LBRO) GO TO 470
   IF (.NOT. FROM ) GO TO 460
   IF (JCOL .NE. JCENT+NSUB2) GO TO 460
   FBOK = .T.
   GO TO 500

460 CONTINUE
   FBOK = .F.
   GO TO 500

470 CONTINUE
   IF (JCOL .EQ. JCENT+NSUB2) LBOK = .T.
   FBOK = .F.
   GO TO 500

C  -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

C  SET UP VALUES FOR AN UNSUBDIVIDED CASE
C  TEST FOR NON-ZERO BOX CODE -
   480 CONTINUE
   IF (NEBOX .EQ. 0) GO TO 1100
   ICENT = IRG
   ICENT = ICENT
   ICENT = ICENT
   JCENT = JCOL
   JCENT = JCOL
   LBOK = .T.
   LBOK = .T.
   NSCDEN = NEDBOX
   FBOK = .T.
   IPCENT = ICENT

C  500 CONTINUE
   IF (NEBOX .GT. 0) GO TO 510
   FULLB(X(JCENT)) = .F.
   IF (LBOK) GO TO 1040
   IF (FBOK) GO TO 515
   GO TO 1100

C  C  THE BOX IS TO BE CONSIDERED. ARE N-HAT TERMS NECESSARY -
   510 CONTINUE
C  C  ARE N-HAT TERMS ALREADY AVAILABLE -
   IF (FBOK) GO TO 515

B138
IF (NCIBOX .NE. 1) GO TO 890
IF (.NOT. WING) GO TO 890
IF (JUCENT .NE. JUCENT) GO TO 830
GO TO 890
C
C                        COMPUTE H-HAT TERMS FOR THIS (UNSUBDIVIDED) BOX
C
515 CONTINUE
ENULL(JUCENT) = (0.,0.)
ENULL(JUCENT) = (0.,0.)
ENULL(JUCENT) = (0.,0.)
C ARE LEFT SURFACE CONTRIBUTIONS POSSIBLE -
IF (JUCENT-IXBU .LT. JUCENT) GO TO 800
IF (PSIS .EQ. 0 .OR. .NOT. DIHS) GO TO 800
C
C GET AIC ARRAYS W AND V FOR LEFT SURFACE INFLUENCE ON RT SURFACE
CALL GETAIC(JUCENT,WW, D, IR)
IF (IR .NE. 0) GO TO 900
MJMN = JUCENT - IXBU
YMUBAR = COS(2*PSIS)* (JUCENT-.5)
JBAR = YMUBAR + 1
C
C GET REFERENCE LOCATION IN AIC ARRAYS
IF (YBAR) 520, 525, 530
520 JINCX = 1
GO TO 535
C
C IAIC = NUBMIN+2
INCAIC = 2*NUBMIN + 1
JINOX = -1
GO TO 540
530 JINOR = -1
535 IAIC = NUBMIN+2 + NUBMIN
INCAIC = 2*NUBMIN + 2
C
C LOOP FORWARD OF BOX FOR WING/WING (TAIL/TAIL) H-HAT TERMS
C
540 CONTINUE
DO 550 NUBAR = NUBMIN, NUBMAX
MJIC1 = MJIC(1,NUBAR+1)
MJIC2 = MJIC(2,NUBAR+1)
IF (MJIC2 .LE. 0) GO TO 585
IF (YBAR .GE. 0) GO TO 550
JCQCL = -JBAR - NUBAR + MJIC1
GO TO 550
550 JCQCL = -JBAR - NUBAR + MJIC1 + 2
540 CONTINUE
C
C LOOP LEFT OF RECEIVING CHORD TO GET LEFT SURFACE CONTRIBUTIONS
C
DO 560 MJAI = MJIC1, MJIC2
IF (JCQCL .LE. 0) GO TO 570
CALL DECODER(BOX, LBXC, 1, JCQCL, 1, JCQCL, F, ICD)
IF (ICD .EQ. 0) GO TO 570
C
C A CONTRIBUTING BOX HAS BEEN FOUND. GET THE AIC LOCATION
RAIC = IAIC + MJAI
C
C GET LOCATION IN N ARRAYS FOR THE VALUES AT BOX (1,JCQCL)
ICB = LCOSD2(JCQCL, I, PNTDW, LNTDW, 1, LNTDW)
VELPOT 00316
VELPOT 00317
VELPOT 00318
VELPOT 00319
VELPOT 00320
VELPOT 00321
VELPOT 00322
VELPOT 00323
VELPOT 00324
VELPOT 00325
VELPOT 00326
VELPOT 00327
VELPOT 00328
VELPOT 00329
VELPOT 00330
VELPOT 00331
VELPOT 00332
VELPOT 00333
VELPOT 00334
VELPOT 00335
VELPOT 00336
VELPOT 00337
VELPOT 00338
VELPOT 00339
VELPOT 00340
VELPOT 00341
VELPOT 00342
VELPOT 00343
VELPOT 00344
VELPOT 00345
VELPOT 00346
VELPOT 00347
VELPOT 00348
VELPOT 00349
VELPOT 00350
VELPOT 00351
VELPOT 00352
VELPOT 00353
VELPOT 00354
VELPOT 00355
VELPOT 00356
VELPOT 00357
VELPOT 00358
VELPOT 00359
VELPOT 00360
VELPOT 00361
VELPOT 00362
VELPOT 00363
VELPOT 00364
VELPOT 00365
VELPOT 00366
VELPOT 00367
VELPOT 00368
VELPOT 00369
VELPOT 00370
VELPOT 00371
VELPOT 00372
B139
C COEF = ( CPSIS2*K(ICA) - SPIS2*K(ICA) ) * SYMTY VELPOT 00373
EMRUL(JUCENT) = EMRUS(IDS)*COEF + EMRUL(JUCENT) VELPOT 00374
EMRLL(JUCENT) = EMRLS(IDS)*COEF + EMRLL(JUCENT) VELPOT 00375
DPMIL(JUCENT) = EMRUS(IDS) - EMRLS(IDS) * C(ICA) * SYMTY +
1 VELPOT 00376
570 CONTINUE VELPOT 00377
C JCOLL = JCOLL + JINCR VELPOT 00378
580 CONTINUE VELPOT 00379
C END OF LOOP FOR LEFT ROW CONTRIBUTIONS VELPOT 00380
C 585 CONTINUE VELPOT 00381
I = I - 1 VELPOT 00382
IF (I .LT. IXU) GO TO 600 VELPOT 00383
IAIC = INCAIC + IAIC VELPOT 00384
INCAIC = INCAIC + 2 VELPOT 00385
590 CONTINUE VELPOT 00386
C END OF LOOP FORWARD ON ROWS, TO COMPUTE LEFT SURFACE CUT-OF-
PLANE EFFECTS, FROM 540 VELPOT 00387
C 600 CONTINUE VELPOT 00388
IF (WING) GO TO 830 VELPOT 00389
ENRURW(JUCENT) = (0.,0.) VELPOT 00390
ENRULW(JUCENT) = (0.,0.) VELPOT 00391
IF (IXUEN .NE. 1) GO TO 860 VELPOT 00392
C COMPUTE THE RIGHT WING CONTRIBUTION TO THE TAIL BOX VELPOT 00393
II = 1 VELPOT 00394
IF (PSIW .EQ. PSIT) II = 2 VELPOT 00395
CALL GETAI(JUCENT,RWT, II, IR) VELPOT 00396
IF (IR .NE. 0) GO TO 700 VELPOT 00397
NUMIN = ABS(EL) + .5 VELPOT 00398
NUMAX = TUCENT - 1 VELPOT 00399
I = TUCENT - NUBMIN VELPOT 00400
YMUAR = (JUCENT - .5) * COS(PSIDIF) + CAR.* SIN(PSIW) VELPOT 00401
JBAR = YMUAR VELPOT 00402
IF (YMUAR .GE. 0) JBAR = JBAR + 1 VELPOT 00403
IF (YBAR) 620,625,630 VELPOT 00404
620 JINCR = -1 VELPOT 00405
GO TO 635 VELPOT 00406
625 IAIC = NUBMIN +2 VELPOT 00407
INCAIC = 2*NUBMIN + 1 VELPOT 00408
JINCR = 1 VELPOT 00409
GO TO 640 VELPOT 00410
630 JINCR = 1 VELPOT 00411
635 IAIC = NUBMIN +2 + NUBMIN VELPOT 00412
INCAIC = 2*NUBMIN + 2 VELPOT 00413
640 CONTINUE VELPOT 00414
C LOOP FORWARD OVER THE RIGHT WING VELPOT 00415
DO 690 NUBAR = NUBMIN,NUMAX VELPOT 00416
NuAiC1 = NuAiC(NUBAR+1) VELPOT 00417
NuAiC2 = NuAiC(NUBAR+1) VELPOT 00418
IF (NuAiC2 .EQ. 0) GO TO 685 VELPOT 00419
B140
IF (VBAR .GE. 0) GO TO 650
   JCOLR = JBAR + NUBAR + MUACIZ + 1
   GO TO 660
650 JCOLR = JBAR - NUBAR - MUACIZ - 1
660 CONTINUE

C LOOP ON A ROW OF WING BOXES, COMPUTING RIGHT HAND WING-TAIL
   CONTRIBUTION
   DO 860 MUAI = MUAC1,MUAC2
   IF (JCOLR .LE. 0) GO TO 670
   CALL DECODER(IBOX, LBNCDW, I, JCOLR, I, JCOLR, .F., ICD)
   IF (ICD .LE. 0) GO TO 670
   C A CONTRIBUTING BOX HAS BEEN FOUND, GET THE AIC LOCATION
   KAIC = IAIC + MUAI
   GO TO 6W0 VELPOT 00432
860 CONTINUE

   SW0 JC(XR = JBAR - NUBAR + IC2 + I
   VELPOT 00433
   £60 CJ5LI 1UE
   VELPOT 00434
C SW0 A CONTRIBUTING BOX HAS BEEN FOUND, GET THE AIC LOCATION
   KAIC = IAIC + MUAI
   GO TO 6W0 VELPOT 00439
   IF (JCICR .LE. 0) GO TO 670
   ENURJW(JUCENT) = (COS(PSIDIF)*KAI C) - SIN(PSIDIF)*KAI C)
   1  = ENURJW(IDS) + ENURJW(JUCENT)
   GO TO 670
665 CONTINUE
   ENURJW(JUCENT) = COS(PSIDIF)*KAI C) * ENURJW(IDS) +
   1  ENURJW(JUCENT)
670 CONTINUE
   JCOLR = JCOLR + IJMPR
   680 CONTINUE
C END OF LOOP FOR RIGHT WING ROW CONTRIBUTIONS
695 CONTINUE
   T = I - 1
   IF (I .LE. 0) GO TO 700
   IAIC = IAIC + INCAIC
   INCAIC = INCAIC + 2
   690 CONTINUE
C END OF LOOP FORWARD ON ROWS, TO COMPUTE RIGHT WING OUT-OF-
   PLANE EFFECTS ON THE TAIL, FROM 640
C DETERMINE WHETHER LEFT WING INFLUENCE IS TO BE COMPUTED
700 CONTINUE
   IF (SYM .EQ. 0) GO TO 800
   VELPOT 08
C GET AIC ARRAYS W AND : FOR LEFT WING INFLUENCE ON TAIL
   II = 1
   IF (-PhIW .EQ. 0, PSIT) II = 2
   CALL SETAIC(JUCENT, LNT, II, IR)
   IF (IR .NE. 0) GO TO 800
   MUBMIN = ABS(EL) + .5
   MUBMAX = JUCENT - 1
   I = JUCENT - MUBMIN
   YMUBAR = - COS(PSI W + PSIT)*(JUCENT-.5) + CAPPSIN(PSIT)
   JBAR = YMUBAR
   IF (YMUBAR .LE. 0) JBAR = JBAR + 1
   IF (VBAR) 720,725,730
720 JINCR = 1
   GO TO 735
725 IAIC = MUBMIN + 2
   INCAIC = 2+MUBMIN + 1
   VELPOT 0465
JIMCR = -1
GO TO 740
730 JIMCR = -1
735 IAIC = NUBMIN + NUBMAX + NUBMIN
INCAI = 2*NUBMIN + 2
740 CONTINUE
C
C LOOP FORWARD TO GET LEFT WING CONTRIBUTION TO THE TAIL
IF (SYM .NE. 0) GO TO 800
DO 790 NUBAR = NUBMIN,NUBMAX
MUAI1 = MUAI1(1,NUBAR+1)
MUAI2 = MUAI2(1,NUBAR+1)
IF (MUAI1 .LE. 0) GO TO 785
IF (MUAI2 .LE. 0) GO TO 790
JCOLL = JBAR - NUBAR + MUAI1 - 1
GO TO 760
750 JCOLL = JBAR + NUBAR - MUAI1 + 1
760 CONTINUE
C
C LOOP ON LEFT WING ROW TO GET LEFT WING CONTRIBUTION TO TAIL
DO 780 MAI = MUAI1,MAI2
IF (JCOLL .LE. 0) GO TO 770
CALL DCEDER(IBOW,LBEXD,M,JCOLL,1,JCOLL,1,JCOLL,1,F,F,1C)
IF (F .NE. 0) GO TO 770
A CONTRIBUTING BOX HAS BEEN FOUND. GET THE AIC LOCATION
KAIC = IAIC + MAI
C GET THE NORMAL WASH LOCATION
IDS = LOCSDM I, JCOLL, I, JCOLL, I, JCOLL, 1, JCOLL
C ADD THIS CONTRIBUTION TO N-HAT
IF (1 .NE. 2) GO TO 765
ENRUW(JUCENT) = (CPISHAW(KAIC) - SPSHAW(KAIC))
1 = ENRU(JDS) + ENRUW(JUCENT)
GO TO 770
765 CONTINUE
ENRUW(JUCENT) = CPISHAW(KAIC) + ENRU(JDS) + ENRUW(JUCENT)
770 CONTINUE
JCOLL = JCOLL + JIMCR
780 CONTINUE
C END OF LOOP FOR LEFT WING ROW CONTRIBUTIONS
C
785 CONTINUE
I = I-1
IF (I .LE. 0) GO TO 800
IAIC = IAIC + INCAI
INCAI = INCAI + 2
790 CONTINUE
C END OF LOOP "FORWARD ON ROWS, TO COMPUTE LEFT WING OUT-OF-PLANE EFFECTS ON THE TAIL, FROM 740
C EFFECTS ON THE TAIL, FROM 740
C
IF (SYM .NE. 0) ENRUW(JUCENT) = - ENRUW(JUCENT)
800 CONTINUE
C
C COMPUTE THE UNSUBDIVIDED NORMAL WASH VALUES, IF THE BOX IS ON- LOCATION
C
C PLANE. IF NOT, GET THE VALUE FROM THE INTERFERENCE TERMS
C AND THE CONDITION THAT DELTA-PHI = 0 ON ANY DIAPHRAGM, WENTAI-
C
C
ED BY MAE EFFECTS WHERE APPLICABLE.

820 CONTINUE

IF (NCRBOX .NE. 1) GO TO 860
GET DEFORMATION AND SLOPE OF UNSUBDIVIDED TAIL BOX CENTER
IDS = LOGSW/IPCENT+1,LPNLP, JUCENT, IPNTRM, LMODES,1,LMODES)
DPL = DEFS2(1,IDS)
SPL = DEFS2(2,IDS)

C COMPUTE TAIL NORMAL WASH VALUES
ENDIF = 2.0*CMPLX(B1*SLP, XXVL*DPL) = ENRV(UJUCENT)

1 ENSUM = - (ENULU(JUCENT)) + ENULU(JUCENT) -ENRV(UJUCENT) * 2.0

C IF (IPCENT .NE. JUCENT) GO TO 852

LOCW = LOGSW/JUCENT+1,LPNLP, JUCENT, IPNTRM, LPNTDW, 1, LPNTDW
ENRUS(LCOW) = (ENSUM + ENDIF) * 0.5
ENRSL(LCOW) = (ENSUM - ENDIF) * 0.5
GO TO 850

C - WING --

830 CONTINUE

IF (NCRBOX .NE. 1) GO TO 860
GET DEFORMATION AND SLOPE OF UNSUBDIVIDED WING BOX CENTER
IDS = LOGSW/IPCENT, JUCENT, IPNTRM, LMODES,1,LMODES)
DPL = DEFS2(1,IDS)
SPL = DEFS2(2,IDS)

C COMPUTE WING NORMAL WASH VALUES
ENSUM = -ENRSL(JUCENT) -ENULU(JUCENT)
ENDIF = ENDIF + CMPLX(B1*SLP, XXVL*DPL) + ENRSL(JUCENT)) * 2.0

C IF (IPCENT .NE. JUCENT) GO TO 852

LOCW = LOGSW/JUCENT+1,LPNLP, JUCENT, IPNTRM, LPNTDW, 1, LPNTDW
ENRUS(LCOW) = (ENSUM + ENDIF) * 0.5
ENRSL(LCOW) = (ENSUM - ENDIF) * 0.5
GO TO 850

C NORMAL WASH IS AVAILABLE IF THE BOX IS ON PLAN FORM

850 CONTINUE

IF (.T. FROM LOCW = LOGSW(JUCENT+1,LPNLP, JUCENT, IPNTRM, LPNTDW, 1, LPNTDW, 1,LPNTDW)

1 IF (SUBOFF) GO TO 855
C GET THE SUBDIVIDED VALUE FOR THE NORMAL WASH TERMS
DELS = ENRSL(LCOW)
IF (DELS .EQ. XUNIT) GO TO 830

C THE NEXT 2 STATEMENTS ARE ONLY HIT FOR A SUBDIVIDED PLAN FORM

C BOX WITH NO ASSOCIATED PLAN FORM CONTROL POINT

832 CONTINUE

FULLX(JUCENT) = .FALSE.
DELM = (ENSUM + ENDIF) * 0.5
DELM = (ENSUM - ENDIF) * 0.5

834 CONTINUE

ENRUS = CMPLX(REAL(DELPHA)/XSUDD, AIMAG(DELPHA)/XSUDD) +
1 * XXVL*(IPCENT*REAL(DELPHA))
ENRSL = CMPLX(REAL(DELPHB)/XSUDD, AIMAG(DELPHB)/XSUDD) +
1 * XXVL*(IPCENT*REAL(DELPHB))
IF (CBX) GO TO 870
GO TO 1030
855 CONTINUE
ENGRUS = ENGRUS(CBXX)
ENGRLS = ENGRLS(CBXX)
GO TO 870
850 CONTINUE
FULLBX(JUCENT) = .F.
C
C DETERMINE THE VELOCITY POTENTIAL CONTRIBUTIONS FROM
C BOXES LYING AHEAD OF THE CURRENT BOX
C
870 CONTINUE
IF (G61, NE. 0. AND, DIMS) GO TO 880
C THE SURFACE IS PLANAR, GET FULL SURFACE CONTRIBUTIONS
DELPH = BIROW, JCL, PKRN, SKEBN, IBX, LBBX, WING, .F. )
GO TO 890
C
C DIHEDRAL ANGLE IS TO BE ACCOUNTED FOR, GET THE PLANAR
C (SUBLIZED) CONTRIBUTION OF THE RIGHT SURFACE
880 CONTINUE
DELPH = BIROW, JCL, PKRN, SKEBN, IBX, LBBX, WING, .T. )
ADD THE SPATIAL LEFT SURFACE CONTRIBUTIONS
DELPH = DELPH + DRPH(JUCENT)
890 CONTINUE
IF (NCBOX = 2) 910, 1000, 980
C
C THE BOX IS ON-PLANAR, CENTER, COMPLETE THE CALCULATION OF
C THE VELOCITY POTENTIAL
C
910 CONTINUE
IDS = LOCEN(JUCENT+1, JCL, JUCENT, IPNTR, L Modes, i, L Modes)
IF (SUBOFF) GO TO 915
DELPH(1IDS) = (ENGRUS, ENGRLS) * SKEBN, (1) + DELPH
GO TO 920
915 CONTINUE
DELPH(1IDS) = (ENGRUS, ENGRLS) * PKRN, (1) + DELPH
920 CONTINUE
C
C COMPUTE ANY TRAILING EDGE VELOCITY POTENTIALS ASSOCIATED
C WITH THIS UNDIVIDED BOX
C
C IS THIS A TRAILING EDGE BOX -
IF (.NOX, COPLAN) GO TO 930
IF (TERLOC(JERLOC) .LT. FLIROW) JERLOC = JERLOC + MARK
IF (TERLOC(JERLOC) .GT. FLIROW) JERLOC = JERLOC - MARK
930 CONTINUE
JJ = JERLOC - NSUDE
TERM1 = TERM1(JJ)
IF (NSUDE, .EQ. 1) GO TO 935
DO 932 J = 2, NSUDE
JJ = JJ + 1
TERM1 = TERM1(JJ)
932 CONTINUE
935 CONTINUE
B144
932 CONTINUE
933 IF (TYPMIN.GT.0) GO TO 1030
C  YES. GET THE BASIC VELOCITY POTENTIAL
C  DELPHI = DELPHI(IDS)
935 IF (NRAD .NE. 1) GO TO 940
C  IT IS ON PLANIFORM. TEST FOR TIP CHORD -
C  IF (JEXO .GE. JCCL .AND. JUCENT .LT. NYBO) GO TO 950
C  IF (JEXO .GT. JCCL .AND. JUCENT .LT. MYST) GO TO 950
C  BOX IS ON THE TIP CHORD. CHECK FOR THIRD TIP BOX -
C  CALL DCDECER(IBOX,LBXCD, IR, JCCL, IR, JCCL, .T., NCDC)
C  IF (NCDC .NE. 1) GO TO 950
C  TRY MACH RAY EXTRAPOLATION. ARE THE 2 RAY BOXES ON-PLANFORM -
C  JC = JCCL - NSUBDV
C  CALL DCDECER(IBOX,LBXCD, IR, JC, IR, JC, .T., NCDD)
C  IF (NCDD .NE. 1) GO TO 945
C  JC = JC - NSUBDV
C  CALL DCDECER(IBOX,LBXCD, IR+NSUBDV, JC, IR+NSUBDV, JC, .T., NCDD)
C  IF (NCDD .NE. 1) GO TO 945
C  MACH RAY EXTRAPOLATION, FOLLOWED BY CHORDWISE LINEAR EXTRA-
C  PHAT = LOC(SWJUCENT-1+1,1,LMODES,0,LMODES)
C  PHAT = LOC(SWJUCENT+1,1,LMODES,0,LMODES)
C  DELPHI = DELPHI(PHAT)
C  DELPH = DELPHI(PHAT)
C  SDELPH = (DELPH - DELPHI)*XSUBDV
C  GO TO 955
C  MACH RAY UNAVAILABLE. ARE THERE 2 BOXES ON THE CHORD -
945 IF (NCDA .NE. 1) GO TO 3020
C  CHORDWISE LINEAR EXTRAPOLATION
950 CONTINUE
C  X = JEXO .LT. IR - 2*XSUBDV
C  CALL DCDECER(IBOX,LBXCD, IR, JCCL, IR, JCCL, .T., NCDC)
C  IF (NCDC .NE. 1) GO TO 950
C  TRY MACH RAY EXTRAPOLATION. ARE THE 2 RAY BOXES ON-PLANFORM -
C  JC = JEXO - NSUBDV
C  CALL DCDECER(IBOX,LBXCD, IR, JC, IR, JC, .T., NCDD)
C  IF (NCDD .NE. 1) GO TO 945
C  JC = JC - NSUBDV
C  CALL DCDECER(IBOX,LBXCD, IR-NSUBDV, JC, IR-NSUBDV, JC, .T., NCDD)
C  IF (NCDD .NE. 1) GO TO 945
C  MACH RAY EXTRAPOLATION, FOLLOWED BY CHORDWISE LINEAR EXTRA-
C  PHAT = LOC(SWJUCENT-1+1,1,LMODES,0,LMODES)
C  PHAT = LOC(SWJUCENT+1,1,LMODES,0,LMODES)
C  DELPHI = DELPHI(PHAT)
C  DELPH = DELPHI(PHAT)
C  SDELPH = (DELPH - DELPHI)*XSUBDV
C  GO TO 955
C  LOOP TO COMPUTE AND STORE TRAILING EDGE VELOCITY POTENTIALS
955 CONTINUE
C  JA = JEXO - NSUBDV
C  JB = JEXO + NSUBDV
C  DO 970 JJ = JA, JB
C  XINOR = JEXO(JJ) .LT. IR
C  IF (XINOR .LT. -XSUBDV/2.0) GO TO 960
C  TYP(JJ) = DELPH + XINOR*SDELPH
970 CONTINUE
C  ALL TRAILING EDGE VALUES HAVE BEEN COMPUTED FOR THIS
C  UNSUBDIVIDED BOX.
C  GO TO 1030
C
C BOX IS IN THE TRAILING EDGE DIAPHRAGM AREA. COMPUTE THE WAKE VELOCITY POTENTIAL CONTRIBUTION

980 CONTINUE VELPOT 00715
IF (COPLAN) GO TO 983 VELPOT 00716
JL = JEWLO C VELPOT 00717
GO TO 980 VELPOT 00718

983 JJ = JCOL VELPOT 00719
IF (JL , GT, MBST) GO TO 990 VELPOT 00720
JJT = JJ + MB5W VELPOT 00721
IF (FLOAT(IROW), GT, TELOC(JJT)) JJ = JJT VELPOT 00722
990 CONTINUE VELPOT 00723
C COMPUTE (X-DISTANCE/B1) * K1

