NASA CONTRACTOR REPORT

AERODYNAMIC DESIGN AND ANALYSIS SYSTEM
FOR SUPersonic AIRCRAFT

Part 3 - Computer Program Description

W. D. Middleton, J. L. Lundry, and R. G. Coleman

Prepared by
BOEING COMMERCIAL AIRPLANE COMPANY
Seattle, Wash. 98124
for Langley Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • MARCH 1975
NASA CR-2522

2. Government Accession No.  

3. Recipient's Catalog No.  

4. Title and Subtitle  
Aerodynamic Design and Analysis System for Supersonic Aircraft.  
Part 3--Computer Program Description

5. Report Date  
March 1975

6. Performing Organization Code  

7. Author(s)  
W. D. Middleton, J. L. Lundry, and R. G. Coleman

D6-41769

9. Performing Organization Name and Address  
Boeing Commercial Airplane Company  
P.O. Box 3707  
Seattle, Washington 98124

10. Work Unit No.  

11. Contract or Grant No.  
NAS1-12052

12. Sponsoring Agency Name and Address  
National Aeronautics and Space Administration  
Washington, D.C. 20546

13. Type of Report and Period Covered  
Contractor Report  
Jan. 1973--Nov. 1974


15. Supplementary Notes  
Three of three final reports

16. Abstract  
An integrated system of computer programs has been developed for the design and analysis of supersonic configurations. The system uses linearized theory methods for the calculation of surface pressures and supersonic area rule concepts in combination with linearized theory for calculation of aerodynamic force coefficients. Interactive graphics are optional at the user's request.

The description of the design and analysis system is broken into three parts:  
Part 1--General Description and Theoretical Development  
Part 3--Computer Program Description

This part contains schematics of the program structure and describes the individual overlays and subroutines.

17. Key Words (Suggested by Author(s))  
Wing design  
Configuration analysis  
Wave drag  
Pressure distribution

18. Distribution Statement  
Unclassified--Unlimited  
New Subject Category 02

19. Security Classif. (of this report)  
Unclassified

20. Security Classif. (of this page)  
Unclassified

21. No. of Pages  
96

22. Price*  
$4.75

*For sale by the National Technical Information Service, Springfield, Virginia 22151
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>2.0 DISCUSSION</td>
<td>3</td>
</tr>
<tr>
<td>2.1 Executive Module</td>
<td>6</td>
</tr>
<tr>
<td>2.2 Geometry Module</td>
<td>8</td>
</tr>
<tr>
<td>2.3 Skin Friction Module</td>
<td>20</td>
</tr>
<tr>
<td>2.4 Near-Field Wave Drag Module</td>
<td>25</td>
</tr>
<tr>
<td>2.5 Wing Design and Optimization Module</td>
<td>45</td>
</tr>
<tr>
<td>2.6 Lift Analysis Module</td>
<td>62</td>
</tr>
<tr>
<td>3.0 REFERENCES</td>
<td>85</td>
</tr>
<tr>
<td>APPENDIX A—Interactive Graphics Subroutines</td>
<td>87</td>
</tr>
</tbody>
</table>
AERODYNAMIC DESIGN AND ANALYSIS SYSTEM
FOR SUPERSONIC AIRCRAFT

PART 3 - COMPUTER PROGRAM DESCRIPTION

W. D. Middleton, J. L. Lundry, and R. G. Coleman
Boeing Commercial Airplane Company

1.0 SUMMARY

An integrated system of computer programs has been developed for the design and analysis of supersonic configurations.

The system consists of an executive driver and seven basic computer programs including a plot module, which are used to build up the theoretical force coefficients of a selected configuration.

Documentation of the system has been broken into three parts:

Part 1 - General Description & Theoretical Development
Part 2 - User's Manual
Part 3 - Computer Program Description

This part, the computer program description, contains schematics of the program structure and written descriptions of the individual overlays and subroutines.

Interactive graphics for use with the system are optional, employing the NASA-LRC CRT display and associated software.

The computer program is written in FORTRAN IV for a SCOPE 3.0 or KRONOS 2.0 operating system and library file. It is designed for the CDC 6600 series of computers and executes in OVERLAY mode. The system requires approximately 110000 (octal) central memory words and uses seven peripheral disc files in addition to the input and output files.
2.0 DISCUSSION

A schematic of the design and analysis system overlay structure is shown in figure 2.0-1. The system is a single overlaid program, with the executive driver as the main overlay and the basic programs as primary overlays. The basic programs manipulate input (geometry module), draw a picture of the configuration (plot module), or perform design or analysis calculations.

The format of the computer program documentation is to present schematics or block diagrams of the major program structure, together with subroutine descriptions, for each module developed under the design and analysis system contract. The plot and far-field wave drag modules are not included in this procedure, since they are described in other NASA documentation (references 1 and 2).

The description of the overlay structure follows the convention of labeling overlays with octal numbers, but calling them with their decimal equivalents.

A typical test case and associated program output is given in the User's Manual (part 2).

File Usage

File usage in the system is assigned as follows:

1  basic geometry storage
2  interface file
3  restart data
5  card input
6  output
9, 10, 11, 12 storage files for far-field wave drag module
9, 11, 12 storage files for plot module

Program Structure

A block diagram of the design and analysis system is shown in figure 2.0-2. The largest element of the system with the NASA-LRC graphics software attached occurs with the geometry display module loaded, and is approximately 110000\(8\) (octal). A "stripped" version of the system (graphics software excluded) would have its largest core requirement with the lift analysis module loaded, and would be approximately 70000\(8\).

These core sizes are for an absolute version of the program, without the operating system loader.
FIGURE 2.0-1.—INTEGRATED SUPersonic DESIGN AND ANALYSIS SYSTEM
* Largest core requirement with graphics attached ≈ 110000\(\text{g}\).

** Largest core requirement without graphics software ≈ 70000\(\text{g}\).

FIGURE 2.0.2.—SYSTEM STRUCTURE
2.1 EXECUTIVE MODULE

The executive level (0,0 overlay) is used to read executive control cards and request execution of the basic programs (primary overlays) as instructed. The executive cards are described in the user's manual (part 2) and summarized under subroutine CHECKIN.

Program SUPERD

PURPOSE: To read program executive cards and call primary overlays.

METHOD: SUPERD is the program name assigned to the executive level (SDA, 0, 0). It reads executive cards, calls subroutine CHECKIN to find the corresponding overlay number, calls the geometry module to read or sort input, then calls the appropriate primary overlay for problem execution.

CRT is a special executive card used only to turn on (or off) the graphics routines as described in the user's manual.

INPUT: Executive cards (see user's manual)

SUBROUTINES CALLED: CHECKIN

Program CHECKIN

PURPOSE: To identify primary overlay number corresponding to executive card.

METHOD: The overlay number for the different primary overlays is given by variable IPRG, in common block/SAVI/. CHECKIN sets the value of IPRG by finding the executive card word corresponding to IPRG. The correspondence is:

<table>
<thead>
<tr>
<th>EXECUTIVE_CARD</th>
<th>IPRG</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLØT</td>
<td>2</td>
</tr>
<tr>
<td>FFWD</td>
<td>3</td>
</tr>
<tr>
<td>SKFR</td>
<td>4</td>
</tr>
<tr>
<td>GEØM</td>
<td>5</td>
</tr>
<tr>
<td>NPWD</td>
<td>6</td>
</tr>
<tr>
<td>ANLØ</td>
<td>7</td>
</tr>
<tr>
<td>WDEM</td>
<td>8</td>
</tr>
<tr>
<td>FSUP</td>
<td>9</td>
</tr>
<tr>
<td>WGUP</td>
<td>10</td>
</tr>
</tbody>
</table>
USE: CALL CHECKIN

INPUT: Executive control card word (ISKIP)

SUBROUTINES CALLED: None

Function TBLU1
Integer Function L00KUP

PURPOSE: To perform linear or second order interpolation from one-dimensional array.

METHOD: TBLU1 and the associated integer function L00KUP are general purpose interpolation routines, for either linear or second order interpolation. The program call is:

\[ Z = TBLU1( X, XX, Y, MD, N ) \]

where

- \( X \) independent variable
- \( XX \) array containing independent variable
- \( Y \) array containing dependent variable
- \( MD \) code defining interpolation type
  - 1 = linear
  - 2 = second order
- \( N \) number of variables in array \( XX \) or \( Y \)
- \( Z \) dependent variable (answer)

Array \( XX \) must be monotonically increasing.

USE: \[ Z = TBLU1( X, XX, Y, MD, N ) \]

INPUT: As described above

SUBROUTINES CALLED: TBLU1 calls associated subroutine L00KUP
2.2 GEOMETRY MODULE

The geometry module is primary overlay 5. It contains subprograms to read, store, or display the configuration geometry, and to set up the input for the other basic programs. A schematic of the geometry module is shown in Figure 2.2-1.

Program PDPACK

PURPOSE: To route geometry handling requests from the executive to geometry routines, based on executive control cards.

METHOD: Program PDPACK is the primary level of the geometry module. It is entered from the executive to store, add, or change input data; to update wing camber surface or fuselage basic geometry; or to enter the interface routines to set up input data for the other modules. A schematic of PDPACK is shown in figure 2.2-2.

The executive control cards corresponding to variables in PDPACK are as follows:

<table>
<thead>
<tr>
<th>EXECUTIVE CARD</th>
<th>PDPACK VARIABLE</th>
<th>PRINCIPAL PROGRAMS OR SUBROUTINES CALLED</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEØM NEW</td>
<td>IFIRST=0</td>
<td>EDITS</td>
</tr>
<tr>
<td>GEØM</td>
<td>IFIRST=1</td>
<td>INPUTS, EDITS</td>
</tr>
<tr>
<td>PLØT</td>
<td>IPRG=2</td>
<td>INPUTS, GEØMPLT</td>
</tr>
<tr>
<td>FFWD</td>
<td>IPRG=3</td>
<td>INPUTS, GEØM80</td>
</tr>
<tr>
<td>SKFR</td>
<td>IPRG=4</td>
<td>INPUTS, GEØM158</td>
</tr>
<tr>
<td>NFWD</td>
<td>IPRG=6</td>
<td>INPUTS, GEØM916</td>
</tr>
<tr>
<td>ANLÆ</td>
<td>IPRG=7</td>
<td>INPUTS, GEØM201</td>
</tr>
<tr>
<td>WDEÆ</td>
<td>IPRG=8</td>
<td>INPUTS, GEØM253</td>
</tr>
<tr>
<td>FSUP</td>
<td>IPRG=9</td>
<td>INPUTS, FUSUPD</td>
</tr>
<tr>
<td>WGUP</td>
<td>IPRG=10</td>
<td>INPUTS, NEWCAM</td>
</tr>
</tbody>
</table>

Configuration geometry is read from cards in program EDITS and stored on tape 1 when the geometry module is not in core. Program INPUTS retrieves the geometry from tape 1 when the geometry module is called. Subroutines WRGEØM and PRINTG write the configuration geometry on tape 1 for storage or print the geometry onto the output tape, respectively.
FIGURE 2.2-2.—PROGRAM PDPACK SCHEMATIC
The interface programs write data onto tape 2, which becomes the input tape for the individual basic modules.

**USE:**

CALL OVERLAY (SDA, 5, 0, 0)

**INPUT:**

Executive card variables listed above.

