THE POWER INDUCED EFFECTS MODULE:
A FORTRAN CODE WHICH ESTIMATES LIFT INCREMENTS DUE TO
POWER INDUCED EFFECTS FOR V/STOL FLIGHT

GRANTOR - NASA AMES RESEARCH CENTER
GRANTEE - CAL POLY STATE UNIVERSITY
GRANT NUMBER NCC 2-664
1/1/90 - 9/30/91

Principal Investigator
Dr. Doral R. Sandlin

Student Investigator
Kipp E. Howard

Cal Poly State University
San Luis Obispo, CA 93407

July 1991
ABSTRACT

THE POWER INDUCED EFFECTS MODULE:
A FORTRAN CODE WHICH ESTIMATES LIFT INCREMENTS DUE TO
POWER INDUCED EFFECTS FOR V/STOL FLIGHT

Kipp E. Howard

The objective of this project was to produce a user friendly FORTRAN code which can be used for preliminary design of V/STOL aircraft. It estimates lift increments, due to power induced effects, encountered by aircraft in V/STOL flight. These lift increments are calculated using empirical relations developed from wind tunnel tests and are due to suckdown, fountain, ground vortex, jet wake, and the reaction control system. The code can be used as a preliminary design tool along with NASA Ames' Aircraft Synthesis preliminary design code or as a stand-alone program for V/STOL aircraft designers.

The Power Induce Effects Module was validated using experimental data supplied by NASA-Ames Research Center, McDonnell Aircraft Company and data computed from lift increment routines developed by Richard E. Kuhn. Results are presented for many flat plate models along with the McDonnell Aircraft Company's MFVT V/STOL preliminary design and a 15% scale model of the YAV-8B, "Harrier", V/STOL aircraft.

It was found the Power Induced Effects Module predicts trends and magnitudes of lift increments verses aircraft height above the ground, well. The code was also found to predict only the magnitudes of lift increments verses aircraft forward velocity, well. More experimental results are needed to determine how well the code predicts lift increments as they vary with jet deflection angle and angle of attack.
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF FIGURES</th>
<th>vi</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOMENCLATURE</td>
<td>viii</td>
</tr>
</tbody>
</table>

CHAPTER 1

Introduction ........................................................... 1

CHAPTER 2

Lift Increment Descriptions ........................................ 3
- Suckdown ................................................................... 3
- Fountain Lift .......................................................... 4
  - Fountain Arms ...................................................... 5
  - Fountain Core ...................................................... 7
- Lift Improvement Devices ......................................... 7
- Ground Vortex ........................................................ 7
- Jet Wake ............................................................... 8
- Reaction Control System ......................................... 10
- Trends and Comparisons of the Lift Increments .......... 12

CHAPTER 3

Computational Development ........................................ 13
- Variables .............................................................. 13
  - Independent Variables ......................................... 13
  - Configuration Variables ....................................... 13
  - Control and Result Variables ................................. 14
- Hover ...................................................................... 14
  - Suckdown ........................................................... 15
  - Fountain Lift ...................................................... 16
  - Lift Improvement Devices ........................................ 18
- Forward Flight / Jet Deflection Angle / Angle of Attack 19
  - Suckdown/Fountain ............................................. 19
  - Ground Vortex .................................................... 20
  - Jet Wake ............................................................ 21
  - Reaction Control System ....................................... 22

CHAPTER 4

Computer Code Description .......................................... 24
- Philosophy ............................................................ 24
- Control ................................................................. 24
- Input ......................................................................... 26
- Preliminary Calculations ........................................... 26
Lift Increments Routines .............................................................. 27
  Height .................................................................................. 27
  Forward Velocity .................................................................. 28
  Height, Forward Velocity, and Jet Deflection Angle ............... 28
  Height, Forward Velocity, Jet Deflection Angle, and Angle of
  Attack .................................................................................. 29
Output ........................................................................................ 29
Limitations .................................................................................. 30

CHAPTER 5
Results and Discussion/Validation .............................................. 31
Configurations ............................................................................. 32
  Flat Plate Models .................................................................. 33
  Mixed-Flow Vectored-Thrust Configuration ................................ 33
  YAV-8B ........................................................................... 33
Hover .......................................................................................... 34
  Disk .................................................................................. 35
  Body/Wing-Body .................................................................. 35
  Delta .................................................................................. 38
  Mixed Flow Vectored Thrust ................................................. 39
  YAV-8B ........................................................................... 40
Forward Flight ............................................................................... 41
  MFVT ........................................................................... 42
  YAV-8B ........................................................................... 45

CHAPTER 6
Conclusions and Recommendations ............................................ 47

BIBLIOGRAPHY ............................................................................ 49

APPENDIX A
PIE FORTRAN Code ....................................................................... 51
  Coppie .................................................................................. 51
  Inpie .................................................................................. 64
  Planmo .............................................................................. 74
  Finvar .................................................................................. 82
  Sdarea ............................................................................... 104
  Integra .............................................................................. 130
  Diabar ............................................................................... 130
  Hcall ............................................................................... 140
  Hov_ge ............................................................................ 145
  Sfcall ............................................................................... 155
  Rscal ............................................................................... 162
  Rcsind ............................................................................... 164
  Gvcall ............................................................................... 167
  Stolgv ............................................................................... 171
  Wkcall ............................................................................... 173
  Jiepie ............................................................................... 183
  Stolwk ............................................................................ 188
  Piep ............................................................................... 190
  Dist ............................................................................... 210
  Perpdis ............................................................................. 211
  Linterp ............................................................................. 211
  Crossin ............................................................................. 211
<table>
<thead>
<tr>
<th>Software</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linecro</td>
<td>212</td>
</tr>
<tr>
<td>Polyar</td>
<td>213</td>
</tr>
<tr>
<td>Centar</td>
<td>214</td>
</tr>
<tr>
<td>Ssen</td>
<td>217</td>
</tr>
<tr>
<td>Jietab</td>
<td>220</td>
</tr>
</tbody>
</table>

APPENDIX B
Power Induced Effects Module User’s/Programmer’s Manuals... 223
LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure #</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>Suckdown a) Out-of Ground Effect and b) In-Ground Effect</td>
<td>4</td>
</tr>
<tr>
<td>#2</td>
<td>Photograph of a jet in a water tunnel illustrating the entrainment of air from the calm surroundings around the edge of the plate. (Reference 11).</td>
<td>4</td>
</tr>
<tr>
<td>#3</td>
<td>Photograph of the flow field between two jets with the same nozzle pressure (Reference 11).</td>
<td>5</td>
</tr>
<tr>
<td>#4</td>
<td>Planform view of jet pattern with four jets and the flow from each of the jets</td>
<td>6</td>
</tr>
<tr>
<td>#5</td>
<td>“Fountain Core and Arms” generated by a configuration with three or more jets - Side view of jet pattern (Reference 6).</td>
<td>6</td>
</tr>
<tr>
<td>#6</td>
<td>Flow field under an aircraft in hover with lift improvement devices</td>
<td>7</td>
</tr>
<tr>
<td>#7</td>
<td>Ground vortex created from the interaction between the freestream and the jets of a V/STOL aircraft near the ground</td>
<td>8</td>
</tr>
<tr>
<td>#8</td>
<td>Photograph of the cross section of the wake six nozzle diameters downstream of a jet exiting perpendicular to the free-stream in a water tunnel. (Reference 11).</td>
<td>9</td>
</tr>
<tr>
<td>#9</td>
<td>Pressure-coefficient contours on a flat plate with a perpendicular exiting jet. (Reference 14).</td>
<td>10</td>
</tr>
<tr>
<td>#10</td>
<td>Roll control reaction control system nozzles and the flow field around the wing</td>
<td>11</td>
</tr>
<tr>
<td>#11</td>
<td>V/STOL induced lift increments (Reference 8).</td>
<td>11</td>
</tr>
<tr>
<td>#12</td>
<td>Fountain variables from a configuration with three jets</td>
<td>17</td>
</tr>
<tr>
<td>#13</td>
<td>Typical variation of fountain lift increment using h’ Method</td>
<td>18</td>
</tr>
<tr>
<td>#14</td>
<td>Lift Improvement Device definitions from a triangular configuration with three jets.</td>
<td>19</td>
</tr>
<tr>
<td>#15</td>
<td>Flow chart of the control program COPPIE. (Note - Program Flow is downward unless specified.)</td>
<td>25</td>
</tr>
</tbody>
</table>
Figure #16. Half-planforms of configurations used for verification.........................32
Figure #17. Comparison of calculated and measured total lift increments in hover for a 20-inch circular disk (Reference 15)..........................................................35
Figure #18. Comparison of calculated and measured lift increments in hover for the Body planform (Reference 15).................................................................36
Figure #19. Comparison of calculated and measured lift increments in hover for the Wing-Body planform (Reference 15)...............................................................37
Figure #20. Comparison of calculated and measured lift increments in hover for the Delta planform (Reference 15).................................................................39
Figure #21. Comparison of calculated lift increments in hover between Kuhn’s program and PIE for the MFVT, triangular 3-jet configuration (Reference 7)..................................................................40
Figure #22. Comparison of calculated and measured total lift increments in hover for 15% scale model of the YAV-8B (Reference 9)......................................................41
Figure #23. Comparison of calculated lift increments in forward flight at an altitude of 6 ft. for the MFVT, lateral 2-jet configuration from Kuhn’s program and PIE (Reference 7).........................................................42
Figure #24. Comparison of calculated lift increments in forward flight at an altitude of 6 ft. for the MFVT, triangular 3-jet configuration from Kuhn’s program and PIE (Reference 7).........................................................44
Figure #25. Comparison of calculated and measured total lift increments in forward flight for a 15% scale model of the YAV-8B (Reference 9)..................46
### NOMENCLATURE

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Total jet exit area</td>
</tr>
<tr>
<td>ACSYNT</td>
<td>NASA-Ames Research Center’s Aircraft Synthesis preliminary design code</td>
</tr>
<tr>
<td>AOA</td>
<td>Angle of attack of the aircraft with respect to the freestream</td>
</tr>
<tr>
<td>COPPIE</td>
<td>Control program for PIE</td>
</tr>
<tr>
<td>Cp</td>
<td>Coefficient of pressure</td>
</tr>
<tr>
<td>ΔCp</td>
<td>Constant in expression for the contribution of the ground vortex</td>
</tr>
<tr>
<td>d</td>
<td>Diameter of single nozzle</td>
</tr>
<tr>
<td>D</td>
<td>Angular mean diameter of the planform</td>
</tr>
<tr>
<td>d_e</td>
<td>Effective jet exit diameter (the diameter of a single jet which would have the same exit area as the sum of all the jets.)</td>
</tr>
<tr>
<td>DFL</td>
<td>Jet deflection angle with respect to the fuselage</td>
</tr>
<tr>
<td>e</td>
<td>Half the distance between adjacent jets (See Figure #12)</td>
</tr>
<tr>
<td>Exp</td>
<td>Experimental results</td>
</tr>
<tr>
<td>FORTRAN</td>
<td>Formula translation - computer programming language</td>
</tr>
<tr>
<td>GE</td>
<td>In-ground effect</td>
</tr>
<tr>
<td>h</td>
<td>Height of aircraft above the ground (See Figure #12)</td>
</tr>
<tr>
<td>Δh</td>
<td>Wing height above nozzles of configuration</td>
</tr>
<tr>
<td>h'</td>
<td>Critical height for h’ Method (See Figure #13)</td>
</tr>
<tr>
<td>h_1</td>
<td>Height at which the rate of change of lift with height changes</td>
</tr>
<tr>
<td>h_2</td>
<td>Maximum height at which the ground vortex effects are felt</td>
</tr>
<tr>
<td>h_f</td>
<td>Height to which fountain effects are felt</td>
</tr>
<tr>
<td>Ht</td>
<td>Height of aircraft above the ground</td>
</tr>
<tr>
<td>K'</td>
<td>Constant in h’ Method</td>
</tr>
<tr>
<td>K_A</td>
<td>Constant in expression for the contribution of the fountain arms</td>
</tr>
<tr>
<td>K_b</td>
<td>Adjustment factor for the proximity of the RCS jet to the wing tip</td>
</tr>
<tr>
<td>K_c</td>
<td>Adjustment factor for the proximity of the RCS jet to the wing trailing edge</td>
</tr>
<tr>
<td>K_C</td>
<td>Constant in expression for the contribution of the fountain core</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>Kh</td>
<td>Wall jet position factor</td>
</tr>
<tr>
<td>KL</td>
<td>Percentage of lift which LIDs increase over configurations without LIDs</td>
</tr>
<tr>
<td>KS</td>
<td>Constant in expression for multiple jet suckdown (correction factor)</td>
</tr>
<tr>
<td>l</td>
<td>Length of configuration</td>
</tr>
<tr>
<td>ΔL</td>
<td>Absolute change in total lift as the result of a power induced effect</td>
</tr>
<tr>
<td>LID</td>
<td>Lift improvement device</td>
</tr>
<tr>
<td>MCAIR</td>
<td>McDonnell Aircraft Company</td>
</tr>
<tr>
<td>MFVT</td>
<td>Mixed-flow vectored-thrust</td>
</tr>
<tr>
<td>N</td>
<td>Number of fountain arms</td>
</tr>
<tr>
<td>NASA</td>
<td>National Aeronautics and Space Administration</td>
</tr>
<tr>
<td>OGE</td>
<td>Out-of-ground effect</td>
</tr>
<tr>
<td>P</td>
<td>Pressure</td>
</tr>
<tr>
<td>PIE</td>
<td>Power Induced Effects Module</td>
</tr>
<tr>
<td>Pn/P</td>
<td>Nozzle pressure ratio</td>
</tr>
<tr>
<td>RCS</td>
<td>Reaction control system</td>
</tr>
<tr>
<td>S</td>
<td>Total planform area</td>
</tr>
<tr>
<td>S'</td>
<td>Actual surface area between the jets (See Figure #12)</td>
</tr>
<tr>
<td>S''</td>
<td>Potential surface area between the jets (See Figure #12)</td>
</tr>
<tr>
<td>sf</td>
<td>Suckdown/Fountain</td>
</tr>
<tr>
<td>STOVL</td>
<td>Short takeoff and vertical landing</td>
</tr>
<tr>
<td>T</td>
<td>Thrust</td>
</tr>
<tr>
<td>V</td>
<td>Velocity</td>
</tr>
<tr>
<td>V/STOL</td>
<td>Vertical/short takeoff and landing</td>
</tr>
<tr>
<td>Ve</td>
<td>Ratio of the aircraft dynamic pressure to the jet dynamic pressure</td>
</tr>
<tr>
<td>Vo</td>
<td>Forward velocity of aircraft</td>
</tr>
<tr>
<td>w</td>
<td>Width of configuration</td>
</tr>
<tr>
<td>x</td>
<td>Distance of RCS jet ahead of wing trailing edge</td>
</tr>
<tr>
<td>y</td>
<td>Spanwise distance of RCS jet from the wing tip (RCS section only)</td>
</tr>
<tr>
<td>y</td>
<td>Spanwise extent of the jet on the planform (See Figure #12)</td>
</tr>
<tr>
<td>Y</td>
<td>Maximum spanwise extent of jet on planform (See Figure #12)</td>
</tr>
<tr>
<td>YAV-8B</td>
<td>Harrier, a V/STOL jet fighter aircraft (Aircraft shown in Figure #7)</td>
</tr>
</tbody>
</table>

**Greek**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>δ</td>
<td>Jet deflection angle</td>
</tr>
<tr>
<td>θ</td>
<td>Half the angle between adjacent jets and the center of the jet pattern</td>
</tr>
</tbody>
</table>
\( \lambda' \quad \) Exponent in \( h' \) Method

\( \lambda_A \quad \) Exponent in expression for contribution of fountain arms

\( \lambda_C \quad \) Exponent in expression for contribution of fountain core

\( \lambda_S \quad \) Exponent in expression for multiple jet suckdown

\( \Sigma \quad \) Summation

**Subscripts**

- \( A \quad \) Fountain arm
- \( B \) or \( b \quad \) Body
- \( C \quad \) Fountain core
- \( F \quad \) Fountain
- \( f \quad \) Front jets
- \( \text{hover} \quad \) Results from hover calculations
- \( \text{hw} \quad \) High wing
- \( j \quad \) Jets
- \( \text{JW or wake} \quad \) Jet Wake
- \( L \quad \) LIDs
- \( O \quad \) Unaffected by pressure ratio
- \( \text{OGE} \quad \) Out-of-ground effect
- \( P \quad \) Affected by Pressure ratio
- \( r \quad \) Rear jets
- \( \text{RCS} \quad \) Reaction Control System
- \( S \quad \) Suckdown
- \( \text{sf} \quad \) Suckdown/Fountain
- \( \text{square} \quad \) Square planform
- \( V \quad \) Ground Vortex
- \( W \quad \) Wing
- \( \text{wb} \quad \) Wingbody
- \( x \quad \) Individual fountain arm of multiple fountain arm configuration
- \( \infty \quad \) Conditions at infinity or of the freestream
CHAPTER 1

Introduction

Preliminary design of modern aircraft is labor intensive, time consuming and requires detailed knowledge of several engineering disciplines. Interactive computer programs have been developed which reduce the time necessary to perform the calculations involved and permit the engineer to quickly size an aircraft and determine if the design can meet specifications. The Aircraft Synthesis (ACSYNT) code developed by NASA is one of the existing preliminary design and analysis codes. This code will assist an aircraft designer in sizing and design of conventional takeoff and landing aircraft. An effort is now being made to update the code to include sizing and design of vertical and short takeoff and landing (V/STOL) aircraft.

V/STOL aircraft experience changes in the overall lift due to power induced effects. Even though the magnitude of these changes in lift, or lift increments, might be small compared to the installed thrust of an aircraft, they must be estimated if accurate predictions of aircraft performance are to be made. For example, an error of three percent in the total lifting capability in hover is about a 10% error in the prediction of the aircraft range (Reference 3).

In the past, lift increments for simple configurations were predicted by using wind tunnel data to generate empirical models. As more configurations were tested, the empirical models became increasingly more complicated to the point where hand calculations have become impractical and tedious. Computer programs have been developed to calculate each lift increment individually. These programs have proved adequate although the user needs to be knowledgeable about the lift increments, to make many pre-calculations for the
inputs, and to combine the results in order to find the total lift increment. A need exists for a program which does not require detailed knowledge of lift increments, requires only a minimum amount of input from the user and can output individual lift increments and/or total lift increments in any form desired by the user.

The objective of this project was to produce a user friendly FORTRAN code that estimates the lift increments, due to power induced effects, encountered by aircraft in V/STOL flight using the method developed by Kuhn and Stewart (References 6 and 13). The code was developed to enable the programmer to easily include moment increment calculations and customized output. The method uses empirically derived relations based on aircraft configuration and flight regime to calculate the lift increments. The code, which is titled the Power Induced Effect Module (PIE), calculates lift increments due to aerodynamic forces that result from the interaction of the lifting jets with the surrounding environment. These lift increments are due to the suckdown, fountain, ground vortex, jet wake, and the reaction control system. PIE is a stand-alone application, written in FORTRAN, which creates lookup tables that are compatible with ACSYNT. PIE can be easily incorporated into ACSYNT at a later date and made transparent to the user.
CHAPTER 2
Lift Increment Descriptions

Descriptions of the lift increments are presented to assist the reader to better understand this report. Lift increments are considered to be changes in the lift of an aircraft. This project accounts for lift increments which result from the operation of the power plant of a V/STOL aircraft and the interaction between its lifting jets, surrounding air mass, airframe, and landing surface. These increments either add or subtract from the total lift of the aircraft during hover or transition. They are a result of the suckdown, fountain, ground vortex, jet wake and reaction control system suckdown effects. The power induced lift increments which are not accounted for in this project are those dealing with turning and pressure losses in the lifting nozzles, recirculation, and hot-gas ingestion.

Suckdown

The jet exhaust which supports a V/STOL aircraft, entrains its surrounding airmass around the aircraft (Figure #1a). This entrainment is due to viscosity and it is greatest near the jet. The downward flow produced from this entrainment causes a downward force on the airframe (Reference 12) because the wings and body of the aircraft act as flat plates perpendicular to the flow.

Figure #2 is a photograph of a jet flow exiting normal to a flat plate. It can be seen that the entrainment of the flow causes the fluid particles to separate as they flow from above the plate around to the jet (Reference 11). This separated region causes negative pressures on the underside of the plate, sucking it down.
When a single jet is near and perpendicular to the ground, a wall jet is radiated out in all directions from the point of impingement, similar to Figure #1b. When two or more jets are present and widely separated, their wall jets meet on the axis of symmetry between
the jets and since there is no other place to go, they merge and rise upwards, creating a fountain underneath the aircraft.

This effect can be seen in Figure #3 which is a side-view photograph of two nozzles exhausting perpendicular to the ground. The streamlines of the flow are shown by paint streaks on the ground plane and the vertical plane. The plane of symmetry and the actual fountain can be easily seen between the two nozzles.

![Figure #3. Photograph of the flow field between two jets with the same nozzle pressure (Reference 11).](image)

**Fountain Arms**

The fountain arms are created when exhaust from any two neighboring jets meet on the ground at the plane of symmetry between the jets. This is shown between any two adjacent jets in Figure #4. When the two flows meet, they merge and rise upward in a relatively thin fan-shaped sheet as seen in Figure #5. This upward flowing sheet pushes
upward on the lower surface of the aircraft which gives the aircraft a positive lift increment. The upward flowing sheets exiting the vicinity of the jet pattern are referred to as the arms of the fountain or "fountain arms."

Figure #4. Planform view of jet pattern with four jets and the flow from each of the jets (Reference 6).

Figure #5. "Fountain Core and Arms" generated by a configuration with three or more jets - Side view of jet pattern (Reference 6).
**Fountain Core**

The fountain core is generated when the fountain arms created from three or more jets meet at the center of the jet pattern. This fountain core has higher upward velocities than those of the fountain arms (Reference 11) because multiple fountain arms converged at the center of the jet pattern as shown in Figure #4 and #5. These higher upward velocities increase the pressure underneath the aircraft imparting an upward force on its lower surface.

**Lift Improvement Devices**

The lift due to the fountain effect can be increased by the use of lift improvement devices or LIDs. These LIDs are strakes on the lower surface of the aircraft. The LIDs redirect the fluid downward which produces larger upward forces on the aircraft. The flow field underneath an aircraft with LIDs is shown in Figure #6.

![Flow field under an aircraft in hover with lift improvement devices](image)

**Ground Vortex**

A ground vortex forms when the wall jet, from a vertical jet impinging on the ground, physically interacts with the freestream. The freestream causes the wall jet along the ground to roll up into a vortex which is swept in the direction of the freestream.
Figure #7 is a schematic of an aircraft operating close to the ground and shows the ground vortex due to the interaction between the wall jet and the crossflow.

The ground vortex contributes to the lift increment by causing increased flow underneath and near the aircraft. This flow reduces the pressure on the underside of the wing and body which in turn reduces the lift on the aircraft. The magnitude of the lift increment caused by the ground vortex is dependant on ground proximity and aircraft speed. If the aircraft is high enough or fast enough, the ground vortex is swept completely under the airframe causing no decrease in lift. As the aircraft approaches the ground and/or slows down, the loss in lift increases as more of the vortex interacts directly with the airframe.

**Jet Wake**

A jet wake occurs when a jet exits perpendicular to the freestream. The jet is deflected in the direction of the freestream and is quickly transformed into a rolled up vortex pair. Initially, shedding of the vortex pairs are analogous to the vortices which are shed from a solid cylinder placed in a crossflow. Due to the curvature and mixing of the jet in the freestream, the jet is transformed into a vortex pair and their axes become parallel to the freestream. These vortex pairs can be seen in Figure #8, which is a photograph of a cross section of an actual jet exiting perpendicular to the freestream in a water tunnel. The
white portion is the jet flow and the white dots are air bubbles which are induced into the vortices surrounding the jet flow (Reference 11). The vortices created from the jet wake are the primary cause of the interference effects in transition flight. The vortices change the flow field near the airframe which induces additional suction pressures on the surfaces of the aircraft (Reference 10).

Figure #8. Photograph of the cross section of the wake six nozzle diameters downstream of a jet exiting perpendicular to the free-stream in a water tunnel. (Reference 11)

The negative lift increment created by the jet wake is due to large negative pressures developed behind the jet, as shown in Figure #9. This figure depicts pressure contours around a jet exiting perpendicular to the free-stream from a flat plate. It can be seen the jet produces a region of positive pressures upstream of the jet and a much larger region of stronger negative pressures laterally and downstream of the jet (Reference 14).
Reaction Control System

Vertical/short take-off and landing aircraft require pitch, roll, and yaw control when in hover or transition. Typically a portion of the exhaust is bled off from the engine and ducted to the extremities of the aircraft to the reaction control system (RCS). Nozzles on the wing tips are used for roll control and nozzles on the nose and tail can be used for pitch and yaw control. Studies have been performed on the nozzles used for roll control to determine their effectiveness in hover or forward flight. (This report/program will only account for the lift increments resulting from the roll control RCS nozzles due to the unavailability of data for pitch and yaw control nozzles.) It was found the roll control nozzles typically encounter the same type of suckdown and jet wake effects which the main lift engines encounter. These losses are due to the entrainment action of the jet, and the interaction of the free stream and the jet flow (jet wake losses). (See Figure #10.) Both of these effects induced suction pressures on the lower surface of the wing beside and behind
the jet causing a down load on the wing which reduced the effectiveness of the jet. (The magnitude of this lift increment is small compared to the total lift increment, but it is important when determining the maximum rolling moment which the RCS can produce. This will become much more important when moment increment routines are added to the code.)

Figure #10. Roll control reaction control system nozzles and the flow field around the wing

Figure #11. V/STOL induced lift increments (Reference 8)
**Trends and Comparisons of the Lift Increments**

Figure #11 is a graph that displays how lift increments vary with height above the ground. The normalized parameters are used so comparisons between different types and scales of configurations can be made. As can be noted from Figure #11, the ground has a profound effect on the lift increments. They vary near the ground and then level off to a constant out of ground effect (OGE) value as the altitude is increased.
CHAPTER 3
Computational Development

All relations developed for this project are empirical equations which were developed from wind tunnel data by Kuhn (Reference 6) and Stewart and Kuhn (Reference 13). These relations include many configuration and flight regime specific equations. The general equations will be presented with a mention of the specific changes to these equations for given configurations and flight regimes.

Variables

The variables which are present in the PIE routines can be divided into three categories: independent, configuration and control variables. All variables have the capability of being changed by the user. See Appendix B, the User’s/Programmer’s Manual, for more details.

Independent Variables

The independent variables are the flight conditions of the aircraft. These variables are aircraft height above the landing surface, forward velocity, nozzle deflection angle, and angle of attack. They were chosen as independent variables because they define the flight conditions of any particular aircraft design. Also ACSYNT requires lift increments based on these independent variables.

Configuration Variables

Configuration variables define the configuration of a particular aircraft. These variables can be further divided into three categories. The first category are those variables
which define the aircraft wing, body and wing/body. These include widths, lengths, heights, areas, and positions of aircraft components. The second category defines the number, positions, diameters, areas, configurations, and performance of the nozzles. The third category of variables define the positions, configuration, and affected areas of fountain arms and core.

**Control and Result Variables**

The control variables track and control execution, and record errors. These variables include the indices which allow quick and convenient access to results. The result variables store results for each lift increment for each change in the independent variables. These result variables are stored and retrieved using indexed notation \((u(i,j))\) where ‘i’ and ‘j’ can be any positive integer. Due to memory constraints on the computer systems, some of the results are stored in disk files instead of computer memory but are still stored and accessed using the same indexed techniques. Many of the results are stored as components of the total lift increments. Frequently, these components are the lift increments calculated separately for the body, wing, front nozzles, and rear nozzles. Later these components area combined using the principal of superposition.

**Hover**

The only independent variable used in the hover calculations is height. The other independent variables are set to zero velocity, 90° nozzle deflection angle, and 0° angle of attack.

The method used in this study was developed by Kuhn (Reference 6) and its major assumption is the total induced lift increment developed on a V/STOL aircraft can be expressed as:

\[
\frac{\Delta L}{T} = \frac{\Delta L_\infty}{T} + \frac{\Delta L_S}{T} + \frac{\Delta L_F}{T} + \frac{\Delta L_L}{T}
\]  

(1)
Suckdown

$\frac{\Delta L}{T}$ is the out-of-ground effect (OGE) suckdown lift increment. The equation for the OGE suckdown is:

$$\frac{\Delta L_{\infty}}{T} = -0.00022 \sqrt{\frac{S}{A}} \left[ \frac{P_n}{P_{\infty}}^{0.64} \Sigma \pi d \right]^{1.58}$$

where $S$ is the total planform area, $A$ is the total jet exit area, $P_n/P_{\infty}$ is the nozzle pressure ratio, $\Sigma \pi d$ is the total jet perimeter, and $d_e$ is the diameter of an equivalent single jet having area $A$. Experiments have shown that the OGE lift increment is reduced if the nozzles are extended below the surface of a body. The OGE lift increment for a high wing configuration is lower than the low wing configuration and the decrease is related to the square root of the wing height over nozzle diameter as shown in equation (3).

$$\frac{\Delta L_{\infty}}{T} = \left[ \frac{\Delta L_{\infty}}{T} \right]_{hw} + \left[ \frac{\Delta L_{\infty}}{T} \right]_{b} - \left[ \frac{\Delta L_{\infty}}{T} \right]_{wb} \left[ 1 - 0.4 \sqrt{\frac{\Delta h}{d_e}} \right]$$

where $\Delta h$ indicates the wing height above the nozzles of the configuration, and the subscripts $hw$ indicates a high wing, $b$ indicates the body, and $wb$ indicates the wingbody.

$\frac{\Delta L}{T}$ is the additional lift increment due to suckdown experienced in-ground effect (GE). It was first determined experimentally for single jet configurations and then a correction factor was developed for multiple jet configurations. The equation for in-ground effect suckdown is:

$$\frac{\Delta L_{S}}{T} = K_S (-0.015) \left[ \frac{h}{d_e} \left( \frac{D}{d_e} - 1.0 \right) \right]^{2.2 \cdot 0.24 \frac{P_n}{P_{\infty} - 1}}$$

Where in addition to the variables mentioned above, $D$ is the angular mean diameter of the planform, and $K_S$ is the multiple jet correction factor which is a function of configuration geometry. This GE suckdown also changes with wing height similar to the OGE.
suckdown. This is because the wing does not experience the same magnitude of entrainment as the lower surface of the body.

**Fountain Lift**

Two methods were developed by Kuhn to calculate the fountain lift, one for widely spaced jets, the Basic Method, and one for closely spaced jets, the \( h' \) Method.

**Basic Method.** The Basic Method was developed for widely spaced jets with equal thrusts. \( \frac{\Delta L_F}{T} \) is the overall fountain lift increment which is a combination of the lift increment developed by the fountain arms and fountain core.

\[
\frac{\Delta L_F}{T} = \frac{\Delta L_A}{T} + \frac{\Delta L_C}{T}
\]  

(5)

Equations (6) and (7) are used to calculate the fountain arm contribution. Equation (6) is the lift increment for an individual fountain arm ‘x’. Equation (7) combines the fountain arms in the configuration.

\[
\frac{\Delta L_{Ax}}{T} = \frac{2 \left[ Y_x S_{x'} \right]^\gamma}{N \left( e_x S_{x''} \right)} \frac{e_x^2}{(e_x + h) \sqrt{y_x^2 + (e_x + h)^2}}
\]

(6)

\[
\frac{\Delta L_A}{T} = \frac{1}{2} \left( \frac{\Delta L_{A,1}}{T} + \frac{\Delta L_{A,2}}{T} + \ldots + \frac{\Delta L_{A,N}}{T} \right) \left( 7 \sqrt{\frac{h}{d_e}} \frac{d_e}{D - 1} \right)
\]

(7)

\( y_x \) is the spanwise extent of the fountain on the planform, \( Y_x \) is the maximum spanwise extent of the fountain on the planform, \( e \) is half the distance between adjacent jets, \( h \) is the height of the lowest surface above the ground, \( S' \) is the actual surface area between the jets, \( S'' \) is the potential surface area between the jets, and \( N \) is the number of fountain arms. These variables can be seen in Figure #12.
The calculations for the fountain core are similar to those of the fountain arms except the contribution of each fountain arm on the fountain core is:

\[
\frac{\Delta L_{Cx}}{T} = K_C \left( \frac{e_x}{e_x+h} \right)^{\lambda C} \cos \theta_x
\]  

(8)

\[
\frac{\Delta L_{C,1}}{T} + \frac{\Delta L_{C,2}}{T} + \ldots + \frac{\Delta L_{C,N}}{T}
\]  

(9)

Where the constant \( K_C \) and \( \lambda_C \) depend on configuration geometry and altitude, \( e \) is half the distance between adjacent jets, \( h \) is the height of the lowest surface above the ground, and \( \theta \) is half the angle between adjacent jets and the center of the jet pattern.

**h' Method.** The h' Method was developed because configurations with closely spaced jets exhibit an abrupt increase in fountain lift at low altitudes which is not predicted with the Basic Method. Closely spaced jets have higher pressures between the jets which the Basic Method fails to predict. These higher pressures increase rapidly as the jet spacing is reduced. The ability to contain this pressure breaks down as the height is increased because the jets tend to merge into a single jet before they reach the ground.

Two different equations are used for the fountain increment. The decision for which equation to use is based on an altitude, \( h' \).
where \( K' \) and \( \lambda' \) are parameters calculated based on jet spacing, diameter, number, surface area, etc..., \( w \) is width of the configuration, and \( l \) is the length of the configuration. The two curves do not intersect but are joined by a straight line which is tangent to the curve of equation (10) and is projected to \( \frac{\Delta L_F}{T} = 0 \) at a height defined as \( h' \). This is shown in Figure #13.

![Figure #13. Typical variation of fountain lift increment using h’ Method](image)

**Lift Improvement Devices**

\[
\frac{\Delta L_l}{T} = \text{LID lift increment. This lift increment is an addition to the total fountain by a percentage, } K_L, \text{ of the total fountain lift. } K_L \text{ can be determined using the area inclosed by the LIDs, ratio of perimeter by LIDs to the total}
\]
perimeter area enclosed by LIDs, area enclosed by jet centers, length-to-width ratio of jet pattern, and altitude of aircraft. These variables can be seen in Figure #14.

![Diagram of Lift Improvement Device definitions from a triangular configuration with three jets.](image)

**Figure #14.** Lift Improvement Device definitions from a triangular configuration with three jets.

**Forward Flight / Jet Deflection Angle / Angle of Attack**

The calculations based on forward flight, jet deflection angle, and angle of attack use hover calculations as a basis for each lift increment calculation. The method's major assumption is the total lift can be expressed as:

\[
\frac{\Delta L}{T} = \frac{\Delta L_S}{T} + \frac{\Delta L_F}{T} + \frac{\Delta L_{ JW}}{T} + \frac{\Delta L_V}{T}
\]  \hspace{1cm} (12)

where the subscript S refers to the suckdown lift increment (no fountain), F refers to the fountain lift increment, JW refers to the jet wake lift increment, and V refers to the lift increment due to the ground vortex.

**Suckdown/Fountain**

The suckdown and fountain terms are calculated using similar methods. These methods calculate a factor based on the forward extent of the wall jet and the jet spacing of
the front nozzles. This factor, $K_h$, varies between one (the wall jet extending ahead of the configuration) and zero (the wall jet swept behind the configuration.) This factor is multiplied by the square of the sine of the jet deflection angle ($\delta$). (Angle of attack is not used in these calculations.) This calculation produces a factor which modifies the suckdown and fountain terms for the effects of forward speed and jet deflection angle. The suckdown lift increment for forward flight and jet deflection angle is calculated as follows:

$$\frac{\Delta L_s}{T} = \left(\frac{\Delta L_s}{T}\right)_{\text{hover}} K_h \sin^2 \delta \quad (13)$$

The fountain lift increment is calculated using the fountain and LID lift increments from the hover calculations as shown in equation (14).

$$\frac{\Delta L_F}{T} = \left[\frac{\Delta L_F}{T} + \frac{\Delta L_L}{T}\right]_{\text{hover}} \sqrt{\frac{h_f}{h'}} K_h \sin^2 \delta \quad (14)$$

This lift increment calculation is very similar to equation (13) except for the $\sqrt{\frac{h_f}{h'}}$ term. This term is included from the recommendation of Richard Kuhn. It is ratio of $h'$, which was described earlier and $h_f$, which is the maximum height to which the fountain effects are felt on the airframe.

**Ground Vortex**

The ground vortex term is calculated by assuming the configuration is composed of a body and a wing. The lift increment is based on the thrust, diameter and area of the front jets. There are two critical heights associated with the ground vortex lift increments; $h_1$ is the height at which the rate of change of lift with height changes and $h_2$ is the maximum height at which the ground vortex effects are felt. The equations are different for the wing and body and each of those are different for heights based on $h_1$ and $h_2$. An example of these relations is the equation for a body at a height between $h_1$ and $h_2$, equation (14).
\[
\frac{\Delta L_V}{T} = \frac{T_f}{T} \left[ \left( \frac{w_{f}}{w} \right)^{-2} \left( \frac{S_{B}}{A_{f}} \right) \right]^{62} \left( \frac{h}{d_{f}} \right)^{3.2} \sqrt{\frac{1}{V_e}} \Delta C_p \left[ 1 + \sin(\delta-90) \right]^{2} \] (14)

The other equations are similar to equation (14) in complexity and format. The subscript \( f \) refers to the front jets, and \( B \) refers to the body configuration. \( T \) is the thrust, \( w \) is the width, \( l \) is the length, \( S \) and \( A \) are area, \( V_e \) is the ratio of the aircraft dynamic pressure to the jet dynamic pressure, \( \Delta C_p \) is a factor based on \( V_e \) and height, and \( \delta \) is the jet deflection angle.

Jet Wake

The lift increment due to the jet wake is the sum of the increments due to each jet on the wing and body. This is shown in equation (15) where \( B \) refers to the body planform, \( W \) refers to the wing planform, \( f \) refers to the front jets and \( r \) refers to the rear jets.

\[
\frac{\Delta L_{\text{wake}}}{T} = \left[ \left( \frac{\Delta L_f}{T} \right) + \left( \frac{\Delta L_r}{T} \right) \right]_{W} + \left[ \left( \frac{\Delta L_f}{T} \right) + \left( \frac{\Delta L_r}{T} \right) \right]_{B} \] (15)

The lift increment for the wake is figured by first calculating the lift increment out-of-ground effect and then including this term in the overall lift increment in-ground effect.

The OGE jet wake lift increment is calculated using equation (16), which is the equation for the lift increment on a square planform with the jet at the center, and multiplying it by adjustment factors for the planform aspect ratio, longitudinal, vertical, and lateral position of the jet, distance of the jet center from the side of the body, jet deflection angle, non-circular nozzles, jet pressure ratio, and jet flap.

\[
\frac{\Delta L_{\text{wake, square}}}{T} = \left[ -3 \ V_e^2 \left( \sqrt{\frac{S}{A}} - 1 \right)^{67} + 35 \ V_e^{5.5} \left( \sqrt{\frac{S}{A}} - 1 \right) \right] \] (16)

The OGE lift increment is included in equation (17) which is the equation to calculate the jet wake lift increment for an individual jet on a body planform.

\[
\frac{\Delta L_{\text{wake, X}}}{T} = \frac{\Delta L_{\text{wake, OGE}}}{T} \left[ 1 - 0.7 \sqrt{\frac{S}{A_{B}}} \left( \frac{w}{l_{B}} \right)^{1.34} \left( \frac{S}{A} \right)^{0.35} \left( \frac{w}{l_{j}} \right)^{-2} \right] \] (17)
Similar equations are used in calculations for the wing planform. The subscript \( j \) refers to the jets.

If the jets are placed near the trailing edge of the wing, there is a positive lift increment lift increment instead of a negative. This is due to the jet flap effect. In this case a positive lift increment is calculated and replaces the OGE term an equation similar to equation (17).

**Reaction Control System**

The effectiveness of the roll control jets near the wing tip during transition depends on the proximity of the control jets to the wing tip and the wing trailing edge, and jet pressure ratio \( (P_j/P_{\infty}) \). The empirical equations developed for this lift increment are based on an integration of the pressure distributions on the surfaces surrounding the nozzles.

The lift increments are assumed to be made up of two terms, one which is not affected by pressure ratios \( (O) \) and another from the effect of pressure ratios above a critical pressure ratio of 1.893 \( (P) \).

\[
\frac{\Delta L_{RCS}}{T} = \left( \frac{\Delta L}{T} \right)_O + \left( \frac{\Delta L}{T} \right)_P
\]

where:

\[
\left( \frac{\Delta L}{T} \right)_O = \left[ 3V_e^2 - 2.4V_e^2 \right] \sqrt{\frac{S}{A_j}} + 0.41V_e^2 \left( \frac{S}{A_j} \right)^{0.88}
\]

(19)

\[
\left( \frac{\Delta L}{T} \right)_P = -0.017V_e \left( \frac{S}{A} \right)^{0.42} \left( \frac{P_j}{P_{\infty}} - 1.893 \right)^{0.75}
\]

(20)

The variable \( S \) is the area of the wing, \( A_j \) is the jet exit area, \( P_j \) is the jet exit pressure, and \( P_{\infty} \) is the atmospheric pressure. These equations reproduce wind tunnel data with reasonable accuracy up to effective velocity ratios of \( V_e = .1 \), planform-to-jet area ratios up to 7000, and jet pressure ratios up to 45.
The magnitude of the lift increment decreases as the jet is moved closer to the wing tip or trailing edge. This is because the total area near the jet is decreased which reduces the effect of the negative pressures caused by the jet. Equation (21), (22), and (23) show the jet location adjustment factors and how they affect the total RCS lift increment.

\[ K_b = .25 + .2 \left( \frac{y}{d_e} \right)^{58} \]  

\[ K_c = .25 + .06 \frac{x}{d_e} \]  

\[ \frac{\Delta L_{RCS}}{T} = \left[ \left( \frac{\Delta L}{T} \right)_o + \left( \frac{\Delta L}{T} \right)_p \right] K_b K_c \]  

Equation (23) shows the total lift increment as a function of the jet location adjustment factors. \( K_b \) is the adjustment factor for the proximity of the jet to the wing tip, \( y \) is the distance from the trailing edge to the jet, \( K_c \) is the adjustment factor for the proximity of the jet to the wing trailing edge, and \( x \) is the distance from the wing trailing edge to the jet.

This term is usually not included in the total lift increment because its magnitude is much smaller compared to the total lift increment and therefore need not be calculated for the general configuration. These routines were included in PIE to allow rolling moment increment routines, which use this lift increment, to be included at a later date.
CHAPTER 4

Computer Code Description

Philosophy

The Power Induced Effects Module is written in a modular structure which makes it easy to understand and modify. The code is divided into five fundamental sections; control, input, pre-calculations, lift increments, and output. Each of these modules are divided into individual parts to make logic and coding easier to understand. (See Appendix A for a listing of all routines used in PIE.)

Control

The Power Induced Effects (PIE) module is controlled by the control program, COPPIE, which coordinates the execution of PIE. COPPIE’s first task is to make calls to input and pre-calculation subroutines. These calls are made first so all variables are either defined by the user, assigned a default value, or calculated. Within these variables the number, limits, and steps are defined for each independent variable; height, velocity, nozzle deflection angle, and angle of attack. COPPIE steps from one to the maximum number of variations for each independent variable (i.e. If 15 heights are to be considered then COPPIE steps from one to 15 for height). The actual value for the independent variable is calculated and then the control routine for a lift increment(s) is called and/or the next independent variable is incremented. After all independent variables have been incremented up to their maximum values then the output subroutine is called. Figure #15 is a flow chart which shows the basic structure of COPPIE.
Figure #15. Flow chart of the control program COPPIE. (Note - Program Flow is downward unless specified.)
Input

Input routines for PIE perform two functions: input of variables, and check/modification of variables for consistency. The first routine of the input routines reads two input files. The first file contains all user defined variables. The second file contains X and Y coordinates for half the configuration planform. This routine first sets default values for all variables and then reads user defined variables from the first file. The advantage of this style of input over the input in the original programs (variables are entered in the code, hard-coded) is it allows the user to input as many or as few variables (above the minimum configuration variables) as the user desires. It also allows the user to save the configuration in a small, easy to manipulate files and rerun particular configurations as desired. The planform file allows a majority of calculation-intensive variables to be calculated by PIE, which alleviates the user from making tedious calculations.

The second routine assures planforms defined in the second file are consistent with the coordinate system defined in PIE. If the nose of the configuration does not lie on the origin, the routines modify all coordinates of the planforms and necessary configuration variables so the nose does lie on the origin. This makes sure PIE is able to interpret user inputs correctly.

Preliminary Calculations

The original programs have a few serious problems which make them tedious and difficult to use. First, these programs need extensive input in order to calculate the lift increment. These inputs include calculations for actual and potential surface areas between jets, and angular mean diameter of the planform which are tedious calculations because they require an integration over portions of the planform. There are also a number of less tedious calculations, but the number of them make the calculations time consuming for the user. Second, the user needs to know exactly how calculations are made so they can adjust variables so the results are computed correctly. For example, when the configuration has a high wing, areas for the fountain arms are smaller compared to configurations with low
wings, critical angular mean diameters are different, and some lengths based on the nozzles change. These procedures require detailed knowledge of lift increments and the overall method.

A desirable feature of the Power Induced Effects Module (PIE) is it contains a complete set of variables which defines the configuration. This allows lift increment control routines to pick and choose variables needed by lift increment routines. Many pre-calculations need to be made to obtain this complete set of variables. These pre-calculations take a number of hours by hand for each configuration and they change with each new configuration. This means if a new configuration needs to be completed, the calculations need to be refigured, which a computer can do quickly.

**Lift Increments Routines**

The lift increment routines were divided into four sections. The sections are based on which independent variables are used to calculate the lift increment. The first section is lift increments calculated for hover. These lift increments vary with height. The second section contains lift increments which vary with velocity. The third section contains lift increments which vary with height, velocity, and jet deflection angle. The fourth section contains lift increments which change with height, velocity, jet deflection angle, and angle of attack.

**Height**

The hover routines calculate initial lift increments for the suckdown and fountain which vary with height. The RCS, ground vortex, and jet wake lift increments occur when a freestream is present so they are not included in these calculations. The out-of-ground effect (OGE) suckdown is calculated first and then corrections are made for in-ground effect (GE) based on the altitude of the configuration. The fountain terms are calculated based on their proximity to the ground using the “Basic” and “h’” methods outlined by Kuhn (Reference 6).
The specific function of the hover control routine is to recognize when a high wing is present and make appropriate adjustment to variables which it sends to the lift increment routine.

**Forward Velocity**

The reaction control system (RCS) lift increment varies with velocity. Ground effects can be neglected since the size of the RCS jets are much smaller than their height above the ground. Corrections for jet placement are calculated first. The lift increment based on pressure ratio is calculated next and adjustment factors are applied to the results.

The specific function of the RCS control routine is to either execute the lift increment routine or set its result to zero. This is controlled with a flag variable which can be changed by the user.

**Height, Forward Velocity, and Jet Deflection Angle**

Suckdown, fountain, and jet wake lift increments vary with height, forward velocity and jet deflection angle. Suckdown and fountain terms are calculated inside the same increment routine because they both change similarly with the independent variables. A height factor is calculated based on the forward extent of the wall jet. Then the calculations use hover values multiplied by height, forward flight and nozzle deflection angle factors. The control routine for suckdown and fountain routine track the positions of the nozzles and whether there is a high wing and passes the necessary information to the suckdown and fountain lift increment routine.

The jet wake term is calculated using methods described by Stewart (Reference 13). Since the methods makes separate calculations for different planforms and for front and rear jets, the control program is much more detailed than the others. The control program first determines which kind of planforms are to be used: wing and body or wing/body (one planform which is a combination of the wing and body). It checks if the nozzles are near the trailing edge of the wing, for the jet flap effect, and then calculates the
OGE and overall lift increment for the jet wake from front and rear nozzles. This process is repeated for the next planform. The OGE and overall lift increments are calculated in separate routines due to the complexity of each calculation. The calculations for the wing and body are stored separately to allow contributions of each planform to be observed.

**Height, Forward Velocity, Jet Deflection Angle, and Angle of Attack**

The ground vortex term is calculated based on height, forward velocity, nozzle deflection angle, and angle of attack. This is the only term which varies with angle of attack.

The control routine for the ground vortex term is similar to the jet wake control routine with respect to the planforms. The rear nozzles do not contribute to the ground vortex so only the front nozzles are used in the calculations. Lift increments for the wing and body are calculated and stored separately to allow contributions of each planform to be observed.

**Output**

Storage of the results of PIE are in files and arrays which can be recalled with the use of an index notation. This is done so any type of lookup table can be created by the user/programmer. This method of storage allows the programmer to configure an output table in any format using simple index notation when requesting a value (i.e. if the user wants a lift increment for the eighth height, sixth velocity, third deflection angle and fifth angle of attack, it can be recalled by setting the appropriate index variables; i, j, k, and l to 8, 6, 3, and 5 respectively). This feature allows easy access to the lift increments which facilitates many different types of look-up table output.

Currently, PIE creates one three-dimensional look-up table for ACSYNT. This table contains multiple two-dimensional tables based on deflection angle with each individual table based on height and velocity. This table is generated by a single, short subroutine which demonstrates how many other types of tables can be generated.
Other output options are created for viewing purposes. These tables are more examples of the many tables which can be generated. Each lift increment as well as the total lift increments can be viewed based on two of the four independent variables. Currently, outputs are text files which can be read by a plotting routine, developed at NASA Ames for the Silicon Graphics IRIS workstations, and a Macintosh graphing application called KaleidaGraph™.

Limitations

PIE cannot calculate lift increments for configurations with five or more jets or configurations with three or more jets in a straight line either laterally or longitudinally. A temporary solution to this problem is to combined closely spaced jets into single jets, which can be rectangular or oval, until the total number of jets is reduced to four or less.

The methods used in this code are intended for use in preliminary design analysis and to give an indication of the effects of changes to primary configuration variables. Power induced effects are a complex function of many configuration variables and development of a V/STOL aircraft will require careful experimental investigations to accurately determine the induced forces (Reference 13).

Refer to Appendix B, the User's and Programmer's manuals, for additional information on configuration variables, how to run PIE, and more detail on the structure and organization of the code.
CHAPTER 5
Results and Discussion/Validation

When a new computer code is written there are invariably going to be flaws in the code due to calculation or logic errors. This is the reason all codes must to be verified and corrected before they are used reliably for the purposes they were created.

The purpose of this section is to compare the Power Induced Effects Module (PIE) with experimental results and results from Kuhn's programs. PIE is verified using height and forward velocity data from Kuhn’s programs and experimental results. The other two independent variables, jet deflection angle and angle of attack, are not used to verify PIE due to lack of both experimental and computed data. It is difficult to isolate individual experimental lift increment from the total experimental lift increment which means only the total lift increments from PIE can be compared to experimental results. This is not a problem when verifying PIE results with Kuhn’s programs. Since PIE and Kuhn’s routines are based on the same methods, they can be compared lift increment for lift increment.

Many other conceptual configurations were tested that are not included in this report. These configuration were mainly used for error location purposes. They exercised the individual sections of the code to find blatant errors which caused PIE to abort or the lift increments to become unrealistic values. Only general trends and relative magnitudes in the lift increments were examined for these tests.
**Configurations**

The configurations used for verification start as simple flat plate models and progress in complexity to scale models of actual aircraft. These configurations are shown in Figure #16. The first four configurations, disk, body, wing-body and delta, are flat plate models which were experimentally tested in the Hover Test Rig at Rye Canyon by the STOVL/Powered Lift Division of NASA Ames (Reference 15). The next configuration, McDonnell Aircraft Company's (MCAIR) mixed-flow vectored thrust (MFVT), is a V/STOL design which is still in the preliminary design stages. The last configuration is a 15% scale model of an actual V/STOL aircraft, the YAV-8B (Harrier).

![Figure #16. Half-planforms of configurations used for verification.](image-url)
Flat Plate Models

Flat plate models have been experimentally hover tested for suckdown, fountain, and total lift increments encountered in and out of ground effect. The flat plate models tested were: disk, body, wing-body, and delta planforms.

The 20-inch circular disk was tested with one jet in the center. This configuration allowed the suckdown equations to be verified without the effect of any of the other lift increments. The body, wing-body, and delta planforms were tested with two jets placed longitudinally along the planforms. A fountain is created between the two jets on each of the last three planforms. This allows the fountain equations to be verified along with the suckdown equations.

Mixed-Flow Vectored-Thrust Configuration

The mixed-flow vectored-thrust (MFVT) configuration was proposed by the McDonnell Douglas, MCAIR division. Many different nozzle configurations were examined with the previous lift increment programs and considerable data is available. The MFVT was a conceptual design of a V/STOL aircraft that was examined with Kuhn's programs. Even though no experimental tests were conducted with the configuration, Kuhn's program results are valid because the PIE routines results are based on the same methods Kuhn's programs use.

The particular configurations examined in this report are a lateral two-jet case and a triangular three-jet case (one nozzle in front and two nozzles in the rear). The results from Kuhn's programs are very helpful due to the lack of experimental data for forward velocities.

YAV-8B

The YAV-8B, or harrier, configuration is the only configuration which has actual wind tunnel test data from a 15% scale model tested at the McDonnell Aircraft Company (Reference 5). In addition to hover data, a few forward velocity data points are available.
The YAV-8B also has some important configuration changes which were not fully tested in the previous cases. These are lift improvement devices (LIDS) on the underside of the aircraft which trap the fountain, and the jet flap which increases the overall positive circulation about the wing due to the position of the nozzles near the trailing edge of the wing. Both of these configuration changes add to the total lift increment. The experimental data (Reference 9) obtained is for the total lift increments, so only the total values are compared.

**Hover**

The hover calculations were conducted with no forward velocity, 90° jet deflection angle, and zero angle of attack. Most of the results were calculated from heights of two nozzle diameters to twenty nozzle diameters. The two nozzle diameter height was chosen because the method loses accuracy at very low heights due to a strong uncertain flow field created between the jets and the surface. The two nozzle diameter height is also chosen to be lower than the gear height for most aircraft configurations. The gear height is the height of the aircraft while it is on the ground with its gear extended. The twenty nozzle diameter maximum height was chosen because all lift increments either converge to zero or to a constant value. In Figures #17 to #21, the horizontal axis is labeled ‘h/de’, where ‘h’ is the actual height of the aircraft and ‘de’ is the effective nozzle diameter or the diameter of a single nozzle which would have the same exit area as the sum of the areas of all nozzles on a configuration. This label refers to the number of effective nozzle diameters the aircraft is above the ground.

A general reason for the differences between PIE and experimental results for hover can be attributed to the assumption that all the lift increments are linear and can be summed together with no adjustment factors.
Disk

The 20-inch circular disk with a single jet is the simplest configuration tested. The curves in Figure #17 show the computed lift increment compared to the experimental lift increment. At altitudes larger than eight nozzle diameters the curves agree quite well, but for heights less than eight nozzle diameters the curves diverge slightly. This is most likely due to the large rate-of-change of the suckdown lift increment at low heights. Because of this large rate-of-change, any differences between the curves will cause sizable deviations between lift increment values at a given heights.

![Comparison of calculated and measured total lift increments in hover for a 20-inch circular disk (Reference 15)](image)

**Figure #17.** Comparison of calculated and measured total lift increments in hover for a 20-inch circular disk (Reference 15)

Body/Wing-Body

The body and wing-body configuration have two jets as opposed to the single jet present in the disk planform. Also the planform areas of the body and wing-body are much smaller than the planform area of the disk. These two facts cause the fundamental
differences between the magnitudes of the lift increments for the body, wing-body, and disk.

In Figure #18, PIE underpredicts the body total lift increment until a height of seven nozzle diameters is reached and overpredicts the total lift increment after seven nozzle diameters. The same underprediction of the lift increments at lower heights are found on the body as are found on the disk. This implies at lower heights the large slope of the curve magnifies any deviations between lift increment values. At greater heights, differences between experimental and predicted lift increments are not great and can be explained by inaccuracies in the methods for predicting the multiple jet suckdown at high altitudes. This is determined by examining the figures for the body, wing-body and delta planforms, all of which have multiple jets, where the lift increments at high altitudes are always overpredicted from the experimental results. Overall, the exact reasons for the differences are not entirely known. As mentioned by Kuhn (Reference 6), the multiple jet suckdown is the most important and most difficult element of the method to determine.

![Figure #18. Comparison of calculated and measured lift increments in hover for the Body planform (Reference 15)](image-url)
The calculated curve for the wing-body planform in Figure #19 matches closely with the experimental curve for altitudes less than 4.5 nozzle diameters. After this height, PIE overpredicts the lift increment. The good agreement for the lower altitudes is, most likely, a coincidence. Another problem with the method which could have caused the overprediction of the suckdown lift increment at higher altitudes could be the method was developed for low pressure ratios (less than 2.0) but was modified to extrapolate for higher pressure ratios. A higher jet pressure ratio (6.0) was used in the experimental tests. The extrapolation could have caused some differences. In general, higher pressure ratios will tend to entrain more flow around the configuration and thereby causing a greater suckdown increment. The higher pressure ratios also create stronger fountain increments due to the increase in strength of the lifting jets and therefore an increase in strength of the fountain.

![Graph showing comparison of calculated and measured lift increments in hover for the Wing-Body planform (Reference 15)](image)

Figure #19. Comparison of calculated and measured lift increments in hover for the Wing-Body planform (Reference 15)

The point, labeled "fountain point", is most likely an effect of the fountain created between the two jets. For configuration with larger fountains, a very perceptible hump is
usually created near this position in the curve. (A large hump can be seen in Figure #22 from the YAV-8B configuration.) It is also possible this point could be a stray data point from the test, but the first explanation is most probable. PIE does not predict this hump most likely because 1) the hump is in its beginning stages and 2) since PIE is overpredicting the suckdown, the hump is still being washed out before it starts.

Notice the magnitude of the lift increments for the body and wing-body (Figure #18 and #19) are much smaller than the lift increments for the circular disk (Figure #17). This is due the differences in area near the jet exits. Suckdown is caused by the decrease in pressure on the underside of a configuration from the entrainment action of a jet. Since pressure is force per area, then there would be a larger force on a configuration which has more area near the jet. The disk is inches in diameter which is five times larger than the body width of four inches and therefore the disk experiences a larger negative lift increment. This can also be seen by comparing the lift increments of the body and wing-body configurations. The wing-body has a larger lift increment magnitude because it has the wing and the body near the nozzles. Basically the wing area near the nozzles causes the increase in the suckdown lift increment over that of the body.

Delta

The calculations for the delta planform agree well with the experimental data from the hover test rig. The two curves in Figure #20. are almost identical for low altitudes and diverge by only a small amount at the higher altitudes. Also, as expected, the magnitude of the lift increments were higher than the wing-body increments, due to the increase in area around the nozzles of the delta planform.
Figure #20. Comparison of calculated and measured lift increments in hover for the Delta planform (Reference 15)

Mixed Flow Vectored Thrust

The MFVT configuration with 3-jets is examined because a two jet cases was examined with the flat plate tests and the YAV-8B tests examine the 4-jet case. Figure #21 is a comparison between the original program developed by Kuhn and PIE. As can be seen, all the lift increments are approximately the same except for a slight divergence between the suckdown terms. This deviations between the two codes can be attributed to the differences in the configuration variables. A great effort was made to ensure all configurations variables were the same throughout each program, but there were some variables which were calculated differently in PIE than were input by the user of Kuhn’s program. These variables included the planform areas and $\bar{D}$, the angular mean diameters of the planforms.
YAV-8B

Figure #22 is a comparison between the total lift increments from wind tunnel tests of a 15% scale model of the YAV-8B and the total lift increments calculated by PIE. The hump in the data at the beginning of the graph is a result of the fountain arms and core caused by the four lifting jets of the configuration. The fountain effect tapers off at heights around 15 ft. for the calculated curve. The experimental curves has some fountain effects up to 24 ft. After the fountain effect diminishes, both curves reflect the trend of the suckdown increments. Some of the differences at the higher heights could be explained by the methods which were developed for the effect of lift improvement devices, wing height and body contour. (The body contour of some configurations are typically rounded to some extent, especially on the YAV-8B. Since the experimental results were found mostly
from test data of flat plate models, a multiplication factor was developed to correct for the flat plate data.) The development of methods for these effects were all based on very limited experimental data and therefore more experimental tests need to be completed to refine and gain confidence in the methods. Overall the results are promising for this case.

**Forward Flight**

Two additional lift increments are calculated for forward flight with nozzle deflection than are calculated for hover. These are the ground vortex and jet wake increments. Since forward flight contains all lift increments, it should be carefully verified. The only problem is experimental data for forward flight cases are very limited. The only actual experimental data available is for the YAV-8B. Due to the lack of data, more testing should be done to further validate the forward flight predictions.

Forward flight with nozzle deflection is an important flight regime for V/STOL aircraft. This typically occurs when a V/STOL aircraft is in transition from jet born to
wing born flight or visa versa. Transition usually occurs near the ground and is a critical
time for the aircraft. It can be a great advantage to the aircraft designer, while working on
preliminary design, to know the lift increment characteristics of the aircraft in transition.
This allows the designer to modify their configuration to obtain the most favorable forward
flight lift increment characteristics.

MFVT

Again, the programs developed by Kuhn were used to check the validity of PIE for
the MFVT 2-jet and 3-jet cases. The graphs created for this section were created to observe
the variation of the lift increment with respect to \( V_e \), the dynamic pressure ratio. The
dynamic pressure ratio is the square root of the ratio of the dynamic pressure of the aircraft
to the dynamic pressure of the jet.

\[
\text{sf} = \frac{\text{Suckdown/Fountain}}{\text{Ve}}
\]

Figure #23. Comparison of calculated lift increments in forward flight at
an altitude of 6 ft. for the MFVT, lateral 2-jet configuration
from Kuhn's program and PIE (Reference 7)
Figure #23 is a comparison of the lift increments calculated from PIE and Kuhn’s program for the MFVT configuration, with two laterally placed jets, in forward flight. There are a number of discrepancies in the two codes which are explained next.

PIE overpredicts the suckdown/fountain from Kuhn’s program by almost 80% at low velocities. The magnitude of overprediction decreases as the velocity is increased. The reason for this overprediction is when Kuhn’s program calculates this term, it uses a single jet equation, equation (4) without the multiple jet factor Ks, to calculate the multiple jet suckdown. This single jet equation calculates a much smaller suckdown effect than the multiple jet equation. Another small discrepancy in the suckdown calculation is the addition of an extra OGE suckdown term, equation (3), present in the fountain calculation. This would tend to offset the previous miscalculation, but the magnitude of the OGE term is significantly less than the suckdown term in ground effect.

The ground vortex terms agree closely until the dynamic pressure ratio ($V_e$) reaches a value of .1. After this point Kuhn’s calculations tend towards zero faster than PIE’s calculations. This is attributed to the variation in the algorithms between the two codes. The maximum height for which the ground vortex effects are felt is not the same for the body as for the wing. Kuhn’s code assumes the total ground vortex term is zero when the maximum ground vortex height for the body is reached. The method described in reference 12 indicates the ground vortex effects still occur on the wing after the effects on the body reach zero. From the graph it can be seen when Kuhn’s ground vortex term reaches zero, PIE’s ground vortex term changes slope. This is because the ground vortex effects are still occurring on the wing after the body’s ground vortex effects become zero. This is the method is presented in reference 13.

The wake terms also are the same at low velocities and diverge with higher velocity. These differences can be attributed to three reasons:

1) Kuhn’s program does not include the lift adjustment factors for longitudinal position and flap extension which are used with equation (16).
2) A mistake in the formula for the wing aspect induced lift adjustment factor was found in Kuhn's code.

3) A term is used in Kuhn's code which is only to be calculated when there are longitudinal placed jets. This configuration has laterally placed jets.

Figure #24 is a comparison of lift increments calculated from PIE and Kuhn's program for the MFVT configuration, with three jets, in forward flight. The discrepancies found in this figure for the suckdown/fountain and the ground vortex lift increments are due to the same reasons suggested above for the two-jet case.

The differences between the wake terms are also for the same reason as the two-jet case except there are a few other distinctions. When Kuhn's program calculates the aspect induced lift adjustment factor, which is used with equation (16), it uses the lowest body width instead of the width at the wing which is the width PIE uses. PIE does not use the lowest body width so therefore the values are perturbed slightly. Another difference
between the two codes, is when PIE calculates the longitudinal position adjustment factor for the wing, it uses an equation specific for the wing. Kuhn’s program uses a value of 1.0 for the wing. Also when there is a jet near the trailing edge, Kuhn’s program uses a different equation for the lift increment than the wing equation, similar to equation (17), used by PIE. This is the reason for the different trends during the low velocities of the graph. This wing equation calculates a positive value for the wake lift increment due to the placement of the jets in relation to the trailing edge of the wing (the jet flap effect).

The total lift increments for each configuration follow from the summation of all the individual lift increments. Since each individual lift increment from the two codes are not the same then it follows the total lift increments will not be identical either.

The comparison between the two codes helped debug PIE and also helped find some discrepancies between the original programs developed by Kuhn and the methods developed by Kuhn and Stewart.

**YAV-8B**

The forward velocity experimental data for the YAV-8B is very limited, and provides only three velocity points; 0, 17 and 24 knots. These correspond to dynamic pressure ratios of 0.0, 0.023 and 0.033 respectively. More experimental data points are needed, especially in the higher velocity region, to draw more meaningful conclusions.

The comparisons between experimental and calculated lift increments for the YAV-8B are shown in Figure #25. Three different altitudes are shown in the graph: 4, 10, and 25 ft. The trends of the experimental test can not be accurately determined from three data point so the only analysis which can be accomplished is to examine the magnitudes of the lift increments. The lift increments at lower heights tend to agree more than those at greater heights but for the limited velocity range, PIE predicted the total lift increments well. A possible reason for the differences between PIE and experimental results can be attributed to the assumption that all lift increments are linear and can be summed together
with no adjustment factors. (The experimental data in Figure #25 are connected with lines to allow the reader to distinguish between the three groups of curves).

Figure #25. Comparison of calculated and measured total lift increments in forward flight for a 15% scale model of the YAV-8B (Reference 9)
CHAPTER 6
Conclusions and Recommendations

An easy to use, and modify computer code was developed which makes preliminary estimates of lift increments encountered by V/STOL aircraft due to the suckdown, fountain, ground vortex, and jet wake. This code provides the user with a quick and easy-to-use way to evaluate the trends and magnitudes of lift increments encountered by V/STOL aircraft in hover and transition.

The Power Induced Effects Module is able to reproduce the trends of the lift increments encountered in hover accurately for the flat plate models. The magnitude of the lift increments are reasonably accurate for the range of values investigated. The values of the MFVT hover lift increments calculated using Kuhn’s program and PIE agree quite well and any inaccuracies are explained by the differences between the configuration variables. The hover lift increments calculated by PIE for the YAV-8B agree well with the experimental results.

The lift increments encountered in forward flight are calculated by PIE and Kuhn’s code for two MFVT configurations. The magnitudes and trends of these lift increments agree well except for small differences which are caused by slight deviations between the algorithms of the two programs. There are not enough experimental lift increment points to accurately evaluate the trends from PIE for the YAV-8B. The magnitudes of the calculated lift increments fall close to those of the experimental lift increments. Given the number of configuration variables, these accuracies are quite good.

The most obvious drawback of the power induced effect module is the lack of forward flight, jet deflection angle, and angle of attack validation. Further validation of
PIE needs to be accomplished for these independent variables. Also the code needs to be fully incorporated into the ACSYNT design program so it can be used to its fullest potential.

An enhancement to the code would be the addition of routines which would calculate the moment increments which are caused by the non-symmetric lift increments. This would not be very difficult because the code was initially written with the intent to include these moment increments so most of the necessary variables have already been incorporated within the code. Lack of time was the reason this was not done for this project. Much testing and validation needed to be completed for the moment increment routines, similar to the amount done for the lift increment routines.
BIBLIOGRAPHY

1. Aircraft Synthesis Program (ACSYNT), National Aeronautics and Space Administration, Ames Research Center, Moffett Field, California, 1975.