XDKVL = (FLOAT(IROW)-TELOC(JJ)) * WKL*SUBDV VELPOT 00724
AZERO = TWP(JJ) + CNPLX(COS(XDKVL), -SIN(XDKVL)) VELPOT 00725
IF (.NOT. CBOK) GO TO 1010 VELPOT 00726
IF (JUENT+1OMLP, GT, NHTR) GO TO 1010 VELPOT 00727
C SET DELPHI VALUE TO ZERO TO (CLEAR UP LEFT OVER MACE SHAPES)
1 = LOCSIN(JUENT, IOMLP, JUENT, IPHTRM, LMODES, I,MODES) VELPOT 00728
IF (I, NE, 0) DELPHI(I) = (0., 0.) VELPOT 00729
GO TO 1010 VELPOT 00720
C BOX IS IN A LEADING EDGE OR TIP DIAPHRAGM AREA
1000 CONTINUE VELPOT 00731
AZERO = (0., 0.) VELPOT 00732
C COMPUTE NORMAL-WASH VALUES FOR A (SUBDIVIDED) DIAPHRAGM BOX
1010 CONTINUE VELPOT 00733
IF (SUBOFF) GO TO 1015 VELPOT 00734
ENDIF = (AZERO - DELPHI) / SKFRML(11) VELPOT 00735
ENSUM = - (ENRULJ(JUENT) + ENRLLL(JUENT)) VELPOT 00736
ENSUM = CNPLX(REAL(ENSUM), AIMAG(ENSUM)) + WVL*SUBDV/(IROW - 1)
ICONT) *REAL(ENSUM) / XSUBDV VELPOT 00737
GO TO 1020 VELPOT 00738
1013 CONTINUE VELPOT 00739
ENDIF = (AZERO - DELPHI) / PKFRVL(11) VELPOT 00740
ENSUM = - (ENRULJ(JUENT) + ENRLLL(JUENT)) VELPOT 00741
C 1020 CONTINUE VELPOT 00742
ENRUL = 0.5*(ENSENUM + ENDIF) VELPOT 00743
ENRULL = 0.5*(ENSENUM - ENDIF) VELPOT 00744
IF (SUBOFF) GO TO 1030 VELPOT 00745
GO TO 1035 VELPOT 00746
C STORE THE NORMAL WASH VALUES VELPOT 00747
C 1030 CONTINUE VELPOT 00748
IF (SUBOFF) GO TO 1100 VELPOT 00749
C STORE THE COMPUTED SUBDIVIDED NORMAL WASHES VELPOT 00750
1035 CONTINUE VELPOT 00751
CALL STOSW (IROW, JCOL, ENRUL, IBOK, IBXCD, IXBS, MB5BS, MBBS, IRR) VELPOT 00752
IF (I, NE, 0) GO TO 3030 VELPOT 00753
IF (.NOT. IBOK) GO TO 1100 VELPOT 00754
B146
C DETERMINE THE UNSUBDIVIDED NORMAL WASH VALUES

1040 IF (ICENT = 1) 1040, 1050, 1060
C IF THE BOX IS COMPLETE, THE VALUE IS ALREADY STORED -
1050 IF (FULLBOX(JUCENT)) GO TO 1100
C THE UNSUBDIVIDED VALUE EQUALS THE AVERAGE OF ALL ITS
C SUBDIVIDED BOXES

1060 CONTINUE
II = ICENT - NSUBD2
JJJ = JCENT + NSUBD2
ENGUS = (0., 0.)
ENGRLS = (0., 0.)
DO 1080 I = 1, NSUBD2
JJ = JCENT - NSUBD2
CALL DCOOR3(IBOX, LBOXCD, II, JJ, I,, JJJ, J,, IC0DE)
DO 1075 J = 1, NSUBD2
IF (IC0DE(J) .EQ. 0) GO TO 1070
LOCNM = LOCSDW(II, JJ, IPNTFL, IPNTIN, IPNTOT, IPNTLS)
ENGUS = ENSUBD1(LOCNM) + ENRUS
ENGRLS = ENSUBD2(LOCNM) + ENGRLS
1070 CONTINUE
JJ = JJ + 1
1075 CONTINUE
II = II + 1
1080 CONTINUE
ENGUS = ENGUS/NSUBD2
ENGRLS = ENGRLS/NSUBD2
C RESTORE THE PARTIAL BOX FLAG FOR THE NEXT ROW
FULLBX(JUCENT) = .T.
C STORE THE UNSUBDIVIDED NORMAL WASHES
1090 CONTINUE
LOCNM = LOCSDW(JUCENT+IOMLPH, JUCENT, IPNTDW, LPNTDW, 1, LPNTDW)
IF (LOCNM .EQ. 0) GO TO 1040
ENGUS(LOCNM) = ENRUS
IF (IXNG) GO TO 1095
ENGRLS(LOCNM) = ENGRLS
GO TO 1100
1095 ENGRLS(LOCNM) = ENGRLS
1100 CONTINUE
JEXLOC = JEXLOC + 1
C
C END OF LOOP ON (SUBDIVIDED) CHORDS, STARTING AT 3556
C
C END OF LOOP ON (SUBDIVIDED) ROWS, STARTING AT 120
C
C RETURN
C
C DIAGNOSTICS - ALL CALL FLUH
C

3010 WRITE (NT6,9010)
VLPOT 00829
WRITE (NT6,9999) IROW,JCOL, IUCENT, JUCENT
VELPOT 00830
GO TO 420
VELPOT 00831
VELPOT 00832
3020 WRITE (NT6,9020)
VELPOT 00833
GO TO 3999
VELPOT 00834
3030 WRITE (NT6,9030)
VELPOT 00835
GO TO 3999
VELPOT 00836
3040 WRITE (NT6,9040)
VELPOT 00837
GO TO 3999
VELPOT 00838
C
VELPOT 00839
3999 WRITE (NT6,9999) IROW,JCOL, IUCENT, JUCENT
VELPOT 00840
CALL FASTFLUSH(1)
VELPOT 00841
C
VELPOT 00842
VELPOT 00843
9010 FORMAT(5SH4)*** WARNING - NO PLANFORM CONTROL POINT FOUND FOR SUBDIV
VELPOT 00844
1 5SUBDIVIDED BOX DURING VELOCITY POTENTIAL CALCULATIONS *** )
VELPOT 00845
9020 FORMAT(5SH4)*** ERROR - THE TIP BOX PATTERN DOES NOT ALLOW TRAILING
VELPOT 00846
1 44H EDGE VELOCITY POTENTIALS TO BE COMPUTED *** )
VELPOT 00847
9030 FORMAT(5SH4)*** ERROR - FAILURE IN STORING SUBDIVIDED NORMAL-WASHES
VELPOT 00848
1 1H *** )
VELPOT 00849
9040 FORMAT(5SH4)*** ERROR - FAILURE IN STORING CONTROL POINT NORMAL-
VELPOT 00850
1 1SH-WASHES *** )
VELPOT 00851
9999 FORMAT(14X,1G)SUBDIVIDED BOX (I3,I4,I3,I4), CONTROL POINT (I3,
VELPOT 00852
1 I2,I1H,I2,I1H)
VELPOT 00853
C
VELPOT 00854
END
VELPOT 00855
COMPLEX FUNCTION B (IRON, IJOL, PKERNL, IBOK, LBXC, 1, WING, DIN) B 00002

COMPUTES B = SUM OVER IJUX(SUM OVER JUX((DONUTASH) * (KERNEL))) B 00003

N, M.E. B 00004

B IS USED TO COMPUTE VELOCITY POTENTIALS OF ON-

COMPUTER BOXES, OR DONUTASHES OF DIAPHRAGM BOXES B 00005

PARAMETERS - B 00006

IRON = ROW LOCATION OF BOX FOR WHICH B IS TO BE B 00007

COMPUTED B 00008

JCOL = COLUMN LOCATION OF BOX B 00009

PKERNL = PRIMARY KERNEL ARRAY B 00010

SKERNL = SUBDIVIDED KERNEL ARRAY B 00011

IBOK = ARRAY OF BOX CODES B 00012

LBXC = LENGTH OF BOX CODE ARRAY B 00013

WING = .T., WING, ...F., TAIL B 00014

DIN = .T., LEFT SIDE TO BE IGNORED (SURFACE HAS DIHRL) B 00015

= .F., LEFT SIDE TO BE INCLUDED B 00016

VALUES FROM COMMON - B 00017

NSUBDV = NUMBER OF SUBDIVISIONS B 00018

TDB = CENTER OF FIRST SUBDIVIDED BOX RELATIVE TO THE B 00019

SUBDIVIDED PATTERN B 00020

NSUBR = NUMBER OF SUBDIVIDED ROWS B 00021

WBB = NUMBER OF SUBDIVIDED CODES, INCLUDING DIAPHR. B 00022

WBBS = NUMBER OF SUBDIVIDED CODES, INCLUDING DIAPHR. B 00023

PKRKN = SIZE OF SUBDIVIDED KERNEL B 00024

SYM = SYMMETRY INDICATOR B 00025

LUX = NSUBDV + NSUBDV/2 + 1 B 00026

ENSLD = SUBDIVIDED NORMAL-WASHES B 00027

ENSUS, EMBLS = UNSUBDIVIDED NORMAL-WASHES B 00028

IPMTW = POINTER ARRAY FOR UNSUBDIVIDED NORMAL WASHES B 00029

IPMTSD = POINTER ARRAY FOR SUBDIVIDED NORMAL WASHES (END- B 00030

AROUND B 00031

IPMTIN = NEXT AVAILABLE POINTER B 00032

IPMTOR = FIRST POINTER IN USE B 00033

IPMTLS = DIMENSION OF ARRAY IPMTSD B 00034

LINSD = DIMENSION OF SUBDIVIDED NORMAL-WASH ARRAYS B 00035

COMMON /GEOM/ / COPLAN, NSUBDV, XSUBDV, ySUBDV, NSUBK, NSURF, GEOMY 00036

1 B1,BIBETA,B2,BIBETA,WYLA, WYLA, PSIVW. 00037

2 GABBW,GABBBW,ABWW,ABBBW,ABBBW,ABBBW, 00038

3 ECGW, ECGE 00039

LOGICAL COPLAN GEOMY 00040

COMMON /GEOM2 / TLAJ,YLAJ,PSIT, YBT, MYBT, MYBT, MYBT,YBTST, MYBTST, GEOMY 00041

1 MYBBY, INBT, IYBT, CAF. 00042

COMMON / NODES/ SYM, SYM, SYM, PEN, NEY MET. 00043

COMMON /DONUTASH/ IPMTW(250), ENSLD(2,600), IPMTW, IPMTOR, IPMTLS 00044

ENSLDB(250), ENSLDB(250), IPMTSD, 00045

COMMON / IPMTSD, ENSLDB(250), IPMTSD(250), ENSLDB(250), ENSLDB(250), ENSLDB(250), ENSLDB(250) 00046

COMMON / COMPLEX / B8149
COMPLEX, TYP DELPHI

DIMENSION DEFSL(2,100)

EQUIVALENCE (DEPSL, DEPSL(2,100))

C ARRAYS DEPSL AND DEFSL ARE EQUIVALENCED TO GIVE A 2 ROW UN-OVERLAPPED SECTION

COMMON /LATOM/LVOM

C

COMPLEX PKERNL(1), SKERNL(1)

LOGICAL WINQ, DWN

C

DIMENSION I3XED(1,50)

LOGICAL LEFT, LSIDE

C

IA = IPROW

BL = 0.0

C

IF (IA > 100) GO TO 20

C

IXM = IXBIT

IXBS = IXBIT

IIP = IXBIT - (IXBIT - IXBS) / NSUBS

MIBBS = MIBBST

IOLPN = IOLAPN

SYMTY = SYMT

GO TO 25

20 CONTINUE

IXB = IXBW

IXBS = 1

MIB = MIBBW

IF (ECOPLAN) MIB = MIBT

MIBBS = MIBBSW

IOLPN = 0

SYMTY = SYMT

GO TO 25

25 CONTINUE

LSIDE = SYMTY .NE. 0.0 AND .NOT. DWN

C

IS SUBDIVISION REQUESTED -

IF (NSUBD > 0.1) GO TO 410

C

YES. DETERMINE THE NUMBER OF POCS WHICH CAN BE HANDLED (IXBI)

C

NSIRW = 1NPTRN - IPHTOP - 1

IF (NSIRW .LT. 0) NSIRW = NSIRW + IPHTLS

NSIRW = MIBD (NSIRW, MSIRW-1)

IF (IA - IXBS .GT. NSIRW) GO TO 120

C

ALL SUBDIVIDED. ALLOW TO GO TO BEYOND TO TRIG RETURN.

IXBI = IA - IXBS > 1

GO TO 200

C

PARTIAL SUBDIVIDED

120 CONTINUE

I = IA - NSIRW - IXS

I = MOD(1, NSUBD)

I = IREC - 1

IXBI = NSIRW - MOD(1, NSUBD)

C

B150
DETERMINE THE CONTRIBUTION TO B FROM A FORWARD CONE OF SUB-  
DIVIDED BOXES  
FORWARD COLUMN OF CONTRIBUTING BOXES  
DIVIDED BOXES  
CONTRIBUTION RELATIVE TO RECEIVING BOX  
RECEIVING BOX  
CONTRIBUTION RELATIVE TO RECEIVING BOX  
IS THE FORWARD EDGE OF THE PATTERN BEEN REACHED -  
NO, GET BOX TYPE CODES FOR CURRENT ROW.  
LEFT = .T.  
IIA = MOD(IA-1,IPMTLS) + 1  
IIAPL = MOD(IA,IPMTLS) + 1  
IPMTPL = IPMTSD(1,IIAPL)  
IF (IPMTPL .LE. 1) GO TO 2G8  
MWB = IPMTPL - IPMTSD(1,IIA) + IPMTSD(2,IIA) - 1  
GO TO 210  
MB = MBBS  
CONTINUE  
CALL DECODER (IBXCD,IBXCL,IA,IA,NW,.T.,IBACD)  
IF (IBXCD .EQ. 0) GO TO 215  
DO 212 I = 1,MBBS  
IF (IBXCD(MB) .NE. 0) GO TO 215  
MB = MB - 1  
GO TO 215  
IBXCD = ROW OF BOX CODES  
MWB = NUMBER FOUND  
GET LOCATION IN THE SUBDIVIDED LOWMASH ARRAY FOR BOX(IA,JCOL)  
IDM = LOCSW(IA,JCOL,IPMTSD,IPNTLS,IDM)  
N = (NM(NM+1))/2 + 1  
KERNEL(NM,MU) = SKERN((NM+MU)*1/2 + ACS(MU) + 1), DO  
lies in SUBSCRIPT FOR KERNEL (STARTING WITH MU,0)  
CENTER BOX OF ROW IN CONE  
IF (JCOL .GT. MB) GO TO 220  
IF (IBXCD(JCOL) .EQ. 0) GO TO 220  
B = B + SKERN(N) - (ENSUBD(1,IDM) - ENSUBD(2,IDM))  
GO TO 220  
GOING OUT FROM CENTER CHORD OF CONE IN BOTH DIRECTIONS  
IDMR,IDML = POINTERS IN LOWMASH ARRAY FOR RIGHT, LEFT SIDES  
IBMR,IBML = POINTERS IN BOX CODES ARRAY, AS ABOVE  
L = LEFT SIDE POINTED INCREMENTER (CHANGES SIGN WHEN  
THE PLANFORM CENTER-LINE IS ENCOUNTERED)  
E = LEFT SIDE MULTIPLIER, USED TO DETERMINE SYMANTI  
B + B + SKERN(N) * (ENSUBD(1,IDM) - ENSUBD(2,IDM))  
IDMR = IDM+1  
IDML = IDM-1  
IBMR = JCOL+1  
IBML = IBMR-2
DO 280 MU = 1,NU
C
RIGHT SIDE
IF (1SM + GT. NWB) GO TO 230
IF (IBCD(IBX)) .EQ. 0) GO TO 230
B = B + SKERNL(N) * (ENSUBD(1,1IDW) - ENSUBD(1,1IDW))
230: CONTINUE
C
LEFT SIDE
IF (.NOT. LEFT) GO TO 270
C
HAS PLATFORM CENTER LINE BEEN ENCOUNTERED -
IF (IBXL .GT. 250)

C
YES. SET PARAMETERS TO SWEEP BACK ACROSS RIGHT HALF AS A
C
SYMMETRIC/ANTISYMMETRIC IMAGE OF THE LEFT SIDE.
C
LEFT = LSIDE
IF (.NOT. LEFT) GO TO 270
E = SYM1Y
L = 1
IBXL = 1
IDWL = IDWL + 1
GO TO 280
B
C IF THE CENTER LINE HAS PREVIOUSLY BEEN ENCOUNTERED, IBXL WILL
C BE INCREASING. IF IBXL HAS EXCEEDED THE NUMBER OF BOXES ON
C THIS ROW, THIS ROW IS COMPLETE, TRANSFER TO LOOP ON NU.
C
250 IF (IBXL .GT. NWB) GO TO 270
260 IF (IBCD(IBXL)) .EQ. 0) GO TO 270
B = B + SKERNL(N) * (ENSUBD(1,1IDW) - ENSUBD(2,1IDW)) * E
270 CONTINUE
C
SET COUNTERS FOR NEXT STEP OUTWARD
IDWR = IDWR + 1
IDWL = IDWL + L
IBWR = IBWR + 1
IBXL = IBXL + L
N = N + 1
280 CONTINUE
C
END OF LOOP ON MU (SUBDIVIDED COLUMNS OUTBOARD)
C
300 CONTINUE
C
END OF LOOP ON NU (SUBDIVIDED ROWS FORWARD) FROM 200
C
C
IS THERE AT LEAST ONE FULL UNSUBDIVIDED ROW LEFT AFRAID OF
C CURRENT POSITION -
C
310 CONTINUE
IF (IA .LT. IXB) GO TO 800
C
C UNSUBDIVIDED BOXES
C
Determine row and column numbers in subdivided arrays corre-
C pending to unsubdivided box centers.
C
IA = ROW LOCATION OF CONTRIBUTING SUBDIVIDED BOX
C
B152
C IIA = ROW LOCATION OF UNSUBDIVIDED BOX
C INU = FIRST ROW OF UNSUBDIVIDED BOXES TO USE, COUNTING
C OUTWARD.
C JJJ = UNSUBDIVIDED CHORD NUMBER OF RECEIVING BOX
C
IA = IA + NSUBD2
IIA = (IA - 1)B/W/NSUBD + 1
INU = WB1 /NSUBD + 1
JJJ = (JCOL-1)/NSUBD + 1
GO TO 420
C
C SET UP POINTERS IF NO SUBDIVISION WAS REQUESTED
C
410 CONTINUE
C 420 DO 500 NU = INU,WB
IA = IA - NSUBD
IIA = IIA - 1
C HAS THE FORWARD EDGE OF THE PATTERN BEEN REACHED?
IF (IA .LT. WB) GO TO 600
C NO. GET BOX TYPE CODES FOR CURRENT ROW, UNSUBDIVIDED BOX
C CENTERS ONLY
LEFT = .T.
IIA,PN = IIA + IOV,PN
NB = IPNDW(1,IIA,PN1) - IPNDW(1,IIA,PN) + IPNDW(2,IIA,PN)
1 = -1
CALL DCDER(IBOX,LBXCD, IIA,1, IIA,NUB, .F., IBXCD)
C IBXCD = ROW OF BOX CODES
C NUB = NUMBER FOUND
C
GET LOCATION IN UNSUBDIVIDED DOWNWASH ARRAY FOR BOX(IIA,JJJ)
IDW = LOCISM(IIA,PN,JJJ, IPNDW, IIA,PN,1,IIA,PN2)
C
N = (NUM(NUM1))/2 + 1
C N = UNSUBDIVIDED KERNEL SUBSCRIPT FOR MU = 0.
C
CENTER BOX
IF (JJJ .GT. NUB) GO TO 425
IF (IBXCD(JJJ) .NE. 0) B = B + PECHL(N)*ENDUS(IDW-EDRLS(IDW))
C
425 CONTINUE
C
GOING OUT FROM CENTER CHORD IN BOTH DIRECTIONS
IDWR = IDW+1
IDWL = IDW-1
IBWR = JJJ+1
IBWL = JJJ-2
L = -1
E = 1.0
B153
M = M + 1

DO 480 MU = 1, NU

RIGHT SIDE
IF (IBXR .GT. NYB) GO TO 450
IF (IBXCD(IBX).EQ. D) GO TO 450
B = B + PERNL(N) * (ENRUS(IDXR) - ENRLS(IDXR))
450 CONTINUE

LEFT SIDE
IF (.NOT. LEFT) GO TO 470
IF (IBML .GT. D) GO TO 450

YES. SET PARAMETERS TO SWEEP BACK ACROSS RIGHT SIDE
LEFT = LSIDE
IF (.NOT. LEFT) GO TO 470
E = SYMTY
L = 1
IBXL = 1
IBDL = IDWL + 1

TEST FOR ROW COMPLETE, AS IN SUBDIVIDED LOGIC
470 IF (IBXL .GT. NYB) GO TO 470
480 IF (IBXCD(IBX) .EQ. D) GO TO 470
B = B + PERNL(N) * (ENRUS(IDWL) - ENRLS(IDWL)) * E
470 CONTINUE

SET COUNTERS FOR NEXT STEP OUTWARD

END OF LOOP ON MU (CHORDS OUTWARD)

END OF LOOP ON MU (ROWS FORWARD) FROM 420

END
SUBROUTINE GETAIC(JUENT, ITPE, ICODE, IR)  
GETAIC 00002
GETAIC 00003
GETAIC 00004
GETAIC 00005
GETAIC 00006
GETAIC 00007
GETAIC 00008
GETAIC 00009
GETAIC 00010
GETAIC 00011
GETAIC 00012
GETAIC 00013
GETAIC 00014
GETAIC 00015
GETAIC 00016
GETAIC 00017
GETAIC 00018
GETAIC 00019
GETAIC 00020
GETAIC 00021
GETAIC 00022
GETAIC 00023
GETAIC 00024
GETAIC 00025
GETAIC 00026
GETAIC 00027
GETAIC 00028
GETAIC 00029
GETAIC 00030
GETAIC 00031
GETAIC 00032
GETAIC 00033
GETAIC 00034
GETAIC 00035
GETAIC 00036
GETAIC 00037
GETAIC 00038
GETAIC 00039
GETAIC 00040
GETAIC 00041
GETAIC 00042
GETAIC 00043
GETAIC 00044
GETAIC 00045

GETAIC 00003
GETAIC 00004
GETAIC 00005
GETAIC 00006
GETAIC 00007
GETAIC 00008
GETAIC 00009
GETAIC 00010
GETAIC 00011
GETAIC 00012
GETAIC 00013
GETAIC 00014

C GETAIC 00003
C GETAIC 00004
C GETAIC 00005
C GETAIC 00006
C GETAIC 00007
C GETAIC 00008
C GETAIC 00009
C GETAIC 00010
C GETAIC 00011
C GETAIC 00012
C GETAIC 00013
C GETAIC 00014

COMMON /FILES / NT5,NT6,INTAPE,INFSP,NPLAIC,NPSAIIC,NOLTP;  
FILES 00002
I IQUFS,MODESIC,TVPSI,ICBESC,INTFSC,IAICSC  
FILES 00003
COMMON /TAPEIO/ NFS,NHL5,NR,ID(20),NRD,ITYPE,LRS,LWS,M,N;  
TAPEIO 00004
1 PARM(10),IR  
TAPEIO 00005
DIMENSION PARM(10)  
TAPEIO 00006
EQUIVALENCE (PARM,PARM)  
TAPEIO 00007
COMMON /IAICS/ YBAR,EL,MAICC2,50),NRO5,SURF,  
MAICS 00008
1 YEARL,ELL,MAIC2(2,50),NROSL,SURFL,PSICIF  
MAICS 00009
LOGICAL SURF,SURFL  
MAICS 00010
COMMON /PAICS / NAAK, NTTK, NRWK, NLMK, PAIC(4,50)  
PAICS 00011
INTEGER PAIC  
PAICS 00012
DIMENSION NK(4)  
PAICS 00013
EQUIVALENCE (NAAK,NK(I))  
PAICS 00014
COMMON /AICS / XRL, (C(1640),W(1640)),Y(1640)  
AICS 00015
COMPLEX C, W, V  
AICS 00016
COMMON /ARRAYS/ KBXCDW,LBXCIDW,LBOC,KXCDT,LKCDT,KBXCDT,LKCDT  
ARRAYS 00017
2 KZPB,FKZPB,LFZPB,LFZPB  
ARRAYS 00018
KZK,FPKZPB,LPKZPB,LPKZPB  
ARRAYS 00019
KZPB,PFKZPB,LFZPB,LFZPB  
ARRAYS 00020
LLOGICAL KWEIT,KWEU  
GETAIC 00021
DATA KWEIT,KWEU /-1,-1/  
GETAIC 00022
DATA IPAP,IPAIC,IPNT, IFAX,IFLAG,ICDEP /640/  
GETAIC 00023
FNTXL 00061
C GETAIC 00024
GETAIC 00025
GETAIC 00026
GETAIC 00027
GETAIC 00028
GETAIC 00029
GETAIC 00030
GETAIC 00031
GETAIC 00032
GETAIC 00033
GETAIC 00034
GETAIC 00035
GETAIC 00036
GETAIC 00037
GETAIC 00038
GETAIC 00039
GETAIC 00040
GETAIC 00041
GETAIC 00042
GETAIC 00043
GETAIC 00044
GETAIC 00045

C GETAIC 00024
GETAIC 00025
GETAIC 00026
GETAIC 00027
GETAIC 00028
GETAIC 00029
GETAIC 00030
GETAIC 00031
GETAIC 00032
GETAIC 00033
GETAIC 00034
GETAIC 00035
GETAIC 00036
GETAIC 00037
GETAIC 00038
GETAIC 00039
GETAIC 00040
GETAIC 00041
GETAIC 00042
GETAIC 00043
GETAIC 00044
GETAIC 00045