**SUBROUTINES CALLED:**

See schematic on page 10.

---

Subroutine **WRGEOM**

**PURPOSE:**

To write geometry data onto tape 1 for storage, or onto tape 2 for plot program or far-field wave drag program.

**METHOD:**

Subroutine **WRGEOM** uses the input format of the NASA-LRC plot program and is used to write the basic geometry data onto tape 1 when the geometry module leaves core, or to write the geometry data onto the interface tape (2) as a part of the plot or far-field wave drag program interfaces.

**USE:**

CALL **WRGEOM**

**INPUT:**

Configuration geometry (see user's manual)

**SUBROUTINES CALLED:**

None

---

Subroutine **2P#D**

**PURPOSE:**

To calculate the distance between the nacelle centerline and the local wing camber surface.

**METHOD:**

The basic configuration geometry input (read in EDITS) allows the nacelles to be input with either of two dimensions: in the configuration coordinates, or with measured to the local wing camberline. Subroutine **2P#D** is used, regardless of which definition is input, to calculate the other definition and store it in the basic geometry. If the nacelle origin is forward or aft of the wing planform, the distance is measured from the wing leading edge or trailing edge (whichever is closer) at the same Y station.
As a special case, $\text{2P0D}$ is also used to compute the distance between the fuselage centerline and the wing camber line at the side of the fuselage (in program GEOM916).

**USE:**

CALL $\text{2P0D}(\text{NN})$

**INPUTS:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{NN}$</td>
<td>Nacelle origin index</td>
</tr>
<tr>
<td>$\text{P0D0RG}$</td>
<td>Nacelle origin data</td>
</tr>
<tr>
<td>$\text{XAF}$</td>
<td></td>
</tr>
<tr>
<td>$\text{WAF0RG}$</td>
<td>Wing definition (planform, thickness, camberline)</td>
</tr>
<tr>
<td>$\text{T3RD}$</td>
<td></td>
</tr>
</tbody>
</table>

**SUBROUTINES CALLED:** None

Subroutine **PRINTG**

**PURPOSE:** To write configuration geometry onto output tape (tape 6).

**METHOD:** After configuration geometry is read or changed, subroutine PRINTG is used to write the current geometry definition. In the case where the fuselage or wing camber surface is updated (executive cards FSUP or WGUP), only the fuselage or camber surface is output. This is controlled by variable ISKIP1 (=98 for fuselage only, =99 for camber surface only).

**USE:**

CALL PRINTG

**INPUT:**

Geometry definition (see user's manual)

ISKIP1

**SUBROUTINES CALLED:** None

Subroutine **NEWCAM**

**PURPOSE:** To interpolate a wing camber surface for basic geometry definition ($T3RD$) from wing design or analysis definition ($W3RD$).

**METHOD:** The camber surface definition generated or used by the wing design or analysis modules, called $W3RD$, may be different from the basic geometry definition ($T3RD$). This can happen either because the definition is being created by the design program,
or because the analysis program input option for W2ØRD was used. NEWCAM linearly interpolates for TØRD from WØRD, and is called by program PDPACK if executive card WGUP is read. NEWCAM is also automatically called by the analysis program interface if the WØRD option is used.

USE:
CALL NEWCAM (FACTØR)

INPUT:
FACTØR Multiplier used to scale camber surface ordinates, if desired.
NØPCT, P,
JBYMAX, Camber surface definition WØRD contained in common block /CAMBER/
Y,
WØRD, IFØC
XAF, Camber surface definition (TØRD) of basic geometry contained in common block /WING/
WAFØRG,
TØRD

SUBROUTINES CALLED:
INTERP

Subroutine INTERP

PURPOSE: To interpolate two-dimensional array.

METHOD: INTERP is used to linearly interpolate WØRD definition (used in conjunction with subroutine NEWCAM).

USE:
CALL INTERP (A, Ø, N, X, Y, M)

INPUT
A First array location for abscissa values
Ø First array location for ordinate values
N Number of values in A and Ø arrays
X Array of abscissa for ordinate outputs
M Number of values in X and Y arrays output
Y Array of calculated ordinates.

SUBROUTINES CALLED: None
Program GE0MPLT

PURPOSE: To write input tape for PLOT module.

METHOD: GE0MPLT is the interface program between the executive and the PLOT module, entered when the executive card PL0T is read. The interface reads the plot title card and configuration codes, then uses WRGE0M to write the corresponding basic geometry data onto tape 2 (the program input tape). GE0MPLT then reads the plot view cards and writes them onto tape 2.

USE: CALL OVERLAY (SDA, 5, 2, 0)

INPUT: See user's manual for input description

SUBROUTINES CALLED: WRGE0M

Program GE0M80

PURPOSE: To write input tape for far-field wave drag module.

METHOD: GE0M80 is the interface program between the executive and the far-field wave drag program, entered when the executive card FFWD is read. The interface reads a title card and the configuration codes, then uses WRGE0M to write the corresponding basic geometry data onto tape 2. GE0M80 then reads the case and restraint cards (if used) and writes them onto tape 2.

USE: CALL OVERLAY (SDA, 5, 3, 0)

INPUT: See user's manual for input description

SUBROUTINES CALLED: WRGE0M

Program GE0M158

PURPOSE: To write input tape for skin friction module.

METHOD: GE0M158 is the interface program between the executive and the skin friction drag module, entered when the executive card SKFR is read. It reads title information, flight conditions, and configuration data, and structures those inputs together with basic geometry to write the input tape (tape 2) for the skin friction module.
USE: CALL OVERLAY (SDA, 5, 4, 0)

INPUT: See user's manual for input description

SUBROUTINES CALLED: None

Program GEØM916

PURPOSE: To write input tape for near-field wave drag module.

METHOD: GEØM916 is the interface program between the executive and the near-field wave drag module, entered when the executive card NFWD is read. It reads title information, Mach number, etc., and structures those inputs together with basic geometry to write the input tape (tape 2) for the near-field wave drag module.

The near-field program considers the configuration to be uncambered; the wing and fuselage have thickness but have a flat mean-line. Wing height in the side of the fuselage is preserved, however, being the distance from the fuselage centerline to the wing at the side of the fuselage. In GEØM916, the wing height dimensions are computed by a special use of_BPØD, in which the fuselage centerline is treated as a nacelle origin in a series of calculations.

The nacelle B dimension used in the near-field program is the distance from the local wing camberline.

USE: CALL OVERLAY (SDA, 5, 6, 0)

INPUT: See user's manual for input description.

SUBROUTINES CALLED: _BPØD, NEWAREA

Subroutine NEWAREA

PURPOSE: To check for and remove "steps" in input fuselage area definition corresponding to inlet or exit stream tubes.

METHOD: The basic geometry fuselage definition allows for input of the fuselage in four segments. If the
first cross-section of segment 2 is input at the same X station as the last cross-section of segment 1, and the cross-sectional areas of the two inputs are different, then the far-field wave drag program extends a streamtube to account for the area "step", so that no area discontinuity occurs in the wave drag calculations. This can occur, also, between other segments. The near-field wave drag fuselage calculations cannot accommodate an area step at a fuselage station other than the first or last. If one occurs, subroutine NEWAREA removes the step by collapsing the fuselage to a solid body having the same area growth.

NEWAREA also reduces the fuselage area definition to 50 values in X, Z and area if more than 50 total values were input in all fuselage segments, since the near-field program allows only 50 points.

USE:

```
CALL NEWAREA (X, Z, A, L)
```

INPUT:

- X: Output array of fuselage X values for analysis program.
- Z: Output array of fuselage camberline Z values.
- A: Output array of fuselage cross-sectional areas.
- L: Number of values in X, Z, or A arrays

Common block containing fuselage basic geometry definition.

SUBROUTINES CALLED: None

PROGRAM: GEOM201

PURPOSE: To write input tape for lift analysis module.

METHOD: GEOM201 is the interface program between the executive and the lift analysis module, entered when the executive card ANL2 is read. It reads title information, Mach number, etc., and structures those inputs together with basic geometry to write the input tape (tape 2) for the lift analysis module.

The lift analysis interface allows the wing camber surface to be read as the WBRD definition, or passed in through common block /CAMBER/.
addition to the basic geometry definition. If \texttt{W2ORD} is used, the basic geometry is automatically updated by means of subroutines \texttt{NEWCAM} and \texttt{WRGEM}.

In addition, since the lift analysis program logic does not permit step discontinuities in fuselage area between the most forward and aft X stations, subroutine \texttt{NEWAREA} is used to prepare the fuselage cross-sectional area definition. (\texttt{NEWAREA} is described in connection with \texttt{GEM916}).

The nacelle origin \( z \) dimension is the vertical distance to the local wing camber line.

\textbf{USE:} \\
\texttt{CALL OVERLAY (SDA, 5, 7, 0)}

\textbf{INPUT:} \\
See user's manual for input description

\textbf{SUBROUTINES CALLED:} \\
\texttt{NEWAREA, NEWCAM, WRGEM}

\textbf{Program GEM253}

\textbf{PURPOSE:} \\
To write input tape for wing design module.

\textbf{METHOD:} \\
\texttt{GEM253} is the interface between the executive and the wing design module, entered when the executive card \texttt{WDE2} is read. It reads title information, Mach number, loadings data, etc., and structures those inputs together with basic geometry to write the input tape (tape 2) for the wing design module.

The wing design program may employ pressure fields generated by the near-field wave drag module or analysis module. The existence of these pressure fields is tested in the corresponding common blocks. This involves the codes \texttt{BGDCFX}, \texttt{BGDUPX}, and \texttt{CPNACX}. The actual use of these pressure fields is controlled by input of the corresponding loading numbers.

The nacelle \( z \) origin is the vertical dimension to the local wing camber surface.

The restart data deck is substantial. Logic for this part of the interface is copied from the wing design program input.

\textbf{USE:} \\
\texttt{CALL OVERLAY (SDA, 5, 8, 0)}

\textbf{INPUT:} \\
See user's manual for input description
SUBROUTINES CALLED: None

Program FUSUPD

PURPOSE: To update the fuselage definition in the basic geometry to the optimum area distribution generated by the far-field wave drag program.

METHOD: The far-field wave drag program contains a fuselage optimization feature, which area-rules the fuselage subject to input constraints. FUSUPD uses the optimized definition to update the basic fuselage geometry, and is accessed by means of the executive card PSUP.

FSUP first reads the value of $0PH0W$, which controls the optimization interpolation. ($0PH0W = -$ , interpolate at original fuselage $X$ stations; $0PH0W = +$, interpolate at 50 equally spaced $X$ stations). It then changes the basic geometry definition to the optimum body area distribution, contained in common block/$0PBOD$/ . The original fuselage definition is preserved. Fuselage perimeters are proportioned to the new area distribution.

USE: CALL $\odot$OVERLAY (SDA, 5, 9, 0)

INPUT: $0PH0W$ Interpolation code
$X0P$ Optimum fuselage definition. $X0P = X$
$A0P$ array, $A0P = area$ array, $J0P =$
$J0P$ number of $X0P$ or $A0P$ values.

SUBROUTINES CALLED: TBLU1

Program INPTS

PURPOSE: To read basic geometry data from tape into core.