15. Wardwell, D.A. "Hover Test Rig at Rye Canyon, CA (Computerized data)." STOVL/Powered Lift Division (FAP), NASA Ames Research Center. 1989
APPENDIX A

PIE FORTRAN Code

PROGRAM COPPIE
C--------------------------------------------------
C ACRONYM: COntrol Program for Powered Induced Effects.
C PURPOSE: The purpose of COPPIE is to act as the control program (or
driver) for the Power Induced Effects Module. Its function
is to know everything about the aircraft that is being
analyzed. In other words, it will provide all the
subroutines that it calls with what whatever input is
needed to complete the calculations for a particular
configuration.
C LOCAL VARIABLES (in addition to the above parameters):
C NAME TYPE I/O UNITS DESCRIPTION
C ICALC I - ---- Flag which ACSYNT uses to control
C execution of subroutines.
C 1 = Input
C 2 = Calculations
C 3 = Output
C (Not used at this time)
C GLOBAL VARIABLES (in addition to the above parameters and local vars):
C (This variable list contains all variables in the Power Induced
Effects module separated by their respective common blocks. The rest
of the subroutines contains only those variables which are present
within that individual subroutine.)
C CONPIE Common Block
C NAME TYPE I/O UNITS DESCRIPTION
C AEND R I deg Ending value for angle of attack
C ASTART R I deg Starting value for angle of attack
C ASTEP R I deg Step value for angle of attack
C DEND R I deg Ending value for jet deflection
C angle
C DSTART R I deg Starting value for jet deflection
C angle
C DSTEP R I deg Step value for jet deflection angle
C HEND R I Ft Ending value for altitude
C HSTART R I Ft Starting value for altitude
C HSTEP R I Ft Step value for altitude
C VEND R I kts Ending value for aircraft velocity
C VSTART R I kts Starting value for aircraft
### FIGPIE Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Velocity</td>
<td></td>
<td></td>
<td></td>
<td>Step value for aircraft velocity</td>
</tr>
<tr>
<td>VSTEP</td>
<td>R</td>
<td>I</td>
<td>kts</td>
<td>Width of Body</td>
</tr>
<tr>
<td>I/O UNITS</td>
<td></td>
<td></td>
<td></td>
<td>Width of Center Section</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Width of Jet Pattern</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Width of Wing (Wing span)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Width of Wing-Body</td>
</tr>
<tr>
<td>CONFIG</td>
<td>C*18</td>
<td>I</td>
<td></td>
<td>Short title of current config</td>
</tr>
<tr>
<td>CONFIG</td>
<td>C*18</td>
<td>I</td>
<td></td>
<td>Diameter of each Front jet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Diameter of each Rear jet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Diameter of Roll RCS nozzle</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dbar of Body</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dbar of Center Section</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dbar of Wing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dbar of Wing-Body</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Effective Diameter of Front jets</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Effective Diameter of Rear jets</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Effective Diameter of Front &amp; Rear jets combined</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Density Ratio (Jet / Atm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Half dist between adjacent jets (1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Half dist between adjacent jets (3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Half dist between adjacent jets (4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Name of file which contains body &amp; wing planform coordinates</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Boundary layer factor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Adjustment factor for flap extension when calculating the lift loss due to jet induced effects.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Body contour factor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Length of body</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Length of Center Section</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Length of Front jet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Length of Rear jet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Length of Wing-Body</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean Aerodynamic Chord of wing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mass flow rate for one RCS nozzle</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Number of data points for Body planform</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Number of data points for Center Section planform</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Number of data points for Wing planform</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Number of data points for Wing-Body planform</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Number of divisions to be used when calculating suckdown areas (SDAREA) and Dbars (DIABAR)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Total NUMBER of jets</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NUMBER of Front jets</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NUMBER of Rear jets</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Total perimeter of jets</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Ratio of perimeter enclosed by lids to total perimeter</td>
</tr>
<tr>
<td>Variable</td>
<td>Type</td>
<td>Units</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>-------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>PR_F</td>
<td>R</td>
<td>I</td>
<td>Jet Pressure Ratio for front jets</td>
<td></td>
</tr>
<tr>
<td>PR_FR</td>
<td>R</td>
<td>I</td>
<td>Jet Pressure Ratio for all jets</td>
<td></td>
</tr>
<tr>
<td>PR_R</td>
<td>R</td>
<td>I</td>
<td>Jet Pressure Ratio for rear jets</td>
<td></td>
</tr>
<tr>
<td>PR_RCS</td>
<td>R</td>
<td>I</td>
<td>Roll RCS Pressure Ratio</td>
<td></td>
</tr>
<tr>
<td>PT_RCS</td>
<td>R</td>
<td>I</td>
<td>Total pressure for one roll RCS jet</td>
<td></td>
</tr>
<tr>
<td>Q_F</td>
<td>R</td>
<td>lb/sq ft</td>
<td>Dynamic pressure for the front jets</td>
<td></td>
</tr>
<tr>
<td>Q_FR</td>
<td>R</td>
<td>lb/sq ft</td>
<td>Dynamic pressure for Front and Rear jets</td>
<td></td>
</tr>
<tr>
<td>Q_R</td>
<td>R</td>
<td>lb/sq ft</td>
<td>Dynamic pressure for the rear jets</td>
<td></td>
</tr>
<tr>
<td>Q_RCS</td>
<td>R</td>
<td>lb/sq ft</td>
<td>Dynamic pressure for roll RCS jets</td>
<td></td>
</tr>
<tr>
<td>R_B</td>
<td>R</td>
<td>Ft</td>
<td>Corner radius of body sides.</td>
<td></td>
</tr>
<tr>
<td>S_B</td>
<td>R</td>
<td>Sq Ft</td>
<td>Area of Body</td>
<td></td>
</tr>
<tr>
<td>S_CS</td>
<td>R</td>
<td>Sq Ft</td>
<td>Area of Center Section</td>
<td></td>
</tr>
<tr>
<td>S_F</td>
<td>R</td>
<td>Sq Ft</td>
<td>Area of Front jets</td>
<td></td>
</tr>
<tr>
<td>S_FA1</td>
<td>R</td>
<td>Sq Ft</td>
<td>Area affected by fountain arm (1)</td>
<td></td>
</tr>
<tr>
<td>S_FA3</td>
<td>R</td>
<td>Sq Ft</td>
<td>Area affected by fountain arm (3)</td>
<td></td>
</tr>
<tr>
<td>S_FA4</td>
<td>R</td>
<td>Sq Ft</td>
<td>Area affected by fountain arm (4)</td>
<td></td>
</tr>
<tr>
<td>S_FR</td>
<td>R</td>
<td>Sq Ft</td>
<td>Total jet exit area</td>
<td></td>
</tr>
<tr>
<td>S_JP</td>
<td>R</td>
<td>Sq Ft</td>
<td>Area enclosed by Jet Pattern</td>
<td></td>
</tr>
<tr>
<td>S_LID</td>
<td>R</td>
<td>Sq Ft</td>
<td>Area enclosed by LIDS</td>
<td></td>
</tr>
<tr>
<td>S_R</td>
<td>R</td>
<td>Sq Ft</td>
<td>Area of Rear jets</td>
<td></td>
</tr>
<tr>
<td>S_W</td>
<td>R</td>
<td>Sq Ft</td>
<td>Area of Wing</td>
<td></td>
</tr>
<tr>
<td>S_WB</td>
<td>R</td>
<td>Sq Ft</td>
<td>Area of Wing-Body</td>
<td></td>
</tr>
<tr>
<td>SCALE</td>
<td>R</td>
<td>I</td>
<td>Actual scale factor which adjust the area of fountain arm (3) to account for the fact the curvature of the aircraft's nose tends not to hold fountain arm very well causing the area that fountain arm affects to be scaled down.</td>
<td></td>
</tr>
<tr>
<td>SCALE3</td>
<td>R</td>
<td>I</td>
<td>Fountain arm (3) scaler (Percentage measured from the front jets to the nose of the aircraft.) Tries to approximate SCALE but does not do a great job (better than nothing.)</td>
<td></td>
</tr>
<tr>
<td>SF_FB</td>
<td>R</td>
<td>Sq Ft</td>
<td>Area ahead of Front jets using body planform</td>
<td></td>
</tr>
<tr>
<td>SF_FW</td>
<td>R</td>
<td>Sq Ft</td>
<td>Area ahead of Front jets using wing planform</td>
<td></td>
</tr>
<tr>
<td>SF_FWB</td>
<td>R</td>
<td>Sq Ft</td>
<td>Area ahead of Front jets using the wingbody planform</td>
<td></td>
</tr>
<tr>
<td>SF_RB</td>
<td>R</td>
<td>Sq Ft</td>
<td>Area ahead of Rear jets using the body planform</td>
<td></td>
</tr>
<tr>
<td>SF_RW</td>
<td>R</td>
<td>Sq Ft</td>
<td>Area ahead of Rear jets using the wing planform</td>
<td></td>
</tr>
<tr>
<td>SF_RWB</td>
<td>R</td>
<td>Sq Ft</td>
<td>Area ahead of Rear jets using the wingbody planform</td>
<td></td>
</tr>
<tr>
<td>SP_FA1</td>
<td>R</td>
<td>Sq Ft</td>
<td>Potential area affected by fountain arm (1)</td>
<td></td>
</tr>
<tr>
<td>SP_FA3</td>
<td>R</td>
<td>Sq Ft</td>
<td>Potential area affected by fountain arm (3)</td>
<td></td>
</tr>
<tr>
<td>SP_FA4</td>
<td>R</td>
<td>Sq Ft</td>
<td>Potential area affected by fountain arm (4)</td>
<td></td>
</tr>
<tr>
<td>SP_JP</td>
<td>R</td>
<td>Sq Ft</td>
<td>Actual surface area within area enclosed by nozzles</td>
<td></td>
</tr>
<tr>
<td>SPLY_F</td>
<td>R</td>
<td>Ft</td>
<td>SPLY angle of Front jet</td>
<td></td>
</tr>
<tr>
<td>SPLY_R</td>
<td>R</td>
<td>Ft</td>
<td>SPLY angle of Rear jet</td>
<td></td>
</tr>
<tr>
<td>T_F</td>
<td>R</td>
<td>lb</td>
<td>Thrust of Front jets</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Size</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
<td>------</td>
<td>--------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>T_FR</td>
<td>R</td>
<td>I</td>
<td>lb</td>
<td></td>
</tr>
<tr>
<td>T_R</td>
<td>R</td>
<td>I</td>
<td>lb</td>
<td></td>
</tr>
<tr>
<td>T_RCS</td>
<td>R</td>
<td>I</td>
<td>lb</td>
<td></td>
</tr>
<tr>
<td>THA_1</td>
<td>R</td>
<td>I</td>
<td>deg</td>
<td></td>
</tr>
<tr>
<td>THA_3</td>
<td>R</td>
<td>I</td>
<td>deg</td>
<td></td>
</tr>
<tr>
<td>THA_4</td>
<td>R</td>
<td>I</td>
<td>deg</td>
<td></td>
</tr>
<tr>
<td>TP_RCS</td>
<td>R</td>
<td>I</td>
<td>Rankine</td>
<td></td>
</tr>
<tr>
<td>TT_F</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td></td>
</tr>
<tr>
<td>TT_R</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td></td>
</tr>
<tr>
<td>W_F</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>W_R</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>WIDE</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td></td>
</tr>
<tr>
<td>WL_B</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td></td>
</tr>
<tr>
<td>WL_CS</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td></td>
</tr>
<tr>
<td>WL_F</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td></td>
</tr>
<tr>
<td>WL_JP</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td></td>
</tr>
<tr>
<td>Pattern</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WL_R</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td></td>
</tr>
<tr>
<td>WL_WB</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td></td>
</tr>
<tr>
<td>X_BODY (500)</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>X_CORE</td>
<td>R</td>
<td>FT</td>
<td>X-coordinate of fountain core</td>
<td></td>
</tr>
<tr>
<td>X_CS (30)</td>
<td>R</td>
<td>O</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>X_FR</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>X_RCS</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>X_WB (500)</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>X_WING (500)</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>XCA_CG</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>X_CA</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>X.CG</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>XC_G</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>XCG_C2</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>XCG_C4</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>XCG_F</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>XCG_I</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>XCG_R</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>XNOZ_F</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>XNOZ_R</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>XTE_F</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>XTE_R</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>Y_1</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>Y_3</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>Y_4</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>Y_BODY (500)</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td>Y_CORE</td>
<td>R</td>
<td>FT</td>
<td>Y-coordinates of fountain core</td>
<td></td>
</tr>
<tr>
<td>Y_CS (30)</td>
<td>R</td>
<td>O</td>
<td>Ft</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Total thrust in all jets</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Thrust of Rear jets</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Thrust of roll RCS nozzle</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Half angle between jets (1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Half angle between jets (3)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Half angle between jets (4)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Temperature of flow in roll RCS nozzle</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Front Thrust / total Thrust</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Rear Thrust / total Thrust</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Width of Front jet</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Width of Rear jets</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Flow Weight of Inlet / Flow Weight of Exit</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Width to Length ratio of Body</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Width to Length ratio of Center Section</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Width to Length ratio of Front jets</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Width to Length ratio of Jet</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Width to Length ratio of Rear jets</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Width to Length ratio of Wing-Body</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X-coordinates of Body planform</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X-coordinate of fountain core</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X-coordinates of Center Section planform</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Distance between Front &amp; Rear jets</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Distance of roll RCS nozzle ahead of wing trailing edge</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X-coordinates of Wing-Body planform</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X-coordinates of Wing planform</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Distance of center of area ahead of CG</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X-coordinate of Center of Area</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X-coordinate of Center of Gravity</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Distance from CG to MAC/2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Distance from CG to MAC/4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Distance Front jet is ahead of CG</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Inlet longitudinal distance ahead of CG</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Distance Rear jet is ahead of CG</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X-coordinates of Front NOZZle</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X-coordinates of Rear NOzzle</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X-coordinate of trailing edge for front nozzle (root TE if Nozzle within body)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X-coordinate of trailing edge for rear nozzle (root TE if Nozzle within body)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Spanwise extent of planform on fountain arm (1) center line.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Spanwise extent of planform on fountain arm (3) center line.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Spanwise extent of planform on fountain arm (4) center line.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Y-coordinates of Body planform</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Y-coordinate of fountain core</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Y-coordinates of Center Section</td>
<td></td>
</tr>
</tbody>
</table>
C Y F R
C YNOZ F R
C YNOZ R R
C Y R R
C Y RCS R
C
C Y WB(500) R I Ft
C Y WING(500) R I Ft
C Y_ F R
m
C
C YB R R
C
C
C YP 1 R
C YP 3 R
C YP--4 R
C _F R
C
C YWB R R
C
C Z B R
C
C ZCG I R
C Z W R
C

### HOVPIE Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>AJ</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
<td>Total jet exit area.</td>
</tr>
<tr>
<td>BF1</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Half length of fountain arm (1).</td>
</tr>
<tr>
<td>BF3</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Half length of fountain arm (3).</td>
</tr>
<tr>
<td>BF4</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Half length of fountain arm (4).</td>
</tr>
<tr>
<td>BF01</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Maximum spanwise extent of fountain arm (1).</td>
</tr>
<tr>
<td>BF03</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Maximum spanwise extent of fountain arm (3).</td>
</tr>
<tr>
<td>DB</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Maximum spanwise extent of fountain arm (4).</td>
</tr>
<tr>
<td>DC</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Body D bar.</td>
</tr>
<tr>
<td>DE</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Configuration D bar.</td>
</tr>
<tr>
<td>DH</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Equivalent diameter of jets.</td>
</tr>
<tr>
<td>E</td>
<td>I</td>
<td></td>
<td>Ft</td>
<td>Delta height of wing.</td>
</tr>
<tr>
<td>HEIGHT</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Length to width ratio of jet pattern.</td>
</tr>
<tr>
<td>HW</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Altitude at which calculations are to be made.</td>
</tr>
<tr>
<td>KRB</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Half width of body for jets outside outside of body, else use half distance between adjacent jets.</td>
</tr>
<tr>
<td>L</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Body contour factor</td>
</tr>
<tr>
<td>NJETS</td>
<td>I</td>
<td></td>
<td>----</td>
<td>Length of configuration.</td>
</tr>
<tr>
<td>PER</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Total number of jets on config</td>
</tr>
<tr>
<td>PP</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td>Total perimeter of jets (all jets)</td>
</tr>
<tr>
<td>PR</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td>Fraction of lid perimeter enclosed</td>
</tr>
<tr>
<td>SB</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
<td>Jet pressure ratio.</td>
</tr>
<tr>
<td>SC</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
<td>Planform area of body.</td>
</tr>
<tr>
<td>SCP</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
<td>Area enclosed by jet pattern.</td>
</tr>
</tbody>
</table>

Actual surface area inside jet pattern.
Area affected by fountain arms (1).
Area affected by fountain arms (3).
Area affected by fountain arms (4).
Potential surface area between jets (1).
Potential surface area between jets (3).
Potential surface area between jets (4).
Area enclosed by lids.
Planform area of configuration
Front jet splay angle.
Half angles between jets (1).
Half angles between jets (3).
Half angles between jets (4).
Width of body.
Width of configuration.
Half distance between adjacent jets (1).
Half distance between adjacent jets (3).
Half distance between adjacent jets (4).
Distance center of area ahead of CG
Distance between front jets.
Distance between rear jets.

**MISC Common Block**

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>POINTS</td>
<td>I</td>
<td>I</td>
<td>-----</td>
</tr>
<tr>
<td>XPTS</td>
<td>R</td>
<td>I</td>
<td>-----</td>
</tr>
<tr>
<td>YPTS</td>
<td>R</td>
<td>I</td>
<td>-----</td>
</tr>
<tr>
<td>TOTAL</td>
<td>R</td>
<td>I</td>
<td>-----</td>
</tr>
</tbody>
</table>

**PIEFLAG Common Block**

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLGRCS</td>
<td>L</td>
<td>O</td>
<td>-----</td>
</tr>
<tr>
<td>HDEOUT</td>
<td>L</td>
<td>I</td>
<td>-----</td>
</tr>
<tr>
<td>PRTFLG</td>
<td>L</td>
<td>I</td>
<td>-----</td>
</tr>
<tr>
<td>RCSFLG</td>
<td>L</td>
<td>O</td>
<td>-----</td>
</tr>
<tr>
<td>TYPE_F</td>
<td>C*4</td>
<td>O</td>
<td>-----</td>
</tr>
<tr>
<td>TYPE R</td>
<td>C*4</td>
<td>O</td>
<td>-----</td>
</tr>
<tr>
<td>VEOU_</td>
<td>L</td>
<td>I</td>
<td>-----</td>
</tr>
</tbody>
</table>

**DESCRIPTION**

Number of points which define polygon (not to exceed 500)
X-coordinates of polygon
Y-coordinates of polygon
Total area enclosed by the polygon.

Flag which signals if there is an RCS on the configuration
(TRUE - RCS, FALSE - No RCS)
Signals when to output tables based on height or height/De
TRUE - Print based on Height/DE,
FALSE - Print based on height
Signals when to output to screen.
Flag which identifies if RCS lift loss has calculation exceeded the area ratio (Swing / Ajet > 7000)
TRUE - Exceeded
FALSE - Not Exceeded
Description of type of front nozzle (CIRCular, OVAL, RECTangular)
Same as TYPE_F but for rear nozzles
Signals when to output tables based on VE
TRUE - Print Based on VE
FALSE - Print Based on Velocity
Identifies when WingBody planform has been enter by the user. Used to determine when to use wingbody in calculations.

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFACT</td>
<td>R</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>AJ</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
</tr>
<tr>
<td>DE</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>DF</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>DH</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>EX</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>LTV</td>
<td>R</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>I</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>R</td>
<td>I</td>
<td>lb/sq ft</td>
</tr>
<tr>
<td>TFT</td>
<td>R</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>WLB</td>
<td>R</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>WLJ</td>
<td>R</td>
<td>I</td>
<td></td>
</tr>
</tbody>
</table>

DESCRIPTION
Area ratio - based on weather the wing or body is being used
Area of the front jets
Equivalent Diameter of front jets
Diameter of front jets
Height of wing above jets
Distance from jet pattern center to center of front jets.
Lift loss due to ground vortex.
Number of front jets
Dynamic pressure of the nozzles
Front thrust split
Width to length ratio of the body.
Width to length ratio of the jets.

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>R</td>
<td>I</td>
<td>feet</td>
</tr>
<tr>
<td>DRCS</td>
<td>R</td>
<td>I</td>
<td>feet</td>
</tr>
<tr>
<td>NPR</td>
<td>R</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>R</td>
<td>I</td>
<td>lb/sq ft</td>
</tr>
<tr>
<td>S</td>
<td>R</td>
<td>I</td>
<td>sq ft</td>
</tr>
<tr>
<td>SAJ</td>
<td>R</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>R</td>
<td>I</td>
<td>lb</td>
</tr>
<tr>
<td>X</td>
<td>R</td>
<td>I</td>
<td>feet</td>
</tr>
<tr>
<td>Y</td>
<td>R</td>
<td>I</td>
<td>feet</td>
</tr>
</tbody>
</table>

DESCRIPTION
Wing span
Roll RCS jet Diameter.
Roll jet nozzle pressure ratio.
Roll jet dynamic pressure.
Wing area
Wing area / jet area
RCS roll jet thrust
Jet distance ahead of wing trailing edge.
Jet distance in from wing tip.

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBERR</td>
<td>I</td>
<td>O</td>
<td></td>
</tr>
</tbody>
</table>

DESCRIPTION
Error flag returned by DIABAR. If IFLAG = 1, then an explanation of the problem is given at the time of the error.

0 = no errors.
1 = values of dbar & area probably bad.
2 = moved x-position of dbar point/nozzle point to geometric center.
3 = input NANGLE was inappropriate (set to absolute value or 1000).
4 = Points entered in wrong direction, routine aborted. Must enter points from left to right.
ERROR FLAG RETURNED BY SDAREA. IF IFLAG = 1, THEN AN EXPLANATION OF THE PROBLEM IS GIVEN AT TIME OF THE ERROR.

0 = NO ERRORS.
1 = THERE WAS AT LEAST ONE VERTICAL LINE WITHIN PLANFORM DATA. (SECOND VALUE PERTURBED BY .05 UNITS).
2 = NOZZLE HAVE BEEN PLACED EITHER IN FRONT OF OR BEHIND THE AIRCRAFT (ROUTINE ABORTED).
3 = INPUT NSLICE WAS INAPPROPRIATE (SET TO ABSOLUTE VALUE OR 1000).
4 = POINTS ENTERED IN WRONG DIRECTION, ROUTINE ABORTED. MUST ENTER POINTS FROM LEFT TO RIGHT.
5 = 1ST AND LAST POINTS NOT ON X-AXIS, ROUTINE ABORTED.
6 = THE NUMBER OF SLICES THAT WERE ADDED BY COMPUTER EXCEEDS MAXIMUM VALUE OF 2000 INTERNAL SLICES. DECREASE NUMBER OF SLICES.

Thrust variables (T_F, T_R, T_FR, TT_F, TT_R) were entered in the wrong combination

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>AJ</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
<td>Area of front and rear nozzles.</td>
</tr>
<tr>
<td>DC</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Dbar of the configuration used</td>
</tr>
<tr>
<td>DF</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Diameter of individual front jet.</td>
</tr>
<tr>
<td>DE</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Equivalent diameter of all jets.</td>
</tr>
<tr>
<td>DH</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Wing height above nozzles</td>
</tr>
<tr>
<td>HEIGHT</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Altitude at which calculations are to be made.</td>
</tr>
<tr>
<td>HLTF</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td>Hover lift gain due to fountain effects</td>
</tr>
<tr>
<td>HPRIME</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td>H' calculated from HOVPIE</td>
</tr>
<tr>
<td>KBL</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td>Boundry layer factor</td>
</tr>
<tr>
<td>LP</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td>Distance between front and rear nozzles</td>
</tr>
<tr>
<td>LTS</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td>Hover lift loss due to suckdown.</td>
</tr>
<tr>
<td>N</td>
<td>I</td>
<td>I</td>
<td>----</td>
<td>Total number of nozzles.</td>
</tr>
<tr>
<td>NF</td>
<td>I</td>
<td>I</td>
<td>----</td>
<td>Total number of front nozzles.</td>
</tr>
<tr>
<td>PR</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td>Pressure ratio of nozzles.</td>
</tr>
<tr>
<td>Q</td>
<td>R</td>
<td>I</td>
<td>lb/Sq Ft</td>
<td>Average dynamic pressure of nozzles.</td>
</tr>
<tr>
<td>WL</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td>Average width to length of front and rear nozzles</td>
</tr>
<tr>
<td>YF</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td>Distance between front nozzles.</td>
</tr>
</tbody>
</table>
### PIEWK Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFAC</td>
<td>R</td>
<td>I</td>
<td></td>
<td>Area ratios used in calculating the lift loss due to the jet wake.</td>
</tr>
<tr>
<td>AJ</td>
<td>R</td>
<td>I</td>
<td>Sq. Ft</td>
<td>Total area of the jets in question.</td>
</tr>
<tr>
<td>AR</td>
<td>R</td>
<td>I</td>
<td></td>
<td>Aspect ratio of the planform.</td>
</tr>
<tr>
<td>B</td>
<td>R</td>
<td>I</td>
<td></td>
<td>Width of current body config.</td>
</tr>
<tr>
<td>CU</td>
<td>R</td>
<td>I</td>
<td></td>
<td>Momentum coef. based on nozzle.</td>
</tr>
<tr>
<td>DH</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Vertical position of the jet based on the planform.</td>
</tr>
<tr>
<td>DI</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Diameter of the jets in question.</td>
</tr>
<tr>
<td>FIGWK</td>
<td>C</td>
<td></td>
<td></td>
<td>Flag which notifies JIEPIE which planform is in use.</td>
</tr>
<tr>
<td>HEIGHT</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Height of aircraft</td>
</tr>
<tr>
<td>KL_F</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Adjustment factor used for flap extension</td>
</tr>
<tr>
<td>LT</td>
<td>R</td>
<td>O</td>
<td></td>
<td>Lift loss returned by STOLWK</td>
</tr>
<tr>
<td>LT0</td>
<td>R</td>
<td>-</td>
<td></td>
<td>Lift loss out of ground effect.</td>
</tr>
<tr>
<td>LT_FLP</td>
<td>R</td>
<td>O</td>
<td></td>
<td>Basic lift gain due to jet flap action.</td>
</tr>
<tr>
<td>LT_OG</td>
<td>R</td>
<td>O</td>
<td></td>
<td>Out of ground lift loss</td>
</tr>
<tr>
<td>MC</td>
<td>R</td>
<td>I</td>
<td></td>
<td>Mean Aerodynamic Chord.</td>
</tr>
<tr>
<td>N</td>
<td>I</td>
<td>I</td>
<td></td>
<td>Number of nozzles based on which nozzle used (Front, Rear)</td>
</tr>
<tr>
<td>PR</td>
<td>R</td>
<td>I</td>
<td></td>
<td>Pressure Ratio of local jets.</td>
</tr>
<tr>
<td>Q</td>
<td>R</td>
<td>I</td>
<td>lb/Sq Ft</td>
<td>Jet dynamic pressure</td>
</tr>
<tr>
<td>Q0</td>
<td>R</td>
<td>O</td>
<td>lb/Sq Ft</td>
<td>Free stream dynamic pressure.</td>
</tr>
<tr>
<td>S</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
<td>Area of the configuration.</td>
</tr>
<tr>
<td>SF</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
<td>Area of configuration that is in front of nozzles.</td>
</tr>
<tr>
<td>T</td>
<td>R</td>
<td>I</td>
<td>lb</td>
<td>Thrust of the jets in question.</td>
</tr>
<tr>
<td>TEFALG</td>
<td>L</td>
<td></td>
<td></td>
<td>Flag which notifies STOLWK how close nozzle is to the trailing edge of the wing.</td>
</tr>
<tr>
<td>WL</td>
<td>R</td>
<td>I</td>
<td></td>
<td>Width to length ratio of nozzles.</td>
</tr>
<tr>
<td>XC</td>
<td>R</td>
<td>-</td>
<td></td>
<td>Position of nozzle with respect to wing trailing edge (in % chord)</td>
</tr>
<tr>
<td>Y</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Lateral spacing of jets.</td>
</tr>
<tr>
<td>YP</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Distance of jet center to body side.</td>
</tr>
<tr>
<td>VEFAC</td>
<td>R</td>
<td>I</td>
<td></td>
<td>Effective velocity factor which accounts for reduction in velocity at rear nozzles due to front nozzle.</td>
</tr>
</tbody>
</table>

### RESPIE Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>I</td>
<td>O</td>
<td></td>
<td>Angle of attack number counter</td>
</tr>
<tr>
<td>AOA(500)</td>
<td>R</td>
<td>I</td>
<td>deg</td>
<td>Actual Angle Of Attack value.</td>
</tr>
<tr>
<td>D</td>
<td>I</td>
<td>O</td>
<td></td>
<td>Nozzle deflection angle number counter.</td>
</tr>
<tr>
<td>DFL(500)</td>
<td>R</td>
<td>I</td>
<td>deg</td>
<td>Actual nozzle deflection angle value</td>
</tr>
<tr>
<td>GV_LT</td>
<td>R</td>
<td>O</td>
<td></td>
<td>Total lift loss due to the ground vortex.</td>
</tr>
<tr>
<td>Description</td>
<td>I/O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------</td>
<td>-----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lift loss due to the ground vortex effect on the body alone</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lift loss due to the ground vortex effect on the wing alone</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Height number counter</td>
<td>O Ft</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Height to which tangent line intersects $\Delta L/T=0$ (Used in Hover hprime method, obviously)</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total lift loss calculated while in hover.</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lift gain due to fountain effects while in hover.</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lift gain due to the addition of LIDs while in hover.</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lift loss out of ground effect while in hover.</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lift loss out of ground effect while in hover based on deflection angle.</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lift loss due to suckdown while in hover.</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual height value</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total number of Angle of Attack values</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total number of nozzle deflection angles</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total number of height values</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total number of velocity values</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Record number for any file which replaces a $3 \times 3$ matrix</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Record number for any file which replaces a $4 \times 4$ matrix</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total lift loss from the Reaction Control System.</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total lift/gain due to the suckdown and fountain effects while in forward flight.</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lift gain due to fountain effects while in forward flight.</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lift loss due to suckdown effects while in forward flight</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Velocity number counter</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual velocity value</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total lift loss due to the jet wake system produced in forward flight</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lift loss due to jet wake system produced on the body.</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lift loss due to jet wake system produce on the wing.</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**FILES USED:**

<table>
<thead>
<tr>
<th>Logical Unit</th>
<th>I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>68</td>
<td>O</td>
</tr>
<tr>
<td>70</td>
<td>O</td>
</tr>
<tr>
<td>71</td>
<td>O</td>
</tr>
</tbody>
</table>

- Sequential file for table output
- Direct access file for storage of lift increment due to suckdown
- Direct access file for storage of lift increment due to fountain
Direct access file for storage of lift increment due to the ground vortex on the body
Direct access file for storage of lift increment due to the ground vortex on the wing
Direct access file for storage of lift increment due to the jet wake on the body
Direct access file for storage of lift increment due to the jet wake on the wing (without center section)

COMMONS USED: (All common blocks used throughout PIE)

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONPIE</td>
<td>Controls execution and coordination of Power Induced Effects Module</td>
</tr>
<tr>
<td>FIGPIE</td>
<td>Sets defaults for variables and reads user-defined input file</td>
</tr>
<tr>
<td>HOVPIE</td>
<td>Modifies planforms for consistency and integrity</td>
</tr>
<tr>
<td>PIEFLAGS</td>
<td>Calculates all variables which have not been defined by the user or set to a default value</td>
</tr>
<tr>
<td>PIEGV</td>
<td>Calculates areas necessary to make calculations for fountain effect</td>
</tr>
<tr>
<td>PIEERROR</td>
<td>Calculates angular mean diameter of a planform</td>
</tr>
<tr>
<td>PIEESF</td>
<td>Isolates and controls execution of HOV_GE</td>
</tr>
<tr>
<td>PIEWK</td>
<td>Calculates hover lift increments for OGE and GE suckdown, and fountain effects</td>
</tr>
<tr>
<td>POLYGON</td>
<td>Calculates stationary lift increments while</td>
</tr>
<tr>
<td>RESPIE</td>
<td>Controls execution and coordination of Power Induced Effects Module</td>
</tr>
</tbody>
</table>

List of all routines:

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPPIE</td>
<td>Controls execution and coordination of Power Induced Effects Module</td>
</tr>
<tr>
<td>INPIE</td>
<td>Sets defaults for variables and reads user-defined input file</td>
</tr>
<tr>
<td>PLANMO</td>
<td>Modifies planforms for consistency and integrity</td>
</tr>
<tr>
<td>FINVAR</td>
<td>Calculates all variables which have not been defined by the user or set to a default value</td>
</tr>
<tr>
<td>SDAREA</td>
<td>Calculates areas necessary to make calculations for fountain effect</td>
</tr>
<tr>
<td>DIABAR</td>
<td>Calculates angular mean diameter of a planform</td>
</tr>
<tr>
<td>HCALL</td>
<td>Isolates and controls execution of HOV_GE</td>
</tr>
<tr>
<td>HOV_GE</td>
<td>Calculates hover lift increments for OGE and GE suckdown, and fountain effects</td>
</tr>
<tr>
<td>SFCALL</td>
<td>Isolates and controls execution of STOLSF</td>
</tr>
<tr>
<td>STOLSF</td>
<td>Calculates suckdown and fountain lift increments while</td>
</tr>
</tbody>
</table>
in STOL flight

GVCALL: Isolates and controls execution of STOLGV
STOLGV: Calculates ground vortex lift increments while in STOL flight
WKCALL: Isolates and controls execution of STOLWK
JIEPIE: Calculates change in lift caused by the jet induce effects on a flat plate
STOLWK: Calculates jet wake lift increments while in STOL flight
RCSCAL: Isolates and controls execution of RCSIND
RCSIND: Calculates RCS lift increments while in forward flight
PIEP: Printing routine for PIE
JIEETAB: Creates output file which is used by ACSYNT
CENTAR: Calculates the center of area, area and area in front of a point for a given planform
CROSSIN: Calculates the intersection of two lines, with each line defined by two points
DIST (FUNC): Calculates the distance between two points
INTEGRA: Calculates the areas of thin rectangles
LINECRO: Calculates the intersection of two lines, each defined by a slope and Y-intercept
LINTERP: Linear interpolates between two X and Y values give an intermediate X value
PERPDIS: Calculates the perpendicular distance between a point and a line
POLYAR: Calculates the area of any polygon
RTOTAL: Sums the total lift increment given array type indices
SEN: Calculates start, stop, end, and number variables which have not been defined

Routines Called:

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPIE</td>
<td>Sets defaults for variables and reads user defined input file</td>
</tr>
<tr>
<td>PLANMO</td>
<td>Modifies planforms for consistecy and integrity</td>
</tr>
<tr>
<td>FINV</td>
<td>Calculates all variables which have not been defined by the user or set to a default value</td>
</tr>
<tr>
<td>HCALL</td>
<td>Isolates and controls execution of HOV GE</td>
</tr>
<tr>
<td>RCSCAL</td>
<td>Isolates and controls execution of RCSIND</td>
</tr>
<tr>
<td>SFCALL</td>
<td>Isolates and controls execution of STOLSF</td>
</tr>
<tr>
<td>WKCALL</td>
<td>Isolates and controls execution of STOLWK</td>
</tr>
<tr>
<td>GVCALL</td>
<td>Isolates and controls execution of STOLGV</td>
</tr>
<tr>
<td>PIEP</td>
<td>Printing routine for PIE</td>
</tr>
</tbody>
</table>

Notes: None.

References: (Used throughout the PIE routines)


Environment:

Fortran 77, VAX 6800, VAX 11/785, SGI IRIS 4D series.

Non-Standard Code:
#include "figpie.inc"
#include "pieflag.inc"
#include "conpie.inc"
#include "respie.inc"

INTEGER ICALC

C When ICALC is set to 1 then call the variable input subroutine, INPIE
C Set ICALC = 1 for testing purposes
ICALC = 1

C IF (ICALC .EQ. 1) THEN
C CALL INPIE
END IF

C Open necessary input, output, and scratch files
C Table Output File
OPEN (UNIT=68,STATUS='UNKNOWN')
C Forward velocity suckdown term
OPEN (UNIT=70,FORM='FORMATTED',
$ ACCESS='DIRECT',RECL=11,STATUS='SCRATCH')
C Forward velocity fountain term
OPEN (UNIT=71,FORM='FORMATTED',
$ ACCESS='DIRECT',RECL=11,STATUS='SCRATCH')
C Ground vortex for the Body
OPEN (UNIT=72,FORM='FORMATTED',
$ ACCESS='DIRECT',RECL=11,STATUS='SCRATCH')
C Ground vortex for the Wing
OPEN (UNIT=73,FORM='FORMATTED',
$ ACCESS='DIRECT',RECL=11,STATUS='SCRATCH')
C Jet wake for the Body
OPEN (UNIT=74,FORM='FORMATTED',
$ ACCESS='DIRECT',RECL=11,STATUS='SCRATCH')
C Jet wake for the Wing
OPEN (UNIT=75,FORM='FORMATTED',
$ ACCESS='DIRECT',RECL=11,STATUS='SCRATCH')
C The meat of the Power Induced Effects module.
C Set ICALC = 2 for testing purposes
ICALC = 2

C IF (ICALC .EQ. 2) THEN
C Make planform modifications
CALL PLANMO
C Finish calculating all initial variables
CALL FINVAR
C
C Start do loops
C Altitude (Ft)
DO H = 1, LH
HT(H) = HSTART + (H - 1) * HSTEP
Lift loss in hover
CALL HCALL

Velocity (kts)
DO V = 1, LV
   VO(V) = VSTART + (V - 1) * VSTEP
   IF (H .EQ. 1) CALL RCSCALL
   Jet Deflection Angle (Deg)

DO D = 1, LD
   DFL(D) = DSTART + (D - 1) * DSTEP
   STOL Suckdown and Fountain Calcs.
   CALL SFCALL
   CALL WKCALL

Angle of attack (Deg)
DO A = 1, LA
   AOA(A) = ASTART + (A - 1) * ASTEP
   CALL GVCALL
   MOCALL not implemented yet.
   CALL MOCALL
END DO

END DO

END DO

Set H, D, V & A to its last value
H = H - 1
D = D - 1
V = V - 1
A = A - 1
END IF

Call PIEP subroutine to print out all variables.
Set ICALC = 3 for testing purposes
ICALC = 3

IF (ICALC .EQ. 3) CALL PIEP
Close all scratch files
CLOSE (70)
CLOSE (71)
CLOSE (72)
CLOSE (73)
CLOSE (74)
CLOSE (75)

END

Inpie

SUBROUTINE INPIE

ACRONYM: INput routine for Powered Induced Effects.
C PURPOSE: The purpose of INPIE is to input all the variables that the
user has placed in the input file with the extension .PIE.

C LOCAL VARIABLES (in addition to the above parameters):

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>I</td>
<td>I</td>
<td>---</td>
<td>Counter</td>
</tr>
<tr>
<td>HEADER</td>
<td>C*8</td>
<td>I</td>
<td>---</td>
<td>Flag which defines which planforms are contained within *.pie file</td>
</tr>
</tbody>
</table>

C GLOBAL VARIABLES (in addition to the above parameters and local vars):

C CONPIE Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEND</td>
<td>R</td>
<td>I</td>
<td>deg</td>
<td>Ending value for angle of attack</td>
</tr>
<tr>
<td>ASTART</td>
<td>R</td>
<td>I</td>
<td>deg</td>
<td>Starting value for angle of attack</td>
</tr>
<tr>
<td>ASTEP</td>
<td>R</td>
<td>I</td>
<td>deg</td>
<td>Step value for angle of attack</td>
</tr>
<tr>
<td>DEND</td>
<td>R</td>
<td>I</td>
<td>deg</td>
<td>Ending value for jet deflection angle</td>
</tr>
<tr>
<td>DSTART</td>
<td>R</td>
<td>I</td>
<td>deg</td>
<td>Starting value for jet deflection angle</td>
</tr>
<tr>
<td>DSTEP</td>
<td>R</td>
<td>I</td>
<td>deg</td>
<td>Step value for jet deflection angle</td>
</tr>
<tr>
<td>HEND</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Ending value for altitude</td>
</tr>
<tr>
<td>HSTART</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Starting value for altitude</td>
</tr>
<tr>
<td>HSTEP</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Step value for altitude</td>
</tr>
<tr>
<td>VEND</td>
<td>R</td>
<td>I</td>
<td>kts</td>
<td>Ending value for aircraft velocity</td>
</tr>
<tr>
<td>VSTART</td>
<td>R</td>
<td>I</td>
<td>kts</td>
<td>Starting value for aircraft velocity</td>
</tr>
<tr>
<td>VSTEP</td>
<td>R</td>
<td>I</td>
<td>kts</td>
<td>Step value for aircraft velocity</td>
</tr>
</tbody>
</table>

C FIGPIE Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_B</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Width of Body</td>
</tr>
<tr>
<td>B_CS</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Width of Center Section</td>
</tr>
<tr>
<td>B_JP</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Width of Jet Pattern</td>
</tr>
<tr>
<td>B_W</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Width of Wing (Wing span)</td>
</tr>
<tr>
<td>B_WB</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Width of Wing-Body</td>
</tr>
<tr>
<td>CONFIG</td>
<td>C*18</td>
<td>I</td>
<td>-----</td>
<td>Short title of current config</td>
</tr>
<tr>
<td>D_F</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Diameter of each Front jet</td>
</tr>
<tr>
<td>D_R</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Diameter of each Rear jet</td>
</tr>
<tr>
<td>D_RCS</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Diameter of Roll RCS nozzle</td>
</tr>
<tr>
<td>D_B</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Dbar of Body</td>
</tr>
<tr>
<td>D_CS</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Dbar of Center Section</td>
</tr>
<tr>
<td>D_W</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Dbar of Wing</td>
</tr>
<tr>
<td>D_WB</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Dbar of Wing-Body</td>
</tr>
<tr>
<td>DE_F</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Effective Diameter of Front jets</td>
</tr>
<tr>
<td>DE_FR</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Effective Diameter of Front &amp; Rear jets combined</td>
</tr>
<tr>
<td>DE_R</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Effective Diameter of Rear jets</td>
</tr>
<tr>
<td>DR</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Density Ratio (Jet / Atm)</td>
</tr>
<tr>
<td>E_1</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Half distance between adjacent jets (1)</td>
</tr>
<tr>
<td>E_3</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Half distance between adjacent jets (3)</td>
</tr>
<tr>
<td>E_4</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Half distance between adjacent jets (4)</td>
</tr>
<tr>
<td>F_NAME</td>
<td>C*50</td>
<td>I</td>
<td>-----</td>
<td>Name of file which contains body &amp; wing planform coordinates</td>
</tr>
<tr>
<td>Variable</td>
<td>Type</td>
<td>Format</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>KB</td>
<td>R</td>
<td>I</td>
<td>Boundary layer factor</td>
<td></td>
</tr>
<tr>
<td>KLF</td>
<td>R</td>
<td>I</td>
<td>Adjustment factor for flap extension when calculating the lift loss due to jet induced effects.</td>
<td></td>
</tr>
<tr>
<td>KR</td>
<td>R</td>
<td>I</td>
<td>Body contour factor</td>
<td></td>
</tr>
<tr>
<td>L_B</td>
<td>R</td>
<td>I</td>
<td>Length of body</td>
<td></td>
</tr>
<tr>
<td>L_CS</td>
<td>R</td>
<td>I</td>
<td>Length of Center Section</td>
<td></td>
</tr>
<tr>
<td>L_F</td>
<td>R</td>
<td>I</td>
<td>Length of Front jet</td>
<td></td>
</tr>
<tr>
<td>L_R</td>
<td>R</td>
<td>I</td>
<td>Length of Rear jet</td>
<td></td>
</tr>
<tr>
<td>L_WB</td>
<td>R</td>
<td>I</td>
<td>Length of Wing-Body</td>
<td></td>
</tr>
<tr>
<td>MAC</td>
<td>R</td>
<td>I</td>
<td>Mean Aerodynamic Chord of wing</td>
<td></td>
</tr>
<tr>
<td>MD_RCS</td>
<td>R</td>
<td>I</td>
<td>Mass flow rate for one RCS nozzle</td>
<td></td>
</tr>
<tr>
<td>N_BODY</td>
<td>I</td>
<td>I</td>
<td>Number of data points for Body planform</td>
<td></td>
</tr>
<tr>
<td>N_WING</td>
<td>I</td>
<td>I</td>
<td>Number of data points for Wing-Body planform</td>
<td></td>
</tr>
<tr>
<td>N_WB</td>
<td>I</td>
<td>I</td>
<td>Number of data points for Wing-Body planform</td>
<td></td>
</tr>
<tr>
<td>N_DIV</td>
<td>I</td>
<td>I</td>
<td>Number of divisions to be used when calculating suckdown areas (SDAREA) and Dbars (DIABAR)</td>
<td></td>
</tr>
<tr>
<td>NUM</td>
<td>I</td>
<td>I</td>
<td>Total number of jets</td>
<td></td>
</tr>
<tr>
<td>NUM_F</td>
<td>I</td>
<td>I</td>
<td>Number of Front jets</td>
<td></td>
</tr>
<tr>
<td>NUM_R</td>
<td>I</td>
<td>I</td>
<td>Number of Rear jets</td>
<td></td>
</tr>
<tr>
<td>PER_FR</td>
<td>R</td>
<td>I</td>
<td>Total perimeter of jets</td>
<td></td>
</tr>
<tr>
<td>PP_LID</td>
<td>R</td>
<td>I</td>
<td>Ratio of perimeter enclosed by lids to total perimeter</td>
<td></td>
</tr>
<tr>
<td>PR_F</td>
<td>R</td>
<td>I</td>
<td>Jet Pressure Ratio for front jets</td>
<td></td>
</tr>
<tr>
<td>PR_FR</td>
<td>R</td>
<td>I</td>
<td>Jet Pressure Ratio for all jets</td>
<td></td>
</tr>
<tr>
<td>PR_R</td>
<td>R</td>
<td>I</td>
<td>Jet Pressure Ratio for rear jets</td>
<td></td>
</tr>
<tr>
<td>PR_RCS</td>
<td>R</td>
<td>I</td>
<td>Roll RCS Pressure Ratio</td>
<td></td>
</tr>
<tr>
<td>PT_RCS</td>
<td>R</td>
<td>I</td>
<td>Total pressure for one roll RCS jet</td>
<td></td>
</tr>
<tr>
<td>Q_F</td>
<td>R</td>
<td>I</td>
<td>Dynamic pressure for the front jets</td>
<td></td>
</tr>
<tr>
<td>Q_FR</td>
<td>R</td>
<td>I</td>
<td>Dynamic pressure for the rear jets</td>
<td></td>
</tr>
<tr>
<td>Q_R</td>
<td>R</td>
<td>I</td>
<td>Dynamic pressure for roll RCS jets</td>
<td></td>
</tr>
<tr>
<td>Q_RCS</td>
<td>R</td>
<td>I</td>
<td>Corner radius of body sides</td>
<td></td>
</tr>
<tr>
<td>R_B</td>
<td>R</td>
<td>I</td>
<td>Area of Body</td>
<td></td>
</tr>
<tr>
<td>S_B</td>
<td>R</td>
<td>I</td>
<td>Area of Center Section</td>
<td></td>
</tr>
<tr>
<td>S_CS</td>
<td>R</td>
<td>I</td>
<td>Area of Front jets</td>
<td></td>
</tr>
<tr>
<td>S_F</td>
<td>R</td>
<td>I</td>
<td>Area affected by fountain arm (1)</td>
<td></td>
</tr>
<tr>
<td>S_FA1</td>
<td>R</td>
<td>I</td>
<td>Area affected by fountain arm (3)</td>
<td></td>
</tr>
<tr>
<td>S_FA3</td>
<td>R</td>
<td>I</td>
<td>Area affected by fountain arm (4)</td>
<td></td>
</tr>
<tr>
<td>S_FA4</td>
<td>R</td>
<td>I</td>
<td>Total jet exit area</td>
<td></td>
</tr>
<tr>
<td>S_FR</td>
<td>R</td>
<td>I</td>
<td>Area enclosed by Jet Pattern</td>
<td></td>
</tr>
<tr>
<td>S_JP</td>
<td>R</td>
<td>I</td>
<td>Area enclosed by LIDS</td>
<td></td>
</tr>
<tr>
<td>S_LID</td>
<td>R</td>
<td>I</td>
<td>Area of Rear jets</td>
<td></td>
</tr>
<tr>
<td>S_R</td>
<td>R</td>
<td>I</td>
<td>Area of Wing</td>
<td></td>
</tr>
<tr>
<td>S_W</td>
<td>R</td>
<td>I</td>
<td>Area of Wing-Body</td>
<td></td>
</tr>
<tr>
<td>S_WB</td>
<td>R</td>
<td>I</td>
<td>Actual scale factor which adjusts area of fountain arm (3) to account for the fact that curvature of aircraft's nose tends not to hold fountain arm very well causing area that fountain arm affects to be scaled down.</td>
<td></td>
</tr>
<tr>
<td>SCALE</td>
<td>R</td>
<td>I</td>
<td>Fountain arm (3) scaler (Percentage)</td>
<td></td>
</tr>
<tr>
<td>SCALE3</td>
<td>R</td>
<td>I</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
measured from the front jets to the
nose of the aircraft.) Tries to
approximate SCALE but does not do a
great job (better than nothing.)

Area ahead of Front jets using body
planform
Area ahead of Rear jets using the
wing planform
Area ahead of Rear jets using the
wingbody planform

Actual surface area within area
enclosed by nozzles

Front Thrust / total Thrust
Rear Thrust / total Thrust

Width of Front jet
Width of Rear jets

Flow Weight of Inlet / Flow Weight
of Exit

Width to Length ratio of Body
Width to Length ratio of Center
Section

Width to Length ratio of Front jets
Width to Length ratio of Jet
Pattern

Width to Length ratio of Rear jets
Width to Length ratio of Wing-Body

X-coordinates of Body planform
Distance between Front & Rear jets
Distance of roll RCS nozzle ahead
der wing trailing edge

X-coordinates of Wing-Body planform
X-coordinates of Wing planform
Distance of center of area ahead of
CG

X-coordinate of Center of Area
### X-Coordinate of Center of Gravity
Distance from CG to MAC/2
Distance from CG to MAC/4
Distance Front jet is ahead of CG
Inlet longitudinal distance ahead of CG
Distance Rear jet is ahead of CG
X-coordinates of Front NOZZle
X-coordinates of Rear NOZZle
X-coordinate of trailing edge for front nozzle (root TE if Nozzle within body)
X-coordinate of trailing edge for rear nozzle (root TE if Nozzle within body)
Spanwise extent of planform on fountain arm (1) center line.
Spanwise extent of planform on fountain arm (3) center line.
Spanwise extent of planform on fountain arm (4) center line.
Y-coordinates of Body planform
Distance between Front jets
Y-coordinates of Front NOZZle
Y-coordinates of R NOZZle
Distance between Rear jets
Distance of roll RCS nozzle in from wingtip
Y-coordinates of Wing-Body planform
Y-coordinates of Wing planform
Lateral distance from Body to Front jets (for external jets)
Lateral distance from Body to Rear jets (for external jets)
Max spanwise extent of planform (1)
Max spanwise extent of planform (3)
Max spanwise extent of planform (4)
Lateral distance from Wingbody to Front jets (for external jets)
Lateral distance from wingbody to Rear jets (for external jets)
Height of body base above nozzle
Inlet vertical distance above CG
Height of wing above nozzle

---

### DESCRIPTION

Signals when to output tables based on height or height/De
(TRUE - Print based on Height/DE, FALSE - Print based on height)
Signals when to output tables based on VE
TRUE - Print Based on VE
FALSE - Print Based on Velocity

Identifies when WingBody planform has been enter by the user. Used
to determine when to use the wingbody incalculations.

### RESPIE Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Total number of Angle of Attack values</td>
</tr>
<tr>
<td>LD</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Total number of nozzle deflection angles values.</td>
</tr>
<tr>
<td>LH</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Total number of height values.</td>
</tr>
<tr>
<td>LV</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Total number of velocity values.</td>
</tr>
</tbody>
</table>

### FLAG VARIABLES (in addition to the above parameters and local vars):

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### FILES USED:

<table>
<thead>
<tr>
<th>LOGICAL UNIT</th>
<th>I/O</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>I</td>
<td>Data file containing planform coordinates of body &amp; wing separately or wingbody coordinates only.</td>
</tr>
<tr>
<td>10</td>
<td>I</td>
<td>Data file containing namelist PIENAM variables.</td>
</tr>
</tbody>
</table>

### COMMONS USED:

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONPIE</td>
<td>CONtrol variables for Power Induced Effects - Contains variables which control the execution of the PIE module.</td>
</tr>
<tr>
<td>FIGPIE</td>
<td>conFIGuration for Power Induced Effects - Contains all variables needed to define the configuration and other parameters of the aircraft.</td>
</tr>
<tr>
<td>PIEFLAGS</td>
<td>Power Induced Effects FLAGS - Contains variables which help keep track of the configuration of the aircraft.</td>
</tr>
<tr>
<td>RESPIE</td>
<td>RESults from all POwer Induced Effects subroutines - Contains results from each subroutine along with variables for height, velocity, jet deflection angle, and angle of attack and their counters.</td>
</tr>
</tbody>
</table>

### CALLED BY:

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPPIE</td>
<td>Controls execution and coordination of PIE</td>
</tr>
</tbody>
</table>

### ROUTINES CALLED:

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### NOTES: None.

### REFERENCES:

1. None.

### ENVIRONMENT:

FORTRAN 77, VAX 6800, VAX 11/785, SGI IRIS 4D series.

### NON-STANDARD CODE:

? 

### AUTHOR(S):

Kipp E. Howard (KEH), Cal Poly San Luis Obispo, NASA Ames Moffett

### REVISION HISTORY:

<table>
<thead>
<tr>
<th>DATE</th>
<th>INITIALS &amp; DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>04/03/90</td>
<td>KEH -- Initial coding complete.</td>
</tr>
<tr>
<td>07/13/91</td>
<td>KEH -- Eliminated extraneous spacing and fixed comments</td>
</tr>
</tbody>
</table>
# include "conpie.inc"
# include "figpie.inc"
# include "pieflag.inc"
# include "respie.inc"

C

C Variable Declarations

CHARACTER HEADER*8

C

C Define namelist PIENAME

  B_BB, DB_BB, L_BB, S_BB, WL_BB,
  B_JP, PP_LID, S_JP, S_LID, SP_JP, WL_JP,
  B_CS, DB_CS, L_CS, S_CS, WL_CS,
  D_F, DE_F, L_F, NUM_F, PR_F, Q_F, S_F, SF_FB,
  SF_FB, SPLY_F, T_F, TT_F, W_F, WL_F, XCG_F, XNOZ_F,
  Y_F, YB_F, YNOZ_F, XTE_F, YWB_F,
  DE_R, L_R, NUM_R, Q_R, S_R, SF_RB,
  Y_R, YB_R, YNOZ_R, XTE_R, YWB_R,
  DE_FR, FNAME, K, KF, PR_FR, Q_FR, S_FR, T_FR, X_FR,
  DR, KLF, PR_FR, WIWE, XCG_I, XCG_I, CONFIG,
  D_RCS, MD_RCS, PR_RCS, PT_RCS, T_RCS,
  TP_RCS, Q_RCS, X_RCS, Y_RCS,
  KB, NDIV, X_CG, XCG_C2, XCG_C4,
  F_NAME, HDOUT, VEOUT,
  SF_A1, S_FA3, S_FA4, SP_FA1, SP_FA3, SP_FA4, SCALE3, SCALE,
  E_1, E_3, E_4, THA_1, THA_3, THA_4, Y_1, Y_3, Y_4,
  YF_1, YP_3, YP_4,
  HSTART, HEND, HSTEP, LH, VSTART, VEND, VSTEP, LV,
  DSTART, DEND, DSTEP, LD, ASTART, AEND, ASTEP, LA, SKIP

C

C Initialization of all variables:

Variable that has been set to 9999. is a variable which will be calculated if the user does not set it through the input file.

AEND = 9999.
ASTART = 9999.
ASTEP = 9999.
B_B = 9999.
B_CS = 9999.
B_JP = 9999.
B_W = 9999.
B_BB = 9999.
CONFIG = '????????????????????
D_F = 9999.
D_R = 9999.
D_RCS = 9999.
DE_B = 9999.
DE_CS = 9999.
DE_W = 9999.
DE_BB = 9999.
DE_F = 9999.
DE_FR = 9999.
DE_R = 9999.
DR = 1.0
DEND = 9999.
DSTART = 9999.
DSTEP = 9999.
E1 = 9999.
E3 = 9999.
E4 = 9999.
F.NAME = 'test.pie'
HDEOUT = .TRUE.
HEND = 9999.
HSTART = 9999.
HSTEP = 9999.
KB = .666667
KLF = 1.0
KR = 9999.
LB = 9999.
LC = 9999.
LF = 9999.
LR = 9999.
LWB = 9999.
LH = 9999
LV = 9999
LD = 9999
LA = 9999
MAC = 0.0
MD.RCS = 0.0
NDIV = 500
NUM = 9999
NUM.F = 9999
NUM.R = 9999
PER.FR = 9999.
PLID = 9999.
PF = 9999.
PF.FR = 9999.
PR.R = 9999.
PR.RCS = 0.0
PRFLG = .TRUE.
PT.RCS = 9999.
QF = 9999.
QR = 9999.
Q.FR = 9999.
Q.RCS = 9999.
RB = 9999.
SB = 9999.
SCS = 9999.
SJF = 9999.
SF = 9999.
SFA1 = 9999.
SFA3 = 9999.
SFA4 = 9999.
SF.R = 9999.
SLID = 0.0
SR = 9999.
SW = 9999.
SWB = 9999.
SCALE = 9999.
SCALE3 = 1.0
SF_FB = 9999.
SF FWB = 9999.
SF_RB = 9999.
SF R WB = 9999.
SKIP = .FALSE.
SP_JP  =  9999.
SP_FA1 =  9999.
SP_FA3 =  9999.
SP_FA4 =  9999.
SPLY_F =  0.0
SPLY_R =  0.0
T_F   =  9999.
T_FR  =  9999.
T_R   =  9999.
T_RCCS =  0.0
THA_1 =  9999.
THA_3 =  9999.
THA_4 =  9999.
TP_RCCS =  0.0
TT_F  =  9999.
TT_R  =  9999.
VEND  =  9999.
VEOUT  = .FALSE.
VSTART =  9999.
VSTEP  =  9999.
W_F   =  9999.
W_R   =  9999.
WFLAG = .FALSE.
WWE   =  0.0
WL_B  =  9999.
WL_CS  =  9999.
WL_JP =  9999.
WL_F  =  9999.
WL_R  =  9999.
WL_WB =  9999.
X_FR  =  9999.
X_RCCS =  9999.
XCA_CG =  9999.
X_CG  =  0.0
X_CA  =  9999.
XCG_C2 =  9999.
XCG_C4 =  0.0
XCG_F =  9999.
XCG_I  =  0.0
XCG_R  =  9999.
XNOZ_F =  0.0
XNOZ_R =  9999.
XTE_F =  9999.
XTE_R =  9999.
Y_1   =  9999.
Y_3   =  9999.
Y_4   =  9999.
YP_1  =  9999.
YP_3  =  9999.
YP_4  =  9999.
YNOZ_F =  0.0
YNOZ_R =  9999.
Y_F   =  9999.
Y_R   =  9999.
Y_RCCS =  9999.
YB_F  =  9999.
YB_R  =  9999.
YWB_F =  9999.
YWE_R = 9999.
Z_E = 0.0
ZCG_I = 0.0
Z_W = 0.0

C Read user entered data in PIENAM Name list
READ (9, NML=PIENAM)

C Read in planform coordinates from F_NAME. Iris is case sensitive
C when it comes to naming files.
C
OPEN(UNIT = 10, FILE = F_NAME, TYPE = 'OLD')

C Set up loop to read planform coordinates
I = 1

C Read header for planform coordinates.
READ (10,7,END=50) HEADER
7 FORMAT (A8)

IF (HEADER .EQ. 'BODYPLAN') THEN
WBFLAG = .FALSE.

READ (10,*) X_BODY(I), Y_BODY(I)
C Upon reaching the flag (9999.) for X_BODY, go to next planform.
IF (X_BODY(I) .EQ. 9999.) GO TO 40

C Keep track of the number of body points
N_BODY = I
C
Increment counter
I = I + 1
GO TO 10

ELSE IF (HEADER .EQ. 'WINGPLAN') THEN
WBFLAG = .FALSE.

READ (10,*) X_WING(I), Y_WING(I)
C Upon reaching the flag (9999.) for X_WING, go to next planform.
IF (X_WING(I) .EQ. 9999.) GO TO 40

C Keep track of the number of wing points
N_WING = I
C
Increment counter
I = I + 1
GO TO 20

ELSE IF (HEADER .EQ. 'WINGBODY') THEN
WBFLAG = .TRUE.

READ (10,*) X WB(I), Y WB(I)
C Upon reaching the flag (9999.) for X_WB, go to next planform.
IF (X_WB(I) .EQ. 9999.) GO TO 40

C Keep track of the number of wingbody points
N WB = I
C
Increment counter
I = I + 1
GO TO 30

END IF
40

GO TO 5
C

Assign the correct value to WBFLAG. WBFLAG is true if only the
C wingbody is entered alone.
C
50 CONTINUE
SUBROUTINE PLANMO

ACRONYM: PLANform MOdifications.

PURPOSE: The purpose of PLANMO is to
1) Adjust any two planform coordinates that have the same X-coordinate. (This is because the slope of a line between two points with the same X-coordinate is infinity.)
2) Combine the individual body and wing planforms into one planform for the wingbody.
3) Move the most forward part of the body planform (generally the nose) to the origin of the coordinate system and move the wing planform likewise.

LOCAL VARIABLES:

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP(20)</td>
<td>I</td>
<td>-</td>
<td>----</td>
<td>Body point counter which keep track of which body point a crossing has taken place before.</td>
</tr>
<tr>
<td>CROSS(20)</td>
<td>I</td>
<td>-</td>
<td>----</td>
<td>Flag which identifies when a crossing point has been reached.</td>
</tr>
<tr>
<td>DELTAX</td>
<td>R</td>
<td>Ft</td>
<td></td>
<td>Amount to add to a point that has the same X-coordinate as the previous point. (= .1% of the total length of the configuration)</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>----</td>
<td></td>
<td>Counter for body</td>
</tr>
<tr>
<td>J</td>
<td>I</td>
<td>----</td>
<td></td>
<td>Counter for wing</td>
</tr>
<tr>
<td>K</td>
<td>I</td>
<td>----</td>
<td></td>
<td>Counter for wingbody</td>
</tr>
<tr>
<td>L</td>
<td>I</td>
<td>----</td>
<td></td>
<td>Counter for center section</td>
</tr>
<tr>
<td>M</td>
<td>I</td>
<td>----</td>
<td></td>
<td>Crossing counter</td>
</tr>
<tr>
<td>W_FLAG</td>
<td>I</td>
<td>I</td>
<td>----</td>
<td>Flag which signifies when to use the wing planform when creating the wingbody planform.</td>
</tr>
<tr>
<td>WP(20)</td>
<td>I</td>
<td>-</td>
<td>----</td>
<td>Wing point counter which keep track of which wing point a crossing has taken place before.</td>
</tr>
<tr>
<td>XCR(20)</td>
<td>R</td>
<td>-</td>
<td>----</td>
<td>X-coordinate at which M crossing occurs</td>
</tr>
<tr>
<td>XCROSS</td>
<td>R</td>
<td>Ft</td>
<td></td>
<td>The X-coordinate of the intersection of two lines defined by two body points and two wing points</td>
</tr>
<tr>
<td>YCR(20)</td>
<td>R</td>
<td>-</td>
<td>----</td>
<td>Y-coordinate at which M crossing occurs</td>
</tr>
<tr>
<td>YCROSS</td>
<td>R</td>
<td>Ft</td>
<td></td>
<td>See XCROSS (Y-coordinate)</td>
</tr>
<tr>
<td>XMAX</td>
<td>R</td>
<td>Ft</td>
<td></td>
<td>Most aft point of configuration</td>
</tr>
<tr>
<td>XMINT</td>
<td>R</td>
<td>Ft</td>
<td></td>
<td>Most forward point of the configuration.</td>
</tr>
</tbody>
</table>

GLOBAL VARIABLES (in addition to the above parameters and local vars):

FIGPIE Common Block
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>N_BODY</td>
<td>I</td>
<td>I</td>
<td></td>
<td>Number of data points for Body planform</td>
</tr>
<tr>
<td>N_CS</td>
<td>I</td>
<td>O</td>
<td></td>
<td>Number of data points for Center Section planform</td>
</tr>
<tr>
<td>N_WING</td>
<td>I</td>
<td>I</td>
<td></td>
<td>Number of data points for Wing planform</td>
</tr>
<tr>
<td>N_WB</td>
<td>I</td>
<td>I</td>
<td></td>
<td>Number of data points for Wing-Body planform</td>
</tr>
<tr>
<td>X_BODY(500)</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>X-coordinates of Body planform</td>
</tr>
<tr>
<td>X_CS(30)</td>
<td>R</td>
<td>O</td>
<td>Ft</td>
<td>X-coordinates of Center Section planform</td>
</tr>
<tr>
<td>X_RCS</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Distance of roll RCS nozzle ahead of wing trailing edge</td>
</tr>
<tr>
<td>X_WB(500)</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>X-coordinates of Wing-Body planform</td>
</tr>
<tr>
<td>X_WING(500)</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>X-coordinates of Wing planform</td>
</tr>
<tr>
<td>X_CA</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>X-coordinate of Center of Area</td>
</tr>
<tr>
<td>X_CG</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>X-coordinate of Center of Gravity</td>
</tr>
<tr>
<td>XNOZ_F</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>X-coordinates of Front NOZZle</td>
</tr>
<tr>
<td>XNOZ_R</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>X-coordinates of Rear NOZZle</td>
</tr>
<tr>
<td>XTE_F</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>X-coordinate of trailing edge for front nozzle (root TE if Nozzle within body)</td>
</tr>
<tr>
<td>XTE_R</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>X-coordinate of trailing edge for rear nozzle (root TE if Nozzle within body)</td>
</tr>
<tr>
<td>Y_CS(30)</td>
<td>R</td>
<td>O</td>
<td>Ft</td>
<td>Y-coordinates of Center Section</td>
</tr>
</tbody>
</table>

**PIEFLAGS** Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>WBFLAG</td>
<td>L</td>
<td>O</td>
<td></td>
<td>Identifies when WingBody planform has been entered by the user. Used to determine when to use the wingbody in calculations.</td>
</tr>
</tbody>
</table>

**FILES USED:**

<table>
<thead>
<tr>
<th>LOGICAL UNIT</th>
<th>I/O</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>(none)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**COMMONS USED:**

**CALLED BY:**

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPPIE</td>
<td>Controls execution and coordination of PIE</td>
</tr>
</tbody>
</table>

**ROUTINES CALLED:**

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CROSSING</td>
<td>Calculates the intersection of two lines defined by four points. Line one is defined with (X1,Y1) &amp; (X2,Y2).</td>
</tr>
</tbody>
</table>
Line two is defined with (X3,Y3) & (X4,Y4).

C NOTES: None.
C REFERENCES:
C 1) None.
C ENVIRONMENT:
C FORTRAN 77, VAX 6800, VAX 11/785, SGI IRIS 4D series.
C NON-STANDARD CODE:
C ?
C AUTHOR(S):
C Kipp E. Howard (KEH), Cal Poly San Luis Obispo, NASA Ames Moffett
C REVISION HISTORY:
C DATE INITIALS & DESCRIPTION
C 03/15/90 KEH -- Initial coding complete.
C 07/13/91 KEH -- Eliminated extraneous spacing and fixed comments

#include "figpie.inc"
#include "pieflag.inc"

C Variable declarations
REAL CROSS(20), XCR(20), YCR(20)
INTEGER W_FLAG, BP(20), WP(20), I, J, K, L, M

C ADJUST TWO PLANFORM COORDINATES IF THEY HAVE THE SAME X-COORDINATE
C Find the maximum and minimum points on the planform
C Use the wingbody planform if WFLAG has been set
IF (W_FLAG) THEN
DO I = 1, N_WB
  Initialize XMAX and XMIN to the first points of planform.
  IF (I .EQ. 1) THEN
    XMIN = X_WB(I)
    XMAX = X_WB(I)
  END IF
  Assign wingbody planform coordinate to XMIN if the wingbody planform coordinate is less than XMIN.
  IF (X_WB(I) .LT. XMIN) XMIN = X_WB(I)
  Assign wingbody planform coordinate to XMAX if the wingbody planform coordinate is greater than XMAX.
  IF (X_WB(I) .GT. XMAX) XMAX = X_WB(I)
END DO
ELSE
  Use body planform
  DO I = 1, N_BODY
    Initialize XMAX and XMIN to the first points of planform.
    IF (I .EQ. 1) THEN
      XMIN = X_BODY(I)
      XMAX = X_BODY(I)
    END IF
    Assign body planform coordinate to XMIN if the body planform coordinate is less than XMIN.
    IF (X_BODY(I) .LT. X_MIN) X_MIN = X_BODY(I)
    Assign body planform coordinate to XMAX if the body planform coordinate is greater than XMAX.
    IF (X_BODY(I) .GT. XMAX) XMAX = X_BODY(I)
  END DO
ELSE
  Use body planform
  DO I = 1, N_BODY
    Initialize XMAX and XMIN to the first points of planform.
    IF (I .EQ. 1) THEN
      X_MIN = X_BODY(I)
      X_MAX = X_BODY(I)
    END IF
    Assign body planform coordinate to XMIN if the body planform coordinate is less than X_MIN.
    IF (X_BODY(I) .LT. X_MIN) X_MIN = X_BODY(I)
    Assign body planform coordinate to XMAX if the body planform coordinate is greater than X_MAX.
    IF (X_BODY(I) .GT. XMAX) XMAX = X_BODY(I)
  END DO
END ELSE
END IF
END DO
Calculate a relative delta X to adjust the given points
DELTA X = .001 * (XMAX - XMIN)
Use the wingbody planform if WBFLAG is set

IF (WBFLAG) THEN
  DO 30 I = 2, N WB
    Adjust the current point if the previous point has the
    same X-coordinate.
    IF (X WB(I-1) .EQ. X WB(I))
      X WB(I) = X WB(I) + DELTAX
  CONTINUE
ELSE
  Body adjustment
  DO 40 I = 2, N BODY
    Adjust the current point if the previous point has the
    same X-coordinate.
    IF (X BODY(I-1) .EQ. X BODY(I))
      X BODY(I) = X BODY(I) + DELTAX
  CONTINUE
Wing adjustment
  DO 50 I = 2, N WING
    same X-coordinate.
    IF (X WING(I-1) .EQ. X WING(I))
      X WING(I) = X WING(I) + DELTAX
  CONTINUE
END IF

COMBINE BODY AND WING INTO WINGBODY PLANFORM AND FIND THE
CENTER SECTION PLANFORM.

IF (.NOT. WBFLAG) THEN
  Initialize counters
  Wingbody counter
  K = 2
  Crossing counter
  M = 1
  Determine where the wing planform intersects the body planform
  and store those points in an array
  Step through the body planform
  DO I = 2, N BODY
    Step through the wing planform
    DO J = 2, N WING
      Check to see if any of the points on the body and wing
      coincide.
      IF (X BODY(I) .NE. X WING(J) .AND.
          X BODY(I-1) .NE. X WING(J-1)) THEN
        Find intersection of line defined by wing points and
        line defined by body points.
        CALL CROSSIN(X BODY(I-1),Y BODY(I-1),
                    X BODY(I),Y BODY(I),
                    X WING(J-1),Y WING(J-1),
                    X WING(J),Y WING(J),
                    XCROSS, YCROSS)
      ELSE
Some code continues here...
XCROSS = X_BODY(I)
YCROSS = Y_BODY(I)
END IF

Check to see if lines cross
IF (((XCROSS .GT. X_BODY(I-1)) .AND. XCROSS.LE.X_BODY(I))
  .OR. (XCROSS .LT.X_BODY(I-1)) .AND. XCROSS.GE.X_BODY(I)))
  THEN
    Check to see if one of the crossing points is at the
    same point as a body or wing point.
    IF (XCROSS .EQ. X_BODY(I)) .OR.
    XCROSS .EQ. X_WING(I)) THEN
      CROSS(M) = 1
    ELSE
      CROSS(M) = -1
    END IF

BP(M) = I
WP(M) = J
XCR(M) = XCROSS
YCR(M) = YCROSS
M = M + 1
END IF

END DO

Set end condition for wing and body planform
BP(M) = 0
WP(M) = 0

Initialize Body counter
I = 2

Initialize Wing counter
J = 2

Initialize Wingbody counter
K = 2

Initialize Center section counter
M = 1

Calculate wingbody planform
Determine which planform to start with.
IF (X_BODY(1) .EQ. 0.0) THEN
  The body planform
  X WB(1) = X_BODY(1)
  W FLAG = -1
ELSE
  The wing planform
  X WB(1) = X_WING(1)
  W FLAG = 1
END IF

Termination condition - occurs when the wingbody point equals
the last point of either the body or wing.
IF ((X WB(K-1) .EQ. X_BODY(N_BODY)) .AND.
  Y WB(K-1) .EQ. Y BODY(N BODY)) .OR.
  (X WB(K-1) .EQ. X_WING(N_WING)) .AND.
  Y WB(K-1) .EQ. Y_WING(N_WING))) GO TO 80
IF (W_FLAG .EQ. -1) THEN
    Use the body points for the wingbody planform
    IF (I .LT. BP(M) .OR. BP(M) .EQ. 0) THEN
        X_WB(K) = X_BODY(I)
        Y_WB(K) = Y_BODY(I)
        Increment body counter
        I = I + 1
        Increment wingbody counter
        K = K + 1
    Run through tests again
    GO TO 70

A crossing point has been reached so use it in the wingbody planform.
ELSE IF (I .EQ. BP(M)) THEN
    X_WB(K) = XCR(M)
    Y_WB(K) = YCR(M)
    Increment wingbody counter
    K = K + 1

Determine counter for wing
    IF (CROSS(M) .EQ. 1) THEN
        J = WP(M) + 1
    ELSE IF (CROSS(M) .EQ. -1) THEN
        J = WP(M)
    END IF

Calculate W_FLAG
    W_FLAG = W_FLAG * -1
    Increment crossing counter
    M = M + 1
    Run through tests again
    GO TO 70
END IF

ELSE IF (W_FLAG .EQ. 1) THEN
    Use the wing points for the wingbody planform
    IF (J .LT. WP(M) .OR. WP(M) .EQ. 0) THEN
        X_WB(K) = X_WING(J)
        Y_WB(K) = Y_WING(J)
        Increment wing counter
        J = J + 1
        Increment wingbody counter
        K = K + 1
    Run through tests again
    GO TO 70

A crossing point has been reached so use it in the wingbody planform.
ELSE IF (J .EQ. WP(M)) THEN
    X_WB(K) = XCR(M)
    Y_WB(K) = YCR(M)
    Increment wingbody counter
    K = K + 1

Determine counter for body
    IF (CROSS(M) .EQ. 1) THEN
I = BP(M) + 1 
ELSE IF (CROSS(M) .EQ. -1) THEN
  I = BP(M)
END IF

Calculate W_FLAG
W_FLAG = W_FLAG * -1
Increment crossing counter
M = M + 1
Run through tests again
GO TO 70
END IF

END IF

Finish wingbody calculations
N WB = K - 1
Calculate center section planform
Initialize counters
I = 2
J = 2
L = 2
K = 2
M = 1

Determine which planform to start with.
IF (X_BODY(1) .GT. X_WING(1)) THEN
  Body planform
  X_CS(1) = X_BODY(1)
  W_FLAG = -1
ELSE
  Wing planform
  X_CS(1) = X_WING(1)
  W_FLAG = 1
END IF

Termination condition - occurs when the center section point equals the last point of either the body or wing.
IF ((X_CS(L-1) .EQ. X_BODY(N BODY) .AND.
  Y_CS(L-1) .EQ. Y_BODY(N BODY)) .OR.
  (X_CS(L-1) .EQ. X_WING(N WING) .AND.
  Y_CS(L-1) .EQ. Y_WING(N WING))) GO TO 100

IF (W_FLAG .EQ. -1) THEN
  Use the body points for the center section planform
  IF (I .LT. BP(M) .OR. BP(M) .EQ. 0) THEN
    X_CS(L) = X_BODY(I)
    Y_CS(L) = Y_BODY(I)
  Increment body counter
  I = I + 1
  Increment center section counter
  L = L + 1
  Run through tests again
  GO TO 90
  A crossing point has been reached so use it in the wingbody planform.
ELSE IF (I .EQ. BP(M)) THEN
C
C X_CS(L) = XCR(M)
Y_CS(L) = YCR(M)
Increment center section counter
L = L + 1
Determine counter for wing
C
C IF (CROSS(M) .EQ. 1) THEN
J = WP(M) + 1
ELSE IF (CROSS(M) .EQ. -1) THEN
J = WP(M)
END IF
C
C Calculate W_FLAG
W_FLAG = W_FLAG * -1
Increment crossing counter
M = M + 1
Run through tests again
GO TO 90
END IF
C
ELSE IF (W_FLAG .EQ. 1) THEN
Use the wing points for the center section planform.
IF (J .LT. WP(M) .OR. WP(M) .EQ. 0) THEN
X_CS(L) = X_WING(J)
Y_CS(L) = Y_WING(J)
Increment wing counter
J = J + 1
Increment center section counter
L = L + 1
Run through tests again
GO TO 90
A crossing point has been reached so use it in the wingbody planform.
ELSE IF (J .EQ. WP(M)) THEN
X_CS(L) = XCR(M)
Y_CS(L) = YCR(M)
Increment center section counter
L = L + 1
Determine counter for body
C
IF (CROSS(M) .EQ. 1) THEN
I = BP(M) + 1
ELSE IF (CROSS(M) .EQ. -1) THEN
I = BP(M)
END IF
C
C Calculate W_FLAG
W_FLAG = W_FLAG * -1
Increment crossing counter
M = M + 1
Run through tests again
GO TO 90
END IF
C
END IF
C
C Finish calculations for the center section planform.
ADJUST PLANFORMS AND ANY OTHER VARIABLES THAT ARE BASED ON THIS
COORDINATE SYSTEM SO THAT THE NOSE OF THE AIRCRAFT IS LOCATED AT THE
ORIGIN.

IF (XMIN .NE. 0.0) THEN
  Move body planform
  DO I = 1, N BODY
      X_BODY(I) = X_BODY(I) - XMIN
  END DO
  Move wing planform
  DO I = 1, N_WING
      X_WING(I) = X_WING(I) - XMIN
  END DO
  Move wingbody planform
  DO I = 1, N WB
      X WB(I) = X WB(I) - XMIN
  END DO
  Move center section planform
  DO I = 1, N CS
      X_CS(I) = X_CS(I) - XMIN
  END DO
  Move nozzles
  Front
      XNOZ_F = XNOZ_F - XMIN
  Rear
      IF (YNOZ_R .NE. 9999.) XNOZ_R = XNOZ_R - XMIN
  Move Center of Area and Center of Gravity
      X_CG = X_CG - XMIN
  IF (X_CA .NE. 9999.) X_CA = X_CA - XMIN
  Move Trailing edge placement
      IF (XTE_F .NE. 9999.) XTE_F = XTE_F - XMIN
  IF (XTE_R .NE. 9999.) XTE_R = XTE_R - XMIN
  Move RCS nozzle placement
      IF (X_RCS .NE. 9999.) X_RCS = X_RCS - XMIN
  END IF
  
RETURN
END

SUBROUTINE FINVAR

ACRONYM: FINish VARIABLE calculations

PURPOSE: The purpose of FINVAR is to make all calculations necessary
to variables that the user has not declared through the
PIENAM namelist. An undeclared variable is one that has the
value of 9999.