IR = 0  
IF (IPNT .NE. 0) GO TO 100  
C GETAIC 00028
GETAIC 00029
GETAIC 00030
GETAIC 00031
GETAIC 00032
GETAIC 00033
GETAIC 00034
GETAIC 00035
GETAIC 00036
GETAIC 00037
GETAIC 00038
GETAIC 00039
GETAIC 00040

C INITIAL CALL. SET UP FILES AND POINTERS  
REMIND TALESC  
GETAIC 00028
GETAIC 00029
GETAIC 00030
GETAIC 00031
GETAIC 00032
GETAIC 00033
GETAIC 00034
GETAIC 00035
GETAIC 00036
GETAIC 00037
GETAIC 00038
GETAIC 00039
GETAIC 00040
GETAIC 00041
GETAIC 00042
GETAIC 00043
GETAIC 00044
GETAIC 00045

C REMIND TALESC  
GETAIC 00028
GETAIC 00029
GETAIC 00030
GETAIC 00031
GETAIC 00032
GETAIC 00033
GETAIC 00034
GETAIC 00035
GETAIC 00036
GETAIC 00037
GETAIC 00038
GETAIC 00039
GETAIC 00040
GETAIC 00041
GETAIC 00042
GETAIC 00043
GETAIC 00044
GETAIC 00045

C EXPAND PAIC ARRAY  
GETAIC 00031
GETAIC 00032
GETAIC 00033
GETAIC 00034
GETAIC 00035
GETAIC 00036
GETAIC 00037
GETAIC 00038
GETAIC 00039
GETAIC 00040
GETAIC 00041
GETAIC 00042
GETAIC 00043
GETAIC 00044
GETAIC 00045

100 CONTINUE  
GETAIC 00031
GETAIC 00032
GETAIC 00033
GETAIC 00034
GETAIC 00035
GETAIC 00036
GETAIC 00037
GETAIC 00038
GETAIC 00039
GETAIC 00040
GETAIC 00041
GETAIC 00042
GETAIC 00043
GETAIC 00044
GETAIC 00045

110 CONTINUE  
GETAIC 00041
GETAIC 00042
GETAIC 00043
GETAIC 00044
GETAIC 00045

120 CONTINUE  
GETAIC 00041
GETAIC 00042
GETAIC 00043
GETAIC 00044
GETAIC 00045

B155
DO 129 J = 1, NMIX
   PAIC(I,J) = PAIC(I,J)
   PAIC(I,J) = 0
129 CONTINUE
   I = I + 1
130 CONTINUE
   IF (NTK .EQ. 0) GO TO 140
   IF (I .EQ. 2) GO TO 140
   DO 135 J = 1, NTK
      PAIC(I,J) = PAIC(I,J)
      PAIC(I,J) = 0
135 CONTINUE
140 CONTINUE
   C ZERO OUT THE AIC ARRAYS
   DO 150 I = 1, LKernel
      C(I) = (0., 0., 0.)
      W(I) = (0., 0., 0.)
150 CONTINUE
   C GET THE AIC LOCATION
160 CONTINUE
   IF (INC(NYPE) .LT. JUCENT) GO TO 290
   ILOC = PAIC(NYPE, JUCENT)
   C ARE THE DESIRED ARRAYS ALREADY IN CORE -
   IF (ILOC .EQ. IPAIC) GO TO 300
   IPAIC = ILOC
   ILOC = (ILOC - 1) * 4 + 1
   C GET THE MUSIC ARRAY FROM THE NON-PLANAR AIC SCRATCH FILE
   C SPACE AND READ MUSIC
   CALL REDIT
   IF (ILOC - IPHT) 200, 220, 210
   MUSIC ARE BEHIND CURRENT LOCATION
200 REWIND IATCSC
   NMU = ILOC - 1
   GO TO 220
   C REQUIRED MUSIC ARE AHEAD OF CURRENT POSITION
210 CONTINUE
   NMU = ILOC - IPHT
   C READ MUSIC FROM IATCSC
220 CONTINUE
   NMU = 94
   MUCI
   K = 2
   CALL READMUS(IATCSC, NWRITE, RANDOU, NPS, NMU, LB, NMU, X, NID, ID,
   IYPE, LB4, MUSIC, M, N, PARM, IRR)
   IF (I8R .NE. 0) GO TO 3000
   NMUNS = N
   IPHT = ILOC + 1
   EL = PARM(5)
   YBAR = PARM(4)
   IDC = IPARM(6)
   CALL REDIT
C IS THE C ARRAY DESIRED?
IF (ICCE .NE. 0) GO TO 250
C YES, IS IT AVAILABLE?
IF (ICD .EQ. 0) GO TO 240
C NO, SET THE ERROR FLAG AND CONTINUE
IR = 1
GO TO 250
C READ THE C ARRAY FROM IAICSC
240 CONTINUE
WMARRY = 9HSpatial C
CALL READMX( IAICSC, WWRITE, RANDOMU, NFS, NMS, LS, NMR, K, NID, ID,
1 ITYPE, LRS, W, M, N, PARM, IRR)
IF (IRR .NE. 0) GO TO 3000
CALL RDINIT
GO TO 260
250 CONTINUE
NMS = 1
260 CONTINUE
IPNT = IPNT + 1
C READ THE W ARRAY FROM IAICSC
WMARRY = 9HSpatial W
CALL READMX( IAICSC, WWRITE, RANDOMU, NFS, NMS, LS, NMR, K, NID, ID,
1 ITYPE, LRS, W, M, N, PARM, IRR)
IF (IRR .NE. 0) GO TO 3000
IPNT = IPNT + 1
CALL RDINIT
C IS THE V ARRAY DESIRED?
IF (ICCE .EQ. 2) GO TO 300
C YES, IS IT AVAILABLE?
IF (ICD .NE. 2) GO TO 280
C NO, SET ERROR FLAG
IR = IR + 1
GO TO 300
C READ THE V ARRAY FROM IAICSC
280 CONTINUE
WMARRY = 9HSpatial V
CALL READMX( IAICSC, WWRITE, RANDOMU, NFS, NMS, LS, NMR, K, NID, ID,
1 ITYPE, LRS, V, M, N, PARM, IRR)
IF (IRR .NE. 0) GO TO 3000
IPNT = IPNT + 1
GO TO 300
C NO AICS CAN BE FOUND OF THE TYPE DESIRED FOR THIS CHORE
290 IR = 3
C 300 CONTINUE
RETURN
C DIAGNOSTIC
C 3100 CONTINUE
WRITE (418,90000) IAICSC, IRR
WRITE (418,9192) WMARRY, W
CALL FLUSH(1)
9000 FORMAT(20H*** ERROR - WHILE READING FROM SPATIAL AIC FILE .AIC, 34H*** )
1 14H, ERROR CODE = 14, 4H *** )
9192 FORMAT(14X,A10, 20HARRAY, DIMENSIONED (14,1H,14,11N) WAS BEING
SUBROUTINE STOCSW(IROW,JCOL,EN,IBOX,LIBXCD,IXB,IXBS,MYB,IRR)  
STOCSW 00002

STORES A COMPUTED DOWNWASH VALUE IN THE END-AROUND SUBDIVIDED  
DOWNSWASH ARRAY, AND UPDATES POINTERS WHEN NEEDED  
STOCSW 00003

IROW = BOX CHORIWISE LOCATION  
STOCSW 00004
JCOL = BOX SPANWISE LOCATION  
STOCSW 00005
EN = COMPLEX NORMAL-WASHES TO BY STORED  
STOCSW 00006
IBOX = ARRAY OF BOX CODES  
STOCSW 00007
LIBXCD = LENGTH OF BOX CODE ARRAY  
STOCSW 00008
IXB = FIRST SUBDIVIDED ROW OF THE PLANFORM  
STOCSW 00009
IXBS = MAXIMUM CHORD LENGTH OF SUBDIVIDED PATTERN  
STOCSW 00010
MYB = MAXIMUM ROW LENGTH  
STOCSW 00011

RETURNS -  
STOCSW 00012
IRR = ERROR RETURN, C = SUCCESSFUL  
STOCSW 00013
= 1, FUNCTION LOCDSW FOUND THE POINTER OUTSIDE  
STOCSW 00014
THE DEFINED SET OF DOWNWASHES  
STOCSW 00015
ENSUBD = SUBDIVIDED NORMAL-WASH ARRAY WITH ADDED VALUE  
STOCSW 00016

COMMON PARAMETERS USED  
STOCSW 00017
KSIYRN = MAXIMUM SIZE OF THE SUBDIVIDED KERNEL  
STOCSW 00018
LSDW = DIMENSION OF SUBDIVIDED NORMAL-WASH ARRAY  
STOCSW 00019
IPNTSD = POINTER ARRAY FOR SUBDIVIDED NORMAL-WASH ARRAY  
STOCSW 00020
IPNTIN = NEXT AVAILABLE CELL IN IPNTSD  
STOCSW 00021
IPNTOR = FIRST CURRENTLY VALID CELL IN IPNTSD  
STOCSW 00022
IPNTLS = DIMENSION OF IPNTSD  
STOCSW 00023

COMMON / KERN / ERR,WSKRN,IPKERN,RFKERN,NEPATK,NGPNEA  
STOCSW 00024
COMMON / ARRAYS/ KBXCDM,LIBXCDM,LIBXCD,KBXCD,LIBXCD,JKPH,JKPH,  
STOCSW 00025
KALPHA,JKERN,IKERN,IPNTMH,IPNTRM,RDEFS,IKAPHI,  
STOCSW 00026
UMODES,IPNTSD,IPNTSD,KSDW,LSDW,IPNTDW,LPNTDW,  
STOCSW 00027
KROW,LSOW,KTVWP,LTVWP  
STOCSW 00028

COMMON / SWASH/ IPNTSD(250), ENSUBD(12,800), IPNTIN,IPNTOR,IPNTLS  
STOCSW 00029
SMWASH 00030

COMMON / CHECKPR/ DPPCR,GEOPR,NCDCPR,ALCPR,ASCPR,GPFCPR  
STOCSW 00031
CHECKPR 00032

LOGICAL DPPCR,GEOPR,NCDCPR,ALCPR,ASCPR,GPFCPR  
STOCSW 00033

EQUALVALUE(EQUALSUB)  
STOCSW 00034

LOGICAL CHECKPR  
STOCSW 00035

DIMENSION TITL(3)  
STOCSW 00036

COMPLEX EN(2)  
STOCSW 00037

IRR = 0  
STOCSW 00038

IS THIS THE INITIAL CALL -  
STOCSW 00039

IF (IROW.EQ. IXB .AND. JCOL .EQ. 1) GO TO 700  
STOCSW 00040

C NO. IS A NEW ROW BEING CONSIDERED -  
STOCSW 00041

IF (IROW.GT. IROW) GO TO 200  
STOCSW 00042

C NO. GET THE LOCATION FOR THE VALUE IN THE SUBDIVIDED DOWN-  
STOCSW 00043
WASH ARRAY  
STOCSW 00044

IJ = LOCDSW(IROW,JCOL,IPNTSD,IPNTIN,IPNTOR,IPNTLS)  
STOCSW 00045

IF (IJ) 900,900,550  
STOCSW 00046

C MUST UPDATE POINTERS AND ADD A ROW TO THE SUBDIVIDED BOX ARRAY  
STOCSW 00047

200 CONTINUE  
STOCSW 00048

IROW = IROW  
STOCSW 00049

B159
SET THE NEXT VALUES OF THE POINTER ARRAY (1 ROW)

INCREMENT IPTOT, ALLOWING FOR END-AROUND INCREMENTAL

1 = IPTIM IN

II = IPTSD(I, IPTIN)

CALL POINTR (ROW, 1, M, B, . . . , BOX, LBD, IPNTLS, 1

LOOP INCREMENTING IPTOT, IF OVER-LAP OCCURS.

IF (IPNTIN - IPTOT) > 200, 225, 235

CONTINUE

IF (IPNTIN - IPTOT) < 1

CONTINUE

CALL PRINTR (TITL, 0, ENSUD, 2, IPTOT, IPTIN - 1, MB, IPTSD)

CALL PRINTR (TITL, 0, ENSUD, 2, IPTOT, IPTNL - 1, MB, IPTSD)

CONTINUE

IF (IPNTSD(I, IPTIN)) = MOD(IPNTOT + ISROG - MASKRN - 1)

CONTINUE

KEEP SUBDIVIDED DOWNSHAPES END-AROUND.

HAS THE ARRAY LIMIT BEEN EXCEEDED -

ILD = IPTSD(I, IPTIN)

IF (IPNTSD(I, IPTIN) , IE, LSTW + 1) GO TO 405

LIMIT EXCEEDED BY CURRENT ROW. PLACE AT BEGINNING OF THE ARRAY

IF (.NOT. CHECK) GO TO 400

TITL(1) = IOKEY SUBDIV

TITL(2) = IOKEY, UPPER

TITL(3) = IOH, PARTIAL

IF (IP0NIN, LT, IPTOT) GO TO 395

CALL PRINTR (TITL, 0, ENSUD, 2, IPTOT, IPTIN - 1, MB, IPTSD)

GO TO 400

CALL PRINTR (TITL, 0, ENSUD, 2, IPTOT, IPTNL - 1, MB, IPTSD)

CONTINUE

IF (IPNTSD(I, IPTIN)) = IPTSD(I, IPTIN) - ILD + 1

IPNSTD(I, IPTIN) = IPTSD(I, IPTIN) - ILD + 1

CONTINUE

IF (ILD, GT, IP0) GO TO 330

ARRAY WAS ALREADY END-AROUND PRIOR TO LATEST ADDITION

IF (ILD, EQ, IP0) GO TO 340

ADDED ROW WENT END-AROUND AS WELL

IF (IPNTOT = MOD(IPNTOT, IPNTLS) + 1

IF (IPNTOT, NE, 1) GO TO 410

IPNTOT = MOD(IPNTOT, IPNTLS) + 1

IF (IP0 = IPNTSD(I, IPTOT)

GO TO 440

ARRAY WAS SEQUENTIAL. CHECK WHETHER IT HAS GONE END-AROUND

CALL FRENTR(TITL, 0, ENSUD, 2, IPTOT, IPTIN - 1, MB, IPTSD)

CONTINUE

IF (ILD, EQ, IP0) GO TO 310

IT HAS GONE END-AROUND
C 440 CONTINUE
C HAS THE ARRAY BEEN OVERRITTEN IN GOING END-AROUND?
IF (IPO .GT. IP) GO TO 500
C YES, MOVE IPNTOT UNTIL CLEAR.
IPNTOT = MOD(IPNTOT,IPNTLS) + 1
IPO = IPNTSD(I IPNTOT)
IF (IPO .NE. 1) GO TO 440
C ALL POINTERS HAVE BEEN RESET. GET LOCATION
900 CONTINUE
IJ = LOCSDW(IROW,ICOL,IPNTSD,IPNTIN,IPNTOT,IPNTLS)
IF (IJ .EQ. 0) GO TO 900
C STORE THE DOWNHILL VALUE
550 CONTINUE
ENSUBD(1, IJ) = EN(1)
ENSUBD(2, IJ) = EN(2)
600 RETURN
C
C INITIAL CALL
700 CONTINUE
IPNTOR = MOD(IB-1,IPNTLS) + 1
IPNTIN = IPNTOR
MX = MIND(MW,B-1, BSKRN, IPNTLS-1)
C SET UP POINTER ARRAY FOR FIRST PASS
CALL PONTR(IB, MX, MY, .., .., IBX0L, IBXCD, IPNTLS, 1, 1, IPNTIN, IPNTSD)
720 CONTINUE
IF (IPNTSD(1, IPNTIN) .LE. LSDW ) GO TO 730
IPNTIN = IPNTIN - 1
MX = MX - 1
GO TO 720
730 CONTINUE
IROW = MX + IB - 1
C
ENSUBD(1, 1) = EN(1)
ENSUBD(2, 1) = EN(2)
GO TO 600
C
C ERROR
900 IRR = 1
GO TO 600
C
END
SUBROUTINE PRINT ( TITL, IMODE, ARRAY, K, IXB, MXB, MYB, IPNTR )
C
C  TITL - TITLE TO PRINT FOR THE ARRAY
C  IMODE - MODE SHAPE NUMBER
C  ARRAY - ARRAY TO BE PRINTED

DIMENSION ARRAY(K,1), TITL(3)
COMPLEX ARRAY
DIMENSION IPNTR(2, 50)
COMMON /CTRL/ PRVECH, TITLE(8), PRVECH, PRMODE, DHHW, DHT, DEFAULT
I   DEFAULT
LOGICAL PRVECH, PRMODE, DHHW, DHT, DEFAULT
COMMON /PROCLAIM/ XMACH, NMODES, NTSLOP, NXVAL, SMOOTH, MDEG, GRDFIT,
I   EXAIC, SUBB, PLYWOOD
LOGICAL SMOOTH, GRDFIT, EXAIC, SUBB, PLYWOOD
COMMON /FILES/ NT5, NT6, INTAPE, INFSP, NPLAC, NPLAC, NOUTP,
I   IOUSP, MDESC, TFSP, ITDESC, TFSP, IAICSC
COMMON /KVAL/ IKVAL, XKVAL(20), XS(50)
DIMENSION PC(2)
DIMENSION S(50), D(50)
EQUIVALENCE (S(1), BUFF(1)), (D(1), BUFF(1251))
REAL K1

INTEGER PAGE
COMMON /RDBUFF/ RFCCOE, IBFCONT, BUFFS0
DATA PC / 100PAGE CONTI, RPMUED /
DATA BLANK / 1H /
DATA XINIT / -1.0 /
K1 = XKVAL(IKVAL)
IF (XS(KVAL).NE.XINIT) K1 = XS(KVAL)

C

PAGE = 0
M = 1
M = 4
IF (M.GT.MYB) M = MYB
100 LINE = 100
200 DO 1400 J = 1, IXB, MXB
      DO 300 JN = M
      S(J) = 0.0
      D(J) = 0.0
300 CONTINUE
      IF (LINE.LE.50) GO TO 900
      PAGE = PAGE + 1
      LINE = 4
      WRITE (NT6,9001) TITLE, TITL, XMACH, K1, IMODE

C
      IF (PAGE.EQ.1) GO TO 700
      WRITE (NT6,9005) PC
      GO TO 800
700 WRITE(NT6,9005)
800 CONTINUE
      WRITE(NT6,6006) (BLANK, J; J = N, M)
      WRITE(NT6,6007) (BLANK, J; J = N, M)
900 CONTINUE
      J = 1, IPNTR(2, 1)
      IF (J .LE. 0) GO TO 1400
      I = 1, IPNTR(1, 1)

B162
JE = 1; PRINTM(I,1+1), IDX + JS -1
IF(JE.EQ.0) GO TO 1400
DO 1000 J=JS,JE
  S(J) = REAL(ARRAY(I,IDX))
  D(J) = AIMAG(ARRAY(I,IDX))
  IDX = IDX + 1
1000 CONTINUE
DO 1200 J=N,M
  IF((S(J)) .GT. 1000, 1100, 1300
1100 CONTINUE
  IF((D(J)) .GT. 1000, 1200, 1300
1200 CONTINUE
  GO TO 1400
1300 WRITE (INT6,9013) I, (S(J), D(J), J=N,M)
  LINE = LINE + 1
1400 CONTINUE
  M = M+4
  N = N+4
  IF(N .GT. WB) GO TO 1500
  IF(M .GT. WB) M=WB
  IF(LINE .GT. 45) GO TO 100
  WRITE(INT6,906)  (BLANK, J=N,M)
  WRITE(76,907)  (BLANK, J=N,M)
  LINE = LINE+3
  GO TO 200
1920 CONTINUE
RETURN
9001 FORMAT(1H1,20X,8A10/ 50X,3A10/ 46X, 7H( HACH F5.3,5X,10FRED. FREQ. PRTNM 00081
  1 *=* F8.5, * *) /52X,4MORE SHAPE, I3)
9005 FORMAT(44X,42(1H-),20X,4A10,4A)
9206 FORMAT(4HORGN, A1,14X, SHCHRD, I3, 3(A1,22X, SHCHRD, I3))
9207 FORMAT(3X, 4(A1,9X, #FREAL, 8X, 9H(IMGINARY) )
9013 FORMAT(4,8E15.8)
END
SUBROUTINE DCODED(JBOX,LBOX, IA, JA, IL, JL, SUBD, ICODE)

DIMENSION IBOX(LBOX,1), ICODE(1)

C     IBOX - ARRAY OF BOX CODES IN PACKED WORD FORMAT
C     LBOX - ROW DIMENSION OF BOX CODES ARRAY
C     IA - J-TH INDEX OF FIRST CODE TO RETRIEVE
C     JA - J-TH INDEX OF FIRST CODE TO RETRIEVE
C     IL - LAST BOX CODE ON THE JA-TH CHORD TO RETRIEVE
C     JL - LAST BOX ON THE IA-TH ROW TO RETRIEVE
C     SUBD - SUBDIVIDED BOX CODES DESIRED, If. UNSUBDIVIDED
C     ICODE - ARRAY INTO WHICH BOX CODE WILL BE STORED.

COMMENT ON USAGE

BOX CODES CAN BE RETRIEVED FOR ONE BOX, A ROW OR PART OF
A ROW, OR A COLUMN OR PART OF A COLUMN. A ROW OR COLUMN CAN
NOT BE RETRIEVED AT THE SAME TIME. IF ONLY 1 BOX IS DESIRED
SET IL = IA AND JL = JA. IF BOTH IL NE. IA AND JL NE. JA,
ONE ROW WILL BE RETURNED, IL BEING IGNORED.

COMMON /GEO/ COPLAN, NSUBD, XSUBD, NSUBD, NSUBC, NSURF.
1      B1, BIBET, B1S, BIBTE, XMAX, XMIN, PSW,
2      XBBX, XBBW, MBBW, MYBBW, MXBBW, MYBBX, MBBX,
3      XBBW, XCENTR

LOGICAL COPLAN
LOGICAL SUBD
INTEGER SHIFT
DATA NEWRD /20/
MASK = 7
IB = 1

IF (SUBD) GO TO 50
I = NSUBD * (JA-1) + XBBW
J = NSUBD * (IA-1) + NSURC
ISKIP = NSUBD
IEND = NSUBD + (IL-1) + XBBW
JEND = NSUBD + (JL-1) + NSURC
GO TO 60

50 CONTINUE
I = IA
J = JA
ISKIP = 1
IEND = IL
JEND = JL

60 CONTINUE
IF (JL .EQ. JA) GO TO 1100

C     PROGRAM WILL RETRIEVE N BOXES FROM ROW I

100 CONTINUE
DO 1000 JJ = J, JEND, ISKIP
JB = (JJ-1) / NEWRD + 1
IWORD = IBOX(I, JB)
JR = (NEWRD - MOD(JJ-1, NEWRD)) + 3
IF (JR .EQ. 0) JB = 0
C     J = NUMBER OF BITS TO SHIFT LEFT.
IJMASK = SHIFT(MASK, JB)
ICODE = IJWORD.AND.IJMASK
JW8 = 1 + JB
ICODE(B) = SHIFT(ICODE, JW8)

B164
IB = IB + 1
1000 CONTINUE
  GO TO 3000
C
C       PROGRAM WILL RETRIEVE NJ BOXES FROM CHORD J
1100 CONTINUE
  JSB = (J-1)/NBWRD + 1
  JB = (NBWRD - MOD(J,NBWRD ) ) * 3
  IF(JB.EQ.0) JB = 0
  IJMASK = SHIFT(MASK,JB)
  NWB = -JB
  D* 2000 II = I,END,ISKIP
  IJWORD = IBOX(II,JSB)
  IJCODE = IJWORD.AND.IJMASK
  ICODE(IJ) = SHIFT(IJCODE,NWB)
  IB = IB + 1
2000 CONTINUE
C
3000 CONTINUE
RETURN
END
SUBROUTINE POINTER( IX, MX, MYB, SUBD, DIAPH, IBOX, LBXCD, IPNTR )

C Generates a pointer array which serves as an index to a condensed array of box values (modes, downshakes, etc.)
C
C IX = Center of first box to use
C MX = Number of rows to process
C MYB = Maximum row length
C SUBD = ..., subdivided boxes to be used
C     = ..., subdivided boxes to be used
C DIAPH = ..., diaphragms to be included
C     = ..., only planform boxes
C IBOX = Array of box codes
C LBXCD = Row dimension of box codes array
C MXIR = Size of IPNTR array
C IPNT1 = Value to be used for first pointer (normally 1)
C IOUT -
C IPNTIN = Location of next available location in IPNTR array
C (Overlap of tail is accounted for here.)
C
C Outputs -
C IPNTR = Pointer array - output from the routine
C IPNTR(1,1) = Location of the first value for row I
C IPNTR(2,1) = Chord location (subscript J) of that value
C
C Common values used -
C
DIMENSION IBOX(LBXCD,1), IPNTR(2,1), ICODE(150)
LOGICAL DIAPH, SUBD, WING

C ITOFBX = IPOINT
IL = IX + MX - 1
DO 100 IROW = IX, IL
IPBSUG = 0
ILUSUG = 0
IPBX = 1
IBXSUM = 0
CALL DCODE(IBOX, LBXCD, IROW, 1, IROW, MYB, SUBD, ICODE)