METHOD: Program INPTS is used to reread the configuration basic geometry back into core when the geometry module is accessed. INPTS uses the same format as program WRGEQM, which was used to store the basic geometry on tape 1.

USE: CALL $\odot$OVERLAY (SDA, 5, 10, 0)

INPUT: Tape 1
Program EDITS

PURPOSE: To read configuration geometry from cards.

METHOD: Program EDITS is used to read the configuration geometry from cards, and is accessed by either the GEOM NEW or GEOM executive cards. (GEOM NEW zeroes all the configuration J1, J2, etc., codes, so that EDITS reads all new data; GEOM preserves all existing codes, so that new geometry read replaces or adds to existing geometry read by INPTS).

The input format of program EDITS is basically that if the NASA-LRC plot program, with a few additional input variables added as noted in the user's manual.

USE: CALL OVERLAY (SDA, 5, 11, 0)

INPUT: See user's manual for input description

SUBROUTINES CALLED: None

SUBROUTINES CALLED: BP0D
2.3 SKIN FRICTION MODULE

The skin friction module is primary overlay U. Its principal subprograms calculate the wetted areas of the configuration, and then the skin friction drag for a series of input flight conditions. A schematic of the skin friction module is shown in figure 2.3-1.

Program TEA-158A

PURPOSE: To read configuration data, flight conditions, and calculate wetted areas and lengths to be used in skin friction drag calculations.

METHOD: Program first reads all input data, which consists of configuration geometry and either Mach number-altitude or Mach number-Reynolds number flight conditions.

For each configuration component, the corresponding wetted areas and reference lengths are computed. In the case of the wing or canard(s) of fin(s), the parts are broken into strips to permit more accuracy in determining an average skin friction coefficient. (The wing is broken into approximately 50 strips, canard or fin into 10 strips.)

The wetted areas and references lengths are passed to subroutine DRAG for the skin friction coefficient calculations.

USE: CALL OVERLAY (SDA, 4, 0)

INPUT: Input is read from interface tape (tape 2), set up by skin friction interface in geometry module.

SUBROUTINES CALLED: DRAG

Subroutine DRAG

PURPOSE: To calculate configuration skin friction coefficients and print answers for all input flight conditions.

METHOD: Given the set of configuration wetted areas and input flight conditions, the skin friction coefficients for each component are computed and summed to obtain the total skin friction drag. If
FIGURE 2.3.1. SKIN FRICTION MODULE SCHEMATIC

- Displays results on CRT
- TEA-158 (4,0)
  - Reads input
  - Computes wetted areas and reference lengths
- DRAG
  - Calculates friction drag for input flight conditions
- FICT
  - Skin friction coefficient from T' method
- SKIN
- ATMO 62
  - Standard atmosphere data
the graphics routines are activated, the friction drag coefficients are also printed on the CRT.

**USE:**

CALL DRAG

**INPUT:**

Input is passed in common blocks from program TEA-158A

**PRINCIPAL VARIABLES:**

Same as program TEA-158A, plus:

- **SCAM** Scale factor to convert configuration reference lengths to feet for use in subroutine FICT
- **SREF** Wing reference area
- **SWETT** Total configuration wetted area
- **CDFT** Total skin friction drag coefficient
- **AM**, **AL**, **DELT**, **SCAMØD**
  - AM, Mach number - altitude flight
  - AL, condition input AM = Mach number, AL = altitude (feet), DELT = temperature deviation from standard (°F), SCAMØD = input scale factor.
- **AM**, **RNPFL**, **SCAMØD**, **TØTEM**
  - AM, Mach number - Reynolds number
  - RNPFL, condition input. RNPFL = Reynolds number per foot/1,000,000. TØTEM = total temperature (°Rankine)
- **SWETRB** Fuselage wetted area
- **FUSL** Fuselage reference length
- **SWETB** Fuselage wetted area corrected for overlap areas of wing, fin(s), and canard(s)
- **AREAJ** Planform area of a wing strip
- **WINGL** Length of wing strip
- **LCOUNT** Total number of wing strips
- **SWETW** (nominally 50) total wing wetted area
- **TSWTNA** Wetted area of nacelle(s) at first input origin, corresponding length.
- **TPØDL** Total wetted area for all nacelles
- **NPØD** Number of nacelle origins
- **SWTPN** Planform area of a fin strip (10 total strips), corresponding strip length, number of fins
- **NFIN** Corresponding strip length
- **TCCAN** Number of canards
- **NCAN** Number of canards

22
SWETXP  Wetted area of an arbitrary configuration part, corresponding
RLXP    reference length, number of
NXTPT   arbitrary parts.

OUTPUT:  Output consists of flight conditions, wetted area and skin friction coefficient buildups, total skin friction drag coefficient and configuration wetted area.

SUBROUTINES CALLED:  FICT, DISP158

Subroutine FICT

PURPOSE:  To calculate the turbulent skin friction drag coefficient for a given reference length, Mach number, Reynolds number, and total temperature.

METHOD:  The skin friction coefficient is calculated from the T' method described in the aerodynamic theory document (part 1).

USE:  CALL FICT (AMX, ALX, EEL, C)

INPUT:  AMX  Mach number
ALX  Altitude, feet
EEL  Reference length

PRINCIPAL VARIABLES:  TI  Total temperature, degrees Rankine
SCAM  Scale factor to convert input reference length to feet
SI  Free stream Reynolds number

OUTPUT:  C  Skin friction coefficient

SUBROUTINES CALLED:  ATMØ62, SKIN

Subroutine ATMØ62

PURPOSE:  To provide standard atmospheric data

METHOD:  Program uses 1962 standard altitude definition (reference 3).

USE:  CALL ATMØ62 (Z, TEMP, SIGMA, AX)
INPUT:  
Z  Geometric altitude, feet

OUTPUT:  
TEMP  Temperature, degrees centigrade
SIGMA  Density ratio
AX  Speed of sound, knots

SUBROUTINES

CALLED:  None

Subroutine SKIN

PURPOSE:  To iterate for skin friction coefficient.

METHOD:  Program is used to solve the Karman-Schoenherr equation.

using \[ \frac{0.242}{\sqrt{\text{CF}}} = \log_{10} (\text{CF} \times \text{REY}) \]

value for CF. Solution is iterative, and is satisfied when successive iterations agree within .0001 per cent. A maximum of 50 iterations is allowed.

USE:  CALL SKIN (REY, CF)

INPUT:  REY  Reynolds number

OUTPUT:  CF  Skin friction coefficient

SUBROUTINES CALLED:  None

ERROR RETURN:  Program uses 50th iteration for CF if convergence does not occur, and prints error message.
2.4 NEAR-FIELD WAVE DRAG MODULE

The near-field wave drag module is primary overlay 6. It contains principal subprograms to calculate the near-field pressure data and drag coefficients and to display the calculated results. A schematic of the principal program structure is given in figures 2.4-1 and 2.4-2.

Program TEA-356

PURPOSE: Near-field wave drag primary overlay

METHOD: Program TEA-356 is the primary level of the near-field wave drag program. It calls the input overlay (1), and contains the Mach number loop which calls the main program (overlay 2) and the graphics display (overlay 3).

USE: CALL OVERLAY (SDA, 6, 0, 0)

INPUT: Executive card NFWD

SUBROUTINES CALLED: Overlays 6,1 to 6,3

Program P916

PURPOSE: To perform thickness pressure calculations in near-field wave drag module.

METHOD: Program P916 is the main program of the near-field wave drag module. It solves for the thickness pressure distribution on the surface of an arbitrary wing-fuselage-nacelle configuration, and integrates the pressures over the surface to obtain wave drag.

The equations used in the thickness pressure solution are given in the theory document (part 1). A consistent nomenclature is used, as possible, between the theory equations and the Fortran variable names.

The major subprogram of P916 is NACPF, which calculates the nacelle thickness pressures and all nacelle interference terms except for the nacelle-on-wing and wing-on-nacelle terms.

The principal program logic is as follows:

1) Convert planform geometry into wing
Wing thickness pressures and drag coefficient summary

Nacelle terms (See figure 2.4-2)

TRNSFM
INTSEC
BODY
PSIG
DZCALC
GETNCP
GRAB
WONAC
BDRAG
ADDUM

Setup grid system
Wing-fuselage intersection
Fuselage F(y) and thickness pressure
Fuselage pressure field
Grid system slopes
Nacelle pressure field signature
Wing or fuselage pressure field interpolation
Wing-on-nacelle term
Fuselage drag
Nacelle drag summary

FIGURE 2.4-1.—NEAR-FIELD WAVE DRAG MODULE SCHEMATIC
Y stations for nacelle pressure field

Separate nacelles, above and below wing

Nacelle wave drag, \( F(y) \)

Calculate nacelle pressure field acting on wing

Fuselage-on-nacelle term

Nacelle-on-fuselage term

Nacelle-on-nacelle term

FIGURE 2.4-2.—NACELLE TERMS SCHEMATIC
2) Locate wing-fuselage intersection (INTSEC)

3) Calculate nacelle data (NACPF)

4) Calculate fuselage pressure field acting on wing. This is done for each selected semi-span station (TYB2) in loop DØ 130. The resultant fuselage pressure field is stored in PBWG, which is printed in loop DØ 150.

5) Calculate wing thickness pressure field. This is done for the selected semi-span stations in loop DØ 830. The semi-span element row is NSTAR and the associated x elements are LSTAR. For each NSTAR and LSTAR, the upstream region of integration contains elements located at N and LVAR. The local wing slopes are DDX, contained in array TBC. The loop DØ 450 performs the influence function times slope calculation described in the theory document. The velocity potential (PHI) is then calculated, and differenced to get pressure coefficient (CP).

After all pressure coefficients are calculated at a given semi-span station, they are smoothed in loop 710, resulting in CPAVG.

Calculation of the drag terms (pressure times wing slope) employs these variables names:

- **DBWG**: fuselage pressure acting on wing
- **DAVG**: wing thickness pressure
- **DNAC**: nacelle pressure acting on wing.

6) Interpolate wing thickness pressures for output. This is done at the selected semi-span stations in loop DØ 770. The wing thickness pressures are then output in loop DØ 840. At the same time, the common block containing the wing thickness pressures, /PLIM/, is completed to make it self-sufficient for use in other modules, where:

- **FYB2**: per cent semi-span array, MYFR values
- **TNMPC**: per cent chord array, NMPC values
- **VAR(X,Y)**: thickness pressure array

The wing thickness pressure plus fuselage pressure field is then summed and output in loop DØ 860.
7) Integrate drag terms spanwise to get total wing drag. This is done in loop DØ 900.

8) Calculate wing thickness pressure on nacelle volume term (subroutine WØNAC).

9) Calculate fuselage pressure drag plus wing thickness pressure on fuselage drag terms. This is done in subroutine BDRAG and results in drag terms DBD and DWØB, respectively.

10) Convert drag terms to coefficient form, output wing section drag coefficients, nacelle drag coefficients, and final drag summary. This is done starting at loop DØ 920, using the following nomenclature:

- YB2: semi-span fraction
- CHORD: local wing chord
- CAS: wing section drag coefficient
- BEF: section drag of fuselage on wing
- CNACS: section drag of nacelle on wing
- CINT: sum of CAS, BEF, CNACS
- DFR: fraction of total wing drag occurring at YB2
- CDAVG: wing thickness drag coefficient
- CDB: fuselage thickness drag coefficient
- CDBOW: interference drag coefficient of fuselage on wing
- CDWOB: interference drag coefficient of nacelle on wing
- CDTOT: total drag coefficient

The nacelle drag coefficient summary is printed in subroutine ADDUM.