LOCAL VARIABLES (in addition to the above parameters):

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A_RCS</td>
<td>R</td>
<td>-</td>
<td>sq Ft</td>
<td>Area of roll RCS nozzle</td>
</tr>
<tr>
<td>DAEND</td>
<td>R</td>
<td>-</td>
<td>deg</td>
<td>Default value for ending value of angles of attack</td>
</tr>
</tbody>
</table>
GLOBAL VARIABLES (in addition to the above parameters and local vars):

CONPIE Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEND</td>
<td>R</td>
<td>I</td>
<td>deg</td>
<td>Ending value for angle of attack</td>
</tr>
<tr>
<td>ASTART</td>
<td>R</td>
<td>I</td>
<td>deg</td>
<td>Starting value for angle of attack</td>
</tr>
<tr>
<td>ASTEP</td>
<td>R</td>
<td>I</td>
<td>deg</td>
<td>Step value for angle of attack</td>
</tr>
</tbody>
</table>
### FIGPIE Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_B</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>B_CS</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>E_JF</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>B_w</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>B_WB</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>D_F</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>D_R</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>D_RCS</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>D_B</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>D_CS</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>D_W</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>D_WB</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>D_E</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>D_EFR</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>DE_R</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>E_1</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>E_3</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>E_4</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>KR</td>
<td>R</td>
<td>I</td>
<td>----</td>
</tr>
<tr>
<td>L_B</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>L_CS</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>L_F</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>L_R</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>L_WB</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>MAC</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>MD_RCS</td>
<td>R</td>
<td>I</td>
<td>lbm/s</td>
</tr>
<tr>
<td>N_BODY</td>
<td>I</td>
<td>I</td>
<td>----</td>
</tr>
<tr>
<td>N_CS</td>
<td>I</td>
<td>O</td>
<td>----</td>
</tr>
<tr>
<td>N_WING</td>
<td>I</td>
<td>I</td>
<td>----</td>
</tr>
<tr>
<td>N_WB</td>
<td>I</td>
<td>I</td>
<td>----</td>
</tr>
<tr>
<td>NDIV</td>
<td>I</td>
<td>I</td>
<td>----</td>
</tr>
</tbody>
</table>

**DESCRIPTION**

- **Ending value for jet deflection angle**
- **Starting value for jet deflection angle**
- **Step value for jet deflection angle**
- **Ending value for altitude**
- **Starting value for altitude**
- **Step value for altitude**
- **Ending value for aircraft velocity**
- **Starting value for aircraft velocity**
- **Step value for aircraft velocity**

Width of Body
- Width of Center Section
- Width of Jet Pattern
- Width of Wing (Wing span)
- Width of Wing-Body
- Diameter of each Front jet
- Diameter of each Rear jet
- Diameter of Roll RCS nozzle
- Dbar of Body
- Dbar of Center Section
- Dbar of Wing
- Dbar of Wing-Body
- Effective Diameter of Front jets
- Effective Diameter of Front & Rear jets combined
- Effective Diameter of Rear jets
- Half distance between adjacent jets (1)
- Half distance between adjacent jets (3)
- Half distance between adjacent jets (4)
- Body contour factor
- Length of Body
- Length of Center Section
- Length of Front jet
- Length of Rear jet
- Length of Wing-Body
- Mean Aerodynamic Chord of wing
- Mass flow rate for one RCS nozzle
- Number of data points for Body planform
- Number of data points for Center Section planform
- Number of data points for Wing planform
- Number of data points for Wing-Body planform
- Number of divisions used when calculating suckdown areas (SDAREA) and Dbars (DIABAR)
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Type</th>
<th>I</th>
<th>Ft</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUM</td>
<td>I</td>
<td></td>
<td></td>
<td>Total number of jets</td>
</tr>
<tr>
<td>NUM_F</td>
<td>I</td>
<td></td>
<td></td>
<td>Number of front jets</td>
</tr>
<tr>
<td>NUM_R</td>
<td>I</td>
<td></td>
<td></td>
<td>Number of rear jets</td>
</tr>
<tr>
<td>PER_FR</td>
<td>R</td>
<td></td>
<td></td>
<td>Total perimeter of jets</td>
</tr>
<tr>
<td>PP_LID</td>
<td>R</td>
<td></td>
<td></td>
<td>Ratio of perimeter enclosed by lids to total perimeter</td>
</tr>
<tr>
<td>PR_F</td>
<td>R</td>
<td></td>
<td></td>
<td>Jet pressure ratio for front jets</td>
</tr>
<tr>
<td>PR_FR</td>
<td>R</td>
<td></td>
<td></td>
<td>Jet pressure ratio for all jets</td>
</tr>
<tr>
<td>PR_R</td>
<td>R</td>
<td></td>
<td></td>
<td>Jet pressure ratio for rear jets</td>
</tr>
<tr>
<td>PR_RCS</td>
<td>R</td>
<td>1</td>
<td>lb/sq ft</td>
<td>Roll RCS pressure ratio</td>
</tr>
<tr>
<td>PT_RCS</td>
<td>R</td>
<td>1</td>
<td>lb/sq ft</td>
<td>Total pressure for one roll RCS jet</td>
</tr>
<tr>
<td>Q_F</td>
<td>R</td>
<td>1</td>
<td>lb/sq ft</td>
<td>Dynamic pressure for the front jets</td>
</tr>
<tr>
<td>Q_FR</td>
<td>R</td>
<td>1</td>
<td>lb/sq ft</td>
<td>Dynamic pressure for the rear jets</td>
</tr>
<tr>
<td>Q_R</td>
<td>R</td>
<td>1</td>
<td>lb/sq ft</td>
<td>Dynamic pressure for roll RCS jets</td>
</tr>
<tr>
<td>Q_RCCS</td>
<td>R</td>
<td>1</td>
<td>lb/sq ft</td>
<td>Corner radius of body sides</td>
</tr>
<tr>
<td>R_B</td>
<td>R</td>
<td></td>
<td></td>
<td>Area of body</td>
</tr>
<tr>
<td>S_B</td>
<td>R</td>
<td></td>
<td>sq ft</td>
<td>Area of center section</td>
</tr>
<tr>
<td>S_CS</td>
<td>R</td>
<td></td>
<td>sq ft</td>
<td>Area of front jets</td>
</tr>
<tr>
<td>S_F</td>
<td>R</td>
<td></td>
<td>sq ft</td>
<td>Area affected by fountain arm (1)</td>
</tr>
<tr>
<td>S_FA1</td>
<td>R</td>
<td></td>
<td>sq ft</td>
<td>Area affected by fountain arm (2)</td>
</tr>
<tr>
<td>S_FA3</td>
<td>R</td>
<td></td>
<td>sq ft</td>
<td>Area affected by fountain arm (3)</td>
</tr>
<tr>
<td>S_FA4</td>
<td>R</td>
<td></td>
<td>sq ft</td>
<td>Area affected by fountain arm (4)</td>
</tr>
<tr>
<td>S_FR</td>
<td>R</td>
<td></td>
<td>sq ft</td>
<td>Total jet exit area</td>
</tr>
<tr>
<td>S_JP</td>
<td>R</td>
<td></td>
<td>sq ft</td>
<td>Area enclosed by jet pattern</td>
</tr>
<tr>
<td>S_LID</td>
<td>R</td>
<td></td>
<td>sq ft</td>
<td>Area enclosed by lids</td>
</tr>
<tr>
<td>S_R</td>
<td>R</td>
<td></td>
<td>sq ft</td>
<td>Area of rear jets</td>
</tr>
<tr>
<td>S_W</td>
<td>R</td>
<td></td>
<td>sq ft</td>
<td>Area of wing</td>
</tr>
<tr>
<td>S_WB</td>
<td>R</td>
<td></td>
<td>sq ft</td>
<td>Area of wing-body</td>
</tr>
<tr>
<td>SF_FB</td>
<td>R</td>
<td></td>
<td>sq ft</td>
<td>Area ahead of front jets using body planform</td>
</tr>
<tr>
<td>SF_FW</td>
<td>R</td>
<td></td>
<td>sq ft</td>
<td>Area ahead of front jets using wing planform</td>
</tr>
<tr>
<td>SF_FWB</td>
<td>R</td>
<td></td>
<td>sq ft</td>
<td>Area ahead of front jets using the wingbody planform</td>
</tr>
<tr>
<td>SF_RB</td>
<td>R</td>
<td></td>
<td>sq ft</td>
<td>Area ahead of rear jets using the body planform</td>
</tr>
<tr>
<td>SF_RW</td>
<td>R</td>
<td></td>
<td>sq ft</td>
<td>Area ahead of rear jets using the wing planform</td>
</tr>
<tr>
<td>SF_RWB</td>
<td>R</td>
<td></td>
<td>sq ft</td>
<td>Area ahead of rear jets using the wingbody planform</td>
</tr>
<tr>
<td>SP_FA1</td>
<td>R</td>
<td></td>
<td>sq ft</td>
<td>Potential area affected by fountain arm (1)</td>
</tr>
<tr>
<td>SP_FA3</td>
<td>R</td>
<td></td>
<td>sq ft</td>
<td>Potential area affected by fountain arm (3)</td>
</tr>
<tr>
<td>SP_FA4</td>
<td>R</td>
<td></td>
<td>sq ft</td>
<td>Potential area affected by fountain arm (4)</td>
</tr>
<tr>
<td>SP_JP</td>
<td>R</td>
<td></td>
<td>sq ft</td>
<td>Actual surface area within area enclosed by nozzles</td>
</tr>
<tr>
<td>T_F</td>
<td>R</td>
<td>1</td>
<td>lb</td>
<td>Thrust of front jets</td>
</tr>
<tr>
<td>T_FR</td>
<td>R</td>
<td>1</td>
<td>lb</td>
<td>Total thrust in all jets</td>
</tr>
<tr>
<td>T_R</td>
<td>R</td>
<td>1</td>
<td>lb</td>
<td>Thrust of rear jets</td>
</tr>
<tr>
<td>T_RCS</td>
<td>R</td>
<td>1</td>
<td>lb</td>
<td>Thrust of roll RCS nozzle</td>
</tr>
<tr>
<td>THA_1</td>
<td>R</td>
<td></td>
<td>deg</td>
<td>Half angle between jets (1)</td>
</tr>
<tr>
<td>THA_3</td>
<td>R</td>
<td></td>
<td>deg</td>
<td>Half angle between jets (3)</td>
</tr>
<tr>
<td>THA_4</td>
<td>R</td>
<td></td>
<td>deg</td>
<td>Half angle between jets (4)</td>
</tr>
<tr>
<td>TP_RCCS</td>
<td>R</td>
<td></td>
<td>Rankine</td>
<td>Temperature of flow in roll RCS nozzle</td>
</tr>
<tr>
<td>Variable</td>
<td>Type</td>
<td>Unit</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>TT_F</td>
<td>R</td>
<td>I</td>
<td>Front Thrust / total Thrust</td>
<td></td>
</tr>
<tr>
<td>TT_R</td>
<td>R</td>
<td>I</td>
<td>Rear Thrust / total Thrust</td>
<td></td>
</tr>
<tr>
<td>W_F</td>
<td>R</td>
<td>Ft</td>
<td>Width of Front jet</td>
<td></td>
</tr>
<tr>
<td>W_R</td>
<td>R</td>
<td>Ft</td>
<td>Width of Rear jets</td>
<td></td>
</tr>
<tr>
<td>WL_B</td>
<td>R</td>
<td>I</td>
<td>Width to Length ratio of Body</td>
<td></td>
</tr>
<tr>
<td>WL_CS</td>
<td>R</td>
<td>I</td>
<td>Width to Length ratio of Center Section</td>
<td></td>
</tr>
<tr>
<td>WL_F</td>
<td>R</td>
<td>I</td>
<td>Width to Length ratio of Front jets</td>
<td></td>
</tr>
<tr>
<td>WL_UP</td>
<td>R</td>
<td>I</td>
<td>Width to Length ratio of Jet Pattern</td>
<td></td>
</tr>
<tr>
<td>WL_R</td>
<td>R</td>
<td>I</td>
<td>Width to Length ratio of Rear jets</td>
<td></td>
</tr>
<tr>
<td>WL_WB</td>
<td>R</td>
<td>I</td>
<td>Width to Length ratio of Wing-Body Planform</td>
<td></td>
</tr>
<tr>
<td>X_BODY(500)</td>
<td>R</td>
<td>Ft</td>
<td>X-coordinates of Body planform</td>
<td></td>
</tr>
<tr>
<td>X_CS(30)</td>
<td>R</td>
<td>Ft</td>
<td>X-coordinates of Center Section planform</td>
<td></td>
</tr>
<tr>
<td>X_FR</td>
<td>R</td>
<td>Ft</td>
<td>Distance between Front &amp; Rear jets</td>
<td></td>
</tr>
<tr>
<td>X_RCS</td>
<td>R</td>
<td>Ft</td>
<td>Distance of roll RCS nozzle ahead of wing trailing edge</td>
<td></td>
</tr>
<tr>
<td>X_WB(500)</td>
<td>R</td>
<td>Ft</td>
<td>X-coordinates of Wing-Body planform</td>
<td></td>
</tr>
<tr>
<td>X_WING(500)</td>
<td>R</td>
<td>Ft</td>
<td>X-coordinates of Wing planform</td>
<td></td>
</tr>
<tr>
<td>X_CA</td>
<td>R</td>
<td>Ft</td>
<td>Distance of center of area ahead of CG</td>
<td></td>
</tr>
<tr>
<td>X_CG</td>
<td>R</td>
<td>Ft</td>
<td>X-coordinate of Center of Area</td>
<td></td>
</tr>
<tr>
<td>XCG_C2</td>
<td>R</td>
<td>Ft</td>
<td>X-coordinate of Center of Gravity</td>
<td></td>
</tr>
<tr>
<td>XCG_C4</td>
<td>R</td>
<td>Ft</td>
<td>Distance from CG to MAC/2</td>
<td></td>
</tr>
<tr>
<td>XCG_F</td>
<td>R</td>
<td>Ft</td>
<td>Distance from CG to MAC/4</td>
<td></td>
</tr>
<tr>
<td>XCG_R</td>
<td>R</td>
<td>Ft</td>
<td>Distance Front jet is ahead of CG</td>
<td></td>
</tr>
<tr>
<td>XNO2_F</td>
<td>R</td>
<td>Ft</td>
<td>Distance Rear jet is ahead of CG</td>
<td></td>
</tr>
<tr>
<td>XNO2_R</td>
<td>R</td>
<td>Ft</td>
<td>X-coordinates of Front NOZZle</td>
<td></td>
</tr>
<tr>
<td>XNOZ</td>
<td>R</td>
<td>Ft</td>
<td>X-coordinates of Rear NOZZle</td>
<td></td>
</tr>
<tr>
<td>XTE</td>
<td>R</td>
<td>Ft</td>
<td>X-coordinate of trailing edge for front nozzle (root TE if Nozzle within body)</td>
<td></td>
</tr>
<tr>
<td>XTE_R</td>
<td>R</td>
<td>Ft</td>
<td>X-coordinate of trailing edge for rear nozzle (root TE if Nozzle within body)</td>
<td></td>
</tr>
<tr>
<td>X_1</td>
<td>R</td>
<td>Ft</td>
<td>Spanwise extent of planform on fountain arm (1) center line.</td>
<td></td>
</tr>
<tr>
<td>X_3</td>
<td>R</td>
<td>Ft</td>
<td>Spanwise extent of planform on fountain arm (3) center line.</td>
<td></td>
</tr>
<tr>
<td>X_4</td>
<td>R</td>
<td>Ft</td>
<td>Spanwise extent of planform on fountain arm (4) center line.</td>
<td></td>
</tr>
<tr>
<td>Y_BODY(500)</td>
<td>R</td>
<td>Ft</td>
<td>Y-coordinates of Body planform</td>
<td></td>
</tr>
<tr>
<td>Y_CS(30)</td>
<td>R</td>
<td>Ft</td>
<td>Y-coordinates of Center Section planform</td>
<td></td>
</tr>
<tr>
<td>Y_F</td>
<td>R</td>
<td>Ft</td>
<td>Distance between Front jets</td>
<td></td>
</tr>
<tr>
<td>YNO2_F</td>
<td>R</td>
<td>Ft</td>
<td>Y-coordinates of Front NOZZle</td>
<td></td>
</tr>
<tr>
<td>YNO2_R</td>
<td>R</td>
<td>Ft</td>
<td>Y-coordinates of R NOZZle</td>
<td></td>
</tr>
<tr>
<td>Y_R</td>
<td>R</td>
<td>Ft</td>
<td>Distance between Rear jets</td>
<td></td>
</tr>
<tr>
<td>Y_RCS</td>
<td>R</td>
<td>Ft</td>
<td>Distance of roll RCS nozzle in from wingtip</td>
<td></td>
</tr>
<tr>
<td>Y_WB(500)</td>
<td>R</td>
<td>Ft</td>
<td>Y-coordinates of Wing-Body planform</td>
<td></td>
</tr>
<tr>
<td>Y_WING(500)</td>
<td>R</td>
<td>Ft</td>
<td>Y-coordinates of Wing planform</td>
<td></td>
</tr>
<tr>
<td>YB_F</td>
<td>R</td>
<td>Ft</td>
<td>Lateral distance from Body to Front jets (for external jets)</td>
<td></td>
</tr>
<tr>
<td>YB_R</td>
<td>R</td>
<td>Ft</td>
<td>Lateral distance from Body to Rear jets (for external jets)</td>
<td></td>
</tr>
<tr>
<td>YP_1</td>
<td>R</td>
<td>Ft</td>
<td>Maximum spanwise extent of</td>
<td></td>
</tr>
</tbody>
</table>
planform (1)
Maximum spanwise extent of planform (3)
Maximum spanwise extent of planform (4)
Lateral distance from Wingbody to Front jets (for external jets)
Lateral distance from wingbody to Rear jets (for external jets)
Height of wing above nozzle

---

### MISC Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>POINTS</td>
<td>I</td>
<td>I</td>
<td>---</td>
</tr>
<tr>
<td>XPTS(500)</td>
<td>R</td>
<td>I</td>
<td>---</td>
</tr>
<tr>
<td>YPTS(500)</td>
<td>R</td>
<td>I</td>
<td>---</td>
</tr>
<tr>
<td>TOTAL</td>
<td>R</td>
<td>I</td>
<td>---</td>
</tr>
</tbody>
</table>

**DESCRIPTION**

Number of points which define polygon (not to exceed 500)
X-coordinates of polygon
Y-coordinates of polygon
Total area enclosed by the polygon.

---

### PIEFLAG Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLGRCS</td>
<td>L</td>
<td>O</td>
<td>---</td>
</tr>
<tr>
<td>TYPE_F</td>
<td>C*4</td>
<td>O</td>
<td>---</td>
</tr>
<tr>
<td>TYPE_R</td>
<td>C*4</td>
<td>O</td>
<td>---</td>
</tr>
<tr>
<td>WBFLAG</td>
<td>L</td>
<td>O</td>
<td>---</td>
</tr>
</tbody>
</table>

**DESCRIPTION**

Flag which signals if there is an RCS on the configuration (TRUE - RCS, FALSE - No RCS)
Discription of type of front nozzle (CIRCular, OVAL, RECTangular)
Same as TYPE_F but for rear nozzles
Identifies when WingBody planform has been enter by the user. Used to determine when to use the wingbody in calculations.

---

### PIEERROR Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBERR</td>
<td>I</td>
<td>O</td>
<td>---</td>
</tr>
</tbody>
</table>

**DESCRIPTION**

Error flag returned by DIABAR. If IFLAG = 1, then an explanation of the problem is given at the time of the error.

- 0 = no errors.
- 1 = values of dbar & area probably bad.
- 2 = moved x-position of dbar point/nozzle point to geometric center.
- 3 = input NANGLE was inappropriate (set to absolute value or 1000).
- 4 = Points entered in wrong direction, routine aborted. Must enter points from left to right.
- 5 = 1st and last points not on x-axis, routine aborted.
- 6 = the sweeping ray intersected more than 4 points or zero
RESPIE Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td>Total number of Angle of Attack values</td>
</tr>
<tr>
<td>LD</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td>Total number of nozzle deflection angles.</td>
</tr>
<tr>
<td>LH</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td>Total number of height values.</td>
</tr>
<tr>
<td>LV</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td>Total number of velocity values.</td>
</tr>
</tbody>
</table>

FILES USED:

LOGICAL UNIT I/O DESCRIPTION

(none)

COMMONS USED:

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONPIE</td>
<td>CONtrol variables for Power Induced Effects - Contains variables which control the execution of the PIE module.</td>
</tr>
<tr>
<td>FIGPIE</td>
<td>conFIGuration for Power Induced Effects - Contains all variables needed to define the configuration and other parameters of the aircraft.</td>
</tr>
<tr>
<td>PIEFLAGS</td>
<td>Power Induced Effects FLAGS - Contains variables which help keep track of the configuration of the aircraft.</td>
</tr>
<tr>
<td>PIERROR</td>
<td>Power Induced Effects for eERRORs - Contains all error flags within PIE.</td>
</tr>
<tr>
<td>POLYGON</td>
<td>Contains the points used in POLYAR when calculating the area of a polygon and CENTAR when calculating the center of area.</td>
</tr>
<tr>
<td>RESPIE</td>
<td>REsults from all POwer Induced Effects subroutines - Contains results from each subroutine along with variables for height, velocity, jet defelction angle, and angle of attack and their counters.</td>
</tr>
</tbody>
</table>

CALLED BY:

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPPIE</td>
<td>Main driver routine for the PIE module.</td>
</tr>
</tbody>
</table>

ROUTINES CALLED:

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINTERP</td>
<td>Linear interpolates between two X and Y values give an intermediate X value</td>
</tr>
<tr>
<td>CENTAR</td>
<td>Calculates the center of area, area and area in front of a point for a given planform</td>
</tr>
<tr>
<td>POLYAR</td>
<td>Calculates the area of any polygon</td>
</tr>
<tr>
<td>SDAREA</td>
<td>Calculates areas necessary to make calculations for the fountain effect</td>
</tr>
<tr>
<td>DIAEABAR</td>
<td>Calculates angular mean diameter of a planform</td>
</tr>
<tr>
<td>SSEN</td>
<td>Calculates start, stop, end, and number variables which have not been defined</td>
</tr>
</tbody>
</table>

NOTES: None.

REFERENCES:

1) None.
C ENVIRONMENT:
C FORTRAN 77, VAX 6800, VAX 11/785, SGI IRIS 4D series.
C NON-STANDARD CODE:
C ?
C AUTHOR(S):
Kipp E. Howard (KEH), Cal Poly San Luis Obispo, NASA Ames Moffett
C
C REVISION HISTORY:
C DATE INITIALS & DESCRIPTION
C 04/16/90 KEH -- Initial coding complete.
C 07/13/91 KEH -- Eliminated extraneous spacing and fixed comments

#include "conpie.inc"
#include "figpie.inc"
#include "misc.inc"
#include "pieflag.inc"
#include "pierror.inc"
#include "respie.inc"

REAL PI, X_DB(500), Y_DB(500)
INTEGER N_DB, DLA, DLD, DLH, DLV

PI = 3.1415926

C Determine the number and placement of the nozzles
C Check front nozzle(s) if necessary
IF (NUM_F .EQ. 9999.) THEN
   IF (YNOZ_F .EQ. 0.0) THEN
      NUM_F = 1
   ELSE
      NUM_F = 2
   END IF
END IF

C Check rear nozzle(s) if necessary
IF (NUM_R .EQ. 9999.) THEN
   IF (YNOZ_R .EQ. 0.0) THEN
      NUM_R = 1
   ELSE
      NUM_R = 2
   END IF
END IF

C Calculate total number of nozzles if necessary
IF (NUM .EQ. 9999.) NUM = NUM_F + NUM_R

If there are no rear nozzles, assign the values for rear nozzles
equal to the value of the front nozzles (this is because SDAREA
needs to have at two nozzles input and for the side-by-side
configuration only one nozzle is shown on the planform. SDAREA
knows this configuration if both nozzles have the same
coordinates.
IF (NUM R .EQ. 0 .AND. NUM_F .GT. 1) THEN
  XNOZ_R = XNOZ_F
  YNOZ_R = YNOZ_F
END IF

C Find the extremes (maximums and minimums) for each planform
C Initialize XMAX and XMIN to the first points of the wingbody planform.
  XMINWB = X WB(1)
  XMAXWB = X WB(1)
  YMAXWB = Y WB(1)
  DO 10 I = 2, N WB
    Assign wingbody planform coordinate to XMINWB if the wingbody planform coordinate is less than XMINWB.
    IF (X WB(I) .LT. XMINWB) XMINWB = X WB(I)
    Assign wingbody planform coordinate to XMAXWB if the wingbody planform coordinate is greater than XMAXWB.
    IF (X WB(I) .GT. XMAXWB) XMAXWB = X WB(I)
    Assign wingbody planform coordinate to YMAXWB if the wingbody planform coordinate is greater than YMAXWB.
    IF (Y WB(1) .GT. YMAXWB) YMAXWB = Y WB(1)
    $ DO 10 I = 2, N WB

C If (X WB(I) .GT. XNOZ_F .AND. X WB(I-1) .LE. XNOZ_F)
    THEN
      Determine the Y-coordinate at the nozzle placement using a linear interpolation method.
      CALL LINTERP(XNOZ_F, X WB(I), X WB(I-1), Y WB(I), Y WB(I-1), YB)
      Calculate distance from body side to jet center.
      IF (YWB_F .EQ. 9999.) YWB_F = YNOZ_F - YB
END IF

C IF (X WB(I) .GT. XNOZ_R .AND. X WB(I-1) .LE. XNOZ_R .AND. NUM R .GE. 1) THEN
    Determine the Y-coordinate at the nozzle placement using a linear interpolation method.
    CALL LINTERP(XNOZ_R, X WB(I), X WB(I-1), Y WB(I), Y WB(I-1), YB)
    Calculate distance from body side to jet center.
    IF (YWB_R .EQ. 9999.) YWB_R = YNOZ_R - YB
END IF

CONTINUE 10

IF (.NOT. WBFLAG) THEN
C Find the extremes (maximums and minimums) for each planform
C Initialize XMAX and XMIN to the first points of the body planform.
  YMAX_B = Y BODY(1)
  XMAX_B = X BODY(1)
  DO 20 I = 2, N BODY
    Assign body planform coordinate to YMAX_B if the body planform coordinate is greater than YMAX_B.
    IF (Y BODY(I) .GT. YMAX_B) YMAX_B = Y BODY(I)
    Assign wingbody planform coordinate to XMAXWB if wingbody planform coordinate is greater than XMAXWB.
    IF (X BODY(I) .GT. XMAX_B) XMAX_B = X BODY(I)
    Find distance from body to jets if jets are outside body.
END IF

CONTINUE 20
IF (X_BODY(I) .GT. XNOZ_F .AND. X_BODY(I-1) .LE. XNOZ_F)
THEN
  Determine the Y-coordinate at the nozzle placement using a linear interpolation method.
  CALL LINTERP(XNOZ_F, X_BODY(I), X_BODY(I-1), Y_BODY(I), Y_BODY(I-1), YB)
END IF

Calculate distance from body side to jet center.
IF (YB .EQ. 9999.) YB = YNOZ_F - YB
END IF

IF (X_BODY(I) .GT. XNOZ_R .AND. X_BODY(I-1) .LE. XNOZ_R .AND. NUM_R .GE. 1) THEN
  Determine the Y-coordinate at the nozzle placement using a linear interpolation method.
  CALL LINTERP(XNOZ_R, X_BODY(I), X_BODY(I-1), Y_BODY(I), Y_BODY(I-1), YB)
END IF

Calculate distance from body side to jet center.
IF (YB .EQ. 9999.) YB = YNOZ_R - YB
END IF

CONTINUE
END IF

Initialize XMAX and XMIN to the first points of the center section planform.
YMAXCS = Y_CS(I)
XMAXCS = X_CS(I)
XMINCS = X_CS(I)
DO 25 I = 2, N_CS
  IF (X_CS(I) .LT. XMINCS) XMINCS = X_CS(I)
  IF (X_CS(I) .GT. XMAXCS) XMAXCS = X_CS(I)
  IF (Y_CS(I) .GT. YMAXCS) YMAXCS = Y_CS(I)
25 CONTINUE

----------------------------------------
Find dimensions of all pieces (body, wing, wingbody, Jet Pattern, center section, front and rear nozzles)
IF (WBFLAG) THEN
  Wingbody dimensions
  Width
    IF (B_WB .EQ. 9999.) B_WB = 2.0 * YMAXWB
  Length
    IF (L_WB .EQ. 9999.) L_WB = XMAXWB
  Width to length ratio
    IF (WL_WB .EQ. 9999.) WL_WB = B_WB / L_WB
B_B = 0.0
L_B = 0.0
WL_B = 0.0
B_CS = 0.0
L_CS = 0.0
WL_CS = 0.0
ELSE
  Body dimensions
  Width
    IF (B_B .EQ. 9999.) B_B = 2.0 * YMAX_B
  Length
    IF (L_B .EQ. 9999.) L_B = XMAX_B
  Width to length ratio
IF (WL_B .EQ. 9999.) WL_B = B_B / L_B

Wing dimensions

Width
IF (B_W .EQ. 9999.) B_W = 2.0 * YMAXWB

Wingbody dimensions

Width
IF (B_WB .EQ. 9999.) B_WB = 2.0 * YMAXWB

Length
IF (L_WB .EQ. 9999.) L_WB = XMAXWB

Width to length ratio
IF (WL_WB .EQ. 9999.) WL_WB = B_WB / L_WB

Center Section

Width
IF (B_CS .EQ. 9999.) B_CS = 2.0 * YMAXCS

Length
IF (L_CS .EQ. 9999.) L_CS = XMAXCS - XMINCS

Width to length ratio
IF (WL_CS .EQ. 9999.) WL_CS = B_CS / L_CS

END IF

Jet Pattern dimensions

Width
IF (B_JP .EQ. 9999 .AND. NUM .GE. 3) THEN
  Front nozzle is wider than the rear nozzle
  IF (YNOZ_F .GE. YNOZ_R) B_JP = 2 * YNOZ_F
  Rear nozzle is wider than the front nozzle
  IF (YNOZ_R .GT. YNOZ_F) B_JP = 2 * YNOZ_R
ELSE IF (B_JP .EQ. 9999) THEN
  B_JP = 0
END IF

Width to length ratio
IF (WL_JP .EQ. 9999 .AND. NUM .GE. 3) THEN
  WL_JP = B_JP / (XNOZ_R - XNOZ_F)
ELSE IF (WL_JP .EQ. 9999) THEN
  WL_JP = 0.0
END IF

Front nozzles

Determine what type of nozzles
IF (D_F .NE. 9999 .AND. ((W_F .EQ. 9999 .AND. L_F .EQ. 9999.) $ .OR. (W_F .EQ. L_F))) THEN
  TYPE_F = 'CIRC'
ELSE IF (D_F .EQ. 9999 .AND. W_F .NE. 9999 .AND. L_F .NE. 9999.) $ THEN
  TYPE_F = 'RECT'
ELSE IF (D_F .NE. 9999 .AND. W_F .NE. 9999 .AND. L_F .NE. 9999.) $ THEN
  TYPE_F = 'OVAL'
END IF

Width to length ratio
IF (TYPE_F .EQ. 'CIRC') THEN
  W_F = D_F
  L_F = D_F
  WL_F = 1.0
ELSE
  WL_F = W_F / L_F
Distance between front jets
IF (NUM_F .GT. 1 .AND. Y_F .EQ. 9999.) THEN
    Y_F = 2.0 * YNOZ_F
ELSE IF (NUM_F .EQ. 1 .AND. Y_F .EQ. 9999.) THEN
    Y_F = 0.0
END IF

Equivalent diameter
IF (DE_F .EQ. 9999.) THEN
    For circular nozzles
    IF (TYPE_F .EQ. 'CIRC') THEN
        DE_F = SQRT(NUM_F * D_F**2.0)
    END IF
    For oval nozzles
    ELSE IF (TYPE_F .EQ. 'OVAL') THEN
        DE_F = SQRT(NUM_F * W_F * L_F)
    END IF
    For rectangular nozzles
    ELSE IF (TYPE_F .EQ. 'RECT') THEN
        DE_F = SQRT(4.0/PI*NUM_F*W_F*L_F)
    END IF
END IF

Diameter of individual jets if jets are not circular
IF (D_F .EQ. 9999. ) D_F = DE_F/SQRT(REAL(NUM_F))

Calculate distance from CG to front nozzle
IF (XCG_F .EQ. 9999.) XCG_F = X_CG - XNOZ_F

Rear nozzles
Rear nozzles present?
IF (NUM_R .NE. 0) THEN
    Determine what type of nozzles
    IF (D_R .NE. 9999. .AND. (W_R .EQ. 9999. .AND. L_R .EQ. 9999.)
        .OR. (W_R .EQ. L_R)) THEN
        TYPE_R = 'CIRC'
    ELSE IF (D_R .EQ. 9999. .AND. W_R .NE. 9999. .AND.
        L_R .NE. 9999.) THEN
        TYPE_R = 'RECT'
    ELSE IF (D_R .NE. 9999. .AND. W_R .NE. 9999. .AND.
        L_R .NE. 9999.) THEN
        TYPE_R = 'OVAL'
    END IF

Width to length ratio
IF (TYPE_R .EQ. 'CIRC') THEN
    W_R = D_R
    L_R = D_R
    WL_R = 1.0
ELSE
    WL_R = W_R / L_R
END IF

Distance between rear jets
IF (NUM_R .GT. 0 .AND. YNOZ_R .NE. 0.0 .AND. Y_R .EQ. 9999.)
$ \text{THEN} \\
Y_R = 2.0 \times YNOZ_R \\
\text{ELSE} \\
Y_R = 0.0 \\
\text{END IF}$

**Equivalent diameter**

IF (DE_R .EQ. 9999.0 .AND. NUM_R .GT. 0) THEN
  FOR CIRCULAR NOZZLES
    IF (TYPE_R .EQ. 'CIRC') DE_R = SQRT(NUM_R \times D_R**2.0)
  END IF
  FOR OVAL NOZZLES
    IF (TYPE_R .EQ. 'OVAL') DE_R = SQRT(NUM_R \times W_R \times L_R)
  END IF
  FOR RECTANGULAR NOZZLES
    IF (TYPE_R .EQ. 'RECT') DE_R = SQRT(4.0/PI*NUM_R*W_R*L_R)
  END IF
END IF

**Equivalent diameter**

IF (DE_R .EQ. 9999.0 .AND. NUM_R .GT. 0) THEN
  IF (TYPE_R .EQ. 'CIRC') DE_R = SQRT(NUM_R \times D_R**2.0)
END IF

**Diameter of individual jets if jets are not circular**

IF (D_R .EQ. 9999.0) D_R = DE_R/SQRT(REAL(NI/M_R))

**Calculate distance from CG to rear nozzle**

IF (XCG_R .EQ. 9999.0 .AND. NUM_R .GE. 1) XCG_R = X.CG - XNOZ_R

ELSE
  SET ALL PERTINENT VALUES TO 0.0
  D_R = 0.0
  W_R = 0.0
  L_R = 0.0
  W_L = 0.0
  DE_R = 0.0
  XCG_R = 0.0
  S_R = 0.0
  Y_R = 0.0
  Y_R = 0.0
  SF_RB = 0.0
  SF_RW = 0.0
  SF_RW = 0.0
  W_R = 0.0
  TYPE_R = 'NONE'
END IF

**Both nozzles**

DISTANCE BETWEEN FRONT AND REAR NOZZLES

IF (NUM_R .GT. 0 .AND. X_FR .EQ. 9999.) THEN
  X_FR = XNOZ_R - XNOZ_F
END IF

IF (NUM_R .GT. 0) THEN
  DE_FR = SQRT(DE_F**2.0 + DE_R**2.0)
ELSE
  DE_FR = DE_F
END IF
Total perimeter of jets
IF (PER_FR .EQ. 9999.) THEN

Front jets
IF (TYPE_F .EQ. 'CIRC') THEN
PER = PI * D_F
ELSE IF (TYPE_F .EQ. 'OVAL') THEN
PER = 2.0 * PI * SQRT((W_F**2.0 + L_F**2.0) / 2.0)
ELSE IF (TYPE_F .EQ. 'RECT') THEN
PER = 2.0 * W_F + NUM_F * L_F
END IF

PER_FR = NUM_F * PER

Rear jets
IF (NUM_R .GT. 0) THEN

IF (TYPE_R .EQ. 'CIRC') THEN
PER = PI * D_R
ELSE IF (TYPE_R .EQ. 'OVAL') THEN
PER = 2.0 * PI * SQRT((W_R**2.0 + L_R**2.0) / 2.0)
ELSE IF (TYPE_R .EQ. 'RECT') THEN
PER = 2.0 * W_R + 2.0 * L_R
END IF

PER_FR = PER_FR + NUM_R * PER
END IF

END IF

--------------------
Find values dealing with areas for planforms (DBAR, CENTER OF AREA, AREA AHEAD AND BEHIND JETS, ETC.)

Find the center of area for the wingbody configuration
IF (X_CA .EQ. 9999.) CALL CENTAR(I, 9999., X_WB, Y_WB, N_WB, X_CA)

Calculate XCA CG
IF (XCA_CG .EQ. 9999.) XCA_CG = X_CG - X_CA

Find total area of body
IF (S_B .EQ. 9999.) THEN

IF (.NOT. WBFLAG) THEN
CALL CENTAR(2, 9999., X_BODY, Y_BODY, N_BODY, S_B)
The area has to be multiplied by 2 because only half the
the planform is defined.
S_B = 2.0 * S_B
ELSE
S_B = 0.0
END IF

END IF

Find total area of wing
IF (S_W .EQ. 9999.) THEN

IF (.NOT. WBFLAG) THEN
CALL CENTAR(2, 9999., X_WING, Y_WING, N_WING, S_W)
The area has to be multiplied by 2 because only half the
the planform is defined.
S_W = 2.0 * S_W
ELSE
S_W = 0.0
ENDIF

C

ENDIF

C

Find total area of wingbody
IF (S_WB .EQ. 9999.) THEN
CALL CENTAR(2, 9999., X WB, Y WB, N WB, S WB)
The area has to be multiplied by 2 because only half the
the planform is defined.
S_WB = 2.0 * S_WB
ENDIF

C

Find total area of center section
IF (S_CS .EQ. 9999.) THEN
  IF (.NOT. WBFLAG) THEN
    CALL CENTAR(2, 9999., X_CS, Y_CS, N_CS, S_CS)
The area has to be multiplied by 2 because only half the
the planform is defined.
    S_CS = 2.0 * S_CS
  ELSE
    S_CS = 0.0
  END IF
ENDIF

C

Find the area ahead of the front nozzle
Using the body planform
IF (SF_FB .EQ. 9999.) THEN
  IF (.NOT. WBFLAG) THEN
    CALL CENTAR(3, X_NOZ_F, X_BODY, Y_BODY, N_BODY, SF_FB)
The area has to be multiplied by 2 because only half the
the planform is defined.
    SF_FB = 2.0 * SF_FB
  ELSE
    SF_FB = 0.0
  END IF
ENDIF

C

Using the wing planform
IF (SF_FW .EQ. 9999.) THEN
  IF (.NOT. WBFLAG) THEN
    CALL CENTAR(3, X_NOZ_F, X_WING, Y_WING, N_WING, SF_FW)
The area has to be multiplied by 2 because only half the
the planform is defined.
    SF_FW = 2.0 * SF_FW
  ELSE
    SF_FW = 0.0
  END IF
ENDIF

C

END IF

C
Using the wingbody planform
IF (SF_FW .EQ. 9999.) THEN
CALL CENTAR(3, XNOZ_F, XWB, YWB, N WB, SF FW)
The area has to be multiplied by 2 because only half the
planform is defined.
SF_FW = 2.0 * SF_FW
END IF

Find the area ahead of the rear nozzles if necessary.
Using the body planform
IF (NUM_R .GT. 0) THEN
IF (SF_RB .EQ. 9999.) THEN
IF (.NOT. WBFLAG) THEN
CALL CENTAR(3, XNOZ_R, XBODY, YBODY, N BODY, SF_RB)
The area has to be multiplied by 2 because only half the
planform is defined.
SF_RB = 2.0 * SF_RB
ELSE
SF_RB = 0.0
END IF
END IF

Using the wing planform
IF (SF_RW .EQ. 9999.) THEN
IF (.NOT. WBFLAG) THEN
CALL CENTAR(3, XNOZ_R, X_WING, Y_WING, N_WING, SF_RW)
The area has to be multiplied by 2 because only half the
planform is defined.
SF_RW = 2.0 * SF_RW
ELSE
SF_RW = 0.0
END IF
END IF

Using the wingbody planform
IF (SF_RWB .EQ. 9999.) THEN
CALL CENTAR(3, XNOZ_R, X_WB, Y WB, N WB, SF RWB)
The area has to be multiplied by 2 because only half the
planform is defined.
SF_RWB = 2.0 * SF_RWBC
END IF

Find the area enclosed by the Jet Pattern
IF (S JP .EQ. 9999. .AND. NUM .GE. 3) THEN
XPTS(1) = XNOZ_F
YPTS(1) = 0.0
XPTS(2) = XNOZ_F
YPTS(2) = YNOZ_F
XPTS(3) = XNOZ_R
YPTS(3) = YNOZ_R
XPTS(4) = XNOZ_R
YPTS(4) = 0.0
POINTS = 4
CALL POLYAR

The area has to be multiplied by 2 because only half the
the planform is defined.
S_JP = 2.0 * TOTAL
ELSE IF (S_JP .EQ. 9999.) THEN
S_JP = 0.0
END IF

Set default value for actual surface area enclosed by jet pattern
IF (SP_JP .EQ. 9999. .OR. SP_JP .GT. S_JP) THEN

IF (NUM .GE. 3) THEN
SP_JP = S_JP
ELSE
SP_JP = 0.0
END IF

END IF

Determine the area of each nozzle
Front nozzles (using DE because the shape of nozzle has
been accounted for when DE was calculated)
IF (S_F .EQ. 9999.) S_F = PI / 4.0 * DE_F**2.0

IF (NUM_R .GT. 0) THEN
Rear nozzles (using DE because the shape of nozzle has
been accounted for when DE was calculated)
IF (S_R .EQ. 9999.) S_R = PI / 4.0 * DE_R**2.0
Front and rear nozzles (using DE because the shape of nozzles
have been accounted for when DE was calculated)
IF (S_FR .EQ. 9999.) S_FR = S_F + S_R
ELSE
S_R = 0.0
S_FR = S_F
END IF

Misc. Calculations
Jet dynamic pressures
IF (PR_FR .HI. 9999.) THEN
Assume choked, isentropic flow at nozzle exit, const. Gamma.
Qjet = 1/2 * RHOe * Vexit^2
RHOe = Pe/(R * Te)
Vexit = SQRT(Gamma * R * Te)
Po = PR * Patm
Using isentropic pressure ratio for choked flow (Mach # = 1),
atmospheric pressure, and Pressure Ratio and after reducing
all numbers the following equation applies.

Constants used:
Pe/Po = .52828
R = 53.34 #f*ft/(#m*R)
Gamma = 1.4

IF (Q_FR .EQ. 9999.) Q_FR = 782.92 * PR_FR
Q_F = Q_FR
PR_F = PR_FR

IF (NUM_R .GE. 1) THEN
QR = Q_FR
PR_R = PR_FR
ELSE
QR = 0.0
PR_R = 0.0
END IF

ELSE IF (PR_F .NE. 9999.) THEN
   Same as above.
   IF (Q_F .EQ. 9999.) Q_F = 782.92 * PR_F

   IF (PR_R .NE. 9999.) THEN
      Q_R = 782.92 * PR_R
      Q_FR = (Q_F + Q_R) / 2.0
      PR_FR = (PR_F + PR_R) / 2.0
   ELSE
      IF (NUM_R .GE. 1) THEN
         QR = Q_F
         PR_R = PR_F
      ELSE
         QR = 0.0
         PR_R = 0.0
      END IF
   END IF

   Q_FR = Q_F
   PR_FR = PR_F
END IF

END IF

Account for different thrust inputs
IF (NUM_R .EQ. 0) THEN
   IF (T_F .NE. 9999.) THEN
      T_F = T_F
   ELSE
      T_F = T_FR
   END IF
   TT_F = 1.0
   TR = 0.0
   TT_R = 0.0
- Input of front and rear thrusts only
ELSE IF (T_F .NE. 9999. .AND. T_R .NE. 9999.) THEN
   T_FR = T_F + T_R
   TT_F = T_F / T_FR
   TT_R = 1.0 - TT_F
- Input of front thrust split and total thrust.
ELSE IF (TT_F .NE. 9999. .AND. T_FR .NE. 9999.) THEN
   T_F = T_FR * TT_F
   T_R = T_FR - T_F
   TT_R = T_R / T_FR
- Input of rear thrust split and total thrust.
ELSE IF (TT_R .NE. 9999. .AND. T_FR .NE. 9999.) THEN
   T_R = T_FR * TT_R
   T_F = T_FR - T_R
   TT_F = T_F / T_FR
- Input of front thrust split and front thrust.
ELSE IF (T\textsubscript{F} .NE. 9999. .AND. T\textsubscript{FR} .NE. 9999.) THEN
\[ T\textsubscript{FR} = \frac{T\textsubscript{F}}{TT\textsubscript{F}} \]
\[ T\textsubscript{R} = T\textsubscript{FR} - T\textsubscript{F} \]
\[ TT\textsubscript{R} = T\textsubscript{R} / T\textsubscript{FR} \]

- Input of rear thrust split and rear thrust.
ELSE IF (T\textsubscript{R} .NE. 9999. .AND. T\textsubscript{R} .NE. 9999.) THEN
\[ T\textsubscript{R} = \frac{T\textsubscript{R}}{TT\textsubscript{F}} \]
\[ T\textsubscript{R} = T\textsubscript{FR} - T\textsubscript{F} \]
\[ TT\textsubscript{R} = T\textsubscript{R} / T\textsubscript{FR} \]

- Input of total thrust and rear thrust.
ELSE IF (T\textsubscript{FR} .NE. 9999. .AND. T\textsubscript{R} .NE. 9999.) THEN
\[ T\textsubscript{F} = T\textsubscript{FR} - T\textsubscript{R} \]
\[ TT\textsubscript{F} = T\textsubscript{F} / T\textsubscript{FR} \]
\[ TT\textsubscript{R} = T\textsubscript{R} / T\textsubscript{FR} \]

- Input of total thrust and front thrust.
ELSE IF (T\textsubscript{FR} .NE. 9999. .AND. T\textsubscript{F} .NE. 9999.) THEN
\[ T\textsubscript{R} = T\textsubscript{FR} - T\textsubscript{F} \]
\[ TT\textsubscript{F} = T\textsubscript{F} / T\textsubscript{FR} \]
\[ TT\textsubscript{R} = T\textsubscript{R} / T\textsubscript{FR} \]

- Error with thrust inputs
ELSE
\[ T\textsubscript{ERR} = 1 \]
END IF

Calculate the distance from the CG to the mid-span point of the MAC.
IF (XCG\textsubscript{C2} .EQ. 9999.) XCG\textsubscript{C2} =XCG\textsubscript{C4} / 4.0

Set default values for PP\textsubscript{LID} based on configuration.
IF (PP\textsubscript{LID} .EQ. 9999. ) THEN
\[ IF (NUM .GE. 3 .AND. S\textsubscript{LID} .NE. 0.0) THEN \]
\[ PP\textsubscript{LID} = 1.0 \]
ELSE
\[ PP\textsubscript{LID} = 0.0 \]
END IF

Calculate placement of trailing edge of wing relative to the rear nozzle if there are rear nozzles and trailing edge placement has not been entered.
IF (NUM\textsubscript{R} .GT. 0 .AND. XTE\textsubscript{R} .EQ. 9999.) XTE\textsubscript{R} = XTE\textsubscript{F}

Calculate body contour factor (KR) (NADC-80246-60 Pg 38 Eq. 26)
IF (KR .EQ. 9999.) THEN
\[ IF (R\textsubscript{B} .EQ. 9999.) THEN \]
Set default value
\[ KR = .666667 \]
ELSE IF (NUM\textsubscript{R} .EQ. 0) THEN
Length wise fountain
\[ KR = .05 * (R\textsubscript{B} / (Y\textsubscript{F}/2.0))**-1.0 \]
ELSE
Core-and-Arm, and cross wise fountains
\[ KR = .54 * (R\textsubscript{B} / (X\textsubscript{FR}/2.0))**-.20 \]
END IF
C  END IF
C--------------------
C Calculate all variables that define fountain arms
C Body planform
IF (Z_W .GT. 0.0 .AND. .NOT. WBFLAG) THEN
  POINTS = N_BODY
  DO 30 I = 1, POINTS
     XPTS(I) = X_BODY(I)
     YPTS(I) = Y_BODY(I)
  CONTINUE
30 C Wingbody planform
ELSE
  POINTS = N_WB
  DO 40 I = 1, POINTS
     XPTS(I) = X_WB(I)
     YPTS(I) = Y_WB(I)
  CONTINUE
40 END IF
C
C IF (NUM .GT. 1) THEN
    CALL SDAREA(0, SDAERR, XPTS, YPTS, POINTS, NDIV)
ELSE
    E_1 = 0.0
    E_3 = 0.0
    E_4 = 0.0
    Y_1 = 0.0
    Y_3 = 0.0
    Y_4 = 0.0
    YP_1 = 0.0
    YP_3 = 0.0
    YP_4 = 0.0
    THA_1 = 0.0
    THA_3 = 0.0
    THA_4 = 0.0
    S_FA1 = 0.0
    S_FA3 = 0.0
    S_FA4 = 0.0
    SF_FA1 = 0.0
    SF_FA3 = 0.0
    SF_FA4 = 0.0
END IF
C
C Calculate DBAR for Wingbody if only wingbody is given
C IF (WBFLAG) THEN
C
C Wingbody planform
IF (DB_WB .EQ. 9999.) THEN
  POINTS = N_WB
  DO I = 1, POINTS
     XPTS(I) = X_WB(I)
     YPTS(I) = Y_WB(I)
  END DO
  CALL DIABAR(0, DBERR, XPTS, YPTS, POINTS, XNOZ_F*X_FR/2.0,
                  NDIV, DB_WB)
END IF
C
DB_B = 0.0
Calculate DBAR for each planform

Body planform
IF (DB_B .EQ. 9999.) THEN
  POINTS = N_BODY
  DO 50 I = 1, POINTS
       XPTS(I) = X_BODY(I)
       YPTS(I) = Y_BODY(I)
  CONTINUE
  CALL DIABAR(0, DBERR, XPTS, YPTS, POINTS, XNOZ_F+X_FR/2.0, NDIV, DB_B)
END IF

Wing planform
IF (DB_W .EQ. 9999.) THEN
  POINTS = N_WING
  DO 60 I = 1, POINTS
       XPTS(I) = X_WING(I)
       YPTS(I) = Y_WING(I)
  CONTINUE
  CALL DIABAR(0, DBERR, XPTS, YPTS, POINTS, XNOZ_F+X_FR/2.0, NDIV, DB_W)
END IF

Wingbody planform
IF (DB WB .EQ. 9999.) THEN
  POINTS = N_WB
  DO 70 I = 1, POINTS
       XPTS(I) = X WB(I)
       YPTS(I) = Y WB(I)
  CONTINUE
  CALL DIABAR(0, DBERR, XPTS, YPTS, POINTS, XNOZ_F+X_FR/2.0, NDIV, DB WB)
END IF

Center section planform
IF (DB CS .EQ. 9999.) THEN
  POINTS = N_CS
  DO 80 I = 1, POINTS
       XPTS(I) = X CS(I)
       YPTS(I) = Y CS(I)
  CONTINUE
  CALL DIABAR(0, DBERR, XPTS, YPTS, POINTS, XNOZ_F+X_FR/2.0, NDIV, DB_CS)
END IF

END IF

Determine roll RCS variables when necessary.
IF (X_RCS .EQ. 9999. .OR. Y_RCS .EQ. 9999.) THEN
  FLGRCS = .FALSE.
ELSE
  FLGRCS = .TRUE.
END IF
C Determine roll RCS variables when necessary.
   IF (FLGRCS) THEN
   
C Find pressure ratio of nozzle
   IF (PR_RCS .EQ. 9999.) then
     
C if (PT_RCS .eq. 9999.) then
     Assume Q_RCS is known
     PT_RCS = Q_RCS / 144.0
     PR_RCS = PT_RCS / 14.7
   ELSE IF (Q_RCS .EQ. 9999.) THEN
     Assume PT_RCS is known
     PR_RCS = PT_RCS / 14.7
   END IF
   
C Find total pressure of the nozzle
   IF (PT_RCS .EQ. 9999.) PT_RCS = PR_RCS * 14.7
   
C Find roll jet dynamic pressure
   IF (Q_RCS .EQ. 9999.) Q_RCS = PT_RCS * 144.0
   
C Find roll jet diameter
   IF (D_RCS .EQ. 9999.) THEN
     Velocity (assume choked flow at the nozzle)
     V_RCS = SQRT(1.4 * 53.34 * 32.2 * TP__RCS)
     Area of nozzle
     A_RCS = (T_RCS - MD_RCS * V_RCS / 32.2) / (PT_RCS - 14.7) / 144.
     %
     Diameter of Nozzle
     D_RCS = 2.0 * SQRT(A_RCS / PI)
   END IF
   
C ELSE
   PT_RCS = 0.0
   Q_RCS = 0.0
   D_RCS = 0.0
   END IF

C Make looping variable calculations
C Set default values for looping variables
C Angle of Attack
   DAEND = 30.
   DASTAR = 0.
   DLA = 4
C Deflection angle
   DDEND = 30.
   DDSTAR = 90.
   DLD = 7
C Height
   DHEND = 20 * DE_FR
   DHSTAR = 2 * DE_FR
   DLH = 25
C Velocity
   DVEND = 100.
   DVSTAR = 1
   DLV = 20
C Obtain starting, stepping, ending and number (SSEN) values for
C the following looping variables
C Angle of Attack
CALL SSEN(ASTART, ASTEP, AEND, LA, DASTAR, DAEND, DLA, 9999.)
C Deflection Angle
CALL SSEN(DSTART, DSTEP, DEND, LD, DDSTAR, DDEND, DLD, 9999.)
C Height
CALL SSEN(HSTART, HSTEP, HEND, LH, DHSTAR, DHEND, DLH, 9999.)
C Velocity
CALL SSEN(VSTART, VSTEP, VEND, LV, DVSTAR, DVEND, DLV, 9999.)
C RETURN
END

Sderea

SUBROUTINE SDAREA(IFLAG, SDAERR, X, Y, NPTS, NSLICE)
C PURPOSE: THIS ROUTINE WILL CALCULATE THE POSITION OF THE FOUNTAIN
CORE IN RELATION TO THE NOZZLE POSITION, HALF THE INCLUDED
ANGLE BETWEEN THE FOUNTAIN ARMS, HALF THE DISTANCE BETWEEN
ADJACENT FOUNTAIN ARMS, SPANWISE EXTENT OF PLANFORM ON
FOUNTAIN ARM CENTERLINE, MAXIMUM SPANWISE EXTENT OF
PLANFORM BETWEEN JETS, ACTUAL SURFACE AREA AND POTENTIAL
SURFACE AREA BETWEEN FOUNTAIN ARMS. AN INPUT FILE
CONTAINING PLANFORM COORDINATES (IN A CLOCKWISE DIRECTION
STARTING FROM THE NOSE) IS REQUIRED ALONG WITH THE
COORDINATES OF THE NOZZLE.
C PARAMETERS:

C NAME TYPE I/O UNITS DESCRIPTION
------- ----- --- ---- ----------------------------------------
IFLAG I I ---- USER INTERACTION FLAG:
SDAERR O I ---- ERROR FLAG RETURNED BY SDAREA. IF
IFLAG = 1, THEN AN EXPLANATION OF THE
PROBLEM IS GIVEN AT TIME OF THE ERROR.
X (500) R FEET X-VALUES OF THE PLANFORM (LIMIT = 500).
VALUES MUST START FROM THE LEFT AND
PROCEED COUNTERCLOCKWISE FOR ONE HALF OF
THE AIRCRAFT (THE AIRCRAFT IS ASSUMED
SYMMETRICAL).

Y-VALUES OF THE PLANFORM (LIMIT = 500).
NUMBER OF POINTS USED TO DEFINE THE
PLANFORM.
NUMBER OF SLICES IN THE PLANFORM USED
IN THE INTEGRATION TECHNIQUE.

TO THE ABOVE PARAMETERS):

<p>| LOCAL VARIABLES (IN ADDITION TO THE ABOVE PARAMETERS): |
|---------------------------------|-----------|</p>
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>R</td>
<td>FEET</td>
<td>USED AS UPPER LIMIT FOR INTEGRATION</td>
</tr>
<tr>
<td>ATYPE</td>
<td>I</td>
<td>----</td>
<td>FLAG WHICH SIGNALS WHICH AREA TO ADD</td>
</tr>
<tr>
<td>B</td>
<td>R</td>
<td>FEET</td>
<td>USED AS LOWER LIMIT FOR INTEGRATION</td>
</tr>
<tr>
<td>BLINE1</td>
<td>R</td>
<td>FEET</td>
<td>Y-INTERCEPT FOR LINE 1</td>
</tr>
<tr>
<td>BLINE2</td>
<td>R</td>
<td>FEET</td>
<td>Y-INTERCEPT FOR LINE 2</td>
</tr>
<tr>
<td>BLINE3</td>
<td>R</td>
<td>FEET</td>
<td>Y-INTERCEPT FOR LINE 3</td>
</tr>
<tr>
<td>BLSPAN</td>
<td>R</td>
<td>FEET</td>
<td>Y-INTERCEPT FOR SPAN LINE (FOUNTAIN LINE)</td>
</tr>
<tr>
<td>BMSPAN</td>
<td>R</td>
<td>FEET</td>
<td>Y-INTERCEPT OF LINE WHICH INTERSECTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>POINT WHERE MSPAN(1) OCCURS.</td>
</tr>
<tr>
<td>BTEST</td>
<td>R</td>
<td>FEET</td>
<td>Y-INTERCEPT USED TO TEST WHERE MSPAN(1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OCCURS.</td>
</tr>
<tr>
<td>C</td>
<td>R</td>
<td>FEET</td>
<td>USED AS UPPER LIMIT FOR INTEGRATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>WHEN A AND B ARE BEING USED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(COMPENSATORY TO 0.0)</td>
</tr>
<tr>
<td>DX</td>
<td>R</td>
<td>FEET</td>
<td>MAXIMUM WIDTH OF EACH SLICE IN THE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PLANFORM.</td>
</tr>
<tr>
<td>DXX</td>
<td>R</td>
<td>FEET</td>
<td>ACTUAL WIDTH OF EACH SLICE USED IN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CALCULATIONS.</td>
</tr>
<tr>
<td>E1</td>
<td>R</td>
<td>FEET</td>
<td>HALF THE DISTANCE BETWEEN EACH FOUNTAIN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ARM AND ITS NEAREST NOZZLE (IN THE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CLOCKWISE DIRECTION, STARTING AT THE SIDE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FOUNTAIN ARM.</td>
</tr>
<tr>
<td>E2</td>
<td>R</td>
<td>FEET</td>
<td>SEE E1</td>
</tr>
<tr>
<td>E3</td>
<td>R</td>
<td>FEET</td>
<td>SEE E1</td>
</tr>
<tr>
<td>E4</td>
<td>R</td>
<td>FEET</td>
<td>SEE E1</td>
</tr>
<tr>
<td>FACT</td>
<td>R</td>
<td>----</td>
<td>FACTOR WHICH ADDS OR SUBTRACTS AREA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ACCORDING TO WHICH DIRECTION THE SLICER</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IS MOVING.</td>
</tr>
<tr>
<td>HOLE(3)</td>
<td>R</td>
<td>----</td>
<td>HOLE DETERMINATION VALUE:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.0 = THERE IS NO HOLE IN FUSELAGE.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.0 = THERE IS A HOLE IN THE FUSELAGE.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DO NOT INCLUDE THE PARTICULAR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FOUNTAIN ARM IN CALCULATIONS.</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>----</td>
<td>GENERAL PURPOSE COUNTER</td>
</tr>
<tr>
<td>II</td>
<td>I</td>
<td>----</td>
<td>TEMPORARY VALUE OF I</td>
</tr>
<tr>
<td>IMAX</td>
<td>I</td>
<td>----</td>
<td>THE POINT NUMBER AT WHICH THE MAXIMUM X</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VALUE OCCURS</td>
</tr>
<tr>
<td>IMIN</td>
<td>I</td>
<td>----</td>
<td>THE POINT NUMBER AT WHICH THE MINIMUM X</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VALUE OCCURS</td>
</tr>
<tr>
<td>LINE1</td>
<td>R</td>
<td>FEET</td>
<td>VALUE OF LINES RADIATING FROM NOZZLE 1,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PARALLEL TO THE FOUNTAIN ARM.</td>
</tr>
<tr>
<td>LINE2</td>
<td>R</td>
<td>FEET</td>
<td>VALUE OF LINE BETWEEN BOTH NOZZLE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LINE3</td>
<td>R</td>
<td>FEET</td>
<td>VALUE OF LINES RADIATING FROM NOZZLE 2,</td>
</tr>
</tbody>
</table>
PARALLEL TO THE FOUNTAIN ARM.

VALUE OF LINES RADIATING FROM NOZZLE 2,
PARALLEL TO THE AIRCRAFT CENTER LINE.

VALUE OF LINES RADIATING FROM NOZZLE 1,
PARALLEL TO THE AIRCRAFT CENTER LINE.

VALUE OF LINE THAT REPRESENTS FOUNTAIN
ARM.

PREVIOUS VALUE OF LSPAN.

SLOPE OF LINE FROM NOZZLE 1 TO FOUNTAIN
CORE

SLOPE OF LINE FROM NOZZLE 2 TO FOUNTAIN
CORE

SLOPE OF LINE 1

SLOPE OF LINE 2

SLOPE OF LINE 3

SLOPE OF SPAN LINE (FOUNTAIN ARM).

SLOPE OF LINE WHICH HELPS DEFINE MSPAN
IN REGION 1.

SLOPE OF MIDDLE FOUNTAIN ARM IN THE
FOUR NOZZLE CONFIGURATION.

MAXIMUM SPANWISE EXTENT OF PLANFORM
BETWEEN JETS.

ANOTHER GENERAL PURPOSE COUNTER.

AREA OF NOZZLE WHICH CAN BE SUBTRACTED
FROM THE TOTAL AREA FOUND UNDER EACH
FOUNTAIN ARM.

NUMBER OF NOZZLE (MAXIMUM OF 4)

TEXT DESCRIBING HOW MANY NOZZLE THERE
ARE AT A PARTICULAR PLACE ON PLANFORM.

RADIUS OF EACH NOZZLE

TEMPORARY VALUE OF RADIUS USED IN
SORTING THE NOZZLE POSITIONS.

AREA BETWEEN 'A' AND 'B' THAT THE
INTEGRA SUBROUTINE CALCULATES.

AREA BETWEEN 'C' AND 0.0 THAT THE
INTEGRA SUBROUTINE CALCULATES.

AREA OF PLANFORM THAT IS AFFECTED BY
FOUNTAIN ARM IN REGION 1.

AREA OF PLANFORM THAT IS AFFECTED BY
FOUNTAIN ARM IN REGION 2.

AREA OF PLANFORM THAT IS AFFECTED BY
FOUNTAIN ARM IN REGION 3.

POTENTIAL AREA IN REGION 1.

POTENTIAL AREA IN REGION 2.

POTENTIAL AREA IN REGION 3.

SPANWISE EXTENT OF PLANFORM BETWEEN JETS.

TOTAL AREA FOUND UNDER EACH FOUNTAIN ARM.

HALF THE INCLUDED ANGLE BETWEEN ADJACENT
FOUNTAIN ARMS.

SEE THETA1

SEE THETA1

SEE THETA1

TEXT DESCRIBING WEATHER THE NOZZLE ARE
INTERNAL OR EXTERNAL.

TEXT USED TO ALLOW THE DESCRIPTION
SENTENCE TO BE GRAMATICALLY CORRECT.

1ST X-COORDINATE USED TO DEFINE THE
POTENTIAL AREA OF REGION 1.
2nd x-coordinate used to define the potential area of region 1.
3rd x-coordinate used to define the potential area of region 1.
4th x-coordinate used to define the potential area of region 1.
X-coordinate of the fountain core.
Most aft point on the planform.
Most forward point on the planform.
X-coordinate of each nozzle position indicator across planform.
X-coordinate of spanwise extent of planform between jets.
General purpose temporary value of x used in many different calculations.
The x-coordinate that corresponds to the y-coordinate that is the farthest point from the aircraft centerline (y_max).
1st y-coordinate used to define the potential area of region 1.
2nd y-coordinate used to define the potential area of region 1.
3rd y-coordinate used to define the potential area of region 1.
4th y-coordinate used to define the potential area of region 1.
Y-coordinate of the fountain core.
Farthest point from the aircraft centerline.
Y-coordinate of each nozzle.
Value used to determine M2 for 3 nozzle y-value of planform at the slice point.
Y-coordinate of spanwise extent of planform between jets.
General purpose temporary value of y used in many different calculations.
The y-coordinate that corresponds to the most aft point on the planform (y_max).
The y-coordinate that corresponds to the most forward point on the planform (y_min).

Global variables:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>I/O</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANG_FR</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Front and Rear jet deflection angle</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>I</td>
<td>Ft</td>
<td>Width of Body</td>
</tr>
<tr>
<td>B_JP</td>
<td>B</td>
<td>I</td>
<td>Ft</td>
<td>Width of Jet Pattern</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>I</td>
<td>Ft</td>
<td>Width of Wing (Wing span)</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>I</td>
<td>Ft</td>
<td>Width of Wing-Body</td>
</tr>
<tr>
<td>B_WB</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Diameter of each Front jet</td>
</tr>
<tr>
<td>D_F</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Diameter of each Rear jet</td>
</tr>
<tr>
<td>D_R</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Diameter of Roll RCS nozzle</td>
</tr>
<tr>
<td>D_RCS</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Dbar of Body</td>
</tr>
<tr>
<td>DB_B</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Dbar of Wing</td>
</tr>
<tr>
<td>DB_W</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Dbar of Wing-Body</td>
</tr>
<tr>
<td>DB_WB</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Effective Diameter of Front jets</td>
</tr>
<tr>
<td>DE_F</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Effective Diameter of Front &amp; Rear</td>
</tr>
<tr>
<td>DE_FR</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td></td>
</tr>
</tbody>
</table>
jets combined
Effective Diameter of Rear jets
Density Ratio (Jet / Atm)
Name of file which contains nozzle
placement & size, body & wing
planform coordinates
Height of nozzle exit above ground
Global execution flag for ACSYNT
1) Input data
2) Make calculations
3) Output necessary data
Boundary layer factor
Body contour factor
Length of body
Length of Front jet
Length of Rear jet
Length of Wing-Body
Mean Aerodynamic Chord of wing
Mass flow rate for one RCS nozzle
Number of data points for Body
planform
Number of data points for Wing
planform
Number of data points for Wing-Body
planform
Total NUMBER of jets
NUMBER of Front jets
NUMBER of Rear jets
Total perimeter of jets
Ratio of perimeter enclosed by lids
to total perimeterC
Jet Pressure Ratio for all jets
Roll RCS Pressure Ratio
Total pressure for one roll RCS jet
Dynamic pressure for Front & Rear
jets
Dynamic pressure for roll RCS jets
Area of Body
Area enclosed by Jet Pattern
Area of Front jets
Total jet exit area
Area enclosed by LIDS
Area of Rear jets
Area of Wing
Area of Wing-Body
Area ahead of Front jets using body
planform
Area ahead of Front jets using
wingbody planform
Area ahead of Rear jets
Area ahead of Rear jets using the
wingbody planform
Actual surface area within Jet
Pattern
SPLAY angle of Front jet
SPLAY angle of Rear jet
Thrust of Front jets
Thrust of Rear jets
Thrust of roll RCS nozzle
Temperature of flow in roll RCS nozzle
Front Thrust / total Thrust
Rear Thrust / total Thrust
Forward velocity of aircraft
Width of Front jet
Width of Rear jets
Flow Weight of Inlet / Flow Weight of Exit
Width to Length ratio of Body
Width to Length ratio of Jet
Pattern
Width to Length ratio of Front jets
Width to Length ratio of Rear jets
Width to Length ratio of Wing-Body
X-coordinates of Body planform
Distance between Front & Rear jets
Distance of roll RCS nozzle ahead of wing trailing edge
X-coordinates of Wing-Body planform
X-coordinates of Wing planform
Distance of center of area ahead of CG
X-coordinate of Center of Area
X-coordinate of Center of Gravity
Distance from CG to MAC/2
Distance from CG to MAC/4
Distance Front jet is ahead of CG
Inlet longitudinal distance ahead of CG
Distance Rear jet is ahead of CG
X-coordinates of Front NOZZle
X-coordinates of Rear NOZZle
Y-coordinates of Body planform
Distance between Front jets
Y-coordinates of Front NOZZle
Y-coordinates of R NOZZle
Distance between Rear jets
Distance of roll RCS nozzle in from wingtip
Y-coordinates of Wing-Body planform
Y-coordinates of Wing planform
Lateral distance from Body to Front jets (for external jets)
Lateral distance from Body to Rear jets (for external jets)
Height of body base above nozzle
Inlet vertical distance above CG
Height of wing above nozzle

COMMONS USED: (All common blocks used throughout PIE)

CONTol variables for Power Induced Effects - Contains variables which control the execution of the PIE module.
conFIGuration for Power Induced Effects - Contains all variables needed to define the configuration and other parameters of the aircraft.
C FILES USED:
C LOGICAL UNIT I/O DESCRIPTION
---- ----------------------------
? I Input file, used when IFLAG is set to 1.

C NON-STANDARD CODE:
C
C NOTES: CAN BE RUN AS A STAND ALONE ROUTINE IF CALLED WITH IFLAG
C SET = 1.
C CALLED BY:
NAME DESCRIPTION
--- ---------------------------------------------------------------
COPPIE Controls execution and coordination of Power Induced
Effects Module

C SUBROUTINES CALLED:
NAME DESCRIPTION
---- ---------------------------------------------------------------
CROSSIN Calculates the intersection of two lines, with each
line defined by two points
DIST (FUNC) Calculates the distance between two points
INTEGRA Calculates the areas of thin rectangles
LINECRO Calculates the intersection of two lines, each defined
by a slope and Y-intercept
LINECRO Calculates the intersection of two lines, each defined
by a slope and Y-intercept
LINTERP Linear interpolates between two X and Y values give an
intermediate X value
PERPDIS Calculates the perpendicular distance between a point
and a line

C ENVIRONMENT:
C VAX, VMS, FORTRAN

C AUTHOR(S):
C Kipp E. Howard (KEH), Cal Poly San Luis Obispo, NASA Ames Moffett

C REVISION HISTORY:
C DATE INITIALS & DESCRIPTION
C 02/12/90 KEH -- ORIGINAL CODE COMPLETED
C 07/14/91 KEH -- Eliminated extraneous spacing and fixed comments

#include "figpie.inc"

REAL DX, DXX, E1, E2, E3, E4, LINE1, LINE2, LINE3, LINE4, LINE5, M1, M2,
$ RAD(2), RADIEM, THETA1, THETA2, THETA3, THETA4, X(500), XNOZ(2),
$ XMIN, XMAX, YMAX, XTEMP, Y(500), YTEMP, YNOZ(2), XCORE, YCORE,
$ YPRIME, YSLICE(2000), XSLICE(2000), STOT(3), A, B, C, MMSPAN,
$ SPAN(3), MSPAN(3), XSPAN, LSPAN, LSPANO, MLINE1, MLINE2, HOLE(3),
$ MLINE3, MLSPAN, BLINE1, BLINE2, BLINE3, BLSPAN, NOZARE(2)
INTEGER ATYPE, I, II, IFLAG, SDAERR, NSLICE, NOZZLE, NPTS, FACT
CHARACTER USERFI*50, PLACE(2)*15, TYPE(2)*9, VERB(2)*3

C Initialization of variables
HOLE(1) = 0.0
HOLE(2) = 0.0
HOLE(3) = 0.0

C I/O WITH THE USER (IFLAG = 1).
IF (IFLAG .EQ. 1) THEN
WRITE(6,*)
WRITE(6,*) THIS SUBROUTINE IS DESIGNED TO WORK ONLY FOR
$ SYMMETRICAL.'
WRITE(6,*) 'AIRPLANES. ALSO THE THRUST IN EACH NOZZLE IS'
WRITE(6,*) 'ASSUMED TO BE CONSTANT AND THE NOZZLE ARE ASSUMED'
WRITE(6,*) 'TO BE CIRCULAR.'
WRITE(6,*) ' '
WRITE(6,*) 'USE A TEXT FILE TO LIST THE COORDINATES FOR THE
$ NOZZLE FIRST, THEN LIST THE COORDINATES FOR YOUR AIRCRAFT
$ IN A CLOCKWISE DIRECTION SEQUENTIALLY FROM THE LEFT (THE
$ NOSE OF THE AIRCRAFT).'.
WRITE(6,*) '
WRITE(6,*) '
WRITE(6,*) 'INPUT FILENAME THAT CONTAINS NOZZLE POSITIONS, X
$ & Y COORDINATES:'
READ(*,5) USERFI
FORMAT(A50)
ENDIF
C
6 IF (IFLAG .EQ. 1) THEN
WRITE(6,*) '
WRITE(6,*) 'HOW MANY SLICES IN THE PLANFORM WOULD YOU LIKE TO
$ USE WHEN DETERMINING THE AREAS? (MAXIMUM = 1000)'
READ *, NSLICE
ENDIF
C
SDAERR = 0
IF (NSLICE.LT.1) THEN
IF (IFLAG.EQ.0) THEN
NSLICE = ABS(NSLICE)
SDAERR = 3
ENDIF
IF (NSLICE.LT.1) THEN
NSLICE = 1000
ENDIF
ELSE
WRITE (6,*) '
WRITE (6,*) 'SORRY DAVE, I CAN NOT DO THAT (HAL).'.
GO TO 6
ENDIF
C
ELSE IF (NSLICE.GT.1000) THEN
NSLICE = 1000
SDAERR = 3
ENDIF
C
READ DATA FROM INPUT FILE (NOZZLE PLACEMENT, NOZZLE RADIUS AND
COORDINATES OF PLANFORM. ALSO FIND MOST FORWARD, MOST AFT, AND
WING TIP POINTS.
IF (IFLAG .EQ. 1) THEN
OPEN (UNIT = 7, NAME = USERFI, TYPE = 'OLD')
READ (7,*) XNOZ(1),YNOZ(1),RAD(1),XNOZ(2),YNOZ(2),RAD(2)
I = 1
10 READ (7, *, END = 12) X(I), Y(I)
GO TO 10
12 NPTS = I-1
ELSE
XNOZ(1) = XNOZ_F
YNOZ(1) = YNOZ_F
XNOZ(2) = XNOZ_R
YNOZ(2) = YNOZ_R

END IF

DO 13 I = 1, NPTS
   IF (I .EQ. 1) THEN
      XMIN = X(I)
      YXMIN = Y(I)
      XMAX = X(I)
      YMAX = Y(I)
      IMIN = I
      IMAX = I
   END IF
   IF ((X(I).EQ.X(I-I)).AND.(I.NE.1)) THEN
      SDAERR = 1
      IF (IFLAG.EQ.1) THEN
         WRITE (6,*) 'ONE LINE SEGMENT IN THE PLANFORM DATA WAS VERTICAL.'
         WRITE (6,*) '(THE SECOND VALUE WAS MOVED ALONG THE X-AXIS BY .05.)'
      END IF
      X(I) = X(I) + .0001*X(I)
   END IF
   IF (X(I) .LT. XMIN) THEN
      XMIN = X(I)
      YXMIN = Y(I)
      IMIN = I
   END IF
   IF (X(I) .GT. XMAX) THEN
      XMAX = X(I)
      YXMAX = Y(I)
      IMAX = I
   END IF
   IF (Y(I) .GT. YMAX) THEN
      YMAX = Y(I)
      XYMAX = X(I)
   END IF
CONTINUE

13 CONTINUE

C MOVE ALL POINTS SO THAT THE MOST FORWARD PART OF THE AIRCRAFT IS AT X=0.