C Find last box on row
W = MYB
IF (DIAPH) GO TO 20
C Only planform boxes desired
DO 15 JCOL = 1, MYB
IF (ICODE(JCOL) .EQ. 1) GO TO 30
W = W - 1
15 CONTINUE
GO TO 92
C Planform and diaphragm desired
20 CONTINUE
DO 21 JCOL = 1, MYB
IF (ICODE(JCOL) .NE. 0) GO TO 30
W = W - 1
21 CONTINUE
GO TO 92
C
C   LOOP ON CHORDS IN THE ROW
30 CONTINUE
   DO 90 JCOL = 1,M
   IF (ICODE(JCOL) .EQ. 1) GO TO 35,40,50
C   ICODE = 0
C
   35 IF (IDISUM .NE. 0) GO TO 40
      IPBX = IPBX + 1
      GO TO 90
C   ICODE = 1
C
   40 IPBSUM = IDISUM + 1
C   ICODE = 2 OR 3
90 CONTINUE
   IDISUM = IDISUM + 1
90 CONTINUE
C
92 CONTINUE
   IPNR(1,IPNR) = ITOTBX
   IPNR(2,IPNR) = IPBX
   IPNR = MOD(IPNR,MXR) + 1
   IF (DIAG) GO TO 95
      ITOTBX = ITOTBX + IPBSUM
      GO TO 100
95 ITOTBX = ITOTBX + IDISUM
100 CONTINUE
   IPNR(1,IPNR) = ITOTBX
   IPNR(2,IPNR) = 0
   RETURN
END
FUNCTION LOCSDW(IRW, JCCL, IPNTSD, IPNTIN, IPNTOT, IPNTLS)

C RETURNS THE LOCATION OF THE WORD IN THE END-AROUND SUBDIVIDED
C DOWNWASH ARRAY CORRESPONDING TO BOX(IRW,JCCL) OF THE SUB-
C DIVIDED BOX ARRAY
C
C IRW = BOX CHORDWISE LOCATION
C JCCL = BOX SPANWISE LOCATION
C IPNTSD = ARRAY OF POINTERS
C IPNTIN = NEXT AVAILABLE (UNUSED) CELL IN IPNTSD (END-
C AROUND)
C IPNTOT = FIRST CURRENTLY AVAILABLE CELL IN IPNTSD
C IPNTLS = LAST CELL OF IPNTSD (LENGTH OF ARRAY)
C
C RETURN -
C LOCSDW = LOCATION OF DESIRED DOWNWASH, IF SUCCESSFUL
C = 0, IF LOCPTN LIES OUTSIDE THE DEFINED AREA.
C
C DIMENSION IPNTSD(2,IPNTLS)
C
C LOCPTN = MOD(IRW-1,IPNTLS) + 1
C LOCPTN = LOCATION OF CELL IN IPNTSD WHICH WAS OR IS TO BE
C USED
C IF(IPNTIN-IPNTOT) 100, 300, 200
C END AROUND HAS OCCURRED
C 100 IF (LOCPTN-IPNTIN) 400, 300, 150
C NOT IN UPPER PART, IS LOCPTN WITHIN BOTTOM PART -
C 150 IF (LOCPTN-IPNTOT) 300, 400, 400
C
C NO END AROUND, NORMAL SEQUENCE
C 200 IF (LOCPTN-IPNTIN) 250, 300, 300
C LESS THAN UPPER LIMIT, IS LOCPTN .GE. LOWER LIMIT -
C 250 IF (LOCPTN .GE. IPNTOT) GO TO 400
C
C ERROR OR INITIAL CONDITION ENCOUNTERED (SHOULD NEVER OCCUR)
C 300 LOCSDW = 0
C GO TO 500
C
C SUCCESSFUL, BOX HAS BEEN DEFINED
C 400 IFB = IPNTSD(2,LOCPTN)
C IF(JCCL.LT.IFB) GO TO 300
C LOCSDW = IPNTSD(1,LOCPTN) + JCCL-IFB
C
C 500 CONTINUE
C RETURN
C END
SUBROUTINE SMPLW( IBOK, LBXCD, JOHID, JT, IFRST, ILAST)

COMPUTES DOWNWASH, SIDEWASH AND VELOCITY POTENTIAL FOR A
SAMPLE CHORD LOCATED IN THE WING FLOW FIELD

IBOK = ARRAY OF BOX CODES FOR THE WING
LBXCD = LENGTH OF BOX CODE ARRAY
JOHID = SAMPLE-MATCH CHORD NUMBER
JT = J-LOCATION OF THE CHORD
IFIRST = NUMBER OF FIRST SAMPLE BOX
ILAST = NUMBER OF LAST SAMPLE BOX

DIMENSION IBOK(LBXCD,1)
COMMON /CTRL/ FREQM, CHACH, TITLE(8), FREQM, FRVACD, DIAH, DINT, CTRL
1 DEFAULT
LOGICAL FREQM, FRVACD, DIAH, DINT, DEFAULT
COMMON /PROBLV/ XMAC, NMEDS, TSLOF, RVALS, SMOOTH, NECD, GRDFIT
1 EXAC, SUBD, FLYWOOD
LOGICAL SMOOTH, GRDFIT, EXAC, SUBD, FLYWOOD
COMMON /SNASH/ ITRANS(2,50), ENSUBD(2,60), ITRANS, ITRANS, ITRANS
COMMON /SNASH/ ITRANS(2,50), ENSUBD(2,60), ITRANS, ITRANS, ITRANS

COMPLEX SUBD
COMMON /MULACS/ YBAR, EL, MULACS(2, 50), NCSW, SURF,
1 YBAR, ELL, MULACS(2, 50), NCSW, SURF, PSI
LOGICAL SURF, SURF
COMMON /GEOM/ CPLAN, NSUBD, NSUBD, NSUBD, NSUBC, NSURF,
1 B1, B1, B1, B1, B1, B1, B1, B1, B1, B1, B1, B1, B1,
2 MODW, MWTW, MYFW, MYCCS, MYCSW, MYBGS, MYBGS,
3 XEQW, XQCTR
LOGICAL CPLAN
COMMON /FILES/ MTS, MTS, INTAPE, INFSP, MULACS, NSPAC, NSPAC,
1 IQFSP, MQS, MQS, MQS, MQS, MQS, MQS,
LOGICAL MQS, MQS, MQS, MQS, MQS, MQS, MQS
COMMON /IOCNT/ OPLAC, OPLAC, OPLAC, OPLAC, OPLAC, OPLAC,
1 FRAIC, FRAIC, FRAIC, FRAIC, FRAIC, FRAIC,
2 FRAIC, FRAIC, FRAIC, FRAIC, FRAIC, FRAIC
COMMON /MODS/ SYM, SYM, SYM, SYM, SYM, SYM

COMMON /NAMES/ ITRANS(2,100), ENRUS(1275), ENRUS(1275), IOCLAP

COMPLEX ENRUS, ENRUS
COMMON /AICS/ WX1, C(1640), W(1640), V(1640)
COMPLEX C, W, V

COMMON /DCOL(52), SW(50), LW(50), PHI(50)
COMMON /EQUIVALENCE (SW, ITRANS), (LW, ENSUBD), (IW, ENSUBD(1,26))
COMMON CPUs, CPU, CPU, CPU, CPU, CPU

DATA RVOL, LVOL / 3, 4 /

C
C SET CONSTANTS
CPRI  = COS(PSI)
PSI  = ATN(PSI)
BIMN = 1. / B1

C COMPUTE THE RIGHT WING CONTRIBUTION TO THE SAMPLE CHORD
GET THE NECESSARY AIC ARRAYS

CALL ETAIC(JORD,RTY,0,IR)

IF (IR .NE. 0) GO TO 800

YMUBAR = (JT-.5 + EL4PSI)/CPSI
JBAR = YMUBAR

IF (YMUBAF .GE. 0) JBAR = JBAR + 1

NUMBIN = ABS(EL) + .5

IBX = 1

LOOP ON BOXES ALONG THE SAMPLE CHORD

DO 200 IBX = IP1ST,ILAST

NUMAX = IBX - 1

I = IBX - NUMIN

ZERO OUT THE SUMMATION VARIABLES

WSUM = (.0,.0)
WSUM = (.0,.0)
NSUM = (.0,.0)

IF (YBAR) 120,125,130

120 JINCR = -1

GO TO 135

125 IAIC = NUMBIN+2

INCAIC = 2*NUMIN + 1

JINCR = 1

GO TO 140

130 JINCR = 1

135 IAIC = NUMBIN+2 + NUMIN

INCAIC = 2*NUMIN + 2

140 CONTINUE

LOOP FORWARD OVER THE RIGHT WING

DO 190 NUMAR = NUMIN,NUMAX

MUAI C1 = MUAI C(I,NUMAR+1)

MUAI C2 = MUAI C(2,NUMAR+1)

IF (MUAI C2 ,GE. 0) GO TO 185

IF (YBAR .GE. 0) GO TO 150

JCLR = JBAR + NUMAR - MUAI C2 + 1

GO TO 160

150 JCLR = JBAR - NUMAR + MUAI C1 - 1

160 CONTINUE

LOOP ON A ROW OF WING BOXES, COMPUTING RIGHT WING CONTRIBUTION

DO 180 MUAI = MUAI C1,MUAI C2

IF (JCLR ,LE. 0) GO TO 170

CALL DEODEH(IBOX,LBXCD,1,JCUR,1,JCLR,F4,IICD)

IF (IICD ,GE. 0) GO TO 170

A CONTRIBUTING BOX HAS BEEN FOUND. GET THE AIC LOCATION

KAIC = IAIC + MUAI

GET THE NORMAL-WASH LOCATION

IDS = LOCSDH1,JCUR,LPNTDW,LPNTDO,1,LPNTDO

ADD THIS CONTRIBUTION TO THE SUMS

EN = ENRUS(IJS)

GO TO 165

165 EN = ENRUS(IJS)

8170
165 CONTINUE  
WSUM = WSUM + W(KAIC) * EN  
VSUM = VSUM + V(KAIC) * EN  
PHISUM = PHISUM + C(KAIC) * EN  
170 CONTINUE  
JCOLR = JCOLR + JINC  
180 CONTINUE  
END OF LOOP FOR RIGHT WING ROW CONTRIBUTIONS  

181 CONTINUE  
I = I - 1  
IF (I .LE. 0) GO TO 195  
IAIC = IAIC + INCAIC  
INCAIC = INCAIC + 2  
190 CONTINUE  
END OF LOOP FORWARD ON RIGHT WING ROWS, FROM 1400  

193 CONTINUE  
DW(IBX) = BINV * (CPSI+VSUM + PSI+WSUM)  
SW(IBX) = BINV * (CPSI+VSUM - PSI+WSUM)  
PHI(IBX) = PHISUM  
IBX = IBX + 1  
200 CONTINUE  
END OF LOOP ON RECEIVING BOXES, FOR RIGHT WING CONTRIBUTIONS  

C  
NBX = IBX - 1  
C  
IS LEFT WING CONTRIBUTION NEEDED -  
IF (SYM .EQ. 0) GO TO 310  
C  
YES, GET THE AIC ARRAYS FOR LEFT WING CONTRIBUTIONS.  
CALL RETAIIC(JORD, LWT, O, IR)  
IF (IR .NE. 0) GO TO 600  
C  
NUBIN = ABS(EL) + .5  
IBX = 1  
YMBAR = (-JT+.5 + EL*PSI) / CPSI  
JBAR = YMBAR  
IF (YMBAR .GE. 0) JBAR = JBAR + 1  
C  
LOOP ON BOXES ALONG THE SAME CHORD  
DO 300 IBXX = IPRT,1-LAST  
NUMAX = IBXX - 1  
I = IBXX - NUBIN  
C  
ZERO OUT THE SUMMATION VARIABLES  
WSUM = 0.0,0.0  
VSUM = 0.0,0.0  
PHISUM = 0.0,0.0  
C  
IF (YBAR) 220, 225, 230  
220 JINC = 1  
GO TO 235  
225 IAIC = NUBIN+2  
INCAIC = 2*NUBIN + 1  
JINC = -1  
GO TO 240  
230 JINC = -1  
235 IAIC = NUBIN+2 + NUBIN  
INCAIC = 2*NUBIN + 2  
240 CONTINUE  
C  
B171
C LOOP FORWARD OVER THE LEFT WING
DO 290 NUBAR = NUBMIN, NUBMAX
MUAI1 = MUAI1(NUBAR + 1)
MUAI2 = MUAI2(NUBAR + 1)
IF (MUAI2 .LE. 0) GO TO 285
IF (YBAR .GE. 0) GO TO 250
JCOLL = JBAR - NUBAR + MUAI1 - 1
GO TO 260
250 JCOLL = JBAR + NUBAR - MUAI1 - 1
260 CONTINUE

C LOOP ON A ROW OF WING BOXES, COMPUTING LEFT WING CONTRIBUTIONS
DO 300 NUAIC = MUAI1, MUAI2
IF (JCOLL .LE. 0) GO TO 270
CALL CCDCR(IBX, IBXED, I, JCOLL, I, JCOLL, .F., ICD)
IF (ICD .LE. 0) GO TO 270
C A CONTRIBUTING BOX HAS BEEN FOUND. GET THE AIC LOCATION
KAIC = IAIC + NUAIC
C GET THE NORMAL-WASH LOCATION
ICD = LOCSDW(I, JCOLL, IPNTSW, LNPTSW, 1, LNPTSW)
C ADD THIS CONTRIBUTION TO THE SUMS
EN = ENUS(IUS)
GO TO 265
265 EN = ENUS(IUS)

C END OF LOOP FOR LEFT WING ROW CONTRIBUTIONS
260 CONTINUE

C 285 CONTINUE
I = I - 1
IF (I .LE. 0) GO TO 295
IAIC = IAIC + INCAIC
INCAIC = INCAIC + 2
290 CONTINUE
C END OF LOOP FORWARD ON LEFT WING ROWS, FROM 240
280 CONTINUE

C 295 CONTINUE
DWH(IBX) = DWH(IBX) + BINV*(CPSIM*VSUSUM - SPSIM*VSUSUM) * SYM
DSW(IBX) = DSW(IBX) + BINV*(CPSIS*VSUSUM + SPSIS*VSUSUM) * SYM
PSH(IBX) = PSH(IBX) + PSUSUMSYM
IBX = IBX + 1
300 CONTINUE
C END OF LOOP ON RECEIVING BOXES, FOR LEFT WING CONTRIBUTIONS
300 CONTINUE

C DETERMINE WHAT TO PRINT
310 CONTINUE
WRITE (NT6, 6001) TITLE, XNACH, XKYL, JT, IFRST, ILAST
IF (.NOT. PREW) GO TO 330
WRITE (NT6, 6010)
WRITE (NT6, 6011) (SW(I), I = 1, NIBX)
B172
330 CONTINUE
   IF (.NOT. PRSLW) GO TO 340
   WRITE (NT5,6011)
   WRITE (NT5,6013) (SW(I), I = 1, NBXS)
340 CONTINUE
   IF (.NOT. PRSLW) GO TO 400
   IF (NBXS .LT. 2) GO TO 400
   ISUB = 0
   PHI(ISUB) = 2.*PHI(1) - PHI(2)
   PHI(NBXS+1) = 2.*PHI(NBXS) - PHI(NBXS-1)
   FACTOR = B1BETA/2.0 * BINV**2
   DO 390 I = 1, NBXS
      LW(I) = FACTOR*(PHI(I+1) - PHI(I-1))
390 CONTINUE
   WRITE (NT5,6012)
   WRITE (NT5,6013) (LW(I), I = 1, NBXS)
C 400 RETURN
C 800 WRITE (NT6,8000) IR
   GO TO 400
C 8001 FORMAT(1X,20X,8A10/51X,16HFLOWFIELD SAMPLING /40X,TH( MACH ,
   1 F5.5,5X,16HRED. FREQ. = ,F8.5, ZH )/41X,16HSAMPLED AT CHORD SHWLW 00232
   2 I3, 8H, BOX 12, 8H TO BOX 12 /1HD,2X, 4110X,4HREAL,8X,
   3 9H(IMAGINARY) )
8010 FORMAT(1SH0 UP-WASHES - )
8011 FORMAT(1SH0SIDE-WASHES - )
8012 FORMAT(1SH0LONGITUDINAL-WASHES - )
8013 FORMAT(4X,8E16.8)
C 800 FORMAT(1SH0 WARNING - PROBLEMS ENCOUNTERED WHILE GETTING AICS
   1 39H FOR FLOW-FIELD SAMPLING. ERROR CODE = ,15, 4H **** )
   END
PROGRAM SMTH
C
C THIS PROGRAM WILL FIT A LEAST SQUARES POLYNOMIAL SURFACE
C THROUGH THE VELOCITY POTENTIALS FOR A PLANE FORM. A NEW SET OF
C VELOCITY POTENTIALS WILL BE CALCULATED FROM THE POLYNOMIALS
C AND WRITTEN ON THE IVPSC FILE.
C
COMMON /ARRAYS/ KBKCDW, LBKCDW, LBOKC, KBKXCD, LBKXCD, KJALPH, LJALPH,
1       KALPHA, KERNA, IKERNA, KPNTRM, LPNTRM, KDEFSL, KDELPH,
2       LMODES, KPNTRD, LPNTRD, KDSWM, LSWM, KPNTRD, LPNTRD,
3       KDLM, LDNM, KTNVP, LTVP
COMMON /FILES/ MT5, MT6, INRTPC, INFSP, NPLAIC, NSPAIC, NOUTP,
1       IQUSPS, MODESC, IVPSC, IGEOSC, IWTSC, IAICSC
COMMON /IOCONT/ OPLAIC, OPAIC, WGEOM, WTSAR, WTSCL, WTBX, FRBOX,
1       PRPAIC, PRSAIC, PRMODES, PRCOEF, PRDM, PRSW, PRVP,
2       PRBL, PRDCP, PRGAMAF, PRGAMC, PRSL, PRWR, PRNW, PRCN
EQUIVALENCE (PRDM, PRSW, PRVP, PRBL, PRDCP, PRGAMAF, PRGAMC, PRSL)
LOGICAL OPLAIC, OPAIC, WGEOM, WTSAR, WTSCL, WTBX, FRBOX, PRPAIC,
1       PRSAIC, PRMODES, PRCOEF, PRDM, PRSW, PRVP, PRBL, PRDCP,
2       PRGAMAF, PRGAMC, PRSL, PRWR, PRNW, PRCN
COMMON /PROBJW/ RACH, NODES, NSLOP, NKVALS, SMOOTH, NDEG, CRFFIT,
1       EXAC, SUBDV, PLYWOOD
LOGICAL SMOOTH, CRFFIT, EXAC, SUBDV, PLYWOOD
COMMON /KVAL / NKVAL, NKVAL(20), NW(20)
COMMON /GEONTY/ COPLAN, NSUBDS, XSUBDV, NSUBZ, NSUBCN, NSURF,
1       NW, MBMB, MBWB, MBMB, MBWB, MBWB, MBWB,
2       NW, MBMB, MBWB, MBMB, MBWB, MBWB, MBWB,
3       NW, XCENTR
LOGICAL COPLAN
COMMON /GEO2 / TLAX, TLAZ, FSAT, XMBT, XMBT, XMBBT, XMBBT, XMBST,
1       XMBST, XBST, XBST, CARL
COMMON /TAPEIOV/ NFS, NS5, LS, NR, ID(20), ND, IYPE, LRS, LWS, M, N,
1       IARM(10), IPR
DIMENSION IARM(10), IPR
EQUIVALENCE (IPARM, IPR)
COMMON /CHECKPR/ DPPCR, GEOCR, MOCPR, AICCR, NASCCR, SMCCR, GAFCR
LOGICAL DPPCR, GEOCR, MOCPR, AICCR, NASCCR, SMCCR, GAFCR
EQUIVALENCE (CHECKPR, SMCCR, GAFCR)
LOGICAL CHECKPR
C
C DELPHI (NBOOKS2), TVP (NOCLS1 + NOCLS2 * NSUBDV)
C COMPLEX DELPHI (1000), TVP (250), AVPS (1250)
C X(ND, DELPHI + NO. TVP), Y (SAME)
C COMPLEX DELPHI
C DIMENSION X (1250), Y (1250)
C A (NO. COEFF.)
C COMPLEX A (66)
C COMMON /INDEX/ IS (100), JC (100), JS (100), JOC (100)
C XPI (NO. COEFF. + 1), YP (SAME)
C DIMENSION XPI (11), YP (11)
C FELOC (MBW, MYBT, NSUBDV), TELOC (SAME)
C DIMENSION FELOC (250), TELOC (250)
C IFPNTRM (2, NBOOKS2), TSPNTR (SAME)
C DIMENSION IFPNTRM (2, 150)
C DIMENSION TITL (5)
C
REAL K1
COMPLEX VP
LOGICAL MREAD, MWFIT, RANDN, RANDC
K1 = MKVAL(KVAL)
MREAD = .FALSE.
RANDN = .FALSE.
MWFIT = .FALSE.
RANDC = .FALSE.
MKVAL = MKVAL
IF(COPLAN) MKB = MKBT

C

PUT NAME OF SCRATCH FILE FOR SMOOTHED VALUES INTO PLACE

C

NVPSC = TVPSC
REND NVPSC
C
GET THE PLANFORM POINTERS FROM THE MODESC FILE
C
REND MODESC
CALL RDINIT
ITYPE = SHIMIXED
NBRARY = SHIPTRNM
CALL READMX(MODESC, MREAD, RANDN, NFS, NMS, LS, NR, 2, NID, ID, ITYPE, 1
LRS, IPTRNM, 2, NPHRS, PARM, IRR)
IONLAP = IFARM(3)
IF(IRR .NE. 0) GO TO 6020
C
CALL RDINIT
ITYPE = SHMIXED
NFS = 1
NBRARY = THIS PT.
CALL READMX(MODESC, MREAD, RANDN, NFS, NMS, LS, NR, 1, NID, ID, ITYPE, 1
LRS, 1, M, N, PARM, IRR)
IF(IRR .NE. 0) GO TO 6020
C
C
READ THE FEXLC AND TEXLOC ARRAYS FROM THE GEOMETRY SCRATCH
FILE. THESE ARE NEEDED TO INTERPOLATE VELOCITY POTENTIALS AT
BOX EDGES.
C
REND IGEOGC
CALL RDINIT
NMS = 2
IF(NSURF.EQ.1 .OR. COPLAN) NMS=1
ITYPE = SHMKIXED
NBRARY = SHFEXLOC
CALL READMX(IGEOGC, MREAD, RANDN, NFS, NMS, LS, NR, 1, NID, ID, ITYPE, 1
LRS, FEXLOC, M, N, PARM, IRR)
IF(IRR .NE. 0) GO TO 6019
C
CALL RDINIT
ITYPE = SHMIXED
NBRARY = SHTEXLOC
CALL READMX(IGEOGC, MREAD, RANDN, NFS, NMS, LS, NR, 1, NID, ID, ITYPE, 1
LRS, TEXLOC, M, N, PARM, IRR)
IF(IRR .NE. 0) GO TO 6010

B175
C REORDER THE PXLOC AND TXLOC ARRAYS SO THAT THERE ARE
VALUES FOR UNSUBDIVIDED CHORDS ONLY.