USE: CALL ØVERLAY (SDA, 6, 1, 6HRECALL)

INPUT: MK Mach number index variable (from overlay 6, 0). Configuration data from overlay 6, 1.

SUBROUTINES CALLED: See schematic, figure 2.4-1.

Subroutine ADDUM

PURPOSE: To sum nacelle drag terms, convert them to coefficient form (based on wing reference area) and write them onto the output file.
METHOD: Subroutine NACPF generates all nacelle thickness drag terms except for the wing on nacelle and nacelle on wing terms. These are saved in common block /NDRAG/ and printed by ADDUM when the main program (P916) writes the final drag summary. The drag coefficient nomenclature is as follows:

- TNISOL: isolated nacelle wave drag coefficient
- TFB0N: interference drag coefficient of fuselage on nacelles
- TPN0B: interference drag coefficient of nacelle on fuselage
- TPNON: interference drag of other nacelles on nacelle
- TFIMAG: "image" drag coefficient of nacelle on itself
- TP0IMG: "image" drag coefficient of other nacelles
- TDW0N: interference drag coefficient of wing on nacelles

USE: CALL ADDUM (SUMM)

INPUT: SUMM Sum of nacelle drag coefficients

SUBROUTINES CALLED: None

Subroutine BDRAG

PURPOSE: To integrate fuselage thickness pressures over fuselage to get wave drag, and calculate wing-on-fuselage interference drag term.

METHOD: The fuselage is broken into TNCUT lengthwise strips, and 5 radial segments (per half section) per strip. The fuselage thickness pressures and wing pressures acting on the carry-over region covered by the fuselage are integrated over the fuselage segments. Wing pressures are transferred aft along Mach lines to obtain the point of application on the fuselage.

USE: CALL BDRAG (TNCUT)

INPUT: TNCUT Number of fuselage strips. (Set at 50 in main program.
TNMPC, Wing pressure field extended along fuselage centerline. TPWE = pressure coefficient array, TNMPC = X
SUBROUTINES CALLED: GRAB, TBLU1

Subroutine BØDY

PURPOSE: To calculate the Whitham \( F(y) \) function for an arbitrary body of revolution, then compute surface pressure distribution, wave drag, and wetted area.

METHOD: Subroutine BØDY first computes the Whitham \( F(y) \) function, according to the Stieltjes integral formulation. The equations used are summarized in the theoretical document (part 1). The \( F(y) \) function is computed at \( TNCUT+1 \) \( X \)-stations (where \( TNCUT \) is an input in the main program, usually 50). To remove minor irregularities in the final \( F(y) \) function, extra points are inserted between each of the \( X \)-stations, \( F(y) \) is computed there also, and a five point smoothing formula applied.

The \( F(y) \) function is also computed at \( X \)-stations downstream of the last body station, to define the "tail" on the \( F(y) \) function which is needed in the area-balancing technique for pressure coefficient. The loop for the \( F(y) \) equations begins at statement 60, which results in the unsmoothed \( F(y) \) array stored in TFY, with corresponding \( y \) values in BTØ, JFY total values. After smoothing and extending the \( F(y) \) function aft of the body, the final \( F(y) \) function is stored in TFTAU, with corresponding \( y \) values in TTAU, JSTØ total values.

The body surface pressure calculations loop starts at DØ 330, using the equations given in the theory document (part 1). The body is again broken into \( TNCUT+1 \) segments, and the resulting thickness pressures stored in TCPBY, with corresponding \( X \) values in TXCPBY. The wetted area calculation and the integration of the thickness pressures to get wave drag is performed in loop DØ 340.
Subroutine **BØDY** will handle bodies having open front or aft ends, but assumes smooth geometry in-between.

**USE:**

```plaintext
CALL BØDY (TXB, TRB, TNXB, TNCUT, TTAU, TPTAU, JSTØ, WAVE, SNWET)
```

**INPUT:**

- **TXB**: Body definition. TXB = x array,
- **TRB**: TRB = radii array, TNXB values in
- **TNXB**: Number of body segments in F(y)
- **TNCUT**: and thickness pressure calculations
- **TTAU**: F(y) function. TTAU = y,
- **TPTAU**: TPTAU = F(y), JSTØ values in y or
- **JSTØ**: F(y).
- **WAVE**: \( \int C_p \, dA = \) wave drag/\( q \)
- **SNWET**: Wetted area of body, assuming circular cross-sections

**SUBROUTINES CALLED:**

- **TBLU1**

Subroutine **BPINT**

**PURPOSE:**

To integrate the nacelle(s) pressure signature over fuselage surface to obtain interference drag.

**METHOD:**

The composite nacelle pressure signature (contained in XVAL, CPVAL, JNEXT values) is applied to the fuselage area distribution obtained in subroutine **BSETUP**. The process is repeated for each nacelle origin. The integral of \( C_p \) area is called **BFØRCE**.

**USE:**

```plaintext
CALL SUBROUTINE (ZNAC, BFØRCE, JNEXT)
```

**INPUT:**

- **ZNAC**: Nacelle vertical deminsion, relative to wing
- **BFØRCE**: Resulting pressure force
- **JNEXT**: Number of values in nacelle pressure signature

**SUBROUTINES CALLED:**

- **TBLU1**

Subroutine **BSETUP**

**PURPOSE:**

To calculate fuselage area growth for use in integrating nacelle-on-fuselage interference drag term.
METHOD: The fuselage is broken into 3 pieces: forebody, mid-body region (wing part), and aft body. The mid-body region covers the X interval of the wing-fuselage intersection.

Each fuselage piece is broken into 50 segments, and the area growth associated with each segment stored in an array (for use later in subroutine BPINT). The arrays are:

FBX, FBDA  forebody X and area
BMX  mid-body X
BMDAU  area growth above wing
BMDAL  area growth below wing
ABX, ABDA  aft-body X and area

USE: CALL BSETUP

INPUT: Fuselage geometry contained in /FUSLGE/ wing-fuselage intersection contained in /WBINT/

SUBROUTINES CALLED: TBLU1

Subroutine COMBINE

PURPOSE: To combine two nacelle pressure fields, for nacelles located above and below the wing, into a single composite pressure field.

METHOD: If nacelles are located both above and below the wing, subroutine NACPF generates two nacelle pressure fields: TCPDN (nacelles below) and TCPN (nacelles above). COMBINE sums these two nacelle pressure fields, to use in computing the thickness pressure interference drag. The resulting pressure field is stored in TXPN (per cent chord array) and TCPN (pressure coefficient array), with NYP semi-span per cents contained in TYP

USE: CALL COMBINE

INPUT: Nacelle pressure fields described above

SUBROUTINES CALLED: TBLU1
Subroutine **CUTOFF**

**PURPOSE:**
To delete portions of a nacelle pressure field.

**METHOD:**
In constructing nacelle pressure signatures at an arbitrary distance from the generating body, subroutine **NACPF** calls **CUTOFF** to delete parts of the signature intercepted (or not reflected) by intermediate configuration components. The pressure signature is contained in **BT0** AND **CT0** (x and pressure coefficients, respectively).

**USE:**
CALL **CUTOFF** \((x_1, x_2)\)

**INPUT:**
\(x_1, x_2\) Bounding values of \(x\), between which the pressure coefficients are set to zero.

**SUBROUTINES CALLED:**
**TBLU1**

Subroutine **DBCALC**

**PURPOSE:**
To calculate wing slopes for all grid elements representing wing planform.

**METHOD:**
The wing thickness definition is contained in common block **THICK**, consisting of the array **TBRD** (thickness profile), **TPCT** (per cent chord array, **NPTCT** values). The \(y\) stations corresponding to **TBRD** are the planform input stations, in array **YLED**. **DBCALC** first interpolates the **TBRD** array linearly at the midpoint of each semi-span element, storing the results in **ATEC**. The **ATEC** array is then interpolated chordwise to obtain the \(dz/dx\) slope for each element, storing the slopes in **TCEC**. The chordwise interpolation may be either linear or quadratic depending upon an input code. (The **TCEC** storage scheme is described under subroutine **TRANSFM**.)

Inboard of the wing side-of-body \(y\) value (input **ANYBOD**, if **ANYBOD** = +), the wing slopes are set to zero.

**USE:**
CALL **DBCALC**

**INPUT:**
Wing planform, thickness, **ANYBOD** (described above).

**SUBROUTINES CALLED:**
**TBLU1**
Subroutine GETNCP

PURPOSE: To interpolate a nacelle pressure signature at a given wing semi-span station, for use in calculating nacelle-on-wing interference drag.

METHOD: The nacelle pressure field, TCPN, is a two-dimensional array (for percent chord and span). The wing program, P916, performs wing thickness pressure calculations at selected span stations. It is convenient to have a one-dimensional array, versus X only at that span station, for use in interpolating nacelle pressures for the nacelle-on-wing interference drag.

GETNCP interpolates TCPN to obtain the required pressure signature, which is stored in TXNCW (percent chord) and TCPNCW (pressure coefficient).

USE: CALL GETNCP(YB2)

INPUT: YB2 Semi-span fraction
/NPF/ nacelle pressure field definition common block.

SUBROUTINES CALLED: None

Subroutine GLANCE

PURPOSE: To perform cutoff of nacelle pressure signature according to glance feature of nacelle pressure field.

METHOD: If the "glance" option of calculating the nacelle pressure field is used, this subroutine calculates what part of a pressure signature from a selected nacelle acting at a given spanwise station should be deleted. The approximation is made that the pressure signature propagates along Mach lines for this calculation.

If deletion of part of the signature is required, subroutine CUTOFF is called.

USE: CALL GLANCE (Y, YNAC, BETA)

INPUT: Y Y station on wing
YNAC Y station of nacelle generating pressure signature
BETA Mach number parameter, $\sqrt{M^2-1}$
SUBROUTINES
CALLED: CUT0FF

Subroutine GRAB

PURPOSE: To interpolate pressure coefficient at a given planform location from the wing thickness pressure table or fuselage-on-wing interference pressure table.

METHOD: Linear interpolation for pressure coefficient from a two-dimensional array.

USE: CALL GRAB (X, Y, XCP, JTELL)

INPUT: X X coordinate on wing planform
Y Y coordinate on wing planform
XCP Interpolated pressure coefficient
JTELL Variable to select interpolation array
1 = wing thickness pressure
2 = fuselage-on-wing pressure
VAR Wing thickness pressure array
PBWG Fuselage-on-wing pressure array

SUBROUTINES
USED: None

Subroutine INTPLT

PURPOSE: To generate a detailed definition of the tilted F(y) function for a selected right-running leg.