C IF (XMIN .NE. 0.0) THEN
   DO 15 I = 1, NPTS
      X(I) = X(I) - XMIN
   CONTINUE
   XNOZ(1) = XNOZ(1) - XMIN
   XNOZ(2) = XNOZ(2) - XMIN
15 CONTINUE

C SORT NOZZLE SO THAT NOZZLE ONE IS CLOSEST TO THE NOSE OF THE AIRCRAFT.

C IF (XNOZ(1) .GT. XNOZ(2)) THEN
   XTEMP = XNOZ(1)
   YTEMP = YNOZ(1)
   RADTEM = RAD(1)
   XNOZ(1) = XNOZ(2)
   YNOZ(1) = YNOZ(2)
C CATCHING FAULTY INPUT SO AS TO AVOID INACCURATE DATA IF POSSIBLE.
C IF (X(1).GT.X(NPTS)) THEN
C IF (IFLAG.EQ.1) THEN
WRITE(6,*) 'ERROR IN DATA. PLEASE ENTER COORDINATES FROM
$ LEFT TO RIGHT.'
SDAERR = 4
GO TO 500
ELSE
SDAERR = 4
GO TO 500
ENDIF
C ENDIF
C IF ((Y(1) .NE. Y(NPTS)) .OR. (Y(1) .NE. 0)) THEN
C IF (IFLAG.EQ.1) THEN
WRITE(6,*) 'PLEASE, TRY TO FOLLOW THE DIRECTIONS. HALF AN
$ AIRPLANE ON THE X-AXIS. I.E. BOTH YOUR FIRST AND LAST Y VALUES
$ SHOULD BE 0. THANK YOU.'
SDAERR = 5
GO TO 500
ELSE
SDAERR = 5
GO TO 500
ENDIF
C ENDIF
C CHECK TO SEE IF NOZZLE ARE IN FRONT OR BEHIND PLANFORM
C IF ((XNOZ(1).LT.XMIN) .OR. (XNOZ(2).GT.XMAX)) THEN
C IF (IFLAG.EQ.1) THEN
WRITE(6,*) 'THE NOZZLES HAVE BEEN LOCATED IN FRONT OR
$ BEHIND THE AIRCRAFT. THE NOZZLE MUST BE PLACED BETWEEN THE
$ NOSE AND THE TAIL OF THE AIRCRAFT.'
SDAERR = 2
GO TO 500
ELSE
SDAERR = 2
GO TO 500
ENDIF
C END IF
C IDENTIFY THE NUMBER AND PLACEMENT OF NOZZLES AS A CHECK FOR THE
C USER.
C NUMBER OF NOZZLES:
C IF (((YNOZ(1).EQ.0.0).AND.(YNOZ(2).EQ.0.0)).OR.
$ (XNOZ(1).EQ.XNOZ(2))) THEN
NOZZLE = 2
ELSE IF ((YNOZ(1).GT.0.0) .AND. (YNOZ(2).GT.0.0)) THEN
NOZZLE = 4
ELSE
NOZZLE = 3
END IF

VERBS FOR THE NUMBER OF NOZZLE
IF (NOZZLE.EQ.3) THEN
IF (YNOZ(i) .GT.0.0) THEN
PLACE(1) = 'ARE TWO NOZZLE'
VERB(1) = 'ARE'
PLACE(2) = 'IS ONE NOZZLE'
VERB(2) = 'IS'
ELSE
PLACE(1) = 'IS ONE NOZZLE'
VERB(1) = 'IS'
PLACE(2) = 'ARE TWO NOZZLE'
VERB(2) = 'ARE'
END IF
ELSE IF (NOZZLE.EQ.2) THEN
IF (XNOZ(1).EQ.XNOZ(2)) THEN
PLACE(1) = 'ARE TWO NOZZLE'
VERB(1) = 'ARE'
ELSE
PLACE(1) = 'IS ONE NOZZLE'
PLACE(2) = PLACE(1)
VERB(1) = 'IS'
VERB(2) = VERB(1)
END IF
ELSE
PLACE(1) = 'ARE TWO NOZZLE'
PLACE(2) = PLACE(1)
VERB(1) = 'ARE'
VERB(2) = VERB(1)
END IF

PLACEMENT OF NOZZLE
DO 20 I=1,NPTS
IF ((X(I).GE.XNOZ(1)) .AND. (X(I-1).LT.XNOZ(1))) THEN
CALL LINTERP(XNOZ(1),X(I),X(I-1),Y(I),Y(I-1),YTEMP)
IF (YNOZ(1).GT.YTEMP) THEN
TYPE(1) = 'EXTERNAL.'
ELSE
TYPE(1) = 'INTERNAL.'
END IF
END IF
END IF
IF ((X(I).GE.XNOZ(2)) .AND. (X(I-1).LT.XNOZ(2))) THEN
CALL LINTERP(XNOZ(2),X(I),X(I-1),Y(I),Y(I-1),YTEMP)
IF (YNOZ(2).GT.YTEMP) THEN
TYPE(2) = 'EXTERNAL.'
ELSE
TYPE(2) = 'INTERNAL.'
END IF
END IF

DETERMINE WHETHER THERE IS A HOLE IN THE FRONT OR REAR OF THE AIRCRAFT.
IF (I.LT.IMIN.AND.Y(I).EQ.0.0.AND.Y(I-1).GT.Y(I))
$ $ HOLE(3) = 1.0
$ IF (I-1.GT.IMAX.AND.Y(I-1).EQ.0.0.AND.Y(I).GT.Y(I-1))
$ $ HOLE(2) = 1.0
20 CONTINUE

C C MORE PLACEMENT OF NOZZLES
IF (IFLAG.EQ.1) THEN
  WRITE (6,*), ' '
  WRITE (6,22) NOZZLE
  IF (XNOZ(1).NE.XNOZ(2)) THEN
    WRITE (6,23) PLACE(1), 'FRONT ', VERB(1), TYPE(1)
    WRITE (6,23) PLACE(2), 'REAR ', VERB(2), TYPE(2)
  ELSE
    WRITE (6,23) PLACE(1), 'CENTER', VERB(1), TYPE(1)
  END IF
END IF

C C CALCULATE PARAMETERS THAT DO NOT CHANGE AS AREAS ARE BEING
C CALCULATED (IE. FOUNTAIN CORE POSITION, SLOPES, ANGLES).
C
C TWO NOZZLES
IF (NOZZLE.EQ.2) THEN
  XCORE = (XNOZ(1) + XNOZ(2))/2.0
  YCORE = 0.0
  IF (XNOZ(1).NE.XNOZ(2)) THEN
    E1 = (XNOZ(2) - XNOZ(1))/2.0
    XSPAN = XCORE
  ELSE
    E1 = YNOZ(1)
    IF (YXMIN.EQ.0.0) THEN
      SPAN(3) = XCORE - XMIN
      MSPAN(3) = SPAN(3)
    END IF
    IF (YXMAX.EQ.0.0) THEN
      SPAN(2) = XMAX - XCORE
      MSPAN(2) = SPAN(2)
    END IF
  END IF
  THETA1 = 3.1415926/2.0
ELSE
  SLOPE OF IMPORTANT LINES
  MLINE2 = (YNOZ(2) - YNOZ(1))/(XNOZ(2) - XNOZ(1))
  IF (MLINE2.NE.0.0) THEN
    FILSPAN = -1.0/MLINE2
    MLINE1 = FILSPAN
    MLINE3 = MLINE1
  ELSE
    Y-INTERCEPT OF IMPORTANT LINES
    BLINE1 = -MLINE1 * XNOZ(1) + YNOZ(1)
    BLINE2 = -MLINE2 * XNOZ(1) + YNOZ(1)
    BLINE3 = -MLINE3 * XNOZ(2) + YNOZ(2)
    BLSPAN = -MSPAN * (XNOZ(1) + XNOZ(2))/2.0 + (YNOZ(1) + YNOZ(2))/2.0
    XCORE = - (BLSPAN / MSPAN)
    YCORE = 0.0
    M1 = (YCORE - YNOZ(1))/(XCORE - XNOZ(1))
    M2 = (YNOZ(2) - YCORE)/(XNOZ(2) - XCORE)
    THETA4 = ABS(ATAN(M1))
  END IF
END IF

C C THREE AND FOUR NOZZLES
ELSE
C SLOPE OF IMPORTANT LINES
  MLINE2 = (YNOZ(2) - YNOZ(1))/(XNOZ(2) - XNOZ(1))
  IF (MLINE2.NE.0.0) THEN
    MSPAN = -1.0/MLINE2
    MLINE1 = MSPAN
    MLINE3 = MLINE1
  END IF
C Y-INTERCEPT OF IMPORTANT LINES
  BLINE1 = -MLINE1 * XNOZ(1) + YNOZ(1)
  BLINE2 = -MLINE2 * XNOZ(1) + YNOZ(1)
  BLINE3 = -MLINE3 * XNOZ(2) + YNOZ(2)
  BLSPAN = -MSPAN * (XNOZ(1) + XNOZ(2))/2.0 + (YNOZ(1) + YNOZ(2))/2.0
  XCORE = - (BLSPAN / MSPAN)
  YCORE = 0.0
  M1 = (YCORE - YNOZ(1))/(XCORE - XNOZ(1))
  M2 = (YNOZ(2) - YCORE)/(XNOZ(2) - XCORE)
  THETA4 = ABS(ATAN(M1))
END IF

C C
THETA2 = ABS(ATAN(M2))
IF (ATAN(MLSPAN).GT.0.0) THEN
    THETA1 = ATAN(MLSPAN) - THETA2
ELSE
    THETA1 = 3.1415926 - THETA2 + ATAN(MLSPAN)
END IF
THETA3 = 3.1415926 - THETA1 - THETA2 - THETA4
E1 = DIST(XNOZ(1),YNOZ(1),XNOZ(2),YNOZ(2))/2.0
E3 = E1
E4 = YNOZ(1)
E2 = YNOZ(2)
IF (NOZZLE.EQ.3.AND.YNOZ(1).GT.YNOZ(2)) THEN
    E2 = YNOZ(1)
    E4 = 0.0
END IF
IF (YXMAX.EQ.0.0) THEN
    SPAN(2) = XMAX - XNOZ(2)
    MSPAN(2) = SPAN(2)
END IF
IF (YXMIN.EQ.0.0) THEN
    SPAN(3) = XNOZ(1) - XMIN
    MSPAN(3) = SPAN(3)
    IF (YNOZ(2).EQ.0.0) THEN
        SPAN(2) = SPAN(3)
        MSPAN(2) = SPAN(2)
        SPAN(3) = 0.0
        MSPAN(3) = 0.0
    ELSE IF (YNOZ(1).EQ.0.0) THEN
        SPAN(3) = 0.0
        MSPAN(3) = 0.0
    END IF
END IF
END IF
C C CALCULATE WIDTH OF EACH SLICE.
DX = XMAX/NSLICE
C SET SLICER AT NOSE AND BEGIN MAKING SLICES ALONG THE THE YSLICE(N) OF THE AIRCRAFT.
C WRITE (6,32)
N = 0
X(NPTS+1) = X(NPTS)
II = 1
30 I = II
N = N + 1
IF (N-I .EQ. 0) THEN
    PSLICE = XSLICE(1)
ELSE
    PSLICE = XSLICE(N-1)
END IF
C C DX IS GOING TO THE RIGHT WITH NO TURNS IN SIGHT
IF ((X(I).LT.X(I+1)).AND.(X(I+1).LE.X(I+2))) THEN
C THE NEXT SLICE COMES BEFORE THE NEXT OUTLINE POINT
IF ((PSLICE+DX).LT.X(I+1)) THEN
    DXX = DX
    II = I
C THE NEXT SLICE OCCURS AFTER THE NEXT OUTLINE POINT AND BEFORE
C THE POINT AFTER THAT.
ELSE IF (((PSLICE+DX).GE.X(I+1)).AND. (PSLICE+DX).LE.X(I+2)) THEN
  DXX = DX
  II = I + 1
END IF

C THE NEXT SLICE OCCURS AFTER THE NEXT TWO OUTLINE POINTS
ELSE IF (((PSLICE+DX).GT.X(I+2)).AND. (X(I+1).LT.X(I+2))) THEN
  II = I + 1
  DXX = ((X(II) + X(II+1))/2.0) - PSLICE
ELSE IF (X(I+1).EQ.X(I+2)) THEN
  GO TO 100
END IF

FACT = 1

C
C DX IS GOING TO THE LEFT WITH NO TURNS IN SIGHT
ELSE IF ((X(I).GT.X(I+1)).AND.((X(I+1).GE.X(I+2)))) THEN
C THE NEXT SLICE OCCURS BEFORE THE NEXT OUTLINE POINT
  IF ((PSLICE-DX).GT.X(I+1)) THEN
    DXX = -DX
    II = I
  ELSE IF (PSLICE.LT.X(I+2)) THEN
    II = I + 1
  ELSE IF (PSLICE.GT.((X(I+1)+X(I+2))/2.0)) THEN
    II = I + 1
    DXX = 0.0
  ELSE
    II = I + 1
    DXX = ((X(II)+X(II+1))/2.0) - PSLICE
  END IF
  FACT = -1
END IF

C
C DX IS GOING TO THE RIGHT AND A TURN IS COMING UP.
ELSE IF ((X(I).LT.X(I+1)).AND.((X(I+1).GT.X(I+2)))) THEN
C THE NEXT SLICE OCCURS BEFORE THE NEXT OUTLINE POINT
  IF ((PSLICE+DX).LE.X(I+1)) THEN
    DXX = DX
    II = I
    FACT = 1
  ELSE
    II = I + 1
    DXX = ((X(II)+X(II+1))/2.0) - PSLICE
    ELSE IF (PSLICE.GT.((X(I+1)+X(I+2))/2.0)) THEN
      II = I + 1
      DXX = 0.0
    ELSE
      II = I + 1
      DXX = ((X(II)+X(II+1))/2.0) - PSLICE
    END IF
    FACT = -1
  END IF
END IF

C
C DX IS GOING TO THE LEFT AND A TURN IS COMING UP.
ELSE IF ((X(I+1).LT.X(I)).AND.(X(I+2).GT.X(I+1))) THEN
  IF ((PSLICE-DX).GE.X(I+1)) THEN
    DXX = -DX
    II = I
    FACT = -1
  ELSE
    IF (PSLICE.GT.X(I+2)) THEN
      II = I + 1
      DXX = ((X(II)+X(II+1))/2.0) - PSLICE
    ELSE IF (PSLICE.LT.((X(I+1)+X(I+2))/2.0)) THEN
      II = I + 1
      DXX = 0.0
    ELSE
      II = I + 1
      DXX = ((X(II)+X(II+1))/2.0) - PSLICE
    END IF
    FACT = 1
  END IF
END IF

C CALCULATE NEW SLICE POSITION GIVEN 'DXX' FROM ABOVE AND DETERMINE THE
C CORRESPONDING YSLICE (N) FOR THAT POSITION.
  IF (N-I .EQ. 0) THEN
    XSLICE(N) = DXX
  ELSE
    XSLICE(N) = XSLICE(N-1) + DXX
  END IF
  CALL LINTERP (XSLICE(N),X(II),X(II+1),Y(II),Y(II+1),yslice(n))
  IF (N.GE.2000) THEN
    SDA ERR = 6
    IF (IFLAG.EQ.1) WRITE (6,*) 'THE NUMBER OF INTERNAL SLICES HAS
    $ NUMBER OF SLICES.'
    GO TO 500
  END IF

C DETERMINE WHERE THE SLICER IS ON THE PLANFORM AND SET THE VALUES OF
C A, B, AND C TO THEIR RESPECTIVE VALUES. ALSO DETERMINE THE POSITIONS
C OF THE SPANWISE AND MAXIMUM EXTENT OF THE PLANFORM ON THE FOUNTAIN
C ARM.
  C TWO NOZZLES
    IF (NOZZLE.EQ.2) THEN
      C ONE NOZZLE IN THE FRONT AND ONE IN THE REAR.
      IF (YNOZ(1).EQ.0.0) THEN
        IF (((XSLICE(N).GT.XNOZ(1)).AND.(XSLICE(N).LT.XNOZ(2)))
        $ THEN
          A = YSLICE(N)
          B = 0.0
        ELSE
          A = 0.0
          B = 0.0
        END IF
        CALL LINTERP (XSPAN,XSLICE(N),XSLICE(N-1),YSLICE(N),
      $ THEN

$\text{YSLICE}(N-1), \text{ytemp})$

\text{IF} (\text{SPAN}(1) .LT. \text{ytemp}) \text{SPAN}(1) = \text{ytemp}$

\text{END IF}

\text{IF} (\text{MSPAN}(1) .LT. A) \text{MSPAN}(1) = A

\text{ATYPE} = 1

\text{TWO NOZZLES SIDE BY SIDE}

\text{ELSE}

\text{IF} (\text{YSLICE}(N) .LT. \text{YNOZ}(1)) \text{THEN}

\text{A} = \text{YSLICE}(N)

\text{B} = 0.0

\text{ELSE}

\text{A} = \text{YNOZ}(1)

\text{B} = 0.0

\text{END IF}

\text{DETERMINE MAXIMUM SPANWISE AND SPANWISE EXTENT OF PLANFORM}

\text{ON FOUNTAIN ARM CENTERLINE.}

\text{IF} (\text{YSLICE}(N) .LT. \text{EI} \text{AND} .YXMIN.NE.0.0) \text{THEN}

\text{IF} (\text{N}.EQ.1) \text{THEN}

\text{SPAN}(1) = \text{XCORE} - \text{X}(1)

\text{SPAN}(2) = \text{X(NPTS)} - \text{XCORE}

\text{ELSE IF} (\text{YSLICE}(N).EQ.0.0) \text{THEN}

\text{IF} (\text{XCORE}-\text{XSLICE}(N)) .GT. 0.0) \text{THEN}

\text{IF} (\text{SPAN}(1) .LT. (\text{XCORE}-\text{XSLICE}(N))) \text{THEN}

\text{SPAN}(1) = \text{XCORE} - \text{XSLICE}(N)

\text{ELSE IF} (\text{XSLICE}(N)-\text{XCORE}) .GT. 0.0) \text{THEN}

\text{IF} (\text{SPAN}(2) .LT. (\text{XSLICE}(N)-\text{XCORE})) \text{THEN}

\text{SPAN}(2) = \text{XSLICE}(N) - \text{XCORE}

\text{END IF}

\text{IF} ((\text{XSLICE}(N)-\text{XCORE}) .LT. (-\text{MSPAN}(1))) \text{THEN}

\text{MSPAN}(1) = \text{XCORE} - \text{XSLICE}(N)

\text{ELSE IF} ((\text{XSLICE}(N)-\text{XCORE}) .GT. (\text{MSPAN}(2))) \text{THEN}

\text{MSPAN}(2) = \text{XSLICE}(N) - \text{XCORE}

\text{END IF}

\text{IF} \text{sllicer is before the nozzles then add to STOT}(3)

\text{else add to STOT}(2)

\text{IF} (\text{XSLICE}(N) .LT. \text{XNOZ}(1)) \text{THEN}

\text{ATYPE} = 3

\text{ELSE}

\text{ATYPE} = 2

\text{END IF}

\text{C} = 0.0

\text{THREE NOZZLE}

\text{ELSE IF} (\text{NOZZLE.EQ.3}) \text{THEN}

\text{LINE2} = \text{MLINE2} \text{XSLICE}(N) + \text{BLINE2}

\text{LINE1} = \text{MLINE1} \text{XSLICE}(N) + \text{BLINE1}

\text{LINE3} = \text{MLINE3} \text{XSLICE}(N) + \text{BLINE3}

\text{LSPAN0} = \text{LSPAN}

\text{LSPAN} = \text{MLSPAN} \text{XSLICE}(N) + \text{BLSPAN}

\text{IF} (\text{YNOZ}(1).GT.0.0) \text{THEN}

\text{LINE4} = \text{YNOZ}(1)

\text{ELSE}

\text{LINE4} = \text{YNOZ}(2)

\text{END IF}
ONE NOZZLE IN THE FRONT AND TWO NOZZLES IN THE REAR

IF (LINE1 .LT. LINE3) THEN
  IF (LINE1 .GE. LINE2) THEN
    IF (LINE1 .LT. YSLICE (N)) THEN
      IF (LINE3 .LT. YSLICE (N)) THEN
        A = LINE3
        B = LINE1
      ELSE
        A = YSLICE (N)
        B = LINE1
      END IF
    ELSE
      A = 0.0
      B = 0.0
    END IF
    C = 0.0
    ATYPE = 1
  ELSE IF (LINE3 .GE. LINE2) THEN
    IF (LINE3 .LT. YSLICE (N)) THEN
      A = LINE3
      B = LINE2
    ELSE IF (LINE2 .LT. YSLICE (N)) THEN
      A = YSLICE (N)
      B = LINE2
    ELSE
      A = 0.0
      B = 0.0
    END IF
    C = 0.0
    ATYPE = 1
  ELSE IF (LINE4 .LT. LINE2) THEN
    IF (LINE4 .GT. YSLICE (N)) THEN
      A = YSLICE (N)
    ELSE
      A = LINE4
    END IF
    B = 0.0
    C = 0.0
    ATYPE = 2
    IF (YSLICE (N) .EQ. 0.0 .AND. SPAN (2) .LT. XSLICE (N) - XNOZ (2))
      SPAN (2) = XSLICE (N) - XNOZ (2)
    ELSE IF ((YSLICE (N) .LT. LINE4) .AND. ((XSLICE (N) - XNOZ (2)) .GT. MSPAN (2)))) MSPAN (2) = XSLICE (N) - XNOZ (2)
  END IF
ENDIF

Skip test for X, YSPAN if N=1, because subscript out of range

IF (N .NE. 1) THEN
  IF ((LSSPAN - YSLICE (N-1)) * (LSSPAN - YSLICE (N)) .LE. 0.0) THEN
    CALL CROSSIN (XSLICE (N-1), YSLICE (N-1), XSLICE (N), YSLICE (N), XSLICE (N-1), LSPAN, LSPANO, XSLICE (N), LSPAN, 
    xtemp, ytemp)
  END IF
ENDIF
TWO NOZZLE IN THE FRONT AND ONE IN THE REAR.

ELSE

   IF (LINE3.GE.LINE2) THEN
      IF (LINE3.LT.YSLICE(N)) THEN
         IF (LINE1.LT.YSLICE(N)) THEN
            A = LINE1
            B = LINE3
         ELSE
            A = YSLICE(N)
            B = LINE3
         END IF
      ELSE
         A = 0.0
         B = 0.0
      END IF

   ELSE
      A = 0.0
      B = 0.0
   END IF

   ATYPE = 1

ELSE IF (LINE1.GE.LINE2) THEN
   IF (LINE1.LT.YSLICE(N)) THEN
      A = LINE1
      B = LINE2
   ELSE IF (LINE2.LT.YSLICE(N)) THEN
      A = YSLICE(N)
      B = LINE2
   ELSE
      A = 0.0
      B = 0.0
   END IF

   C = 0.0

   ATYPE = 1

ELSE IF (LINE4.LT.LINE2) THEN
   IF (LINE4.GT.YSLICE(N)) THEN
      A = YSLICE(N)
   ELSE
      A = LINE4
   END IF

   B = 0.0

   C = 0.0

   ATYPE = 2

   IF (YSLICE(N).EQ.0.0.AND.SPAN(2).LT.XNOZ(1)-XSLICE(N)) THEN
      SPAN(2) = XNOZ(1) - XSLICE(N)
   ELSE
      SPAN(2) = SPAN(1) - SPAN(2)
   END IF

   END IF

END IF

FOUR NOZZLES

ELSE

   LINE2 = MLINE2*XSLICE(N)+BLINE2
   IF (YNOZ(1).NE.YNOZ(2)) THEN
      LINE1 = MLINE1*XSLICE(N)+BLINE1
      LINE3 = MLINE3*XSLICE(N)+BLINE3
      LSPAN0 = LSPAN
      LSPAN = MSPAN*XSLICE(N)+BLSPAN
   END IF

   LINE4 = YNOZ(2)
   LINE5 = YNOZ(1)

   BOTH NOZZLES ARE THE SAME DISTANCE FROM THE CENTER LINE.
IF (LINE5.EQ.LINE4) THEN
IF (XSLICE(N).LT.XNOZ(1)) THEN
  IF (LINE5.LT.YSLICE(N)) THEN
    A = LINE5
  ELSE
    A = YSLICE(N)
  END IF
  B = 0.0
  C = 0.0
  ATYPE = 3
ELSE IF (XSLICE(N).GT.XNOZ(2)) THEN
  IF (LINE4.LT.YSLICE(N)) THEN
    A = LINE4
  ELSE
    A = YSLICE(N)
  END IF
  B = 0.0
  C = 0.0
  ATYPE = 2
ELSE IF (LINE2.LT.YSLICE(N)) THEN
  A = YSLICE(N)
  B = LINE2
  C = 0.0
  ATYPE = 1
END IF
THE REAR NOZZLE ARE CLOSER TOGETHER THAN THE FRONT NOZZLE.
ELSE IF (LINE5.GT.LINE4) THEN
IF (LINE2.GT.LINE5) THEN
  IF (LINE5.LT.YSLICE(N)) THEN
    A = LINE5
  ELSE
    A = YSLICE(N)
  END IF
  B = 0.0
  C = 0.0
  ATYPE = 3
ELSE IF ((LINE2.LE.LINE5).AND.(LINE1.LT.YSLICE(N))) THEN
  A = LINE1
  B = LINE2
  C = 0.0
  ATYPE = 1
ELSE IF (LINE2.LT.LINE4) THEN
  IF (LINE3.LT.YSLICE(N)) THEN
    A = YSLICE(N)
    B = LINE3
    C = LINE4
    ATYPE = 4
  ELSE IF (LINE4.LT.YSLICE(N)) THEN
    A = LINE4
    B = 0.0
    C = 0.0
    ATYPE = 2
  ELSE
    A = YSLICE(N)
    B = 0.0
    C = 0.0
    ATYPE = 2
END IF
ELSE IF (LINE2.LT.YSLICE(N)) THEN
    A = YSLICE(N)
    B = LINE2
    C = 0.0
    ATYPE = 1
ELSE
    A = 0.0
    B = 0.0
    C = 0.0
END IF

THE FRONT NOZZLE ARE CLOSER TOGETHER THAN THE REAR NOZZLE.
ELSE

IF (LINE2.GT.LINE4) THEN
    IF (LINE4.LT.YSLICE(N)) THEN
        A = LINE4
    ELSE
        A = YSLICE(N)
    END IF
    B=0.0
    C=0.0
    ATYPE = 2
ELSE IF ((LINE2.LE.LINE4).AND.(LINE3.LT.YSLICE(N))) THEN
    A = LINE3
    B = LINE2
    C = 0.0
    ATYPE = 1
ELSE IF (LINE2.LT.LINE5) THEN
    IF (LINE1.LT.YSLICE(N)) THEN
        A = YSLICE(N)
        B = LINE1
        C = LINE5
        ATYPE = 5
    ELSE IF (LINE5.LT.YSLICE(N)) THEN
        A = LINE5
        B = 0.0
        C = 0.0
        ATYPE = 3
    ELSE
        A = YSLICE(N)
        B = 0.0
        C = 0.0
        ATYPE = 3
    END IF
ELSE IF (LINE2.LT.YSLICE(N)) THEN
    A = YSLICE(N)
    B = LINE2
    C = 0.0
    ATYPE = 1
ELSE
    A = 0.0
    B = 0.0
    C = 0.0
END IF
END IF

IF (YNOZ(I).EQ.YNOZ(2)) THEN
    XSPAN = XCORE
    IF ((XSLICE(N).GE.XSPAN).AND.(XSLICE(N-1).LT.XSPAN)) THEN

$
CALL LINTERP (XSPAN, XSLICE (N), XSLICE (N-1), YSLICE (N), YSLICE (N-1), ytemp)
ENDIF
IF (SPAN(1).LT.YTEMP) SPAN(1) = YTEMP
ENDIF
IF (MSPAN(1).LT.A) MSPAN(1) = A
ELSE
IF (YSLICE(N).EQ.0.0) THEN
IF (SPAN(2).LT.XSLICE(N) - XNOZ(2)) SPAN(2) = XSLICE(N) - XNOZ(2)
ENDIF
IF (((YSLICE(N).LT.LINE4).AND.((XSLICE(N) - XNOZ(2)).GT.  
MSPAN(2)))) MSPAN(2) = XSLICE(N) - XNOZ(2)
ENDIF
IF (((YSLICE(N).LT.LINE5).AND.((XNOZ(1) - XSLICE(N)).GT.  
MSPAN(3)))) MSPAN(3) = XNOZ(1) - XSLICE(N)
ENDIF
C
C CALCULATE THE X & Y COORDINATES THAT CORRESPOND TO THE MAXIMUM
C SPANWISE EXTENT OF PLANFORM ON FOUNTAIN ARM CENTERLINE IN REGION 1
C (THE SIDE OF THE AIRCRAFT.)
IF (NOZZLE.GT.2.AND.A.EQ.YSLICE(N).AND.B.NE.0.0) THEN
IF ((LSPAN-YSLICE(N-1))*(LSPAN-YSLICE(N)).LE.0.0) THEN
CALL CROSSIN(XSLICE(N-1), YSLICE(N-1), XSLICE(N), YSLICE(N),  
XSLICE(N-1), LSPAN, XSLICE(N), LSPAN, xtemp, ytemp)
IF (YSPAN.LT.YTEMP) THEN
YSPAN = YTEMP
XSPAN = XTEMP
ENDIF
BTEST = YSLICE(N) - MLINE2 * XSLICE(N)
CALL LINECRO(MLINE1, BLINE1, MLINE2, BTEST, xtemp, ytemp)
IF (YTEMP.GT.YTEMPOLD) THEN
YTEMPOLD = YTEMP
XTEMPOLD = YTEMP
XMSSPAN = XSLICE(N)
YMSSPAN = YSLICE(N)
ENDIF
ENDIF
C
C CALCULATE THE AREA FOR EACH DXX AND ADD OR SUBTRACT THAT FROM THE
C TOTAL AREA IN EACH REGION.
IF (DXX.EQ.0.0) DXX = DX
DXX = ABS(DXX)
CALL INTEGRA(A, B, C, DXX, SAB, SC)
IF (ATYPE.EQ.4) THEN
STOT(1) = STOT(1) + FACT * SAB
STOT(2) = STOT(2) + FACT * SC
ELSE IF (ATYPE.EQ.5) THEN
STOT(1) = STOT(1) + FACT * SAB
STOT(3) = STOT(3) + FACT * SC
ELSE
STOT(ATYPE) = STOT(ATYPE) + FACT * SAB
ENDIF
C
GO TO 30
C
C CALCULATE MAXIMUM SPANWISE AND SPANWISE EXTENT OF PLANFORM ON FOUNTAIN
C AREA CENTERLINE FOR REGION 1 (THE SIDE OF THE AIRCRAFT.)

100 IF (NOZZLE.GT.2 .AND. YMSPAN.NE.0.0) THEN
   CALL PERPDIS(XSPAN, YSPAN, MLINE2, BLINE2, span(I))
   CALL PERPDIS(XMSPAN, YMSPAN, MLINE2, BLINE2, mspan(I))
END IF

C CALCULATE AND SUBTRACT THE AREAS OF THE NOZZLE FROM THE TOTAL AREAS

IF (IFLAG .EQ. i) THEN
   NOZARE(1) = RAD(1)**2.0 * 3.1415926
   NOZARE(2) = RAD(2)**2.0 * 3.1415926
ELSE
   NOZARE(1) = S_F/NUM_F
   NOZARE(2) = S_R/NUM_R
ENDIF

IF (TYPE(1).EQ. 'INTERNAL. ') THEN
   IF (XNOZ(1).EQ.XNOZ(2)) THEN
      STOT(2) = STOT(2) - NOZARE(1)/4.0
      STOT(3) = STOT(3) - NOZARE(1)/4.0
   ELSE
      STOT(1) = STOT(1) - NOZARE(1)/4.0
   END IF
   IF (NOZZLE.EQ.3) THEN
      IF (YNOZ(2).EQ.0.0) STOT(2) = STOT(2) - NOZARE(1)/4.0
   ELSE IF (NOZZLE.EQ.4) THEN
      STOT(2) = STOT(2) - NOZARE(1)/4.0
   END IF
ENDIF

IF (TYPE(2).EQ. 'INTERNAL. ') THEN
   IF (XNOZ(1).NE.XNOZ(2)) STOT(1) = STOT(1) - NOZARE(2)/4.0
   IF (NOZZLE.EQ.3) THEN
      IF (YNOZ(1).EQ.0.0) STOT(2) = STOT(2) - NOZARE(2)/4.0
   ELSE IF (NOZZLE.EQ.4) THEN
      STOT(2) = STOT(2) - NOZARE(2)/4.0
   END IF
ENDIF

END IF

FINISH CALCULATIONS ON ALL AREAS TO BE SENT BACK TO CALLING ROUTINE.

IF (NOZZLE.EQ.2) THEN
   IF (XNOZ(1).NE.XNOZ(2)) THEN
      SP1 = STOT(1)
      SPPI = 2 * E1 * MSPAN(1)
   ELSE
      SP2 = (1.0 - HOLE(2)) * 2.0 * STOT(2)
      SP3 = (1.0 - HOLE(3)) * 2.0 * STOT(3)
      SPP2 = (1.0 - HOLE(2)) * 2.0 * E1 * MSPAN(2)
      SPP3 = (1.0 - HOLE(3)) * 2.0 * E1 * MSPAN(3)
   END IF
ELSE
   IF (NOZZLE.EQ.3.AND.YNOZ(2).EQ.0.0) THEN
      HOLE(1) = HOLE(2)
      HOLE(2) = HOLE(3)
      HOLE(3) = HOLE(1)
   END IF
   SP1 = STOT(1)
   SP2 = (1.0 - HOLE(2)) * 2.0 * STOT(2)
   SP3 = (1.0 - HOLE(3)) * 2.0 * STOT(3)
\[ SPP2 = (1.0 - HOLE(2)) \times 2.0 \times E2 \times MSPAN(2) \]
\[ SPP3 = (1.0 - HOLE(3)) \times 2.0 \times E4 \times MSPAN(3) \]

CALCULATE SPP1
IF \((YMS\text{SPAN} \cdot \equiv 0.0)\) THEN
\[ SPP1 = 0.0 \]
ELSE
\[ X1 = X\text{NOZ}(1) \]
\[ Y1 = Y\text{NOZ}(1) \]
\[ X2 = X\text{NOZ}(2) \]
\[ Y2 = Y\text{NOZ}(2) \]
\[ MM\text{SPAN} = -1.0/MLINE1 \]
\[ BM\text{SPAN} = Y\text{MS\text{SPAN}} - MM\text{SPAN} \times XM\text{SPAN} \]
CALL LINECRO \((MM\text{SPAN}, BM\text{SPAN}, MLINE1, BLINE1, x4, y4)\)
CALL LINECRO \((F_L\text{SPAN}, BM\text{SPAN}, MLINE3, BLINE3, x3, y3)\)
\[ SPP1 = 0.5 \times (X1\times Y2 + X2\times Y3 + X3\times Y4 + X4\times Y1 - Y1\times X2 - Y2\times X3 - Y3\times X4 - Y4\times X1) \]
END IF
END IF

MAKE FINAL CALCULATIONS FOR ALL SPANS AND MSPANS.
IF \((\text{SPAN}(1) \lt 0.0)\) THEN \(\text{SPAN}(1) = 0.0\)
\[ \text{SPAN}(2) = (1.0 - HOLE(2)) \times \text{SPAN}(2) \]
\[ \text{SPAN}(3) = (1.0 - HOLE(3)) \times \text{SPAN}(3) \]
\[ \text{MSPAN}(2) = (1.0 - HOLE(2)) \times \text{MSPAN}(2) \]
\[ \text{MSPAN}(3) = (1.0 - HOLE(3)) \times \text{MSPAN}(3) \]

C Assign the correct variables to the variables from the calling
C routine COPPIE
C
Calculate estimated nose configuration
SCALE3 is a percentage measured from the nozzle to the tip of the
nose.
IF \((\text{SCALE3} \not= 1.0 \text{ OR SCALE \cdot EQ. 9999.})\) THEN
\[ \text{SCALE} = 1.0 - (-0.00129228 + 0.31836 \times (1.0 - \text{SCALE3}) + 4.41157 \times (1.0 - \text{SCALE3})^2 - 10.376 \times (1.0 - \text{SCALE3})^3 + 6.65561 \times (1.0 - \text{SCALE3})^4) \]
ELSE
\[ \text{SCALE} = 1.0 \]
END IF

Four nozzle configuration

\[
\begin{align*}
222 & \quad X \quad X \quad 222 \\
22 & \quad X \quad X \quad 22 \\
22 & \quad X \quad X \quad 22 \\
22 & \quad X \quad X \quad 22 \\
22 & \quad X \quad X \quad 22 \\
22222 & \quad X \quad X \quad 22222 \\
\end{align*}
\]

IF \((\text{NUM_F} \cdot \equiv 2 \text{ AND NUM_R \cdot EQ. 2})\) THEN
Half distance between fountain arms
IF \((E_1 \cdot EQ. 9999.)\) THEN \(E_1 = E1\)
IF \((E_3 \cdot EQ. 9999.)\) THEN \(E_3 = E4\)
IF \((E_4 \cdot EQ. 9999.)\) THEN \(E_4 = E2\)
Half angles between fountain arms
IF (THA_1 .EQ. 9999.) THA_1 = THETA_1
IF (THA_3 .EQ. 9999.) THA_3 = THETA_4
IF (THA_4 .EQ. 9999.) THA_4 = THETA_2

Half length of fountain arms
IF (Y_1 .EQ. 9999.) Y_1 = SPAN(1)
IF (Y_3 .EQ. 9999.) Y_3 = SPAN(3) * SCALE_3
IF (Y_4 .EQ. 9999.) Y_4 = SPAN(2)

Maximum spanwise extent of planform
IF (YP_1 .EQ. 9999.) YP_1 = MSPAN(1)
IF (YP_3 .EQ. 9999.) YP_3 = MSPAN(3) * SCALE_3
IF (YP_4 .EQ. 9999.) YP_4 = MSPAN(2)

Area affected by fountain arm
IF (SF_A1 .EQ. 9999.) SF_A1 = SP1
IF (SF_A3 .EQ. 9999.) SF_A3 = SP3 * SCALE
IF (SF_A4 .EQ. 9999.) SF_A4 = SP2

Potential area affected by fountain arm
IF (SP_FA1 .EQ. 9999.) SP_FA1 = SP1
IF (SP_FA3 .EQ. 9999.) SP_FA3 = SP3 * SCALE
IF (SP_FA4 .EQ. 9999.) SP_FA4 = SP2

Three nozzle configuration (One nozzle in front, two in rear)

```
  1 X X 222
 11 X X 2 2
 1 1 X X 2
  1 X 2
  1 X X 2
 111111 X X 222222
```

ELSE IF (NUM_F .EQ. 1 .AND. NUM_R .EQ. 2) THEN

Pieces at -- 4 have been moved to -- 3 due to hov_ge which when making calculations for a three nozzle configuration uses -- 3 instead of -- 4 in its calculations no matter which configuration is being used (1 X 2 or 2 X 1).

Half distance between fountain arms
IF (E_1 .EQ. 9999.) E_1 = E1
IF (E_3 .EQ. 9999.) E_3 = E2
IF (E_4 .EQ. 9999.) E_4 = 0.0

Half angles between fountain arms
IF (THA_1 .EQ. 9999.) THA_1 = THETA_1
IF (THA_3 .EQ. 9999.) THA_3 = THETA_2
IF (THA_4 .EQ. 9999.) THA_4 = 0.0

Half length of fountain arms
IF (Y_1 .EQ. 9999.) Y_1 = SPAN(1)
IF (Y_3 .EQ. 9999.) Y_3 = SPAN(2)
IF (Y_4 .EQ. 9999.) Y_4 = 0.0

Maximum spanwise extent of planform
IF (YP_1 .EQ. 9999.) YP_1 = MSPAN(1)
IF (YP_3 .EQ. 9999.) YP_3 = MSPAN(2)
IF (YP_4 .EQ. 9999.) YP_4 = 0.0

Area affected by fountain arm
IF (SF_A1 .EQ. 9999.) SF_A1 = SP1
IF (SF_A3 .EQ. 9999.) SF_A3 = SP2
IF (SF_A4 .EQ. 9999.) SF_A4 = 0.0
Potential area affected by fountain arm
IF (SP_FAI .EQ. 9999.) SP_FAI = SPPI
IF (SP_FA3 .EQ. 9999.) SP_FA3 = SPP2
IF (SP_FA4 .EQ. 9999.) SP_FA4 = 0.0

Three nozzle configuration (Two nozzles in front, one in rear)

```
  2 2 X X 1
  2 X X 1 1
  2 X 1
  2 X X 1
  22222 X X 11111
```

ELSE IF (NUM_F .EQ. 2 .AND. NUM_R .EQ. 1) THEN

Half distance between fountain arms
IF (E_1 .EQ. 9999.) E_1 = E1
IF (E_3 .EQ. 9999.) E_3 = E2
IF (E_4 .EQ. 9999.) E_4 = 0.0

Half angles between fountain arms
IF (THA_1 .EQ. 9999.) THA_1 = THETA1
IF (THA_3 .EQ. 9999.) THA_3 = THETA4
IF (THA_4 .EQ. 9999.) THA_4 = THETA2

Half length of fountain arm
IF (Y_1 .EQ. 9999.) Y_1 = SPAN(1)
IF (Y_3 .EQ. 9999.) Y_3 = SPAN(2) * SCALE3
IF (Y_4 .EQ. 9999.) Y_4 = 0.0

Maximum spanwise extent of planform
IF (YP_1 .EQ. 9999.) YP_1 = MSPAN(1)
IF (YP_3 .EQ. 9999.) YP_3 = MSPAN(2) * SCALE3
IF (YP_4 .EQ. 9999.) YP_4 = 0.0

Area affected by fountain arm
IF (S_FAI .EQ. 9999.) S_FAI = SP1
IF (S_FA3 .EQ. 9999.) S_FA3 = SP2 * SCALE
IF (S_FA4 .EQ. 9999.) S_FA4 = 0.0

Potential area affected by fountain arm
IF (SP_FAI .EQ. 9999.) SP_FAI = SPPI
IF (SP_FA3 .EQ. 9999.) SP_FA3 = SPP2 * SCALE3
IF (SP_FA4 .EQ. 9999.) SP_FA4 = 0.0

Two nozzle configuration (One nozzle in front, one in rear)

```
  1 X X 1
  11 X X 11
  1 1 X X 1
  1 X 1
  1 X X 1
  1 X 1
  11111 X X 11111
```

ELSE IF (NUM_F .EQ. 1 .AND. NUM_R .EQ. 1) THEN

Half distance between fountain arms
IF (E_1 .EQ. 9999.) E_1 = E1
IF (E_3 .EQ. 9999.) E_3 = 0.0
IF (E_4 .EQ. 9999.) E_4 = 0.0
C Half angles between fountain arms
IF (THA_1 .EQ. 9999.) THA_1 = THETA1
IF (THA_3 .EQ. 9999.) THA_3 = 0.0
IF (THA_4 .EQ. 9999.) THA_4 = 0.0
C Half length of fountain arms
IF (Y_1 .EQ. 9999.) Y_1 = SPAN(1)
IF (Y_3 .EQ. 9999.) Y_3 = 0.0
IF (Y_4 .EQ. 9999.) Y_4 = 0.0
C Maximum spanwise extent of planform
IF (YP_1 .EQ. 9999.) YP_1 = MSPAN(1)
IF (YP_3 .EQ. 9999.) YP_3 = 0.0
IF (YP_4 .EQ. 9999.) YP_4 = 0.0
C Area affected by fountain arm
IF (S_FA1 .EQ. 9999.) S_FA1 = 2.0 * SP1
IF (S_FA3 .EQ. 9999.) S_FA3 = 0.0
IF (S_FA4 .EQ. 9999.) S_FA4 = 0.0
C Potential area affected by fountain arm
IF (SP_FA1 .EQ. 9999.) SP_FA1 = 2.0 * SP1
IF (SP_FA3 .EQ. 9999.) SP_FA3 = 0.0
IF (SP_FA4 .EQ. 9999.) SP_FA4 = 0.0

Two nozzle configuration (Two nozzles in front)

```
 222  X  X  000
  2  2  X  X  0  0
  2  X  X  0  0
  2  X  X  0  0
  2  X  X  0  0
 22222  X  X  000
```

ELSE IF (NUM_F .EQ. 2 .AND. NUM_R .EQ. 0) THEN

C Half distance between fountain arms
IF (E_1 .EQ. 9999.) E_1 = E1
IF (E_3 .EQ. 9999.) E_3 = 0.0
IF (E_4 .EQ. 9999.) E_4 = 0.0
C Half angles between fountain arms
IF (THA_1 .EQ. 9999.) THA_1 = THETA1
IF (THA_3 .EQ. 9999.) THA_3 = 0.0
IF (THA_4 .EQ. 9999.) THA_4 = 0.0
C Half length of fountain arms
IF (Y_1 .EQ. 9999.) Y_1 = (SPAN(2) + SPAN(3) * SCALE3) / 2.0
IF (Y_3 .EQ. 9999.) Y_3 = 0.0
IF (Y_4 .EQ. 9999.) Y_4 = 0.0
C Maximum spanwise extent of planform
IF (YP_1 .EQ. 9999.) YP_1 = (MSPAN(2) + MSPAN(3) * SCALE3) / 2.0
  IF (YP_3 .EQ. 9999.) YP_3 = 0.0
  IF (YP_4 .EQ. 9999.) YP_4 = 0.0
C Area affected by fountain arm
IF (S_FA1 .EQ. 9999.) S_FA1 = (SP2 + SP3 * SCALE) / 2.0
IF (S_FA3 .EQ. 9999.) S_FA3 = 0.0
IF (S_FA4 .EQ. 9999.) S_FA4 = 0.0
C Potential area affected by fountain arm
IF (SP_FA1 .EQ. 9999.) SP_FA1 = (SPP2 + SPP3 * SCALE3) / 2.0
IF (SP_FA3 .EQ. 9999.) SP_FA3 = 0.0
IF (SP_FA4 .EQ. 9999.) SP_FA4 = 0.0
END IF

Core coordinates
XCORE = XCORE
YCORE = YCORE

Print out everything when necessary
IF (IFLAG.EQ.1) THEN
WRITE (6,35) STOT(1), STOT(2), STOT(3)
WRITE (6,40) XCORE, YCORE, SP1, SP2, SP3, SPP1, SPP2, SPP3, SPAN(1),
$ SPAN(2), SPAN(3), MSPAN(1), MSPAN(2), MSPAN(3)
WRITE (6,"*') XCORE "*') YCORE
WRITE (6,"*') M1 "') M2 " ') M3
WRITE (6,"*') THEETA1, "') THEETA2, " ') THEETA3,
$ THEETA4
WRITE (6,"*') E1 " ') E2, " ') E3
WRITE (6,"*') E1, E2, E3, E4
22 FORMAT (IX,"THERE ARE A TOTAL OF '11', NOZZLE ON THIS
$ AIRCRAFT.")
23 FORMAT (4X,"THERE ',A15,' IN THE ',A6,' WHICH 'A3,' ',A9)
32 FORMAT (IX,"POINT ',1X,' XSLICE ',1X,' YSLICE ',1X,' DX ',1X,
$ ',1X,' ',1X,' ',1X,' ',1X,' A ',1X,' B')
35 FORMAT (1X,"STOT(1) = ',F6.2,2X,"STOT(2) = ',F6.2,2X,"STOT(3) = ',
$ F6.2)
40 FORMAT (/1X,"XCORE = ',F10.2,4X,"YCORE = ',F5.2//1X,"SP1 = ',
$ F6.2,5X,"SP2 = ',F6.2,5X,"SP3 = ',F6.2//1X,"SPP1 = ',F6.2,5X,
$ SPP2 = ',F6.2,5X,"SPP3 = ',F6.2//1X,"SPAN1 = ',F6.2,5X,
$ SPAN2 = ',F6.2,5X,"SPAN3 = ',F6.2//1X,"MSPAN1 = ',F6.2,5X,
$ MSPAN2 = ',F6.2,5X,"MSPAN3 = ',F6.2)
END IF

500 RETURN

---

Integra

SUBROUTINE INTEGRA(A, B, C, DX, AREA_AB, AREA_C)
THIS SUBROUTINE CALCULATES THE AREA THAT IS BOUNDED BY TWO LINES
PLUS ANOTHER ONE AND ZERO.
C
AREA_AB = (A - B) * DX
C
REAL A, B, C, DX, AREA_AB, AREA_C
AREA_AB = (A - B) * DX
AREA_C = C * DX
RETURN
END

---

Diabar

Subroutine DIABAR (Iflag, lerror, X, Y, NPTS, xctr, nangle, dbar)
**ACRONYM:** DIABAR since DBAR is some reserved function.

**PURPOSE:** DBAR is the Angular Mean Diameter of a Planform. This routine will calculate DBAR and areas forward and aft of the DBAR point. An input file containing planform coordinates (in a clockwise direction starting from the nose) is required along with the x-position from which to calculate DBAR.

**PARAMETERS:**

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFLAG</td>
<td>I</td>
<td>I</td>
<td>----</td>
<td>User interaction flag:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 = user interaction (can run as a stand alone routine.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>otherwise = no user interaction (all inputs from the calling routine).</td>
</tr>
<tr>
<td>IERROR</td>
<td>I</td>
<td>O</td>
<td>----</td>
<td>Error flag returned by DIABAR. If IFLAG = 1, then an explanation of the problem is given at the time of the error.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 = no errors.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 = values of dbar &amp; area probably bad.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 = moved x-position of dbar point/nozzle point to geometric center.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3 = input NANGLE was inappropriate (set to absolute value or 1000).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4 = Points entered in wrong direction, routine aborted. Must enter points from left to right.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5 = 1st and last points not on x-axis, routine aborted.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6 = the sweeping ray intersected more than 4 points or zero points; values returned may be bad.</td>
</tr>
<tr>
<td>X (500)</td>
<td>R</td>
<td>feet</td>
<td>X-values of the planform (limit = 500). Values must start from the left and proceed counterclockwise for one half of the aircraft (the aircraft is assumed symmetrical).</td>
<td></td>
</tr>
<tr>
<td>Y (500)</td>
<td>R</td>
<td>feet</td>
<td>Y-values of the planform (limit = 500).</td>
<td></td>
</tr>
<tr>
<td>NPTS</td>
<td>I</td>
<td>----</td>
<td>The number of points read in from the planform data file.</td>
<td></td>
</tr>
<tr>
<td>XCTR</td>
<td>R</td>
<td>I</td>
<td>feet</td>
<td>The x-coordinate of the point at which DBAR and/or areas are to be calculated.</td>
</tr>
<tr>
<td>NANGLE</td>
<td>I</td>
<td>I</td>
<td>----</td>
<td>Number of angle divisions to use in the calculation of DBAR and/or areas.</td>
</tr>
<tr>
<td>DBAR</td>
<td>R</td>
<td>O</td>
<td>feet</td>
<td>Angular Mean Diameter of the planform.</td>
</tr>
<tr>
<td>FAREA</td>
<td>R</td>
<td>O</td>
<td>ft²</td>
<td>The planform area in front (to the left) of the XCTR point.</td>
</tr>
</tbody>
</table>
The planform area behind (to the right) of the XCTR point.

### LOCAL VARIABLES (in addition to the above parameters):

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHA</td>
<td>R</td>
<td>radians</td>
<td>Current sweep angle.</td>
</tr>
<tr>
<td>AREA1</td>
<td>R</td>
<td>ft²</td>
<td>Temporary variable for the total area.</td>
</tr>
<tr>
<td>ARM1</td>
<td>R</td>
<td>ft²</td>
<td>Partial area for the intersection point in consideration.</td>
</tr>
<tr>
<td>ARM2</td>
<td>R</td>
<td>ft²</td>
<td>See ARM1.</td>
</tr>
<tr>
<td>ARM3</td>
<td>R</td>
<td>ft²</td>
<td>See ARM1.</td>
</tr>
<tr>
<td>ARM4</td>
<td>R</td>
<td>ft²</td>
<td>The step angle (PI/NANGLE) for the DBAR and area calculations.</td>
</tr>
<tr>
<td>DELTA</td>
<td>R</td>
<td>radians</td>
<td>The geometric center (average x-value) based on the first and last points of the planform data file.</td>
</tr>
<tr>
<td>GEOCTR</td>
<td>R</td>
<td>feet</td>
<td>Temporary counter variable.</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td></td>
<td>XCTR position flag: 0 = XCTR inside planform.</td>
</tr>
<tr>
<td>IND</td>
<td>I</td>
<td></td>
<td>1 = XCTR behind planform and will cause farea to equal tlares and bkarea to equal 0 for output.</td>
</tr>
<tr>
<td>IND2</td>
<td>I</td>
<td></td>
<td>2 = XCTR forward of planform and will cause bkarea to equal tlares and farea to equal 0 for output.</td>
</tr>
<tr>
<td>IND3</td>
<td>I</td>
<td></td>
<td>Flag for setting XCTR = GEOCTR: 0 = Leave XCTR as is.</td>
</tr>
<tr>
<td>NINCEP</td>
<td>I</td>
<td></td>
<td>1 = Set XCTR = GEOCTR</td>
</tr>
<tr>
<td>NINMAR</td>
<td>I</td>
<td></td>
<td>Trouble message flag (for IFLAG = 1): 0 = No trouble in NINMAR or NINCEP.</td>
</tr>
<tr>
<td>PI</td>
<td>R</td>
<td></td>
<td>1 = NINMAR = 0, and/or NINCEP &gt; 4 or &lt; 1 (IERROR = 6).</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
<td>feet</td>
<td>The number of intercepts for a particular radius.</td>
</tr>
<tr>
<td>RINCEP (5)</td>
<td>R</td>
<td>feet</td>
<td>Last (previous) number of line intersections (goes with LASTR).</td>
</tr>
<tr>
<td>RINMR1-4</td>
<td>R</td>
<td>feet</td>
<td>Pi.</td>
</tr>
<tr>
<td>RLAST</td>
<td>R</td>
<td>feet</td>
<td>Current total radius point determined from RINCEP(1-4).</td>
</tr>
<tr>
<td>SLOPE (500)</td>
<td>R</td>
<td></td>
<td>The radius at the intersection of the ray and line in question. Last radii points at intercepts 1-4.</td>
</tr>
<tr>
<td>SUBST</td>
<td>R</td>
<td></td>
<td>Last total radius point. The slopes of the lines between data sets.</td>
</tr>
<tr>
<td>SUMR</td>
<td>R</td>
<td>feet</td>
<td>Temporary value for the calculation of the radii. The summation of the different radii values.</td>
</tr>
<tr>
<td>TEMP</td>
<td>R</td>
<td></td>
<td>Temporary variable (used in sort loops). The angle, theta, corresponding to each x,y data set.</td>
</tr>
<tr>
<td>THETA (500)</td>
<td>R</td>
<td>radians</td>
<td>The total planform area.</td>
</tr>
<tr>
<td>TLAREA</td>
<td>R</td>
<td>ft²</td>
<td>Temporary value for the calculation of the radii.</td>
</tr>
<tr>
<td>TRANS</td>
<td>R</td>
<td></td>
<td>Temp. variable for run again questions (for IFLAG = 1).</td>
</tr>
<tr>
<td>YESNO</td>
<td>C+3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
REAL ALPHA, AREAl, ARM1, ARM2, ARM3, ARM4, BKEARe, DBAR, DELTA, 
$ FRAREA, GEOCTR, PI, R, RINCEP(5), RINMAR1, RINMAR2, RINMAR3, 
$ RINMAR4, RLAST, SLOPE(500), SUBST, SUMR, TEMP, THETA(500), 
$ TLAREA, TRANS, X(500), XCTR, Y(500)
INTEGER IND, IND2, IND3, IERROR, IFLAG, NANGLE, NINCEP, NINMAR, 
$ NPTS

character*3 yesno
character*50 userfi

I/O with the user (IFLAG = 1).
if (iflag .eq. 1) then
  write(6,*),' This program is designed to work only for symmetrical
  write(6,*),' airplanes -- which are probably the only kind of
  write(6,*),' airplanes that you are working with, but I thought
  write(6,*),' that I'd tell you anyway.'
  write(6,*),' Enter coordinates for your aircraft in a clockwise
  $direction sequentially from the left (the nose of the aircraft).' end if
3  if (iflag .eq. 1) then
    write(6,*),' Input filename for X and Y coordinates:'
    read(*,5) userfi
    format(a50)
  endif
Set/Reset indicators.

\begin{verbatim}
ind = 0
ind2 = 0
ind3 = 0
if (iflag .eq. I) then
  write(6,*) 'How many angle divisions would you like in the half circle used in determining D-Bar and the area for the aircraft?
  $ Areas for the front and back of the aircraft are calculated using only half of the angles entered here.'
  read*, nangle
endif
error = 0
if (nangle.lt.l) then
  if (iflag.eq.0) then
    nangle = abs(nangle)
    error = 3
  if (nangle.lt.l) then
    nangle = 1000
  endif
else
  write(6,*) 'Get real! (Just can not let you do it Jeffrey.)'
  go to 6
endif
if (iflag.eq.l) then
  write(6,*) 'Input the X-coordinate of the point at which you wish D-Bar to be calculated from or the X-coordinate of the nozzle location for an area value.'
  read*, xctr
endif
if (iflag.eq.l) then
  write(6,*) ' ' 
endif
\end{verbatim}
x(I) = X(I)
I = I + 1
goto 10
npts = I-1

END IF

geoctr = (x(1)+x(npts))/2.0

Set x-position for dbar calculation equal to GEOCTR if points outside of planform and/or instructed to do so.

if (ind2.eq.1) XCTR = geoctr

Move input x-values to place XCTR at 0,0.

x(I) = x(I) - XCTR

Find theta's.
do 20 I=2,npts

x(I) = x(I) - XCTR
if (abs(x(I)).le.0.0001) then
  theta(I) = pi/2
else if (x(I).lt.0.0) then
  theta(I) = abs(atan (y(I)/x(I)))
else if (x(I).gt.0.0) then
  theta(I) = pi - abs(atan (y(I)/x(I)))
endif
if (iflag.eq.1) then
  write(6,*)'angle to coordinate = ',theta(I)*180.0/pi
endif

Find slopes.
if (x(I).eq.x(I-1)) then
  slope(I) = 9999.9
  slope of 9999.9 signifies an infinite slope (vertical line).
else
  slope(I) = (y(I)-y(I-1))/(x(I)-x(I-1))
endif
continue

Catching faulty input so as to avoid inaccurate data if possible.
npts = I-1
if (x(1).gt.x(npts)) then
  if (iflag.eq.1) then
    write(6,*)'Error in data. Please enter coordinates from left to right.'
  go to 3
else
  ierror = 4
  go to 500
endif
else
  theta(1) = 0
endif
if (y(1).ne.y(npts).or.y(1).ne.0.or.y(npts).ne.0) then
  if (iflag.eq.1) then
    write(6,*)'Please, try to follow the directions. Half an airplane on the x-axis.
$i.e. both your first and last y values should be 0. Thank you.'
  go to 3
else
  theta(1) = 0
endif
else
  ierror = 5
  go to 500
endif
endif
if (x(npts).le.0.or.x(1).ge.0) then
  if (iflag.eq.0) then
    ierror = 2
    if (x(npts).lt.0) then
      ind = 1
    else
      ind = 2
    endif
  ind2 = 1
  rewind (7)
goto 9
else
  write(6,*),''Are you sure that you want your point outside of
$ the aircraft (or at the very edge)?''
  read (*,1000) yesno
  if (yesno(1:1).eq.'y') then
    write(6,*),'With this value D-Bar is inaccurate.'
    write(6,*),'Along with the area values. An accurate
$ area measurement can be taken at the edge of the aircraft or
$ inside. While an accurate D-Bar measurement should be taken from
$ the center of gravity or thereabouts.'
  else
    write(6,*),''Would you like your point to calculate the
$ total area from the geometric center?This is probably the best
$ option if your initial point is outside the aircraft, since
$ D-Bar is invalid. ' 
    read (*,1000) yesno
  if (yesnc(1:1).eq.'y') then
    ind = 1
  else
    ind = 2
  endif
  ind2 = 1
  rewind (7)
goto 9
else
  rewind (7)
goto 8
endif
endif
endif
if (iflag.eq.1) then
  write(6,*),'' data points received from file.'
endif

Determine intersections/ # of int's/ radii.
alpha = 0.0
trans = 0.0
subst = 0.0
r = 0.0
ninmar = 1
lastr = abs(x(1))
do 70 I = 1, nangle
    nincep = 0
    alpha = alpha + delta
    if (alpha.gt.pi) then
        alpha = pi
    endif
    do 80 J = 1, npts-1
        if (theta(J).lt.alpha.and.alpha.le.theta(J+1).or.theta(J+1).le.alpha
           .and.alpha.lt.theta(J)) then
            nincep = nincep + 1
            if (slope(J+1).ge.9999.0) then
                Vertical line!
                rincep(nincep) = abs(x(j)/cos(alpha))
            else
                if (slope(J+1).eq.-tan(alpha)) then
                    The intersection is a line (instead of a point).
                    rincep(nincep) = abs((x(J) +x(J+1))/2*cos(alpha))
                endif
            else
                if ((pi/2.0 - 0.01).le.alpha.and.alpha.le. (pi/2.0 + 0.01)) then
                    rincep(nincep) = abs(-slope(J+1)*x(J)+y(J))
                else
                    if (alpha.lt.pi/2.0) then
                        trans = (-slope(J+l)*x(J) + y(J))/(abs(tan(alpha))-slope(J+l))
                    else
                        trans = (-slope(J+l)*x(J) + y(J))/(abs(tan(alpha))-slope(J+l))
                    endif
                    rincep(nincep) = abs(trans/cos(alpha))
                endif
            endif
        endif
    enddo
70 continue
80 continue

Determining # of intercepts at each angle and the radius thereof
if (nincep.eq.1) then
    r = abs(rincep(1))
    areal = ((rlast+r)/2.0)**2.0*pi/nangle
else if
    $(nincep.eq.2.and.rincep(1).le.(rincep(2)+0.01).and.(rincep(2)-$ 0.01).le.rincep(1)) then
        r = abs(rincep(1))
        areal = ((rlast+r)/2.0)**2.0*pi/nangle
    else if (nincep.eq.2.and.rincep(1).ne.rincep(2)) then
        areal = ((rlast+r)/2.0)**2.0*pi/nangle
    endif
r = abs(rincep(2) - rincep(1))
arm1 = abs(((rincep(1) + rinmrl)/2.0)**2.0*pi/nangle
arm2 = abs(((rincep(2) + rinmr2)/2.0)**2.0*pi/nangle
areal = abs(arm2 - arm1)
else if (nincep.eq.3) then
c Sort radii values (three intersection points).
do 90 K = 1,2
do 100 L = K+1,3
if (rincep(K).gt.rincep(L)) then
temp = rincep(L)
  rincep(L) = rincep(K)
  rincep(K) = temp
endif
100 continue
90 continue
if (rincep(1).eq.rincep(2)) then
  r = abs(rincep(3))
  areal = ((rlast+r)/2.0)**2.0*pi/nangle
else
  r = abs(rincep(1) + rincep(3) - rincep(2))
  if (ninmar.eq.1.or.ninmar.eq.2) then
    areal = ((rlast+r)/2.0)**2.0*pi/nangle
  else if (ninmar.eq.3) then
    arm1 = abs(((rincep(1) + rinmrl)/2.0)**2.0*pi/nangle
    arm2 = abs(((rincep(2) + rinmr2)/2.0)**2.0*pi/nangle
    arm3 = abs(((rincep(3) + rinmr3)/2.0)**2.0*pi/nangle
    areal = arm1 - arm2 + arm3
  else if (ninmar.eq.4) then
    arm1 = abs(((rincep(1) + rinmrl)/2.0)**2.0*pi/nangle
    arm2 = abs(((rincep(2) + rinmr2)/2.0)**2.0*pi/nangle
    arm3 = abs(((rincep(3) + rinmr3)/2.0)**2.0*pi/nangle
    areal = arm1 - arm2 + arm3
  else
    areal = ((rlast+r)/2.0)**2.0*pi/nangle
  endif
endif
e else if (nincep.eq.4) then
c Sort radii values (four intersection points).
do 110 K = 1,3
do 120 L = K+1,4
if (rincep(K).gt.rincep(L)) then
temp = rincep(L)
  rincep(L) = rincep(K)
  rincep(K) = temp
endif
120 continue
110 continue
if (rincep(1).eq.rincep(2).and.rincep(3).eq.rincep(4)) then
  r = abs(rincep(3))
  areal = ((rlast+r)/2.0)**2.0*pi/nangle
else if (rincep(1).eq.rincep(2) then
  r = abs(rincep(1) + rincep(4) - rincep(3))
else if (ninmar.eq.4) then
  arm1 = abs(((rincep(1) + rinmrl)/2.0)**2.0*pi/nangle
  arm3 = abs(((rincep(3) + rinmr3)/2.0)**2.0*pi/nangle

arm4 = abs(((rincep(4) + rinmr4)/2.0)**2.0*pi/nangle
areal = arm1 - arm3 + arm4
else if (ninmar.eq.3) then
    arm1 = abs(((rincep(1) + rinmr1))/2.0)**2.0*pi/nangle
    arm3 = abs(((rincep(3) + rinmr2)/2.0)**2.0*pi/nangle
    arm4 = abs(((rincep(4) + rinmr3)/2.0)**2.0*pi/nangle
    areal = arm1 - arm3 + arm4
else if (ninmar.eq.2.or.ninmar.eq.1) then
    areal = ((rlast+r)/2.0)**2.0*pi/nangle
else
    ind3 = 1
    r = abs(rincep(1))
    areal = ((rlast+r)/2.0)**2.0*pi/nangle
endif
endif
endif
endif
sumr = sumr + r

Calculate the total area of the aircraft (or planform).
tlarea = areal + tlarea
if (alpha.le.pi/2.0) then
    frarea = areal + frarea
else
    bkarea = areal + bkarea
endif

Resetting for next go-around.
rlast = r
ninmar = nincep
if (nincep.eq.4) then
    go to 300
else if (nincep.eq.3) then
    go to 310
else if (nincep.eq.2) then
    go to 320
else if (nincep.eq.1) then
    go to 330
else
    Blow Up!
    NINCEP is not equal to 1,2,3, or 4
    if (nincep.eq.0 .and. iflag.eq.1) then
        write(6,*),'trouble'
    else if (nincep.gt.4 .and. iflag.eq.1) then
        write(6,*),'more trouble'
    else if (iflag .eq. 1) then
        write(6,*),'this statement should never print up'
    endif
    ind3 = 1
    ierror = 6
endif

300    rinmr4 = rincep(4)
310    rinmr3 = rincep(3)
320    rinmr2 = rincep(2)
330    rinmr1 = rincep(1)
continue

Warn user in case of possible error.

if (ind3.eq.1.and.iflag.eq.1) then
  write(6,*),'Possible error in D-Bar calculation from odd
$ airplane configurations or irregular placement of center
$ point. Try evaluation again except with a different
$ number of angles this time.'
endif

D-Bar calculation.
dbar = 2*sumr/nangle
if (iflag.eq.1) then
  write(6,*),' '　
dwrite(6,*),'D-Bar =',dbar
  write(6,*),' '　
endif
if (ind.eq.1) then
  frarea = tlarea
  bkarea = 0.0
else if (ind.eq.2) then
  bkarea = tlarea
  frarea = 0.0
endif
if (iflag.eq.1) then
  write(6,*),'Total aircraft area =',tlarea
  write(6,*),'Front aircraft area =',frarea
  write(6,*),'Back aircraft area =',bkarea
  write(6,*),' '　
  write(6,*),'Would you like to use the same file again?'
  read (*,1000) yesno
  if (yesno(1:1).eq.'y') then
    rewind(7)
    goto 6
  else
    close (7)
  endif
else
  write(6,*),'Would you like to use a different file?'
  read (*,1000) yesno
1000 format(a3)
  if (yesno(1:1).eq.'y') then
    goto 3
  endif
endif
if (ind3.eq.1) then
  ierror = 6
endif
close (7)
500 return
end

**HCALL**

SUBROUTINE HCALL

C--人生 collects all correct variables,
C ACRONYM: Hover CALLing routine
C ---
C PURPOSE: The purpose of HCALL is to set up all the correct variables,
based on the configuration, that are to be sent to the
HOV_GE routine. Another reason for this subroutine is all
subroutines that need to be called have variables with the
same name which need to be kept separate when using common
blocks to pass variables back and forth between routines.

**LOCAL VARIABLES** (in addition to the above parameters):

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<list of local variables in routine>

**GLOBAL VARIABLES** (in addition to the above parameters and local vars):

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**FIGPIE Common Block**

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DESCRIPTION**

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **B_B**: Width of Body
- **B WB**: Width of Wing-Body
- **DB B**: Dbar of Body
- **DB WB**: Dbar of Wing-Body
- **DE FR**: Effective Diameter of Front & Rear
- **E_1**: Half distance between adjacent jets (1)
- **E_3**: Half distance between adjacent jets (3)
- **E_4**: Half distance between adjacent jets (4)
- **KR**: Body contour factor
- **L WB**: Length of Wing-Body
- **NUM**: Total NUMBER of jets
- **PER FR**: Total perimeter of jets
- **PP LID**: Ratio of perimeter enclosed by lids
- **PF FR**: Jet Pressure Ratio for all jets
- **S_B**: Area of Body
- **S_FA1**: Area affected by fountain arm (1)
- **S_FA3**: Area affected by fountain arm (3)
- **S_FA4**: Area affected by fountain arm (4)
- **S_FR**: Total jet exit area
- **S_JP**: Area enclosed by Jet Pattern
- **S_LID**: Area enclosed by LIDS
- **S WB**: Area of Wing-Body
- **SP_FA1**: Potential area affected by fountain arm (1)
- **SP FA3**: Potential area affected by fountain arm (3)
- **SP FA4**: Potential area affected by fountain arm (4)
- **SP JP**: Actual surface area within area enclosed by nozzles
- **SPLY F**: SPLAy angle of Front jet
- **THA_1**: Half angle between jets (1)
- **THA_3**: Half angle between jets (3)
- **THA_4**: Half angle between jets (4)
- **WL JP**: Width to Length ratio of Jet Pattern
- **X FR**: Distance between Front & Rear jets
- **XCA CG**: Distance of center of area ahead of
- **Y_1**: Spanwise extent of planform on fountain arm (1) center line.
Spanwise extent of planform on fountain arm (3) center line.
Spanwise extent of planform on fountain arm (4) center line.
Distance between Front jets
Distance between Rear jets
Lateral distance from Body to Front jets (for external jets)
Max spanwise extent of planform (1)
Max spanwise extent of planform (3)
Max spanwise extent of planform (4)
Lateral distance from Wingbody to Front jets (for external jets)
Height of wing above nozzle

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AJ</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
</tr>
<tr>
<td>BF1</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>BF3</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>BF4</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>BFPI</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>BFP3</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>BFP4</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>DB</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>DC</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>DE</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>DH</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>E</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>HEIGHT</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>HW</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>KRB</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>L</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>NJETS</td>
<td>I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PER</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>PP</td>
<td>R</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>PR</td>
<td>R</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>SB</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
</tr>
<tr>
<td>SC</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
</tr>
<tr>
<td>SCP</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
</tr>
<tr>
<td>SFA1</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
</tr>
<tr>
<td>SFA3</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
</tr>
<tr>
<td>SFA4</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
</tr>
<tr>
<td>SFA1P</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
</tr>
<tr>
<td>SFA3P</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
</tr>
<tr>
<td>SFA4P</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
</tr>
</tbody>
</table>

DESCRIPTION

Total jet exit area.
Half length of fountain arm (1).
Half length of fountain arm (3).
Half length of fountain arm (4).
Maximum spanwise extent of fountain arm (1).
Maximum spanwise extent of fountain arm (3).
Maximum spanwise extent of fountain arm (4).
Body D bar.
Configuration D bar.
Equivalent diameter of jets.
Delta height of wing.
Length to width ratio of jet pattern.
Altitude at which calculations are to be made.
Half width of body for jets outside outside of body, else use half distance between adjacent jets.
Body contour factor
Length of configuration.
Total number of jet on config.
Total perimeter of jets (all jets)
Fraction of lid perimeter enclosed
Jet pressure ratio.
Planform area of body.
Area enclosed by jet pattern.
Actual surface area inside jet pattern.
Area affected by fountain ares(1).
Area affected by fountain ares(3).
Area affected by fountain ares(4).
Potential surface area between jets (1)
Potential surface area between jets (3)
Potential surface area between jets (4)
C SL R I Sq Ft Area enclosed by lids.
C SP R I Sq Ft Planform area of configuration
C SPLAY R I Deg Front jet splay angle.
C THA1 R I Deg Half angles between jets (1).
C THA3 R I Deg Half angles between jets (3).
C THA4 R I Deg Half angles between jets (4).
C WB R I Ft Width of body.
C WC R I Ft Width of configuration.
C X1 R I Ft Half distance between adjacent jets (1).
C X3 R I Ft Half distance between adjacent jets (3).
C X4 R I Ft Half distance between adjacent jets (4).
C XCA R I Ft Distance center of area ahead of CG
C YF R I Ft Distance between front jets.
C YR R I Ft Distance between rear jets.

RESPIE Common Block
NAME TYPE I/O UNITS DESCRIPTION
H I O ---- Height number counter
HT(500) R O Ft Actual height value

FILES USED:
LOGICAL UNIT I/O DESCRIPTION
(none)

COMMONS USED:
NAME DESCRIPTION
FIGPIE conFIGuration for Power Induced Effects - Contains all variables needed to define the configuration and other parameters of the aircraft.
HOVPIE HOVer variables for Power Induced Effects - Contains variables which are sent to the HOV_GE subroutine.
RESPIE RESults from all POwer Induced Effects subroutines - Contains results from each subroutine along with variables for height, velocity, jet deflection angle, and angle of attack and their counters.