IF(NSUBDV.EQ.1) GO TO 120
XSLIDE = NSUBDV - 1*BW
JCCL = NSUBCN
NCCS = MYSB + MYP
DO 110 I=1,NCCS
TXLOC(I) = (TXLOC(JCCL) + XSLIDE)/NSUBDV
PXLOC(I) = (PXLOC(JCCL) + XSLIDE)/NSUBDV
JCCL = JCCL + NSUBDV
110 CONTINUE
120 CONTINUE

C LOOP ON NUMBER OF MODES (ALSO NO. OF V.P.)
RENEW IVPSC
DO 2000 J=1,NMODES

C READ CDPH1 ARRAY FROM IVPSC. THE TVP ARRAY MUST BE SKIPPED
C IF NH IS NOT 1
C CALL RINIT

C ITYPE = SHR Champ
CALL READX(IPVPS,ICREAD,RSN,SNS,LS,SHR,2,NID,ID,ITYPE,
1 LRS,CDPH1,M,N,$PH1,IRR)
IF(IRR.NE.0) GO TO 440
C CALL RINIT
C ITYPE = SHR Champ
CALL READX(IPVPS,ICREAD,RSN,SNS,LS,SHR,2,NID,ID,ITYPE,
1 LRS,TVP,M,N,PH1,IRR)
IF(IRR.NE.0) GO TO 440
C LOOP ON Nsurf TO FIT EACH PLANFORM INDEPENDENTLY.
C DO 1000 NS=1,Nsurf
C MOVE DELPHI FOR PLANFORM NS TO AVP ARRAY DELETING ZERO
C VALUES AND OBTAINING THE (X,Y) COORDINATES IN (I,J) INDICES
C IF(NS.EQ.2) GO TO 100
C FIRST PLANFORM
IBEG = !
ILIM = MYSB
IC = 0
NOH = 0
GO TO 200
C SECOND PLANFORM
100 CONTINUE
IFBT = (ICBT-1*BW)/NSUBDV + 1
IBEG = IFBT

B176
ILIM = MBT
IC = 0
NCH = MBW
IUP = MBW
IF(COPLAN) IUP = IFBT - 1
DO 125 I=1,IUP
IC = IC + JOC(I)
125 CONTINUE
200 CONTINUE
ICS = IC
INO = 0
DO 400 J=IBEG,ILIM
I = IX
IF(NS.EQ.2) I = IX + IOMAP
JST = JS(I)
JEND = JS(I) + JOC(I)-1
DO 400 J=JST,JEND
IC = IC + 1
IB = IS(I+1)
IT = IB + ICH(J+1) -1
IF(IX.LT.IB) GO TO 400
IF(IX.GT.IT) GO TO 400
INO = INO + 1
AVPS(INO) = DELPHI(IC)
X(INO) = I
Y(INO) = J
400 CONTINUE
C ADD THE LEADING EDGE VELOCITY POTENTIAL TO THE AVP ARRAY
C VEL. POT. = 0. UNLESS IT IS FOR SECOND PLANFORM IN COPLANAR
C ANALYSIS
C
JLAST = MBT
IF(NS.EQ.2) JLAST = MBT
DO 600 J=1,JLAST
IB = IS(J+1)
INO = INO + 1
X(INO) = FERLOC(J)
Y(INO) = J
AVPS(INO) = (0.,0.,)
IF(NS.EQ.2) GO TO 600
IF(NCH.CAP.CAP) GO TO 600
DDA = (FERLOC(J+1)-TFLOC(J)) * KL
JT = J
IF(NSUBDV.EQ.0.) JT = NSUBDV * (J-1) + NSUBCN
AVPS(INO) = TVP(JT) + OMPY(COS(DDAV),-SIN(DDAV))
600 CONTINUE
C CALL LEAST SQUARES SURFACE FITTING ROUTINE
C
IDIM = 2
CN = 1.0
IDEG = NDEG
IF(NBEG.EQ.0) GO TO 675
DO 650 I=1,10
IDEG = IDEG + 1
IM = IDEG + 1
650 CONTINUE
675 CONTINUE
W = 2.0
NC = XM * XM + XM + 1.0E-04
KM = (3 * NC) / 2
IF(NC,LE,1MO) GO TO 575
550 CONTINUE
575 CONTINUE
CALL FITTUM (IDEC, IMO, XYA,AVPS,A,CN,IDM)
C
C EVALUATE THE POLYNOMIAL EQUATION FOR DELPHI
C
MEGS = IDEC + 1
IC = 1CS
DO 900 IX = 1BEQ, ILLIM
I = IX
IF(MS.EQ.2) I = IX + 1OMAP
XP(I) = 1.
DO 700 IP = 2, MCSEG
700 XP(IP) = XP(IP-1) + FLOAT(IP)
JI = JS(I)
JT = JC(I) + JI - 1
DO 900 J = JI, JT
IC = IC + 1
IB = IS(J+MCN)
IT = IB + NOC(J+MCN) - 1
IF(IX, LT, IB) GO TO 900
IF(IX, GT, IT) GO TO 900
YP(I) = 1.0
DO 800 JP = 1, MCSEG
800 YP JP = YP JP-1) + FLOAT(JP)
VP = A(I)
IA = 1
DO 850 L2 = 2, MCSEG
DO 850 L3 = 1, L2
JA = L2 - L3 + 1
IA = IA + 1
VP = VP + XP(IP) * YP(L3) * A(IA)
850 CONTINUE
DELPHI (IC) = VP
900 CONTINUE
C
C CALCULATE THE TRAILING EDGE VELOCITY POTENTIALS (TVP ARRAY)
C
IF(MS.EQ.2) GO TO 910
NTST = 1
NTVPS = MB5W
JJ = 0
GO TO 920
910 CONTINUE
NTST = NTVPS + 1
NTVPS = MB5W + MBST
920 CONTINUE
DO 930 J = NTST, NTVPS
930 TVP(J) = (0., 0.)
C
NBEG = NTST + NSUE2
NEND = NTVPS - NSUE2
JC = 0
SMOOTH 00211
SMOOTH 00212
SMOOTH 00213
SMOOTH 00214
SMOOTH 00215
SMOOTH 00216
SMOOTH 00217
SMOOTH 00218
SMOOTH 00219
SMOOTH 00220
SMOOTH 00221
SMOOTH 00222
SMOOTH 00223
SMOOTH 00224
SMOOTH 00225
SMOOTH 00226
SMOOTH 00227
SMOOTH 00228
SMOOTH 00229
SMOOTH 00230
SMOOTH 00231
SMOOTH 00232
SMOOTH 00233
SMOOTH 00234
SMOOTH 00235
SMOOTH 00236
SMOOTH 00237
SMOOTH 00238
SMOOTH 00239
SMOOTH 00240
SMOOTH 00241
SMOOTH 00242
SMOOTH 00243
SMOOTH 00244
SMOOTH 00245
SMOOTH 00246
SMOOTH 00247
SMOOTH 00248
SMOOTH 00249
SMOOTH 00250
SMOOTH 00251
SMOOTH 00252
SMOOTH 00253
SMOOTH 00254
SMOOTH 00255
SMOOTH 00256
SMOOTH 00257
SMOOTH 00258
SMOOTH 00259
SMOOTH 00260
SMOOTH 00261
SMOOTH 00262
SMOOTH 00263
SMOOTH 00264
SMOOTH 00265
SMOOTH 00266
SMOOTH 00267
DO 980 J = BEG, MEND, NSUBDV
C   \( \text{J} = \text{LOCATION IN THE TVP ARRAY (SUBDIVIDED VALUES)} \)
C   \( \text{JJ} = \text{LOCATION IN ARRAYS TEXTLOC, NOC, IS, ETC.} \)
C   \( \text{JC} = \text{UNSUBDIVIDED CHORD NUMBER} \)
C   \( \text{IROW} = \text{ROW NUMBER OF LAST BOX ON THE CHORD} \)
C   \( \text{I} = \text{ROW LOCATION OF IROW IN ARRAY IPNTRM} \)
      JJ = JJ + 1
      JC = JC + 1
      I = TEXTLOC(JJ)
      INOW = I
      XINOR = TEXTLOC(JJ) - 1
      IF (.NOT. COPLAN .AND. JC .GT. MBWD) I = I + IPOVR
      INDB = IPNTRM(I, I) + JC - IPNTRM(2, I)
      IF (INOW .LT. INDB) GO TO 980
C   \( \text{TEST FOR 3 BOXES ON CHORD JJ} \)
      IF(NOC(JJ), LT, 3) GO TO 940
C   \( \text{2 BOXES AND NO MACH RAY AVAILABLE, OR} \)
C   \( \text{3 BOXES OR MORE. DO LINEAR EXTRAPOLATION.} \)
      980 CONTINUE
      INE = IPNTRM(I, I-1) + JC - IPNTRM(2, I-1)
      SDELPH = DELPH(INDB) - DELPH(INDE)
      GO TO 990
C   \( \text{TEST FOR MACH RAY EXTRAPOLATION.} \)
      940 CONTINUE
      IB = IS(JJ-1)
      IX = IB + NOC(JJ-1) + 1
      IF (IROW .LT. IB .OR. IROW .GT. IX) GO TO 945
      IB = IS(JJ-2)
      IX = IB + NOC(JJ-2) + 1
      IM = IROC+1
      IF (IM .GE. IB .AND. IM .LE. IX) GO TO 948
C   \( \text{MACH RAY CANNOT BE USED. TEST FOR 2 BOXES ON CHORD JJ} \)
      945 CONTINUE
      IF (NOC(JJ), LT, 2) GO TO 710
      GO TO 935
C   \( \text{MACH RAY CAN BE USED} \)
      948 CONTINUE
      INCA = IPNTRM(I, I-1) + JC - IPNTRM(2, I-1) - 2
      INC = IPNTRM(I, I) + JC - IPNTRM(2, I) - 2
      SDELPH = 2.0 * DELPH(INDE) - DELPH(INCA) - DELPH(INCB)
      GO TO 950
C   \( \text{JJ = JJ} \)
      IF(NSUBDV, .NE. 1) JT = NSUBDV + NSUBDV * (JJ-1)
      TVP(JT) = DELPH(INDB) + XINOR + SDELPH
      950 CONTINUE
C   \( \text{WRITE THE DELPHI \& TVP ARRAY ON THE KIVPSC FILE} \)
      CALL RDINIT
      ITYPE = 5
      N = IPNTRM(I, NFNTRS) - 1
      NOARR = SDELPH

B179
CALL WRITEX(NIVPS,C,WAVIT,RANDOUC,WNS,NS,LG,LWS,2,ID)
1       DELPHI, ITYPE, 2, N, PARM, IRR)
IF(IRR .NE. 0) GO TO 6040
C
M = NTVPSC
WARRAY = BHTYP
CALL WRITEX(NIVPS,C,WAVIT,RANDOUC,WNS,NS,LG,LWS,2,ID)
1       TVP, ITYPE, 2, N, PARM, IRR)
IF(IRR .NE. 0) GO TO 6040
C
C
PRINT THE SMOOTHED VEL. POT. ARRAY
IF (.NOT. TVP) GO TO 1500
TITL(1) = 8K       WING
TITL(2) = 10HVELOCITY P
TITL(3) = 10HATENTIALS
IF (COPLAN) TITL(1) = 10HMAIN/TTAIL
CALL PRINTR(TITL, NM, DELPHI, 1, 1, MNB, MNBW, IPNTRM)
IF (NSURF. EQ. 1. OR. COPLAN) GO TO 1500
TITL(1) = 8H       TAIL
CALL PRINTR(TITL, NM, DELPHI, 1, IFBT, MNB, MNBW,
1       IPNTRM(1,1,OMP,1))
1500 CONTINUE
IF (.NOT. CHECKPR) GO TO 2000
WRITE (NIT6, 9400) (TVP(I), I=1, NTVPSC)
9400 FORMAT (1X,8F16.8)
C
2000 CONTINUE
C
END FILE NIVPS
RENAME NIVPS
C
CHANGE FILE NAMES
C
IAICSC = 1IVPS
IVPS = NIVPS
C
RETURN
C
610 CONTINUE
WRITE (NIT6, 9010) IGEOSC, IRR
WRITE (NIT6, 9011) WARRAY
GO TO 6100
C
6100 CONTINUE
WRITE (NIT6, 9010) IGEOSC, IRR
WRITE (NIT6, 9041) NM
9100 CONTINUE
WRITE (NIT6, 9101) ID(I), ID(J)
WRITE (NIT6, 9102) PARM, IPARM
WRITE (NIT6, 9103) NF5, NM5
WRITE (NIT6, 9104) IDPRO, M, N
WRITE (NIT6, 9900)
GO TO 9000
C
T010 CONTINUE
WRITE (MT6,9020) SMOOTH 00368
WRITE (MT6,2021) IRW,JC SMOOTH 00369
IC
8000 CONTINUE SMOOTH 00370
CALL FLUSH(1) SMOOTH 00371
CSMOOTH 00372
C
9020 FORMAT(7SMG 100373
1 BE COMPUTED. ) SMOOTH 00374
9021 FORMAT(5X,15HCOORDINATES (I2,1H,I2,1H) ) SMOOTH 00375
9010 FORMAT(5X,15HERROR - WHILE READING THE GEOMETRY SCRATCH FILE A10 SMOOTH 00376
1, 1SH, ERROR CODE = I4,4H ) SMOOTH 00377
9011 FORMAT(5X,15HAN ATTEMPT WAS MADE TO READ THE A6, 8H MATRIX./) SMOOTH 00378
9041 FORMAT(5X,15HAN ATTEMPT WAS MADE TO READ THE VEL. POT. ARRAY NUMBE SMOOTH 00379
1R 13,1H. ) SMOOTH 00380
9050 FORMAT(5X,15HERROR - WHILE WRITING ON THE VEL. POT. SCRATCH FILE SMOOTH 00381
1 A10, 1SH, ERROR CODE = I4,4H ) SMOOTH 00382
9051 FORMAT(5X,15HATTEMPTING TO WRITE VEL. POT. NUMBER 13 ) SMOOTH 00383
9101 FORMAT(5X,15H , A10, 110) SMOOTH 00384
9102 FORMAT(5X,15H PARAMETERS ,I10E11.3, / 10X,*(INTEGER), I7, 9111 ) SMOOTH 00385
9103 FORMAT(5X,15H ,FILE SPACING = ,I3, , MATRIX SPACING = ,I3 ) SMOOTH 00386
9104 FORMAT(5X,15H ,MATRIX TYPE = ,A10, , DIMENSIONED (I4,2H X I4,1H ) SMOOTH 00387
9800 FORMAT(40 ERROR OCCURRED IN SMOOTHING SECTION. *) PTNG 000388
END SMOOTH 00389

B181
SUBROUTINE PRINTR(TITL,MODE,ARRAY,K,IXB,MB,YBI,PTRM)

C TITLE - TITLE TO PRINT FOR THE ARRAY
C MODE - MODE SHAPE NUMBER
C ARRAY - ARRAY TO BE PRINTED

DIMENSION ARRAY(K,1), TITL(3)
COMPLEX ARRAY
DIMENSION IPTRM(2,100)
COMMON /CONTROL/ PREVEX,OMACH, TITLE(8), PRVGEOM, PRVMODE, DIHM, DIHT, CONTL
1 DEFAULT
LOGICAL PRVGEOM, PRVMODE, DIHM, DIHT, DEFAULT
COMMON /PROBLM/ XMAC, NMACH, NTSLP, NKVALS, SMOOTH, NDEG, CRDFIT,
1 EXAI, SUBD, PLYWOOD
LOGICAL SMOOTH, CRDFIT, EXAI, SUBD, PLYWOOD
COMMON /FILES/ NT5, NT6, INTAPE, INFSP, NPLAI, NSPAIC, NCPUP,
1 ICUPS, MDESC, TPS, ICES, ICESND, TCPS, IAICES
COMMON /KVAL/ IKVAL, XKVAL(20), XK(20)
DIMENSION PC(2)
DIMENSION (50,100)
EQUIVALENCE (S(1),BUFF(1)), (D**,BUFF(125))
REAL KI
INTEGER PAGE
COMMON /RWBUFF/ BFCODE, IBFCONT, BUFF(3280)
DATA PC /10PAGE CONTI,4HNUED /
DATA BLANK /1 /
DATA XINIT /-1.0 /
KI = XKVAL(IKVAL)
IF (KX(KI)<.NE. XINIT) KI = XKVAL(IKVAL)

PAGE = 0
M = 1
M = 4
IF(M.GT.MB) M = MB
100 LINE = 100
200 DO 1400 I = IXB, MB
1400 DO 300 J=N,M
300 CONTINUE
IF(LINE.LE.50) GO TO 900
PAGE = PAGE + 1
LINE = 4
WRITE (NT6,9001) TITLE, TITL, XMACH, KI, MODE
IF(PAGE.GT.1) GO TO 700
WRITE (NT6,9005) PC
GO TO 800
700 WRITE(NT6,9005)
800 CONTINUE
WRITE(NT6,6005) (BLANK, J=N,M)
WRITE(NT6,6007) (BLANK, J=N,M)
900 CONTINUE
JS = IPTRM(2,1)
IDX = IPTRM(1,1)
JE = IPTRM(1,1+1) - IDX + JS -1
B182
IF(JE.EQ.0) GO TO 1400
DO 1000 J=5,JE
S(J) = REAL(ARRAY(1,IDX))
D(J) = AIMAG(ARRAY(1,IDX))
IDX = IDX + 1
1000 CONTINUE
DO 1200 J=N,M
IF(S(J)) 1300,1100,1300
1100 CONTINUE
IF(D(J)) 1300,1200,1300
1200 CONTINUE
GO TO 1400
1300 WRITE(NF6,9013) I,(S(J),D(J),J=N,M)
LINE = LINE + 1
1400 CONTINUE
M = M+1
N = N+1
IF(M.GT.MW8) GO TO 1500
IF(M.GT.MWB) M=WB
IF(LINE.GT.45) GO TO 100
WRITE(NF6,6005) (BLANK, J=N,M)
WRITE(NF6,6007) (BLANK, J=N,M)
LINE = LINE+3
GO TO 200
1500 CONTINUE
RETURN
END
9001 FORMAT(1H1,20X,8A10/ 46X,4SMOOTHED *,3A10/ 46X,7H(MACH F5.5,5X,
1 12RED. FREQ. =FB,5, * )/ 52X,4MODE SHAPE*, 13)
9002 FORMAT(44X,42:(H-),20X,A10,A4)
9006 FORMAT(4HORC, A1,14X,9HCHORD,13,3(A1,22X,9HCHORD,13))
9007 FORMAT(3X, 4D11.9X,4REAL,8X,9HIMAGINARY))
9013 FORMAT(I4,8E16.8)
END
SUBROUTINE FITTER(M,N,X,Y,Z,C,ZN,IDIM)
DIMENSION X(100), Y(100), Z(100), C(IDIM,66)
DIMENSION A(66,66), X'Y(I), Y(11)
DIMENSION VS(10)
LOGICAL COMPLEX
C
C M - DEGREE OF POLYNOMIAL EQUATION
C N - NUMBER OF DATA POINTS TO FIT CURVE THROUGH
C X - X COORDINATE OF DATA POINT
C Y - Y COORDINATE OF DATA POINT
C Z - Z COORDINATE OF DATA POINT
C C - OUTPUT COEFFICIENT ARRAY
C CN - SCALE FACTOR
C CN - SCALE FACTOR
C IDIM - INDICATOR OF REAL OR COMPLEX FUNCTION
C = 1, FUNCTION IS REAL
C = 2, FUNCTION IS COMPLEX
C IF COMPLEX SET DIMENSIONS OF FUNCTION AND COEFFICIENTS
C TO (IDIM = --- )
C DETERMINE NUMBER OF COEFFICIENTS
C EPS = 1.0E-04
COMPLEX = .FALSE.
IF(IDIM.EQ.2) COMPLEX = .TRUE.
C SCALE DATA TO REDUCE MAGNITUDE OF MATRIX TERMS.
C SHOULD AVOID BOUNDARY DUE TO OVERFLOW CONDITIONS.
C IF(CN.EQ.0) CN=1.0
IF(CN.EQ.1.0) GO TO 15
DO 5 1=1,N
X(1) = X(1)/CN
Y(1) = Y(1)/CN
5 CONTINUE
15 CONTINUE
MM = M + 1
MC = MM/2.
NC = XMM/2 + YC + EPS
IF(NC.LT.N) GO TO 25
MM = M
GO TO 15
25 CONTINUE
C
MC = NC
C DETERMINE THE MAXIMUM DEGREE THAT CAN BE COMPUTED N
C EACH DIRECTION AND SET UP ORDER OF SOLUTION.
C NEV = 1
MBX = M
VS(1) = X(1)
DO 60 J=1, M
DO 50 J=1,NEV
IF(X(I).EQ.VS(J)) GO TO 55
50 CONTINUE
NEV = NEV + 1
VS(NEV) = X(1)
IF (NDV-1.EQ.M) GO TO 85
93 CONTINUE
90 CONTINUE
NDX = NDV - 1
85 CONTINUE

C
NDV = 1
NDY = M
VS(I) = Y(I)
DO 80 I=1,N
DO TO J=1,NDV
IF (Y(I).NE.VS(J)) GO TO 75
70 CONTINUE
NDV = NDV + 1
VS(NDV) = Y(I)
IF (NDV-1.EQ.M) GO TO 85
75 CONTINUE
80 CONTINUE
NDV = NDV - 1
85 CONTINUE

C
IT0 = NC + 1
IT01 = IT0
IF (COMPLX) IT0 = IT0 + 1

C
ZERO OUT THE A ARRAY

C
DO 95 I=1,NC
V(I) = 0.0
IF (.NOT.COMPLX) GO TO 90
C2(I) = 0.0
90 CONTINUE
DO 95 J=1,IT0
95 A(I,J) = 0.0

C
DETERMINE DEVIATION EQUATION AND SQUARE THE EQUATION

C
AI(1) =1.0
XP(1) =1.0
YP(1) =1.0
M = M + 1
DO 20 K=1,N
K0 10 L=2,MN
Y(L) = XP(L-1)*X(K)
YP(L) = YP(L-1)*Y(K)
10 CONTINUE

C
I = 1
DO 40 L=2,MN
DO 20 LL=1,L
IL = L - LL +1
IF (IL-1.GT.M) GO TO 30
IF (IL-1.GT.MD) GO TO 20
I = I + 1
AI(I) = XP(IL)*YP(IL)
20 CONTINUE
30 CONTINUE
40 CONTINUE
AI(I+1) = Z(I,K)
IF(COMPLEX) AI(I+2) = Z(I,K)
IF(K,GT,1) GO TO 45
NC = I
ITOT = NC + 1
ITOT1 = ITOT
IF(COMPLEX) ITOT = ITOT + 1
45 CONTINUE
C
DO 1100 I=1,NC
DO 1100 J=1,ITOT
ASAV = AI(I)*AI(J)
A(I,J) = A(I,J) + ASAV
1100 CONTINUE
200 CONTINUE
C
SQUARE ROOT METHOD
INTERMEDIATE MATRIX
DO 1200 I=1,NC
IM = I-1
TMP=0.0
IF(I.EQ.1) GO TO 1150
DO 1120 L=1,IM
1120 TMP = TMP + A(L,I)**2
1150 CONTINUE
T = A(I,1) - TMP
IF(T.GT.EPS) GO TO 4
A(I,1) = 0.0
GO TO 1200
4 CONTINUE
A(I,1) = SQRT(T)
IF(A(I,1).GT.EPS) GO TO 1155
A(I,1),ITOT = 0.0
GO TO 1200
1155 CONTINUE
C
JS = J+1
DO 1180 J = JS,ITOT
TMP=0.0
IF(J.EQ.1) GO TO 1175
DO 1180 L=1,IM
1175 A(I,J) = A(I,J) - TMP/A(I,1)
1180 CONTINUE
1200 CONTINUE
C
BACK SUBSTITUTE FOR COEFFICIENTS
DO 1300 K=1,NC
I = NC - K + 1
IP = I+1
TMP1 = 0.0
TMP2 = 0.0
IF(A(I,1).GT.EPS) GO TO 1325
C(I,1) = 0.0
IF(COMPLEX) C(2,1) = 0.0
1300 CONTINUE
GO TO 1400

1325 CONTINUE
   IF (I.EQ.NC) GO TO 1375
   DO 1390 L=1,NC
      TMP1 = TMP1 + A(I,L)*C(I,L)
   IF (.NOT.COMPLEX) GO TO 1350
      TMP2 = TMP2 + A(I,L)*C(2,L)
   1350 CONTINUE
1375 CONTINUE
   C(1,1) = A(I,1) - TMP1/A(I,1)
   IF (.NOT.COMPLEX) GO TO 1400
   C(2,1) = A(I,1) - TMP2/A(I,1)

1400 CONTINUE
C
C REORDER THE COEFFICIENTS IN CORRECT POWERS
C OF X AND Y.
C
   IF (MC.EQ.NC) GO TO 1475

C
IZ = 1
I = 1
DO 1440 L=2,MC
   DO 1420 LL=1,L
      IL = L - LL + 1
      I = I + 1
   IF (LL-1.LE.MDXY.AND.IL-1.LE.MDXY) GO TO 1410
      X(I) = 0.0
      Y(I) = 0.0
   GO TO 1420
C
1410 CONTINUE
   IZ = IZ + 1
   X(I) = C(1,IZ)
   Y(I) = C(2,IZ)
   IF (COMPLEX) Y(I) = C(2,IZ)

1420 CONTINUE
1440 CONTINUE
C
   DO 1450 I=2,MC
      C(1,I) = X(I)
   IF (COMPLEX) C(2,I) = Y(I)

1450 CONTINUE
1475 CONTINUE
C
C ELIMINATE THE SCALE FACTOR FROM THE COEFFICIENTS.
C
   IF (CN.EQ.1.0) GO TO 1700
   I=1
   CP = 1.0/CN
   DO 1600 L=2,MC
      DO 1500 LL=1,L
         I = I + 1
         C(1,I) = C(1,I)*CP
         C(2,I) = C(2,I)*CP
      1500 CONTINUE
   CP = CP/CN

1600 CONTINUE
1700 CONTINUE
C THE C ARRAY NOW CONTAINS THE COEFFICIENTS.
C
RETURN
END

<table>
<thead>
<tr>
<th>FITTER 00230</th>
</tr>
</thead>
<tbody>
<tr>
<td>FITTER 00231</td>
</tr>
<tr>
<td>FITTER 00232</td>
</tr>
<tr>
<td>FITTER 00233</td>
</tr>
<tr>
<td>FITTER 00234</td>
</tr>
</tbody>
</table>
This program will fit a curve through the velocity potentials along each chord independently. These curves will be used to evaluate a smoothed velocity potential.
$\text{R1} = \text{IVAL}(\text{IRVAL})$