METHOD: Subroutine PSIG identifies all right-running legs of the tilted F(y) function. INTPLT is used to enrich the definition for use in later interpolations. The selected leg may either be a "base" leg, or a leg that possibly contains an area-balancing solution. The maximum number of points used to enrich the leg definition is 300. The enriched definitions end up in the following arrays:

<table>
<thead>
<tr>
<th>Base leg</th>
<th>Other leg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tilted F(y)</td>
<td>BT0</td>
</tr>
<tr>
<td>F(y)</td>
<td>BPTAU</td>
</tr>
<tr>
<td>Integrated area</td>
<td>BAREA</td>
</tr>
</tbody>
</table>
USE: CALL INTPLT (SUM, TPRE, FPRE, NL, NR, NSET, TFTAU)

INPUT: SUM Area under tilted F(y) function corresponding to left end of leg
TPRE, FPRE Value of tilted Y, (F(y)) at left end of leg
NL, NR Storage index of tilted F(y) array (arrays TT0 = tilted y, TFTAU = F(y))
NSET Index for leg definition
1 = base leg
2 = succeeding leg
TFTAU F(y) array

SUBROUTINES CALLED: None

Subroutine INTSEC

PURPOSE: To locate the wing-fuselage intersection and compute corresponding exposed wing wetted area.

METHOD: The wing-fuselage intersection code is controlled by input ANYB0D. If positive, ANYB0D defines the inboard end of the wing for purposes of the thickness definition, whether a fuselage is present or not. If ANYB0D is negative, the intersection of the wing and fuselage is calculated by solving for the point at which each constant per cent chord meanline of the wing crosses through the fuselage surface (assuming the fuselage to be circular).

The wing-fuselage intersection is stored in common block /WBINT/.
WX X location of intersection
WY, W5, WT Corresponding Y, S, and thickness values

USE: CALL INTSEC (WSW)

INPUT: WSW Exposed wing wetted area

SUBROUTINES CALLED: TBLU1

Subroutine MERGE

PURPOSE: To create a single composite pressure signature from multiple superimposed signatures.
METHOD:
The nacelle pressure field in NACPF at a given Y station is built up from contributions from all nacelles. The pressure signature from a single nacelle is contained in arrays BTØ and CTØ (x and pressure coefficient), NW values of each. The composite signature becomes XVAL and CPVAL, with JNEXT values of each.

All values of the new pressure signature are retained, and are merged with any previous values in XVAL and CPVAL, except that the merged pressure signature is cut off aft of an input X value (XTE). In addition, the location of the bow shock of all pressure signatures in CPVAL is saved (in TSX, NWWHAT values).

USE:
CALL MERGE (JNEXT, NWWHAT, XTE)

INPUT:
JNEXT Number of values in composite CPVAL array
NWWHAT Number of combined pressure signatures
XTE Aft-most X value of interest

SUBROUTINES CALLED:
TBLU1

Subroutine NACPF

PURPOSE:
To calculate pressure field of nacelles and associated drag terms.

METHOD:
NACPF is a major subprogram of P916, which is used to calculate the nacelle thickness pressures, and all nacelle interference terms except the wing-on-nacelle and nacelle-on-wing terms. A schematic of the program is shown in Figure 2.4-2. The calculation sequence is as follows:

1. Construct the pressure field of the nacelles acting on the wing. To do this, a series of wing semi-span stations is set up (in YSETUP). The program then calculates the composite pressure signature from all nacelles acting at those Y stations, for nacelles first below the wing, then for nacelles above the wing (if any). Either glance or wrap nacelle pressure field option may be used.
The nacelle pressure field is shrunk to 20 X and pressure coefficient values at each y station, stored in common block NPF:

TXPN x array, per cent chord
TCPN pressure coefficient array
TYP y semi-span stations (per cent), NYP values

2. Calculate composite pressure signatures and interference drag terms between nacelles or between nacelles and fuselage. The composite pressure signatures are applied to the area growth of the affected component to get the drag force.

3. "Image" effects of the nacelles are calculated. If the nacelle is located next to the wing, the pressure signature from itself reflects off the wing and back onto the generating nacelle. Similarly, the reflected signature from other nacelles may cause an interference drag. Drag interference due to reflected pressure signatures are calculated separately from the direct effects under (2).

USE: CALL NACPF

INPUT: Configuration geometry, passed to NACPF in common blocks.

SUBROUTINES CALLED: See schematic, figure 2.4-2.

Subroutine NSETUP

PURPOSE: To separate nacelles into those above or below wing, write nacelle geometry, and calculate nacelle wave drag, wetted area, and F(y) function.

METHOD: NSETUP is used for general bookkeeping involving the nacelles, and is called by NACPF. It scans the nacelle origins and separates them into those below and above the wing. It prints the nacelle input, and then zeroes the tables used to sum the nacelle interference drag terms. It then calls subroutine BØDY to calculate the nacelle F(y) function, wave drag, and wetted areas.
If all nacelles have the same geometry, NSETUP calculates the nacelle $F(y)$ only once, and shifts it appropriately for other nacelle origins.

**USE:**
CALL NSETUP

**INPUT:**
Nacelle data in common blocks

**SUBROUTINES CALLED:**
BODY

---

**Subroutine ORDER**

**PURPOSE:**
To arrange an arbitrary set of numbers into a monotonically increasing array.

**METHOD:**
The nacelle bow shock locations contained in TSX (NWHAT values) are in random order. For interpolation by TBLUI, they must be in monotonically increasing order, and cannot be double-valued. ORDER is used to rework TSX as required.

**USE:**
CALL ORDER (TSX, NWHAT)

**INPUT:**
TSX Array of bow shock X locations
NWHAT Number of values in TSX array

**SUBROUTINES CALLED:**
None

---

**Subroutine PINGRT**

**PURPOSE:**
To integrate pressure field over surface of nacelle.

**METHOD:**
Given a pressure signature defined by the XVAL, CPVAL arrays, PINGRT is used to integrate the signature as a buoyancy field over a nacelle area distribution.

**USE:**
CALL PINGRT (J, FORCE, JNEXT)

**INPUT:**
J Index value of nacelle origin
FORCE Resultant pressure drag term, $\int_{C} P \, dA$
JNEXT Number of values in XVAL or CPVAL array

**SUBROUTINES CALLED:**
40
Subroutine PSIG

PURPOSE:
To calculate the near-field pressure signature for a body of revolution at a given distance from the body.

METHOD:
Given an $F(y)$ function, Mach number, and radial distance from the body, PSIG is used to calculate the resultant pressure signature. The method is described in the aerodynamic theory document (part 1).

The basic steps in the calculation sequence are as follows:

1. The tilted $F(y)$ function is constructed from the input $F(y)$ function, and the running sum of area under the tilted $F(y)$ function computed. (The input function is defined by arrays TTAU, TFTAU and the tilted function by TTØ, TPTAU, with the area corresponding to TTØ in TAREA).

2. The bow shock position is identified by the left-most value of TTØ for which TAREA = 0. (Interpolation of the TAREA array is used for accuracy.)

3. The tilted $F(y)$ function remaining after the bow shock is located is searched to identify all right-running legs.

4. All right-running legs are compared for common TAREA values, which would identify possible shock wave locations. (Again, interpolation is used to enrich the TAREA definition, using subroutine INTPLT.)

5. Valid shock wave solutions are identified, based on the left-most criteria described in the theory document.

6. The resulting tilted and area-balanced $F(y)$ function is converted to pressure coefficient by means of the multiplier CØNST and stored in arrays BTØ and CTØ (x and pressure coefficient). The BTØ and CTØ arrays are cut off if x becomes greater than input value YED.
CALL PSIG (TTAU, TPTAU, JSTØ, R, YED)

TTAU  F(y) definition, Y = TTAU array,
TPTAU F(y) = TPTAU array, JSTØ values
JSTØ
R  Radial distance from body centerline
YED  Largest x-value of interest for pressure signature

SUBROUTINES CALLED: INTPLT, TBLU1

Subroutine TRNSFM

PURPOSE: To convert input planform geometry to program grid system.

METHOD: The wing planform is represented in the near-field wave drag program as a set of recilinear elements, described in the theory document (part 1). Given the number of semi-span rows (TNØN) used to define the right-hand wing, TRNSFM interpolates the planform definition at the mid-point of each semi-span row for the leading edge and trailing edge x value. These x values are converted to program scale (by means of RATIØ = Ywing-tip * BETA/TNØN) and stored in arrays TXLE and TXTE, with XLEØ and XTEØ defining the wing centerline grid points.

Storage of wing pressure coefficients and surface slopes uses a space-conserving technique: The leading edge of a spanwise element row is stored immediately after the trailing edge of the adjacent inboard row. The index array for the row storage is JØ, which totals the number of elements stored up to the leading edge of a selected row. The allowable total number of elements representing the right hand wing is 2750; if this total is exceeded for a given value of TNØN, the program reduces TNØN until the 2750 limit is not exceeded.

CALL TRNSFM

Input geometry in common blocks

SUBROUTINES CALLED: TBLU1
Subroutine WØNAC

PURPOSE: To calculate buoyancy effect of wing thickness pressures acting on nacelle.

METHOD: The wing-on-nacelle interference drag term, consisting of wing thickness pressures acting on the nacelle area distribution, is computed in WØNAC. Wing thickness pressures are transferred aft along Mach lines to a series of nacelle first runs. The resulting interference drag terms, $J_p dA$, are stored in array, TDWØN, with storage index corresponding to the nacelle origin number.

USE: CALL WØNAC

INPUT: Wing thickness pressure definition (VAR), configuration geometry passed in through common blocks.

SUBROUTINES CALLED: TBLUI

Subroutine YSETUP

PURPOSE: To set up semi-span Y values for calculation of nacelle pressure field.

METHOD: A set of Y stations on the right hand wing are identified in YSETUP for use in defining the nacelle pressure field. This set of Y stations consists of the semi-span Y values used in the wing thickness pressure calculations (contained in TYB2) plus extra stations immediately inboard and outboard of each nacelle centerline.

The set of Y values is arranged in monotonically increasing order by calling subroutine ØRDER. The resulting array is TYP, with NYP values.

USE: CALL YSETUP

INPUT: Configuration geometry passed in through common blocks

SUBROUTINES CALLED: ØRDER
Program LOAD6

PURPOSE: To read input geometry for near-field wave drag module.

METHOD: LOAD6 reads the input set up by the near-field program interface in the geometry module.

USE: CALL OVERLAY (SDA, 6, 2, 0)

INPUT: See input instructions in user's manual.

SUBROUTINES CALLED: TBLUI
2.5 WING DESIGN AND OPTIMIZATION MODULE

The wing design and optimization module is primary overlay 8. It contains secondary overlays to calculate:

- The flat wing solution for the input planform and Mach number
- The wing shape and force characteristics for a given loading
- The optimum combination of loadings for least drag, subject to imposed constraints.

A schematic of the wing design and optimization module is shown in figure 2.5-1.

Program TEA-253

PURPOSE: To call the three secondary overlays of the wing camber design program and to call the interactive graphics displays (if used).

METHOD: Standard FORTRAN statements

USE: Call OVERLAY (SDA, 10, 0)

INPUT: Input is in secondary OVERLAY (SDA, 10, 1)

SUBROUTINES/OVERLAYS CALLED: JLTIME, OVERLAY (SDA, 10, 1), OVERLAY (SDA, 10, 2), OVERLAY (SDA, 10, 3)

Function CPARB

PURPOSE: To define lifting pressure coefficient in the arbitrarily defined planform region for component loading 11.

METHOD: At a given span station, linear interpolation is used to establish the chordwise coordinate of the arbitrary region's leading edge. Then, \( Cp \) is proportional to distance aft of this location.