CALLED BY:
NAME DESCRIPTION
COPPIE Controls execution and coordination of Power Induced Effects Module

ROUTINES CALLED:
NAME DESCRIPTION
HOV_GE Calculates hover lift increments for OGE and GE suckdown, and fountain effects

NOTES: None.
REFERENCES:
1) None.
ENVIRONMENT:
FORTRAN 77, VAX 6800, VAX 11/785, SGI IRIS 4D series.
NON-STANDARD CODE:
AUTHOR(S):
Kipp E. Howard (KEH), Cal Poly San Luis Obispo, NASA Ames Moffett

REVISION HISTORY:
DATE    INITIALS & DESCRIPTION
04/25/90 KEH -- Completed initial coding and debugging
07/13/91 KEH -- Eliminated extraneous spacing and fixed comments

#include "figpie.inc"
#include "hovpie.inc"
#include "respie.inc"

Assign local HOV_GE variables to their respective global configuration
variables.
Start with variables that do not change with planform configuration
HEIGHT = HT(H)
PR = PR_Fr
DE = DE_Fr
AJ = S_Fr
PER = PER_Fr
NJETS = NUM
SFA1 = S_Fa1
SFA3 = S_Fa3
SFA4 = S_Fa4
SFA1P = SP_Fa1
SFA3P = SP_Fa3
SFA4P = SP_Fa4
X1 = E_1
X3 = E_3
X4 = E_4
BF1 = Y_1
BF3 = Y_3
BF4 = Y_4
BFP1 = YP_1
BFP3 = YP_3
BFP4 = YP_4
THA1 = THA_1
THA3 = THA_3
THA4 = THA_4
E = 1.0/WL_JP
SC = S_JP
SCP = SP_JP
SL = S_LID
PF = PP_LID
SPLAY = SPLY_F
YF = Y_F
YR = Y_R
XCA = XCA_CG
KRB = KR

Assign variables that do change with planform configuration
DC = DB_WB
DB = DB_B
SP = S_WB
SB = S_B
WC = B_WB
WB = B_B
L = L_WB
DH = Z_W
SUBROUTINE HOV_GE

PURPOSE: Calculation of jet induced lift loss for powered lift aircraft due to jet effects in ground proximity.

LOCAL VARIABLES (in addition to the above parameters):

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DI</td>
<td>R</td>
<td>-</td>
<td>Ft</td>
<td>Average diameter of individual nozzles.</td>
</tr>
<tr>
<td>EAV</td>
<td>R</td>
<td>-</td>
<td>Ft</td>
<td>Average half distance between adjacent jets.</td>
</tr>
<tr>
<td>HDR</td>
<td>R</td>
<td>-</td>
<td>----</td>
<td>Aircraft Height/Effective diameter</td>
</tr>
<tr>
<td>HFLAG</td>
<td>I</td>
<td>-</td>
<td>----</td>
<td>HFLAG= 1: the 2nd point tried.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>= 2: at least 3rd point tried.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>= 3: on transition line.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>= 4: past transition line.</td>
</tr>
<tr>
<td>HP</td>
<td>R</td>
<td>O</td>
<td>Ft</td>
<td>Height to which tangent line intersects dL/T=0 (Used in Hover h'prime method, obviously)</td>
</tr>
<tr>
<td>IERROR</td>
<td>I</td>
<td>-</td>
<td>----</td>
<td>IERROR= 1: Using lower point and possible error</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>= 2: Past H'/De on first try.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>= 3: Tangent line does not intersect lower curve</td>
</tr>
<tr>
<td>ITRAN</td>
<td>I</td>
<td>-</td>
<td>----</td>
<td>ITRAN = 0: Intersection of tangent line has not been found.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>= 1: Intersection of tangent line has been past.</td>
</tr>
<tr>
<td>KL</td>
<td>R</td>
<td>-</td>
<td>----</td>
<td>Augmentation factor for LIDs</td>
</tr>
<tr>
<td>KLA</td>
<td>R</td>
<td>-</td>
<td>----</td>
<td>First augmentation factor for LIDs</td>
</tr>
<tr>
<td>KLB</td>
<td>R</td>
<td>-</td>
<td>----</td>
<td>Second augmentation factor for LIDs</td>
</tr>
<tr>
<td>KP</td>
<td>R</td>
<td>-</td>
<td>----</td>
<td>Constant used in h' method.</td>
</tr>
</tbody>
</table>
| KS   | R    | -   | ----  | Ratio of the multijet suckdown to
that for an equivalent single jet.
Constant exponent used in h' method
Exponent in expression for multijet
suckdown.
Total lift increment resulting
from all the fountain arms.
Lift loss resulting from fountain
arm (1).
Lift loss resulting from fountain
arm (3).
Lift loss resulting from fountain
arm (4).
Total lift loss resulting from
fountain core.
Total Lift loss resulting from
fountain core (Method A).
Lift loss from the fountain core
due to fountain arm (1).
Lift loss from the fountain core
due to fountain arm (3).
Lift loss from the fountain core
due to fountain arm (4).
Total Lift loss resulting from
fountain core (Method B).
Lift loss from the fountain core
due to fountain arm (1).
Lift loss from the fountain core
due to fountain arm (3).
Lift loss from the fountain core
due to fountain arm (4).
Total lift loss from all fountain
arms.
Lift loss from fountain arms, used
determine transition line.
Lift loss from fountain arms, used
determine transition line (Lower
line).
Lift loss from fountain arms, used
to determine transition line (Transition line).
Lift loss from fountain arms, used
determine transition line. (Upper
line).
Total lift loss from lift
improvement devices.
Net lift loss for aircraft.
Total lift resulting from suckdown
out of ground effect.
Total lift resulting from OGE
suckdown effect for body alone.
Total lift resulting from suckdown
out of ground effect for wingbody.
Total lift loss resulting from
suckdown.
Lift loss resulting from suckdown
using the body planform and
multiple jets.
Lift loss resulting from suckdown
using the body planform and a single jet.

Lift loss resulting from suckdown using the wingbody planform and a single jet.

Often used parameter = HDE/(DB/DE-1)

Height ratio for fountain arm (1).

Height ratio for fountain arm (3).

Height ratio for fountain arm (4).

Slope of temporary transition line.

Slope of lower line.

Previous slope of temporary transition line.

Slope of transition line.

Slope of upper line.

Distance to center of jet pattern when jet reaches the ground.

X-intercept of current line.

Previous X-intercept of current line.

Previous X-coordinate of last point.

X-coordinate of upper line.

Previous Y-coordinate of last point.

Y-coordinate of Upper Line

GLOBAL VARIABLES (in addition to the above parameters and local vars):

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>AJ</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
<td>Total jet exit area.</td>
</tr>
<tr>
<td>BF1</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Half length of fountain arm (1).</td>
</tr>
<tr>
<td>BF3</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Half length of fountain arm (3).</td>
</tr>
<tr>
<td>BF4</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Half length of fountain arm (4).</td>
</tr>
<tr>
<td>BFP1</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Maximum spanwise extent of fountain arm (1).</td>
</tr>
<tr>
<td>BFP3</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Maximum spanwise extent of fountain arm (3).</td>
</tr>
<tr>
<td>BFP4</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Maximum spanwise extent of fountain arm (4).</td>
</tr>
<tr>
<td>DB</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Body D bar.</td>
</tr>
<tr>
<td>DC</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Configuration D bar.</td>
</tr>
<tr>
<td>DE</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Equivalent diameter of jets.</td>
</tr>
<tr>
<td>DH</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Delta height of wing.</td>
</tr>
<tr>
<td>E</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Length to width ratio of jet pattern.</td>
</tr>
<tr>
<td>HEIGHT</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Altitude at which calculations are to be made.</td>
</tr>
<tr>
<td>HW</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Half width of body for jets outside outside of body, else use half distance between adjacent jets.</td>
</tr>
<tr>
<td>KRB</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Body contour factor</td>
</tr>
<tr>
<td>L</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Length of configuration.</td>
</tr>
<tr>
<td>NJETS</td>
<td>I</td>
<td>I</td>
<td></td>
<td>Total number of jet on config.</td>
</tr>
<tr>
<td>PER</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Total perimeter of jets (all jets)</td>
</tr>
<tr>
<td>PP</td>
<td>R</td>
<td>I</td>
<td></td>
<td>Fraction of lid perimeter enclosed</td>
</tr>
<tr>
<td>NAME</td>
<td>TYPE</td>
<td>I/O</td>
<td>UNITS</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
<td>-----</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>H</td>
<td>I</td>
<td>O</td>
<td>-----</td>
<td>Height number counter</td>
</tr>
<tr>
<td>H_HPRI</td>
<td>R</td>
<td>O</td>
<td>Ft</td>
<td>Height to which tangent line intersects $dL/T=0$ (Used in Hover hprime method, obviously)</td>
</tr>
<tr>
<td>H_LT(500)</td>
<td>R</td>
<td>O</td>
<td>-----</td>
<td>Total lift loss calculated while in hover.</td>
</tr>
<tr>
<td>H_LT(500)</td>
<td>R</td>
<td>O</td>
<td>-----</td>
<td>Lift gain due to fountain effects while in hover.</td>
</tr>
<tr>
<td>H_LT(500)</td>
<td>R</td>
<td>O</td>
<td>-----</td>
<td>Lift gain due to the addition of LIDs while in hover.</td>
</tr>
<tr>
<td>H_LTOG</td>
<td>R</td>
<td>O</td>
<td>-----</td>
<td>Lift loss OGE effect while in hover.</td>
</tr>
<tr>
<td>H_LTS(500)</td>
<td>R</td>
<td>O</td>
<td>-----</td>
<td>Lift loss due to suckdown while in hover.</td>
</tr>
</tbody>
</table>

**RESPIE Common Block**

**DESCRIPTION**

- **HOVPIE**: HOver variables for Power Induced Effects - Contains variables which are sent to the HOV_GE subroutine.
- **RESPIE**: Results from all POWer Induced Effects subroutines.
Contains results from each subroutine along with variables for height, velocity, jet deflection angle, and angle of attack and their counters.

**CALLED BY:**
**NAME** | **DESCRIPTION**
--- | ---
**HCALL** | Isolates and controls execution of HOV_GE

**ROUTINES CALLED:**
**NAME** | **DESCRIPTION**
--- | ---
None.

**NOTES:** None.

**REFERENCES:**

**ENVIRONMENT:**
FORTRAN 77, VAX 6800, VAX 11/785, SGI IRIS 4D series.

**NON-STANDARD CODE:**
?

**AUTHOR(S):**
Douglas A. Wardwell (DAW), NASA Ames Research Center
Kipp E. Howard (KEH), Cal Poly San Luis Obispo, NASA Ames Moffett

**REVISION HISTORY:**
**DATE** | **INITIALS & DESCRIPTION**
--- | ---
11/28/89 | DAW -- FORTRAN version of Dick Kuhn's basic program.
04/22/90 | KEH -- Modifications and corrections of routine
07/13/91 | KEH -- Eliminated extraneous spacing and fixed comments

```fortran
#include "hovpie.inc"
#include "respie.inc"
```

**Calculation variables.**

- INTEGER IERROR, I, HFLAG, ITRAN
- REAL LTS, LTN, LTA, LTA1, LTA3, LTA4, LTCA, TEMP,
- LTCA1, LTCA3, LTCA4, LTCB, LTCB1, LTCB3, LTCB4, SHR1,
- SHR3, SHR4, KCA, KCB, KLA, KLB, KL, GS, KS, LS,
- LTA, LTF, LTL, SLOPE, SLOPEL, SLOPEU, YP, XP, KP, LP,
- LTFF, LTFFP, LTFFU, LTFF, XINTER, YUL, YTRAN, LTFTP, SLOPET,
- LTGB, LTGBW, LTGB, MESS, LTSM, LTSSB, LTSSW
- LTSMB, EAV, DI, X1C, HDA, HP, YFC, SCC, EC, HPDE, MTD, LCA, LCB

**SAVE** LTFTP, SLOPET, XINTER, XP, YP, YTRAN, HFLAG, ITRAN

```fortran
IF (H .EQ. 1) THEN
  HFLAG = 0
  ITRAN = 0
END IF
```

```fortran
IF (E .LT. 1.0) E = 1.0 / E
```

**Estimated HOVER Suckdown and Fountain Lift (?) using the**
**Basic method -- X/D > 3.0, or h' method -- X/D <= 3.0**

```fortran
IF (DH .NE. 0.0) THEN
  High Wing configuration
  LTGB = -0.00022*SQRT(SB/AJ)*(PR**-0.64)*(PER/DE)**1.58
ELSE
  Low Wing configuration
```
LTOGB = -0.00022*SQRT(SP/AJ)*(PR**-0.64)*(PER/DE)**1.58

END IF

C

LTOGBW = -0.00022*SQRT(SP/AJ)*(PR**-0.64)*(PER/DE)**1.58
LTOG = LTOGB + (LTOGBW-LTOGB)*(1.0-0.4*SQRT(DH/DE))
LIF = 0.0
LTL = 0.0
LTN = 0.0
IERROR = 0
HDR = HEIGHT/DE

C

C Lift loss due to suckdown
IF (DH .NE. 0.0) THEN
  High wing configuration
  MESS = (HDR)/(DB/DE-1.0)
  IF (NJETS .EQ. I) THEN
    KS = 1.0
  ELSE
    LS = -1.7*((WB/L)*((SB/(WB*L))**0.36)**1.38
    KS = 4.5*(MESS)**0.25 * (1.0-(HDR)/
      (0.08*(DB/DE)*((WB/L)))**LS)
  END IF
  ELSE
    Assume PR = 2.0
    LTSMB = KS * (-0.015) * (MESS**-1.96)
    LTSSB = -0.015*(((HEIGHT+DH)/DE)/(DB/DE-1.0))**-1.96
    LTSSWB = -0.015*(((HEIGHT+DH)/DE)/(DC/DE-1.0))**-1.96
  ELSE
    LTSMB = KS *(-0.015) * (MESS**-(2.2-.24*(PR-1.0)))
    LTSSB = -0.015*(((HEIGHT+DH)/DE)/(DB/DE-1.0))
      **-(2.2-.24*(PR-1.0))
    LTSSWB = -0.015*(((HEIGHT+DH)/DE)/(DC/DE-1.0))
      **-(2.2-.24*(PR-1.0))
  END IF
  LTS = LTSMB+LTSSWB-LTSSB
END IF

C

C Low wing configuration
  MESS = (HDR)/(DC/DE-1.0)
  IF (NJETS .EQ. 1) THEN
    KS = 1.0
  ELSE
    LS = -1.7*((WC/L)*((SP/(WC*L))**0.36)**1.38
    KS = 4.5*(MESS)**0.25 * (1.0-(HDR)/
      (0.08*(DC/DE)*((WC/L)))**LS)
  END IF
  IF (PR .GE. 2.0) THEN
    LTS = KS * (-0.015)*MESS**-1.96
  ELSE
    LTS = KS *(-0.015) * (MESS**-(2.2-.24*(PR-1.0)))
  END IF

C

C ENDIF
DI = DE/SQRT(REAL(NJETS))
EAV = (2*X1+X3+X4)/NJETS
IF (NJETS .EQ. 2) EAV = X1

*** Fountain Effects Calculation.
IF (EAV/DI .GT. 3.0) THEN
   IF (NJETS .EQ. 2) THEN
      *** Jet Fountain -- 2 jets (Basic method).
      IF (SPLAY .GT. 0) THEN
         X1C = X1-HEIGHT*TAN(SPLAY/57.296)
         IF (X1C .LT. 0.0) X1C=0.0
         SFA1 = SFA1*X1C/X1
         SC = SC*X1C/X1
      END IF
      IF (SFA1 .EQ. 0.0) THEN
         LTA1 = 0.0
         GO TO 1210
      ELSE
         X1=X1C
      END IF
   END IF
   END IF

   LTF = LTA1
   LTN = LTOG+LTS+LTF
   HP = 1.0
   GO TO 2330
ELSE IF (NJETS .GT. 2) THEN
   *** Jet Fountain -- more than 2 jets (Basic method).
   *** Fountain Arms.
   SHR1 = X1/(X1+HEIGHT)
   LTA1 = (2.0/NJETS*((BFP1 * SFA1)/(X1 * SFA1P))**0.835) *
         SHR1**2.0*BF1/SQRT(BF1*BF1+(X1+HEIGHT)**2.0)
   LTA3 = (2.0/NJETS*((BFP3 * SFA3)/(X3 * SFA3P))**0.835) *
         SHR3**2.0*BF3/SQRT(BF3*BF3+(X3+HEIGHT)**2.0)
   LTA4 = (2.0/NJETS*((BFP4 * SFA4)/(X4 * SFA4P))**0.835) *
         SHR4**2.0*BF4/SQRT(BF4*BF4+(X4+HEIGHT)**2.0)
   ELSE
      SHR4 = 0.0
      LTA4 = 0.0
   END IF

   LTA = ((2.0*LTA1+LTA3+LTA4)/2.0)*0.7* SQRT((HDR)/(DB/DE-1))

   *** Fountain Core.
   KCA = 0.12*NJETS*((DB/DE)*(WB/L)*E**0.25)*
        DE/SQRT(SC)
   LCA = 2.5
   LTCA1 = KCA*(SHR1)**LCA*COS(THA1)
$\text{LTCA}_3 = \text{KCA} \times (\text{SHR}_3)^2 \times \text{LCA} \times \cos (\text{THA}_3) \\
\text{LTCA}_4 = \text{KCA} \times (\text{SHR}_4)^2 \times \text{LCA} \times \cos (\text{THA}_4) \\
\text{LTCA} = (2.0 \times \text{LTCA}_1 + \text{LTCA}_3 + \text{LTCA}_4) \tag{1}
$  

$\text{KCB} = 0.31 \times \text{NJETS} \times (\text{DB}/\text{DE})^{0.35} \times (\text{WB}/\text{L})^{0.65} \times (\text{SCP}/\text{SC})^{0.5} \times ((\text{E} \times \text{DE}) / \sqrt{\text{SC}})^{1.8} \tag{2}
$  

$\text{LCB} = \text{NJETS} \times \text{E} \times \text{DE} / \sqrt{\text{SC}} \tag{3}
$  

$\text{LTCB}_1 = \text{KCB} \times (\text{SHR}_1)^2 \times \text{LCB} \times \cos (\text{THA}_1) \\
\text{LTCB}_3 = \text{KCB} \times (\text{SHR}_3)^2 \times \text{LCB} \times \cos (\text{THA}_3) \\
\text{LTCB}_4 = \text{KCB} \times (\text{SHR}_4)^2 \times \text{LCB} \times \cos (\text{THA}_4) \\
\text{LTCB} = (2.0 \times \text{LTCB}_1 + \text{LTCB}_3 + \text{LTCB}_4) \tag{4}
$  

$\text{IF} (\text{LTCA} \gt \text{LTCB}) \text{ THEN} \\
\quad \text{LTC} = \text{LTCA} \\
\text{ELSE} \\
\quad \text{LTC} = \text{LTCB} \\
\text{ENDIF} \tag{5}
$  

$\text{LTF} = \text{LTA} + \text{LTC} \tag{6}
$  

$\text{HP} = 1.0 \tag{7}
$  

$\text{ELSE IF} (\text{NJETS} \gt 1) \text{ THEN} \\
\quad *** \text{ method for fountain effects.} \\
\text{IF} (\text{NJETS} \text{ EQ. 2}) \text{ THEN} \\
\quad *** \text{ 2 jet configuration.} \\
\quad *** \text{ Use a pressure ratio (PR) of 2.0 if PR is greater than two.} \\
\quad \text{IF} (\text{PR} \text{ GE. 2.0}) \text{ THEN} \\
\quad \quad \text{HP} = 3.6 \times (\text{HW}/\text{DI})^{0.62} \times \sqrt{2.0} \times \text{DE} \\
\text{ELSE} \\
\quad \quad \text{HP} = 3.6 \times (\text{HW}/\text{DI})^{0.62} \times \sqrt{\text{PR}} \times \text{DE} \\
\text{ENDIF} \tag{8}
$  

$\text{For 2 jet side-by-side use 2*E instead of BF1 since method was developed for longitudinal jets not lateral jets.} \\
\text{IF} (\text{IF EQ. 0.0} \text{ AND X1 EQ. 0.0}) \text{ THEN} \\
\quad \text{KP} = 0.084 \times (\text{EAV}/\text{DI})^{0.39} \times ((\text{BF1}/\text{DI}) \times (\text{SFA1}/\text{SFA1P}))^{1.1} \tag{9}
$  

$\text{ELSE} \\
\quad \text{KP} = 0.084 \times (\text{EAV}/\text{DI})^{0.39} \times ((2.0 \times \text{X1}/\text{DI}) \times (\text{SFA1}/\text{SFA1P}))^{1.1} \tag{10}
$  

$\text{ENDIF} \tag{11}
$  

$\text{LP} = -1.35 \times (\text{HW}/\text{X1}) \tag{12}
$  

$\text{ELSE} \\
\quad *** \text{ more than 2 jet configuration.} \\
\quad *** \text{ Use a pressure ratio (PR) of 2.0 if PR is greater than two.} \\
\quad \text{IF} (\text{PR} \text{ GE. 2.0}) \text{ THEN} \\
\quad \quad \text{HP} = 2.0 \times \sqrt{\text{EAV}/\text{DI}} \times \sqrt{2.0} \times \text{DE} \\
\text{ELSE} \\
\quad \quad \text{HP} = 2.0 \times \sqrt{\text{EAV}/\text{DI}} \times \sqrt{\text{PR}} \times \text{DE} \\
\text{ENDIF} \tag{13}
$  

$\text{ENDIF} \tag{14}$
LP = -2.4*(DB*WB/(DE*L))**.4/(EAV/DI*SQRT(E))
ENDIF

SLOPEL = 0.033*(DB/DE*WB/L)
LTFPU = KP*(HDR)**LP
LTFPL = SLOPEL/(HDR)

IF ((HDR) .LT. HP/DE .OR. HFLAG .LT. 3) THEN

IF (HFLAG .EQ. 0) THEN
*** Use 1st point from upper curve.
LTFP = LTFPU
HFLAG = 1
ELSE IF (HFLAG .LT. 3) THEN
*** HFLAG = 1: the 2nd point tried.
*** HFLAG = 2: at least the 3rd point tried.
SLOPEP = SLOPE
SLOPE = (YP-LTFPU)/(XP-(HDR))
IF (SLOPE .GE. 0.0) THEN
*** Use the lower point and give possible error
*** message.
HFLAG = 4
IERROR = 1
ELSE
*** The x-intercept (h/d) point.
XINTP = XINTER
XINTER = (HDR) - 1.0/SLOPE*LTFPU
ENDIF
ELSE
*** The x-intercept (h/d) point.
XINTP = XINTER
XINTER = (HDR) - 1.0/SLOPE*LTFPU
IF (XINTER .GT. HP/DE) THEN
ENDIF
ELSE IF (XINTER .LT. HP/DE) THEN
*** Use upper line.
ELSE IF (XINTER .EQ. HP/DE) THEN
SLOPET = SLOPE
HFLAG = 3
XINTER = HP/DE
ELSE
*** Draw line from point to HP/DE and use
*** if intersection < HP.
SLOPET = (LTFP-0.0)/(HP/DE-XP)
XINTER = HP/DE
HFLAG = 3
IERROR = 2
ENDIF
ELSE IF (XINTER .EQ. HP/DE) THEN
*** Set up the transition line.
SLOPET = SLOPE
HFLAG = 3
ELSE IF (XINTER .LT. HP/DE) THEN
*** Use upper line.
LTFP = LTFPU
ENDIF
C
ENDIF
C
IF (HFLAG .EQ. 1) HFLAG = 2
ENDIF
C
YP = LTFPU
XP = (HDR)
ENDIF
C
IF (HFLAG .EQ. 3 .AND. (HDR) .LT. HP/DE) THEN
*** Use the Transition Tangent line.

IF (ITRAN .EQ. 0) THEN

YUL = LTFPT

IF ((HP/DE*SLOPET)**2.0 .LT. ABS(4.0*SLOPEL*SLOPET)) THEN
*** Tangent line DOES NOT intersect the lower curve.
LTFP = LTFPL
HFLAG = 4
IERROR = 3
ELSE
*** Find the intersection of the tangent line and lower curve.
YTRAN = (- (HP/DE)*SLOPET - SQRT((HP/DE*SLOPET)**2.0 + 4.0*SLOPEL*SLOPET))/2.0
ITRAN = 1
ENDIF
ENDIF
C
LTFP = SLOPET * ((HDR) - XINTER)
C
IF (LTFP .LE. YTRAN) THEN
*** Past the intersection of the transition line and the lower curve--now use lower curve.
LTFP = LTFPL
HFLAG = 4
ENDIF
C
ELSE IF ((HDR).GE. HP/DE .OR. HFLAG .EQ. 4) THEN
*** Use lower curve.
LTFP = LTFPL
HFLAG = 4
ENDIF
C
ELSE
LTF = LTFP
ELSE
For the single jet case
LTF = 0.0
END IF
C
*** LID's
C
C************************************
C *** LID'S
IF (NJETS .GT. 2) THEN
  IF (SL .GT. 0.0 .AND. SCP .GT. 0.0) THEN
    KLA = 1.25*PP*(SL/SCP)*(DE/DB)**0.44*SQRT(I/E)
    KLB = 0.22*(HEIGHT/SQRT(SL))*E**2.0*(SC/SCP)
  ENDIF
  IF (KLA .LT. KLB) THEN
    KL = KLA
  ELSE
    KL = KLB
  ENDIF
  LTL = LTF*KL
ENDIF

CONTINUE

*** Output ***
LTF = LTF*KRB
LTN = LTOG+LTS+LTF+LTL
MTD = LTS*XCA/DE

Assign lift losses calculated to the appropriate lift loss result variable
H LTOG = LTOG
H LTS(H) = LTS
H LTF(H) = LTF
H LTL(H) = LTL
H LT(H) = LTN
H HPRI = HP

RETURN
END

SUBROUTINE SFCALL

ACRONYM: StolSF CALLing routine.

PURPOSE: The purpose of SF CALL is to set up all correct variables, based on the configuration, that are to be sent to the STOLSF routine. Another reason for this subroutine is all the subroutines that need to be called have variables with the same name which need to be kept separate when using common blocks to pass the variables back and forth between routines.

LOCAL VARIABLES (in addition to the above parameters):

NAME TYPE I/O UNITS DESCRIPTION
--- ---- --- ---------- --------------------------------- 
(none)

GLOBAL VARIABLES (in addition to the above parameters and local vars):

FIGPIE Common Block

NAME TYPE I/O UNITS DESCRIPTION
--- ---- --- ---------- --------------------------------- 
DB_B R I Ft Dbar of Body
DB_WB R I Ft Dbar of Wing-Body
**PIESF Common Block**

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AJ</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
</tr>
<tr>
<td>DC</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>DF</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>DE</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>DH</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>HEIGHT</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>HLTF</td>
<td>R</td>
<td>I</td>
<td>----</td>
</tr>
</tbody>
</table>

**PIEFLAG Common Block**

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLGRCS</td>
<td>L</td>
<td>O</td>
<td>----</td>
</tr>
<tr>
<td>HDEOUT</td>
<td>L</td>
<td>I</td>
<td>----</td>
</tr>
<tr>
<td>PRTFLG</td>
<td>L</td>
<td>I</td>
<td>----</td>
</tr>
<tr>
<td>RCSFLG</td>
<td>L</td>
<td>O</td>
<td>----</td>
</tr>
<tr>
<td>TYPE_F</td>
<td>C*4</td>
<td>O</td>
<td>----</td>
</tr>
<tr>
<td>TYPE_R</td>
<td>C*4</td>
<td>O</td>
<td>----</td>
</tr>
<tr>
<td>VEOUT</td>
<td>L</td>
<td>I</td>
<td>----</td>
</tr>
<tr>
<td>WBFLAG</td>
<td>L</td>
<td>O</td>
<td>----</td>
</tr>
</tbody>
</table>

**DESCRIPTION**

- **Effective Diameter of Front jets**: Effective Diameter of Front & Rear nozzles
- **Boundary layer factor**: Total Number of jets
- **Total Number of jets**: Number of Front jets
- **Number of Rear jets**: Jet Pressure Ratio for all jets
- **Dynamic pressure for Front and Rear nozzles**: Total jet exit area
- **Width to Length ratio of Front jets**: Width to Length ratio of Rear jets
- **Distance between Front & Rear jets**: Distance between Front jets
- **Height of wing above nozzle**: height of wing above nozzle

- **FLGRCS**: Flag which signals if there is an RCS on the configuration (TRUE - RCS, FALSE - No RCS)
- **HDEOUT**: Signals when to output tables based on height or height/De
  - TRUE - Print based on Height/DE,
  - FALSE - Print based on height
- **PRTFLG**: Signals when to output to screen.
- **RCSFLG**: Flag which identifies if RCS lift loss has calculation exceeded the area ratio (Swing / Ajet > 7000)
  - TRUE - Exceeded
  - FALSE - Not Exceeded
- **TYPE_F**: Discription of type of front nozzle (CIRCular, OVAL, RECTangular)
- **TYPE_R**: Same as TYPE_F but for rear nozzles
- **VEOUT**: Signals when to output tables based on VE
  - TRUE - Print Based on VE
  - FALSE - Print Based on Velocity
- **WBFLAG**: Identifies when WingBody planform has been enter by the user. Used to determine when to use wingbody in calculations.

- **AJ**: Area of front and rear nozzles. _Dbar_ of the configuration used
  (Based on the wing height).
- **DC**: Diameter of individual front jet.
- **DF**: Equivalent diameter of all jets.
- **DE**: Wing height above nozzles.
- **HEIGHT**: Altitude at which calculations are to be made.
- **HLTF**: Hover lift gain due to fountain effects.
Lift gain due to the addition of LIDs while in hover.

H' calculated from HOVPIE

Boundary layer factor

Distance between front and rear nozzles.

Hover lift loss due to suckdown.

Total number of nozzles.

Total number of front nozzles.

Pressure ratio of nozzles.

Average dynamic pressure of nozzles

Average width to length of front and rear nozzles

Distance between front nozzles.
in STOL flight

C NOTES: None.
C REFERENCES:
C 1) None.
C ENVIRONMENT:
C FORTRAN 77, VAX 6800, VAX 11/785, SGI IRIS 4D series.
C NON-STANDARD CODE:
C ?
C AUTHOR(S):
C Kipp E. Howard (KEH), Cal Poly San Luis Obispo, NASA Ames Moffett
C REVISION HISTORY:
C DATE INITIALS & DESCRIPTION
C 05/01/90 KEH -- Completed initial coding and debugging.
C 07/13/91 KEH -- Eliminated extraneous spacing and fixed comments

#include "figpie.inc"
#include "pieflag.inc"
#include "piesf.inc"
#include "respie.inc"

C Assign local STOLSF variables to their respective global configuration
C variables.
    HEIGHT = HT(H)
    PR = PR_FR
    DE = DE_FR
    AJ = S_FR
    DH = Z_W
    N = NUM
    NF = NUM_F
    YF = Y_F
    LP = X_FR

    IF (NUM_R .NE. 0) THEN
        WL = (WL_F + WL_R) / 2.0
    ELSE
        WL = WL_F
    END IF

    KBL = KB
    DF = D_F
    Q = Q_FR
    HPRI = H_HPRI
    LTS = H_LTS(H)
    HLTF = H_LTF(H)
    HLTL = H_LTL(H)

    IF (Z_W .NE. 0.0) THEN
        DC = DB_GB
    ELSE
        DC = DB_B
    END IF

CALL STOLSF
RETURN
END
SUBROUTINE STOLSF
ACRONYM: STOL operation - estimation of Suckdown and Fountain effects.

PURPOSE: The purpose of STOLSF is to calculate the suckdown and fountain effects while in the STOL configuration.

LOCAL VARIABLES (in addition to above parameters):

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>HD</td>
<td>R</td>
<td>-</td>
<td>-----</td>
<td>Aircraft height/front nozzle diam.</td>
</tr>
<tr>
<td>HF</td>
<td>R</td>
<td>-</td>
<td>Ft</td>
<td>Height where fountain effects are diminish to zero.</td>
</tr>
<tr>
<td>KH</td>
<td>R</td>
<td>-</td>
<td>-----</td>
<td>Wall jet placement factor</td>
</tr>
<tr>
<td>KX</td>
<td>R</td>
<td>-</td>
<td>-----</td>
<td>Longitudinal adjustment factor</td>
</tr>
<tr>
<td>KY</td>
<td>R</td>
<td>-</td>
<td>-----</td>
<td>Lateral adjustment factor</td>
</tr>
<tr>
<td>LTF</td>
<td>R</td>
<td>-</td>
<td>-----</td>
<td>Storage variable for fountain lift increment</td>
</tr>
<tr>
<td>LTN</td>
<td>R</td>
<td>-</td>
<td>-----</td>
<td>Storage variable for net lift increment</td>
</tr>
<tr>
<td>LTSS</td>
<td>R</td>
<td>-</td>
<td>-----</td>
<td>Storage variable for suckdown lift increment</td>
</tr>
<tr>
<td>VE</td>
<td>R</td>
<td>-</td>
<td>-----</td>
<td>Ratio of freestream dynamic pressure to jet dynamic pressure.</td>
</tr>
<tr>
<td>XD</td>
<td>R</td>
<td>-</td>
<td>-----</td>
<td>Forward extent of wall jet in units of jet diameters</td>
</tr>
</tbody>
</table>

GLOBAL VARIABLES (in addition to local variables):

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>AJ</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
<td>Area of front and rear nozzles.</td>
</tr>
<tr>
<td>DC</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Dbar of the configuration used (Based on the wing height).</td>
</tr>
<tr>
<td>DF</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Diameter of individual front jet.</td>
</tr>
<tr>
<td>DE</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Equivalent diameter of all jets.</td>
</tr>
<tr>
<td>DH</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Wing height above nozzles</td>
</tr>
<tr>
<td>HEIGHT</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Altitude at which calculations are to be made.</td>
</tr>
<tr>
<td>HLTF</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Hover lift gain due to fountain effects</td>
</tr>
<tr>
<td>HLTL</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Lift gain due to the addition of LIDs while in hover.</td>
</tr>
<tr>
<td>HPRIME</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>H' calculated from HOVPIE</td>
</tr>
<tr>
<td>KEL</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Boundry layer factor</td>
</tr>
<tr>
<td>LP</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Distance between front and rear nozzles.</td>
</tr>
<tr>
<td>LTS</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Hover lift loss due to suckdown.</td>
</tr>
<tr>
<td>N</td>
<td>I</td>
<td>I</td>
<td>-----</td>
<td>Total number of nozzles.</td>
</tr>
<tr>
<td>NF</td>
<td>I</td>
<td>I</td>
<td>-----</td>
<td>Total number of front nozzles.</td>
</tr>
<tr>
<td>PR</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Pressure ratio of nozzles.</td>
</tr>
<tr>
<td>Q</td>
<td>R</td>
<td>I</td>
<td>1b/Sq Ft</td>
<td>Average dynamic pressure of nozzles</td>
</tr>
<tr>
<td>WL</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Average width to length of front and rear nozzles</td>
</tr>
<tr>
<td>YF</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Distance between front nozzles.</td>
</tr>
</tbody>
</table>

RESPIE Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>I</td>
<td>O</td>
<td>-----</td>
<td>Nozzle deflection angle number</td>
</tr>
</tbody>
</table>
Suckdown Calculations

#include "piesf.inc"
#include "respie.inc"

C Local variable declarations
REAL VE, LTSS, KH, KY, KX, HD, XD, HF, LTF, LTN

VE=SQRT((0.00119* (I.69*VO(V))**2.0)/Q)

Suckdown Calculations
C

HD=HEIGHT/DF

IF (VE .EQ. 0.) THEN
    KH = 1.
ELSE IF (NF .EQ. 2) THEN

(AFVAL-TR-87-3019 Volume II, 14-6)
Make calculations for jets that are side-by-side.

IF ((YF/DF) .GE. 4.25 .OR. NF .EQ. 1) THEN
    KY = 1.0
ELSE
    KY = 1. + 6.2 * VE * (1. - (YF/DF)/4.25)
END IF

Calculate forward extent of jet flow.

XD=HD*TAN((DFL(D)-90.)/57.296)+(KBL* (.75/VE)-1.75*WL**.3*
VE**2.25*(1.-SIN((DFL(D)-90.)/57.296))*HD**2.5)*
(DFL(D)/90.)**2.*KY

KH = (XD + (LP/2.)/DF) * DF/DC + .5

IF (KH .GE. 1.) THEN
    KH = 1.0
ELSE IF (KH .LT. 0.0) THEN
    KH = 0.0
END IF

ELSE IF (NF .EQ. 1) THEN

(AFVAL-TR-87-3019 Volume II, 14-5)
Make calculations for tandem jets.

IF ((LP/2.)/DF .LT. 2.12) THEN
    KX = 1. + .64 * (1. - (LP/2./DF)/2.12)
ELSE IF ((LP/2.)/DF .GE. 2.12 .OR.
    (LP/2.)/DF .EQ. 0.) THEN
    KX = 1.
END IF

Calculate forward extent of jet wake

XD=HD*TAN((DFL(D)-90.)/57.296)+(KBL* (.75/VE)-1.75*WL**.3*
VE**2.25*(1.-SIN((DFL(D)-90.)/57.296))*HD/KX)**2.5)*
(DFL(D)/90.)**2.

KH = (XD + (LP/2.)/DF) * DF/DC + .5

IF (KH.GE.1.) THEN
    KH=1.0
ELSE IF (KH .LT. 0.0) THEN
    KH=0.0
END IF

END IF

LTSS=LTS*KH*(SIN(DFL(D)/57.296))**2

Fountain lift calculations

IF (N.LT.2)THEN
    LTF=0.
ELSE
IF (VE.EQ.0.) THEN
  HF=HPRIME
ELSE
  HF=DE*2.5/SQRT(VE)
ENDIF

IF (HEIGHT.GT.HF .AND. VE .NE. 0.0) THEN
  LTF=0.
ELSE IF(HF.GT.HPRIME) THEN
  HF=HPRIME
  LTF=(HLTF+HLTL)*(HF/HPRIME)**0.5*KH
  *(SIN(DFL(D)/57.296))**2.0
ELSE
  LTF=(HLTF+HLTL)*(HF/HPRIME)**0.5*KH
  *(SIN(DFL(D)/57.296))**2.0
ENDIF

IF(LTF.LT.0.) LTF=0.0
END IF

Out-of-Ground-Effect Term
The following has been commented out because of the
recommendation of Richard Kuhn.

This section must be included because the OGE lift loss
does vary with velocity and deflection angle.

HLTOGE(D) = H_LTOG * (SIN(DFL(D) / 57.296))**2

HLTOGE(D) = H_LTOG

OUTPUT
LTN = LTSS + LTF
HDR=HEIGHT/DE
Assign lift losses calculated to the appropriate lift loss
result variable
REC3 = LH * LV * (D - 1) + LH * (V - 1) + H
WRITE (70,100,REC=REC3) LTSS
WRITE (71,100,REC=REC3) LTF

RETURN
Format statements
100 FORMAT(F11.5)

SUBROUTINE RCSCAL

ACRONYM: Reaction Control System CALL routine (power induced effects)
PURPOSE: RCSCAL sets up all the correct variables, based on the
configuration, that are sent to the RCSCAL routine.

LOCAL VARIABLES (in addition to the above parameters):
NAME  TYPE  I/O  UNITS  DESCRIPTION
---  ----  --  ----  "
none

GLOBAL VARIABLES (in addition to the above parameters and local vars):
### FIGPIE Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_W</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Width of Wing (Wing span)</td>
</tr>
<tr>
<td>D_RCS</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Diameter of Roll RCS nozzle</td>
</tr>
<tr>
<td>PR_RCS</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td>Roll RCS Pressure Ratio</td>
</tr>
<tr>
<td>Q_RCS</td>
<td>R</td>
<td>I</td>
<td>lb/sq ft</td>
<td>Dynamic pressure for roll RCS jets</td>
</tr>
<tr>
<td>S_W</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
<td>Area of Wing</td>
</tr>
<tr>
<td>T_RCS</td>
<td>R</td>
<td>I</td>
<td>lb</td>
<td>Thrust of roll RCS nozzle</td>
</tr>
<tr>
<td>X_RCS</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Distance of roll RCS nozzle ahead of wing trailing edge</td>
</tr>
<tr>
<td>Y_RCS</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Distance of roll RCS nozzle in from wingtip</td>
</tr>
</tbody>
</table>

### PIEFLAGS Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLGRC</td>
<td>L</td>
<td>O</td>
<td>Flag which signals if there is an RCS on the configuration</td>
</tr>
</tbody>
</table>

### PIERCS Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>R</td>
<td>I</td>
<td>feet</td>
<td>Wing span</td>
</tr>
<tr>
<td>DRCS</td>
<td>R</td>
<td>I</td>
<td>feet</td>
<td>Roll RCS jet Diameter.</td>
</tr>
<tr>
<td>NPR</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td>Roll jet nozzle pressure ratio.</td>
</tr>
<tr>
<td>Q</td>
<td>R</td>
<td>I</td>
<td>lb/sq ft</td>
<td>Roll jet dynamic pressure.</td>
</tr>
<tr>
<td>S</td>
<td>R</td>
<td>I</td>
<td>sq ft</td>
<td>Wing area</td>
</tr>
<tr>
<td>T</td>
<td>R</td>
<td>I</td>
<td>lb</td>
<td>RCS roll jet thrust</td>
</tr>
<tr>
<td>X</td>
<td>R</td>
<td>I</td>
<td>feet</td>
<td>Jet distance ahead of wing trailing edge.</td>
</tr>
<tr>
<td>Y</td>
<td>R</td>
<td>I</td>
<td>feet</td>
<td>Jet distance in from wing tip.</td>
</tr>
</tbody>
</table>

### RESPIE Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCS_LT(1,500)</td>
<td>R</td>
<td>O</td>
<td>----</td>
<td>Total lift loss from the Reaction Control System.</td>
</tr>
</tbody>
</table>

### Files Used:

<table>
<thead>
<tr>
<th>LOGICAL UNIT</th>
<th>I/O</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>&lt;list of files used in the routine&gt;</td>
</tr>
</tbody>
</table>

### Commons Used:

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIGPIE</td>
<td>configuration for Power Induced Effects - Contains all variables needed to define the configuration and other parameters of the aircraft.</td>
</tr>
<tr>
<td>PIEFLAGS</td>
<td>Power Induced Effects FLAGS - Contains variables which help keep track of the configuration of the aircraft.</td>
</tr>
<tr>
<td>PIERCS</td>
<td>Power Induced Effects for the Reaction Control System - Contains variables which are sent to the RCSIND subroutine.</td>
</tr>
<tr>
<td>RESPIE</td>
<td>RESULTS from all Power Induced Effects subroutines - Contains results from each subroutine along with</td>
</tr>
</tbody>
</table>
variables for height, velocity, jet deflection angle, and angle of attack and their counters.

---

ROUTINES CALLED:

NAME       DESCRIPTION
----------- -----------------------------
RCSIND     Calculates RCS lift increments while in forward flight

NOTES: None.

REFERENCES:
1) None.

ENVIRONMENT:
FORTRAN 77, VAX 6800, VAX 11/785, SGI IRIS 4D series.

NON-STANDARD CODE:

AUTHOR(S):
Kipp E. Howard (KEH), Cal Poly San Luis Obispo, NASA Ames Moffett

REVISION HISTORY:
05/15/90 KEH -- Completed initial coding and debugging.
07/14/91 KEH -- Eliminated extraneous spacing and fixed comments

#include "figpie.inc"
#include "pieflag.inc"
#include "piercs.inc"
#include "respie.inc"

C Assign local RCSIND variables to their respective global configuration
C variables.

IF (FLGRCS) THEN
B = BW
DRCS = D_RCS
NFR = PR_RCS
Q = Q_RCS
S = SW
T = T_RCS
X = X_RCS
Y = Y_RCS
CALL RCSIND
ELSE
Assign the zero to RCS results.
RCS_LT(1,V) = 0.0
ENDIF

RETURN
END

SUBROUTINE RCSIND

ACRONYM: RCS jet INDuced effects.

PURPOSE: Calculation of jet induced lift loss for powered lift
aircraft due to jet effects in ground proximity.
LOCAL VARIABLES (in addition to the above parameters):

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>KB</td>
<td>R</td>
<td>----</td>
<td>-----</td>
<td>Chordwise position factor.</td>
</tr>
<tr>
<td>KC</td>
<td>R</td>
<td>----</td>
<td>-----</td>
<td>Spanwise position factor.</td>
</tr>
<tr>
<td>LTO</td>
<td>R</td>
<td>----</td>
<td>-----</td>
<td>Lift increment for subcritical pressure ratio (does not vary with pressure ratio)</td>
</tr>
<tr>
<td>LTP</td>
<td>R</td>
<td>----</td>
<td>-----</td>
<td>Lift increment due to pressure ratio</td>
</tr>
<tr>
<td>LVOLO</td>
<td>R</td>
<td>----</td>
<td>-----</td>
<td>Lift increment for RCS</td>
</tr>
<tr>
<td>MO</td>
<td>R</td>
<td>lb ft</td>
<td></td>
<td>Rolling moment without losses</td>
</tr>
<tr>
<td>MV</td>
<td>R</td>
<td>lb ft</td>
<td></td>
<td>Rolling moment with losses</td>
</tr>
<tr>
<td>PI</td>
<td>R</td>
<td>rad</td>
<td></td>
<td>3.1415927</td>
</tr>
<tr>
<td>VE</td>
<td>R</td>
<td>----</td>
<td>-----</td>
<td>Effective velocity ratio</td>
</tr>
</tbody>
</table>

GLOBAL VARIABLES (in addition to the above parameters and local vars):

PIEFLAG Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCSFLG</td>
<td>L</td>
<td>O</td>
<td>----</td>
<td>Flag which identifies if RCS lift loss has calculation exceeded the area ratio (Swing / Ajet &gt; 7000)</td>
</tr>
<tr>
<td>TRUE</td>
<td>Exceeded</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FALSE</td>
<td>Not Exceeded</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PIERCS Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>R</td>
<td>I</td>
<td>feet</td>
<td>Wing span</td>
</tr>
<tr>
<td>DRCS</td>
<td>R</td>
<td>I</td>
<td>feet</td>
<td>Roll RCS jet Diameter.</td>
</tr>
<tr>
<td>NPR</td>
<td>R</td>
<td>I</td>
<td></td>
<td>Roll jet nozzle pressure ratio.</td>
</tr>
<tr>
<td>Q</td>
<td>R</td>
<td>I</td>
<td>lb/sq ft</td>
<td>Roll jet dynamic pressure.</td>
</tr>
<tr>
<td>S</td>
<td>R</td>
<td>I</td>
<td>sq ft</td>
<td>Wing area</td>
</tr>
<tr>
<td>SAJ</td>
<td>R</td>
<td>----</td>
<td>----</td>
<td>Wing area / jet area</td>
</tr>
<tr>
<td>T</td>
<td>R</td>
<td>I</td>
<td>lb</td>
<td>RCS roll jet thrust</td>
</tr>
<tr>
<td>X</td>
<td>R</td>
<td>I</td>
<td>feet</td>
<td>Jet distance ahead of wing trailing edge.</td>
</tr>
<tr>
<td>Y</td>
<td>R</td>
<td>I</td>
<td>feet</td>
<td>Jet distance in from wing tip.</td>
</tr>
</tbody>
</table>

RESPIE Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCS_LT(1,500)</td>
<td>R</td>
<td>O</td>
<td>----</td>
<td>Total lift loss from the Reaction Control System.</td>
</tr>
<tr>
<td>V</td>
<td>I</td>
<td>O</td>
<td></td>
<td>Velocity number counter</td>
</tr>
<tr>
<td>VO(500)</td>
<td>R</td>
<td>O</td>
<td>----</td>
<td>Actual velocity value</td>
</tr>
</tbody>
</table>

FILES USED:

<table>
<thead>
<tr>
<th>LOGICAL UNIT</th>
<th>I/O</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>None.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

COMMONS USED:

PIEFLAGS

Power Induced Effects FLAGS - Contains variables which help keep track of the configuration of the aircraft.

PIERCS

Power Induced Effects for the Reaction Control System - Contains variables which are sent to the RCSIND.
subroutine.
RESusts from all POwer Induced Effects subroutines -
Contains results from each subroutine along with
variables for height, velocity, jet deflection angle,
and angle of attack and their counters.

CALLED BY:
NAME    DESCRIPTION
--------- ----------------------------------
RCSCAL   Isolates and controls execution of RCSIND

ROUTINES CALLED:
NAME    DESCRIPTION
--------- ----------------------------------
None.

NOTES: None.
REFERENCES:
1) None.
ENVIRONMENT:
FORTRAN 77, VAX 6800, VAX 11/785, SGI IRIS 4D series.
NON-STANDARD CODE:
AUTHOR(S):
Douglas A. Wardwell (DAW), NASA Ames Research Center
Kipp E. Howard (KEH), Cal Poly San Luis Obispo, NASA Ames Res Ctr

REVISON HISTORY:
DATE        INITIALS & DESCRIPTION
02/22/90    DAW -- FORTRAN version of Dick Kuhn's basic program.
02/26/90    KEH -- Finished conversion of BASIC program to FORTRAN.
07/14/91    KEH -- Eliminated extraneous spacing and fixed comments

#include "piercs.inc"
#include "respie.inc"
#include "pieflag.inc"

*** Local variables.
REAL PI, KB, KC, MV, MO, LTO, LTP, VE, LVOLO

PI = 3.1415926
MO = T*(B/2.0-Y)
VE = SQRT((.00119*(1.69*VO(V))**2.0)/Q)
SAJ = (S/2.0)/((PI/4.0)*DRCS**2.0)

C Calculate chordwise position factor
if (Y/DRCS .gt. 10) then
KB = 1.0
else
KB = .25+.2*(Y/DRCS)**.58
end if

C Calculate spanwise position factor
if (X/DRCS .gt. 12.0) then
KC = 1.0
else
KC = .25+.06*(X/DRCS)
end if

C Limitations of this method
IF (VE .GT..1) THEN
LTO = 0.0
LTP = 0.0
go to 15
END IF

C IF (SAJ .GT. 7000) THEN
  RCSFLG = .TRUE.
ELSE
  RCSFLG = .FALSE.
END IF

C if (NPR .LT. 1.893) then
  LTP = 0.0
else
  LTP = -.017*VE*(SAJ**.42)*(NPR-1.893)**.75
end if

C LTO = ((3.0*VE**3.0)-2.4*VE**2.2)*SAJ**.688
  \( \frac{S}{.41*VE**2.2)*SAJ**.688} \)
LVOLO = (LTO+LTP)*KB*KC
MV = MO*LVOLO

C Assign the correct value to the overall results.
RCS_LT(I,V) = LVOLO

100 format (lx,' V, kts ', ' M/Mo ', ' M')
110 format (lx,3(f9.3,3x))

C 15 RETURN
END

---

Gvcall

SUBROUTINE GVCALL

C ACRONYM: stolGV CALLing routine for power induced effects

C PURPOSE: The purpose of GVCALL is to set up all correct variables,
           based on the configuration, that are to be sent to the
           STOLGV routine. Another reason for this subroutine is all
           subroutines that need to be called have variables with the
           same name which need to be kept separate when using common
           blocks to pass variables back and forth between routines.

C LOCAL VARIABLES (in addition to the above parameters):
C NAME   TYPE I/O UNITS DESCRIPTION
C

C GLOBAL VARIABLES (in addition to the above parameters and local vars):
C
C FIGPIE Common Block
C NAME   TYPE I/O UNITS DESCRIPTION
C B WB    R I Ft Width of Wing-Body
C D F     R I Ft Diameter of each Front jet
C DE F    R I Ft Effective Diameter of Front jets
C NUM F   I I ---- NUMBER of Front jets
C NUM R   I I ---- NUMBER of Rear jets
C Q FR    R I lb/sq ft Dynamic pressure for Front and Rear
C S B     R I Sq Ft Area of Body
C S CS    R I Sq Ft Area of Center Section
C S F R I Sq Ft
C S W R I Sq Ft
C S WB R I Sq Ft
C T T F R I ----
C W L B R I ----
C W L F R I ----
C X FR R I Ft
C Z B R I Ft
C Z W R I Ft
C PIEGV Common Block
--- --- I/O UNITS
AFACT R I ----
AJ R I Sq Ft
DE R I Ft
DF R I Ft
DH R I Ft
EX R I Ft
LTV R O ----
N I I ----
O R I ----
TFT R I ----
WL B R I ----
WL J R I ----
C PIEFLAG Common Block
--- --- I/O UNITS
WBFLAG L O ----
C RESPIE Common Block
--- --- I/O UNITS
A I O ----
D I O ----
H I O ----
LA R I ----
LD R I ----
LH R I ----
LV R I ----
REC4 I O ----
V I O ----
--- --- I/O
72 O Direct access file for storage of lift increment due to the ground vortex on the body

**DESCRIPTION**

- **C S F R I Sq Ft**: Area of Front jets
- **C S W R I Sq Ft**: Area of Wing
- **C S WB R I Sq Ft**: Area of Wing-Body
- **C T T F R I ----**: Front Thrust / total Thrust
- **C W L B R I ----**: Width to Length ratio of Body
- **C W L F R I ----**: Width to Length ratio of Front jets
- **C X FR R I Ft**: Distance between Front & Rear jets
- **C Z B R I Ft**: Height of body base above nozzle
- **C Z W R I Ft**: Height of wing above nozzle

**DESCRIPTION**

- **C PIEGV Common Block**
  - **AFACT R I ----**: Area ratio - based on whether the wing or body is being used
  - **AJ R I Sq Ft**: Area of the front jets
  - **DE R I Ft**: Equivalent Diameter of front jets
  - **DF R I Ft**: Diameter of front jets
  - **DH R I Ft**: Height of wing above jets
  - **EX R I Ft**: Distance from jet pattern center to center of front jets.
  - **LTV R O ----**: Lift loss due to ground vortex.
  - **N I I ----**: Number of front jets
  - **O R I ----**: Dynamic pressure of the nozzles
  - **TFT R I ----**: Front thrust split
  - **WL B R I ----**: Width to length ratio of the body.
  - **WL J R I ----**: Width to length ratio of the jets.

**DESCRIPTION**

- **C PIEFLAG Common Block**
  - **WBFLAG L O ----**: Identifies when WingBody planform has been entered by the user. Used to determine when to call MAXEWB subroutine.

**DESCRIPTION**

- **C RESPIE Common Block**
  - **A I O ----**: Angle of attack number counter
  - **D I O ----**: Nozzle deflection angle number counter.
  - **H I O ----**: Height number counter
  - **LA R I ----**: Total number of Angle of Attack values
  - **LD R I ----**: Total number of nozzle deflection angles.
  - **LH R I ----**: Total number of height values.
  - **LV R I ----**: Total number of velocity values.
  - **REC4 I O ----**: Record number for any file which replaces a 4 X 4 matrix
  - **V I O ----**: Velocity number counter

**DESCRIPTION**

- **FILES USED**
  - **LOGICAL UNIT I/O**: Direct access file for storage of lift increment due to the ground vortex on the body
Direct access file for storage of lift increment due to the ground vortex on the wing.

Commons Used:
- **FIGPIE**
  Configuration for Power Induced Effects - Contains all variables needed to define the configuration and other parameters of the aircraft.
- **PIEFLAGS**
  Power Induced Effects FLAGS - Contains variables which help keep track of the configuration of the aircraft.
- **PIEGV**
  Power Induced Effects for stolGV - Contains variables which are sent to the STOLGV subroutine.
- **RESPIE**
  RESULTS from all Power Induced Effects subroutines - Contains results from each subroutine along with variables for height, velocity, jet deflection angle, and angle of attack and their counters.

Called by:
- **COPPIE**
  Controls execution and coordination of Power Induced Effects Module.

Routines Called:
- **STOLGV**
  Calculates ground vortex lift increments while in STOL flight.

Notes: None.
References:
1) None.

Environment:
FORTRAN 77, VAX 6800, VAX 11/785, SGI IRIS 4D series.

Non-Standard Code:
?

Author(s):
Kipp E. Howard (KEH), Cal Poly San Luis Obispo, NASA Ames Moffett

Revision History:
- 05/14/90 KEH -- Completed initial coding and debugging.
- 07/13/91 KEH -- Eliminated extraneous spacing and fixed comments.

```
#include "figpie.inc"
#include "pieflag.inc"
#include "piegv.inc"
#include "respie.inc"

REC4 = LH * LV * LD * (A - 1) + LH * LV * (D - 1) + LH * (V - 1) + H

C Assign local STOLGV variables to their respective global configuration variables.
C Variables that do not change with each configuration (Wing, Body, CS)
TFT = TT_F
WLJ = WL_F
AJ = S_F
N = NUM_F
DE = DE_F
DF = D_F
Q = Q_FR
```
IF (NUM_F .EQ. 2 .AND. NUM_R .EQ. 0) THEN
  EX = 0.0
ELSE
  EX = X_FR / 2.0
END IF

IF (WBFLAG) THEN
  Variables that STOLGV needs while calculating for the Wingbody
  IF (B_WB**2 / S_WB .GT. 1.0) THEN
    WLB = 1.0
  ELSE
    WLB = WL_WB
  END IF

  DH = 0.0
  AFACT = (S_WB/S_F)**.62

  IF (VO(V) .EQ. 0.0) THEN
    LTV = 0.0
  ELSE
    CALL STOLGV
  END IF

  WRITE (72,100,REC=REC4) LTV
  WRITE (73,100,REC=REC4) 0.0
ELSE
  Variables that STOLGV needs while calculating for the Body.
  WLB = WL_B
  DH = Z_B
  AFACT = (S_B / S_F * WL_B)**.62

  IF (VO(V) .EQ. 0.0) THEN
    LTV = 0.0
  ELSE
    CALL STOLGV
  END IF

  WRITE (72,100,REC=REC4) LTV
  WRITE (73,100,REC=REC4) 0.0
END IF

Variables that STOLGV needs while calculating for the Wing
  WLB = 1.0
  DH = Z_W
  AFACT = ((S_W/S_F)**.62 - (S_CS/S_F)**.62)

  IF (VO(V) .EQ. 0.0) THEN
    LTV = 0.0
  ELSE
    CALL STOLGV
  END IF

  WRITE (73,100,REC=REC4) LTV
END IF

RETURN

Format statements
100 FORMAT (F11.5)
**STOLGV**

**ACRONYM:** STOL operation - estimation of Ground Vortex term.

**PURPOSE:** The purpose of STOLGV is to calculate the lift loss due to the ground vortex generated from the proximity of the ground.

**LOCAL VARIABLES** (in addition to the above parameters):

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP</td>
<td>R</td>
<td>-</td>
<td>-</td>
<td>Ground Vortex factor</td>
</tr>
<tr>
<td>HD</td>
<td>R</td>
<td>-</td>
<td>Ft</td>
<td>Height at which the rate of change of lift with height changes.</td>
</tr>
<tr>
<td>H2D</td>
<td>R</td>
<td>-</td>
<td>Ft</td>
<td>Maximum height at which the ground vortex effects are felt.</td>
</tr>
<tr>
<td>INC</td>
<td>R</td>
<td>I</td>
<td>Deg</td>
<td>Total angle nozzles are facing taking into account the jet deflection angle and the angle of attack.</td>
</tr>
<tr>
<td>KX</td>
<td>R</td>
<td>-</td>
<td>-</td>
<td>Factor used in calculating H2D.</td>
</tr>
<tr>
<td>VE</td>
<td>R</td>
<td>-</td>
<td>-</td>
<td>Ratio of the dynamic pressure of the aircraft over the dynamic pressure of the nozzles.</td>
</tr>
</tbody>
</table>

**GLOBAL VARIABLES** (in addition to the above local variables):

<table>
<thead>
<tr>
<th>PIEGV Common Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
</tr>
<tr>
<td>AFACT</td>
</tr>
<tr>
<td>AJ</td>
</tr>
<tr>
<td>DE</td>
</tr>
<tr>
<td>DF</td>
</tr>
<tr>
<td>DH</td>
</tr>
<tr>
<td>EX</td>
</tr>
<tr>
<td>LTV</td>
</tr>
<tr>
<td>N</td>
</tr>
<tr>
<td>Q</td>
</tr>
<tr>
<td>TFT</td>
</tr>
<tr>
<td>WLB</td>
</tr>
<tr>
<td>WLJ</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RESPIE Common Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>D</td>
</tr>
<tr>
<td>H</td>
</tr>
<tr>
<td>LA</td>
</tr>
<tr>
<td>LD</td>
</tr>
<tr>
<td>LH</td>
</tr>
<tr>
<td>LV</td>
</tr>
</tbody>
</table>
| REC4                | I    | O   | ----  | Record number for any file which
C FILES USED:
LOGICAL UNIT I/O DESCRIPTION
------------------------ ------------
<None>

C COMMONS USED:
NAME DESCRIPTION
------------ ------------------------
PIEFLAGS Power Induced Effects FLGS - Contains variables which help keep track of the configuration of the aircraft.
PIEGV Power Induced Effects for stolGV - Contains variables which are sent to the STOLGV subroutine.
RESPIE RESULTS from all Power Induced Effects subroutines - Contains results from each subroutine along with variables for height, velocity, jet deflection angle, and angle of attack and their counters.

C CALLED BY:
NAME DESCRIPTION
---------- ------------------------
<None>

C NOTES: None.

C REFERENCES:

C ENVIRONMENT:
FORTRAN 77, VAX 6800, VAX 11/785, SGI IRIS 4D series.

C NON-STANDARD CODE:
?

C AUTHOR(S):
Kipp E. Howard (KEH), Cal Poly San Luis Obispo, NASA Ames Moffett

C REVISION HISTORY:
DATE INITIALS & DESCRIPTION
05/08/90 KEH -- Initial completion of code.
05/14/90 KEH -- Code in working order.
07/13/91 KEH -- Eliminated extraneous spacing and fixed comments

#include "piegv.inc"
#include "respie.inc"

C Variable declaration
REAL CP, H1D, H2D, INC, XX, VE

C Variable initialization
INC = DFL(D) + AOA(A)
VE=SQRT((0.00119*(I.69*VO(V))**2.)/Q)

IF (EX/DF .LT. 2.12) THEN
  KX = 1. + .64 * (1. - (EX/DF)/2.12)
ELSE IF (EX/DF .GE. 2.12) .OR. EX/DF .EQ. 0.) THEN

replaces a 4 X 4 matrix
Velocity number counter
KX = 1.
END IF

H1D=1.19*((1/VE)**.68)*((1.+SIN((INC-90.0)/57.296))**.53-DH/DF
H2D=(.14/VE**.4 + 2.0)*SQRT(INC/90.0)*WLB**.4 * WLJ**-.12 * KX
CP=-2.3/SQRT(VE) * (1.0 - HT(H)/(H2D*DF))
IF (CP .LT. -3.3) CP=-3.3
IF (CP .GT. 0.0) CP=0.0

Calculate LTV according to its relationship to H1 and H2.
IF (HT(H) .LT. H1D*DF) THEN
  LTV = TFT * 0.13 * WLJ**(-.2) * AFACT * VE**.8 * CP *
      *(1.0 + SIN((INC - 90.0)/57.296)) *
      ((HT(H) + DH)/DF)**-1.3
ELSE IF (HT(H) .GE. H1D*DF .AND. HT(H) .LE. H2D*DF) THEN
  LTV = TFT * 0.18 * WLJ**(-.2) * AFACT * VE**(-.5) * CP *
      *(1.0 + SIN((INC - 90.0)/57.296))*2.0 *
      ((HT(H) + DH)/DF)**-3.2
ELSE
  LTV = 0.0
END IF

RETURN
END

SUBROUTINE WKCALL

Wkcall

ACRONYM: stolWK CALLing routine for power induced effects

PURPOSE: The purpose of GVCALL is to set up all the correct
variables, based on the configuration, that are to be sent
to the STOLWK and JIEPIE routine.

LOCAL VARIABLES (in addition to the above parameters):

NAME TYPE I/O UNITS DESCRIPTION
DE R I ---- Effective diameter of the jets in question.
LT_WK R O ---- Total Lift loss of the config. due to jet wake.
LT_WKB R O ---- Lift loss of the body due to jet wake.
LT_WKW R O ---- Lift loss of the wing due to jet wake.
LT0 R - ---- Lift loss out of ground effect.
FER R - ---- Perimeter of jets in question.
PI R - ---- Constant - 3.1415926

GLOBAL VARIABLES (in addition to the above parameters and local vars):

FIGPIE Common Block

NAME TYPE I/O UNITS DESCRIPTION
B_B R I Ft Width of Body
B_CS R I Ft Width of Center Section
B_W R I Ft Width of Wing (Wing span)
B_WB R I Ft Width of Wing-Body
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>D_F</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Diameter of each Front jet</td>
</tr>
<tr>
<td>D_R</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Diameter of each Rear jet</td>
</tr>
<tr>
<td>DE_F</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Effective Diameter of Front jets</td>
</tr>
<tr>
<td>DE_R</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Effective Diameter of Rear jets</td>
</tr>
<tr>
<td>KLF</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Adjustment factor for flap extension when calculating the lift loss due to jet induced effects.</td>
</tr>
<tr>
<td>MAC</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Mean Aerodynamic Chord of wing</td>
</tr>
<tr>
<td>NUM_F</td>
<td>I</td>
<td>I</td>
<td>-----</td>
<td>NUMBER of Front jets</td>
</tr>
<tr>
<td>NUM_R</td>
<td>I</td>
<td>I</td>
<td>-----</td>
<td>NUMBER of Rear jets</td>
</tr>
<tr>
<td>PR_F</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Jet Pressure Ratio for front jets</td>
</tr>
<tr>
<td>PR_FR</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Jet Pressure Ratio for all jets</td>
</tr>
<tr>
<td>PR_R</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Jet Pressure Ratio for rear jets</td>
</tr>
<tr>
<td>Q_F</td>
<td>R</td>
<td>I</td>
<td>lb/sq ft</td>
<td>Dynamic pressure for the front jets</td>
</tr>
<tr>
<td>Q_R</td>
<td>R</td>
<td>I</td>
<td>lb/sq ft</td>
<td>Dynamic pressure for the rear jets</td>
</tr>
<tr>
<td>S_B</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
<td>Area of Body</td>
</tr>
<tr>
<td>S_CS</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
<td>Area of Center Section</td>
</tr>
<tr>
<td>S_F</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
<td>Area of Front jets</td>
</tr>
<tr>
<td>S_R</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
<td>Area of Rear jets</td>
</tr>
<tr>
<td>S_W</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
<td>Area of Wing</td>
</tr>
<tr>
<td>S_WB</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
<td>Area of Wing-Body</td>
</tr>
<tr>
<td>SF_FB</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
<td>Area ahead of Front jets using body planform</td>
</tr>
<tr>
<td>SF_RB</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
<td>Area ahead of Rear jets using the body planform</td>
</tr>
<tr>
<td>T_F</td>
<td>R</td>
<td>I</td>
<td>lb</td>
<td>Thrust of Front jets</td>
</tr>
<tr>
<td>T_R</td>
<td>R</td>
<td>I</td>
<td>lb</td>
<td>Thrust of Rear jets</td>
</tr>
<tr>
<td>WL_B</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Width to Length ratio of Body</td>
</tr>
<tr>
<td>WL_F</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Width to Length ratio of Front jets</td>
</tr>
<tr>
<td>WL_R</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Width to Length ratio of Rear jets</td>
</tr>
<tr>
<td>X_FR</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Distance between Front &amp; Rear jets</td>
</tr>
<tr>
<td>XNOZ_F</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>X-coordinates of Front NOZZle</td>
</tr>
<tr>
<td>XNOZ_R</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>X-coordinates of Rear NOZZle</td>
</tr>
<tr>
<td>XTE_F</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>X-coordinate of trailing edge for front nozzle (root TE if Nozzle within body)</td>
</tr>
<tr>
<td>XTE_R</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>X-coordinate of trailing edge for rear nozzle (root TE if Nozzle within body)</td>
</tr>
<tr>
<td>Y_F</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Distance between Front jets</td>
</tr>
<tr>
<td>Y_R</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Distance between Rear jets</td>
</tr>
<tr>
<td>YB_F</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Lateral distance from Body to Front jets (for external jets)</td>
</tr>
<tr>
<td>YB_R</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Lateral distance from Body to Rear jets (for external jets)</td>
</tr>
<tr>
<td>YWB_F</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Lateral distance from Wingbody to Front jets (for external jets)</td>
</tr>
<tr>
<td>YWB_R</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Lateral distance from wingbody to Rear jets (for external jets)</td>
</tr>
<tr>
<td>Z_B</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Height of body base above nozzle</td>
</tr>
<tr>
<td>Z_W</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Height of wing above nozzle</td>
</tr>
</tbody>
</table>

---

PIEFLAG Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>WBFLAG</td>
<td>L</td>
<td>O</td>
<td>-----</td>
<td>Identifies when the WingBody planform has been enter by the user. Used to determine when to</td>
</tr>
</tbody>
</table>
call MAKEWB subroutine.

### PIEWK Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFACCT</td>
<td>R</td>
<td>I</td>
<td>----</td>
</tr>
<tr>
<td>AJ</td>
<td>R</td>
<td>I</td>
<td>Sq. Ft</td>
</tr>
<tr>
<td>AR</td>
<td>R</td>
<td>I</td>
<td>----</td>
</tr>
<tr>
<td>B</td>
<td>R</td>
<td>I</td>
<td>----</td>
</tr>
<tr>
<td>CU</td>
<td>R</td>
<td>I</td>
<td>----</td>
</tr>
<tr>
<td>DH</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>DI</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>FIGWK</td>
<td>C</td>
<td>-</td>
<td>----</td>
</tr>
<tr>
<td>HEIGHT</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
</tr>
<tr>
<td>KL_F</td>
<td>R</td>
<td>I</td>
<td>----</td>
</tr>
</tbody>
</table>

### RESPIE Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>I</td>
<td>O</td>
<td>----</td>
</tr>
<tr>
<td>H</td>
<td>I</td>
<td>O</td>
<td>----</td>
</tr>
<tr>
<td>LD</td>
<td>R</td>
<td>I</td>
<td>----</td>
</tr>
<tr>
<td>LH</td>
<td>R</td>
<td>I</td>
<td>----</td>
</tr>
<tr>
<td>LV</td>
<td>R</td>
<td>I</td>
<td>----</td>
</tr>
</tbody>
</table>

### DESCRIPTION

Area ratios used in calculating the lift loss due to the jet wake.
Total area of the jets in question.
Aspect ratio of the planform.
Width of current body config.
Momentum coef. based on nozzle.
Vertical position of the jet based on the planform.
Diameter of the jets in question.
Flag which notifies JIEPIE which planform is in use.
Height of aircraft.
Adjustment factor used for flap extension

1.0 - Flaps extended
0.25 - Flaps not extended
Lift loss retured by STOLWK
Lift loss out of ground effect.
Out of ground lift loss
Mean Aerodynamic Chord.
Number of nozzles based on which nozzle used (Front, Rear)
Pressure Raito of local jets.
Jet dynamic pressure
Free stream dynamic pressure.
Area of the configuration.
Area of configuration that is in front of nozzles.
Thrust of the jets in question.
Flag which notifies STOLWK how close the nozzle is to the trailing edge of the wing.
Width to length ratio of nozzels.
Position of nozzle with respect to wing trailing edge (in % chord)
Lateral spacing of jets.
Distance of jet center to bodyside.
Effective velocity factor which accounts for the reduction in velocity at the rear nozzles due to the front nozzle.
Vertical distance of the nozzles.

Nozzle deflection angle number counter.
Height number counter
Total number of nozzle deflection angles.
Total number of height values.
Total number of velocity values.
Record number for any file which replaces a 3 X 3 matrix
Velocity number counter
Actual velocity value

<table>
<thead>
<tr>
<th>I/O</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>o</td>
<td>Direct access file for storage of lift increment due to the jet wake on the body</td>
</tr>
<tr>
<td>o</td>
<td>Direct access file for storage of lift increment due to the jet wake on the wing (without center section)</td>
</tr>
</tbody>
</table>

Configuration for Power Induced Effects - Contains all variables needed to define the configuration and other parameters of the aircraft.
Power Induced Effects Flags - Contains variables which help keep track of the configuration of the aircraft.
Power Induced Effects for STOLWK and JIEPIE - Contains all variables which are passed to the STOLWK and JIEPIE subroutines.
Results from all Power Induced Effects subroutines - Contains results from each subroutine along with variables for height, velocity, jet deflection angle, and angle of attack and their counters.

Controls execution and coordination of Power Induced Effects Module

Calculates change in lift caused by the jet induced effects on a flat plate
Calculates jet wake lift increments while in STOL flight

#include "figpie.inc"
#include "pieflag.inc"
#include "piewk.inc"
#include "respie.inc"

Declaration of local variables
REAL LT_WK, LT_WKB, LT_WKC, LT_WKW, PI
C Assign local STOLWK variables to their respective global configuration
C variables.
C Define variables that do not change with planform configuration
C and nozzle definition.
HEIGHT = HT(H)
MC = MAC
LT_WK = 0.0
LT_WKB = 0.0
LT_WKW = 0.0
KL_F = KLF
PI = 3.1415926
Q0 = 0.00119*(1.69*VO(V))**2.0

IF (WBFLAG) THEN
WINGBODY PLANFORM
Assign wingbody planform variables.
AR = B_WB**2.0 / S_WB
B = B_WB
S = S_WB
Z = 0.0
FIGWK = 'WBDY'
Assign variables for the front nozzles.
AJ = S_F
DE = DE_F
DI = D_F
N = NUM_F
PER = PI * 2.0 * SQRT(AJ / 3.14159)
PR = PR_F
Q = Q_F
SF = 0.0
T = T_F
WL = WL_F
Y = Y_F
YP set to the following value to insure that KL_YP equals 1.0.
YP = YWB_F
VEFACT = 1.0
XC = (XNOZ_F - (XTE_F - MAC))/MAC
IF (VO(V) .EQ. 0.0) THEN
L0 = 0.0
LFP = 0.0
ELSE
CU = T / (Q0*VEFACT**2.) / S
Calculate Lift loss due to the front nozzle
CALL JIEPIE
END IF

Determine how close the nozzle is to the trailing edge of
the wing minus the flap.
IF (ABS(XTE_F - XNOZ_F) .LT. MAC/4.0) THEN
TEFLAG = .TRUE.
ELSE
TEFLAG = .FALSE.
END IF

AFACT = SQRT(S/AJ * B_WB**2.0/S_WB)
DH = 0.0
\[ LT_{OG} = -0.00022 \times \sqrt{S/AJ} \times (PR_{FR}^{0.64}) \times (PER/DE)^{1.58} \]

Calculate lift loss due to the jet wake

CALL STOLWK

\[ LT_{WKB} = LT \times TT_F \]

C

Assign variables for the rear nozzles if necessary

IF (NUM_R .GT. 0) THEN

AJ = SR
SF = 0.0
T = TR
DE = DE_R
DI = D_R
N = NUM_R
PER = PI * 2.0 * SQRT(AJ / 3.14159)
PR = PR_R
Q = Q_R
Y = Y_R
YP = YWB_R
WL = WL_R
VEFACT = ((X_FR/D_F) - 1.0) / ((X_FR/D_F) + .75)
XC = (XNOZ_R - (XTE_R - MAC)) / MAC

IF (VO(V) .EQ. 0.0) THEN

LT0 = 0.0
LT_FLP = 0.0
ELSE

CU = T / (Q0*VEFACT**2.) / S
CALL JIEPIE
END IF

C

IF (ABS(XTE_R - XNOZ_R) .LT. MAC/4.0) THEN

TEFLAG = .TRUE.
ELSE

TEFLAG = .FALSE.
END IF

C

AFACT = SQRT(S/AJ * B_WB**2.0/S_WB)
DH = 0.0
LT_{OG} = -0.00022*SQRT(S/AJ)*(PR_{FR}**-0.64)*(PER/DE)**1.58

Calculate lift loss due to the jet wake

CALL STOLWK

\[ LT_{WKB} = LT_{WKB} + LT \times TT_R \]

LT_{WKW} = 0.0

END IF

C

ELSE

BODY PLANFORM

Assign body planform variables.