$\text{MREAD} = \text{.FALSE.}$

$\text{RANCM} = \text{.FALSE.}$

$\text{MWRIT} = \text{.FALSE.}$

$\text{RANDO} = \text{.FALSE.}$

$\text{MGB} = \text{MGBW}$

$\text{IF(COPLAN)} \text{MGB} = \text{MGBT}$

\[ \begin{align*}
\text{C} & \quad \text{PUT NAME OF SCRATCH FILE FOR SMOOTHED VALUES INTO PLACE} \\
\text{C} & \quad \text{MVPSG} = \text{IAICSC} \\
\text{C} & \quad \text{RENEW MVPSG} \\
\text{C} & \quad \text{GET THE FINER POINTERS FROM THE MODESC FILE} \\
\text{C} & \quad \text{RENEW MODESC} \\
\text{C} & \quad \text{CALL RNINIT} \\
\text{C} & \quad \text{ITYPE} = \text{SHMIXED} \\
\text{C} & \quad \text{MARRAY} = \text{SHHPTMT} \\
\text{C} & \quad \text{CALL READX(MODESC,MREAD,RANCM,NFS,NLS,NMR,2,NID,ID,ITYPE,} \\
\text{C} & \quad \text{LRS,IPTRM,2,IPNTRS,PARM,IRR)} \\
\text{C} & \quad \text{IOLAP} = \text{IPARM(3)} \\
\text{C} & \quad \text{IF(IRR.NE.0) GO TO 6020} \\
\text{C} & \quad \text{CALL RNINIT} \\
\text{C} & \quad \text{ITYPE} = \text{SHMIXED} \\
\text{C} & \quad \text{NFS} = 1 \\
\text{C} & \quad \text{MARRAY} = \text{SHHPS PT} \\
\text{C} & \quad \text{CALL READX(MODESC,MREAD,RANCM,NFS,NLS,NMR,1,NID,ID,ITYPE,} \\
\text{C} & \quad \text{LRS,IS,M,N,PARM,IRR)} \\
\text{C} & \quad \text{IF(IRR.NE.0) GO TO 6020} \\
\text{C} & \quad \text{READ THE FEXLOC AND TEXLOC ARRAYS FROM THE GEOMETRY SCRATCH} \\
\text{C} & \quad \text{FILE. THESE ARE NEEDED TO INTERPOLATE VELOCITY POTENTIALS AT} \\
\text{C} & \quad \text{BOX EDGES.} \\
\text{C} & \quad \text{RENEW IGEOSGC} \\
\text{C} & \quad \text{CALL RNINIT} \\
\text{C} & \quad \text{NFS} = 2 \\
\text{C} & \quad \text{IF(NSURF.EQ.1.OR.CPLAN) NFS=1} \\
\text{C} & \quad \text{ITYPE} = \text{SHMIXED} \\
\text{C} & \quad \text{MARRAY} = \text{SHFEXLOC} \\
\text{C} & \quad \text{CALL READX(IGEOSGC,MREAD,RANCM,NFS,NLS,NMR,1,NID,ID,ITYPE,} \\
\text{C} & \quad \text{LRS,FEXLOC,M,N,PARM,IRR)} \\
\text{C} & \quad \text{IF(IRR.NE.0) GO TO 6010} \\
\text{C} & \quad \text{CALL RNINIT} \\
\text{C} & \quad \text{ITYPE} = \text{SHMIXED} \\
\text{C} & \quad \text{MARRAY} = \text{SHFTEXLOC} \\
\text{C} & \quad \text{CALL READX(IGEOSGC,MREAD,RANCM,NFS,NLS,NMR,1,NID,ID,ITYPE,} \\
\text{C} & \quad \text{LRS,TEXLOC,M,N,PARM,IRR)} \\
\text{C} & \quad \text{IF(IRR.NE.0) GO TO 6010} \\
\text{C} & \quad \text{REORDER THE FEXLOC AND TEXLOC ARRAYS SO THAT THERE ARE} \\
\text{C} & \quad \text{VALUES FOR UNSUBDIVIDED CHORDS ONLY.}
\end{align*} \]
C IF(NSUBDV.EQ.1) GO TO 120
XSLIDE = NSUBDV - IXBW
JCOL = NSUBDV
NCOLS = MYBW + MYBT
DO 110 I=1,NCOLS
TEXLOC(I) = (TEXLOC(JCOL) + XSLIDE)/NSUBDV
FEXLOC(I) = (FEXLOC(JCOL) + XSLIDE)/NSUBDV
JCOL = JCOL + NSUBDV
110 CONTINUE
120 CONTINUE
C IFBT = (IXBT-IXBW)/NSUBDV + 1
C LOOP ON NUMBER OF MODES (ALSO NO. OF V.P.)
REWIND IVPSC
DO 200 N=1,NMODES
C READ DELPHI ARRAY FROM IVPSC. THE IVPSC ARRAY MUST BE SKIPPED
C IF NW IS NOT 1
C CALL REINIT
ITYPE = NHMIXED
NNS = 1
IF(NW.EQ.1) NNS = 0
CALL READM(IVPSC,NREAD,RANDIN,NFS,NNS,LS,NR,2,NID,ITYPE,1
URS,DELPHI,M,N,PARK,IRR)
IF(IRR.NE.0) GO TO 6040
C LOOP ON NUMBER OF CHORDS
C NCORDS = MYBW
IF(NSURF.EQ.2) NCORDS = MYBW + MYBT
DO 100 J=1,NCORDS
NC = 1
IF(J.GT.MYBW) NC = MYBW + 1
IST = IS(J)
NK = IST + NC(J) - 1
JSUM = 0
ITROW = IST
IF(NOR.COPLAN.AND..GT.MYBW) ITROW = IST + INCLAP
DO 100 I=1,ITROW
100 JSUM = JSUM + JCC(I)
JSUM = JSUM - JCC(ITROW) + 1
IND = 0
DO 200 I=IST,NK
IX = I
IND = IND + 1
IF(NOR.COPLAN.AND..GT.MYBW) IX = I + INCLAP
ISUB = JSUM + J - JS(IX) - NC + 1
X(IND) = I
AVPS(IND) = DELPHI(ISUB)
JSUM = JSUM + JCC(IX)
200 CONTINUE
C FIND THE DERIVATIVE OF DELPHI, AND SMOOTH THESE
C
VC = AVPS(1)
INDM1 = IND - 1
AVPA = 0.5 * (AVPS(1) + AVPS(2))
AVPS(1) = (AVPS(2) - AVPS(1))/(X(2) - X(1))
DO 350 I = 2, INDM1
AVPB = 0.5 * (AVPS(I) + AVPS(I-1))
AVPS(1) = AVP - AVPA
AVPA = AVPB
350 CONTINUE
AVPS(IND) = (AVPS(IND) - AVPA)/0.5
IND = IND + 1
INDM1 = INDM1 + 1
AVPS(IND) = AVPS(INDM1)
X(IND) = X(INDM1) + 0.5
C
XINC = X(I)
DO 375 I = 1, IND
X(I) = X(I) - XINC
375 CONTINUE
IDE = NDEG
C
CALL FITTING ROUTINE LEAST SQUARES ERROR CURVE.
C
CALL CURVE(IDE, IND, X, AVPS, A)
C
EVALUATE THE CURVE FOR SMOOTH DELPHI VALUES
C
NDEG = IDE + 1
JSUM = 0
DO 400 I = 1, ITROW
400 JSUM = JSUM + JCC(I)
C
JSUM = JSUM - JCC(ITROW) + 1
DO 500 I = 1, NK
IX = I
IF(NOT.COPLAN.AND.J, GT, MYPD) IX = I + ITROW
ISUB = JSUM + J - JCC(IX) - NC + 1
VP = VC
XV = FLOAT(I) - XINC
XP = 1.0
DO 450 L = 1, NDEG
XP = XP * XV
XD = L
XPI = XP / XD
VP = VP + A(L) * XPI
450 CONTINUE
DELPHI(ISUB) = VP
JSUM = JSUM + JCC(IX)
500 CONTINUE
C
CALCULATE THE TRAILING EDGE VELOCITY POTENTIALS (TVP ARRAY)
C
L = J
IV(NSUBDV,NE,1) L = NSUBDV * (J-1) + NSUBCN
TVP(I) = (0.,0.)
JJ = J
JC = J
B192
IF (JG .GT. MYB) JC = J - MYB
I = TECLOC(JJ)
KMER = TECLOC(JJ) + 1
IROW = 1
IF (.NOT. COPLAN.AND.J .GT. MYB) I = I + IOCLAP
INEB = IPNTRM(1,1) + JC - IPNTRM(2,1)
C
C TEST FOR 3 BOXES ON CHORD JJ
IF (NOC(JJ) .LT. 3) GO TO 940
C
C 2 BOXES AND NO MACH RAY AVAILABLE, OR
C 3 BOXES OR MORE. NO LINEAR EXTRAPOLATION.
930 CONTINUE
INE = IPNTRM(1,1-1) + JC - IPNTRM(2,1-1)
ADELPH = DELPH(INEB) - DELPH(INDE)
GO TO 990
C
C TEST FOR MACH RAY EXTRAPOLATION.
940 CONTINUE
IB = IS(JJ-1)
IX = IB + NOC(JJ-1) * 1
IF (IROW .LT. IB .OR. IROW .GT. IX) GO TO 945
IB = IS(JJ-2)
IX = IB + NOC(JJ-2) + 1
IM = IRow - 1
IF (IM .GE. IB .AND. IM .LT. IX) GO TO 948
C
C MACH RAY CANNOT BE USED. TEST FOR 2 BOXES ON CHORD JJ
945 CONTINUE
IF (NOC(JJ) .LT. 2) GO TO 7010
GO TO 940
C
C MACH RAY CAN BE USED
948 CONTINUE
INCA = IPNTRM(1,1-1) + JC - IPNTRM(2,1-1) - 2
INDC = IPNTRM(1,1) + JC - IPNTRM(2,1) - 1
ADELPH = DELPH(INEC) - DELPH(INCA) - DELPH(INDE)
990 CONTINUE
JT = JJ
IF (NSUBDV. NE. 1) JT = NSUBDV * (JJ-1) + NSUBCN
TVP(JT) = DELPH(INEB) + XINCH + DELPH
990 CONTINUE
C
C 1000 CONTINUE
C
C WRITE THE LLPHI AND TYP ARRAY ON THE TVPSC FILE
CALL REINIT
ITYPE = SSXRED
N = IPNTRM(1,MPYCRS) - 1
NROHR = GMELPHI
CALL WRTEK(MYVPSC,HRHIT,RANDCU,MS,NS,LS,HR,LW,2,1D, 1
S = DELPH,ITYPE,2,N,PARH,HRH)
IF (ISRC. NE. 0) GO TO 6340
C
MYVP = NSUBDV * NCARDS
* WHAT
NROHR = GMHVP
C
8193
CALL WRITE(NPSC1,NPSC2,NPSC3,NPSC4,NPSC5,NPSC6,NPSC7,NPSC8)
1
WRITE(TYPE,NPSC1,NPSC2,NPSC3,NPSC4,NPSC5,NPSC6,NPSC7,NPSC8)
IF(NPSC1,NE.0) GO TO 6040
C
C PRINT THE SMOOTHED VEL. JT. ARRAY
C
IF(NPSC1,NE.0) GO TO 1500
TID1 = 0.0
TID2 = 0.0
TID3 = 0.0
IF(10000) TID1 = 1.0
TID3 = 1.0
IF(NPSC1,NE.0) GO TO 1500
TID1 = 0.0
TID3 = 0.0
C
1500 CONTINUE
IF(NPSC1,NE.0) GO TO 2000
WRITE(NPSC1,NPSC2) (TIP(1),I=1,NPSC2)
9400 FORMAT(/(1X,RE16.8))
C
2000 CONTINUE
C
END FILE NPSC
RESID NPSC
CHANGE FILE NAMES
C
NPSC = NPSC1
NPSC = NPSC2
C
RETURN
0010 CONTINUE
WRITE(NPSC1,NPSC2) (TIP(1),I=1,NPSC2)
WRITE(NPSC1,NPSC2) (MIPSC(1),I=1,NPSC2)
GO TO 6000
8020 CONTINUE
WRITE(NPSC1,NPSC2) (MIPSC(1),I=1,NPSC2)
WRITE(NPSC1,NPSC2) (MIPSC(1),I=1,NPSC2)
GO TO 6000
B040 CONTINUE
WRITE(NPSC1,NPSC2) (MIPSC(1),I=1,NPSC2)
WRITE(NPSC1,NPSC2) (MIPSC(1),I=1,NPSC2)
GO TO 6000
C
GO00 CONTINUE
WRITE(NPSC1,NPSC2) (TIP(1),I=1,NPSC2)
WRITE(NPSC1,NPSC2) (TIP(1),I=1,NPSC2)
GO TO 6000
C
7010 CONTINUE
WRITE(NPSC1,NPSC2) (TIP(1),I=1,NPSC2)
WRITE(NPSC1,NPSC2) (TIP(1),I=1,NPSC2)
C
6000 CONTINUE
CALL FLUSH(1)
C
6020 FORMAT(33H2554 ERROR - NO TIP TRAILING EDGE VELOCITY POTENTIAL CAN OCCUR)
1 BE COMPUTED. *** )
8021 FORMAT(5X,13MCOORDINATES (I2,1H,12,1H )
8010 FORMAT(5X)*** ERROR - WHILE READING THE GEOMETRY SCRATCH FILE AIG CHORDF 00314
1, 15H, ERROR CODE = I4,4H *** )
8011 FORMAT(5X,32HAN ATTEMPT WAS MADE TO READ THE A6, 8H MATRIX,/// ) CHORDF 00315
8041 FORMAT(5X,54HAN ATTEMPT WAS MADE TO READ THE VEL. POT. ARRAY NUMBE CHORDF 00316
1R 15,1H. )
8050 FORMAT(5G6H)*** ERROR - WHILE WRITING ON THE VEL. POT. SCRATCH FILE CHORDF 00317
1 A10, 15H, ERROR CODE = I4,4H *** )
8051 FORMAT(5X,3HATTEMPTING TO WRITE VEL. POT. NUMBER 13 )
9101 FORMAT(5X,**MATRIX ID = *, A10, I10)
9102 FORMAT(5X,**PARAMETERS *,10E11.3, / 10X,**INTEGER)*, I7, 9I11 )
9103 FORMAT(5X,**FILE SPACING = *,I3,** MATRIX SPACING = *,I3 )
9104 FORMAT(5X,**MATRIX TYPE =*,A10,*, DIMENSIONED (*I4,2H X,14,1H )
9900 FORMAT(40 ERRORS OCCURRED IN CHORD-FIT SMOOTHING SECTION. *)
END
SUBROUTINE PRINTR (TITL, IINDX, ARRAY, X, XK, MYB, IPNTBM)

C TITL - TITLE TO PRINT FOR THE ARRAY
C IINDX - MODE SHAPE NUMBER
C ARRAY - ARRAY TO BE PRINTED
C

DIMENSION ARRAY (K, 1), TITL(3)
COMPLEX ARRAY
COMMON /FILES/ NT5, NT6, INTAPL, IMFSP, NPLAIC, NSFIE, NOUTF,
1 IQUFS, MODESC, IFVSC, ISGSC, IFWFS, IAISSC
COMMON /CONTROL/ PREVEX, XMAN, TITLE(8), PRVGEN, PRVMODE, DIHM, DIHT, PRINTR
1 DEFAULT
LOGICAL PRVGEOM, PRVMODE, DIHW, DIHT, DEFAULT
COMMON /PROBLM/ XMAN, IINDX, NT5LOF, NKSLOF, SMOOTH, NDEG, ORGDF, ILOC
1 EXAI, SUBDI, PLYWOOD
LOGICAL SMOOTH, ORGDF, EXAI, SUBDI, PLYWOOD
COMMON /KVAL/ IKS(20), IK(20)
DIMENSION IPNTBM(2, 50)
DIMENSION PC(2)
DIMENSION S(50), D(50)
EQUIVALENCE (S(1), BUF(1)), (D(1), BUF(1251))
REAL K1
INTEGER PAGE

COMMON /RWBUFF/ BFCOND, IBCONT, BUF(3280)
DATA PC / 160PAGE CONTI, 4MNUED /
DATA BLANK / 1H /
DATA XINIT / -1.0 /
K1 = IKS(IKVAL)
IF(KS(IKVAL), NE, XINIT) K1 = IKS(IKVAL)
PAGE = 0
N = 1
M = 4
IF(I(M, GT, MYB)) M = MYB

100 LINE = 100
200 CONTINUE
   DO 1400 I = 1, XB, XK
       DO 300 J = 1, N, M
          S(J) = 0.0
          D(J) = 0.0
   300 CONTINUE
   IF(LINE, LE, 50) GO TO 900
   PAGE = PAGE + 1
   LINE = 4
   WRITE (NT6, 9001) TITLE, TITL, XMAN, K1, IINDX
   IF(PAGE, EQ, 1) GO TO 700
   WRITE (NT6, 9005) PC
   GO TO 800
   700 WRITE(NT6, 9005)
   800 CONTINUE
   WRITE(NT6, 6005) (BLANK, J, J = N, M)
   WRITE(NT6, 6007) (BLANK, J, J = N, M)

900 CONTINUE
   JS = IPNTBM(2, 1)
   IF(JS, LE, 0) GO TO 1400
   IDX = IPNTBM(1, 1)
   JE = IPNTBM(1, 1) - IDX + JS - 1

B196
IF(JE.EQ.0) GO TO 1400
DO 1000 J=JS,JE
S(J) = REAL ARRAY (1,IDX)
D(J) = AIMAG ARRAY (1,IDX)
IDX = IDX + 1
1000 CONTINUE
DO 1200 J=N,M
IF(S(J)) 1300,1100,1300
1100 CONTINUE
IF(D(J)) 1300,1200,1300
1200 CONTINUE
GO TO 1400
1300 WRITE (NT6,9013) 1, (S(J),D(J),J=N,M)
LINE = LINE + 1
1400 CONTINUE
M = N+4
N = N+4
IF(N.GT.MYB) GO TO 1500
IF(N.GT.MYB) M = MYB
IF(LINE.GT.45) GO TO 100
WRITE (NT6,6006) (BLANK,J,J=N,M)
WRITE (NT6,6007) (BLANK, J=N,M)
LINE = LINE+3
GO TO 200
1500 CONTINUE
RETURN
9000 FORMAT(1H,12D,8A10/ 46X,4SMOOTHED 9,3A10/ 46X,7H MACH FS.5,5X,
1 12RESH. FREQ. =F8.5, * *) / 52X,4MORE SHAPE* (3)
9005 FORMAT(44X,42(1H=),20X,A10,A4)
6006 FORMAT(4HOROW, A1,14X,SHORD,13, 3(A1,22X,SHORD,13))
6007 FORMAT(3X, 4(A1,9X,4REAL,6X,9IMAGINARY))
9013 FORMAT(14,8E16.8)
END
SUBROUTINE CURVE(N,M,X,Z,C)
DIMENSION X(50), Z(2,50), C(2,21)
DIMENSION A(I23), A(I2123), X(6)
C
M = DEGREE OF POLYNOMIAL EQUATION
N = NUMBER OF DATA POINTS TO FIT CURVE THROUGH
X = X COORDINATE OF DATA POINT
Z = Z COORDINATE
C = OUTPUT COEFFICIENT ARRAY
C
EPS = 1.0E-04
IF(N.LT.W-1) N = N-1
NC = N + 1
C
ZERO OUT THE ARRAYS NEEDED
C
ITOR = NC + 2
DO 100 I=1,NC
C(I,1) = 0.0
C2(I) = 0.0
100 CONTINUE
DO 100 I=1,ITOR
A(I,1) = 0.0
100 CONTINUE
C
DETERMINE DEVIATION EQUATION AND SQUARE THE EQUATION
C
A(1) = 1.00
DO 400 K=1,N
A(I) = 1.0
DO 200 L=2,NC
A(L) = A(L-1) * X(K)
200 CONTINUE
A(NC+1) = Z(I,K)
A(NC+2) = Z(2,K)
C
DO 300 I=1,NC
DO 300 J=1,ITOR
ASAV = A(I) * A(J)
A(I,J) = A(I,J) + ASAV
300 CONTINUE
400 CONTINUE
C
SQUARE ROOT METHOD INTERMEDIATE MATRIX
C
DO 1200 I=1,NC
IM = I-1
TMP = 0.0
IF(I.EQ.1) GO TO 600
DO 500 L=1,IM
500 TMP = TMP + A(L,1) ** 2
600 CONTINUE
T = A(I,1) - TMP
IF(T.GT.EPS) GO TO 700
A(I,1) = 0.0
GO TO 1200
700 CONTINUE

6198
A(I,J) = SORT(I)
IF(A(I,J).GT.EPS) GO TO 800
A(I,ITOT) = 0.0
GO TO 1200
800 CONTINUE
C
J8 = I+1
DO 1100 J=JS,ITOT
THP = 0.0
IF(I.EQ.1) GO TO 1000
DO 900 L=1,1M
900 THP = THP + A(L,J)*A(L,J)
1000 A(I,J) = (A(I,J)-THP)/A(I,J)
C100 CONTINUE
1200 CONTINUE
C
C    BACK SUBSTITUTE FOR COEFFICIENTS
C
DO 1500 K=1,NC
II = NC - K + 1
IP1 = II + 1
THP1 = 0.0
THP2 = 0.0
IF(A(I,I).GT.EPS) GO TO 1500
C(I,I) = 0.0
C(2,I) = 0.0
GO TO 1600
1300 CONTINUE
IF(I.EQ.NC) GO TO 1500
DO 1400 L=IP1,NC
THP1 = THP1 + A(I,L)*C(I,L)
THP2 = THP2 + A(I,L)*C(2,L)
1400 CONTINUE
1500 CONTINUE
C(I,I) = (A(I,NC+1)-THP1)/A(I,I)
C(2,I) = (A(I,ITOT)-THP2)/A(I,I)
1600 CONTINUE
RETURN
END
PROGRAM FORCES

COMMON /ARRAYS/ KBXCDW, LBCDWT, LBOGK, KBXCDT, LBCDWT, KJALPH, LJALPH,
           KALPHA, KEREN, LKEREN, KFNTRH, LFNTRH, KDFSL, KELPH,
           LMODES, KFNTRD, LFNTRD, KSDW, LSDW, KFNTRD, LFNTRD,
           KDMLDW, KDMLDWT, LTVP

COMMON /FILES/ NTS, NTG, INTAPE, INFSP, NPLAIC, NPLAIC, NOUTP,
           ICOFSP, MODESC, IVYSC, IGEOSC, IVTFSC, IAICSC

EQUIVALENCE (IVTFSC, IVTSLSC)

COMMON /IOCONT/ OPLAIC, OSPLAIC, WTGEOM, WTGNMF, WTSL, WTDL, PRBOX,
           PRPAIC, PRSAIC, PRCON, PRCCEF, PRDW, PRSW, PRVP,
           PRBL, PRDCP, PRGNAC, PRSL, PRLL, PRHM, PRCH

COMMON /KVAL/ ERR, MOSKEN, IPKERN, NPKERN, NPATK, NGEWE

COMMON /PROBLW/ XAHC, NMXCES, NTSLCP, NVALS, SMOOTH, SNEG, CRDFIT,
           EVAIC, SUBDV, PLYWOOD

LOGICAL OPLAIC, OSPLAIC, WTGEOM, WTGNMF, WTSL, WTDL, PRBOX,

COMMON /MODES/ SYM, SYMH, MTFPEW, MTFPET

COMMON /GEOMY/ COPLAN, NSUBDV, XSUBDV, NSUE2, NSUBCN, NSUR,
           B1, B1ETA, B1TAS, WAX, WLAZ, PSW,
           MXSW, MXBBW, MBBW, MWSW, MWSW, MBBSSH, MBBSSH

LOGICAL IKBR

 COMMON /TAPEO/ NFS, NMS, LS, NR, ID(20), NID, NTYPE, LRS, LWS, N,
           Iparm(10), IRR
DIMENSION Iparm(10)

COMMON /CHECKR/ DPCPR, GECPR, MODCR, AECPR, NACPR, SCNCR, GACPR

LOGICAL DPCPR, GECPR, MODCR, AECPR, NACPR, SCNCR, GACPR

COMMON /RAWF/(1250)

DIMENSION RAWF(1250)

COMMON /RAWF/(BUFF)

DIMENSION CWF(1250)

COMMON /RAWF/(BUFF)

DIMENSION ICXCDF(150), ICXCF(150), ICXCD(150), ICXCD(150)
C  DEF5L (2, NBOXES), DEFLITE(NCCLS)
C  DEF5L (2, 1000),  DEFLITE (50)
C  ALPHA (NCCLS + NSUBS), IJALP (SAME)
C  NBOXES)
C  NALP (50)
C  DEF5L (2, 1000),  DEFLITE (50)
C  ALPHA (NCCLS + NSUBS), IJALP (SAME)
C  NBOXES)
C  NALP (50)
C  DEF5L (2, 1000),  DEFLITE (50)
C  ALPHA (NCCLS + NSUBS), IJALP (SAME)
C  NBOXES)
C  NALP (50)
C  DEF5L (2, 1000),  DEFLITE (50)
C  ALPHA (NCCLS + NSUBS), IJALP (SAME)
C  NBOXES)
C  NALP (50)
C  DEF5L (2, 1000),  DEFLITE (50)
C  ALPHA (NCCLS + NSUBS), IJALP (SAME)
C  NBOXES)
C  NALP (50)
C  DEF5L (2, 1000),  DEFLITE (50)
C  ALPHA (NCCLS + NSUBS), IJALP (SAME)
C  NBOXES)
C  NALP (50)
C  DEF5L (2, 1000),  DEFLITE (50)
C  ALPHA (NCCLS + NSUBS), IJALP (SAME)
C  NBOXES)
C  NALP (50)
C  DEF5L (2, 1000),  DEFLITE (50)
C  ALPHA (NCCLS + NSUBS), IJALP (SAME)
C  NBOXES)
C  NALP (50)
C  DEF5L (2, 1000),  DEFLITE (50)
C  ALPHA (NCCLS + NSUBS), IJALP (SAME)
C  NBOXES)
C  NALP (50)
C  DEF5L (2, 1000),  DEFLITE (50)
C  ALPHA (NCCLS + NSUBS), IJALP (SAME)
C  NBOXES)
C  NALP (50)
C  DEF5L (2, 1000),  DEFLITE (50)
C  ALPHA (NCCLS + NSUBS), IJALP (SAME)
C  NBOXES)
C  NALP (50)
C  DEF5L (2, 1000),  DEFLITE (50)
C  ALPHA (NCCLS + NSUBS), IJALP (SAME)
C  NBOXES)
C  NALP (50)
C  DEF5L (2, 1000),  DEFLITE (50)
C  ALPHA (NCCLS + NSUBS), IJALP (SAME)
C  NBOXES)
C  NALP (50)
C  DEF5L (2, 1000),  DEFLITE (50)
C  ALPHA (NCCLS + NSUBS), IJALP (SAME)
C  NBOXES)
C  NALP (50)
C  DEF5L (2, 1000),  DEFLITE (50)
C  ALPHA (NCCLS + NSUBS), IJALP (SAME)
C  NBOXES)
C  NALP (50)
C  DEF5L (2, 1000),  DEFLITE (50)
C  ALPHA (NCCLS + NSUBS), IJALP (SAME)
C  NBOXES)
C  NALP (50)
C  DEF5L (2, 1000),  DEFLITE (50)
C  ALPHA (NCCLS + NSUBS), IJALP (SAME)
C  NBOXES)
C  NALP (50)
C  DEF5L (2, 1000),  DEFLITE (50)
C  ALPHA (NCCLS + NSUBS), IJALP (SAME)
C  NBOXES)
C  NALP (50)
C READ THE POINTERS FROM THE MDESC FILE.