USE: \( CP = CPARB (XF, YF, CHORD) \)

INPUT: \( XF \) the chordwise and spanwise location on the wing planform in fractions of local chord and semispan at which pressure coefficient is to be provided.

45
OVERLAY (SDA,10,0,0) PROGRAM TEA253

CPARB
CPDEF
CPINTR
CPNACI
CPTDEF
JLTIME

OVERLAY (SDA,10,1,0) PROGRAM P151
MAXMIN
REGRID
TRANSFM

OVERLAY (SDA,10,2,0) PROGRAM P91615
CDWONN
HEADER
SMOOTH3

OVERLAY (SDA,10,3,0) PROGRAM OPTIMUM
MATINV

OVERLAY (SDA,10,4,0) PROGRAM BUCKTP

OVERLAY (SDA,10,5,0) PROGRAM EDBUCK

OVERLAY (SDA,10,6,0) PROGRAM STOPOP

• CRT displays

• Data input
• Flat wing calculations
• Camber surface calculations
• Optimum combination of loadings

FIGURE 2.5-1: WING DESIGN AND OPTIMIZATION MODULE
CHORD  local wing chord at span station YF.

In addition, a definition of the arbitrary region is passed through COMMON/BLOCK4/.

PRINCIPAL VARIABLES:

CPARB  the lifting pressure coefficient in the arbitrarily defined region.
XARB   chordwise and spanwise coordinates of points defining the arbitrary region.
YARB   

Subroutine CPDEF

PURPOSE:
To define lifting pressure coefficient as a function of chordwise and spanwise planform location for each of the seventeen component loadings; further, to also define this pressure coefficient for an optimum combination of loadings.

METHOD:
Lifting pressure coefficients for loadings 1-11 are defined by analytical expressions and for loadings 12-17 by linear interpolation in tables as a function of chordwise and spanwise planform location.

USE:
CALL CPDEF (DX, BETAY, CHORD, KOPT, IFLAG)

INPUT:
DX  is distance aft of leading edge in program units.
BETAY is the product of the Mach number factor ($\sqrt{M^2-1}$) and spanwise distance from the centerline in program units.
CHORD is wing chord in program units at span station BETAY.
KOPT is the index for the loading number table LOADNO.
IFLAG is a parameter which offers the option of deleting the contribution to lifting Cp of the three configuration-dependent loadings. This option is used during camber surface calculations which use lifting pressures, but not during calculation of wing surface pressures.

Input is also provided through the use of common blocks.
PRINCIPAL VARIABLES:

AI  a table of loading factors computed by program OPTIMUM in OVERLAY (SDA, 10, 3), one for each of the component loadings, that defines the optimum combination of loadings.

CPBODL  a table of lifting pressure coefficient acting on the wing and created by nonsymmetric body volume distribution.

CPBUPW  a table of lifting pressure coefficients acting on the wing and created by the body upwash flow field.

CPDEF  lifting pressure coefficient (lower surface \( C_P \) - upper surface \( C_P \)).

FLOAD  a table of scaling factors set in program P151 which produces component loading lift coefficients of order 1.0.

LOADNO  a table of component loading numbers with index KOPT that are input data in program P151.

NLOADS  the number of component loadings

XBODUP  the chordwise and spanwise tables corresponding to the body upwash loading table CPBUPW, in units of percent of local chord and percent of local span, respectively.

YBODUP  the chordwise and spanwise tables corresponding to the body buoyancy loading CPBODL, in units of percent of local chord and percent of local span, respectively.

SUBROUTINES CALLED:  CPARB, CPINTR, CPNACI

Function CPINTR

PURPOSE:  This program interpolates linearly in chordwise and spanwise directions for pressure distributions not defined in COMMON blocks.

METHOD:  Chordwise linear interpolation is used first for two span stations bracketing the span station of interest; then, linear spanwise interpolation is used.

USE:  \( CP = \text{CPINTR (XF, YF, X, Y, CP, NX, NY, ...) \} \)
NXMAX, NYMAX)

INPUT:

XF the chordwise and spanwise coordinates of the point on the plan-

YP form for which pressure coefficient

is to be found, in fractions of

local chord and of semispan, re-

spectively.

X tables of chordwise and spanwise

Y locations at which pressure coef-

ficient CP is defined. NX and NY

values, respectively, are defined

and NXMAX and NYMAX, respectively,

are the maximum values of NX and NY.

CP a two-dimensional array of pressure

coefficients.

PRINCIPAL

VARIABLES: CPINTR interpolated value of pressure

coefficient.

Function CPNACI

PURPOSE: To interpolate linearly both chordwise and

spanwise for lift pressures due to nacelles

that are defined in COMMON. The chordwise

locations at which nacelle pressures are

defined vary from span station to span

station.

METHOD: The span stations bracketing the desired

span station are located first. Linear in-

terpolation chordwise is completed at these

two span stations for the desired chordwise

location, and finally, spanwise interpolation

is carried out at the desired chordwise

location.

USE: CP = CPNACI (XF, YF)

INPUT: XF chordwise and spanwise locations

YP on the wing planform in fractions

of local chord and semispan, re-

spectively.

In addition, the nacelle pressure field CP

and the spanwise and chordwise locations at

which it is defined, Y and X, are defined by

COMMON NPF.
VARIABLES:

- **CP**: A two-dimensional table of nacelle pressure field defined in the analysis program.
- **CPNACI**: Interpolated value of the nacelle pressure coefficient.
- **X**: A two-dimensional table of chord-wise locations corresponding on a one-to-one basis with the CP table.
- **Y**: A one-dimensional table of semispan stations corresponding to the span-wise variations of both X and CP.

Function CPTDEF

**PURPOSE:** To interpolate linearly both chordwise and spanwise for pressure coefficient due to wing thickness and defined in COMMON.

**METHOD:** Linear interpolation is carried out chordwise at the two span stations bracketing the desired span station, and is followed by linear spanwise interpolation.

**USE:**

\[ \text{CPT} = \text{CPTDEF} (XF, YF) \]

**INPUT:**
- **XF**: Chordwise and spanwise stations on the wing planform in fractions of local chord and semispan, respectively.
- **YF**: The wing thickness pressures are defined in a two-dimensional array CPT in COMMON block PLIM; the chordwise locations X and spanwise locations Y corresponding to the elements in CPT are also defined in COMMON.

PRINCIPAL VARIABLES: All defined above.

Subroutine JLTIME

**PURPOSE:** To print out time since job began execution and to print out the time increment since the immediately preceding call to this subroutine.

**METHOD:** Interrogate the system timing subroutine SECOND.
Program P151

PURPOSE: To read input for the wing design and optimization module; to calculate the flat wing loading; and to set normalizing factors for component loadings 1-10 so that their lift coefficients will be approximately 1.0.

METHOD: The initial part of P151 reads the input data (set up by the wing design interface in the geometry module) and writes it out according to the print code selected in the input. The wing grid system is also established (in subroutine TRNSFM). Then (if the RESTART feature is not being used), the flat wing solution for the planform is calculated.

The method of the flat wing solution is essentially the same as described for the lift analysis module (using the equations given in the theory document, part 1), except that only the wing is present (no fuselage, nacelles, etc). The wing is scanned from front to back and centerline to right hand wing tip, computing the pressure coefficients (CP) for all field point elements (LSTAR, NSTAR). The 9-point smoothing equation is applied after all pressure coefficients are calculated.

The lifting pressure distribution is calculated over the surface of the wing to obtain lift coefficient (SCL9), drag coefficient (SCD9), \( \frac{dC_m}{dC_L} \) (DCMCL) and drag-due-to-lift factor (KF), based on input reference geometry.

For the given planform, normalizing factors (array FL\( \text{LOAD} \)) are then calculated which will produce a lift coefficient of approximately unity when used with each of the analytically defined basic loadings (1-10) in program P91615.
INPUT: See User's Manual

SUBROUTINES CALLED: TRANSFM, MAXMIN, REGRID, JLTIME

Subroutine MAXMIN

PURPOSE: To identify and print both the maximum and
the minimum elements in a two-dimensional
array.

METHOD: Standard FORTRAN library subroutines.

USE: CALL MAXMIN (A, NX, NY, NXMAX, NYMAX)

INPUT: A a two-dimensional array with maximum
dimensions NXMAX and NYMAX
containing (NX)(NY) values to be
searched for maximum and minimum.

OUTPUT: THEMEX The maximum element of A
THEMIN The minimum element of A.

Subroutine REGRID

PURPOSE: To define grid system data when the RESTART
option is used.

METHOD: When the RESTART option is used, all basic
loading data is input from cards (or tape)
and grid system calculations normally made
(which will be needed) are bypassed.
REGRID sets up the appropriate arrays, which
include the wing chord at each of the span-
wise calculation stations, and the per cent
chord and per cent wing length associated
with each grid element along those spanwise
stations.

USE: CALL REGRID

INPUT: COMMON block BLOCK1 is used to input a
definition of the wing planform and the
associated parameters required to define
the Mach box grid system.
Subroutine TRNSFM

PURPOSE: To convert input data to program units and set up wing grid system.

METHOD: The wing is represented in the program by a set of rectilinear elements, with the number of semi-span element rows given by input TN\\text{N}. TRNSFM interpolates the planform at the centerline of each element row to define the leading edge and trailing edge values, converts them to program scale (using $\text{RAT} = \text{Y_{wing tip} BETA/TN\\text{N}}$), and stores them in arrays TXLE and TXTE. Special values XLE0 and XTE0 define the wing centerline leading edge and trailing edge.

If the parabolic apex option is selected in the input ($\text{YSN\\text{G\\text{0}T}>0.$), the wing leading edge out to YSN\\text{G\\text{0}T} is altered to a parabolic shape.

USE: CALL TRNSFM

INPUT: A definition of the wing planform in physical units is passed to TRNSFM by COMMON block BLOCK1 and block SNOOT.

SUBROUTINES CALLED: TBLU1

OVERLAY (SDA, 10, 2)
Program P91615

PURPOSE: To calculate the aerodynamic characteristics of a specified lift loading and the camber surface required to support it. Both component loadings and combinations of component loadings are handled. If requested, all data for the RESTART option are punched in this program.

METHOD: Program P91615 solves for the camber surface required to support a specified loading and the associated force coefficients, using the equations given in the theory document (part 1). The program is actually used in two ways:

1) To calculate the force coefficients and interference drag characteristics of a set of basic loadings.

2) To calculate the camber surface for an optimum combination of loadings.
The program code used to define the usage of P91615 is KOPTI. If KOPTI is greater than the number of basic loadings used, then option (2) above is employed.

Option (1):

In the calculation of the camber surface and force coefficients, a series of semi-span stations are selected in the program input (TJBYS). The program then picks a loading, calculates the required surface slopes for all elements at each TJBYS station, and integrates the slope distribution to obtain the camberline. In these calculations, each element along TJBYS is identified by the nomenclature LSTAR (X) and NSTAR (βy), as described in the theory document.

After the camberlines and sectional force coefficients of all spanwise stations have been computed, the characteristics are integrated spanwise to obtain lift coefficient (CL), drag coefficient (CD), and pitching moment coefficient (CMAX). These are converted to input reference geometry basis and become CLR and CDR, lift and drag coefficients. The pitching moment coefficient is adjusted to the value at zero lift, and becomes CM0R. Drag-due-to-lift factor is labeled KE = CDR/(CLR)^2. The interference drag coefficients are stored in array CDI.