AR = B_B**2.0 / S_B
S = S_B
Z = Z_B
FIGWK = 'BODY'

C

Assign variables for the front nozzles.

AJ = S_F
DE = DE_F
DI = D_F
N = NUM_F
PER = PI * 2.0 * SQRT(AJ / 3.14159)
PR = PR_F
Q = Q_F
SF = SF_FB
T = T_F
WL = WL_F
Y = Y_F

C

Use distance to wingbody edge if height of wing is zero,
otherwise use the distance to the body edge.

IF (Z_W .GT. 0.0) THEN
    YP = YB_F
ELSE
    YP = YWB_F
END IF

C

VEFACT = 1.0
XC = 0.0

C

IF (VO(V) .EQ. 0.0) THEN
    LT0 = 0.0
    LT_FLP = 0.0
ELSE
    CU = T / (Q0*VEFACT**2.) / S
Calculate Lift loss due to the front nozzle
    CALL JIEPIE
END IF

C

There is no jet flap effect on the body
TEFLAG = .FALSE.
AFACT = SQRT(S / AJ * WL_B)
DH = Z_B
LT_OG = -0.00022*SQRT(S/AJ)*(PR_FR**-0.64)*(PER/DE)**1.58

C

Calculate lift loss due to the jet wake
CALL STOLWK
LT_WKB = LT * TT_F

C

Assign variables for the rear nozzles if necessary
IF (NUM_R .GT. 0) THEN
    AJ = S_R
    SF = SF_RB
    T = T_R
    DE = DE_R
    DI = D_R
    N = NUM_R
    PER = PI * 2.0 * SQRT(AJ / 3.14159)
    PR = PR_R
    Q = Q_R
    Y = Y_R

C

Use distance to wingbody edge if height of wing is zero,
otherwise use the distance to the body edge.

IF (Z_W .GT. 0.0) THEN
    YP = YB_R
ELSE
    YP = YWB_R
END IF

C

WL = WL_R
XC = 0.0
VEFACT = ((X_FR/D_F) - 1.0)/((X_FR/D_F) + .75)

IF (VO(V) .EQ. 0.0) THEN
  LT0 = 0.0
  LT_FLP = 0.0
ELSE
  CU = T / (Q0*VEFACT**2.) / S
  CALL JIEPIE
END IF

There is no jet flap effect on the body
TEFLAG = .FALSE.
AFACT = SQRT(S / AJ * WL_B)
DH = Z_B
LT_OG = -0.00022*SQRT(S/AJ)* (PR_FR**-0.64)* (PER/DE)**1.58

Calculate lift loss due to the jet wake
CALL STOLWK
LT_WKB = LT_WKB + LT * TT_R
END IF

WING PLANFORM
Assign wing planform variables.
AR = B*W**2.0 / S_W
B = B_W
S = S_W
Z = Z_W
FIGWK = 'WING'
Assign variables for the front nozzles.
AJ = S_F
DE = DE_F
DI = D_F
N = NUM_F
PER = PI * 2.0 * SQRT(AJ / 3.14159)
PR = PR_F
Q = Q_F
SF = 0.0
T = T_F
WL = WL_F
Y = Y_F

YP set to the following value to insure that KL_YP equals 1.0.
YP = -1.0 * DI
VEFACT = 1.0
XC = (XNOZ_F - (XTE_F - MAC))/MAC

IF (VO(V) .EQ. 0.0) THEN
  LT0 = 0.0
  LT_FLP = 0.0
ELSE
  CU = T / (Q0*VEFACT**2.) / S
  Calculate Lift loss due to the front nozzle
  CALL JIEPIE
END IF

Determine how close the nozzle is to the trailing edge of the wing.
IF (ABS(XTE_F - XNOZ_F) .LT. MAC/4.0) THEN
IF (VO(V) .EQ. 0.0) THEN
    LT0 = 0.0
    LT_FLP = 0.0
ELSE
    CU = T / (Q0*VEFACT**2.) / S
    CALL JIEPIE
END IF

IF (ABS(XTE_R - XNOZ_R) .LT. MAC/4.0) THEN
    TEFLAG = .TRUE.
ELSE
    TEFLAG = .FALSE.
END IF

AFACT = SQRT(S/AJ * B_W**2.0/S_W)-SQRT(S/AJ * B_CS**2.0/S_CS)
DH = Z_W
LT_OG = -0.00022*SQRT(S/AJ)* (PR_FR**-0.64)*(PER/DE)**1.58
Calculate lift loss due to the jet wake
CALL STOLWK
LT_WKW = LT * TT_F

Assign variables for the rear nozzles if necessary
IF (NUM_R .GT. 0) THEN
    AJ = S_R
    SF = 0.0
    T = T_R
    DE = DE_R
    DI = D_R
    N = NUM_R
    PER = PI * 2.0 * SQRT(AJ / 3.14159)
    PR = PR_R
    Q = Q_R
    Y = Y_R
    YP = -1.0 * DI
    WL = WL_R
    VEFACn = ((X_FR/D_F) - 1.0)/((X_FR/D_F) + .75)
    XC = (XNOZ_R - (XTE_R - MAC))/MAC

    IF (VO(V) .EQ. 0.0) THEN
        LT0 = 0.0
        LT_FLP = 0.0
    ELSE
        CU = T / (Q0*VEFACT**2.) / S
        CALL JIEPIE
    END IF

    IF (ABS(XTE_R - XNOZ_R) .LT. MAC/4.0) THEN
        TEFLAG = .TRUE.
    ELSE
        TEFLAG = .FALSE.
    END IF

    AFACT = SQRT(S/AJ * B_W**2.0/S_W)-SQRT(S/AJ * B_CS**2.0/S_CS)
    DH = Z_W
    LT_OG = -0.00022*SQRT(S/AJ)* (PR_FR**-0.64)*(PER/DE)**1.58
    Calculate lift loss due to the jet wake
    CALL STOLWK
    LT_WKW = LT_WKW + LT * TT_R
END IF

CENTER SECTION PLANFORM
Assign center section planform variables.
AR = B_CS**2.0 / S_CS
B = B_CS
S = S_CS
Z = Z_W
C FIGWK = 'CSEC'
Assign variables for the front nozzles.
AJ = S_F
DE = D_E_F
DI = D_F
N = NUM_F
PER = PI * 2.0 * SQRT(AJ / 3.14159)
PR = PR_F
Q = Q_F
SF = 0.0
T = T_F
WL = WL_F
YP set to the following value to insure that KL_YP equals 1.0.
YP = -1.0 * DI
VEFACT = 1.0
XC = 0.0
IF (VO(V) .EQ. 0.0) THEN
  LT0 = 0.0
  LT_FLP = 0.0
ELSE
  CU = T / (Q0*VEFACT**2.) / S
Calculate lift loss due to the front nozzle
CALL JIEPIE
END IF
TEFLAG = .FALSE.
AFACT = SQRT(S/AJ * B_W**2.0/S_W)-SQRT(S/AJ * B_CS**2.0/S_CS)
DH = Z_W
LT_OG = -0.00022*SQRT(S/AJ)*(PR_FR**-0.64)*(PER/DE)**1.58
Calculate lift loss due to the jet wake
CALL STOLWK
LT_WKC = LT * TT_F
C Assign variables for the rear nozzles if necessary
IF (NUM_R .GT. 0) THEN
AJ = S_R
DE = D_E_R
DI = D_R
N = NUM_R
PER = PI * 2.0 * SQRT(AJ / 3.14159)
PR = PR_R
Q = Q_R
SF = 0.0
T = T_R
Y = Y_R
YP = -1.0 * DI
WL = WL_R
VEFACT = ((X_FR/D_F) - 1.0)/((X_FR/D_F) + .75)
XC = 0.0
IF (VO(V) .EQ. 0.0) THEN
  LT0 = 0.0
  LT_FLP = 0.0
ELSE
  CU = T / (Q0*VEFACT**2.) / S
CALL JIEPIE
SUBROUTINE JIEPIE

ACRONYM: Jet Induce Effects for the Power Induced Effects module.

PURPOSE: The purpose of JIEPIE is to calculate the change in the lift coefficient caused by the jet induced effects on a flat plate. JIEPIE takes into account aspect ratio, longitudinal, vertical, and lateral position of jets, deflection angle, distance from side of body, and non-circular or closely spaced jets.

LOCAL VARIABLES (in addition to the above parameters):

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CL_BAS</td>
<td>R</td>
<td>-</td>
<td>----</td>
<td>Basic lift loss on a square planform with the jet at the center of the planform.</td>
</tr>
<tr>
<td>FP_BAS</td>
<td>R</td>
<td>-</td>
<td>----</td>
<td>Basic lift gain due to jet flap action</td>
</tr>
<tr>
<td>K_JFH</td>
<td>R</td>
<td>-</td>
<td>----</td>
<td>Adjustment factor Jet Flap</td>
</tr>
<tr>
<td>K_PR</td>
<td>R</td>
<td>-</td>
<td>----</td>
<td>Adjustment factor for Net Jet Pressure Ratio</td>
</tr>
<tr>
<td>KL_A</td>
<td>R</td>
<td>-</td>
<td>----</td>
<td>Adjustment factor for aspect ratio</td>
</tr>
<tr>
<td>KL_B</td>
<td>R</td>
<td>-</td>
<td>----</td>
<td>Adjust for the &quot;jet-flap&quot; span</td>
</tr>
<tr>
<td>KL_D</td>
<td>R</td>
<td>-</td>
<td>----</td>
<td>Adjustment factor for the jet deflection angle.</td>
</tr>
<tr>
<td>KL_N</td>
<td>R</td>
<td>-</td>
<td>----</td>
<td>Adjustment factor for non-circular nozzle configuration (had closely spaced jets).</td>
</tr>
<tr>
<td>KL_X1</td>
<td>R</td>
<td>-</td>
<td>----</td>
<td>Adjustment factor for longitudinal</td>
</tr>
</tbody>
</table>
**GLOBAL VARIABLES (in addition to the above parameters and local vars):**

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFACT</td>
<td>R</td>
<td>I</td>
<td></td>
<td>Area ratios used in calculating the lift loss due to the jet wake.</td>
</tr>
<tr>
<td>AJ</td>
<td>R</td>
<td>I</td>
<td>Sq. Ft</td>
<td>Total area of the jets in question.</td>
</tr>
<tr>
<td>AR</td>
<td>R</td>
<td>I</td>
<td></td>
<td>Aspect ratio of the planform.</td>
</tr>
<tr>
<td>B</td>
<td>R</td>
<td>I</td>
<td></td>
<td>Width of current body config.</td>
</tr>
<tr>
<td>CU</td>
<td>R</td>
<td>I</td>
<td></td>
<td>Momentum coef. based on nozzle.</td>
</tr>
<tr>
<td>DH</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Vertical position of the jet based on the planform.</td>
</tr>
<tr>
<td>DI</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Diameter of the jets in question.</td>
</tr>
<tr>
<td>FIGWK</td>
<td>C</td>
<td></td>
<td></td>
<td>Flag which notifies JIEPIE which planform is in use.</td>
</tr>
<tr>
<td>HEIGHT</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Height of aircraft.</td>
</tr>
<tr>
<td>KL_F</td>
<td>R</td>
<td>I</td>
<td></td>
<td>Adjustment factor used for flap extension.</td>
</tr>
<tr>
<td>LT</td>
<td>R</td>
<td>O</td>
<td></td>
<td>Lift loss retured by STOLWK.</td>
</tr>
<tr>
<td>LT_FLP</td>
<td>R</td>
<td>O</td>
<td></td>
<td>Basic lift gain due to jet flap action.</td>
</tr>
<tr>
<td>LT0</td>
<td>R</td>
<td>-</td>
<td></td>
<td>Lift loss out of ground effect.</td>
</tr>
<tr>
<td>LT_OG</td>
<td>R</td>
<td>O</td>
<td></td>
<td>Out of ground lift loss</td>
</tr>
<tr>
<td>MC</td>
<td>R</td>
<td>I</td>
<td></td>
<td>Mean Aerodynamic Chord.</td>
</tr>
<tr>
<td>N</td>
<td>I</td>
<td>I</td>
<td></td>
<td>Number of nozzles based on which nozzle used (Front, Rear)</td>
</tr>
<tr>
<td>PR</td>
<td>R</td>
<td>I</td>
<td></td>
<td>Pressure Ratio of local jets.</td>
</tr>
<tr>
<td>Q</td>
<td>R</td>
<td>I</td>
<td>lb/Sq Ft</td>
<td>Jet dynamic pressure</td>
</tr>
<tr>
<td>QO</td>
<td>R</td>
<td>O</td>
<td>lb/Sq Ft</td>
<td>Free stream dynamic pressure.</td>
</tr>
<tr>
<td>S</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
<td>Area of the configuration.</td>
</tr>
<tr>
<td>SF</td>
<td>R</td>
<td>I</td>
<td>Sq Ft</td>
<td>Area of configuration that is in front of nozzles.</td>
</tr>
<tr>
<td>T</td>
<td>R</td>
<td>I</td>
<td>lb</td>
<td>Thrust of the jets in question.</td>
</tr>
<tr>
<td>TEFALG</td>
<td>L</td>
<td>-</td>
<td></td>
<td>Flag which notifies STOLWK how close the nozzle is to the trailing edge of the wing.</td>
</tr>
<tr>
<td>WL</td>
<td>R</td>
<td>I</td>
<td></td>
<td>Width to length ratio of nozzles.</td>
</tr>
<tr>
<td>XC</td>
<td>R</td>
<td>-</td>
<td></td>
<td>Position of nozzle with respect to wing trailing edge (in % chord)</td>
</tr>
<tr>
<td>Y</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Lateral spacing of jets.</td>
</tr>
<tr>
<td>YP</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Distance of jet center to body side.</td>
</tr>
<tr>
<td>VEFACT</td>
<td>R</td>
<td>I</td>
<td></td>
<td>Effective velocity factor which accounts for the reduction in velocity at the rear nozzles due to</td>
</tr>
</tbody>
</table>
Z R I

RESPIE Common Block

NAME TYPE I/O UNIT

D I O ----
H I O ----
LD R I ----
LH R I ----
LV R I ----
REC3 I O ----
V I O ----
VO(500) R O ----

FILES USED:

LOGICAL UNIT I/O DESCRIPTION

<None>

COMMONS USED:

NAME DESCRIPTION

PIEWK Power Induced Effects for stolWK and jiepie - Contains all variables which are passes to the STOLWK and JIEPIE subroutines.

DESCRIPTION

RESPIE RESULTS from all PIE subroutines - Contains variables which are the main outputs from each subroutine

CALLED BY:

NAME DESCRIPTION

WKCALL Isolates and controls execution of STOLWK

ROUTINES CALLED:

NAME DESCRIPTION

<None>

NOTES: None.

REFERENCES:


ENVIRONMENT:

FORTRAN 77, VAX 6800, VAX 11/785, SGI IRIS 4D series.

NON-STANDARD CODE:

?.

AUTHOR(S):

Kipp E. Howard (KEH), Cal Poly San Luis Obispo, NASA Ames Moffett

REVISION HISTORY:

DATE INITIALS & DESCRIPTION
05/08/90 KEH -- Initial completion of code.
07/13/91 KEH -- Eliminated extraneous spacing and fixed comments

the front nozzle.
Vertical distance of the nozzles.

DESCRIPTION

Nozzle deflection angle number counter.
Height number counter
Total number of nozzle deflection angles.
Total number of height values.
Total number of velocity values.
Record number for any file which replaces a 3 X 3 matrix
Velocity number counter
Actual velocity value
C Variable declaration
   REAL CL_BAS, FP_BAS, VE, KL_A, KL_B, KL_X1, KL_X2, KL_Y, KL_Z,
   $  KL_YP, KL_D, K_JFH, KL_N, K_PR
C
C Determine the effective velocity ratio
   VE = VEFACT * SQRT(Q0/Q)
C Calculate Cl,basic (lift loss on a square planform with a jet at the
center of the planform. (AFWAL-TR-87-3019 vol III, pg E-4)
   CL_BAS = -3.0 * VE**2.0 * (SQRT(S/AJ) - 1.0)**(2.0/3.0) +
         35.0 * VE**5.5 * (SQRT(S/AJ) - 1.0)
C Calculate Basic lift gain due to jet flap action (AFWAL-TR-87-3019
Vol II Sec. 4.1.3.1 pg 4-13)
   FP_BAS = ((AR + 2.0 * CU / 3.1415926) / (AR + 2.0 + .604 *
         SQRT(CU) + .876 * CU)**.63 - CU*
         SIN(DFL(D)/57.296)) / CU
C
   IF (FP_BAS * CU .GT. 1.94 * AR) THEN
      FP_BAS = 1.94 * AR / CU
   ELSE IF (FP_BAS * CU .GT. 12.57) THEN
      FP_BAS = 12.57 / CU
   ELSE IF (FP_BAS .LT. 0.0) THEN
      FP_BAS = -0.0
   END IF
C
C Adjust for Aspect Ratio
   IF (AR .LT. 1.0) THEN
      KL_A = AR**.05 - .5 * (1.0 - AR)**3.6
   ELSE
      KL_A = 1.0 - .4 * (1.0 - 1.0 / AR)**4.0
   END IF
C
C Adjust for longitudinal position of the jet
C Bodies
   IF (AR .LT. 1.0) THEN
      KL_X1 = 1.35 - .7 * (SF/S) - 2.5 * (ABS(SF/S - 0.5))**2.5
      KL_X2 = 0.0
C Wings
   ELSE IF (AR .GE. 1.0) THEN
C
      IF (XC .LE. .75) THEN
         KL_X1 = 1.0
         KL_X2 = 0.0
      ELSE IF (XC .GT. .75 .AND. XC .LE. 1.5) THEN
         KL_X1 = 0.0
         KL_X2 = 1.0
      ELSE IF (XC .GT. 1.5 .AND. XC .LT. 3.5) THEN
         KL_X1 = -0.5 * XC + 1.75
         KL_X2 = 0.0
      ELSE IF (XC .GT. 3.5) THEN
         KL_X1 = 0.0
         KL_X2 = 0.0
      END IF
C
END IF
C
IF (FIGWK .EQ. 'CSEC') THEN
  KL_X1 = 1.0
  KL_X2 = 0.0
END IF

C Adjust for the vertical position of the jet.
  KL_Z = 1.0 - .15 * ABS(Z/DI)

C Adjust for lateral spacing of the jets
IF (Y/DI .LT. 3.0 .AND. Y .NE. 0.0) THEN
  KL_Y = 1.2 - .1 * (Y/DI - 1.0)
ELSE
  KL_Y = 1.0
END IF

C Adjust for the distance of the jet center from the side of the body.
IF (YP/DI .LT. -.55) THEN
  KL_YP = 1.0
ELSE IF (YP/DI .LT. 0.0) THEN
  KL_YP = .85 - .273 * YP/DI
ELSE IF (YP/DI .LT. 2.0) THEN
  KL_YP = .85 - .425 * YP/DI
ELSE
  KL_YP = 0.0
END IF

C Adjust for the jet deflection angle.
  KL_D = DFL(D) / 90.0

C Adjust for non-circular nozzle configuration (and closely spaced jets).
  KL_N = 1.0 - (14.0 * VE**1.5 - 22.0 * VE**2.5) * (1.0 - WL**.3)

C Adjust for the "jet-flap" span
IF (FIGWK .EQ. 'WING' .OR. AR .GE. 1.0) THEN
  KL_B = N * DI / B
ELSE
  KL_B = 0.0
END IF

C Adjust for Net Jet Pressure Ratio
IF (PR .GE. 1.893) THEN
  K_PR = (PR/1.893)**.25
ELSE
  K_PR = 1.0
END IF

C Adjust Jet Flap calculation for aircraft height.
  K_JFH = 1.0 - .055 * (MC/HEIGHT)**1.3

C****************************************************************************
C Make final calculation for C1
C Lift loss due to jet interference
  LT0 = CL_BAS * KL_A * KL_X1 * KL_Y * KL_Z * KL_YP * KL_D * KL_N
       * K_PR
C Lift gain due to jet flap interaction
  LT_FLP = FP_BAS * KL_X2 * KL_Z * KL_D * KL_B * KL_F * K_JFH
C
RETURN
END
**SUBROUTINE STOLWK**

**ACRONYM:** STOL operation - estimation of jet Wake term

**PURPOSE:** The purpose of STOLWK is to calculate the lift loss due to the jet wake system.

**LOCAL VARIABLES (in addition to the above parameters):**

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>VE</td>
<td>R</td>
<td>I</td>
<td>-</td>
<td>Effective velocity ratio</td>
</tr>
<tr>
<td>XK</td>
<td>R</td>
<td>I</td>
<td>-</td>
<td>Factor used in calculating H2D</td>
</tr>
</tbody>
</table>

**GLOBAL VARIABLES (in addition to the above parameters and local vars):**

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIEWK</td>
<td>Common Block</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PIEWK Common Block**

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFECT</td>
<td>R</td>
<td>I</td>
<td>-</td>
<td>Area ratios used in calculating the lift loss due to the jet wake.</td>
</tr>
<tr>
<td>AR</td>
<td>R</td>
<td>I</td>
<td>-</td>
<td>Aspect ratio of the planform.</td>
</tr>
<tr>
<td>DH</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Vertical position of the jet based on the planform.</td>
</tr>
<tr>
<td>DI</td>
<td>R</td>
<td>I</td>
<td>Ft</td>
<td>Diameter of the jets in question.</td>
</tr>
<tr>
<td>LT</td>
<td>R</td>
<td>O</td>
<td>-</td>
<td>Lift loss returned by STOLWK</td>
</tr>
<tr>
<td>LT_FLX</td>
<td>R</td>
<td>O</td>
<td>-</td>
<td>Basic lift gain due to jet flap action.</td>
</tr>
<tr>
<td>LT0</td>
<td>R</td>
<td>-</td>
<td>-</td>
<td>Lift loss out of ground effect.</td>
</tr>
<tr>
<td>Q</td>
<td>R</td>
<td>I</td>
<td>lb/Sq Ft</td>
<td>Free stream dynamic pressure.</td>
</tr>
<tr>
<td>Q0</td>
<td>R</td>
<td>O</td>
<td>lb/Sq Ft</td>
<td></td>
</tr>
<tr>
<td>TEFALG</td>
<td>L</td>
<td></td>
<td>-</td>
<td>Flag which notifies STOLWK how close the nozzle is to the trailing edge of the wing.</td>
</tr>
<tr>
<td>WL</td>
<td>R</td>
<td>I</td>
<td>-</td>
<td>Width to length ratio of nozzels.</td>
</tr>
<tr>
<td>VEFACT</td>
<td>R</td>
<td>I</td>
<td>-</td>
<td>Effective velocity factor which accounts for the reduction in velocity at the rear nozzles due to the front nozzle.</td>
</tr>
</tbody>
</table>

**RESPIE Common Block**

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>I</td>
<td>O</td>
<td>-</td>
<td>Nozzle deflection angle number counter.</td>
</tr>
<tr>
<td>DFL(500)</td>
<td>R</td>
<td>I</td>
<td>deg</td>
<td>Actual nozzle deflection angle value</td>
</tr>
<tr>
<td>H</td>
<td>I</td>
<td>O</td>
<td>-</td>
<td>Height number counter</td>
</tr>
<tr>
<td>HT(500)</td>
<td>R</td>
<td>O</td>
<td>Ft</td>
<td>Actual height value</td>
</tr>
</tbody>
</table>

**FILES USED:**

<table>
<thead>
<tr>
<th>LOGICAL UNIT</th>
<th>I/O</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>(none)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**COMMONS USED:**

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIEWK</td>
<td>Power Induced Effects for stolWK and jiepie - Contains all variables which are passes to the STOLWK and JIEPIE subroutines.</td>
</tr>
</tbody>
</table>
RESPIE
RESults from all PIE subroutines - Contains variables
which are the main outputs from each subroutine

CALLED BY:
NAME DESCRIPTION
-------------
WKCALL Isolates and controls execution of STOLWK

ROUTINES CALLED:
NAME DESCRIPTION
---------

<None>

NOTES: None.

REFERENCES:
1) Stewart, V.R. and Kuhn, R.E. "A Method for Prediction of the
Aerodynamic Stability and Control Parameters of STOL Aircraft
Configurations." North American Aircraft Operations. Rockwell
International Corporation. AFWAL-TR-87-3019. Volume II & III.
June 1987.

ENVIRONMENT:
FORTRAN 77, VAX 6800, VAX 11/785, SGI IRIS 4D series.

REFERENCES:

AUTHOR(S):
Kipp E. Howard (KEH), Cal Poly San Luis Obispo, NASA Ames Moffett

REVISION HISTORY:
DATE INITIALS & DESCRIPTION
05/09/90 KEH -- Initial completion of code.
07/13/91 KEH -- Eliminated extraneous spacing and fixed comments

#include "respie.inc"
#include "piewk.inc"

REAL VE, KX

Determine the effective velocity ratio
VE=VEFACT * SQRT(Q0/Q)

Set KX equal to 1 because KX is based on the distance between the
center of the jet pattern to the center of the front jet(s). The
lift loss due to the jet wake is calculated for the front jets and
rear jets individually, so the distance from the center of the jet
pattern to the center of the front jet(s) is always going to be zero
which causes KX to equal 1.0.
KX = 1.0

Test to see which equation to use (near or far away from the TE)
IF (AR .GE. 1.0 .AND. TEFLAG) THEN
Near the trailing edge of the wing (Lift gain)
IF (VE .EQ. 0.0) THEN
LT = LT FLP
ELSE
LT = LT FLP * (1.0 + .01/VE**2.0 * AFACT
* (DFL(D)/90.)**1.34 * ((HT(H) + DH)/DI)**-1.27)
END IF
ELSE
Not near the trailing edge of the wing (Lift loss)
LT = LT0 * (1.0 - .7 * AFACT * (DFL(D)/90.)**1.34


SUBROUTINE PIEP

ACRONYM: Powered Induced Effects Printing routine

PURPOSE: The purpose of PIEP is to print all the variables in a
convient, easy to read style.

LOCAL VARIABLES (in addition to the above parameters):

NAME TYPE I/O UNITS DESCRIPTION
--------- ---- ----- -----------------------------
AEND R I deg Ending value for angle of attack
ANG_FR R I Ft Front and Rear jet deflection ANGLE
ASTART R I deg Starting value for angle of attack
ASTEP R I deg Step value for angle of attack
B_B R I Ft Width of Body
B_CS R I Ft Width of Center Section
B_JP R I Ft Width of Jet Pattern
B_W R I Ft Width of Wing (Wing span)
B_WB R I Ft Width of Wing-Body
D_F R I Ft Diameter of each Front jet
D_R R I Ft Diameter of each Rear jet
D_RCS R I Ft Diameter of Roll RCS nozzle
DB_B R I Ft Dbar of Body
DB_CS R I Ft Dbar of Center Section
DB_W R I Ft Dbar of Wing
DB_WB R I Ft Dbar of Wing-Body
DE_F R I Ft Effective Diameter of Front & Rear
DE_FR R I Ft jets combined
DE_R R I Ft Effective Diameter of Rear jets
DEND R I deg Ending value for jet deflection
DR R I ---- Density Ratio (Jet / Atm)
DSTEP R I deg Step value for jet deflection angle
F_NAME C I ---- Name of file which contains nozzle
H R I Ft Height of nozzle exit above ground
HEND R I Ft Ending value for altitude
HSTART R I Ft Starting value for altitude
HSTEP R I Ft Step value for altitude
ICALC I I ---- Global execution flag for ACSYNT
1) Input data
2) Make calculations
3) Output necessary data

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KB</td>
<td>R</td>
<td>I</td>
<td>Boundary layer factor</td>
</tr>
<tr>
<td>KR</td>
<td>R</td>
<td>I</td>
<td>Body contour factor</td>
</tr>
<tr>
<td>L_B</td>
<td>R</td>
<td>I</td>
<td>Length of body</td>
</tr>
<tr>
<td>L_CS</td>
<td>R</td>
<td>I</td>
<td>Length of Center Section</td>
</tr>
<tr>
<td>L_F</td>
<td>R</td>
<td>I</td>
<td>Length of Front jet</td>
</tr>
<tr>
<td>L_R</td>
<td>R</td>
<td>I</td>
<td>Length of Rear jet</td>
</tr>
<tr>
<td>L_WB</td>
<td>R</td>
<td>I</td>
<td>Length of Wing-Body</td>
</tr>
<tr>
<td>MAC</td>
<td>R</td>
<td>I</td>
<td>Mean Aerodynamic Chord of wing</td>
</tr>
<tr>
<td>MD_RCS</td>
<td>R</td>
<td>I</td>
<td>Mass flow rate for one RCS nozzle</td>
</tr>
<tr>
<td>N_BODY</td>
<td>I</td>
<td>I</td>
<td>Number of data points for Body</td>
</tr>
<tr>
<td>N_WING</td>
<td>I</td>
<td>I</td>
<td>Number of data points for Wing planform</td>
</tr>
<tr>
<td>N_WB</td>
<td>I</td>
<td>I</td>
<td>Number of data points for Wing-Body planform</td>
</tr>
<tr>
<td>NUM</td>
<td>I</td>
<td>I</td>
<td>Total NUMBER of jets</td>
</tr>
<tr>
<td>NUM_F</td>
<td>I</td>
<td>I</td>
<td>NUMBER of Front jets</td>
</tr>
<tr>
<td>NUM_R</td>
<td>I</td>
<td>I</td>
<td>NUMBER of Rear jets</td>
</tr>
<tr>
<td>PER_FR</td>
<td>R</td>
<td>I</td>
<td>Total perimeter of jets</td>
</tr>
<tr>
<td>PP_LID</td>
<td>R</td>
<td>I</td>
<td>Ratio of perimeter enclosed by lids to total perimeter C</td>
</tr>
<tr>
<td>PR_FR</td>
<td>R</td>
<td>I</td>
<td>Jet Pressure Ratio for all jets</td>
</tr>
<tr>
<td>PR_RCS</td>
<td>R</td>
<td>I</td>
<td>Roll RCS Pressure Ratio</td>
</tr>
<tr>
<td>PT_RCS</td>
<td>R</td>
<td>I</td>
<td>Total pressure for one roll RCS jet</td>
</tr>
<tr>
<td>Q_FR</td>
<td>R</td>
<td>I</td>
<td>Dynamic pressure for Front and Rear jets</td>
</tr>
<tr>
<td>Q_RCS</td>
<td>R</td>
<td>I</td>
<td>Dynamic pressure for roll RCS jets</td>
</tr>
<tr>
<td>S_B</td>
<td>R</td>
<td>I</td>
<td>Area of Body</td>
</tr>
<tr>
<td>S_CS</td>
<td>R</td>
<td>I</td>
<td>Area of Center Section</td>
</tr>
<tr>
<td>S_JP</td>
<td>R</td>
<td>I</td>
<td>Area enclosed by Jet Pattern</td>
</tr>
<tr>
<td>S_F</td>
<td>R</td>
<td>I</td>
<td>Area of Front jets</td>
</tr>
<tr>
<td>S_FR</td>
<td>R</td>
<td>I</td>
<td>Total jet exit area</td>
</tr>
<tr>
<td>S_LID</td>
<td>R</td>
<td>I</td>
<td>Area enclosed by LIDS</td>
</tr>
<tr>
<td>S_R</td>
<td>R</td>
<td>I</td>
<td>Area of Rear jets</td>
</tr>
<tr>
<td>S_W</td>
<td>R</td>
<td>I</td>
<td>Area of Wing</td>
</tr>
<tr>
<td>S_WB</td>
<td>R</td>
<td>I</td>
<td>Area of Wing-Body</td>
</tr>
<tr>
<td>SF_FB</td>
<td>R</td>
<td>I</td>
<td>Area ahead of Front jets using body planform</td>
</tr>
<tr>
<td>SF_FWB</td>
<td>R</td>
<td>I</td>
<td>Area ahead of Front jets using the wingbody planform</td>
</tr>
<tr>
<td>SF_RB</td>
<td>R</td>
<td>I</td>
<td>Area ahead of Rear jets</td>
</tr>
<tr>
<td>SF_RWB</td>
<td>R</td>
<td>I</td>
<td>Area ahead of Rear jets using the wingbody planform</td>
</tr>
<tr>
<td>SP_JP</td>
<td>R</td>
<td>I</td>
<td>Actual surface area within area</td>
</tr>
<tr>
<td>SPLY_F</td>
<td>R</td>
<td>I</td>
<td>Area enclosed by nozzles</td>
</tr>
<tr>
<td>SPLY_R</td>
<td>R</td>
<td>I</td>
<td>SPLay angle of Front jet</td>
</tr>
<tr>
<td>T_F</td>
<td>R</td>
<td>I</td>
<td>SPLay angle of Rear jet</td>
</tr>
<tr>
<td>T_R</td>
<td>R</td>
<td>I</td>
<td>Thrust of Front jets</td>
</tr>
<tr>
<td>T_RCS</td>
<td>R</td>
<td>I</td>
<td>Thrust of Rear jets</td>
</tr>
<tr>
<td>TF_RCS</td>
<td>R</td>
<td>I</td>
<td>Thrust of roll RCS nozzle</td>
</tr>
<tr>
<td>TT_F</td>
<td>R</td>
<td>I</td>
<td>Temperature of flow in roll RCS nozzle</td>
</tr>
<tr>
<td>TT_R</td>
<td>R</td>
<td>I</td>
<td>Front Thrust / total Thrust</td>
</tr>
<tr>
<td>VEND</td>
<td>R</td>
<td>I</td>
<td>Rear Thrust / total Thrust</td>
</tr>
<tr>
<td>VO</td>
<td>R</td>
<td>I</td>
<td>Ending value for aircraft velocity</td>
</tr>
<tr>
<td>VSTART</td>
<td>R</td>
<td>I</td>
<td>Starting value for aircraft</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VSTEP</td>
<td>Step value for aircraft velocity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W_F</td>
<td>Width of Front jet</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W_R</td>
<td>Width of Rear jet</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WINE</td>
<td>Flow Weight of Inlet / Flow Weight of Exit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WB</td>
<td>Width to Length ratio of Body</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W_LP</td>
<td>Width to Length ratio of Center Section</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W_LT</td>
<td>Width to Length ratio of Jet Pattern</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WL_F</td>
<td>Width to Length ratio of Front jets</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WL_R</td>
<td>Width to Length ratio of Rear jets</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WL_WB</td>
<td>Width to Length ratio of Wing-Body X-coordinates of Body planform</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X_BODY</td>
<td>Distance between Front &amp; Rear jets</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X_FR</td>
<td>Distance of roll RCS nozzle ahead of wing trailing edge</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X_RCS</td>
<td>X-coordinates of Wing-Body planform</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X_WB</td>
<td>X-coordinates of Wing planform</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X_WING</td>
<td>Distance of center of area ahead of CG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X_CA</td>
<td>X-coordinate of Center of Area</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X_CG</td>
<td>X-coordinate of Center of Gravity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XCG_C2</td>
<td>Distance from CG to MAC/2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XCG_C4</td>
<td>Distance from CG to MAC/4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XCG_F</td>
<td>Distance Front jet is ahead of CG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XCG_I</td>
<td>Inlet longitudinal distance ahead of CG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XCG_R</td>
<td>Distance Rear jet is ahead of CG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XNOZ_F</td>
<td>X-coordinates of Front NOZZle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XNOZ_R</td>
<td>X-coordinates of Rear NOZZle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y_BODY</td>
<td>Y-coordinates of Body planform</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y_F</td>
<td>Distance between Front jets</td>
<td></td>
<td></td>
</tr>
<tr>
<td>YNOZ_F</td>
<td>Y-coordinates of Front NOZZle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>YNOZ_R</td>
<td>Y-coordinates of Rear NOZZle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y_R</td>
<td>Distance between Rear jets</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y_RCS</td>
<td>Distance of roll RCS nozzle in from wingtip</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y_WB</td>
<td>Y-coordinates of Wing-Body planform</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y_WING</td>
<td>Y-coordinates of Wing planform</td>
<td></td>
<td></td>
</tr>
<tr>
<td>YB_F</td>
<td>Lateral distance from Body to Front jets (for external jets)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>YB_R</td>
<td>Lateral distance from Body to Rear jets (for external jets)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z_B</td>
<td>Height of body base above nozzle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZCG_I</td>
<td>Inlet vertical distance above CG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z_W</td>
<td>Height of wing above nozzle</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**FILES USED:**

<table>
<thead>
<tr>
<th>Logical Unit</th>
<th>I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>68</td>
<td>O</td>
<td>Sequencial file for table output</td>
</tr>
<tr>
<td>70</td>
<td>O</td>
<td>Direct access file for storage of lift increment due to suckdown</td>
</tr>
<tr>
<td>71</td>
<td>O</td>
<td>Direct access file for storage of lift increment due to fountain</td>
</tr>
<tr>
<td>72</td>
<td>O</td>
<td>Direct access file for storage of lift increment due to the ground vortex on the body</td>
</tr>
</tbody>
</table>
C Direct access file for storage of lift increment due to the ground vortex on the wing
C Direct access file for storage of lift increment due to the jet wake on the body
C Direct access file for storage of lift increment due to the jet wake on the wing (without center section)

COMMONS USED:
NAME   DESCRIPTION
------   -----------------------------------------------
FIGPIE   configuration for Power Induced Effects - Contains all variables needed to define the configuration and other parameters of the aircraft.
PIEFLAGS Power Induced Effects FLAGS - Contains variables which help keep track of the configuration of the aircraft.
PIERROR Power Induced Effects errors - Contains all error flags within PIE.
RESPIE RESults from all POwer Induced Effects subroutines - Contains results from each subroutine along with variables for height, velocity, jet deflection angle, and angle of attack and their counters.

CALLED BY:
NAME   DESCRIPTION
------   -----------------------------------------------
COPPIE Controls execution and coordination of Power Induced Effects Module

ROUTINES CALLED:
NAME   DESCRIPTION
------   -----------------------------------------------
JIETAB Creates output file which is used by ACSYNT

NOTES: None.
REFERENCES:
1) None.
ENVIRONMENT:
FORTRAN 77, VAX 6800, VAX 11/785, SGI IRIS 4D series.
NON-STANDARD CODE:
?
AUTHOR(S):
Kipp E. Howard (KEH), Cal Poly San Luis Obispo, NASA Ames Res Ctr.
REVISION HISTORY:
DATE       INITIALS & DESCRIPTION
04/03/90   KEH -- Initial coding complete.
07/14/91   KEH -- Eliminated extraneous spacing and fixed comments

#include "figpie.inc"
#include "pieflag.inc"
#include "pierror.inc"
#include "respie.inc"

REAL PI, DIV
INTEGER CHOICE

PI = 3.1415926

C Print out all input variables
WRITE (6,997) CONFIG, B_B, B_W, B_JP, B_CS, L_B, L_WB,
$ S_JP, S_CS, S_B, S_W, S_WB, S_LID, S_CS, DB_B, DB_W, DB_WB,
WRITE (6,998)
$ NUM_F, NUM_R, NUM, PR_FR, D_F, D_R, DR, DE_F, DE_R, DE_FR, WIWE,
$ S_F, S_R, S_FR, XCG_I, SPLY_F, SPLY_R, PER_FR, ZCG_I,
$ TT_F, TT_R, Q_F, Q_R, XTE_F, XTE_R,
$ XCG_F, XCG_R, SF_FB, SF_RB, SF_FBW, SF_FBW, XNOZ_F, XNOZ_R, YNOZ_F, YNOZ_R
WRITE (6,999)
$ E_1, D_RCS, XCG_C2, E_3, PR_RCS, XCG_C4, E_4, T_RCS, XCA_CG,
$ THA_1*180./PI, TP_RCS, KB, THA_3*180./PI, PT_RCS, KLF,
$ THA_4*180./PI, MD_RCS, XCG, S_FA1, Q_RCS, X_CA, S_FA3, X_RCS,
$ S_FA4, Y_RCS, SP_FA1, SP_FA3, SP_FA4, Y_1, Y_3, Y_4,
$ YP_1, YP_3, YP_4, SCALE3

C IF (N_BODY .GE. N_WING .AND. N_WB .EQ. 0) THEN
N = N_BODY
ELSE IF (N_BODY .LT. N_WING .AND. N_WB .EQ. 0) THEN
N = N_WING
ELSE
N = N_WB
END IF
C
WRITE (6,990)
DO I = 1,N
C IF (I .LE. N_CS .AND. I .LE. N_WING) THEN
WRITE (6,995) X_BODY(I), Y_BODY(I), X_WING(I), Y_WING(I),
X WB(I), Y WB(I), X_CS(I), Y_CS(I)
$ ELSE IF (I .LE. N_BODY .AND. I .LE. N_WING) THEN
WRITE (6,991) X_BODY(I), Y_BODY(I), X_WING(I), Y_WING(I),
X WB(I), Y WB(I)
$ ELSE IF (I .GT. N_BODY .AND. I .LE. N_WING) THEN
WRITE (6,992) X_WING(I), Y_WING(I), X WB(I), Y WB(I)
ELSE IF (I .LE. N_BODY .AND. I .GT. N_WING) THEN
WRITE (6,993) X_BODY(I), Y_BODY(I), X WB(I), Y WB(I)
ELSE IF (I .LE. N_CS .AND. I .GT. N_WING) THEN
WRITE (6,996) X_BODY(I), Y_BODY(I), X WB(I), Y WB(I),
X_CS(I), Y_CS(I)
ELSE
WRITE (6,994) X WB(I), Y WB(I)
END IF
C END DO
WRITE (6,800) DBERR, SDAERR, T_ERR
990 FORMAT (/7X,'Body',13X,'Wing',11X,'Wingbody',8X,'Center Sec',/5X,
$ 'x',6X,'y',3(9X,'x',6X,'y')
991 FORMAT (1X,2(F7.2),2(3X,2(F7.2)))
992 FORMAT (1X,16X,2(F7.2),3X,2(F7.2))
993 FORMAT (1X,2(F7.2),20X,2(F7.2))
994 FORMAT (35X,2(F7.2))
995 FORMAT (1X,2(F7.2),3(3X,2(F7.2)))
996 FORMAT (1X,2(F7.2),17X,2(3X,2(F7.2)))
997 FORMAT (1X,A18/
$ 1X,'Body',11X,'Wing',11X,'Wing-Body',6X,'Jet Pattern ',
$ 1X,'Center Section',
$ /1X,'------------------',1X,'------------------',
$ 1X,'------------------',1X,'------------------',
CREATE USEFUL TABLES TO GRAPH

IF (PRTFLG) THEN
    WRITE (6, 1100)
    READ *, CHOICE

    IF (CHOICE .LT. 0 .OR. CHOICE .GT. 7) THEN
        WRITE (6, *) '*** Invalid Choice ***'
        GO TO 20
    END IF

    GO TO (11111, 100, 200, 300, 400, 500, 600, 700) CHOICE + 1

    HOVER GROUND EFFECT OUTPUT
    WRITE (68, *) '1'
    WRITE (68, *) CONFIG, ', Hover Ground Effect'

    IF (HDEOUT) THEN
        WRITE (68, *) 'H/De'
        DIV = DE / FR
    ELSE
        WRITE (68, *) 'Height'
        DIV = 1.0
    END IF

    WRITE (68, *) 'dL/dT'
    WRITE (68, *) '6'
    WRITE (68, *) 'OG'
    WRITE (68, *) 'SUCKDOWN'
    WRITE (68, *) 'FOUNTAIN'
    WRITE (68, *) 'LIDS'
    WRITE (68, *) 'TOTAL'
    WRITE (68, *) 'Zero'
    WRITE (68, 1105) H
    DO I = 1, H
        WRITE (68, 1110) HT(I) / DIV, H_LT0G(I), H_LTS(I), H_LTF(I), H_LTL(I), H_LT(I), 0.0
    END DO
    GO TO 9999

STOLSF TABLE OUTPUT

IF (CHOICE .LT. 1 .OR. CHOICE .GT. 3) THEN
    WRITE (6, *) '*** Invalid Choice ***'
    GO TO 200
END IF

GO TO (210, 230, 220) CHOICE

WRITE (6, 1230)
DO ID = 1, D
   WRITE(6,1205) ID, DFL(ID)
END DO
WRITE(6,1330)
READ *, ID

WRITE(6,1220)
DO IV = 1, V
   WRITE(6,1205) IV, VO(IV)
END DO
WRITE(6,1320)
READ*, IV

WRITE(68,*),'1'
WRITE(68,1250) CONFIG,', STOLSF','DFL=',DFL(ID), ' VO=', VO(IV)

IF (HDEOUT) THEN
   WRITE(68,*),'H/De'
   DIV = DE_FR
ELSE
   WRITE(68,*),'Height'
   DIV = 1.0
END IF

WRITE(68,*),'dL/dT'
WRITE(68,*),'4'
WRITE(68,*),'SUCKDOWN'
WRITE(68,*),'FOUNTAIN'
WRITE(68,*),'TOTAL'
WRITE(68,*),'Zero'
WRITE(68,1110) H
DO I = 1, H
   REC3 = LH*LV*(ID-1) + LH*(IV-1) + I
   READ(70,1109,REC=REC3) SF_LTS
   READ(71,1109,REC=REC3) SF_LTF
   SF_LT = SF_LTS + SF_LTF
   WRITE(68,1110) HT(I)/DIV, SF_LTS, SF_LTF,
      SF_LT, 0.0
END DO
GO TO 9999

WRITE(6,1210)
DO IH = 1, H
   WRITE(6,1205) IH, HT(IH)
END DO
WRITE(6,1310)
READ *, IH

WRITE(6,1220)
DO IV = 1, V
   WRITE(6,1205) IV, VO(IV)
END DO
WRITE(6,1320)
READ *, IV

WRITE(68,*),'1'
WRITE(68,1250) CONFIG,', STOLSF','HT=', HT(IH), ' VO=', VO(IV)
WRITE(68,*),'Jet Deflection Angle'
WRITE (68,*) 'dL/dT'
WRITE (68,*) '4'
WRITE (68,*) 'SUCKDOWN'
WRITE (68,*) 'FOUNTAIN'
WRITE (68,*) 'TOTAL'
WRITE (68,*) 'Zero'
WRITE (68,1105) D
DO I = 1, D
   REC3 = LH*LV*(I-1) + LH*(IV-1) + IH
   READ (70,1109,REC=REC3) SF_LTS
   READ (71,1109,REC=REC3) SF_LTF
   SF_LT = SF_LTS + SF_LTF
   WRITE (68,1110) DFL(I),SF_LTS, SF_LTF,
     SF_LT, 0.0
$END DO
GO TO 9999

C WRITE (6,1210)
DO IH = 1, H
   WRITE (6,1205) IH, HT(IH)
END DO
WRITE (6,1310)
READ *, IH

C WRITE (6,1230)
DO ID = 1, D
   WRITE (6,1205) ID, DFL(ID)
END DO
WRITE (6,1330)
READ *, ID

C WRITE (68,*) 'l'
WRITE (68,1250) CONFIG,', STOLSF',' HT=', HT(IH),'DFL=', DFL(ID)

C IF (VEOUT) THEN
   WRITE (68,*) 'Ve'
   DIV = SQRT(Q_FR/.00119/1.69**2)
ELSE
   WRITE (68,*) 'Aircraft Velocity'
   DIV = 1.0
END IF

C WRITE (68,*) 'dL/dT'
WRITE (68,*) '4'
WRITE (68,*) 'SUCKDOWN'
WRITE (68,*) 'FOUNTAIN'
WRITE (68,*) 'TOTAL'
WRITE (68,*) 'Zero'
WRITE (68,1105) V
DO I = 1, V
   REC3 = LH*LV*(ID-1) + LH*(I-1) + IH
   READ (70,1109,REC=REC3) SF_LTS
   READ (71,1109,REC=REC3) SF_LTF
   SF_LT = SF_LTS + SF_LTF
   WRITE (68,1110) VO(I)/DIV, SF_LTS, SF_LTF,
     SF_LT, 0.0
$END DO
GO TO 9999
Print-out for STOLGV (Ground vortex term)

WRITE(6,1300)
READ *, CHOICE

IF (CHOICE .LT. 1 .OR. CHOICE .GT. 4) THEN
    WRITE(6,*) '*** Invalid Choice ***'
    GO TO 300
END IF

GO TO (310, 320, 330, 340) CHOICE

Ground Vortex printout using Altitude as the X-Axis.

WRITE(6,1220)
DO IV = I, V
    WRITE(6,1205) IV, VO(IV)
END DO
WRITE(6,1320)
READ *, IV
WRITE (6, 1230)
DO ID = I, D
    WRITE(6, 1205)
END DO
WRITE (6, 1330)
READ*, ID

WRITE(6,1240)
DO IA = I, A
    WRITE(6,1205) IA,
END DO
READ*, IA

WRITE (68, *) 'I'
WRITE(68,1350) CONFIG,', STOLGV',' VO=',' DFL=',' AOA=',AOA(IA)

IF (HDEOUT) THEN
    WRITE(68,*) 'H/De'
    DIV = DE_FR
ELSE
    WRITE(68,*) 'Height'
    DIV = 1.0
END IF

WRITE (68,*) 'dL/dT'
WRITE (68, *) '4'
WRITE (68, *) 'Body'
WRITE (68, *) 'Wing'
WRITE (68, *) 'Both'
WRITE (68, *) 'Zero'
WRITE (68,1105) H
DO I = I, H
    REC4 = LH*LV*LD*(IA-I)+LH*LV*(ID-I)+LH*(IV-I)+I
    READ(72,1109,REC=REC4) GV_LT
    READ(73,1109,REC=REC4) GV_LTW
    GV_LT = GV_LTB + GV_LTW
    WRITE(68,1110) HT(I)/DIV,GV_LTB,
END DO
GO TO 9999

Ground Vortex printout using Velocity as the X-Axis.

WRITE(6,1210)
DO IH = 1, H
  WRITE(6,1205) IH, HT(IH)
END DO
WRITE(6,1310)
READ *, IH

GO TO 9999

DO ID = i, D
  WRITE (6, 1205)
END DO
WRITE (6, 1330)
READ*, ID
WRITE (6, 1240)
DO IA = I, A
  WRITE (6, 1205)
END DO
READ * , IA
WRITE (68,*) 'I'
WRITE (68, 1350)
  IA, AOA (IA)
CONFIG, ' , STOLGV', ' HT=', HT(IH), 'DFL=',
  DFL(ID), 'AOA=', AOA(IA)

IF (VEOUT) THEN
  WRITE(68,*) 'Ve'
  DIV = SQRT(Q_FR/.00119/1.69**2)
ELSE
  WRITE(68,*) 'Aircraft Velocity'
  DIV = 1.0
END IF

WRITE(68,*) 'dL/dT'
WRITE(68,*) '4'
WRITE(68,*) 'Body'
WRITE(68,*) 'Wing'
WRITE(68,*) 'Both'
WRITE(68,*) 'Zero'
WRITE(68,1105) V
DO I = 1, V
  REC4 = LH*LV*LD*(IA-1)+LH*LV*(ID-1)+LH*(I-1)+IH
  READ(72,1109,REC=REC4) GV_LTB
  READ(73,1109,REC=REC4) GV_LTW
  GV_LT = GV_LTB + GV_LTW
  WRITE(68,1110) VO(I)/DIV, GV_LTB,
  GV_LTW, GV_LT, 0.0
END DO
GO TO 9999

Ground Vortex printout using jet deflection angle as X-Axis.

WRITE(6,1210)
DO IH = 1, H
  WRITE(6,1205) IH, HT(IH)
END DO
WRITE (6, 1310)
READ *, IH
C
WRITE (6, 1220)
DO IV = 1, V
   WRITE (6, 1205) IV, VO(IV)
END DO
WRITE (6, 1320)
READ *, IV
C
WRITE (6, 1240)
DO IA = 1, A
   WRITE (6, 1205) IA, AOA(IA)
END DO
READ *, IA
C
WRITE (68, *) '1'
WRITE (68, 1350) CONFIG, ', STOLGV', ', HT=', HT(IH), ', VO=', VO(IV), ', AOA=', AOA(IA)
S
WRITE (68,*) 'Jet deflection angle (Deg)'
WRITE (68,*) 'dL/dT'
WRITE (68,*) '4'
WRITE (68,*) 'Body'
WRITE (68,*) 'Wing'
WRITE (68,*) 'Both'
WRITE (68,*) 'Zero'
WRITE (68, 1105) D
DO I = 1, D
   REC4 = LH*LV*LD*(IA-1) + LH*LV*(I-1) + LH*(IV-1) + IH
   READ(72, 1109, REC=REC4) GV_LTB
   READ(73, 1109, REC=REC4) GV_LTW
   GV_LT = GV_LTB + GV_LTW
   WRITE (68, 1110) DFL(I), GV_LTB,
                  GV_LTW, GV_LT, 0.0
END DO
GO TO 9999
C
Ground Vortex printout using angle of attack as the X-Axis.
340
WRITE (6, 1210)
DO IH = 1, H
   WRITE (6, 1205) IH, HT(IH)
END DO
WRITE (6, 1310)
READ *, IH
C
WRITE (6, 1220)
DO IV = 1, V
   WRITE (6, 1205) IV, VO(IV)
END DO
WRITE (6, 1320)
READ *, IV
C
WRITE (6, 1230)
DO ID = 1, D
   WRITE (6, 1205) ID, DFL(ID)
END DO
WRITE (6, 1330)
READ *, ID
WRITE (68, *) '1'
WRITE (68, 1350) CONFIG, ' STOLGV', ' HT=', HT(IH), ' VO=', VO(IV), 'DFL=', DFL(ID)
WRITE (68, *) 'AOA (DEG)'
WRITE (68, *) 'dL/dT'
WRITE (68, *) '4'
WRITE (68, *) 'Wing'
WRITE (68, *) 'Both'
WRITE (68, *) 'Zero'
WRITE (68, 1105) A
DO I = 1, A
  REC4 = LH*LV*LD*(I-1)+LH*LV*(ID-I)+LH*(IV-I)+IH
  READ (72, 1109, REC=REC4) GV_LTB
  READ (73, 1109, REC=REC4) GV_LTW
  GV_LT = GV_LTB + GV_LTW
  WRITE (68, 1110) AOA(I), GV_LTB,
                  GV_LTW, GV_LT, 0.0
END DO
GO TO 9999

RCS induced lift loss output
WRITE (68, *) '1'
WRITE (68, 1360) CONFIG, ', RCSIND'

IF (VEOUT) THEN
  WRITE (68, *) 'Ve'
  DIV = SQRT(Q_FR/.00119/1.69**2)
ELSE
  WRITE (68, *) 'Aircraft Velocity'
  DIV = 1.0
END IF

DO I = 1, V
  WRITE (68, 1110) VO(I)/DIV, RCS_LT(I, I), 0.0
END DO

IF (RCSFLG) THEN
  WRITE (6, 1400)
END IF
GO TO 9999

STOLWK table output
WRITE (6, 1500)
READ *, CHOICE

IF (CHOICE .LT. 1 .OR. CHOICE .GT. 3) THEN
  WRITE (6, *) '*** Invalid Choice ***'
  GO TO 500
END IF

GO TO (510, 520, 530) CHOICE
WRITE(6,1230)
DO ID = 1, D
   WRITE(6,1205) ID, DFL(ID)
END DO
WRITE(6,1330)
READ '*, ID

WRITE(6,1220)
DO IV = 1, V
   WRITE(6,1205) IV, VO(IV)
END DO
WRITE(6,1320)
READ '*, IV

WRITE(68,*) '1'
WRITE(68,1550) CONFIG, ', STOLWK', 'DFL=', DFL(ID), ' VO=', VO(IV)

IF (HDEOUT) THEN
   WRITE(68,*) 'H/De'
   DIV = DE_FR
ELSE
   WRITE(68,*) 'Height'
   DIV = 1.0
END IF

WRITE(68,*) 'dL/dT'
WRITE(68,*) '4'
WRITE(68,*) 'BODY'
WRITE(68,*) 'WING'
WRITE(68,*) 'TOTAL'
WRITE(68,*) 'Zero'
WRITE(68,1105) H
DO I = 1, H
   REC3 = LH*IV*(ID-1) + LH*(IV-1) + I
   READ(74,1109,REC=REC3) WK_LTB
   READ(75,1109,REC=REC3) WK_LTW
   WK_LT = WK_LTB + WK_LTW
   WRITE(68,1110) HT(I)/DIV, WK_LTB, WK_LTW,
   WK_LT, 0.0
END DO
GO TO 9999

WRITE(6,1210)
DO IH = 1, H
   WRITE(6,1205) IH, HT(IH)
END DO
WRITE(6,1310)
READ '*, IH

WRITE(6,1230)
DO ID = 1, D
   WRITE(6,1205) ID, DFL(ID)
END DO
WRITE(6,1330)
READ '*, ID

WRITE(68,*) '1'
WRITE (68, 1550) CONFIG, ', STOLWK', ' HT=', HT(IH), 'DL=', DFL(I)

C

IF (VEOUT) THEN
WRITE (68, *) 'Ve'
DIV = SQRT(Q_FR/.00119/1.69**2)
ELSE
WRITE (68, *) 'Aircraft Velocity'
DIV = 1.0
END IF

C

WRITE (68, *) 'dL/dT'
WRITE (68, *) '4'
WRITE (68, *) 'BODY'
WRITE (68, *) 'WING'
WRITE (68, *) 'TOTAL'
WRITE (68, *) 'Zero'
WRITE (68, 1105) V
DO I = 1, V
  REC3 = LH*L*(ID-1) + LH*(I-1) + IH
  READ (74, 1109, REC=REC3) WK_LTB
  READ (75, 1109, REC=REC3) WK_LTW
  WK_LT = WK_LTB + WK_LTW
  WRITE (68, 1110) VO(I)/DIV, WK_LTB, WK_LTW, WK_LT, 0.0
END DO
GO TO 9999

C

WRITE (6, 1210)
DO IH = 1, H
  WRITE (6, 1205) IH, HT(IH)
END DO
WRITE (6, 1310)
READ *, IH

C

WRITE (6, 1220)
DO IV = 1, V
  WRITE (6, 1205) IV, VO(IV)
END DO
WRITE (6, 1320)
READ *, IV

C

WRITE (68, *) '1'
WRITE (68, 1550) CONFIG, ', STOLWK', ' HT=', HT(IH), ' VO=', VO(IV)
WRITE (68, *) 'Jet Deflection Angle'
WRITE (68, *) 'dL/dT'
WRITE (68, *) '4'
WRITE (68, *) 'BODY'
WRITE (68, *) 'WING'
WRITE (68, *) 'TOTAL'
WRITE (68, *) 'Zero'
WRITE (68, 1105) D
DO I = 1, D
  REC3 = LH*L*(I-1) + LH*(IV-1) + IH
  READ (74, 1109, REC=REC3) WK_LTB
  READ (75, 1109, REC=REC3) WK_LTW
  WK_LT = WK_LTB + WK_LTW
  WRITE (68, 1110) DFL(I), WK_LTB, WK_LTW, WK_LT, 0.0
END DO
GO TO 9999

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

Totals table output

WRITE (6, 1600)
READ *, CHOICE

IF (CHOICE .LT. 1 .OR. CHOICE .GT. 4) THEN
    WRITE (6,*) '*** Invalid Choice ***'
    GO TO 600
END IF

GO TO (610, 620, 630, 640) CHOICE

WRITE (6, 1220)
DO IV = I, V
    WRITE (6, 1205)
END DO
WRITE (6, 1320)
READ *, IV

WRITE (6, 1230)
DO ID = I, D
    WRITE (6, 1205)
END DO
WRITE (6, 1330)
READ *, ID

WRITE (6, 1240)
DO IA = I, A
    WRITE (6, 1205)
END DO
WRITE (6, 1340)
READ *, IA

WRITE (68, *) 'i'
WRITE (68, 1650) CONFIG, ' , TOTALS' , '
VO=' , VO(IV), 'DFL=' , DFL(ID), 'AOA=' , AOA(IA)

IF (HDEOUT) THEN
    WRITE (68, *) 'H/De'
    DIV = DE.FR
ELSE
    WRITE (68, *) 'Height'
    DIV = 1.0
END IF

WRITE (68, *) 'dL/dT'
WRITE (68, *) '7'
WRITE (68, *) 'OGE'
WRITE (68, *) 'SF'
WRITE (68, *) 'G-VORTEX'
WRITE (68, *) 'JET WAKE'
WRITE (68, *) 'RCS'
WRITE (68, *) 'TOTAL'
WRITE (68, *) 'Zero'
WRITE (68, 1105) H
DO I = 1, H
    REC3 = LH*LV*(ID-1) + LH*(IV-1) + I
REC4 = LH*LV*LD*(IA-1)+LH*LV*(ID-1)+LH*(IV-1)+I
READ(70,1109,REC=REC3) SF_LTS
READ(71,1109,REC=REC3) SF_LTF
READ(72,1109,REC=REC4) GV_LTDB
READ(73,1109,REC=REC4) GV_LTW
READ(74,1109,REC=REC3) WK_LTDB
READ(75,1109,REC=REC3) WK_LTW
SF_LT = SF_LTS + SF_LTF
GV_LT = GV_LTDB + GV_LTW
WK_LT = WK_LTDB + WK_LTW
TOT = HLTOGE(ID) + SF_LT + GV_LT +
   WK_LT + RCS_LT(I,IV)
WRITE(68,1110) HT(I)/DIV, HLTOGE(ID), SF_LT,
$  GV_LT, WK_LT, RCS_LT(I,IV),
$  TOT, 0.0
END DO
GO TO 9999
WRITE(6,1210)
DO IH = 1, H
   WRITE(6,1205) IH, HT(IH)
END DO
WRITE(6,1310)
READ *, IH
WRITE(6,1230)
DO ID = 1, D
   WRITE(6,1205) ID, DFL(ID)
END DO
WRITE(6,1330)
READ *, ID
WRITE(6,1240)
DO IA = 1, A
   WRITE(6,1205) IA, AOA(IA)
END DO
WRITE(6,1340)
READ *, IA
WRITE(68,*) '1'
WRITE(68,1650) CONFIG, ', TOTALS', ' HT=', HT(IH), ' DFL=', DFL(ID),
   'AOA=', AOA(IA)
IF (VEOUT) THEN
   WRITE(68,*) 'Ve'
   DIV = SQRT(Q_FR/.00119/I.69**2)
ELSE
   WRITE(68,*) 'Aircraft Velocity'
   DIV = 1.0
END IF
WRITE(68,*) 'dL/dT'
WRITE(68,*) '7'
WRITE(68,*) 'OGE'
WRITE(68,*) 'SF'
WRITE(68,*) 'G-VORTEX'
WRITE(68,*) 'JET WAKE'
WRITE(68,*) 'RCS'

WRITE(68,*) 'TOTAL'
WRITE(68,*) 'Zero'
WRITE(68,1105) V
DO I = 1, V
   REC3 = LH*LV* (ID-I) + LH* (I-I) + IH
   REC4 = LH*LV*LD* (IA-I) + LH*LV* (ID-I) + LH* (I-I) + IH
   READ(70,1109,REC=REC3) SF LTS
   READ(71,1109,REC=REC3) SF LTF
   READ(72,1109,REC=REC4) GV_LT
   READ(73,1109,REC=REC4) GV_LTW
   READ(74,1109,REC=REC3) WK_LT
   READ(75,1109,REC=REC3) WK_LTW
   SF LT = SF LTS + SF LTF
   GV_LT = GV_LT + GV_LTW
   WK LT = WK_LT + WK_LTW
   TOT = HLTOGE(ID) + SF LT + GV_LT + WK LT + RCS_LT(I,I)
   WRITE(68,1110) VO (I)/DIV, HLTOGE(ID), SF LT,
               GV_LT, WK LT, RCS_LT(I,I),
               TOT, 0.0
END DO
GO TO 9999
END
DO IH = 1, H
   WRITE(6,1205) IH, HT(IH)
END DO
WRITE(6,1310) *
END
DO IV = I, V
   WRITE(6,1205) IV, VO(IV)
END DO
WRITE(6,1320) *
END
DO IA = I, A
   WRITE(6,1205) IA, AOA(IA)
END DO
WRITE(6,1340) *
END
WRITE(68,*) '1'
WRITE(68,1650) CONFIG, ', TOTALS', ' HT=', HT(IH), ' VO=', VO(IV),
               'AOA=', AOA(IA)
WRITE(68,*) 'Jet Deflection Angle'
WRITE(68,*) 'dL/dT'
WRITE(68,*) '?'
WRITE(68,*) 'OGE'
WRITE(68,*) 'SF'
WRITE(68,*) 'G-VORTEX'
WRITE(68,*) 'JET WAKE'
WRITE(68,*) 'RCS'
WRITE(68,*) 'TOTAL'
WRITE(68,*) 'Zero'
WRITE(68,1105)
DO I = 1, D
REC3 = LH*LV*(I-1) + LH*(IV-1) + IH
REC4 = LH*LV*LD*(IA-1) + LH*LV*(I-1) + LH*(IV-1) + IH
READ(70,1109,REC=REC3) SF_LTS
READ(71,1109,REC=REC3) SF_LTF
READ(72,1109,REC=REC4) GV_LTB
READ(73,1109,REC=REC4) GV_LTW
READ(74,1109,REC=REC3) WK_LTB
READ(75,1109,REC=REC3) WK_LTW
SF_LT = SF_LTS + SF_LTF
GV_LT = GV_LTB + GV_LTW
WK_LT = WK_LTB + WK_LTW
TOT = HLTOGE(I) + SF_LT + GV_LT +
     WK_LT + RCS_LT(I,IV) --
WRITE(68,1110) DFL(I), HLTOGE(I), SF_LT,
     GV_LT, WK_LT, RCS_LT(I,IV),
     TOT, 0.0
END DO
GO TO 9999

WRITE(6,1210)
DO IH = 1, H
    WRITE(6,1205) IH, HT(IH)
END DO
WRITE(6,1310)
READ *, IH

WRITE(6,1220)
DO IV = 1, V
    WRITE(6,1205) IV, VO(IV)
END DO
WRITE(6,1320)
READ *, IV

WRITE(6,1230)
DO ID = 1, D
    WRITE(6,1205) ID, DFL(ID)
END DO
WRITE(6,1320)
READ *, ID

WRITE(68,*)'1'
WRITE(68,1650) CONFIG,' TOTALS',' HT=', HT(IH), ' VO=', VO(IV),
     'DFL=', DFL(ID)
WRITE(68,*)'Aircraft Angle of Attack'
WRITE(68,*)'dL/dT'
WRITE(68,*)'7'
WRITE(68,*)'OGE'
WRITE(68,*)'SF'
WRITE(68,*)'G-VORTEX'
WRITE(68,*)'JET WAKE'
WRITE(68,*)'RCS'
WRITE(68,*)'TOTAL'
WRITE(68,*)'Zero'
WRITE(68,1105) A
DO I = 1, A
REC3 = LH*LV*(ID-1) + LH*(IV-1) + IH
REC4 = LH*LV*LD*(I-1) + LH*LV*(I-1) + LH*(IV-1) + IH
READ (70,1109,REC=REC3) SF_LTS
READ (71,1109,REC=REC3) SF_LTF
READ (72,1109,REC=REC4) GV_LTB
READ (73,1109,REC=REC4) GV_LTW
READ (74,1109,REC=REC3) WK_LTB
READ (75,1109,REC=REC3) WK_LTW
SF_LT = SF_LTS + SF_LTF
GV_LT = GV_LTB + GV_LTW
WK_LT = WK_LTB + WK_LTW
TOT = HLTGE(ID) + SF_LT + GV_LT +
     WK_LT + RCS_LT(I,IV)
WRITE (68,1110) HT(I)/DIV, HLTGE(ID), SF_LT,
     GV_LT, WK_LT, RCS_LT(I,IV),
     TOT, 0.0
END DO
GO TO 9999

C
JETIND table output
700
WRITE(6,1240)
DO IA = I, A
   WRITE (6,1205) IA, AOA (IA)
END DO
WRITE (6, 1320)
READ *, IA

C
CALL JIETAB(IA)
CLOSE (68)
GO TO 20
END IF

1100 FORMAT(15X,'OUTPUT DATA TO TABLE FORMAT'
$ //5X,'0 - Quit table generation section'
$ //5X,'1 - Output table with all hover results (dL/dT vs Height)'$
$ //5X,'2 - Tables with results from STOLSF (any)'$
$ //5X,'3 - Tables with results from STOLGV (any)'$
$ //5X,'4 - Table with results from RCSIND'
$ //5X,'5 - Table with results from STOLWK (any)'$
$ //5X,'6 - Table with totals from all routines'
$ //5X,'7 - Table for JETIND'
$ //5X,'Choose an option')
1105 FORMAT (1X,I4)
1109 FORMAT (F11.5)
1110 FORMAT (1X,F8.4,3X,14(F11.5,2X))

C
1200 FORMAT(1X,'* * * * * * * * * * * * * * * * * * * * * * '
$ /15X,'STOLSF Table generation section (Suckdown/Fountain)',$
$ //5X,'1 - Output table based on height,'
$ //5X,'2 - Output table based on aircraft velocity'
$ //5X,'3 - Output table based on jet deflection angle',$
$ //5X,'Choose an option')
1205 FORMAT (5X,I3,' --> ',F6.1)
1210 FORMAT (3X,'H Number',4X,'Height')
1220 FORMAT (3X,'V Number',4X,'Velocity')
1230 FORMAT (3X,'D Number',4X,'Def Angle')
1240 FORMAT (3X,'A Number',4X,'Angle of Attack')
1250 FORMAT (1X,A18,A8,2(2X,A4,F3.0))
FUNCTION DIST(X1,Y1,X2,Y2)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THIS FUNCTION CALCULATES THE DISTANCE BETWEEN THE TWO GIVEN
C COORDINATES IN THE (X,Y) PLANE.
C
GIVEN: TWO POINTS (X1, Y1), (X2, Y2)
FIND: THE DISTANCE BETWEEN THE TWO POINTS
C
REAL BLEEP, DIST, X1, Y1, X2, Y2
C
BLEEP = (X2-X1)**2 + (Y2-Y1)**2
DIST = SQRT(BLEEP)
C
RETURN
END
**Perpdis**

SUBROUTINE PERPDIS(X, Y, MLINE, BLINE, dist)

C THIS SUBROUTINE CALCULATES THE DISTANCE FROM A POINT TO A LINE IN A
C PARTICULAR DIRECTION DEFINED BY A SLOPE.

REAL MLINE, a,b,c,x,y, dist

A = -MLINE
B = 1.0
C = -BLINE

CALCULATE PERPENDICULAR DISTANCE BETWEEN POINT AND LINE
DIST = (A * X + B * Y + C)/SQRT(A**2 + B)

RETURN
END

**Linterp**

SUBROUTINE LINTERP(X, XI, X2, YI, Y2, Y)

PURPOSE: THIS SUBROUTINE PREFORMS LINEAR INTERPOLATION.
GIVEN: TWO POINTS (XI,YI), (X2,Y2)
FIND: A VALUE (Y) THAT CORRESPONDS TO THE VALUE (X) THAT LIES
ALONG THE LINE DEFINED BY THE TWO POINTS.

REAL X, XI, X2, Y, YI, Y2

Y = YI + (Y2-YI)/(X2-XI)*(X-XI)

RETURN
END

**Crossin**

SUBROUTINE CROSSIN(X1, Y1, X2, Y2, X3, Y3, X4, Y4, x, y)

C THIS SUBROUTINE CALCULATES THE INTERSECTION OF TWO LINES, EACH LINE
C DEFINED BY TWO POINTS. EACH OF THE LINES END-POINTS LIE ON THE SAME
C X-COORDINATE

GIVEN: FOUR POINTS THAT DEFINE TWO CROSSING LINES (AS SHOWN)

```
  1
 /|
X /|
  /|
  |
  |
X 3
  |
  |
  |
  |
  X 2
```
SUBROUTINE LINECRO(SLOPE1, B1, SLOPE2, B2, x, y)

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

THIS SUBROUTINE CALCULATES THE INTERSECTION OF TWO LINES

GIVEN: SLOPE1 = SLOPE OF THE FIRST LINE
       B1 = THE Y-INTERCEPT OF THE FIRST LINE
       SLOPE2 = SLOPE OF THE SECOND LINE
       B2 = THE Y-INTERCEPT OF THE SECOND LINE

Y = SLOPE2 * (B2 - B1) / (SLOPE1 - SLOPE2) + B2

IF (SLOPE1 .EQ. 0.0) THEN
   X = (Y - B2) / SLOPE2
ELSE
   X = (Y - B1) / SLOPE1
END IF

RETURN
END

FIND: THE INTERSECTION OF THE TWO LINES (x, y)
SUBROUTINE POLYAR

ACRONYM: POLYgon Area

PURPOSE: The purpose of POLYAR is to calculate the area of a polygon by defined a number of points.

LOCAL VARIABLES (in addition to the above parameters):

NAME   TYPE I/O UNITS DESCRIPTION
======== ====== ========= =---------------------------------------------------------------------
I       I      ----     General purpose counter
SUM     R I     ----     Portion added to total
SUB     R I     ----     Portion subtracted from total

GLOBAL VARIABLES (in addition to the above parameters and local vars):

NAME   TYPE I/O UNITS DESCRIPTION
======== ====== ========= =---------------------------------------------------------------------
POINTS  I I     ----     Number of points (not to exceed 500)
XPTS(500) R I     ----     X-coordinates of polygon
YPTS(500) R I     ----     Y-coordinates of polygon
TOTAL   R I     ----     Total area enclosed by the polygon.

FILES USED:
LOGICAL UNIT I/O DESCRIPTION
--------------- -------- ---------------------------
(none)

COMMONS USED:
NAME DESCRIPTION
-------------- ------------------------------------------------------------------------
POLYGON Contains the points used in POLYAR when calculating the area of a polygon and CENTAR when calculating the center of area.

CALLED BY:
NAME DESCRIPTION
-------------- ------------------------------------------------------------------------
FINVAR Calculates all variables which have not been defined by the user or set to a default value
CENTAR Calculates the center of area, area and area in front of a point for a given planform

Routines Called:
NAME DESCRIPTION
-------------- ------------------------------------------------------------------------
(none)

NOTES: None.

REFERENCES:
1) None.

ENVIRONMENT:
FORTRAN 77, VAX 6800, VAX 11/785, SGI IRIS 4D series.

NON-STANDARD CODE:
None.