CALL RDINIT
ITYPE = SHMIXED
IMARRY = GHIPTRM
CALL READMX(MDESC, MREAD, RANDIN, NFS, NNS, LS, NMR, 2, NID, ITYPE, 1
LRS, IPTRM, M, N, PARN, IRR)

IOLAP = IPARM(5)
MPNTRM = N
MPB = MPNTRM - 1
MPB = MNI0(MBP4, MPBT)
MPMES = MPB * MPB
IF(IRR, NE, 0) GO TO 6020

C
C REMIND MDSC

C READ BOX CODES INTO STORAGE FROM GEOMETRY SCRATCH FILE

CALL RDINIT
ITYPE = SHMIXED
IMARRY = GHIBOW
CALL READMX(MDESC, MREAD, RANDIN, NFS, NNS, LS, NMR, 150, NID, ID, ITYPE, 1
LRS, IBOW, M, N, PARN, IRR)
IF(IRR, NE, 0) GO TO 6010

NPLS = 1
IF(NSURF, ER, 1, OR, COPLAN) GO TO 10
NPLS = 2

C
C CALL RDINIT
ITYPE = SHMIXED
IMARRY = GHIBOX
CALL READMX(MDESC, MREAD, RANDIN, NFS, NNS, LS, NMR, 90, NID, ID, ITYPE, 1
LRS, IBOX, M, N, PARN, IRR)
IF(IRR, NE, 0) GO TO 6010

C
C CONTINUE

C READ THE TELOC AND FELOC ARRAYS FROM THE GEOMETRY SCRATCH FILE. THESE ARE NEEDED TO INTERPOLATE VELOCITY POTENTIALS AT BOX EDGES.

CALL RDINIT
ITYPE = SHMIXED
IMARRY = GHFELOC
CALL READMX(MDESC, MREAD, RANDIN, NFS, NNS, LS, NMR, 1, NID, ID, ITYPE, 1
LRS, FELOC, M, N, PARN, IRR)
IF(IRR, NE, 0) GO TO 6010

C
C CALL RDINIT
ITYPE = SHMIXED
IMARRY = GHTELC
CALL READMX(MDESC, MREAD, RANDIN, NFS, NNS, LS, NMR, 1, NID, ID, ITYPE, 1
LRS, TELOC, M, N, PARN, IRR)
IF(IRR, NE, 0) GO TO 6010

B202
C READ AREAS AND POINTERS FOR AREAS FROM GEOMETRY SCRATCH FILE.
THESE ARE USED IN CALCULATION OF BOX LIFTS AND GEN. FORCES.

CALL RDINT
ITYPE = SHMIXED
DIMARY = 6HALPHAS
CALL READM(IGEOC,INREAD,RANDIN,NFS,NSL,NMR,1,NID,ID,ITYPE,
1 LS,ALPHA,M,N,FARM,IIR)
IF(IIR.NE.0) GO TO 6010

CALL RDINT
ITYPE = SHMIXED
DIMARY = 6HALPH
CALL READM(IGEOC,INREAD,RANDIN,NFS,NSL,NMR,1,NID,ID,ITYPE,
1 LS,2HALPH,M,N,FARM,IIR)

HALPH = N
HALPHW = IPARM(3)
HALPF = N = HALPHW
IF(IIR.NE.0) GO TO 6010

REORDER THE FELOC AND TEDLOC ARRAYS SO THAT THERE ARE
VALUES FOR UNSUBDIVIDED CHORDS ONLY.

IF(NSUBDV.EQ.1) GO TO 120
XSLIDE = NSUBDV-1XDW
JCOL = NSUBC
NCOLZ = MBN + HTB
DO 110 I=1,NCOLS
TEDLOC(I) = (TEDLOC(JC0L)+XSLIDE)/NSUBDV
FELOC(I) = (FEL0C(JC0L)+XSLIDE)/NSUBDV
JCOL = JCOL + NSUBDV
110 CONTINUE
120 CONTINUE

LOOP ON THICKNESS SLOPE FUNCTIONS (IF NONE WERE REQUESTED,
ONE DUMMY SET OF ONES WILL HAVE BEEN GENERATED.)

REMEMBER ITSLOC
DO 750 ITSLOP=1,NTSLOP

READ THICKNESS SLOPE FUNCTIONS

CALL RDINT
ITYPE = AHREAL
DIMARY = 5TBLPH
CALL READM(ITSLOC,INREAD,RANDIN,NFS,NMS,LS,NMR,1,NID,ID,ITYPE,
1 LAST,TSBLPH,M,N,FARM,IIR)
IF(IIR.NE.0) GO TO 6040

ZERO OUT THE AIR FORCES ARRAY

DO 150 J=1,MAX
150 GENAF(J) = (0,0,0)

LOOP ON NUMBER OF NOSE SHAPES
DO 650 N=1,NMDES

B203
C GET MODE SHAPE NM FROM MODESC SCRATCH FILE
CALL R9INIT
IF(NM.EQ.1) NM = NMSPEC
ITYPE = 4HREAL
MARRAY = @HDEPSL
CALL READMX(MODESC,HREAD,RANDIN,NFS,NMG,NNS,LS,MR,2,NID,ID,ITYPE,
1 LRS,DEPSL,N,N,PARM,IRR)
IF(IRR.NE.0) GO TO 6020
C
C LOOP ON VELOCITY POTENTIALS
REWIND IVPSC
DO 600 JVP=1,NVPS
C
C READ ONE SET OF VELOCITY POTENTIALS
CALL R9INIT
ITYPE = 4HREAL
MARRAY = 3HTVP
CALL READMX(IVPSC,HREAD,RANDIN,NFS,NMG,NNS,LS,MR,2,NID,ID,ITYPE,
1 LRS,TVP,N,N,PARM,IRR)
IF(IRR.NE.0) GO TO 6030
C
CALL R9INIT
ITYPE = 4HREAL
MARRAY = 3HTVP
CALL READMX(IVPSC,HREAD,RANDIN,NFS,NMG,NNS,LS,MR,2,NID,ID,ITYPE,
1 LRS,TVP,N,N,PARM,IRR)
IF(IRR.NE.0) GO TO 6030
C
C CONDENSE THE TRAILING EDGE VELOCITY POTENTIAL ARRAY TO
C UNSUBDIVIDED BOXES.
C ALSO ZERO OUT THE SECTIONAL GENERALIZED FORCES.
C
NVPS = MBV + MBT
DO 210 I=1,NVPS
SECION(I) = (0.,0.)
210 CONTINUE
C
IF(NSUBDV.EQ.1) GO TO 220
JCOL = NSUBCN
DO 215 I=1,NVPS
TVP(I) = TVP(JCOL)
JCOL = JCOL + NSUBDV
215 CONTINUE
220 CONTINUE
C
C ZERO OUT THE BOX LIFT ARRAY
C
IF(NM.NE.1) GO TO 240
IF (.NOT. BLNEED) GO TO 240
NE = 1NPTRN(1,1NPTRS) - 1
DO 230 I=1,NE
DELCP(I) = (0.,0.)
230 CONTINUE
B204
250 DXLIFT(1) = (0.,0.) FORCES 00255
240 CONTINUE FORCES 00256
C FORCES 00257
C FORCES 00258
C
FORCES 00259
C FORCES 00260
C THE FOLLOWING BLOCK OF CODE COMPUTES FORCES 00261
C DXLIFT - BOX LIFTS FORCES 00262
C SLIFT - SECTION (SHOCK) LIFTS FORCES 00263
C TLIFT - TOTAL LIFT FORCES 00264
C GEMAF - GENERALIZED AIRFORCES FORCES 00265
C FORCES 00266
C
FORCES 00267
C APROM(JVP) = (0.,0.) FORCES 00268
C APROW = CURRENT ROW OF GENERALIZED AIRFORCES FORCES 00269
C FORCES 00270
C SET UP INITIAL CONDITIONS FOR DOUBLE LOOP OVER THE ENTIRE FORCES 00271
C BOX PATTERN FORCES 00272
C INIXDF = FORWARD FORCES 00273
C INIXCD = CENTER - ROWS OF BOX CODES, EXPANDED FORCES 00274
C INIXCD = AFF / FORCES 00275
C VPILE = VELOCITY POTENTIAL AT BOX LEADING EDGE FORCES 00276
C VPIE = ARRAY OF BOX TRAILING EDGE VELOCITY POTENTIALS FORCES 00277
C DO 565 NP=1,NPLS FORCES 00278
IF (NP.EQ.2) GO TO 245 FORCES 00279
ISROWA = I FORCES 00280
NBX = INPTRY(1,2) FORCES 00281
CALL DCODE(IHOM,150,ISROWA,1,ISROWA,NBX,F,IBXCD) FORCES 00282
NBX = NBX FORCES 00283
GO TO 290 FORCES 00284
245 CONTINUE FORCES 00285
ISROWA = (IXBY-IXBW)/NSUBIV + 1 FORCES 00286
IXBT = ISROWA FORCES 00287
IDEX = ISROWA + ICOFLAP FORCES 00288
NBX = INPTRY(1,INDEX)-IPNTRY(1,INDEX) FORCES 00289
ISUBT = 2-IXBT FORCES 00290
CALL DCODE(IHOM(ISUBT,1),LBXCDT,ISROWA,1,ISROWA,NBX,F,IBXCD) FORCES 00291
NBX = NBX FORCES 00292
250 CONTINUE FORCES 00293
DO 270 JCL = 1,NBX FORCES 00294
IBXCD(JCL) = IBXCD(JCL) FORCES 00295
IF (IBXCD(JCL).EQ.1) GO TO 260 FORCES 00296
VPIE(JCL) = XINEF FORCES 00297
DEPLIE(JCL) = XINEF FORCES 00298
GO TO 270 FORCES 00299
260 CONTINUE FORCES 00300
VPIE(JCL) = (0.,0.) FORCES 00301
IDC = JCL FORCES 00302
DEPLIE(JCL) = DEFLS(1,1,1) + DEFLS(D,1,1)@1*(JFLOCC(JCL)-1.0) FORCES 00303
270 CONTINUE FORCES 00304
C FORCES 00305
C
FORCES 00306
C LOOP ON ROWS OF THE BOX PATTERN FORCES 00307
IF(NP.EQ.2) GO TO 275 FORCES 00308
IRZ = 1 FORCES 00309
B205
NBT = NBGW
IF (ICOLPLAN) NBT = NBST
NBGW = NBT
GO TO 280
275 CONTINUE
IRS = (NBT-1)*NBV + NSUBDV + 1 + IOMAP
NBT = NBT + IOMAP
280 CONTINUE
DO 560 IROW = IRS, NBT
C
TRANSF BOX CODES TO CORRECT ARRAYS FOR NEW ROW
NBXF = NBX
NBX = NAXA
IF (NBXF .EQ. 0) GO TO 315
DO 310 JCOL = 1, NBX
310 IBXCDF(JCOL) = IBXCD(JCOL)
315 CONTINUE
IF (NBX .EQ. 0) GO TO 325
DO 320 JCOL = 1, NBX
320 IBXCD(JCOL) = IBXCD(JCOL)
325 CONTINUE
IF (IROW, EQ, NBT) GO TO 335
ISROA = ISROA + 1
IF (NP, EQ, 2) GO TO 330
NBXA = IPNTRM1(ISROA + 1) - IPNTRM1(ISROA) + IPNTRM2(ISROA) - 1
CALL DECOR (IBORM, ISROA, 1, ISROA, NBXA, F, IBXCD)
GO TO 340
330 CONTINUE
INDEX = ISROA + IOMAP
NBXA = IPNTRM1(ISROA + 1) - IPNTRM1(ISROA) + IPNTRM2(ISROA) - 1
ISUBT = 2 - ISROA
CALL DECOR (IBORM, ISROA, 1, ISROA, NBXA, F, IBXCD)
GO TO 340
335 CONTINUE
NBXA = 0
340 CONTINUE
C
ITROW = IROW
IF (NP, EQ, 2) ITROW = IROW - IOMAP
C
LOOP ON CHORDS OF THE BOX PATTERN
IF (NBX .EQ. 0) GO TO 560
DO 550 JCOL = 1, NBX
C
IF (IBXCD(JCOL) .NE. 1) GO TO 550
IDC = LCCSEW (IROW, JCOL, IPNTRM, LPNTRM, IPNTRM)
IF (IDC .EQ. 0) GO TO 970
C
GET THE SUBSCRIPT TO USE IN THE EDGE ARRAYS, JJ
JJ = JCOL
IF (NP, EQ, 2) GO TO 350
IF (.NOT. COLPLAN) GO TO 355
IF (JJ .GT. WBT) GO TO 355
IF (IDEC(JJ) .GE. FLOAT (ITROW)) GO TO 355
350 JJ = JJ + NBW
355 CONTINUE
C
DETERMINE BOX LEADING EDGE VALUES

IF (TEXLOC(JJ) .LT. FLOAT(ITROM-1) ) GO TO 410

BOX LEADING EDGE IS INTERNAL TO THE PLANFORM

VPALE = VPTE(JJ)
DEPLE = DEPLTE(JJ)
GO TO 440

BOX IS ON PLANFORM LEADING EDGE. IS IT INFLUENCED BY THE WAKE?

110 CONTINUE

IF ( JJ .EQ. JCOL .OR. NOT. COPLAN ) GO TO 420

LEADING EDGE OF SECOND PLANFORM. VELOCITY POTENTIAL COMPUTED FROM WAKE EQUATION.

XDKVL = (TEXLOC(JJ) - TEXLOC(JCOL)) * XWL
VPALE = TVP(JCOL) * COMPLX(COS(XDKVL),-SIN(XDKVL))
GO TO 425

LEADING EDGE OF SECOND PLANFORM OF SPATIAL ANALYSIS OR LEADING EDGE OF FIRST PLANFORM (WING)

420 CONTINUE

VPALE = (0.,0.)

425 CONTINUE

TEST FOR SINGLE BOX

IF (TEXLOC(JJ) .LT. FLOAT(ITROM-1) ) GO TO 430

BOX IS A SIMPLE LEADING EDGE BOX

IDA = LOC3D(WROM1,JCOL,1,PNTRM1,LPNTRM1,1,LPNTRM)
IF (IDA .EQ. 0) GO TO 970
SLOPE = BI*DEFSL(2,IDC)
XKIF = FLOAT(ITROM) - TEXLOC(JJ)
DEPLE = DEFSL(1,IDC) - SLOPE * XKIF
GO TO 450

SINGLE BOX. GET LEADING AND TRAILING VALUES

430 CONTINUE

SLOPE = BI*DEFSL(2,IDC)
DEPLE = DEFSL(1,IDC) - SLOPE * (FLOAT(ITROM) - TEXLOC(JJ))
VPTE(JCOL) = TVP(JJ)
DEPLTE(JCOL) = DEFSL(1,IDC) + SLOPE * (TEXLOC(JJ) - FLOAT(ITROM))
GO TO 500

DETERMINE BOX TRAILING EDGE VALUES

440 CONTINUE

IF (TEXLOC(JJ) .LT. FLOAT(ITROM-1) ) GO TO 460

BOX TRAILING EDGE IS INTERNAL TO THE PLANFORM

IDA = LOC3D(WROM1,JCOL,1,PNTRM1,LPNTRM1,1,LPNTRM)
450 CONTINUE

VPTE(JCOL) = .5 * (DEPH1(IDC) + DEPH1(IDA))
DEPLTE(JCOL) = .5 * (DEFSL(1,IDC) + DEFSL(1,IDA))
GO TO 500

BOX IS ON SURFACE TRAILING EDGE

460 CONTINUE

VPTE(JCOL) = TVP(JJ)
SLOPE = BI*DEFSL(2,IDC)
DEPLTE(JCOL) = DEFSL(1,IDC) - SLOPE * (TEXLOC(JJ) - FLOAT(ITROM))

C BOX LEADING AND TRAILING EDGE VALUES ARE COMPUTED. GET

B207
C  ALPHA, THE AREA MULTIPLIER
500 CONTINUE
   IF(NP.EQ.2) GO TO 505
   NNL = NNLH
   NNL = NNLH + 1
   GO TO 506
C
505 CONTINUE
   NLS = NNLH
   NNL = NNLH
506 CONTINUE
C
ALPH = 1.0
JCOMJ = JCOL+512
JCOMF = JCOM+512
DO 510 I=NLS,NNL
   IF (I(JALPH(I),LT, JCOMJ) GO TO 510
   IF (I(JALPH(I),GT, JCOMF) ) GO TO 520
   IF (I(JALPH(I),NE, JCOM+1) ) GO TO 510
   ALPH = ALPHA(I)
   GO TO 520
510 CONTINUE
520 CONTINUE
   IF (PLYWOOD) ALPH = 1.0
   IF(NN,NE,1) GO TO 530
C A RE BOX LIFTS DESIRED -
   IF (.NO. BLUEDT) GO TO 530
   BXLIFT(IDC) = (TEMP + ALPH*WTE(JCQL)-VPE)*TWSBET*TSLFN(IDC)
   DELCP(IDC) = BXLIFT(IDC)/(ALPH*B1)
530 CONTINUE
   TEMPL = DEFLTE(JCQL)VPE(JCQL) - DEFU* VPE
   TEMPLS = (B1*ALPHA*CFSL(2,IDC) + DEL.9* IDC)
   GAF = (TEMPL*CFSL(1,IDC) + TEMPL*CFSL(1,IDC) + TEMPL* CFSL(1,IDC))
   APROM(JVP) = APROM(JVP) + GAF
C SECWOM(JJ) = SECWOH(JJ) + GAF
C
590 CONTINUE
C  END OF LOOP ON CHORDS OF THE BOX PATTERN, FROM 340
590 CONTINUE
C  END OF LOOP ON ROWS OF THE BOX PATTERN, FROM 270
C
595 CONTINUE
C  END OF LOOP ON NUMBER OF PLANFORMS
C
CALL BSDMT
   I=TYPE = 7HCOMPLEX
   PHRM(1) = RVL
   PHRM(2) = B1

B208
PARM(3) = XMA
C IF THIS IS THE FIRST WEIGHTING FUNCTION, IT MAY BE NECESSARY
C TO WRITE AND/OR PRINT BOX LIFTS, ETC
C IF (.NOT. BNEED) GO TO 600
C IF (NM,NE,1) GO TO 600
C IF (.NOT. PRL) GO TO 570
C PRINT BOX LIFTS
TITL(1) = 8H WING
TITL(2) = 10MB LIFTS
TITL(3) = 2H
IF (CMPLAN) TITL(1) = 10MB TAIL
CALL PRINTBL(TITL,JVP,8L1,F1,NBOM,MYBT,IPNTRM)
IF (NSURF.EQ.1 .OR. CMPLAN) GO TO 570
TITL(1) = 8H TAIL
CALL PRINTBL(TITL,JVP,8L1,F1,NBOM,MYBT,IPNTRM(1,1CMPLAP1))
570 CONTINUE
C IF (.NOT. PRDCP) GO TO 52
C PRINT PRESSURE DIFFERENTIAL
TITL(1) = 8H WING
TITL(2) = 10MB RING DIFFERANCE
TITL(3) = 10MB TAIL
IF (CMPLAN) TITL(1) = 10MB TAIL
CALL PRINTBL(TITL,JVP,8L1,F1,NBOM,MYBT,IPNTRM)
IF (NSURF.EQ.1 .OR. CMPLAN) GO TO 572
TITL(1) = 8H TAIL
CALL PRINTBL(TITL,JVP,8L1,F1,NBOM,MYBT,IPNTRM(1,1CMPLAP1))
572 CONTINUE
C IF (.NOT. (TITL.CR.PRL)) GO TO 600
C EXPAND BOX LIFTS FOR WRITING ON TAPE. WRITEP FORMAT ONLY
C IF (NOSRT) WRITE (NTS,9999)
C INITIALIZE COUNTERS FOR PASSING OVER ARRAY BACKWARDS
C IJKL = CURRENT LOCATION IN INPUT (COMPRRESSED) ARRAY
C IJ = CURRENT LOCATION IN OUTPUT (EXPANDED) ARRAY
C IJFST = FIRST LOCATION FOR CURRENT ROW IN INPUT ARRAY
C IJPRV = FIRST LOCATION FOR PREVIOUS ROW IN INPUT ARRAY
C IRW = NBOM
C IJFST = IPNTRM(1,NBOM)
C LOCBST = IPNTRM(2,NBOM)
C IJPRV = IPNTRM(1,NBOM+1)
C IJKL = IJPRV
C IJ = NBOMES - WY + IJPRV-IJFST + LOCBST
C DO 755 I = 1, NBOMES
C RWSF(I) = 0.0,
C 755 CONTINUE
C LOOP BACK HERE ON ROWS. AND ON CHORDS WITHIN A ROW
C 980 CONTINUE
C IJ = IJ - 1
C IJKL = IJKL - 1

B209
IL = 
X-IT(IJKL) 
FORCES 00537 
RM(FI) = BL 
IF (IJKL .GT. IJST) GO TO 580 
END OF LOOP ON CHORDS WITHIN ONE ROW. STEP TO NEXT ROW 
IFPRV = IJST 
LOCPR = LOCST 
C LOOP BACK HERE ON EMPTY ROWS (COPLANAR CASE) 
582 CONTINUE 
IROW = IROW - 1 
IJ = IJ - MB 
C DETERMINE WHETHER DONE - 
IF (IROW .EQ. 0) GO TO 584 
C IS THE ROW EMPTY - 
IF (IPRTR(I,IROW) .EQ. IJPRV) GO TO 582 
IJST = IPRTR(I,IROW) 
LOCST = IPRTR(Z,IROW) 
IJ = IJ - LOCPR + IJST + LOCST 
GO TO 580 
C END OF LOOP ON ROWS. 
C********************************************************************** 
C 584 CONTINUE 
C ARE THE SECTION LiftS TO BE PRINTED OR WRITTEN - 
IF (.NOT. PRSL) GO TO 595 
C COMPUTE SECTION LiftS 
TLIFT = (0.,0.) 
TLIFT1 = (0.,0.) 
TLIFT2 = (0.,0.) 
DO 590 JCCL = 1,MB 
BL = (0.,0.) 
BL2 = (0.,0.) 
IROW = 0 
DO 587 IJ = JCCL, MBXES, MB 
IROW = IROW + 1 
IF (IROW .GE. JCCL) GO TO 586 
BL = BL + RM(FI) 
GO TO 587 
586 BL2 = BL2 + RM(FI) 
587 CONTINUE 
TLIFT1 = TLIFT1 + BL 
TLIFT2 = TLIFT2 + BL2 
SLIFT(JCCL,MBW) = BL2 
SLIFT(JCCL) = BL 
590 CONTINUE 
C IF (.NOT. PRSL) GO TO 595 
C PRINT SECTION LiftS AND TOTAL Lift 
CALL PRNSL(JVP, SLIFT, TLIFT1, TLIFT2, MBW, MBT) 
C********************************************************************** 
C 595 CONTINUE 
IF (.NOT. PECM) GO TO 599 
DO 596 J=1, MBVPS 
SECM(I) = SECM(I) + TWOBT 
596 CONTINUE 
NCM = - JVP 
CALL PRNSL(NCM, SECM, TLIFT1, TLIFT2, MBW, MBT) 
CALL PRNSL(NCM, SECM, TLIFT1, TLIFT2, MBW, MBT)
599 CONTINUE
   IF (.NOT. WTEL) GO TO 600
   WRITE BOX LIFTS ON THE BINARY OUTPUT FILE
   ITYPE = 7HCOMPLEX
   M = NKB
   K = - M
   N = NKB
   MXARY = 7HBLJFTS
   ID(2) = 10000000+IKVAL + JVP
   CALL WRITEX( NOUTP, MXWRT, RANDA, NFB, NMB, LS, NMR, LWS, K, ID, 1
   RMB, ITYPE, M, N, FAMR, IRR )
   IF (IRR .NE. 0) GO TO 922

600 CONTINUE
   STORE THE LOOP ON VELOCITY POTENTIALS
   IJ = NM
   DO 620 JVP = 1,NVP
      GENAF(IJ) = AFROWJ(JVP)*TWXSET
      GPAFC(IJ) = B33SET * REAL(GENAF(IJ))
      GPPAFC(IJ) = 0.
      IF(WVL .NE. 0.) GPPAFC(IJ) = BKSABT * AIMAG(GENAF(IJ))
   IJ = IJ + NMODES
620 CONTINUE
680 CONTINUE
   IF(.NOT.WGWF) GO TO 670
   K = NMODES
   M = NMODES
   N = NMODES
   ID(2) = IKVAL
   CALL WRITEX( NOUTP, MXWRT, RANDA, NFB, NMB, LS, NMR, LWS, K, ID, 1
   GENAF, ITYPE, M, N, FAMR, IRR )
   IF (IRR .NE. 0) GO TO 928

   ARE THE FORCES TO BE PRINTED -
670 CONTINUE
   IF (.NOT. PRGF) GO TO 700
   CALL PRINTAF(GENAF, PRGNAC, GPAFC, GPPAFC)
700 CONTINUE
   CALL UNDF(NODESC, NFAMR = 1)
750 CONTINUE
   RETURN

   DIAGNOSTICS - ALL CALL PFLUSH
   READING FROM SCRATCH FILE
8010 CONTINUE
   WRITE (INTS,9100) IGEOSC
   GO TO 950
6020 CONTINUE
   WRITE (NT6,9120) MDESC
   GO TO 930
6030 CONTINUE
   WRITE (NT6,9180) IPSPC
   GO TO 930
C
6040 CONTINUE
   WRITE (NT6,9140) ITSLSC
   GO TO 930