After all calculations are completed for a given loading, the force coefficients are stored in array TDRAG, and the process repeated until all loadings have been used.

The force coefficients and interference drag coefficients of all loadings are then converted to the component and interference forms used in the matrix solution described in the theory document, and stored in common blocks to be passed to program OPTIMUM.

Finally, the RESTART data is written onto tape 3, and also punched into cards, if requested. These data consist of all component and interference drag terms, the configuration-dependent loadings (if used), and grid system data calculated by P91615.

Option (2):

If option 2 was selected, involving the calculation of the wing shape for an optimum combination of loadings, the calculation sequence is the same as if a basic loading was being used. However, the interference characteristics are not required. The resulting camber surface is stored in common block/CAMBER/ in the following form:

TPCT percent chord array, NØPCT values.
The camber surface will also be punched into cards, if requested in the program input.

**USE:**
```
CALL OVERLAY (SDA, 10, 2)
```

**INPUT:**

All input is handled by COMMON blocks, which pass the required data from P151.

**SUBROUTINES CALLED:**

CDWONN, CPDEF, CPINTR, CPNACI, CPTDEF, HEADER, JLTIME, SMOOTH3, TBLU1

**Purpose**

To calculate the axial force acting on nacelles due to the wing-lift flow-field.

**Method:**

The wing lower surface lifting pressures are projected downward in a vertical plane and aft along Mach lines to the nacelle locations where their product with nacelle incremental frontal area is numerically integrated to produce a nacelle axial force.

**Use:**

```
CDN = CDWONN(KOPT)
```

**Input:**

KOPT component loading index.

In addition, wing planform information is passed to CDWONN by COMMON block BLOCK1 and nacelle geometry is passed by block BLOCK11.

**Principal Variables:**

- **CPTERM** wing lower surface pressure coefficient due to wing lift only.
- **XLEN** longitudinal coordinates of the wing leading and trailing edges at the nacelle span station, and the corresponding wing chord.
- **XTEN** wing leading and trailing edges.
- **CRDN** the nacelle span station.
- **RNAC** two-dimensional arrays defining nacelle radius as a function of nacelle longitudinal station. The first parameter is the nacelle number index.
- **XNAC** nacelle radius as a function of nacelle longitudinal station.
- **XYZ** a two-dimensional array specifying the coordinates of each of the
nacelles' forward-most point. The first parameter is the nacelle number index, and the second parameter defines the coordinate being referenced - i.e., first value is X, second is Y, and third is Z.

**SUBROUTINE CALLED:** CPDEF

**Subroutine HEADER**

**PURPOSE:** To write out loading number and case identification.

**METHOD:** Standard FORTRAN statements.

**USE:** CALL HEADER(KOPT)

**INPUT:** KOPT component loading index.

In addition, the loading number index, loading name, and case identification are passed via COMMON statements.

**OUTPUT:** Titling information for each component loading prior to its aerodynamic and camber analysis.

**Subroutine SMOOTH3**

**PURPOSE:** To apply 3-point smoothing to a selected range of elements within a one-dimensional array.

**METHOD:** Each element to be smoothed is replaced according to the following algorithm:

\[ y_i \rightarrow \frac{1}{4} (y_{i-1} + 2y_i + y_{i+1}) \]

where \( y_{i-1} \) and \( y_{i+1} \) are the immediate neighbors of \( y_i \) before smoothing. If \( y_i \) is the first element in the array to be smoothed, then

\[ y_i = \frac{1}{3} (2y_i + y_{i+1}) \]

and if \( y_i \) is the last,

\[ y_i = \frac{1}{3} (y_{i-1} + 2y_i) \]
USE:

CALL SMOOTH3(A, IFIRST, ILAST, N, NARRAY)

INPUT:

A     the one-dimensional array of data
to be smoothed.
IFIRST  first and last elements in A to be
ILAST   smoothed. All elements between
         will be smoothed.
N     the number of times the smoothing
      algorithm is to be applied.
NARRAY  the maximum number of elements in A.

OVERLAY (SDA, 10, 3)
Program OPTIMUM

PURPOSE:
To define various optimum combinations of lift
loadings in terms of their load strength factors
$A_i$.

METHOD:
Lagrange's method of undermined multipliers (as
described in Part I: Theory), as a function of the
aerodynamic characteristics of each of the
component loadings and their mutual interferences.
In program OPTIMUM, the DO loop on statement 610
(index ILOOP) is used twice only if a constraint on
pitching moment coefficient is used. For ILOOP = 1,
program OPTIMUM produces a solution for minimum
drag with only a lift constraint; 21 solutions of
the drag-due-to-lift factor (KE) and zero lift
pitching moment coefficient (CMO) defining the
design "bracket" plot; and if requested, a solution
for minimum drag with lift coefficient and wing
upper surface pressure constraints. If the latter
solution is satisfied by the first solution, the
latter is set equal to the first. For ILOOP = 2,
program OPTIMUM adds a constraint on pitching
moment coefficient at zero lift to both the
solution with lift coefficient constraint and the
solution with lift and pressure constraints.
Within the ILOOP loop, the left-hand-side solution
matrix AMAT and the right-hand-side solution matrix
BMAT are calculated first, corresponding to figure
4.4-4 of the theory document (part I). The left-
hand-side matrix is stored in ATEMP for
multiplication with the solution as a test of its
accuracy; this multiplication should produce the
right-hand-side.

Subroutine MATINV is called to solve for the load
strength factors $A_i$ and the Lagrange multipliers,
corresponding to $A$.
in figure 3.4-3 of the theory document. The solution load strength factors $\lambda_i$ are stored in two arrays - the $A_i$ array so that the current solution is defined in the wing lifting pressure subroutine CPDEF, and in the TAI array, so that the solution will be defined for program P91615 after program OPTIMUM has been exited. The array TAI has capacity for four sets of load strength factors (under IDUM), corresponding to the four types of available solutions as follows:

<table>
<thead>
<tr>
<th>IDUM</th>
<th>Lift Constraint</th>
<th>Pitching Moment Constraint</th>
<th>Wing Pressure Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>3</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

Lift coefficient CLSOL, drag coefficient CDSOL, and pitching moment coefficient CMOS are computed in program units from the load strength factors and the aerodynamic coefficients of the component loadings. These parameters are then converted to the input reference geometry basis, CLR, CDR, and CMOR, respectively. Values of KE and CMO from previous solutions (if any) for IDUM = 4 are shifted one location toward the rear of the "bucket" plot arrays CKE and CMERO, and the current solution data are stored in these arrays.

The wing upper surface pressure coefficient CPUS is calculated and compared with the user-defined upper surface limiting pressure CPLIM everywhere on the planform, and the minimum value of the difference $C_{\text{PSMIN}} = (\text{CPUS} - \text{CPLIM})$ is identified, along with its planform location. This completes the solution corresponding to IDUM = 1 above.

Then, if ILOOP = 1 and if no pressure constraints have been applied, values of KE and CMO are generated by the DO loop on 560 for the "bucket" plot. This solution parallels the one just described, except that a constraint on design pitching moment coefficient at zero lift CMOD is
added and is varied through 21 values. The range of values of CMOD depends on design lift coefficient CMODMIN, and is centered on the pitching moment coefficient corresponding to the solution for IDUM = 1, if available, or on zero. Values of CMOD are truncated for plotting convenience.

After the bucket plot data are generated and stored, program OPTIMUM tests to see if pressure constraints on the wing upper surface have been requested and whether they are necessary, if requested. If both tests are positive, then a loop to statement 10 is used to apply a constraint on wing lifting pressure at the planform location of CPMIN. This loop is within the loop on 560 (index = ILOOP). The wing lifting pressure coefficient CPLMAX is calculated which will provide a wing upper surface pressure coefficient CPUS equal to the limit pressure coefficient CPLIM, and the constraint on lifting pressure is set to 95% of CPLMAX to provide a small margin so that immediately adjacent planform locations do not exceed the limit by a trivial amount. This inner loop to statement 10 (not a DO loop) has index NCPCON, the number of pressure coefficient constraints, and halts when either:

1) The wing upper surface pressure satisfies the user-defined limiting pressures, or
2) the number of solution constraints of all types is equal to the number of component loadings.

In the latter case, a note is printed and the \( A_i \) for the last cycle of the loop on NCPCON are retained in TAI.

For ILOOP = 2, a pitching moment constraint is added to the solution, and both lift coefficient and pressure limiting constraints function as they did for ILOOP = 1. The bucket plot calculations are omitted.

**USE:**
CALL OVERLAY (SDA, 10, 3)

**INPUT:**
All input is by means of COMMON blocks.

**PRINCIPAL VARIABLES:**
- \( A_i \) load strength factor
- AMAT left-hand-side solution matrix
- BMAT right-hand-side solution matrix
- CDIJ interference drag coefficient
CDSOL  solution drag coefficient
CDWON  interference drag coefficient of wing lift acting on nacelles.
CLDSIN  design lift coefficient
CLI  the ith component lift coefficient
CPBODL  lifting pressure coefficient due to unsymmetric body volume distribution
CPBODU  body pressure coefficient acting on the wing upper surface
CPBUPW  lifting pressure coefficient due to the body upwash loading
CPLIMIT  wing upper surface limit pressure coefficient (on input table)
CPNAC  wing lifting pressures due to nacelles
NCMAX  maximum number of solution constraints
NCPCON  number of pressure coefficient constraints
NLCON  number of direct solution loading constraints due to use of configuration-dependent loadings
OPTION  integer array controlling extent of four types of solutions
TAI  up to four sets of load strength factors $A_i$, corresponding to the four types of solutions.
TBDAY  stored parameters which define a solution lifting pressure constraint, namely, spanwise location chordwise location, local chord, and allowable lifting pressure coefficient, respectively.
USEBOY  a logical flag indicating use of body buoyancy loading if true
USEBUP  a logical flag indicating use of body upwash loading if true
USECMC  a logical flag indicating use of pitching moment constraint if true
USECPL  a logical flag indicating use of wing upper surface pressure limiting if true
USEOPT  a logical flag indicating use of wing thickness pressures if true
USENAC  a logical flag indicating use of nacelle buoyancy loading if true

OUTPUT:

Essential output is the set of up to four solution definitions in terms of the loading strength factors $A_i$. These are passed by COMMON blocks back to the camber surface calculation overlay, OVERLAY.
SUBROUTINES CALLED: CPDEF, CPINTR, CPNACI, CPTDEF, MATINV

Subroutine MATINV

PURPOSE: To solve a set of linear, simultaneous equations. This is a NASA-LRC library subroutine.

METHOD: See LRC library.

USE: CALL MATINV (A, N, B, M, DETERM, IPIVOT, INDEX, NMAX, ISCALE)
2.6 LIFT ANALYSIS MODULE

The lift analysis program is primary overlay 7. The program is divided into separate elements to read input, transform input into program units, and perform the lifting pressure calculations as shown schematically in figure 2.6-1.

The graphics displays are called in program FINISH.