AUTHOR(S):
Kipp E. Howard (KEH), Cal Poly San Luis Obispo, NASA Ames Res Cntr

REVISION HISTORY:
DATE INITIALS & DESCRIPTION
05/15/90 KEH -- Initial completion of code.
07/14/91 KEH -- Eliminated extraneous spacing and fixed comments

#include "misc.inc"
C Variable declarations
REAL SUB, SUM
INTEGER I

C
SUM = 0.0
SUB = 0.0
DO 10 I = 1, POINTS
C
IF (I .EQ. POINTS) THEN
SUM = SUM + XPTS(I) * YPTS(1)
SUB = SUB + YPTS(I) * XPTS(1)
ELSE
SUM = SUM + XPTS(I) * YPTS(I + 1)
SUB = SUB + YPTS(I) * XPTS(I + 1)
END IF

C 10 CONTINUE
TOTAL = ABS(.5 * (SUM - SUB))
C
RETURN
END

---

**Centar**

SUBROUTINE CENTAR (MODE, XSTART, X, Y, N, OUT)
---

C ACRONYM: CENTER of Area
C
C PURPOSE: The purpose of CENTAR is to calculate a planform center of area based on the planform defined by X and Y.
C
C PARAMETERS:

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODE</td>
<td>I</td>
<td>I</td>
<td></td>
<td>Flag which tells this routine which values to calculate</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 - Calculate Center of Area</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 - Calculate Area of planform</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3 - Calculate area in front of XSTART</td>
</tr>
<tr>
<td>N</td>
<td>I</td>
<td>I</td>
<td></td>
<td>Number of points in planform</td>
</tr>
<tr>
<td>OUT</td>
<td>R</td>
<td>O</td>
<td></td>
<td>Value to be returned to calling routine. Based on MODE.</td>
</tr>
<tr>
<td>X(500)</td>
<td>R</td>
<td>I</td>
<td></td>
<td>X-coordinate of planform</td>
</tr>
<tr>
<td>XSTART</td>
<td>R</td>
<td>I</td>
<td></td>
<td>Point from which to calculate the area forward of this point</td>
</tr>
<tr>
<td>Y(500)</td>
<td>R</td>
<td>I</td>
<td></td>
<td>Y-coordinate of planform</td>
</tr>
</tbody>
</table>

C LOCAL VARIABLES (in addition to the above parameters):

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALEFT</td>
<td>R</td>
<td></td>
<td></td>
<td>Left side Area difference for bisection method</td>
</tr>
<tr>
<td>ARIGHT</td>
<td>R</td>
<td></td>
<td></td>
<td>Right side Area difference for bisection method</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td></td>
<td></td>
<td>General purpose counter</td>
</tr>
<tr>
<td>TEST</td>
<td>R</td>
<td></td>
<td></td>
<td>Test Area difference for bisection method</td>
</tr>
<tr>
<td>XCA</td>
<td>R</td>
<td>I</td>
<td></td>
<td>X-coordinate of Centr of Area</td>
</tr>
<tr>
<td>XLEFT</td>
<td>R</td>
<td></td>
<td></td>
<td>Left side X-coordinate for</td>
</tr>
</tbody>
</table>
Bisection method
Right side X-coordinate for bisection method.
Y-coordinate of on planform that is directly above the guess of the center of area.

GLOBAL VARIABLES (in addition to the above parameters and local vars):

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>POINTS</td>
<td>I</td>
<td>I</td>
<td>----</td>
<td>Number of points (not to exceed 500)</td>
</tr>
<tr>
<td>XPTS(500)</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td>X-coordinates of polygon</td>
</tr>
<tr>
<td>YPTS(500)</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td>Y-coordinates of polygon</td>
</tr>
</tbody>
</table>

FILES USED:
LOGICAL UNIT I/O DESCRIPTION

(none)

COMMONS USED:

NAME DESCRIPTION

POLYGON Contains the points used in POLYAR when calculating the area of a polygon and CENTAR when calculating the center of area.

CALLED BY:

NAME DESCRIPTION

FINVAR Calculates all variables which have not been defined by the user or set to a default value

ROUTINES CALLED:

NAME DESCRIPTION

LINTERP Linear interpolates between two X and Y values give an intermediate X value

PERPDIS Calculates the perpendicular distance between a point and a line

POLYAR Calculates the area of any polygon

NOTES: None.

REFERENCES:
1) None.

ENVIRONMENT:

FORTRAN 77, VAX 6800, VAX 11/785, SGI IRIS 4D series.

NON-STANDARD CODE:

None.

AUTHOR(S):

Kipp E. Howard (KEH), Cal Poly San Luis Obispo, NASA Ames Moffett

REVISION HISTORY:

DATE INITIALS & DESCRIPTION
04/01/90 KEH -- Code completed and in working order
07/14/91 KEH -- Eliminated extraneous spacing and fixed comments

---
#include "misc.inc"

C Variable declarations
REAL X_CA, X(500), Y(500)
INTEGER I, N, FIRST

C Calculate the geometric center of the planform.
GEOCTR = (X(N) + X(1)) / 2.0
C Initialize X_CA based on the value of XSTART
   IF (XSTART .EQ. 9999.) THEN
     X_CA = GEOCTR
   ELSE
     X_CA = XSTART
   END IF

C Start calculating areas
   DO 20 I = 2, N
   IF (X_CA .LE. X(I) .AND. X_CA .GT. X(I-1)) THEN
     CALL LINTERP(X_CA,X(I-1),X(I),Y(I-1),Y(I),YPT)
     DO 5 J = 1, I-1
        XPTS(J) = X(J)
        YPTS(J) = Y(J)
     CONTINUE
     XPTS(I) = X_CA
     YPTS(I) = YPT
     XPTS(I+1) = X_CA
     YPTS(I+1) = 0.0
     POINTS = I+1
     Calculate front area
     CALL POLYAR
     FAREA = TOTAL
   END IF

C Set up rear half of planform to calculate its area.
   XPTS(1) = X_CA
   YPTS(1) = 0.0
   XPTS(2) = X_CA
   YPTS(2) = YPT
   DO 10 K = I, N
      J = K - I + 3
      XPTS(J) = X(K)
      YPTS(J) = Y(K)
   CONTINUE
   POINTS = J
   Calculate rear area
   CALL POLYAR
   RAREA = TOTAL
   END IF

20 CONTINUE

C Initialize left and right side for bisection method
   IF (FIRST .EQ. 1) THEN
     TAREA = RAREA + FAREA
     ALEFT = 0.0 - TAREA
     XLEFT = X(1)
     ARIGHT = TAREA - 0.0
     XRIGHT = X(N)
     FIRST = 2
   END IF

C Set OUT to the correct value according to the value of MODE.
   IF (MODE .EQ. 2) THEN
     OUT = TAREA
     GO TO 60
ELSE IF (MODE .EQ. 3) THEN
   OUT = FAREA
   GO TO 60
END IF

TEST = FAREA - RAREA

Determine which side TEST is on (Right or Left).
IF (TEST .LT. 0.0) THEN
   ALEFT = TEST
   XLEFT = X_CA
ELSE
   ARIGHT = TEST
   XRIGHT = X_CA
END IF

Check to see if Areas are within .5% of each other
(for Center of area calculations)
IF (ABS(TEST/(TAREA/2.0)) .LT. .005) GO TO 50
X_CA = (XLEFT + XRIGHT)/2.0

Return to calculate new FAREA and RAREA based on the new X_CA
GO TO 1

IF (MODE .EQ. 1) OUT = X_CA

RETURN

END

SUBROUTINE SSEN(START, STEP, END, NUM, DSTART, DEND, DNUM, FLAG)

ACRONYM: Start, Step, End, Number calculator

PURPOSE: The purpose of SSEN is to calculate any of the variables that define the number list which are not defined, using default values when not enough information is given.

PARAMETERS:

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEND</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Default value for ending value</td>
</tr>
<tr>
<td>DNUM</td>
<td>I</td>
<td>I</td>
<td>-----</td>
<td>Default value for number of points</td>
</tr>
<tr>
<td>DSTART</td>
<td>I</td>
<td>I</td>
<td>-----</td>
<td>Default value for starting value</td>
</tr>
<tr>
<td>END</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Ending value of number list</td>
</tr>
<tr>
<td>FLAG</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Flag to which values are set when the values are not defined. (When setting NUM as undefined use the integer equivalent of FLAG.)</td>
</tr>
<tr>
<td>NUM</td>
<td>I</td>
<td>I</td>
<td>-----</td>
<td>Number of points in number list</td>
</tr>
<tr>
<td>START</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Starting value of number list</td>
</tr>
<tr>
<td>STEP</td>
<td>R</td>
<td>I</td>
<td>-----</td>
<td>Stepping value of number list</td>
</tr>
</tbody>
</table>

LOCAL VARIABLES (in addition to the above parameters):

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>(none)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

GLOBAL VARIABLES (in addition to the above parameters and local vars):

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
C (none)
C FILES USED:
C LOGICAL UNIT I/O DESCRIPTION
---------- ----------------
(none)
C COMMINS USED:
C NAME DESCRIPTION
---------- ----------------
(none)
C CALLED BY:
C NAME DESCRIPTION
---------- ----------------
FINVAR Calculates all variables which have not been defined by
the user or set to a default value
C ROUTINES CALLED:
C NAME DESCRIPTION
---------- ----------------
(none)
C NOTES: None.
C REFERENCES:
C 1) None.
C ENVIRONMENT:
C FORTRAN 77, VAX 6800, VAX 11/785, SGI IRIS 4D series.
C NON-STANDARD CODE:
C None.
C AUTHOR(S):
C Kipp E. Howard (KEH), Cal Poly San Luis Obispo, NASA Ames Res Cntr
C REVISION HISTORY:
C DATE INITIALS & DESCRIPTION
C 07/18/90 KEH -- Code completed and in working order
C 07/14/91 KEH -- Eliminated extraneous spacing and fixed comments
C Variable Declaration
REAL START, STEP, END, DSTART, DEND
INTEGER DNUM, NUM
C IF (START .NE. FLAG) THEN
C IF (END .NE. FLAG) THEN
C IF (STEP .NE. FLAG) THEN
C NUM = INT((END - START) / STEP + 1.)
C ELSE IF (NUM .NE. INT(FLAG)) THEN
C IF (NUM .EQ. 1) THEN
C STEP = 0.0
C ELSE
C STEP = (END - START) / (NUM - 1)
C END IF
C ELSE
C STEP = (END - START) / (NUM - 1)
ELSE IF (STEP .NE. FLAG) THEN
    IF (NUM .NE. FLAG) THEN
        END = START + STEP * (NUM - 1)
    ELSE
        END = DEND
        NUM = INT((END - START) / STEP + 1.)
    END IF
ELSE IF (NUM .NE. FLAG) THEN
    END = DEND
ELSE IF (NUM .EQ. 1) THEN
    STEP = 0.0
ELSE
    STEP = (END - START) / (NUM - 1)
END IF
ELSE
    END = DEND
    NUM = DNUM
ELSE IF (NUM .EQ. 1) THEN
    STEP = 0.0
ELSE
    STEP = (END - START) / (NUM - 1)
END IF
ELSE IF (END .NE. FLAG) THEN
    IF (STEP .NE. FLAG) THEN
        IF (NUM .NE. INT(FLAG)) THEN
            START = END - STEP * (NUM - 1)
        ELSE
            START = DSTART
            NUM = INT((END - START) / STEP + 1.)
        END IF
    ELSE IF (NUM .NE. DNUM) THEN
        START = DSTART
        STEP = (END - START) / (NUM - 1)
    ELSE
        START = DSTART
        NUM = DNUM
        STEP = (END - START) / (NUM - 1)
    END IF
ELSE IF (STEP .NE. DSTART) THEN
    IF (NUM .NE. FLAG) THEN
        START = DSTART
    ELSE
        START = DSTART
        NUM = DNUM
        STEP = (END - START) / (NUM - 1)
    END IF
END = START + STEP * (NUM - 1)
ELSE
START = DSTART
END = DEND
NUM = INT((END - START) / STEP + 1.)
END IF

ELSE IF (NUM .NE. FLAG) THEN
START = DSTART
END = DEND
STEP = (END - START) / (NUM - 1)
ELSE
START = DSTART
END = DEND
NUM = DNUM
STEP = (END - START) / (NUM - 1)
END IF

RETURN
END

SUBROUTINE JIETAB(L)

Jietab

ACRONYM: Jet Induced Effect TABle generator - Example table follows:

# of Heights # of Velocities # of Deflection Angles
--> DFL1 Vel1 Vel12 ..... VelN
    Ht1 LT11 LT12 ..... L1IN
    Ht2 LT21 LT22 ..... Lt2N
    :     :     :       :
    :     :     :       :
    HtM LTM1 LTM2 ..... LTMN
--> DFL2 Vel1 Vel12 ..... VelN
    Ht1 LT11 LT12 ..... L1IN
    :     :     :       :
    etc. etc. etc. etc.

Purpose: The purpose of JIETAP is to generate the table which JETIND
uses to determine the lift loss.

Parameters:

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>I</td>
<td>I</td>
<td></td>
<td>Angle of attack which table is to be generated</td>
</tr>
</tbody>
</table>

Local Variables (in addition to the above parameters): |

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>I</td>
<td>I</td>
<td></td>
<td>Height counter</td>
</tr>
<tr>
<td>J</td>
<td>I</td>
<td>I</td>
<td></td>
<td>Velocity counter</td>
</tr>
<tr>
<td>K</td>
<td>I</td>
<td>I</td>
<td></td>
<td>Deflection angle counter</td>
</tr>
<tr>
<td>DUMMY(100)</td>
<td>I</td>
<td>I</td>
<td></td>
<td>Dummy array variable to hold total results so routine can output values in an implied do loop.</td>
</tr>
</tbody>
</table>

Global Variables (in addition to the above parameters and local vars):

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
</table>
RESPIE Common Block

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>I/O</th>
<th>UNITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFL(500)</td>
<td>R</td>
<td>I</td>
<td>deg</td>
<td>Actual nozzle deflection angle value</td>
</tr>
<tr>
<td>HT(500)</td>
<td>R</td>
<td>O</td>
<td>Ft</td>
<td>Actual height value</td>
</tr>
<tr>
<td>LD</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td>Total number of nozzle deflection angles.</td>
</tr>
<tr>
<td>LH</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td>Total number of height values.</td>
</tr>
<tr>
<td>LV</td>
<td>R</td>
<td>I</td>
<td>----</td>
<td>Total number of velocity values.</td>
</tr>
<tr>
<td>VO(500)</td>
<td>R</td>
<td>O</td>
<td>----</td>
<td>Actual velocity value</td>
</tr>
</tbody>
</table>

FILES USED:

<table>
<thead>
<tr>
<th>LOGICAL UNIT</th>
<th>I/O</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>68</td>
<td>O</td>
<td>Sequential file for table output</td>
</tr>
</tbody>
</table>

COMMONS USED:

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESPIE</td>
<td>Results from all POWER Induced Effects subroutines - Contains results from each subroutine along with variables for height, velocity, jet deflection angle, and angle of attack and their counters.</td>
</tr>
</tbody>
</table>

CALLED BY:

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>JIETAB</td>
<td>Creates output file which is used by ACSYNT</td>
</tr>
</tbody>
</table>

ROUTINES CALLED:

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTOTAL</td>
<td>Sums the total lift increment given array type indices</td>
</tr>
</tbody>
</table>

NOTES: None.

REFERENCES:

1) None.

ENVIRONMENT:

FORTRAN 77, VAX 6800, VAX 11/785, SGI IRIS 4D series.

NON-STANDARD CODE:

None.

AUTHOR(S):

Kipp E. Howard (KEH), Cal Poly San Luis Obispo, NASA Ames Res Cntr

REVISION HISTORY:

<table>
<thead>
<tr>
<th>DATE</th>
<th>INITIALS &amp; DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>08/07/90</td>
<td>KEH -- Code completed and in working order</td>
</tr>
<tr>
<td>07/14/91</td>
<td>KEH -- Eliminated extraneous spacing and fixed comments</td>
</tr>
</tbody>
</table>

#include "respie.inc"

REAL RTOTAL, DUMMY(100)
INTEGER I,J,K,L

Output first line of table (Number of Altitudes, Number of Velocities, Number of Nozzle Deflections)

WRITE(68,*) LH, LV, LD

Start Loops

Deflection LoopC
DO K = 1, LD
WRITE(68,100) DFL(K), (VO(J), J = 1, LV)

Altitude Loop
DO I = 1, LH
   Dummy variable has to be used because, for some reason, a
   function with READ statements, within an implied do loop
   function, will cause the WRITE statement to output nothing. A
   function within an implied do loop works fine as long as
   there are no READ statements.

Velocity Loop
DO J = 1, LV
   DUMMY(J) = RTOTAL(I, J, K, L)
END DO
WRITE(68,100) HT(I), (DUMMY(J), J = 1, LV)

END DO
END DO

100 FORMAT(1X,101(G11.5,2X))
RETURN
END

REAL FUNCTION RTOTAL(I, J, K, L)

C

#include "respie.inc"

C

I - Altitude variable
J - Velocity Variable
K - Deflection Variable
L - Angle of Attack Variables

INTEGER I, J, K, L

REC3 = LH*LV*(K-1) + LH*(J-1) + I
REC4 = LH*LV*LD*(L-1) + LH*LV*(K-1) + LH*(J-1) + I
READ(70,100,REC=REC3) SF_LTS
READ(71,100,REC=REC3) SF_LTF
READ(72,100,REC=REC4) GV_LTB
READ(73,100,REC=REC4) GV_LTW
READ(74,100,REC=REC3) WK_LTB
READ(75,100,REC=REC3) WK_LTW
SF_LT = SF_LTS + SF_LTF
GV_LT = GV_LTB + GV_LTW
WK_LT = WK_LTB + WK_LTW
RTOTAL = HLTGEO(J) + SF_LT + GV_LT + WK_LT + RCS_LT(1, J)
RTOTAL = RTOTAL

C

100 FORMAT (F11.5)
RETURN
END
APPENDIX B

Power Induced Effects Module User's/Programmer's Manuals
Power Induced Effects Module
User's/Programmer's Manuals

by

Kipp Edward Howard

July 1991
# Table of Contents

User's Guide ............................................................................................................... 1  
Description ............................................................................................................... 1  
   Philosophy ........................................................................................................... 1  
   Control .................................................................................................................. 1  
Input .......................................................................................................................... 3  
Preliminary Calculations .......................................................................................... 3  
Lift Increments Routines ......................................................................................... 4  
   Height ................................................................................................................... 4  
   Forward Velocity .................................................................................................. 4  
   Height, Forward Velocity, and Jet Deflection Angle ........................................... 4  
   Height, Forward Velocity, Jet Deflection Angle, and Angle of Attack .................. 5  
Output ....................................................................................................................... 5  
Limitations ............................................................................................................... 6  
Variables ................................................................................................................ 6  
Definitions ............................................................................................................... 6  
Minimum Configuration Variables ............................................................................ 7  
   Necessary Group .................................................................................................. 24  
   Thrust Group ....................................................................................................... 24  
   Nozzle Definition Group ...................................................................................... 25  
   Pressure Ratio Group ......................................................................................... 25  
   RCS Group ......................................................................................................... 25  
   LIDs Group ......................................................................................................... 26  
   Potential Group ................................................................................................... 26  
Output ....................................................................................................................... 27  
Example .................................................................................................................... 27  
   Configuration ....................................................................................................... 27  
   Input ...................................................................................................................... 27  
   Execution .............................................................................................................. 29  
   Output ................................................................................................................... 34  
Programmer's Guide ............................................................................................... 40  
Overall Scheme & Assumptions .............................................................................. 40  
Equations for Subroutines ..................................................................................... 40  
Hover ......................................................................................................................... 40  
   Suckdown ............................................................................................................ 40  
   Fountain Lift ....................................................................................................... 40  
      Basic Method ................................................................................................... 41  
      h' Method ....................................................................................................... 41  
      Lift Improvement Devices ............................................................................... 43  
   Forward Flight / Jet Deflection Angle / Angle of Attack ..................................... 45  
   Suckdown/Fountain ............................................................................................. 45  
   Ground Vortex ..................................................................................................... 45  
   Jet Wake ............................................................................................................... 46  
   Reaction Control System .................................................................................... 47
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routine Descriptions</td>
<td>48</td>
</tr>
<tr>
<td>Control</td>
<td>48</td>
</tr>
<tr>
<td>Preliminary Routines</td>
<td>49</td>
</tr>
<tr>
<td>INPIE</td>
<td>49</td>
</tr>
<tr>
<td>PLANMO</td>
<td>49</td>
</tr>
<tr>
<td>FINVAR</td>
<td>49</td>
</tr>
<tr>
<td>SDAREA</td>
<td>50</td>
</tr>
<tr>
<td>DIABAR</td>
<td>50</td>
</tr>
<tr>
<td>Hover</td>
<td>50</td>
</tr>
<tr>
<td>HCALL</td>
<td>51</td>
</tr>
<tr>
<td>HOV_GE</td>
<td>51</td>
</tr>
<tr>
<td>Suckdown/Fountain</td>
<td>51</td>
</tr>
<tr>
<td>SFCALL</td>
<td>51</td>
</tr>
<tr>
<td>STOLSF</td>
<td>51</td>
</tr>
<tr>
<td>Ground Vortex</td>
<td>51</td>
</tr>
<tr>
<td>GVCALL</td>
<td>51</td>
</tr>
<tr>
<td>STOLGV</td>
<td>51</td>
</tr>
<tr>
<td>Jet Wake</td>
<td>52</td>
</tr>
<tr>
<td>WKCALL</td>
<td>52</td>
</tr>
<tr>
<td>JIEPIE</td>
<td>52</td>
</tr>
<tr>
<td>STOLWK</td>
<td>52</td>
</tr>
<tr>
<td>Reaction Control System</td>
<td>52</td>
</tr>
<tr>
<td>RCSCAL</td>
<td>52</td>
</tr>
<tr>
<td>RCSIND</td>
<td>53</td>
</tr>
<tr>
<td>Output</td>
<td>53</td>
</tr>
<tr>
<td>PIEP</td>
<td>53</td>
</tr>
<tr>
<td>JIETAB</td>
<td>53</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>53</td>
</tr>
<tr>
<td>CENTAR</td>
<td>53</td>
</tr>
<tr>
<td>CROSSIN</td>
<td>53</td>
</tr>
<tr>
<td>DIST</td>
<td>53</td>
</tr>
<tr>
<td>INTEGRA</td>
<td>53</td>
</tr>
<tr>
<td>LINECRO</td>
<td>54</td>
</tr>
<tr>
<td>LINTERP</td>
<td>54</td>
</tr>
<tr>
<td>PERPDIS</td>
<td>54</td>
</tr>
<tr>
<td>POLYAR</td>
<td>54</td>
</tr>
<tr>
<td>RTOTAL</td>
<td>54</td>
</tr>
<tr>
<td>SSEN</td>
<td>54</td>
</tr>
<tr>
<td>References</td>
<td>55</td>
</tr>
<tr>
<td>Appendices</td>
<td>56</td>
</tr>
<tr>
<td>Routine Summary</td>
<td>56</td>
</tr>
<tr>
<td>PIE Hierarchy Diagram</td>
<td>58</td>
</tr>
<tr>
<td>Cross Reference - Common Blocks</td>
<td>59</td>
</tr>
<tr>
<td>Cross Reference - Calls To</td>
<td>60</td>
</tr>
<tr>
<td>Cross Reference - Calls By</td>
<td>61</td>
</tr>
</tbody>
</table>
User's Guide

This manual will provide the user of the Power Induced Effects Module with enough information to implement PIE and provide the programmer with adequate information to support and/or enhance the PIE code.

This work was performed by Kipp Edward Howard for his Masters thesis at Cal Poly San Luis Obispo as suggested by the STOVL/Powered-Lift Technology Branch (FAP), NASA Ames Research Center.

Description

The basic purpose of this project was to produce an easy to use, and modify computer code which estimated lift increments due to power induced effects of V/STOL aircraft, in hover and forward flight, using methods developed by Kuhn and Stewart (References 3 and 6).

Philosophy

The PIE FORTRAN code is written in a modular structure which makes it easy to understand and modify. The code is divided into five fundamental sections; control, input, pre-calculations, lift increments, and output. Each of these modules are divided into individual parts to make logic and coding easier to understand.

Control

The Power Induced Effects (PIE) module is controlled by the control program, COPPIE, which coordinates the execution of PIE. COPPIE’s first task is to make calls to input and pre-calculation subroutines. These calls are made first so all variables are either defined by the user or assigned a default value. Within these variables the number, limits, and steps are defined for each independent variable; height, velocity, nozzle deflection angle, and angle of attack. COPPIE steps from one to the maximum number of variations for each independent variable. (i.e. If 15 heights are to be considered then COPPIE steps from one to 15 for height.) The actual value for the independent variable is calculated and then the control routine for a lift increment(s) is called and/or the next independent variable is incremented. After all independent variables have been incremented up to their maximum values then the output subroutine is called. Figure #1 is a flow chart which shows the basic structure of COPPIE.
Figure #1 Flow chart of the control program COPPIE. (Note - Flow is downward unless specified.)
Each lift increment routine has its own control routine. This routine checks all configuration variables and passes only those variables which the lift increment routines require. This control routine isolates configuration variables from changes made inside lift increment routines and makes the code easier to understand. Further descriptions of these routines are provided in the Lift Increment Routines section.

**Input**

Input routines for PIE perform two functions; input of variables, and check/modification of variables for consistency.

The first routine of the input routines reads two input files. The first file contains all user defined variables. The second file contains X and Y coordinates of half the configuration planform. This routine first sets default values for all variables and then reads user defined variables from the first file. The advantage of this style of input over the input in the original programs (variables are entered in the code, hard-coded, Reference 4) is it allows the user to input as many or as few variables (above the minimum configuration variables) as the user desires. It also allows the user to save the configuration in a small, easy to manipulate files and rerun particular configurations as desired. The planform file (config.pie) allows a majority of calculation-intensive variables to be calculated by PIE, which alleviates the user from making tedious calculations.

The second routine assures planforms defined in the second file (config.pie) are consistent with the coordinate system defined in PIE. If the nose of the configuration does not lie on the origin, the routines modify all coordinates of the planforms and necessary configuration variables so the nose does lie on the origin. This makes sure PIE is able to interpret user inputs correctly.

**Preliminary Calculations**

A desirable feature of the Power Induced Effects Module (PIE) is it contains a complete set of variables which defines the aircraft. This allows lift increment control routines to choose variables needed by lift increment routines. Many pre-calculations need to be made to obtain this complete set of variables. These pre-calculations take a number of hours by hand for each configuration and they change with each new configuration.

Some of these preliminary calculations were very extensive, such as calculations for fountain areas (S_FA1, SP_FA2, etc...) and dbar, D, (DB_B, DB_W, etc...). Many other preliminary calculations were very simple calculations such as nozzle areas (S_F, S_FR) and nozzle distances (X_FR, Y_R, etc...) but the number of these simple calculations is large.
Lift Increments Routines

The lift increment routines were divided into four sections. The sections are based on which independent variables are used to calculate the lift increment. The first section is lift increments calculated for hover. These lift increments vary with height. The second section contains lift increments which vary with velocity. The third section contains lift increments which vary with height, velocity, and jet deflection angle. The fourth section contains lift increments which change with height, velocity, jet deflection angle, and angle of attack.

Height

The hover routines calculate initial lift increments for the suckdown, and fountain which vary with height. They are later adjusted for forward velocity and jet deflection angle. The RCS, ground vortex, and jet wake lift increments occur when a freestream is present so they are not included in these calculations. The out-of-ground effect (OGE) suckdown is calculated first and then corrections are made for in-ground effect (GE) based on the altitude of the configuration. The fountain terms are calculated based on their proximity to the ground using the "Basic" and "h'" methods outlined by Kuhn (Reference 6).

The specific function of the hover control routine is to recognize when a high wing is present and make appropriate adjustment to variables which it sends to the lift increment routine.

Forward Velocity

The reaction control system (RCS) lift increment varies with velocity. Ground effects can be neglected since the size of the RCS jets are much smaller and their height above the ground, relative to their diameter, is much greater than the main lifting jets. Corrections for jet placement are calculated first. The lift increment based on pressure ratio is calculated next and adjustment factors are applied to the results.

The specific function of the RCS control routine is to either execute the lift increment routine or set its value to zero. This is controlled with a flag variable which can be changed by the user.

Height, Forward Velocity, and Jet Deflection Angle

Suckdown, fountain, and jet wake lift increments vary with height, forward velocity and jet deflection angle. Suckdown and fountain terms are calculated inside the same lift increment routine because they both change similarly with the independent variables. A height factor is calculated based on the forward extent of the wall jet. Then the calculations use hover values multiplied by height, forward flight and nozzle deflection angle factors. The control routine for suckdown and fountain routine track the positions of the nozzles and whether there is a high wing and passes the correct information to the suckdown and fountain lift increment routine.
The jet wake terms are calculated using methods described by Stewart (Reference 13). Since the methods makes separate calculations for different planforms and for front and rear jets, the control program is much more detailed than the others. The control program first determines which kind of planforms are to be used; wing and body or wing/body (one planform which is a combination of the wing and body.) It checks if the nozzles are near the trailing edge of the wing, for the jet flap effect, and then calculates the OGE and overall lift increment for the jet wake from front and rear nozzles. This process is repeated for the next planform. The OGE and overall lift increments are calculated in separate routines because of the complexity of each calculation. The calculations for the wing and body are stored separately to allow contributions of each planform to be observed.

**Height, Forward Velocity, Jet Deflection Angle, and Angle of Attack**

The ground vortex term is calculated based on height, forward velocity, nozzle deflection angle, and angle of attack. This is the only term which varies with angle of attack.

The control routine for the ground vortex term is similar to the jet wake control routine with respect to the planforms. The rear nozzles do not contribute to the ground vortex so only the front nozzles are used in the calculations. Lift increments for the wing and body are calculated and stored separately to allow contributions of each planform to be observed.

**Output**

Storage of the results of PIE are in files and arrays which can be recalled with the use of an index notation. This is done so any type of lookup table can be created by the user/programmer. This method of storage allows the programmer to configure an output table in any format using simple index notation when requesting a value. (i.e. If the user wants a lift increment for the eighth height, sixth velocity, third deflection angle and fifth angle of attack, it can be recalled by setting the appropriate index variables; i, j, k, and l to 8, 6, 3, and 5 respectively.) This feature allows easy access to the lift increments which facilitates many different types of lookup table output.

Currently, PIE creates one three-dimensional lookup table for ACSYNT. This table contains multiple two-dimensional tables based on deflection angle with each individual table based on height and velocity. This table is generated by a single, short subroutine which demonstrates how many other types of tables can be generated using similar subroutines.

Other output options are created for viewing purposes. These tables are examples of the many tables which can be generated. Each lift increment as well as the total lift increments can be viewed based on two of their four independent variables. Currently, outputs are text files which can be read by a plotting routine, developed at NASA Ames for the Silicon Graphics IRIS workstations, and a Macintosh graphing application called KaleidaGraph™.
**Limitations**

PIE cannot calculate lift increments for configurations with five or more jets or configurations with three or more jets in a straight line either laterally or longitudinally. A temporary solution to this problem is to combined closely spaced jets into single jets, which can be rectangular or oval, until the total number of jets is reduced to four or less.

The methods used in this code are intended for use in preliminary design analysis and to give an indication of the effects of changes to primary configuration variables. Power induced effects are a complex function of many configuration variables and development of a V/STOL aircraft will require careful experimental investigations to accurately determine the induced forces.

**Variables**

**Definitions**

Below are names, default values, units and descriptions for all variables which can be modified. They are divided into sections to make it easier to find a particular variable. These sections are Control Variables, Body Variables, Center Section Variables, Wing Variables, Wingbody Variables, Forward Nozzle Variables, Rear Nozzle Variables, Front/Rear Nozzle Variables, RCS Variables, Fountain Variables, and Miscellaneous Variables.

Most of the variable names were created to enable them to be easily recognized and understood. All variables consist of six characters or less. The majority of them consist of a logical description of the variable along with a description as to what the variable applies, divided by a ".". An example is the variable S_WB, which is the planform area of the wingbody. Some of the other variables were created based on the same variables from other programs (Kuhn, Reference 4).

The Default column contains values for the variables which have default values. Any variable which does not have a default value will contain either "Calculated" or a "*" possibly followed by a letter. The variables with "Calculated" in the Default column signify the variable will be calculated based on other variables unless it is defined by the user. The variables with a "*" only are the minimum necessary input variables, these need to be defined. The variables with "*" followed by a letter are also minimum input variables except they have special considerations given in the Minimum Configuration Variables section. Variables with the same letters are related to one another. All variables which need to be calculated are initially set to a flag value which is 9999.0. This is the only value which a variable cannot be set. The only variables which need to be watched are thrust variables (T_F, T_R, T_FR) which can be 9999.0 for particular configurations.
The Type column contains a one character description of the type of variable which is used. The character “R” refers to a real value, “I” refers to an integer value, “T” refers to a character value, and “L” refers to a logical value.

The Units column contains English units of the variable. The “----” indicates the variable is dimensionless.

The Description of Variable column contains a short description along with any special considerations needed when using the particular variable.

<table>
<thead>
<tr>
<th>Name</th>
<th>Default</th>
<th>Type</th>
<th>Units</th>
<th>Description of Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control Variables</td>
<td></td>
<td></td>
<td></td>
<td>Contains all variables which deal with changing of the independent variables. The number of independent variables are restricted to the amount of memory available from the computer system. There are no error flags when a mismatch between the start, end, step and number variables. Therefore care must be taken to make sure they are consistent. It is possible to mismatch these variables and still have correct output but the start, end, or step values might not be the same as they were input. An example of this is shown in the velocity calculations (VSTEP).</td>
</tr>
<tr>
<td>AEND</td>
<td>30.0</td>
<td>R</td>
<td>deg</td>
<td>Ending value for angle of attack (AOA) - AOA is the angle between the nose of the aircraft and free stream (nose up is positive). This value is typically not greater then 30° to 45°. It could also be less than ASTART. If this variable is set to 9999.0 and ASTART, ASTEP and LA are defined then AEND will be calculated.</td>
</tr>
<tr>
<td>ASTART</td>
<td>0.0</td>
<td>R</td>
<td>deg</td>
<td>Starting value for angle of attack - See AEND for definition of AOA. This value is typically around 0° but it could be greater than AEND. If this variable is set to 9999.0 and AEND, ASTEP, and LA are defined then ASTART will be calculated.</td>
</tr>
<tr>
<td>ASTEP</td>
<td>Calculated</td>
<td>R</td>
<td>deg</td>
<td>Step value for angle of attack - See AEND for definition of AOA. When defining this variable, just to be safe, make sure steps are equally divisible by the difference between AEND and ASTART. This value is already defaulted.</td>
</tr>
<tr>
<td>Name</td>
<td>Default</td>
<td>Type</td>
<td>Units</td>
<td>Description of Variable</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>------</td>
<td>-------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>LA</td>
<td>4</td>
<td>I</td>
<td></td>
<td>Total number of Angle of Attack values - This value is typically small (&lt;10) because only lift increments which change with angle of attack are those for the ground vortex. It is the least important of the four independent variables. If this variable is set to 9999 and AEND, ASTART, and ASTEP are defined then LA will be calculated. It is possible to set this value to 1 and set AEND and ASTART equal so only one angle of attack is used.</td>
</tr>
<tr>
<td>DEND</td>
<td>30.0</td>
<td>R</td>
<td>deg</td>
<td>Ending value for jet deflection angle (DFL) - DFL is the angle between the nozzles and body of the aircraft (0° is pointing aft (forward flight), and 90° is pointing (down)). This value can be anywhere between 0° and 90°. Similar to AEND when calculating this variable.</td>
</tr>
<tr>
<td>DSTART</td>
<td>90.0</td>
<td>R</td>
<td>deg</td>
<td>Starting value for jet deflection angle - See DEND for definition of DFL. Similar to ASTART when calculating this variable.</td>
</tr>
<tr>
<td>DSTEP</td>
<td>Calculated</td>
<td>R</td>
<td>deg</td>
<td>Step value for jet deflection angle - See DEND for definition of DFL. Similar to ASTEP when calculating this variable.</td>
</tr>
<tr>
<td>LD</td>
<td>7</td>
<td>I</td>
<td></td>
<td>Total number of nozzle deflection angles. Similar to LA when calculating this variable. This values is typically smaller compared to height and velocity independent variables.</td>
</tr>
<tr>
<td>HEND</td>
<td>20.0*DE_FR</td>
<td>R</td>
<td>Ft</td>
<td>Ending value for altitude (HT) - HT is the altitude the bottom of the aircraft is above the surface of the ground. This value must be above HSTART. The default value was chosen because at altitudes higher than 20 effective diameters of the nozzle, the lift increments do not change significantly. Similar to AEND when calculating this variable.</td>
</tr>
<tr>
<td>HSTART</td>
<td>2.0*DE_FR</td>
<td>R</td>
<td>Ft</td>
<td>Starting value for altitude - See HEND for definition of HT. This variable must be below HEND. The default value was chosen because the inaccuracy of the</td>
</tr>
<tr>
<td>Name</td>
<td>Default</td>
<td>Type</td>
<td>Units</td>
<td>Description of Variable</td>
</tr>
<tr>
<td>---------</td>
<td>-----------</td>
<td>------</td>
<td>-------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>HSTEP</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>Step value for altitude - See HEND for definition of HT. It is desirable to have small steps between HSTART and HEND so as to distinguish various changes in lift increments. Similar to ASTEP when calculating this variable.</td>
</tr>
<tr>
<td>LH</td>
<td>25</td>
<td>I</td>
<td></td>
<td>Total number of height values. - See HEND for definition of HT. Similar to LA when calculating this variable. This independent variable is one of the two most important independent variables.</td>
</tr>
<tr>
<td>VEND</td>
<td>100.0</td>
<td>R</td>
<td>kts</td>
<td>Ending value for aircraft velocity (VO) - VO is the forward velocity of the aircraft. This value need not be greater then VSTART. The default value was chosen because it was a typical velocity which an aircraft might achieve wingbourn flight. Similar to AEND when calculating this variable.</td>
</tr>
<tr>
<td>VSTART</td>
<td>1.0</td>
<td>R</td>
<td>kts</td>
<td>Starting value for aircraft velocity - See VEND for definition of VO. The value is typically low but could be higher than VEND. Similar to ASTART when calculating this variable.</td>
</tr>
<tr>
<td>VSTEP</td>
<td>Calculated</td>
<td>R</td>
<td>kts</td>
<td>Step value for aircraft velocity - See VEND for definition of VO. The calculated value will cause the ending value to be 101 which is different than the input for VEND. Similar to ASTEP when calculating this variable.</td>
</tr>
<tr>
<td>LV</td>
<td>20</td>
<td>I</td>
<td></td>
<td>Total number of velocity values - See VEND for definition of VO. Similar to LA when calculating this variable. This independent variable is one of the two most important independent variables.</td>
</tr>
</tbody>
</table>

**Body Variables**

Body variables are determined if a body configuration is used as an input in the configuration file (config.PIE). The body planform is always input with a wing planform and never input with a wingbody planform.

<p>| B_B     | Calculated | R    | Ft    | Width of Body planform - B_B is defined as maximum thickness of the body |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>Default</th>
<th>Type</th>
<th>Units</th>
<th>Description of Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB_B</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>Dbar of Body planform - Angular mean diameter. This is the diameter of a circle which has the same perimeter length as the given planform calculated using the equation below. The diagram of a general planform is shown below. Dbar changes with placement of the point of measure. The point of measure (PM) for the body planform is the point equal distance between the front and rear nozzles or if there is only one nozzle the point of measure is at the front nozzles along the X-axis.</td>
</tr>
<tr>
<td>L_B</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>Length of body planform - L_B is calculated by the maximum distance from the nose of the body planform.</td>
</tr>
<tr>
<td>N_BODY</td>
<td>Calculated</td>
<td>I</td>
<td>---</td>
<td>Number of data points for Body planform - PIE counts the number of points used in the body planform definition. There is no reason to input this variable.</td>
</tr>
<tr>
<td>R_B</td>
<td>9999.</td>
<td>R</td>
<td>Ft</td>
<td>Corner radius of body sides - R_B is used in calculating the body contour factor (KR). It is defined as the radius of curvature of the sides of the body</td>
</tr>
<tr>
<td>S_B</td>
<td>Calculated</td>
<td>R</td>
<td>Ft²</td>
<td>Area of Body - S_B is the total planform area of body. Calculated from planform input in the *.pie file.</td>
</tr>
<tr>
<td>WL_B</td>
<td>Calculated</td>
<td>R</td>
<td>---</td>
<td>Width to Length ratio of Body - WL_B is used in calculating lift increments for the ground vortex and the jet wake.</td>
</tr>
<tr>
<td>Name</td>
<td>Default</td>
<td>Type</td>
<td>Units</td>
<td>Description of Variable</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------</td>
<td>-------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>X_BODY(500)</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>X-coordinates of Body planform - Coordinate which defines X-coordinate of body planform. Coordinates are to start at the nose (0.0, not necessary) and progress toward the tail with increasing coordinate values. These values are obtained from the *.pie file.</td>
</tr>
<tr>
<td>Y_BODY(500)</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>Y-coordinates of Body planform - Coordinate which defines Y-coordinate of body planform. Coordinates are to start at the center line of the body and increase in value toward the right (as seen by the pilot). These values are obtained from the *.pie file.</td>
</tr>
<tr>
<td>Z_B</td>
<td>0.0</td>
<td>R</td>
<td>Ft</td>
<td>Height of body base above nozzle - This value can be either positive or negative. A negative value denoting a body base below the nozzle. This variable is used in calculating lift increments for the ground vortex and the jet wake.</td>
</tr>
<tr>
<td>Wing Variables</td>
<td></td>
<td></td>
<td></td>
<td>Wing variables are determined if a wing configuration is used as an input in the configuration file (*.PIE). The wing planform is always input with a body planform and never input with a wingbody planform.</td>
</tr>
<tr>
<td>B_W</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>Width of Wing (Wing span) - B_W is defined as the maximum thickness of the wing planform.</td>
</tr>
<tr>
<td>DB_W</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>Dbar of Wing - Angular mean diameter of wing planform. See DB_B for a more detailed description.</td>
</tr>
<tr>
<td>MAC</td>
<td>*</td>
<td>R</td>
<td>Ft</td>
<td>Mean Aerodynamic Chord of wing</td>
</tr>
<tr>
<td>N_WING</td>
<td>Calculated</td>
<td>I</td>
<td></td>
<td>Number of data points for Wing planform - PIE counts the number of points used in the wing planform definition. There should be no reason to input this variable.</td>
</tr>
<tr>
<td>S_W</td>
<td>Calculated</td>
<td>R</td>
<td>Ft²</td>
<td>Area of Wing - S_W is the total planform area of wing. Calculated from planform input in the *.pie file.</td>
</tr>
<tr>
<td>X_WING(500)</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>X-coordinates of Wing planform - Coordinate which defines the X-coordinate of wing planform. The coordinates are to be placed on the same coordinate axis as the body and progress</td>
</tr>
</tbody>
</table>
### Description of Variable

- **Y_WING(500)**: Calculated **R Ft**
  - Y-coordinates of Wing planform - Coordinate which defines Y-coordinates of the wing planform. The coordinates for the leading edge of the wing are to start at the center line of the wing and have a positive value toward the right (as seen by the pilot). These values are obtained from *.pie file.

- **Z_W**
  - Height of wing above nozzle - This value can be either positive or negative. A negative value denoting a wing below the nozzle (unusual). This variable is used in calculating lift increments for hover, ground vortex and jet wake.

### Wingbody Variables

- **B_WB**: Calculated **R Ft**
  - Width of Wing-Body - B_WB is defined as the maximum thickness of the wingbody planform.

- **DB_WB**: Calculated **R Ft**
  - Dbar of Wing-Body - Angular mean diameter of wing planform. See DB_B for a more detailed description.

- **L_WB**: Calculated **R Ft**
  - Length of Wing-Body - L_WB is calculated by the maximum distance from the nose of the wingbody planform.

- **N_WB**: Calculated **I**
  - Number of data points for Wing-Body planform - PIE counts the number of points used in the wing-body planform definition. There should be no reason to input this variable.

- **S_WB**: Calculated **R Ft^2**
  - Area of Wing-Body planform - S_WB is the total planform area of wing-body. Calculated from planform input in the *.pie file or from planform made from wing and body planform.
<table>
<thead>
<tr>
<th>Name</th>
<th>Default</th>
<th>Type</th>
<th>Units</th>
<th>Description of Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>WL_WB</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>Width to Length ratio of Wing-Body - WL_WB is used in calculating lift increments for the ground vortex and the jet wake.</td>
</tr>
<tr>
<td>X_WB(500)</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>X-coordinates of Wing-Body planform - Coordinate which defines X-coordinates of wing-body planform. The coordinates are to be placed on the same coordinate axis as the body and progress toward the tail with increasing coordinate values. These values are obtained from *.pie file or determined by PIE given the wing and body planforms.</td>
</tr>
<tr>
<td>Y_WB(500)</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>Y-coordinates of Wing-Body planform - Coordinate which defines the Y-coordinate of the wing-body planform. The coordinates are to start at the center line of the wing-body and have a positive value toward the right (as seen by the pilot). These values are obtained from the *.pie file or determined by PIE given the wing and body planforms.</td>
</tr>
</tbody>
</table>

**Center Section Variables**

Center section variables are determined if a wing and body configuration are used as an input in the configuration file (config.PIE). The center section variables are created from a combination of the wing and body planforms.

<table>
<thead>
<tr>
<th>Name</th>
<th>Default</th>
<th>Type</th>
<th>Units</th>
<th>Description of Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_CS</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>Width of Center Section - B_CS is defined as the maximum thickness of the wingbody planform. Calculated when wing and body planforms are input. Used in calculating lift increments for the jet wake.</td>
</tr>
<tr>
<td>B_JP</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>Width of Jet Pattern - Width of pattern defined by the placements of the jets (each jet is a corner). Used in calculating WL_JP.</td>
</tr>
<tr>
<td>DB_CS</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>Dbar of Center Section - Angular mean diameter of center section. See DB_B for a more detailed description.</td>
</tr>
<tr>
<td>L_CS</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>Length of Center Section - L_CS is calculated by the maximum distance from the front to the rear of the center section. Used in calculating WL_JP.</td>
</tr>
</tbody>
</table>
### Name | Default | Type | Units | Description of Variable
--- | --- | --- | --- | ---
PP_LID | •E | R | ---- | Ratio of perimeter enclosed by lift improvement devices (LIDS) to total perimeter of LIDS. See Kuhn (Ref 1)
S_CS | Calculated | R | Ft² | Area of Center Section - S_CS is the total planform area of the center section. Calculated when a wing and body planform are input. Used in calculating lift increments for the ground vortex and jet wake.
S_JP | Calculated | R | Ft² | Area enclosed by Jet Pattern - S_JP is the total planform area inside the jet pattern defined by the nozzles. Used in calculating lift increments for the hover fountain.
S_LID | •E | R | Ft² | Area enclosed by LIDS - S_LID is the total planform area inside the lift improvements devices. Used in calculating lift increments for the fountain.
SP_JP | •E | R | Ft² | Actual surface area within area enclosed by nozzles - SP_JP can be less than or equal than S_JP but never greater. Used in calculating lift increments for the fountain core and LIDS.
WL_CS | Calculated | R | ---- | Width to Length ratio of Center Section - Not used in any current calculations
WL_JP | Calculated | R | ---- | Width to Length ratio of Jet Pattern - WL_JP is used in calculating the lift increments for the hover fountain.

**Forward Nozzle Variables**

Forward nozzle variables are always defined. If there is only one set of nozzles, they will be defined at the forward nozzles. Variables for the nozzles are input using the *.dat files.

D_F | •B | R | Ft | Diameter of each Front jet - D_F is the diameter of an individual front circular or oval nozzle. If there are two nozzles then input only the diameter of one of them. If the nozzle is oval then input the diameter which would give a circle the same area as the oval. If the nozzle is rectangular then only input the length and width.

DE_F | Calculated | R | Ft | Effective Diameter of Front jets - DE_F is the effective diameter of the front jets combined. This is the diameter of a single circle which would give the same area as the total area of the front nozzles.
<table>
<thead>
<tr>
<th>Name</th>
<th>Default</th>
<th>Type</th>
<th>Units</th>
<th>Description of Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>L_F</td>
<td>•B</td>
<td>R</td>
<td>Ft</td>
<td>Length of Front jet - L_F is the length in the Y-direction of the front jet(s). Input this and W_F only if the nozzles are rectangular, not D_F.</td>
</tr>
<tr>
<td>NUM_F</td>
<td>Calculated</td>
<td>I</td>
<td></td>
<td>NUMber of Front jets - NUM_F is the total number of front jets present. This number cannot be greater than two. This is calculated by examining the location of the jets.</td>
</tr>
<tr>
<td>PR_F</td>
<td>•C</td>
<td>R</td>
<td></td>
<td>Jet Pressure Ratio for front jets - PR_F used to calculate the lift increments for the hover suckdown and the jet wake. Also used to calculated the dynamic pressure of the front nozzles.</td>
</tr>
<tr>
<td>Q_F</td>
<td>Calculated</td>
<td>R</td>
<td>Ib/ft²</td>
<td>Dynamic pressure for the front jets - Q_F is used to calculate the lift increments for the jet wake.</td>
</tr>
<tr>
<td>S_F</td>
<td>Calculated</td>
<td>R</td>
<td>ft²</td>
<td>Area of Front jets - Total area of front jets</td>
</tr>
<tr>
<td>SF_FB</td>
<td>Calculated</td>
<td>R</td>
<td>ft²</td>
<td>Area ahead of Front jets using body planform - Total area of body planform ahead of the front jets. Used to calculate the lift increments for the jet wake.</td>
</tr>
<tr>
<td>SF_FW</td>
<td>Calculated</td>
<td>R</td>
<td>ft²</td>
<td>Area ahead of Front jets using wing planform - Total area of wing planform ahead of the front jets. Used to calculate the lift increments for the jet wake.</td>
</tr>
<tr>
<td>SF_FWB</td>
<td>Calculated</td>
<td>R</td>
<td>ft²</td>
<td>Area ahead of Front jets using the wingbody planform - Total area of wingbody planform ahead of the front jets. Used to calculate the lift increments for the jet wake.</td>
</tr>
<tr>
<td>SPLY_F</td>
<td>0.0</td>
<td>R</td>
<td>Ft</td>
<td>SPLAY angle of Front jet - SPLY_F is the angle from the vertical which the nozzles are canted. A positive splay angle is one which is away from the center of the aircraft.</td>
</tr>
<tr>
<td>T_F</td>
<td>•A</td>
<td>R</td>
<td>lb</td>
<td>Thrust of Front jets - Total thrust of all front jets. (Make sure this value is not 9999.0)</td>
</tr>
<tr>
<td>Name</td>
<td>Default</td>
<td>Type</td>
<td>Units</td>
<td>Description of Variable</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>------</td>
<td>-------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>TT_F</td>
<td>A</td>
<td>R</td>
<td>----</td>
<td>Front Thrust / total Thrust</td>
</tr>
<tr>
<td>W_F</td>
<td>B</td>
<td>R</td>
<td>Ft</td>
<td>Width of Front jet - W_F is the width in the X-direction of the front jet(s). Input this and L_F only if the nozzles are rectangular, not D_F.</td>
</tr>
<tr>
<td>WL_F</td>
<td>Calculated</td>
<td>R</td>
<td>----</td>
<td>Width to Length ratio of Front jets</td>
</tr>
<tr>
<td>XCG_F</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>Distance Front jet is ahead of CG - XCG_F is the distance between the center of gravity (CG) and the front nozzles. A positive value denotes that the nozzles are ahead of the CG.</td>
</tr>
<tr>
<td>XNOZ_F</td>
<td>•</td>
<td>R</td>
<td>Ft</td>
<td>X-coordinates of Front NOZZle - Position of the front nozzles along the X-axis.</td>
</tr>
<tr>
<td>XTE_F</td>
<td>•</td>
<td>R</td>
<td>Ft</td>
<td>X-coordinate of trailing edge for front nozzle - Position along the X-axis of the wing trailing edge (TE) directly behind or in front of the front nozzle(s). Use wing root TE if nozzle is within the body planform. (This value can be different from the XTE_R). Use in calculating the lift increment due to the jet flap condition for the jet wake. (This is not the distance between the front nozzle and the trailing edge, but the X-coordinate of the trailing edge behind the front nozzle)</td>
</tr>
<tr>
<td>Y_F</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>Distance between Front jets - Lateral distance (Y-direction) between the front nozzles.</td>
</tr>
<tr>
<td>YB_F</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>Lateral distance from Body to Front jets - YB_F is the lateral distance (Y-direction) from the side of the body to the nozzles. An external nozzle will have a positive YB_F.</td>
</tr>
<tr>
<td>YNOZ_F</td>
<td>•</td>
<td>R</td>
<td>Ft</td>
<td>Y-coordinates of Front NOZZle - Position of the front nozzles along the Y-axis. Only one value for the front nozzles need to be entered due to the symmetry of the configurations. If YNOZ_F is 0.0 then only one nozzle is assumed. If YNOZ_F is not equal to zero then, due to symmetry, two nozzles are assumed.</td>
</tr>
<tr>
<td>YWB_F</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>Lateral distance from Wingbody to Front jets - YWB_F is the lateral distance (Y-direction) from the side of the wing-body to the nozzles. An external nozzle will have a positive YWB_F.</td>
</tr>
</tbody>
</table>
Rear nozzle variables are defined only if there are front and rear nozzles. The variables for the nozzles are input using the *.dat files.

- **D_R**
  - **Description**: Diameter of each Rear jet - D_R is the diameter of an individual rear circular or oval nozzle. If there are two nozzles then input only the diameter of one of them. If the nozzle is oval then input the diameter which would give a circle the same area as the oval. If the nozzle is rectangular then only input the length and width.
  - **Type**: Calculated
  - **Units**: R Ft

- **DE_R**
  - **Description**: Effective Diameter of Rear jets - DE_R is the effective diameter of the rear jets combined. This is the diameter of a single circle which would give the same area as the total area of the rear nozzles.
  - **Type**: Calculated
  - **Units**: R Ft

- **L_R**
  - **Description**: Length of Rear jet - L_R is the length in the Y-direction of the rear jet(s). Input this and W_R only if the nozzles are rectangular, not D_R.
  - **Type**: Calculated
  - **Units**: R Ft

- **NUM_R**
  - **Description**: Number of Rear jets - NUM_R is the total number of rear jets present. This number cannot be greater than two. This is calculated by examining the location of the jets.
  - **Type**: Calculated
  - **Units**: I

- **PR_R**
  - **Description**: Jet Pressure Ratio for rear jets - PR_R used to calculate the lift increments for the hover suckdown and the jet wake. Also used to calculate the dynamic pressure of the rear nozzles.
  - **Type**: Calculated
  - **Units**: R

- **Q_R**
  - **Description**: Dynamic pressure for the rear jets - Q_R used to calculate the lift increments for the jet wake.
  - **Type**: Calculated
  - **Units**: lb/Ft²

- **S_R**
  - **Description**: Area of Rear jets - Total area of the rear jets.
  - **Type**: Calculated
  - **Units**: R Ft²

- **SF_RB**
  - **Description**: Area ahead of Rear jets using body planform - Total area of body planform ahead of the rear jets. Used to calculate the lift increments for the jet wake.
  - **Type**: Calculated
  - **Units**: R Ft²

- **SF_RW**
  - **Description**: Area ahead of Rear jets using wing planform - Total area of wing planform ahead of the rear jets. Used to calculate the lift increments for the jet wake.
  - **Type**: Calculated
  - **Units**: R Ft²

- **SF_RWB**
  - **Description**: Area ahead of Rear jets using the wingbody planform - Total area of wingbody planform ahead of the rear jets. Used to calculate the lift increments for the jet wake.
  - **Type**: Calculated
  - **Units**: R Ft²
<table>
<thead>
<tr>
<th>Name</th>
<th>Default</th>
<th>Type</th>
<th>Units</th>
<th>Description of Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPLY_R</td>
<td>0.0</td>
<td>R</td>
<td>Ft</td>
<td>SPLaY angle of Rear jet - SPLY_R is the angle from the vertical which the nozzles are canted. A positive splay angle is one which is away from the center of the aircraft. See diagram for SPLY_F.</td>
</tr>
<tr>
<td>T_R</td>
<td>•A</td>
<td>R</td>
<td>lb</td>
<td>Thrust of Rear jets - Total thrust of all rear jets. (Make sure this value is not 9999.0)</td>
</tr>
<tr>
<td>TT_R</td>
<td>•A</td>
<td>R</td>
<td>----</td>
<td>Rear Thrust / total Thrust</td>
</tr>
<tr>
<td>W_R</td>
<td>•B</td>
<td>R</td>
<td>Ft</td>
<td>Width of Rear jet - W_R is the width in the X-direction of the rear jet(s). Input this and L_R only if the nozzles are rectangular, not D_R.</td>
</tr>
<tr>
<td>WL_R</td>
<td>Calculated</td>
<td>R</td>
<td>----</td>
<td>Width to Length ratio of Rear jets</td>
</tr>
<tr>
<td>XCG_R</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>Distance Rear jet is ahead of CG - XCG_R is the distance between the center of gravity (CG) and the rear nozzles. A negative value denotes that the nozzles are behind the CG.</td>
</tr>
<tr>
<td>XNOZ_R</td>
<td></td>
<td>R</td>
<td>Ft</td>
<td>X-coordinates of Rear NOZzle - Position of the rear nozzles along the X-axis.</td>
</tr>
<tr>
<td>XTE_R</td>
<td></td>
<td>R</td>
<td>Ft</td>
<td>X-coordinate of trailing edge for rear nozzle - Position along the X-axis of the wing trailing edge (TE) directly behind or in front of the rear nozzle(s). Use wing root TE if Nozzle is within the body planform. (This value can be different from the XTE_F). Use in calculating the lift increment due to the jet flap condition for the jet wake. (This is not the distance between the rear nozzle and the trailing edge, but the X-coordinate of the trailing edge behind or in front of the rear nozzle)</td>
</tr>
<tr>
<td>Y_R</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>Distance between Rear jets - Lateral distance (Y-direction) between the rear nozzles.</td>
</tr>
<tr>
<td>YB_R</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>Lateral distance from Body to Rear jets - YB_R is the lateral distance (Y-direction) from the side of the body to the nozzles. An external nozzle will have a positive YB_R.</td>
</tr>
<tr>
<td>YNOZ_R</td>
<td></td>
<td>R</td>
<td>Ft</td>
<td>Y-coordinates of Rear NOZzle - Position of the rear nozzles along the Y-axis. Only one value for the rear nozzles need to be</td>
</tr>
</tbody>
</table>

Description of Variable:

- **SPLaY angle of Rear jet** - SPLY_R is the angle from the vertical which the nozzles are canted. A positive splay angle is one which is away from the center of the aircraft. See diagram for SPLY_F.
- **Thrust of Rear jets** - Total thrust of all rear jets. (Make sure this value is not 9999.0)
- **Rear Thrust / total Thrust**
- **Width of Rear jet** - W_R is the width in the X-direction of the rear jet(s). Input this and L_R only if the nozzles are rectangular, not D_R.
- **Width to Length ratio of Rear jets**
- **Distance Rear jet is ahead of CG** - XCG_R is the distance between the center of gravity (CG) and the rear nozzles. A negative value denotes that the nozzles are behind the CG.
- **X-coordinates of Rear NOZzle** - Position of the rear nozzles along the X-axis.
- **X-coordinate of trailing edge for rear nozzle** - Position along the X-axis of the wing trailing edge (TE) directly behind or in front of the rear nozzle(s). Use wing root TE if Nozzle is within the body planform. (This value can be different from the XTE_F). Use in calculating the lift increment due to the jet flap condition for the jet wake. (This is not the distance between the rear nozzle and the trailing edge, but the X-coordinate of the trailing edge behind or in front of the rear nozzle)
- **Distance between Rear jets** - Lateral distance (Y-direction) between the rear nozzles.
- **Lateral distance from Body to Rear jets** - YB_R is the lateral distance (Y-direction) from the side of the body to the nozzles. An external nozzle will have a positive YB_R.
- **Y-coordinates of Rear NOZzle** - Position of the rear nozzles along the Y-axis. Only one value for the rear nozzles need to be
<table>
<thead>
<tr>
<th>Name</th>
<th>Default Type</th>
<th>Units</th>
<th>Description of Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>YWB_R</td>
<td>Calculated</td>
<td>R Ft</td>
<td>Lateral distance from Wingbody to Rear jets - YWB_R is the lateral distance (Y-direction) from the side of the wing-body to the nozzles. An external nozzle will have a positive YWB_R.</td>
</tr>
<tr>
<td>Front/Rear Nozzle Variables</td>
<td></td>
<td></td>
<td>Front/rear nozzle variables are always calculated even if there are only front nozzles. The user should only need to input certain front/rear nozzle variables.</td>
</tr>
<tr>
<td>DE_FR</td>
<td>Calculated</td>
<td>R Ft</td>
<td>Effective Diameter of Front and Rear jets combined - DE_FR is the effective diameter of all the jets combined. This is the diameter of a single circle which would give the same area as the total area of the front and nozzles combined.</td>
</tr>
<tr>
<td>NUM</td>
<td>Calculated</td>
<td>I Ft</td>
<td>NUMber of Front/Rear jets - NUM is the total number of all jets present. This number cannot be greater than four. This is calculated summing NUM_F and NUM_R.</td>
</tr>
<tr>
<td>PER_FR</td>
<td>Calculated</td>
<td>R Ft</td>
<td>Total perimeter of jets - PER_FR is the total perimeter of all the jets combined. Calculation of this takes into account whether the jets are circular, oval, or rectangular.</td>
</tr>
<tr>
<td>PR_FR</td>
<td>*C</td>
<td>R</td>
<td>Jet Pressure Ratio for all jets - PR_FR is used to calculate the lift increments for the hover suckdown and the jet wake. Also used to calculate the average dynamic pressure of all the nozzles.</td>
</tr>
<tr>
<td>Q_FR</td>
<td>Calculated</td>
<td>R lb/Ft²</td>
<td>Dynamic pressure for the front/rear jets - Q_FR is used to calculate the lift increments for the jet wake.</td>
</tr>
<tr>
<td>S_FR</td>
<td>Calculated</td>
<td>R Ft²</td>
<td>Area of Front/Rear jets - Total area of front jets.</td>
</tr>
<tr>
<td>T_FR</td>
<td>*A</td>
<td>R lb</td>
<td>Thrust of Front/Rear jets - Total thrust of all jets. (Make sure this value is not 9999.0)</td>
</tr>
<tr>
<td>X_FR</td>
<td>Calculated</td>
<td>R Ft</td>
<td>Distance between Front and Rear jets - X_FR is the longitudinal distance.</td>
</tr>
</tbody>
</table>
### RCS Variables

<table>
<thead>
<tr>
<th>Name</th>
<th>Default</th>
<th>Type</th>
<th>Units</th>
<th>Description of Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>D_RCS</td>
<td>*D</td>
<td>R</td>
<td>Ft</td>
<td>Diameter for Roll RCS nozzle</td>
</tr>
<tr>
<td>MD_RCS</td>
<td>*D</td>
<td>R</td>
<td>lb/s</td>
<td>Mass flow rate for one RCS nozzle</td>
</tr>
<tr>
<td>PR_RCS</td>
<td>*D</td>
<td>R</td>
<td>-----</td>
<td>Roll RCS Pressure Ratio</td>
</tr>
<tr>
<td>PT_RCS</td>
<td>*D</td>
<td>R</td>
<td>lb/Ft²</td>
<td>Total pressure for one roll RCS jet</td>
</tr>
<tr>
<td>Q_RCS</td>
<td>*D</td>
<td>R</td>
<td>lb/Ft²</td>
<td>Dynamic pressure for roll RCS jets</td>
</tr>
<tr>
<td>RCSFLG</td>
<td>Calculated</td>
<td>R</td>
<td>-----</td>
<td>Flag which identifies if the RCS lift increment calculation exceeded the area ratio. This is not a fatal error.</td>
</tr>
</tbody>
</table>

*Typically these variables are only input when moment increments are being calculated. Since the PIE code does not contain moment increments at this time these variables need only be input when the user is interested in the lift increments caused by the RCS.*

### Fountain Variables

<table>
<thead>
<tr>
<th>Name</th>
<th>Default</th>
<th>Type</th>
<th>Units</th>
<th>Description of Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>T_RCS</td>
<td>*D</td>
<td>R</td>
<td>lb</td>
<td>Thrust of one roll RCS nozzle</td>
</tr>
<tr>
<td>TP_RCS</td>
<td>*D</td>
<td>R</td>
<td>°R</td>
<td>Temperature of flow in roll RCS nozzle</td>
</tr>
<tr>
<td>X_RCS</td>
<td>*D</td>
<td>R</td>
<td>Ft</td>
<td>Distance of roll RCS nozzle ahead of wing trailing edge</td>
</tr>
<tr>
<td>Y_RCS</td>
<td>*D</td>
<td>R</td>
<td>Ft</td>
<td>Distance of roll RCS nozzle from wing tip</td>
</tr>
</tbody>
</table>

*All of these variables, except for SCALE3, are calculated by PIE. They should typically not be changed unless the user wants to compare the PIE results with other results which have different values than calculated by PIE. Setting these variable to user defined values will not speed up the calculations because the SDAREA is still executed.*

Refer to the diagram for the fountain variables. The fountain arms are numbered as shown in the figure. There can be a maximum of four fountain arms (configurations with four or more jets.) The number two fountain arm is a reflection of the number one fountain arm so it is not included in the variable listing. The figure shown has only three fountain arms.
<table>
<thead>
<tr>
<th>Name</th>
<th>Default</th>
<th>Type</th>
<th>Units</th>
<th>Description of Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_1</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>Half distance between adjacent jets for fountain arm (1). (‘e’ in above figure)</td>
</tr>
<tr>
<td>E_3</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>See E_1 - use fountain arm (3).</td>
</tr>
<tr>
<td>E_4</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>See E_1 - use fountain arm (4).</td>
</tr>
<tr>
<td>S_FA1</td>
<td>Calculated</td>
<td>R</td>
<td>Ft²</td>
<td>Actual surface area between the jets which is affected by fountain arm (1). (S’ in above figure)</td>
</tr>
<tr>
<td>S_FA3</td>
<td>Calculated</td>
<td>R</td>
<td>Ft²</td>
<td>See S_FA1 - use fountain arm (3).</td>
</tr>
<tr>
<td>S_FA4</td>
<td>Calculated</td>
<td>R</td>
<td>Ft²</td>
<td>See S_FA1 - use fountain arm (4).</td>
</tr>
<tr>
<td>SCALE</td>
<td>Calculated</td>
<td>R</td>
<td>----</td>
<td>Actual scale factor which adjust the area of fountain arm (3) to account for the fact that the curvature of the aircraft's nose tends not to hold the fountain arm causing the area that the fountain arm affects to be scaled down.</td>
</tr>
<tr>
<td>SCALE3</td>
<td>1.0</td>
<td>R</td>
<td>----</td>
<td>Fountain arm (3) scalar (Percentage measured from the front jets to the nose of the aircraft.) Tries to approximate SCALE but does not do a great job (better than nothing.)</td>
</tr>
<tr>
<td>SP_FA1</td>
<td>Calculated</td>
<td>R</td>
<td>Ft²</td>
<td>Potential surface area between the jets which is affected by fountain arm (1). (S'' in above figure)</td>
</tr>
<tr>
<td>SP_FA3</td>
<td>Calculated</td>
<td>R</td>
<td>Ft²</td>
<td>See SP_FA1 - use fountain arm (3).</td>
</tr>
<tr>
<td>SP_FA4</td>
<td>Calculated</td>
<td>R</td>
<td>Ft²</td>
<td>See SP_FA1 - use fountain arm (4).</td>
</tr>
<tr>
<td>THA_1</td>
<td>Calculated</td>
<td>R</td>
<td>deg</td>
<td>Half the angle between two adjacent jets and the fountain core which is bisected by fountain arm (1) (‘θ’ in above figure)</td>
</tr>
<tr>
<td>THA_3</td>
<td>Calculated</td>
<td>R</td>
<td>deg</td>
<td>See THA_1 - use fountain arm (3)</td>
</tr>
<tr>
<td>THA_4</td>
<td>Calculated</td>
<td>R</td>
<td>deg</td>
<td>See THA_1 - use fountain arm (4)</td>
</tr>
<tr>
<td>Name</td>
<td>Default</td>
<td>Type</td>
<td>Units</td>
<td>Description of Variable</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------</td>
<td>------</td>
<td>--------</td>
<td>----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Y_1</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>Spanwise extent of planform on fountain arm (1) center line. ('y' in above figure)</td>
</tr>
<tr>
<td>Y_3</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>See Y_1 - use fountain arm (3)</td>
</tr>
<tr>
<td>Y_4</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>See Y_1 - use fountain arm</td>
</tr>
<tr>
<td>YP_1</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>Maximum spanwise extent of planform along fountain arm (1). ('Y' in above figure)</td>
</tr>
<tr>
<td>YP_3</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>See YP_1 - use fountain arm (3)</td>
</tr>
<tr>
<td>YP_4</td>
<td>Calculated</td>
<td>R</td>
<td>Ft</td>
<td>See YP_1 - use fountain arm (4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Miscellaneous Variables</strong></td>
</tr>
<tr>
<td>CONFIG</td>
<td>????????...</td>
<td>T</td>
<td></td>
<td>Short title of current configuration (18 characters or less)</td>
</tr>
<tr>
<td>DR</td>
<td>-Z</td>
<td>R</td>
<td></td>
<td>Density Ratio (Jet / Atm)</td>
</tr>
<tr>
<td>F_NAME</td>
<td>test.pie</td>
<td>T</td>
<td></td>
<td>Name of file which contains body &amp; wing, or just wingbody planform coordinates. It should always end with '.pie' to distinguish it from the data file (Maximum length - 50 characters) F_NAME can contain a directory path to the '.pie' file.</td>
</tr>
<tr>
<td>HDEOUT</td>
<td>.TRUE</td>
<td>L</td>
<td></td>
<td>Signals when to output tables based on height or height/De</td>
</tr>
<tr>
<td>KB</td>
<td>.666667</td>
<td>R</td>
<td></td>
<td>Boundary layer factor. Used when calculating forward extent of wall jet. Value is based on whether the ground plane is moving ??</td>
</tr>
<tr>
<td>KLF</td>
<td>1.0</td>
<td>R</td>
<td></td>
<td>Adjustment factor for flap extension when calculating the lift increment due to jet induced effects. 1.0 - Flaps extended .25 - No flap extension</td>
</tr>
<tr>
<td>KR</td>
<td>Calculated</td>
<td>R</td>
<td></td>
<td>Body contour factor Calculated from R_B.</td>
</tr>
<tr>
<td>NDIV</td>
<td>500</td>
<td>I</td>
<td></td>
<td>Number of divisions to be used when calculating suckdown areas (SDAREA) and Dbars (DIABAR). This value has not been optimized. Value could be less and still have good accuracy.</td>
</tr>
<tr>
<td>PRTFLG</td>
<td>.TRUE</td>
<td>L</td>
<td></td>
<td>Signals when to output to the screen. Prints all input variables according to logical section (similar to the sections in this manual) and then allows the user to choose which increments to create a</td>
</tr>
</tbody>
</table>
There are a certain number of variables which must be defined before PIE makes calculations. These variables are called minimum configuration variables. These variables are divided into seven groups; Necessary Group, Thrust Group, Nozzle Definition Group, Pressure Ratio Group, RCS Group, LIDs Group, and Potential Group. The variables in each group, except the Necessary Group, are related which is explained in the following sections. The headings for each section are preceded by a letter which corresponds to the letter found after the “.” in the Variable Definition section.

These variables are general variables. The engineer should read through the Variable Definition section to become familiar with the many different subtle configuration changes which might apply to an aircraft design. This section will list variable names and maybe some pertinent description. The complete definition of the variable can be found in the Variable Definition section.
Here are some points which the user should be aware of:

1) All variables must be entered using the same coordinate system.
2) If there are no rear nozzles then the user need not enter any variables which are specifically for the rear nozzles.

**Necessary Group**

The Necessary Group contains all those variables which must be entered in the input file. These variables have no conditions on them and therefore do not belong in one of the lettered groups.

- X_CG
- XNOZ_F
- YNOZ_F
- XNOZ_R Required if rear nozzles are present.
- YNOZ_R Required if rear nozzles are present.
- XTE_F
- XTE_R Required if rear nozzles are present & value is different from XTE_F.
- MAC
- F_NAME

**Thrust Group**

The Thrust group contains all the variables pertaining to the thrust of the nozzles. Any one of the following groups of two can be entered. Do not enter more than two thrust variables. The only exception to this rule is if there are only front nozzles then it is only necessary to enter T_F or T.FR. If a different group of two, other than the ones listed here, is input then the thrust error flag, T_ERR, will be set to one. This error will not cause PIE to abort but it will cause incorrect results and therefore must be corrected.

- T_F, T_R
- TT_F, T_FR
- TT_R, T_FR
- TT_F, T_F
- TT_R, T_R
- T_FR, T_R
- T_FR, T_F
Nozzle Definition Group

The Nozzle Definition Group contains the variables which define the size and type of nozzles. There are three different types of nozzles which PIE can handle; circular, rectangular and oval. If there are front and rear nozzles, each set must be defined as a type of nozzle using the following definitions.

Circular nozzles are defined by entering the diameter of an individual nozzle in front or rear, if necessary. (D_F, D_R)

Rectangular nozzles are defined by entering the width and length of an individual nozzle in front or rear, if necessary. (W_F, L_F, W_R, L_R)

Oval nozzles are defined by entering the width and length of an individual nozzle along with the diameter of a circular individual nozzle which has the same area as the individual oval nozzle. Use rear nozzles only if necessary. (W_F, L_F, D_F, W_R, L_R, D_R)

Pressure Ratio Group

The Pressure Ratio Group contains all variables which define the pressure ratios of the nozzles. The preferable method for entering the nozzle pressure ratios is to enter the pressure ratios for each nozzle (PR_F, PR_R). If these are not known then the pressure ratio for the whole system must be entered (PR_FR).

RCS Group

The reaction control system group contains all variables which define the nozzles of the RCS. The following is a list of variables which are needed by PIE to compute RCS lift increments. If one of the variables cannot be determined and there is an ‘or’ on the same line then the next variable can be entered in its place. e.g. If the diameter of the RCS nozzle (D_RCS) cannot be found then the three values following it must all be used (MD_RCS, TP_RCS, PT_RCS).

If either X_RCS or Y_RCS are not defined then the RCS routine will not be executed.

X_RCS
Y_RCS
T_RCS
Q_RCS or PT_RCS or PR_RCS
D_RCS or MD_RCS
TP_RCS
PT_RCS
LIDs Group

The LIDs group contains all the variables which define the LIDs and how they are positioned on the airframe. These variables are to be included if the configuration contains LIDS.

- S_LID
- PP_LID
- SP_JP  Required if different than S_JP.