C     WRITING ON THE OUTPUT TAPE
922 CONTINUE
   WRITE (NT6,9220) NOUTP
   GO TO 952
928 CONTINUE
   WRITE (NT6,9280) NOUTP
   GO TO 952
C     INCORRECT DIMENSIONS READ
930 CONTINUE
   I = 1
   GO TO 932
931 I = 2
932 WRITE (NT5,9300) I
   IF(WREAD) GO TO 980
   GO TO 982
C     ERROR DETECTED READING A MATRIX
950 CONTINUE
   WRITE (NT6,9500) IRR
   IF(WREAD) GO TO 980
   GO TO 982
C     ERROR DETECTED WRITING A MATRIX
952 CONTINUE
   WRITE (NT6,9520) IRR
   IF(WWRITE) GO TO 980
   GO TO 982
C     MATRIX DESCRIPTION
960 CONTINUE
   WRITE (NT6,9600) (ID(I),I=1,10) , (ID(I),I=1,10)
   WRITE (NT6,9622) FARM,FARM
   WRITE (NT6,9614) NMR,NMR,LRS,LMS
   GO TO 964
962 WRITE (NT6,9620) ID(I),ID(I)
   WRITE (NT6,9622) FARM,FARM
   WRITE (NT6,9624) NFS,NMS
964 WRITE (NT6,9640) ITYPE,M,N
   WRITE (NT6,9650) MARRAY
   GO TO 990
970 CONTINUE
   WRITE (NT6,9700) IROM, JCOL
   GO TO 990
C
990 CONTINUE
   WRITE (NT6,9900)
C
   CALL FLUSH(1)
C
C

DIAGNOSTIC FORMATS

9100 FORMAT(5X,'*** ERROR WHILE READING GEOMETRY SCRATCH FILE ',A10,  
1            4H *** )
9120 FORMAT(44H,'*** ERROR WHILE READING NODES SCRATCH FILE ',A10,  
1            4H *** )
9140 FORMAT(54H,'*** ERROR WHILE READING THICKNESS SLOPE SCRATCH FILE'  
1            4H *** )
9160 FORMAT(51H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH  
1            4H *** )
9180 FORMAT(65H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9200 FORMAT(55H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9220 FORMAT(49H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9240 FORMAT(59H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9260 FORMAT(63H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9280 FORMAT(67H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9300 FORMAT(65H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9320 FORMAT(69H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9340 FORMAT(71H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9360 FORMAT(73H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9380 FORMAT(75H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9400 FORMAT(77H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9420 FORMAT(79H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9440 FORMAT(81H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9460 FORMAT(83H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9480 FORMAT(85H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9500 FORMAT(87H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9520 FORMAT(89H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9540 FORMAT(91H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9560 FORMAT(93H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9580 FORMAT(95H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9600 FORMAT(97H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9620 FORMAT(99H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9640 FORMAT(101H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9660 FORMAT(103H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9680 FORMAT(105H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9700 FORMAT(107H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9720 FORMAT(109H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9740 FORMAT(111H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9760 FORMAT(113H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9780 FORMAT(115H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9800 FORMAT(117H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9820 FORMAT(119H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9840 FORMAT(121H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9860 FORMAT(123H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9880 FORMAT(125H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9900 FORMAT(127H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9920 FORMAT(129H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9940 FORMAT(131H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9960 FORMAT(133H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )
9980 FORMAT(135H,'*** ERROR WHILE READING VELOCITY POTENTIAL SCRATCH'  
1            4H *** )

C

END

B213
SUBROUTINE DCODE(BOX, LBOX, IA, JA, IL, JL, SUBD, ICODE)

C **ARRAY OF BOX CODES IN PACKED WORD FORMAT**
C **DIMENSION OF BOX CODES ARRAY**
C **INDEX OF FIRST CODE TO RETRIEVE**
C **LAST BOX CODE ON THE IA-TH CHORD TO RETRIEVE**
C **LAST BOX ON THE IA-TH ROW TO RETRIEVE**
C **-... SUBDIVIDED BOX CODES DESIRED, .F. UNSUBDIVIDED.**
C **ARRAY INTO WHICH BOX CODE WILL BE STORED.**

C **COMMON**
C **COPIAN,NSUBD,NSUBD,NSUBC,NSURF,**
C **BI,BB,AB,MS,BB,LM,LL,LX,XX,**
C **IXAX,IXAX,IXAX,**
C **LOGICAL COPLAN**
C **LOGICAL SUBD**
C **DATA MEDAD /20/**
C **MASK = 7**
C **SUBD = 1**
C **GO TO 50**
C **IF (SUBD) GO TO 50**
C **I = NSUBD * (IA-1) + IXBW**
C **J = NSUBD * (JA-1) + NSUBCN**
C **ISKIP = NSUBD**
C **IEOD = NSUBD * (IL-1) + IXBW**
C **JEOD = NSUBD * (JL-1) + NSUBCN**
C **GO TO 60**
C **CONTINUE**
C **I = IA**
C **J = JA**
C **ISKIP = 1**
C **IEOD = IL**
C **JEOD = JL**
C **CONTINUE**
C **SUBD = 1**
C **GO TO 50**
C **SUBD = 0**
C **END**
C **CONTINUE**
C **SUBD = 0**
C **RETURN**

C **COMMON /COMMON/ COPLAN,NSUBD,NSUBD,NSUBD,NSUBC,NSURF,**
C **BI,BB,AB,MS,BB,LM,LL,LX,XX,**
C **IXAX,IXAX,IXAX,**
C **LOGICAL COPLAN**
C **LOGICAL SUBD**
C **DATA MEDAD /20/**
C **MASK = 7**
C **SUBD = 1**
C **GO TO 50**
C **IF (SUBD) GO TO 50**
C **I = NSUBD * (IA-1) + IXBW**
C **J = NSUBD * (JA-1) + NSUBCN**
C **ISKIP = NSUBD**
C **IEOD = NSUBD * (IL-1) + IXBW**
C **JEOD = NSUBD * (JL-1) + NSUBCN**
C **GO TO 60**
C **CONTINUE**
C **I = IA**
C **J = JA**
C **ISKIP = 1**
C **IEOD = IL**
C **JEOD = JL**
C **CONTINUE**
C **SUBD = 1**
C **GO TO 50**
C **SUBD = 0**
C **END**
C **CONTINUE**
C **SUBD = 0**
C **RETURN**
IB = IB + 1
1000 CONTINUE
GO TO 3000
C PROGRAM WILL RETRIEVE NJ BOXES FROM CHORD J
1100 CONTINUE
JSB = (J-1)/NBWRD + 1
JB = (NBWRD - MOD(J,NBWRD)) * 3
IF(JB.EQ.60) JB = 0
IJMASK = SHIFT(MASK,JB)
NB = -JB
DO 2000 II = 1,END,ISKIP
IJWORD = IBOM(II,JSB)
IJCODE = IJWORD.AND.IJMASK
ICODE(IB) = SHIFT(IJCODE,NB)
IB = IB + 1
2000 CONTINUE
C
3000 CONTINUE
RETURN
END
SUBROUTINE PRNTBL(TITL, IMODE, ARRAY, IX, MYB, MYB, IPNTRM)
C
PRINTS BOX LIFTS, USES /WBUF/ FOR INTERMEDIATE SCRATCH
C
IMODE - MODE SHAPE NUMBER
C
ARRAY - ARRAY TO BE PRINTED
C
IPNTRM - POINTER ARRAY FOR ROWS IN ARRAY-
C
NPNTS - NUMBER OF POINTERS
C

COMPLEX ARRAY(1)
DIMENSION TITL(3)
DIMENSION IPNTRM(2,50)
C
COMMON /CTRL/ PREXEV,GMACH, TITLE(8), PRVGEOM, PRVMODE, DIMH, DIMT, CONTrl 00002
1 DEFAUlT
LOGICAL PRVGEOM, PRVMODE, DIMH, DIMT, DEFAULT
COMMON /PROBLM/ XMACH, NMODES, NTSLF, NVALS, SMOOTH, NDEC, CRDFIT, 00002
1 EXAIc, SUBDV, PLYWOOD
LOGICAL SMOOTH, CRDFIT, EXAIc, SUBDV, PLYWOOD
COMMON /KVAL / IKVAL, IKVAL(20), IXS(20)
COMMON /FILES / NT5, NT6, INTAPE, INFSF, NSPAIC, NOUTP, 00002
1 IOUFSF, MODESC, IVPSC, IGEOSC, IWFFSC, IAICS C
C
INTEGER PAGE
DIMENSION S(1), D(1)
EQUIVALENCE (S, BUFF), (D, BUFF(1251))
C
COMplex TLift
COMplex TLIFT1, TLIFT2
DIMENSION PC(2), IPII(2)
EQUIVALENCE (IPNTR, TLIFT)
COMMON /WBUF/ BFCCDE, BFCNT, BUFF(3280)
DATA PC / 10HPAGE CONTI,4HNUED /
DATA BLANK / 1M /
DATA XINIT / -1.0 /
DATA LINEMK / 50 /
C
IXS = IXVAL(IKVAL)
IF(IXS(IKVAL).NE.XINIT) IXS = IXS(IKVAL)
PAGE = 0
N = 1
M = 4
IF(M.GT.WD) M = MYB
C
100 LINE = 100
200 DO 1400 I=1,IXB,IXB
   DO 300 J=N,M
      S(J) = 0.0
      D(J) = 0.0
 300 CONTINUE
   IF(LINE.LE.50) GO TO 900
   PAGE = PAGE + 1
   LINE = 4
   WRITE (NT6,9001) TITLE, TITL, XMACH, IXS, IMODE
   IF(PAGE.EQ.1) GO TO 700
   WRITE (NT6,9005) PC
   GO TO 800
700 WRITE (NT6,9005)

8216
500 CONTINUE
WRITE(NT6,6005) (BLANK, J=1,N, M)
WRITE(NT6,6007) (BLANK, J=1,N, M)
C
900 CONTINUE
JS = IPTRM(2,1)
IF(JS.LT.0) GO TO 1400
IDX = IPTRM(1,1)
JE = IPTRM(1,1+1) - IDX + JS-
IF(JE.EQ.0) GO TO 1400
DO 1000 J=JS,JE
B(J) = REAL(ARRAY(IDX))
D(J) = AIMAG(ARRAY(IDX))
IDX = IDX + 1
1000 CONTINUE
DO 1200 J=1,N
IF(B(J)) 1300,1100,1300
1100 CONTINUE
IF(D(J)) 1300,1200,1300
1200 CONTINUE
GO TO 1400
1300 WRITE(NT6,9013) 1, (S(J), D(J), J=1,N, M)
LINE = LINE + 1
1400 CONTINUE
C
N = N+4
N = N+4
IF(N.GT.MB) GO TO 1500
IF(N.GT.MB) M = MB
IF(LINE.GT.45) GO TO 100
WRITE(NT6,6005) (BLANK, J=1,N, M)
WRITE(NT6,6007) (BLANK, J=1,N, M)
LINE = LINE+3
GO TO 200
1500 CONTINUE
RETURN
9001 FORMAT(1H1,20X,8A10,1/50X,3A10,1/ 46X,7H(MACH FS,3,5X,15SHRED,FREQ. 1 (=M,F0.3, 0 ) = 52X,4MODE SHAPE=, 13 ))
9005 FORMAT(44X,32(1H1),20X,A10,A4 )
9006 FORMAT(4H(ORD,4.1,14X,SHCORD,13,3(A1,22X,SHCORD,13))
9007 FORMAT(5X, 4(A1,3X,4HREAL,8X,SHIAGINARY ) )
9013 FORMAT(4,8E16.8)
END
SUBROUTINE PRNTSL(IIMODE,SLIFT,TLIFT1,TLIFT2,TWBW,WYBT)
   C
   C PRINTS THE SECTION LIFTS AND TOTAL LIFTS
   C
   C IMODE - MODE SHAPE NUMBER
   C SLIFT - SECTION LIFT ARRAY
   C TLIFT1 - WING TOTAL LIFT
   C TLIFT2 - TAIL TOTAL LIFT
   C
   IF IMODE IS NEGATIVE THE PROGRAM WILL OUTPUT SECTION MOMENTS
   C
   COMMON /PROBLM/ Xmach,IMODE,NTSLOF,NKVALS,SMOOTH,NDEG,CREDFIT,
                  
                   1 EXAIC,SMOOTH,PLYWOOD
   LOGICAL SMOOTH,CREDFIT,EXAIC,SMOOTH,PLYWOOD
   COMMON /CONRL/ PREVEX,XMACH,TITLE(8),PRVGEOM,PRMODE,DIHW,DIHT,
                    
                    1 DEFAULT
   LOGICAL PRVGEOM,PRMODE,DIHW,DIHT,DEFAULT
   COMMON /KVAL/ /IKV(20),KXS(20)
   COMMON /FILES/ /NTS,NT6,INTAPE,INSP,NETAIC,NEIC,NOUTP,
                  
                  1 IOUFS,MODESC,IVPSC,IGEOSC,IFWFS,IAICSC
   C
   COMPLEX SLIFT(I)
   COMPLEX TLIFT1,TLIFT2,TLIFT
   LOGICAL PRCH
   DATA BLANK/1H/, XINIT /-1.0 /
   C
   IF (IMODE.LT.0) GO TO 100
   PRCH = .FALSE.
   GO TO 200
   100 CONTINUE
   PRCH = .TRUE.
   IMODE = -IMODE
   200 CONTINUE
   C
   XVAL = XKVAL(IKVAL)
   IF (XKVAL .NE. XINIT) XVAL = XKVAL
   IF (PRCH) GO TO 300
   WRITE (NTS,6010) XMACH,XVAL,IMODE
   GO TO 400
   300 CONTINUE
   WRITE (NTS,6010) XMACH,XVAL,IMODE
   400 CONTINUE
   WRITE (NTS,6008)
   WRITE (NTS,6005)
   WRITE (NTS,6020)
   WRITE (NTS,6007) BLANK, BLANK, BLANK, BLANK
   DO 600 I=1,WYBT,4
       11 = I + 3
   IF (11 .GT. WYBT) 11 = WYBT
   WRITE (NTS,6030) I,(SLIFT(I),I=1,II)
   600 CONTINUE
   IF (PRCH) GO TO 650
   WRITE (NTS,6023) TLIFT1
   650 CONTINUE
   IF (WYBT.EQ.0) GO TO 800
   C

B218
WRITE (NT6,6609)
GO TO 675
650 CONTINUE
   IF(WYBT.EQ.0) GO TO 800
WRITE(NT6,6609)
675 CONTINUE
   WRITE (NT6,6620)
   WRITE (NT6,6607) BLANK,BLANK,BLANK
   DO 700 I=1,WYBT,4
   II = I + 3
   IF (II.GT. WYBT) II = WYBT
   I2 = I + WYB
   II2 = II + WYB
   WRITE (NT6,6620) I,SLIFT(IJ),IJ=I2,II2)
700 CONTINUE
   C
   IF(PFNM) GO TO 800
   WRITE (NT6,6624) TLIFT2
   TLIFT = TLIFT1 + TLIFT2
   WRITE (NT6,6625) TLIFT
800 CONTINUE
   RETURN
   C
605 FORMAT(44X,32(1H-),30X,A10,A4)
607 FORMAT(3X,4(1X,9X,4HREAL,8X,9HIMAGINARY)
608 FORMAT(58X,4HNGO)
609 FORMAT(1HD,// 55X,SECTION LIFT= / 58X,ETAILE,/, 44X,32(1H-) / )
610 FORMAT(1HD,52X,14HSECTION LIFTS /44X,*(1H,MACH=,5.3,5X,
1 4RED. FREQ. =# F8.5,* )/ / 52X,MODE SHAPE=,13)
620 FORMAT(80D4)
621 FORMAT(1HD,44X,0 TOTAL LIFT = WING= / 1HD 40X,2E16.8 )
622 FORMAT(1HD,44X,0 TOTAL LIFT = TAIL= / 1HD 40X,2E16.8 )
623 FORMAT(1HD,53X,0 TOTAL LIFT = 1HD 40X,2E16.8 )
624 FORMAT(14,8E16.8)
625 FORMAT(1HD,45X,SECTIONAL MOMENT COEFFICIENTS#58X,ETAILE/
1 44X,32(1H-) / )
626 FORMAT(1HD,45X,SECTIONAL MOMENT COEFFICIENTS#44X,*(1H,MACH=,
1 5.3,5X,RED. FREQ. =# F8.5,* )/ / 52X,MODE SHAPE=,13)
END
SUBROUTINE PRINTAF(ARRAY, PRNGAC, GPAFC, GPPAFC)
C       PRINTS COMPLEX GENERALIZED AIR FORCES, FROM COMPACT FORTRAN
C       STORAGE
C       ARRAY - ARRAY OF GENERALIZED AIR FORCES
C       PRNGAC - LOGICAL FLAG FOR PRINT OPTION
C       GPAFC - AGARD GENERALIZED AERODYNAMIC COEFFICIENT
C       GPPAFC - AGARD GENERALIZED AERODYNAMIC COEFFICIENT
C
C       COMPLEX ARRAY(1)
DIMENSION GPAFC(1), GPPAFC(1)
LOGICAL PRNGAC
C
DIMENSION PC(2)
COMMON /PRBLMN/ XMACH, NMODES, NTSLOP, NVALS, SMOOTH, NDEG, CRDFIT,
               EXAGIC, SUBDIV, PLYWOOD
LOGICAL SMOOTH, CRDFIT, EXAGIC, SUBDIV, PLYWOOD
COMMON /KVAL/ IKVAL, WKS(20), KS(20)
COMMON /FILES/ NT5, NT6, INTA'4, INSP, FILEP, FILEP, ILAICSC
COMMON /VRCBLW/ X4AIRCES, T5LO, NVALS, SI00T1, IMEG, CR0, INTCR,
                KVAL 0002
COMMC /FILES /
DATA /1H/ 100
DATA /-1.0/ 110
*2 = WVAL(IKVAL)
LINE: = LINE: + 10
100 LINE = LINEX + 10
110 DO 201 I = 1, NMODES
    IF (LINE .LE. LINEX) GO TO 170
    PAGE = PAGE + 1
    LINE = 0
    WRITE (NT6, 6001) XMACH, WXML
    IF (PAGE .EQ. 1) GO TO 190
    WRITE (NT6, 6005) PC
    GO TO 160
190 WRITE (NT6, 6005)
160 CONTINUE
C
    WRITE (NT6, 6005) (BLANK, J, J=J1, J2)
    WRITE (NT6, 6007) (BLANK, J=J1, J2)
C
170 CONTINUE
    WRITE (NT6, 6010) I, (ARRAY(IJ), IJ=J1, J2, NMODES)
    LINE = LINE + 1
    I1 = I1 + 1
    I2 = I2 + 1
200 CONTINUE
C
    J1 = J1 + 4
B220
J2 = J2 + 4
IF (J1 .GT. NODES) GO TO 300
IF (J2 .GT. NODES) J2 = NODES
J1 = J1 + 3*MODES
J2 = J1 + (J2-J1) * NODES
IF (LINE .GT. LINEX - 6) GO TO 500
WRITE (*,10000) (BLANK, J = J1,J2)
LINE = LINE+3
GO TO 110
C
300 CONTINUE
C
PRINT THE GENERALIZED AERODYNAMIC COEFFICIENTS
C
IF DESIRED.
C
IF (.NOT. DLDMAC) GO TO 1400
PAGE = 0
DO 1300 IP = 1,2
J1 = 1
J2 = 8
IF (NODES .LT. J2) J2 = NODES
J1 = 1
J2 = (J2-1) * NODES + 1
C
1100 LINE = LINEX + 10
1110 DO 1200 I=1,NODES
IF (LINE .LE. LINEX) GO TO 1170
PAGE = PAGE + 1
LINE = 8
WRITE (*,10000) XMACH, XVL
IF (IP .EQ. 2) GO TO 1140
IF (PAGE .EQ. 1) GO TO 1130
WRITE (*,10000) PC
GO TO 1140
1130 CONTINUE
WRITE (*,10000)
GO TO 1140
1140 CONTINUE
IF (PAGE .EQ. 1) GO TO 1130
WRITE (*,10000) PC
GO TO 1140
1130 WRITE (*,10000)
1140 CONTINUE
C
WRITE (*,10000) XMACH, XVL
WRITE (*,10000) PC
C
1170 CONTINUE
IF (IP .EQ. 2) GO TO 1180
WRITE (*,10000) I, (GPAPC(IJ), IJ=J1,J2,NODES)
GO TO 1180
1180 CONTINUE
WRITE (*,10000) I, (GPAPC(IJ), IJ=J1,J2,NODES)
1180 CONTINUE
LINE = LINE + 1
J1 = J1 + 1
J2 = J2 + 1

B221
1200 CONTINUE
C
J1 = J1 + 8
J2 = J2 + 8
IF(J1.GT.NMODES) GO TO 1300
IF(J2.GT.NMODES) J2 = NMODES
J1 = J1 + 3*NMODES
J2 = J1 + (J2-J1)*NMODES
IF (LINE.GT.LINE+3) GO TO 1100
WRITE (NE,7006)
LINE = LINE + 3
GO TO 1110
C
1300 CONTINUE
1400 CONTINUE
RETURN
C
6001 FORMAT(1H1,5lx,18l'GENERALIZED FORCES /44X,6H(MACH ,F5.3,5X,4RED. F
1REQ. = B8.5,1H )
6005 FORMAT(35X,50(1H-,20X,A10,44)
6006 FORMAT(5D0 WT. / 6H FUNCT. 4(A2,23HVELOCITY POTENTIAL MODE,13,3X))
6007 FORMAT(2X,4(A10,4HREAL,4X,9H(IMAGINARY) )
610 FORMAT(4,8E16.8)
7001 FORMAT(1H1,44X,'GENERALIZED AERODYNAMIC COEFFICIENTS*/, /1
1 44X,6H(MACH ,F5.3,5X,12HRED. FREQ. = B8.5,1H )
7006 FORMAT(5D0 WT. / 6H FUNCT.27X,6VELOCITY POTENTIAL MODES*/ )
7007 FORMAT(13,7(16)
7005 FORMAT(50X,REAL PART*/,35X,50(1H-,20X,A10,44)
7015 FORMAT(50X,*IMAGINARY PART*/35X,50(1H-,20X,A10,44)
END
FUNCTION LOCSDW(IRON, JCOL, IPNTS, IPNTIN, IPNTO, IPNTLS)

C RETURNS THE LOCATION OF THE WORD IN THE END-AROUND SUBDIVIDED
C DOMAIND ARRAY CORRESPONDING TO BOX(IRON, JCOL) OF THE SUB-
C DIVIDED BOX ARRAY
C
C IRON = BOX CHORDWISE LOCATION
C JCOL = BOX SPANWISE LOCATION
C IPNTS = ARRAY OF POINTERS
C IPNTIN = NEXT AVAILABLE (UNUSED) CELL IN IPNTS (END-
C AROUND)
C IPNTO = FIRST CURRENTLY AVAILABLE CELL IN IPNTS
C IPNTLS = LAST CELL OF IPNTS (LENGTH OF ARRAY)
C
C RETURN -
C LOCSDW = LOCATION OF DESIRED DOMAIND, IF SUCCESSFUL
C = 0, IF LOCSDW LIES OUTSIDE THE DEFINED AREA.
C DIMENSION IPNTS(2, IPNTLS)
C
C LOCIN = MOD(IRON-1, IPNTLS) + 1
C LOCIN = LOCATION OF CELL IN IPNTS WHICH WAS OR IS TO BE LOCSDW
C USED
C 1F(IPNTIN = IPNTO) 100, 300, 200
C END AROUND HAS OCCURRED
C 100 1F (LOCIN = IPNTIN 400, 300, 150)
C NOT IN UPPER PART, IS LOCIN WITHIN BOTTOM PART -
C 150 1F (LOCIN - IPNTO) 300, 400, 400
C
C 200 1F (LOCIN - IPNTO 250, 300, 300
C NO END AROUND, NORMAL SEQUENCE
C 250 1F (LOCIN .GE. IPNTO) GO TO 400
C
C 300 1F (LOCIN .LT. IPNTO) GO TO 300
C ERROR OR INITIAL CONDITION ENCOUNTERED (SHOULD NEVER OCCUR)
C 350 LOCSDW = 0
C GO TO 500
C
C 400 1FB = IPNTS(2, LOCIN)
C 1F (JCOL.LT.1FB) GO TO 300
C LOCSDW = IPNTS(1, LOCIN) + JCOL-1FB
C
C 500 CONTINUE
C RETURN
C END

B223
The Mach box technique has been extended to include wing and tail with dihedral angles and vertical separation. A digital computer program, written in FORTRAN, is presented. The program provides for up to nine sweep angles of the leading and trailing edges of each surface. First order piston theory thickness correction is available as an option and two refinement procedures are provided, subdivision with averaging and velocity potential smoothing. For a maximum of twenty oscillatory mode shapes the program calculates normal washes, velocity potentials, lifts, pressures and generalized forces matrices. If only one surface is being analyzed, sampling of wake up-wash, side-wash and longitudinal wash is available.

The methods described in this report are intended to be used by airplane designers to calculate with improved accuracy, the unsteady aerodynamic loads that act on a lifting surface being propelled at supersonic speeds. The new feature of these calculations is that the aerodynamic interference between the wing and tail has been taken into account. These calculations are an essential ingredient of flutter analyses and will improve the confidence level of such calculations in preventing wing-tail flutter. The general requirement for such calculations are contained in Military Specification MIL-A-8870A (USAF).
<table>
<thead>
<tr>
<th>KEY WORDS</th>
<th>LINK A</th>
<th>LINK B</th>
<th>LINK C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsteady Aerodynamics</td>
<td>ROLE</td>
<td>WT</td>
<td>ROLE</td>
</tr>
<tr>
<td>Aeroelasticity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flutter</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supersonic Flow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aerodynamic Interference</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>