Potential Group

The Potential Group of variables are variables which serve no purpose at this time. They are not needed within any calculations as of this writing. They are defined and positioned throughout PIE to allow the easy incorporation of the moment increment routines. These variables generally refer to moment arms which the forces will act upon to cause the moment increments. Since the lift increments do not need these variables, they can be left out of the input files until the moment increment routines are included.

- DR
- WIWE
- XCG_C4
- XCG_I
- ZCG_I

Output

The storage of the results of PIE are in arrays. This is done so any type of lookup table can be created by a user/programmer. This method of storage allows the programmer to configure an output table in any form using simple index notation when requesting increments. This feature allows many different types of lookup table output.

Currently, PIE produces one three-dimensional lookup table, given an angle of attack, for ACSYNT. This table contains multiple tables based on deflection angle with each individual table based on height and velocity. This table is generated by a single, short subroutine which shows how other types of tables can be generated using similar subroutines.

The other output options are created for graphing purposes. These outputs are an example of the many tables which can be generated. Each lift increment, as well as, the total lift increments can be viewed based on two of four independent variables. Currently, outputs are text files which can be read by a plotting routine, developed at NASA Ames for the Silicon Graphics IRIS workstations, and a Macintosh graphing application called KaleidaGraph™.
Example

Configuration

The example configuration is a McDonnell Douglas YAV-8B, "Harrier" V/STOL aircraft which was originally developed by the British. This configuration was chosen because it is the most common jet lift V/STOL aircraft. One addition this example has over common configurations are LIDs.

Input

Input files can be created in any text editor available to the user. Make sure no special control characters are included in any of the text files.

The data file “config.dat” contains all user defined variables. This file contains namelist formatted input. This means the title of the namelist must be included first in the second column (indent one space) of the file. Following this namelist title on the second line the user defined variables can be entered in any order. The variables in the example file, Table #1, have been listed according to the groups listed above for convenience of the user. After these groups, the variables are positioned in no particular order. It is recommended variables be grouped together in logical groups to allow ease of identification of variables. A suggested group structure would be the groups found in the Variables Definition section of this manual. Each variable must be separated by a comma and one or more of the following: space, tab, or return.

Table #1. Sample Data File - YAV8B.dat

$PIENAM

X_CG = 25.8,
XNOZ_F = 25.3, YNOZ_F = 3.39,
XNOZ_R = 30.38, YNOZ_R = 2.8,
XTE_F = 29.348, MAC = 8.31,
T_F = 9994., T_R = 11423.,
D_F = 1.7692, D_R = 2.17,
PR_FR = 2.25,
PP_LID = 1.0, S_LID = 18.27,
SPLY_F = -10., SPLY_R = -10.,
CONFIG = 'YAV-8B Harrier', WL_F = .74,
B_B = 6.36, KR = .6,
XCG_C2 = -3.6, Z_W = 4.2,
F_NAME = 'yav8b.pie',
HDEOUT = .FALSE.,
HSTART = 1, HEND = 25.,
VSTART = 3.0, VSTEP = 3.5, LH = 25.,
LV = 10.
The planform coordinates file, "config.pie", contains Cartesian coordinates of planforms used for the configuration. This file contains listings of coordinates for particular planforms. Only half the planform must be entered due to symmetry of the configurations.

The format for the coordinates starts with a descriptor of the planform which can be one of the following: WINGPLAN, BODYPLAN, or WINGBODY. Following these descriptors the coordinates for the planform are entered from nose to tail of the configuration. The first and last coordinates must be positioned on the center line of the aircraft (Y-coordinates equal to zero.) Each planform listing is terminated with the planform termination flags, 9999.0, in both the X and Y-coordinate positions. After the planform termination flags are entered the next planform can be entered using the same formats. Enter no more than 500 points for each planform. These formats can be seen in Table #2.

The preferred method for entering planforms is to enter the WINGPLAN and BODYPLAN planforms together. If these planforms cannot be determined then enter only one planform for the WINGBODY. Do not enter all three planforms at once.

Table #2. Example Planform Coordinates File - YAV8B.pie

WINGPLAN
21.3, 0.0
31.5, 14.0
32.9, 14.6
35.44, 14.6
34.0, 8.4
34.0, 0.0
9999., 9999.

BODYPLAN
0.0, 0.0
7.97, 0.21
11.0, 1.4
15.9, 1.4
16.7, 3.23
25.2, 2.91
29.4, 3.18
30.8, 2.3
36.3, 2.0
41.4, 1.07
44.5, 5.0
The set up for execution of PIE requires the input file “config.dat” be assigned or linked to the FORTRAN file, unit 9 (fort.9 for Silicon Graphics Iris.) PIE can then be executed by typing “pie”. Depending on the machine, the code will execute for a few moments and the variable lists will be displayed. Table #3 is a reproduction of the variable lists.

Table #3. Sample Output During Execution

<table>
<thead>
<tr>
<th>YAV-8B Harrier</th>
<th>Wing</th>
<th>Wing-Body</th>
<th>Jet Pattern</th>
<th>Center Sec.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Body</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B_B=6.36</td>
<td>B_W=29.20</td>
<td>B_WB=29.20</td>
<td>B_JP=6.78</td>
<td>B_CS=6.36</td>
</tr>
<tr>
<td>L_B=49.55</td>
<td>L_WB=49.55</td>
<td>S_JP=31.45</td>
<td>L_CS=12.75</td>
<td></td>
</tr>
<tr>
<td>S_B=197.48</td>
<td>S_W=224.30</td>
<td>S_WB=357.53</td>
<td>S_LID=18.27</td>
<td>S_CS=64.24</td>
</tr>
<tr>
<td>DB_B=12.39</td>
<td>DB_W=15.76</td>
<td>DB_WB=19.43</td>
<td>SP_JP=31.45</td>
<td>DB_CS=8.72</td>
</tr>
<tr>
<td>WL_B=0.13</td>
<td>MAC=8.31</td>
<td>WL_WB=0.59</td>
<td>WL_JP=1.33</td>
<td>WL_CS=0.50</td>
</tr>
<tr>
<td>Z_B=0.00</td>
<td>Z_W=4.20</td>
<td></td>
<td>PP_LID=1.00</td>
<td></td>
</tr>
</tbody>
</table>
Table 3. (Continued)

<table>
<thead>
<tr>
<th>Front Nozzles</th>
<th>Rear Nozzles</th>
<th>Both Nozzles</th>
<th>Misc. Nozzles</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUM_F=2</td>
<td>NUM_R=2</td>
<td>NUM=4</td>
<td>PR_FR=2.25</td>
</tr>
<tr>
<td>D_F=1.77</td>
<td>D_R=2.17</td>
<td></td>
<td>DR=1.00</td>
</tr>
<tr>
<td>DE_F=2.50</td>
<td>DE_R=3.07</td>
<td>DE_FR=3.96</td>
<td>WIWE=0.00</td>
</tr>
<tr>
<td>S_F=4.92</td>
<td>S_R=7.40</td>
<td>S_FR=12.31</td>
<td>XCG_I=0.00</td>
</tr>
<tr>
<td>SPLY_F=-10.0</td>
<td>SPLY_R=-10.0</td>
<td>PER_FR=24.75</td>
<td></td>
</tr>
<tr>
<td>W_F=1.77</td>
<td>W_R=2.17</td>
<td>Q_FR=1761.6</td>
<td></td>
</tr>
<tr>
<td>L_F=1.77</td>
<td>L_R=2.17</td>
<td>X_FR=5.08</td>
<td></td>
</tr>
<tr>
<td>WL_F=1.00</td>
<td>WL_R=1.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y_F=6.78</td>
<td>Y_R=5.60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>YB_F=0.47</td>
<td>YB_R=0.24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>YWB_F=-2.10</td>
<td>YWB_R=-9.66</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PR_F=2.25</td>
<td>PR_R=2.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T_F=9994.0</td>
<td>T_R=11423.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TT_F=0.47</td>
<td>TT_R=0.53</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q_F=1761.6</td>
<td>Q_R=1761.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XTE_F=29.35</td>
<td>XTE_R=29.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XCG_F=0.50</td>
<td>XCG_R=-4.58</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SF_F=76.75</td>
<td>SF_RB=107.37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SF_FWB=81.48</td>
<td>SF_RWB=172.68</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XNOZ_F=25.30</td>
<td>XNOZ_R=30.38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>YNOZ_F=3.39</td>
<td>YNOZ_R=2.80</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fount. Arms</th>
<th>RCS Nozzles</th>
<th>Misc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_1=2.56</td>
<td>D_RCS=0.00</td>
<td>XCG_C2=-3.60</td>
</tr>
<tr>
<td>E_3=3.39</td>
<td>PR_RCS=0.00</td>
<td>XCG_C4=0.00</td>
</tr>
<tr>
<td>E_4=2.80</td>
<td>T_RCS=0.00</td>
<td>XCA_CG=-4.83</td>
</tr>
<tr>
<td>THA_1=39.37</td>
<td>TP_RCS=0.00</td>
<td>KB=0.67</td>
</tr>
<tr>
<td>THA_3=57.25</td>
<td>PT_RCS=0.00</td>
<td>KLF=1.00</td>
</tr>
<tr>
<td>THA_4=44.00</td>
<td>MD_RCS=0.00</td>
<td>X_CG=25.80</td>
</tr>
<tr>
<td>S_FA1=0.27</td>
<td>Q_RCS=0.00</td>
<td>X_CA=30.63</td>
</tr>
<tr>
<td>S_FA3=76.97</td>
<td>X_RCS=9999.00</td>
<td></td>
</tr>
<tr>
<td>S_FA4=68.71</td>
<td>Y_RCS=9999.00</td>
<td></td>
</tr>
<tr>
<td>SF_FA1=1.29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SF_FA3=171.53</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SF_FA4=107.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y_1=0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y_3=25.30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y_4=19.17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>YP_1=0.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>YP_3=25.30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>YP_4=19.17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCALE3=1.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 3. (Continued)

<table>
<thead>
<tr>
<th>Body X</th>
<th>Wing X</th>
<th>Wing Y</th>
<th>Wingbody X</th>
<th>Wingbody Y</th>
<th>Center Sec X</th>
<th>Center Sec Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>21.30</td>
<td>0.00</td>
<td>0.00</td>
<td>21.30</td>
<td>0.00</td>
</tr>
<tr>
<td>7.97</td>
<td>0.21</td>
<td>31.50</td>
<td>14.00</td>
<td>7.97</td>
<td>0.21</td>
<td>23.47</td>
</tr>
<tr>
<td>11.00</td>
<td>1.40</td>
<td>32.90</td>
<td>14.60</td>
<td>11.00</td>
<td>1.40</td>
<td>25.20</td>
</tr>
<tr>
<td>15.90</td>
<td>1.40</td>
<td>35.44</td>
<td>14.60</td>
<td>15.90</td>
<td>1.40</td>
<td>29.40</td>
</tr>
<tr>
<td>16.70</td>
<td>3.23</td>
<td>34.00</td>
<td>8.40</td>
<td>16.70</td>
<td>3.23</td>
<td>30.80</td>
</tr>
<tr>
<td>25.20</td>
<td>2.91</td>
<td>34.05</td>
<td>0.00</td>
<td>23.47</td>
<td>2.98</td>
<td>34.04</td>
</tr>
<tr>
<td>29.40</td>
<td>3.18</td>
<td>31.50</td>
<td>14.00</td>
<td>34.05</td>
<td>0.00</td>
<td>34.04</td>
</tr>
<tr>
<td>30.80</td>
<td>2.30</td>
<td>32.90</td>
<td>14.60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36.30</td>
<td>2.00</td>
<td>35.44</td>
<td>14.60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41.40</td>
<td>1.07</td>
<td>34.00</td>
<td>8.40</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44.50</td>
<td>5.00</td>
<td>34.04</td>
<td>2.12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45.70</td>
<td>6.70</td>
<td>36.30</td>
<td>2.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47.40</td>
<td>6.70</td>
<td>41.40</td>
<td>1.07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46.20</td>
<td>0.54</td>
<td>44.50</td>
<td>5.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>49.55</td>
<td>0.00</td>
<td>45.70</td>
<td>6.70</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>47.40</td>
<td>6.70</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>46.20</td>
<td>0.54</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>49.55</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Error Codes

- **DBAR error flag** (DBERR) = 0
- **SDAREA error flag** (SDAERR) = 0
- **Thrust input error** (T_ERR) = 0

After the variables are output to the screen, a menu system will appear which allows the user to choose which type of output required. The following menus and choices are the exact output of a typical PIE run. The first choice is for a table with all hover results, the second choice is for a table with total lift increments, from all routines, based on height for a velocity of 34.5 knots, nozzle deflection angle of 81 degrees, and angle of attack of 6 degrees. The third choice is a table of the totals which can be used by ACSYNT for an angle of attack of 6 degrees. (The output for all these choices are shown following this paragraph.)

**OUTPUT DATA TO TABLE FORMAT**

- 0 - Quit table generation section
- 1 - Output table with all hover results (dL/dT vs Height)
- 2 - Tables with results from STOLSF (any)
- 3 - Tables with results from STOLGV (any)
- 4 - Table with results from RCSIND
- 5 - Table with results from STOLWK (any)
- 6 - Table with totals from all routines
- 7 - Table for JETIND
Choose an option
1

OUTPUT DATA TO TABLE FORMAT

0 - Quit table generation section
1 - Output table with all hover results (dL/dT vs Height)
2 - Tables with results from STOLSF (any)
3 - Tables with results from STOLGV (any)
4 - Table with results from RCSIND
5 - Table with results from STOLWK (any)
6 - Table with totals from all routines
7 - Table for JETIND

Choose an option
6

Table generation for Totals

1 - Output table based on height
2 - Output table based on aircraft velocity
3 - Output table based on jet deflection angle
4 - Output table based on Aircraft Angle of Attack

Choose an option
1

V Number  Velocity
1 --> 3.0
2 --> 6.5
3 --> 10.0
4 --> 13.5
5 --> 17.0
6 --> 20.5
7 --> 24.0
8 --> 27.5
9 --> 31.0
10 --> 34.5

Enter your choice for Aircraft Velocity (V Number)
4

D Number  Def Angle
1 --> 81.0
2 --> 83.0
3 --> 85.0
4 --> 87.0
5 --> 89.0
Enter your choice for Deflection Angle (D Number)
1

A Number    Angle of Attack
1 --> 6.0  
2 --> 8.0  
3 --> 10.0 
4 --> 12.0 

Enter your choice for Aircraft Angle of Attack (A Number)
1

OUTPUT DATA TO TABLE FORMAT

0 - Quit table generation section
1 - Output table with all hover results (dL/dT vs Height)
2 - Tables with results from STOLSF (any)
3 - Tables with results from STOLGV (any)
4 - Table with results from RCSIND
5 - Table with results from STOLWK (any)
6 - Table with totals from all routines
7 - Table for JETIND

Choose an option
7

A Number    Angle of Attack
1 --> 6.0  
2 --> 8.0  
3 --> 10.0 
4 --> 12.0 

Enter your choice for Aircraft Velocity (V Number)
1

OUTPUT DATA TO TABLE FORMAT

0 - Quit table generation section
1 - Output table with all hover results (dL/dT vs Height)
2 - Tables with results from STOLSF (any)
3 - Tables with results from STOLGV (any)
4 - Table with results from RCSIND
5 - Table with results from STOLWK (any)
6 - Table with totals from all routines
7 - Table for JETIND
Choose an option
0

Output

The actual output from PIE is in the form of tables from which graphs can be created or look-ups can be performed.

The following tables were created from the menu choices shown in the previous section. The tables are actual output from PIE except for elimination of “zero” data sets to allow the tables to fit on the pages.

Tables 4 and 5 are basic tables used to create graphs using a Macintosh application called KaleidaGraph or a graphing application for the Silicon Graphics Iris. These two tables follow the format which is described next. The first row contains a number which the Iris application uses to distinguish between different types of input (the number is not used by KaleidaGraph.) The second row contains the title of the table. This title is a combination of the CONFIG variable and a description of the table type. The third and fourth rows contain the titles for the X and Y axis respectively. The fifth column contains the number of data sets \( n \) and the following \( n \) rows contain the labels for those data sets. The next row, following the labels, contains the number of data points. After the above headers, all the data sets are listed with the X-axis value followed by each data set value on the same row.
Table 4. Actual Output from PIE for Hover Lift Increments

1 YAV-8B Harrier, Hover Ground Effect

<table>
<thead>
<tr>
<th>Height</th>
<th>dL/dT</th>
<th>Actual Output from PIE for Hover Lift Increments</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>OG</td>
<td>Suckdown, Fountaint, Lids, Total Zero</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0000</td>
<td>-0.01142</td>
<td>-0.47852, 0.42695, 0.06524, 0.00225, 0.00000</td>
</tr>
<tr>
<td>2.0000</td>
<td>-0.01142</td>
<td>-0.21777, 0.21192, 0.06476, 0.04750, 0.00000</td>
</tr>
<tr>
<td>3.0000</td>
<td>-0.01142</td>
<td>-0.13432, 0.14068, 0.06449, 0.05943, 0.00000</td>
</tr>
<tr>
<td>4.0000</td>
<td>-0.01142</td>
<td>-0.09401, 0.10519, 0.06429, 0.06406, 0.00000</td>
</tr>
<tr>
<td>5.0000</td>
<td>-0.01142</td>
<td>-0.07056, 0.08395, 0.05324, 0.05522, 0.00000</td>
</tr>
<tr>
<td>6.0000</td>
<td>-0.01142</td>
<td>-0.05541, 0.06983, 0.04428, 0.04729, 0.00000</td>
</tr>
<tr>
<td>7.0000</td>
<td>-0.01142</td>
<td>-0.04491, 0.05975, 0.03790, 0.04132, 0.00000</td>
</tr>
<tr>
<td>8.0000</td>
<td>-0.01142</td>
<td>-0.03729, 0.05075, 0.03218, 0.03423, 0.00000</td>
</tr>
<tr>
<td>9.0000</td>
<td>-0.01142</td>
<td>-0.03154, 0.04132, 0.02620, 0.02456, 0.00000</td>
</tr>
<tr>
<td>10.0000</td>
<td>-0.01142</td>
<td>-0.02709, 0.03188, 0.02022, 0.01360, 0.00000</td>
</tr>
<tr>
<td>11.0000</td>
<td>-0.01142</td>
<td>-0.02356, 0.02245, 0.01424, 0.00171, 0.00000</td>
</tr>
<tr>
<td>12.0000</td>
<td>-0.01142</td>
<td>-0.02071, 0.01302, 0.00826, -0.01085, 0.00000</td>
</tr>
<tr>
<td>13.0000</td>
<td>-0.01142</td>
<td>-0.01836, 0.00359, 0.00228, -0.02392, 0.00000</td>
</tr>
<tr>
<td>14.0000</td>
<td>-0.01142</td>
<td>-0.01641, 0.00225, 0.00143, -0.02415, 0.00000</td>
</tr>
<tr>
<td>15.0000</td>
<td>-0.01142</td>
<td>-0.01477, 0.00210, 0.00133, -0.02276, 0.00000</td>
</tr>
<tr>
<td>16.0000</td>
<td>-0.01142</td>
<td>-0.01337, 0.00197, 0.00125, -0.02157, 0.00000</td>
</tr>
<tr>
<td>17.0000</td>
<td>-0.01142</td>
<td>-0.01217, 0.00185, 0.00118, -0.02056, 0.00000</td>
</tr>
<tr>
<td>18.0000</td>
<td>-0.01142</td>
<td>-0.01113, 0.00175, 0.00111, -0.01969, 0.00000</td>
</tr>
<tr>
<td>19.0000</td>
<td>-0.01142</td>
<td>-0.01023, 0.00166, 0.00105, -0.01893, 0.00000</td>
</tr>
<tr>
<td>20.0000</td>
<td>-0.01142</td>
<td>-0.00943, 0.00157, 0.00100, -0.01828, 0.00000</td>
</tr>
<tr>
<td>21.0000</td>
<td>-0.01142</td>
<td>-0.00873, 0.00150, 0.00095, -0.01770, 0.00000</td>
</tr>
<tr>
<td>22.0000</td>
<td>-0.01142</td>
<td>-0.00811, 0.00143, 0.00091, -0.01718, 0.00000</td>
</tr>
<tr>
<td>23.0000</td>
<td>-0.01142</td>
<td>-0.00755, 0.00137, 0.00087, -0.01673, 0.00000</td>
</tr>
<tr>
<td>24.0000</td>
<td>-0.01142</td>
<td>-0.00705, 0.00131, 0.00083, -0.01632, 0.00000</td>
</tr>
<tr>
<td>25.0000</td>
<td>-0.01142</td>
<td>-0.00660, 0.00126, 0.00080, -0.01596, 0.00000</td>
</tr>
</tbody>
</table>
Table 5. Actual Output from PIE for Total Lift Increments

<table>
<thead>
<tr>
<th>Height</th>
<th>dL/dT</th>
<th>OGE</th>
<th>SF</th>
<th>G-VORTEX</th>
<th>JET WAKE</th>
<th>RCS</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0000</td>
<td>-0.01142</td>
<td>0.01334</td>
<td>-0.05691</td>
<td>0.00360</td>
<td>0.00000</td>
<td>-0.05139</td>
<td></td>
</tr>
<tr>
<td>2.0000</td>
<td>-0.01142</td>
<td>0.05747</td>
<td>-0.02746</td>
<td>0.00016</td>
<td>0.00000</td>
<td>0.01875</td>
<td></td>
</tr>
<tr>
<td>3.0000</td>
<td>-0.01142</td>
<td>0.06912</td>
<td>-0.01828</td>
<td>-0.00050</td>
<td>0.00000</td>
<td>0.03892</td>
<td></td>
</tr>
<tr>
<td>4.0000</td>
<td>-0.01142</td>
<td>0.07362</td>
<td>-0.01372</td>
<td>-0.00075</td>
<td>0.00000</td>
<td>0.04773</td>
<td></td>
</tr>
<tr>
<td>5.0000</td>
<td>-0.01142</td>
<td>0.06500</td>
<td>-0.01097</td>
<td>-0.00087</td>
<td>0.00000</td>
<td>0.04174</td>
<td></td>
</tr>
<tr>
<td>6.0000</td>
<td>-0.01142</td>
<td>0.05727</td>
<td>-0.00911</td>
<td>-0.00095</td>
<td>0.00000</td>
<td>0.03579</td>
<td></td>
</tr>
<tr>
<td>7.0000</td>
<td>-0.01142</td>
<td>0.05145</td>
<td>-0.00776</td>
<td>-0.00099</td>
<td>0.00000</td>
<td>0.03128</td>
<td></td>
</tr>
<tr>
<td>8.0000</td>
<td>-0.01142</td>
<td>0.04452</td>
<td>-0.00675</td>
<td>-0.00102</td>
<td>0.00000</td>
<td>0.02533</td>
<td></td>
</tr>
<tr>
<td>9.0000</td>
<td>-0.01142</td>
<td>0.03510</td>
<td>-0.00596</td>
<td>-0.00104</td>
<td>0.00000</td>
<td>0.01668</td>
<td></td>
</tr>
<tr>
<td>10.0000</td>
<td>-0.01142</td>
<td>0.02440</td>
<td>-0.00532</td>
<td>-0.00106</td>
<td>0.00000</td>
<td>0.00660</td>
<td></td>
</tr>
<tr>
<td>11.0000</td>
<td>-0.01142</td>
<td>0.01281</td>
<td>-0.00480</td>
<td>-0.00106</td>
<td>0.00000</td>
<td>-0.00447</td>
<td></td>
</tr>
<tr>
<td>12.0000</td>
<td>-0.01142</td>
<td>0.00056</td>
<td>-0.00436</td>
<td>-0.00108</td>
<td>0.00000</td>
<td>-0.01630</td>
<td></td>
</tr>
<tr>
<td>13.0000</td>
<td>-0.01142</td>
<td>-0.01219</td>
<td>-0.00399</td>
<td>-0.00108</td>
<td>0.00000</td>
<td>-0.02868</td>
<td></td>
</tr>
<tr>
<td>14.0000</td>
<td>-0.01142</td>
<td>-0.01242</td>
<td>-0.00367</td>
<td>-0.00109</td>
<td>0.00000</td>
<td>-0.02860</td>
<td></td>
</tr>
<tr>
<td>15.0000</td>
<td>-0.01142</td>
<td>-0.01106</td>
<td>-0.00340</td>
<td>-0.00109</td>
<td>0.00000</td>
<td>-0.02697</td>
<td></td>
</tr>
<tr>
<td>16.0000</td>
<td>-0.01142</td>
<td>-0.00991</td>
<td>-0.00316</td>
<td>-0.00110</td>
<td>0.00000</td>
<td>-0.02559</td>
<td></td>
</tr>
<tr>
<td>17.0000</td>
<td>-0.01142</td>
<td>-0.00892</td>
<td>-0.00295</td>
<td>-0.00111</td>
<td>0.00000</td>
<td>-0.02440</td>
<td></td>
</tr>
<tr>
<td>18.0000</td>
<td>-0.01142</td>
<td>-0.00807</td>
<td>-0.00276</td>
<td>-0.00111</td>
<td>0.00000</td>
<td>-0.02336</td>
<td></td>
</tr>
<tr>
<td>19.0000</td>
<td>-0.01142</td>
<td>-0.00734</td>
<td>-0.00260</td>
<td>-0.00111</td>
<td>0.00000</td>
<td>-0.02247</td>
<td></td>
</tr>
<tr>
<td>20.0000</td>
<td>-0.01142</td>
<td>-0.00669</td>
<td>-0.00244</td>
<td>-0.00111</td>
<td>0.00000</td>
<td>-0.02166</td>
<td></td>
</tr>
<tr>
<td>21.0000</td>
<td>-0.01142</td>
<td>-0.00613</td>
<td>-0.00231</td>
<td>-0.00111</td>
<td>0.00000</td>
<td>-0.02097</td>
<td></td>
</tr>
<tr>
<td>22.0000</td>
<td>-0.01142</td>
<td>-0.00563</td>
<td>-0.00219</td>
<td>-0.00111</td>
<td>0.00000</td>
<td>-0.02035</td>
<td></td>
</tr>
<tr>
<td>23.0000</td>
<td>-0.01142</td>
<td>-0.00519</td>
<td>-0.00208</td>
<td>-0.00111</td>
<td>0.00000</td>
<td>-0.01980</td>
<td></td>
</tr>
<tr>
<td>24.0000</td>
<td>-0.01142</td>
<td>-0.00479</td>
<td>-0.00197</td>
<td>-0.00111</td>
<td>0.00000</td>
<td>-0.01929</td>
<td></td>
</tr>
<tr>
<td>25.0000</td>
<td>-0.01142</td>
<td>-0.00443</td>
<td>-0.00189</td>
<td>-0.00111</td>
<td>0.00000</td>
<td>-0.01885</td>
<td></td>
</tr>
</tbody>
</table>

Table 6. is a lookup table which is compatible with ACSYNT. (This table is too wide to fit on the page so each row is wrapped to the next row and then indented.) The first row contains three values. The first of these values is the number of heights, the second value is the number of velocities and the third number is the number of nozzle deflection angles. The second row contains the nozzle deflection angle and a listing of all the velocities. The next row contains
a height and the total lift increment for each velocity. This is repeated until all heights have been used then the nozzle deflection angle is changed and a new table is created. This happens until all nozzle deflection angles have been used. (A portion of the table is presented due to the length of the whole table.)

Table 6. Actual Output from PIE for Total Lift Increments for ACSYNT

<table>
<thead>
<tr>
<th>Height</th>
<th>81.000</th>
<th>82.000</th>
<th>83.000</th>
<th>84.000</th>
<th>85.000</th>
<th>86.000</th>
<th>87.000</th>
<th>88.000</th>
<th>89.000</th>
<th>90.000</th>
<th>91.000</th>
<th>92.000</th>
<th>93.000</th>
<th>94.000</th>
<th>95.000</th>
<th>96.000</th>
<th>97.000</th>
<th>98.000</th>
<th>99.000</th>
<th>100.000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0000</td>
<td>-0.14986E-01</td>
<td>-0.28866E-01</td>
<td>-0.40856E-01</td>
<td>-0.51386E-01</td>
<td>-0.60806E-01</td>
<td>-0.78350E-01</td>
<td>-0.81280E-01</td>
<td></td>
<td></td>
<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>2.0000</td>
<td>0.37814E-01</td>
<td>0.30784E-01</td>
<td>0.24534E-01</td>
<td>0.18754E-01</td>
<td>0.13274E-01</td>
<td>0.09630E-01</td>
<td>0.08638E-01</td>
<td></td>
<td></td>
<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>3.0000</td>
<td>0.52194E-01</td>
<td>0.47404E-01</td>
<td>0.43044E-01</td>
<td>0.38924E-01</td>
<td>0.34914E-01</td>
<td>0.31444E-01</td>
<td>0.28424E-01</td>
<td></td>
<td></td>
<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>4.0000</td>
<td>0.58044E-01</td>
<td>0.54384E-01</td>
<td>0.51004E-01</td>
<td>0.47734E-01</td>
<td>0.44494E-01</td>
<td>0.41284E-01</td>
<td>0.38064E-01</td>
<td></td>
<td></td>
<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>5.0000</td>
<td>0.50254E-01</td>
<td>0.47264E-01</td>
<td>0.44744E-01</td>
<td>0.41744E-01</td>
<td>0.39044E-01</td>
<td>0.36244E-01</td>
<td>0.33434E-01</td>
<td></td>
<td></td>
<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>6.0000</td>
<td>0.43074E-01</td>
<td>0.40564E-01</td>
<td>0.38174E-01</td>
<td>0.35794E-01</td>
<td>0.33414E-01</td>
<td>0.31130E-01</td>
<td>0.28424E-01</td>
<td></td>
<td></td>
<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>7.0000</td>
<td>0.37664E-01</td>
<td>0.35474E-01</td>
<td>0.33384E-01</td>
<td>0.31284E-01</td>
<td>0.29124E-01</td>
<td>0.27134E-01</td>
<td>0.25064E-01</td>
<td></td>
<td></td>
<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>8.0000</td>
<td>0.31024E-01</td>
<td>0.29104E-01</td>
<td>0.27234E-01</td>
<td>0.25334E-01</td>
<td>0.23374E-01</td>
<td>0.21330E-01</td>
<td>0.19184E-01</td>
<td></td>
<td></td>
<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>9.0000</td>
<td>0.21334E-01</td>
<td>0.20124E-01</td>
<td>0.18424E-01</td>
<td>0.16844E-01</td>
<td>0.15264E-01</td>
<td>0.13844E-01</td>
<td>0.12424E-01</td>
<td></td>
<td></td>
<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>10.000</td>
<td>0.11324E-01</td>
<td>0.09634E-01</td>
<td>0.08234E-01</td>
<td>0.07044E-01</td>
<td>0.05854E-01</td>
<td>0.04664E-01</td>
<td>0.03474E-01</td>
<td></td>
<td></td>
<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>11.000</td>
<td>-0.10625E-03</td>
<td>-0.15262E-02</td>
<td>-0.29562E-02</td>
<td>-0.44663E-02</td>
<td>-0.60662E-02</td>
<td>-0.77662E-02</td>
<td>-0.95862E-02</td>
<td></td>
<td></td>
<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>12.000</td>
<td>-0.12226E-01</td>
<td>-0.13536E-01</td>
<td>-0.14876E-01</td>
<td>-0.16296E-01</td>
<td>-0.17806E-01</td>
<td>-0.19426E-01</td>
<td>-0.21186E-01</td>
<td></td>
<td></td>
<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>13.000</td>
<td>-0.24866E-01</td>
<td>-0.26086E-01</td>
<td>-0.27346E-01</td>
<td>-0.28676E-01</td>
<td>-0.30126E-01</td>
<td>-0.31476E-01</td>
<td>-0.33166E-01</td>
<td></td>
<td></td>
<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>14.000</td>
<td>-0.24996E-01</td>
<td>-0.26126E-01</td>
<td>-0.27366E-01</td>
<td>-0.28706E-01</td>
<td>-0.30126E-01</td>
<td>-0.31586E-01</td>
<td>-0.33266E-01</td>
<td></td>
<td></td>
<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>15.000</td>
<td>-0.23556E-01</td>
<td>-0.24626E-01</td>
<td>-0.25746E-01</td>
<td>-0.27066E-01</td>
<td>-0.28506E-01</td>
<td>-0.30016E-01</td>
<td>-0.31666E-01</td>
<td></td>
<td></td>
<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>16.000</td>
<td>-0.22336E-01</td>
<td>-0.23466E-01</td>
<td>-0.24606E-01</td>
<td>-0.25826E-01</td>
<td>-0.27306E-01</td>
<td>-0.28866E-01</td>
<td>-0.30506E-01</td>
<td></td>
<td></td>
<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>17.000</td>
<td>-0.21286E-01</td>
<td>-0.22236E-01</td>
<td>-0.23256E-01</td>
<td>-0.24396E-01</td>
<td>-0.25626E-01</td>
<td>-0.26996E-01</td>
<td>-0.28416E-01</td>
<td></td>
<td></td>
<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>18.000</td>
<td>-0.20376E-01</td>
<td>-0.21276E-01</td>
<td>-0.22266E-01</td>
<td>-0.23296E-01</td>
<td>-0.24556E-01</td>
<td>-0.25896E-01</td>
<td>-0.27076E-01</td>
<td></td>
<td></td>
<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>19.000</td>
<td>-0.19596E-01</td>
<td>-0.20456E-01</td>
<td>-0.21406E-01</td>
<td>-0.22466E-01</td>
<td>-0.23636E-01</td>
<td>-0.24826E-01</td>
<td>-0.26016E-01</td>
<td></td>
<td></td>
<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></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>1.000</td>
<td>-15476E-01</td>
<td>-29926E-01</td>
<td>-42256E-01</td>
<td>-53096E-01</td>
<td>-62806E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>2.000</td>
<td>0.38094E-01</td>
<td>0.30914E-01</td>
<td>0.24344E-01</td>
<td>0.18364E-01</td>
<td>0.12714E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>3.000</td>
<td>0.73437E-02</td>
<td>0.21737E-02</td>
<td>0.86000E-02</td>
<td>0.37800E-02</td>
<td>0.47400E-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>4.000</td>
<td>0.52674E-01</td>
<td>0.47704E-01</td>
<td>0.43194E-01</td>
<td>0.38924E-01</td>
<td>0.34794E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>5.000</td>
<td>0.30744E-01</td>
<td>0.26774E-01</td>
<td>0.34230E-01</td>
<td>0.30300E-01</td>
<td>0.32140E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>6.000</td>
<td>0.58624E-01</td>
<td>0.45824E-01</td>
<td>0.51334E-01</td>
<td>0.47944E-01</td>
<td>0.44624E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>7.000</td>
<td>0.41304E-01</td>
<td>0.37984E-01</td>
<td>0.46050E-01</td>
<td>0.42670E-01</td>
<td>0.44560E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>8.000</td>
<td>0.55774E-01</td>
<td>0.47694E-01</td>
<td>0.44814E-01</td>
<td>0.41974E-01</td>
<td>0.39144E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>9.000</td>
<td>0.36314E-01</td>
<td>0.33404E-01</td>
<td>0.41870E-01</td>
<td>0.38500E-01</td>
<td>0.40580E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>10.000</td>
<td>0.43534E-01</td>
<td>0.40934E-01</td>
<td>0.38454E-01</td>
<td>0.36004E-01</td>
<td>0.33534E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>11.000</td>
<td>0.31004E-01</td>
<td>0.28404E-01</td>
<td>0.37150E-01</td>
<td>0.34390E-01</td>
<td>0.35920E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>12.000</td>
<td>0.38074E-01</td>
<td>0.35814E-01</td>
<td>0.33654E-01</td>
<td>0.31464E-01</td>
<td>0.29254E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>13.000</td>
<td>0.26954E-01</td>
<td>0.24594E-01</td>
<td>0.33540E-01</td>
<td>0.30980E-01</td>
<td>0.32290E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>14.000</td>
<td>0.31394E-01</td>
<td>0.29404E-01</td>
<td>0.27464E-01</td>
<td>0.25494E-01</td>
<td>0.23474E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>15.000</td>
<td>0.21364E-01</td>
<td>0.19164E-01</td>
<td>0.28270E-01</td>
<td>0.25860E-01</td>
<td>0.26990E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>16.000</td>
<td>0.22104E-01</td>
<td>0.20344E-01</td>
<td>0.18574E-01</td>
<td>0.16774E-01</td>
<td>0.14904E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>17.000</td>
<td>0.12944E-01</td>
<td>0.10874E-01</td>
<td>0.20110E-01</td>
<td>0.17830E-01</td>
<td>0.18790E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>18.000</td>
<td>0.11514E-01</td>
<td>0.99038E-02</td>
<td>0.82937E-02</td>
<td>0.66238E-02</td>
<td>0.48638E-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>19.000</td>
<td>0.30238E-02</td>
<td>0.10638E-02</td>
<td>0.10410E-02</td>
<td>0.82200E-02</td>
<td>0.90300E-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>20.000</td>
<td>0.26248E-04</td>
<td>0.15062E-02</td>
<td>0.29862E-02</td>
<td>0.45462E-02</td>
<td>0.61862E-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>21.000</td>
<td>0.79462E-02</td>
<td>0.38000E-03</td>
<td>0.24900E-02</td>
<td>0.18300E-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>22.000</td>
<td>0.12266E-01</td>
<td>0.15016E-01</td>
<td>0.16486E-01</td>
<td>0.18036E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>23.000</td>
<td>0.19716E-01</td>
<td>0.11990E-01</td>
<td>0.14050E-01</td>
<td>0.13500E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>24.000</td>
<td>0.25026E-01</td>
<td>0.27596E-01</td>
<td>0.28976E-01</td>
<td>0.30456E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>25.000</td>
<td>0.32066E-01</td>
<td>0.24230E-01</td>
<td>0.26210E-01</td>
<td>0.25450E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>26.000</td>
<td>0.25166E-01</td>
<td>0.27576E-01</td>
<td>0.28876E-01</td>
<td>0.30316E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>27.000</td>
<td>0.31866E-01</td>
<td>0.23910E-01</td>
<td>0.27226E-01</td>
<td>0.24740E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>28.000</td>
<td>0.23706E-01</td>
<td>0.25976E-01</td>
<td>0.27226E-01</td>
<td>0.28606E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>29.000</td>
<td>0.30086E-01</td>
<td>0.22060E-01</td>
<td>0.23420E-01</td>
<td>0.22610E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>30.000</td>
<td>0.22466E-01</td>
<td>0.25060E-01</td>
<td>0.25826E-01</td>
<td>0.27136E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>31.000</td>
<td>0.28586E-01</td>
<td>0.20210E-01</td>
<td>0.21520E-01</td>
<td>0.20840E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>32.000</td>
<td>0.21406E-01</td>
<td>0.23456E-01</td>
<td>0.24616E-01</td>
<td>0.25886E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<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>33.000</td>
<td>0.27306E-01</td>
<td>0.28876E-01</td>
<td>0.18600E-01</td>
<td>0.19250E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6. (Continued)
<table>
<thead>
<tr>
<th>Value</th>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
<th>Column 4</th>
<th>Column 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>18.000</td>
<td>-2.0486E-01</td>
<td>-2.1426E-01</td>
<td>-2.2436E-01</td>
<td>-2.3566E-01</td>
<td>-2.4806E-01</td>
</tr>
<tr>
<td>19.000</td>
<td>-2.6176E-01</td>
<td>-2.7476E-01</td>
<td>-2.8220E-01</td>
<td>-1.8520E-01</td>
<td>-1.7760E-01</td>
</tr>
<tr>
<td>20.000</td>
<td>-1.9696E-01</td>
<td>-2.0576E-01</td>
<td>-2.1556E-01</td>
<td>-2.2646E-01</td>
<td>-2.3856E-01</td>
</tr>
<tr>
<td>21.000</td>
<td>-2.5206E-01</td>
<td>-2.6286E-01</td>
<td>-2.7630E-01</td>
<td>-2.1760E-01</td>
<td>-1.6530E-01</td>
</tr>
<tr>
<td>22.000</td>
<td>-1.8986E-01</td>
<td>-1.9846E-01</td>
<td>-2.0786E-01</td>
<td>-2.1836E-01</td>
<td>-2.3016E-01</td>
</tr>
<tr>
<td>23.000</td>
<td>-2.4266E-01</td>
<td>-2.5246E-01</td>
<td>-2.6286E-01</td>
<td>-2.7630E-01</td>
<td>-1.5460E-01</td>
</tr>
<tr>
<td>24.000</td>
<td>-1.8386E-01</td>
<td>-1.9196E-01</td>
<td>-2.0106E-01</td>
<td>-2.1146E-01</td>
<td>-2.2286E-01</td>
</tr>
<tr>
<td>25.000</td>
<td>-2.3366E-01</td>
<td>-2.4376E-01</td>
<td>-2.5406E-01</td>
<td>-2.6530E-01</td>
<td>-1.4570E-01</td>
</tr>
</tbody>
</table>

(Continued)
Overall Scheme & Assumptions

The scheme for the Power Induced Effects module is as follows: 1) Obtain a complete set of variables for the configuration. 2) Isolate variables to send to lift increment routines based on configuration and planform. 3) Calculate lift increments. 4) Store results in indexed arrays and files. 5) Output configuration variables for the user. 6) Output tables for graphing purposes. Most of the empirical equations have been developed for aircraft with high disk loading jets or fans.

Equations for Subroutines

Hover

The only independent variable used in the hover calculations is height. The other independent variables are set to zero velocity, 90° nozzle deflection angle, and 0° angle of attack.

The method used in this study was developed by Kuhn (reference 3) and assumes the total induced lift increment developed on a V/STOL aircraft can be expressed as:

\[ \frac{\Delta L}{T} = \frac{\Delta L_{\infty}}{T} + \frac{\Delta L_S}{T} + \frac{\Delta L_F}{T} + \frac{\Delta L_L}{T} \]  

Suckdown

\[ \frac{\Delta L_{\infty}}{T} \] is the out-of-ground effect (OGE) suckdown lift increment. The equation for the OGE suckdown is:

\[ \frac{\Delta L_{\infty}}{T} = -0.00022 \sqrt{\frac{S}{A}} \left( \frac{P_n}{P_\infty} \right)^{0.64} \frac{\Sigma \pi d}{d_e} \]^{1.58}  

where \( S \) is the total planform area, \( A \) is the total jet exit area, \( P_n/P_\infty \) is the nozzle pressure ratio, \( \Sigma \pi d \) is the total jet perimeter, and \( d_e \) is the diameter of an equivalent single jet having area \( A \). Experiments have shown that the OGE lift increment is reduced if the nozzles are extended below the surface of a body. The OGE lift increment for a high wing configuration is lower than the low wing configuration and the decrease is related to the square root of the wing height over nozzle diameter as shown in equation (3).
where $\Delta h$ indicates the wing height above the nozzles of the configuration, and the subscripts hw indicates a high wing, b indicates the body, and wb indicates the wingbody.

$\frac{\Delta L_s}{T}$ is the additional lift increment due to suckdown experienced in-ground effect (GE). It was first determined experimentally for single jet configurations and then a correction factor was developed for multiple jet configurations. The equation for in-ground effect suckdown is:

$$\frac{\Delta L_s}{T} = K_s \left( 0.15 + \frac{h}{D_{\text{c}}} \right) \left[ 2.2 \cdot .24 \left( \frac{P_d}{P} \right) - 1 \right]$$

(4)

Where in addition to the variables mentioned above, $D$ is the angular mean diameter of the planform, and $K_s$ is the multiple jet correction factor which is a function of configuration geometry. This GE suckdown also changes with wing height similar to the OGE suckdown. This is because the wing does not experience the same magnitude of entrainment as the lower surface of the body.

**Fountain Lift**

Two methods were developed to calculate the fountain lift, one for widely spaced jets, the Basic Method, and one for closely space jets, the $h'$ Method.

**Basic Method**

The Basic Method was developed for widely spaced jets with the equal thrusts. $\frac{\Delta L_F}{T}$ is the overall fountain lift increment which is a combination of the lift increment developed by the fountain arms and core.

$$\frac{\Delta L_F}{T} = \frac{\Delta L_A}{T} + \frac{\Delta L_C}{T}$$

(5)

Equations (6) and (7) are used to calculate the fountain arm contribution. Equation (6) is the lift increment for an individual fountain arm 'x'. Equation (7) combines the fountain arms in the configuration.
\[
\frac{\Delta L_{\text{Ax}}}{T} = 2\left(\frac{Y_x S_x}{N(e_x S_x''')^{1/3}}\right) \left(\frac{e_x}{e_x + h}\right)^2 \frac{y_x}{\sqrt{y_x^2 + (e_x + h)^2}}
\]

(6)

\[
\frac{\Delta L_A}{T} = \frac{1}{2} \left( \frac{\Delta L_{A,1}}{T} + \frac{\Delta L_{A,2}}{T} + \ldots + \frac{\Delta L_{A,N}}{T} \right) \left( \frac{h}{d_c} \right) \sqrt{\frac{d_c}{D - 1}}
\]

(7)

\(y_x\) is the spanwise extent of the jet on the planform, \(Y_x\) is the maximum spanwise extent of the jet on the planform, \(e\) is half the distance between adjacent jets, \(h\) is the height of the lowest surface above the ground, \(S'\) is the actual surface area between the jets, \(S''\) is the potential surface area between the jets, and \(N\) is the number of fountain arms. These variables can be seen in Figure #2.

![Figure #2: Fountain variables from a configuration with three jets.](image)

The calculations for the fountain core are similar to those of the fountain arms except the contribution of each fountain arm on the fountain core is:

\[
\frac{\Delta L_{Cx}}{T} = K_C \left( \frac{e_x}{e_x + h} \right)^{\lambda_C} \cos \theta_x
\]

(8)

\[
\frac{\Delta L_{C,1}}{T} + \frac{\Delta L_{C,2}}{T} + \ldots + \frac{\Delta L_{C,N}}{T}
\]

(9)

Where the constant \(K_C\) and \(\lambda_C\) depend on the configuration geometry and altitude, \(e\) is half the distance between adjacent jets, \(h\) is the height of lowest
The h' Method was developed because configurations with closely spaced jets exhibit an abrupt increase in fountain lift at low altitudes which is not predicted with the Basic Method. Closely spaced jets have higher pressures between the jets which the Basic Method fails to predict. These higher pressures increase rapidly as the jet spacing is reduced. The ability to contain this pressure breaks down as the height is increased because the jets tend to merge into a single jet before they reach the ground.

Two different equations are used for the fountain increment. The decision for which equation to use is based on an altitude, h'.

\[
\frac{\Delta L_F}{T} = K' \left( \frac{h}{d_c} \right)^{\lambda'} 
\]

(10)

\[
\frac{\Delta L_F}{T} = 0.033 \left( \frac{Dw}{d_c l} \right) \left( \frac{d_c}{h} \right) 
\]

(11)

where \( K' \) and \( \lambda' \) are parameters calculated based on jet spacing, diameter, number, surface area, etc..., \( w \) is width of the configuration, and \( l \) is the length of the configuration. The two curves do not intersect but are joined by a straight line which is tangent to the curve of equation (10) and is projected to \( \frac{\Delta L_F}{T} = 0 \) at a height defined as \( h' \). This is shown in the graph of Figure #3.
Lift Improvement Devices

\[ \frac{\Delta L}{T} \] is the lift improvement devices (LIDs) lift increment. This lift increment is an addition to the total fountain by a percentage, \( K_L \), of the total fountain lift. \( K_L \) can be determined using the area inclosed by the LIDs, ratio of perimeter by LIDs to the total perimeter area enclosed by LIDs, area enclosed by jet centers, length-to-width ratio of jet pattern, and altitude of aircraft. These variables can be seen in Figure #4.
Forward Flight / Jet Deflection Angle / Angle of Attack

The calculations based on forward flight, jet deflection angle, and angle of attack use hover calculations as a basis for each lift increment calculation. The method assumes that the total lift can be expressed as:

\[
\frac{\Delta L}{T} = \frac{\Delta L_S}{T} + \frac{\Delta L_F}{T} + \frac{\Delta L_{JW}}{T} + \frac{\Delta L_V}{T}
\]  

(12)

where the subscript S refers to the suckdown lift increment (no fountain), F refers to the fountain lift increment, JW refers to the jet wake lift increment, and V refers to the lift increment due to the ground vortex.

Suckdown/Fountain

The suckdown and fountain terms are calculated using a method developed in Reference 6. This method calculates a factor based on the forward extent of the wall jet and the jet spacing of the front nozzles. This factor, \(K_h\), varies between one (the wall jet extending ahead of the configuration) and zero (the wall jet swept behind the configuration.) This factor is multiplied by the square of the sine of the jet deflection angle (\(\delta\)). (Angle of attack is not used in these calculations.) This calculation produces a factor which modifies the suckdown and fountain terms for the effects of forward speed and jet deflection angle. The suckdown lift increment for forward flight and jet deflection angle is calculated as follows:
The fountain lift increment is calculated using the fountain and LID lift increments from the hover calculations as shown in equation (14).

\[
\frac{\Delta L_{f}}{T} = \left(\frac{\Delta L_{f}}{T}\right)_{\text{hover}} \sqrt{\frac{h'}{h''}} K_h \sin^2 \delta
\]  

(14)

This lift increment calculation is very similar to equation (13) except for the \(\sqrt{\frac{h'}{h''}}\) term. This term is included due to the recommendation of Richard Kuhn. It is ratio of \(h'\), which was described earlier and \(h''\), which is the maximum height to which the fountain effects are felt on the airframe.

**Ground Vortex**

The ground vortex term is calculated by assuming the configuration is composed of a body and a wing. The lift increment is based on the thrust, diameter and area of the front jets. There are two critical heights associated with the ground vortex lift increments; \(h_1\) is the height at which the rate of change of lift with height changes and \(h_2\) is the maximum height at which the ground vortex effects are felt. The equations are different for the wing and body and each of those are different for heights based on \(h_1\) and \(h_2\). An example of these relations is the equation for a body at a height between \(h_1\) and \(h_2\), equation (14).

\[
\frac{\Delta L_v}{T} = \frac{T_f}{T} 18 \left[\frac{w}{l_f}\right]^2 \left[\frac{S_B}{A_f}\right] \left[\frac{l}{l_B}\right] \Delta C_p \left[1 + \sin(\delta - 90)\right] \sqrt{\frac{1}{V_c}} \frac{h}{d_f}^{-3.2}
\]  

(14)

The other equations are similar to equation (14) in complexity and format and they can be found in reference 6. The subscript \(f\) refers to the front jets, and \(B\) refers to the body configuration. \(T\) is the thrust, \(w\) is the width, \(l\) is the length, \(S\) and \(A\) are area, \(V_e\) is the ratio of the aircraft dynamic pressure to the jet dynamic pressure, \(\Delta C_p\) is a factor based on \(V_e\) and height, and \(\delta\) is the jet deflection angle.

**Jet Wake**

The lift increment due to the jet wake is the sum of the increments due to each jet on the wing and body. This is shown in equation (15) where \(B\) refers to the body planform, \(W\) refers to the wing planform, \(f\) refers to the front jets and \(r\) refers to the rear jets.
The lift increment for the wake is figured by first calculating the lift increment out-of-ground effect and then including this term in the overall lift increment in-ground effect. The OGE jet wake lift increment is calculated using equation (16), which is the equation for the lift increment on a square planform with the jet at the center, and multiplying it by adjustment factors for the planform aspect ratio, longitudinal, vertical, and lateral position of the jet, distance of the jet center from the side of the body, jet deflection angle, non-circular nozzles, jet pressure ratio, and jet flap (Reference 2 and 6).

\[
\frac{\Delta L_{\text{w, square}}}{T} = \left[ -3 \, V_c^2 \left( \sqrt{\frac{S}{A}} - 1 \right)^{6.7} + 35 \, V_c^{5.5} \left( \sqrt{\frac{S}{A}} - 1 \right) \right] \tag{16}
\]

The OGE lift increment is included in equation (17) which is the equation to calculate the jet wake lift increment for an individual jet on a body planform.

\[
\frac{\Delta L_{w,x}}{T} = \frac{\Delta L_{w, OGE}}{T} \left[ 1 - 0.7 \, \sqrt{\frac{S}{A_B}} \, \left( \frac{w}{l_B} \right)^{0.7} \left( \frac{\delta}{90} \right)^{1.34} \left( \frac{w}{l_j} \right)^{0.35} \right] \tag{17}
\]

Similar equations are used in calculations for the wing planform (Reference 6). The subscript j refers to the jets.

If the jets are placed near the trailing edge of the wing, there is a positive lift increment lift increment instead of a negative. This is due to the jet flap effect. In this case a positive lift increment is calculated and replaces the OGE term an equation similar to equation (17).

**Reaction Control System**

The effectiveness of the roll control jets near the wing tip during transition depends on the proximity of the control jets to the wing tip and the wing trailing edge, and jet pressure ratio \( (P_j/P_w) \). The empirical equations developed for this lift increment are based on an integration of the pressure distributions on the surface areas around the nozzles.

The lift increments are assumed to be made up of two terms, one which is not affected by pressure ratios \( (O) \) and another from the effect of pressure ratios above a critical pressure ratio of 1.893 \( (P) \).

\[
\frac{\Delta L_{\text{RCS}}}{T} = \left( \frac{\Delta L}{T} \right)_O + \left( \frac{\Delta L}{T} \right)_P \tag{18}
\]

where:
The variable \( S \) is the area of the wing, \( A_j \) is the jet exit area, \( P_j \) is the jet exit pressure, and \( P_0 \) is the atmospheric pressure. These equations reproduce wind tunnel data with reasonable accuracy up to effective velocity ratios of \( V_e = .1 \), planform-to-jet area ratios up to 7000, and jet pressure ratios up to 45.

The magnitude of the lift increment decreases as the jet is moved closer to the wing tip or trailing edge. This is because the total area near the jet is decreased which reduces the effect of the negative pressures caused by the jet. Equation (21), (22), and (23) show the jet location adjustment factors and how they affect the total RCS lift increment.

\[
\left( \frac{\Delta L}{T} \right)_T = \left[ 3 V_e^3 - 2.4 V_e^2 \right] \sqrt{\frac{S}{A_j}} + 0.41 V_e^2 \left( \frac{P_j}{P_0} \right)^{0.75}
\]  

\[
\left( \frac{\Delta L}{T} \right)_P = -0.017 V_e \left( \frac{S}{A_j} \right)^{0.42} \left( \frac{P_j}{P_0} - 1.893 \right)^{0.75}
\]

\[
K_b = 0.25 + 0.2 \left( \frac{y}{d_c} \right)^{5.8}
\]

\[
K_c = 0.25 + 0.06 \frac{x}{d_c}
\]

\[
\frac{\Delta L_{RCS}}{T} = \left[ \left( \frac{\Delta L}{T} \right)_T + \left( \frac{\Delta L}{T} \right)_P \right] K_b K_c
\]

\( K_b \) is the adjustment factor for the proximity of the jet to the wing tip, \( y \) is the distance from the trailing edge to the jet, \( K_c \) is the adjustment factor for the proximity of the jet to the wing trailing edge, and \( x \) is the distance from the wing trailing edge to the jet.

**Routine Descriptions**

All of the following routines make up PIE. They consist of nine groups; Control, Preliminary, Hover, Suckdown/Fountain, Ground Vortex, Jet Wake, Reaction Control System, Output and Miscellaneous. Almost all variables are transferred between routines using multiple, named common blocks. Each common block contains variables which can be grouped logically. The common blocks are; CONPIE, FIGPIE, HOVPIE, PIEFLAGS, PIEGV, PIERCS, PIERROR, PIESF, PIEWK, POLYGON, RESULTS.

**Control**

The Power Induced Effects module (PIE) is controlled by the control program, COPPIE, which coordinates the execution of PIE. COPPIE first calls
INPIE to determine all the user defined variables. Output and scratch files are opened, planform modifications are made, and final calculations for variables are finished. Next, all lift increments are calculated within a nested do-loop structure. Each increment routine is positioned inside the do-loops according to the independent variables used for each increment. After all increments are calculated, the output routine is called and then all necessary files are closed.

COPPIE contains the ACSYNT control variable, ICALC. Currently this variable is set to the appropriate value as execution proceeds through the routine. When PIE is to be incorporated with ACSYNT, the ICALC definitions need to be removed so PIE can be controlled by ACSYNT.

Preliminary Routines

INPIE

This routines first defines the namelist, PIENAM, and then sets all the default values for each variable. INPIE then reads in the user defined variables from unit 9. Next, INPIE opens the file which contains the planform coordinates, reads the header and reads the appropriate planform coordinates. This is repeated until all coordinates from all planforms have been read from the file. If only a WINGBODY planform is entered then the flag WBFLAG is set to true.

PLANMO

PLANMO's first task is to determine the maximum and minimum points of the planform and then determine .1% of the difference between the two to add that to any point which has the same X-coordinate as the previous point. (This is done because the slope of a line between two points with the same X-coordinate is infinity.) At all time PLANMO checks which planform is being used.

The second task is to combine the individual body and wing planforms into one planform, the wingbody. As a by-product of this combination, the center section planform is determined. Both of these planforms are found by determining where the body and wing planforms cross and saving those points and combining the appropriate planforms together.

The last task for PLANMO is to move the most forward part of the planforms (generally the aircraft nose) to the origin of the coordinate system. Then move all other positioned items the same amount. These items include: body planform, wing planform, wingbody planform, center section planform, nozzles, center of gravity, center of area, trailing edge, RCS nozzles.

FINVAR

The purpose of FINVAR is to calculate all variables which have not been defined. Variables which have not been defined are those variables which have been set to the "undefined" flag value of 9999.0. All variables are checked with this flag before they are calculated except those calculated from the
SDAREA routine. Most calculations in this routine are very simple and are too numerous for complete description in this manual. (See code for individual calculations.)

SDAREA

The SDAREA routine was written to determine all variables which are used in fountain calculations. These include half the angle between adjacent jets and the center of the jet pattern ($\theta$), half the distance between adjacent jets ($e$), maximum spanwise extent of jet on planform ($Y$), spanwise extent of the jet on the planform ($y$), actual surface area between the jets ($S'$), potential surface area between the jets ($S''$).

The routine first determines angles and defines important lines on the configuration. Then the routine divides the planform into thin slices which are summed over the entire planform to determine the areas for the configuration. During this summation, areas are calculated for individual regions of the planform the spanwise extents of the planforms are determined.

This routine was originally written as a stand-alone application. It was modified to allow it to be used as a subroutine. The routine returns the values to PIE by determining the nozzle configuration and assigning calculated values to the appropriate variables used in PIE.

DIABAR

This routine calculates the angular mean diameter of a planform which is the diameter of a circle which has the same circumference as the given planform. The routine divides the planform into arcs from some point in the interior of the planform (center of the jet pattern.) The lengths of these arcs are summed and then the angular mean diameter is determined from the total circumference of the planform.

This routine also has the capability of calculating areas of the planform in front and behind the interior point. There is also extensive user prompts, and output sections which are not used by PIE. These sections can be accessed by recoding PIE so IFLAG is set to one instead of zero when DIABAR is called.

Hover

HCALL

HCALL isolates the local variables within HOV_GE from the global configuration variables. The reason for this is HOV_GE modifies some of its local configuration variables and it is desired to keep the original configuration variables from changing.
HOV_GE

The HOV_GE routine is the lift increment calculation routine for configurations in hover. This routine receives its input through the HOVPIE and RESPIE common block and outputs its results to the RESPIE common block. The routine is based on the original BASIC program developed by Richard Kuhn. The lift increments which it calculates are out-of-ground effect suckdown, in-ground effect suckdown, fountain effect, and LID effect.

Suckdown/Fountain

STOLSF

The STOLSF routine calculates the suckdown/fountain lift increment as it varies with height, forward velocity and nozzle deflection angle. The routine receives input through the PIESF and RESPIE common block, makes the calculations, and then outputs the results to direct access scratch files which are used in the output routines. The results are stored in the scratch files using an equation which assign a unique record number to each result. The record numbers are calculated with the same indices used when accessing arrays.

Ground Vortex

STOLGV

The STOLGV routine is more simplified than previous increment routines. This routine almost independent of configuration. The routine receives input through PIEGV and RESPIE common blocks, uses a lift increment equation based on the heights of the configurations and passes the results back to GVCALL through the PIEGV common block.
**Jet Wake**

**WKCALL**

WKCALL isolates the local variables within JIEPIE and STOLWK from the global configuration variables. This routine is the most detailed control routine. It first determines which planform is being used and then using that planform, makes calculations and routine calls based on the front nozzles and then based on the rear nozzles. This is repeated until all necessary planforms have been used in the calculations. All results are stored using similar scratch files mentioned in STOLSF.

**JIEPIE**

The purpose of JIEPIE is to calculate the change in the lift caused by the jet wake induced effects on a flat plate. JIEPIE accounts for aspect ratio, longitudinal, vertical, and lateral position of the jets, deflection angle, distance of nozzles from body, noncircular or closely spaced jets, jet-flap effect, and jet pressure ratio. The routine receives input through PIEWK and RESPIE common blocks and it passes the results back to WKCALL through the PIEWK common block.

**STOLWK**

The purpose of STOLWK is to calculate the lift increment due to the jet wake system. This is calculated by checking the aspect ratio of the planform and the proximity of the nozzles to the wing trailing edge. All variables are passed to STOLWK with the PIEWK and RESPIE common blocks and its results are returned to WKCALL with the PIEWK common block.

**Reaction Control System**

**RCSCAL**

RCSCAL is executed only once for each velocity because the RCS lift increment does not vary with height, nozzle deflection angle or angle of attack. The routine executes the lift increment routine only when the user has set FLGRCS to TRUE, otherwise the routine sets the RCS increment to zero.

**RCSIND**

The purpose of RCSIND is to make all necessary configuration calculation and lift increment calculations. All variables are passed to RCSIND with the PIERCS and RESPIE common blocks and its results are stored in the RESPIE common block using an array.
Output

PIEP

PIEP outputs all results either to the screen or to files for later use. All variables are passed to this routine using the FIGPIE, PIEFLAG, RESPIE, and PIERROR common blocks. The first task of PIEP is output to the screen of all configuration variables in an easy to read format. This is immediately followed by a menu system from which many different tables can be produced. All results which are stored as arrays can be recalled using arrays. The results which were stored using scratch files need the record number equations to recall the specific results from the scratch files.

Most results were stored as components of the lift increments. The suckdown/fountain lift increments are stored as as suckdown component and also as a fountain component. The ground vortex and jet wake increments are stored as wing and as body components. These components must be combined when any total increments are requested.

JIETAB

JIETAB is an example table output routine. This routines outputs a table which is compatible with ACSYNT. The only variable passed as a parameter is the angle of attack number for which the table is to be produced. All other variables are passed to JIETAB using the RESPIE common block.

Miscellaneous

CENTAR

The CENTAR routine calculates the center of area, area, or area in front of a point for a give planform depending on the value of the variable MODE. Important variables are passed as parameters.

CROSSIN

The CROSSIN routine calculates the intersection of two lines, with each line defined by two points (X,Y). The end points of each line lie on the same X-coordinate. All variables are passed using parameters.

DIST

The DIST routine calculates the distance between two points on the Cartesian plane using the distance formula. These points are passed as parameters to the routine.

INTEGRA

The INTEGRA routine calculates the areas of rectangles which are 'dx' wide and bounded by 1) two values defined by 'a' and 'b' and 2) a value defined by
'c' and zero. This routine is used when calculating areas in the SDAREA routine. All variables are passed using parameters.

LINECRO

The LINECRO routine calculates the intersection of two lines, each defined by a slope and Y-intercept. All variables are passed using parameters.

LINTERP

The LINTERP routine linear interpolates between two given points using a value on the X-axis. All variables are passed using parameters.

PERPDIS

The PERPDIS routine calculates the perpendicular distance between a point and a line. The point is defined by its X- and Y-coordinate, the line is defined by its slope and Y-intercept. All variables are passed using parameters.

POLYAR

The POLYAR routine calculates the area of any polygon with up to 500 points. The points are passed to the routine using the MISC common block.

RTOTAL

The RTOTAL routine sums the different lift increments to the total lift increment. This is done given an indexed notation (similar to array notation).

SSEN

The SSEN routine is the Start, Step, End, Number calculator. Given any three values for START, STEP, END, and NUMBER the routine will calculate the undefined value. If less than three of the variables are given then the routine uses default values to calculate any variables not defined. Variables are undefined when their values are set to the flag value of 9999.0. All variables are passed using parameters.
References

1. Aircraft Synthesis Program (ACSYNT), National Aeronautics and Space Administration, Ames Research Center, Moffett Field, California


## Appendices

### Routine Summary

<table>
<thead>
<tr>
<th>Routine</th>
<th>Acronym</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPPIE</td>
<td>COnrol Program for the Power Induced Effects module</td>
<td>Controls execution and coordination of Power Induced Effects Module</td>
</tr>
<tr>
<td>INPIE</td>
<td>INput for Power Induced Effects module</td>
<td>Sets defaults for variables and reads user defined input file</td>
</tr>
<tr>
<td>PLANMO</td>
<td>PLANform MOdifications</td>
<td>Modifies planforms for consistency and integrity</td>
</tr>
<tr>
<td>FINVAR</td>
<td>FINish Variables</td>
<td>Calculates all variables which have not been defined by the user or set to a default value</td>
</tr>
<tr>
<td>SDAREA</td>
<td>SuckDown Area</td>
<td>Calculates areas necessary to make calculations for the fountain effect</td>
</tr>
<tr>
<td>DIABAR</td>
<td>DIAmeter BAR (D) - compiler would not accept DBAR</td>
<td>Calculates angular mean diameter of a planform</td>
</tr>
<tr>
<td>HCALL</td>
<td>Hover CALLing routine</td>
<td>Isolates and controls execution of HOV_GE</td>
</tr>
<tr>
<td>HOV_GE</td>
<td>Hover with Ground Effects</td>
<td>Calculates hover lift increments for OGE and GE suckdown, and fountain effects</td>
</tr>
<tr>
<td>SFCALL</td>
<td>Suckdown/Fountain CALLing routine</td>
<td>Isolates and controls execution of STOLSF</td>
</tr>
<tr>
<td>STOLSF</td>
<td>STOL operation - estimation of Suckdown and Fountain effects</td>
<td>Calculates suckdown and fountain lift increments while in STOL flight</td>
</tr>
<tr>
<td>GVCALL</td>
<td>Ground Vortex CALLing routine</td>
<td>Isolates and controls execution of STOLGV</td>
</tr>
<tr>
<td>STOLGV</td>
<td>STOL operation - estimation of Ground Vortex effects</td>
<td>Calculates ground vortex lift increments while in STOL flight</td>
</tr>
<tr>
<td>WKCALL</td>
<td>jet WaKe CALLing routine</td>
<td>Isolates and controls execution of STOLWK</td>
</tr>
<tr>
<td>JIEPIE</td>
<td>Jet Induced Effects for PIE</td>
<td>Calculates change in lift caused by the jet induce effects on a flat plate</td>
</tr>
<tr>
<td>STOLWK</td>
<td>STOL operation - estimation of jet WaKe effects</td>
<td>Calculates jet wake lift increments while in STOL flight</td>
</tr>
<tr>
<td>RCSCAL</td>
<td>Reaction Control System CALLing routine</td>
<td>Isolates and controls execution of RCSIND</td>
</tr>
<tr>
<td>RCSIND</td>
<td>Reaction Control System</td>
<td>Calculates RCS lift increments while in forward flight</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>PIEP</td>
<td>INDuced effects</td>
<td>Printing routine for PIE</td>
</tr>
<tr>
<td>JIETAB</td>
<td>Power Induced Effects</td>
<td></td>
</tr>
<tr>
<td>CENTAR</td>
<td>TABle generation routine</td>
<td></td>
</tr>
<tr>
<td>CROSSIN</td>
<td>CROSSING routine</td>
<td></td>
</tr>
<tr>
<td>DIST</td>
<td>DISTance function</td>
<td></td>
</tr>
<tr>
<td>INTEGRA</td>
<td>INTEGRAtion routine</td>
<td></td>
</tr>
<tr>
<td>LINECRO</td>
<td>LINE CROssing routine</td>
<td></td>
</tr>
<tr>
<td>LINTERP</td>
<td>Linear INTERPolation</td>
<td></td>
</tr>
<tr>
<td>PERPDIS</td>
<td>PERPendicular distance routine</td>
<td>Linear interpolates between two X and Y values give an intermediate X value</td>
</tr>
<tr>
<td>POLYAR</td>
<td>POLYgon ARea routine</td>
<td></td>
</tr>
<tr>
<td>RTOTAL</td>
<td>Read TOTAL increments</td>
<td></td>
</tr>
<tr>
<td>SSEN</td>
<td>Start, Step, End, Number Calculator</td>
<td>Sums the total lift increment given array type indices</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Cross Reference - Common Blocks

<table>
<thead>
<tr>
<th>Common Routine</th>
<th>conpie</th>
<th>figpie</th>
<th>hovpie</th>
<th>pieflags</th>
<th>piegv</th>
<th>piersc</th>
<th>piersor</th>
<th>piest</th>
<th>piewk</th>
<th>polygon</th>
<th>respie</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPPIE</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>INPIE</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>PLANMO</td>
<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>FINVAR</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SDAREA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>DIABAR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>HCALL</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>HOV_GE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SFCALL</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>STOLSF</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>GVCALL</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STOLGV</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WKCALL</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JIEPIE</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STOLWK</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCSCAL</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>RCSIND</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>PIEP</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>JIETAB</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>CENTAR</td>
<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>CROSSIN</td>
<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>DIST</td>
<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>INTEGRA</td>
<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>LINECRO</td>
<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>LINTERP</td>
<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>PERPDIS</td>
<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>POLYAR</td>
<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>RTOTAL</td>
<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>SSFN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Cross Reference - Calls To

<table>
<thead>
<tr>
<th>Module</th>
<th>Makes calls to:</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPPIE</td>
<td>INPIE, PLANMO, FINVAR, HCALL, RCSCAL, SFCALL, WKCALL, GVCALL, PIEP</td>
</tr>
<tr>
<td>INPIE</td>
<td>NONE</td>
</tr>
<tr>
<td>PLANMO</td>
<td>CROSSIN</td>
</tr>
<tr>
<td>FINVAR</td>
<td>LINTERP, CENTAR, POLYAR, SDAREA, DIABAR, SSEN</td>
</tr>
<tr>
<td>SDAREA</td>
<td>LINTERP, CROSSIN, LINTERP, POLYAR, SDAREA, DIST (FUNCTION)</td>
</tr>
<tr>
<td>DIABAR</td>
<td>NONE</td>
</tr>
<tr>
<td>HCALL</td>
<td>HOV_GE</td>
</tr>
<tr>
<td>HOV_GE</td>
<td>NONE</td>
</tr>
<tr>
<td>SFCALL</td>
<td>STOLSF</td>
</tr>
<tr>
<td>STOLSF</td>
<td>NONE</td>
</tr>
<tr>
<td>GVCALL</td>
<td>STOLGV</td>
</tr>
<tr>
<td>STOLGV</td>
<td>NONE</td>
</tr>
<tr>
<td>WKCALL</td>
<td>JIEPIE, STOLWK</td>
</tr>
<tr>
<td>JIEPIE</td>
<td>NONE</td>
</tr>
<tr>
<td>STOLWK</td>
<td>NONE</td>
</tr>
<tr>
<td>RCSCAL</td>
<td>RCSIND</td>
</tr>
<tr>
<td>RCSIND</td>
<td>NONE</td>
</tr>
<tr>
<td>PIEP</td>
<td>JIEPIE</td>
</tr>
<tr>
<td>JIEPIE</td>
<td>RTOInc</td>
</tr>
<tr>
<td>RTOTAL</td>
<td>LINTERP, POLYAR</td>
</tr>
<tr>
<td>CROSSIN</td>
<td>LINTERP</td>
</tr>
<tr>
<td>DIST</td>
<td>NONE</td>
</tr>
<tr>
<td>INTEGRA</td>
<td>NONE</td>
</tr>
<tr>
<td>LINECRO</td>
<td>NONE</td>
</tr>
<tr>
<td>LINTERP</td>
<td>NONE</td>
</tr>
<tr>
<td>PERPDIS</td>
<td>NONE</td>
</tr>
<tr>
<td>POLYAR</td>
<td>NONE</td>
</tr>
<tr>
<td>RTOTAL</td>
<td>NONE</td>
</tr>
<tr>
<td>SSEN</td>
<td>NONE</td>
</tr>
</tbody>
</table>
# Cross Reference - Calls By

<table>
<thead>
<tr>
<th>Routine</th>
<th>Called by:</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPPIE</td>
<td>NONE</td>
</tr>
<tr>
<td>INPIE</td>
<td>COPPIE</td>
</tr>
<tr>
<td>PLANMO</td>
<td>COPPIE</td>
</tr>
<tr>
<td>FINVAR</td>
<td>COPPIE</td>
</tr>
<tr>
<td>SDAREA</td>
<td>FINVAR</td>
</tr>
<tr>
<td>DIABAR</td>
<td>FINVAR</td>
</tr>
<tr>
<td>HCALL</td>
<td>COPPIE</td>
</tr>
<tr>
<td>HOV_GE</td>
<td>HCALL</td>
</tr>
<tr>
<td>SFCALL</td>
<td>COPPIE</td>
</tr>
<tr>
<td>STOLSF</td>
<td>SFCALL</td>
</tr>
<tr>
<td>GVCALL</td>
<td>COPPIE</td>
</tr>
<tr>
<td>STOLGV</td>
<td>GVCALL</td>
</tr>
<tr>
<td>WKCALL</td>
<td>COPPIE</td>
</tr>
<tr>
<td>JIEPIE</td>
<td>WKCALL</td>
</tr>
<tr>
<td>STOLWK</td>
<td>WKCALL</td>
</tr>
<tr>
<td>RCSCAL</td>
<td>COPPIE</td>
</tr>
<tr>
<td>RCSIND</td>
<td>RCSCAL</td>
</tr>
<tr>
<td>PIEP</td>
<td>COPPIE</td>
</tr>
<tr>
<td>JIETAB</td>
<td>PIEP</td>
</tr>
<tr>
<td>CENTAR</td>
<td>FINVAR</td>
</tr>
<tr>
<td>CROSSIN</td>
<td>PLANMO, SDAREA</td>
</tr>
<tr>
<td>DIST</td>
<td>SDAREA</td>
</tr>
<tr>
<td>INTEGRA</td>
<td>SDAREA</td>
</tr>
<tr>
<td>LINECRO</td>
<td>SDAREA, CROSSIN</td>
</tr>
<tr>
<td>LINTERP</td>
<td>FINVAR, SDAREA, CENTAR</td>
</tr>
<tr>
<td>PERPDIS</td>
<td>SDAREA</td>
</tr>
<tr>
<td>POLYAR</td>
<td>FINVAR, CENTAR</td>
</tr>
<tr>
<td>RTOTAL</td>
<td>JIETAB</td>
</tr>
<tr>
<td>SSEN</td>
<td>FINVAR</td>
</tr>
</tbody>
</table>