Program TEA201

PURPOSE: Primary level of lift analysis module

METHOD: TEA201 sets up the calculation sequence for the drag-due-to-lift analysis program. The calculation loops are:

- \( \text{D}_0 \ 50 \ \text{JD}_0 \) Mach number loop, repeated for each Mach number.
- \( \text{D}_0 \ 40 \ \text{MLMT} \) Pressure limiting loop. LIMIT angles of attack, if limiting requested.
- \( \text{D}_0 \ 30 \ \text{JCALP} \) Canard angle of attack loop. Repeated for each canard alpha, if canard is present.

USE: CALL OVERLAY (SDA, 7, 0, 0)

INPUT: See user's manual.

SUBROUTINES CALLED: RCALC

Subroutine DUBINT

PURPOSE: To perform double interpolation in array

METHOD: Given a two-dimensional array, DUBINT performs double linear interpolation for an answer at a specified location in array.

USE: CALL DUBINT (X1, Y1, TX, TY, NX, NY, TBL, MX, MY, ANS)

INPUT: X1 X location
       Y1 Y location
FIGURE 2.6-1—LIFT ANALYSIS MODULE SCHEMATIC
SUBROUTINES CALLED: None

**Subroutine GETR**

**PURPOSE:** To calculate influence factor for specified grid element.

**METHOD:** Given an element located in the forecone from a selected field point element, GETR provides the corresponding influence factor. (The influence factor equation is discussed in the theory document, part I).

**USE:**

```
CALL GETR (LSTAR, LVAR, NDIF, R)
```

**INPUT:**

- **LSTAR** Field point element
- **LVAR** Specified grid element
- **NDIF** |NSTAR - N|
- **R** Influence factor

**SUBROUTINES CALLED:** None

**Subroutine GGRID**

**PURPOSE:** To define leading edge and trailing edge data for grid system for specified semi-span element row.

**METHOD:** Given a wing semi-span station, GGRID defines the leading edge and trailing X values, the corresponding grid elements, and the associated element fractions. GGRID is used for wing, canard or horizontal tail.

**USE:**

```
CALL GGRID (JBY, XLE, XTE, LLE, LTE, ALE, ATE, NK)
```

**INPUT:**

- **JBY** Semi-span element row
- **XLE, XTE** Leading and trailing edge X-locations of planform at JBY.
- **LLE, LTE** Grid elements corresponding to XLE, XTE.
ALE, ATE Fracitons of elements defining plan-
form at LLE, LTE.
NK Code to identify configuration component
1 = Canard
2 = Wing
3 = Horizontal tail

SUBROUTINES
CALLED: None

Subroutine RCALC

PURPOSE: To calculate a standard set of influence factors

METHOD: The influence factors used in the lift analysis
program are a function only of the relative
position of the field point element and the
influencing element. RCALC is used to precalculate
a standard set of influence factors for use in the
program to reduce computer time associated with
repeated calculations. The factors are stored in
array TRSAVE.

USE: CALL RCALC

INPUT: None

SUBROUTINES
CALLED: None

Program FUSLGE

PURPOSE: To calculate upwash field of fuselage and to
calculate isolated fuselage force coefficients.

METHOD: Fuselage is used to calculate the wing-fuselage
intersection (subroutine INTSEC), then to calculate
the isolated fuselage lift distribution and force
coefficients using slender body theory, and then to
calculate the fuselage upwash field acting in the
plane of the wing, canard, or horizontal tail.

The equations used in the fuselage lift
distribution and upwash field calculations are
given in the theory document (part 1). The
fuselage forces in the presence of the wing
downwash field are later repeated in subroutine
FUSCF under overlay (7,5). The upwash field of
canard, wing, or horizontal tail is defined by an
array of upwash value at specified chord and semi-span percentages:

<table>
<thead>
<tr>
<th></th>
<th>CANARD</th>
<th>WING</th>
<th>TAIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>% Chord</td>
<td>TXUPW</td>
<td>TXPW</td>
<td>TXUPW</td>
</tr>
<tr>
<td>Semi-span fraction</td>
<td>TYCCW</td>
<td>TYUPW</td>
<td>TYHTW</td>
</tr>
<tr>
<td>Upwash angle</td>
<td>TUPCC</td>
<td>TUPWC</td>
<td>TUPHC</td>
</tr>
<tr>
<td>Upwash angle per deg. angle of attack</td>
<td>TUPCF</td>
<td>TUPWP</td>
<td>TUPHP</td>
</tr>
</tbody>
</table>

The upwash angle calculations are performed in subroutine UPWASH.

USE: CALL OVERLAY (SDA, 7, 1, 0)

INPUT: Geometry definition contained in common blocks. MLIMT = loop index variable from (7,0) overlay.

SUBROUTINES CALLED: INTSEC, UPWASH

Subroutine INTSEC

PURPOSE: To locate wing-fuselage intersection

METHOD: If input SYMM is less than zero, then the analysis program is to solve for the wing-fuselage intersection in order to define the exposed and "carry-over" wing pieces. INTSEC selects each percent chord line of the wing camber surface definition and locates the intersection between wing and fuselage. The fuselage area distribution is considered to be made up of circular cross-sections in the intersection calculations. The resultant intersection is stored in common block/WBINT/, with X, Y, and Z values in arrays WX, WY, and WZ.

USE: CALL INTSEC.

INPUT: Configuration geometry contained in common blocks.

SUBROUTINES CALLED: TBLU1
Subroutine UPWASH

PURPOSE: To calculate fuselage upwash

METHOD: UPWASH calculates fuselage upwash angle in the plane of the canard, wing, or horizontal tail, using the slender body equations discussed in the theory document (part 1). Upwash angles are computed for a series of percent chord values at selected semi-span stations.

USE: CALL UPWASH (Y, DELX, I, L)

INPUT: Configuration geometry contained in common blocks, plus:

Y  semi-span y station
I  span storage index for upwash array
L  variable defining component
   1= canard
   2= wing
   3= tail

SUBROUTINES CALLED: TBLU1

Program WING

PURPOSE: To calculate lifting pressure distribution on wing or canard.

METHOD: A schematic of WING is shown in figure 2.6-2.

The equations used in calculation of the wing or canard lifting pressures are given in the theory document (part 1). The program scans the wing/canard grid system from front to back (DØ 470) and from the centerline to right hand wing tip (DØ 450), locating field point elements on the wing or canard. When a field point element (LSTAR, NSTAR) is located, the program (DØ 200) computes the upstream influence of elements located in the Mach forecone from LSTAR, NSTAR. The local pressure coefficient (CP) is then computed, with the fuselage upwash added to the local surface slope to obtain the effective element angle of attack. If the field point element is located inside the side-of-fuselage station, the element angle of attack is set to zero. (Either of the two pressure coefficient smoothing options may be used in these calculations, controlled by input variable SMØGØ.)
Calculation of lifting pressures at LSTAR, NSTAR

- WING 7.2
- P91611
- GETNCP
- GGRID
- GETR
- CKLIM
- SMOOTH
- CPTAB
- CANPRES
- HUPWSH
- WONAC

- Slopes for grid system
- Nacelle pressure signature
- Planform definition (grid system)
- Influence factors
- Limiting pressure check
- Pressure coefficient smoothing
- Lifting pressure tabulation
- Canard pressure integration
- Downwash on horizontal tail
- Wing-on-nacelle term

FIGURE 2.6-2.—PROGRAM WING SCHEMATIC
After CP is calculated, pressure limiting is applied if input LIMIT is greater than zero, using subroutine CKLIM.

After all pressure coefficients are calculated, they are smoothed (Dφ 630) and integrated over the configuration surface to get lift, drag, and pitching moment. At the same time, the local nacelle pressure coefficients (CPNAC) and asymmetric fuselage volume pressure coefficients (CPASYM) are superimposed. Pressure coefficients are applied to the wing slopes in the exposed wing area portion, and to the fuselage slopes interior to the side-of-fuselage station.

Corresponding calculations and summations are carried on simultaneously for the flat wing at 1 degree angle of attack. The interference drag term of flat wing pressure coefficients on the cambered wing slopes is also computed. Separate summations carry the nacelle drag, lift, and pitching moment and the configuration streamwise and spanwise lift distributions.

The wing pressure coefficient arrays (TWCP and TWCPF) are then interpolated over the wing planform for the output pressure summary (subroutine CPTAB). The canard lift, whose direct lift summation has been kept separate from the wing, is totaled in subroutine CANPRES.

The downwash due to the wing/canard combination acting in the plane of the horizontal tail is then computed in subroutine HUPWSH. The effect of the wing lifting pressures acting on the nacelle area distribution is computed in subroutine W0NAC.

Program WING is called by two loops in the 7,0 level: the canard angle of attack loop and the pressure limiting angle of attack loop. Both result in changes to the wing/canard angle of attack distribution that cannot be handled by superposition.

USE:
CALL OVERLAY (SDA, 7, 2, 0)

INPUT:
Configuration data in common blocks.

SUBROUTINES CALLED:
See schematic, figure 2.6-2.
Subroutine CANPRES

PURPOSE: To sum pressure distributions over canard for lift, drag, and pitching moment.

METHOD: The canard pressure distributions are calculated in program WING, and stored in arrays TCCP (at input canard alpha) and TCCPF (per degree alpha). CANPRES integrates these to get lift, drag, and pitching moment. Drag is computed by applying the lifting pressure to the exposed canard slopes or the fuselage slopes, as appropriate.

USE: CALL CANPRES

INPUT: Configuration data in common blocks.

SUBROUTINES CALLED: GGRID, TBLU1

Subroutine CKLIM

PURPOSE: Checks to see if calculated pressure coefficients violate limiting pressure coefficient.

METHOD: The sum of the wing upper surface lifting pressure coefficient (\(-0.5\times\) calculated lifting pressure coefficient) plus the wing thickness pressure plus the fuselage pressure acting on the wing upper surface is checked against the limiting pressure coefficient. If the summed value, CPCHK, is more negative than the limit, the computed value is reset.

USE: CALL CKLIM (YFR, XPC, CP, CPLIMT, PTEST, CPT)

INPUT: Pressure coefficient arrays in common blocks, plus:

<table>
<thead>
<tr>
<th>YFR</th>
<th>XPC</th>
<th>CP</th>
<th>CPLIMT</th>
<th>PTEST</th>
<th>CPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>semi-span wing station</td>
<td>percent chord</td>
<td>calculated lifting pressure coefficient</td>
<td>limiting upper surface pressure coefficient</td>
<td>indicator variable showing limiting application</td>
<td>thickness pressure coefficient</td>
</tr>
</tbody>
</table>

SUBROUTINES CALLED: DUBINT
Subroutine CPTAB

PURPOSE: To interpolate wing pressure arrays for output

METHOD: The wing lifting pressure distributions for the grid system are computed in WING. CPTAB is used to interpolate the grid pressure distributions for each semi-span row at selected per cent chord values. The interpolated pressure coefficients are stored in arrays VAR (basic angle of attack) and VAR1 (flat plate solution per degree alpha).

USE: Call CPTAB

INPUT: Configuration data and pressure coefficients in common blocks.

SUBROUTINES CALLED: GGRID, TBLU1

Subroutine GETNCP

PURPOSE: To interpolate a nacelle pressure signature at a given wing semi-span station, for use in calculating nacelle-on-wing interference drag.

METHOD: Same as described for subroutine GETNCP for the near-field wave drag program.

USE: CALL GETNCP (YB2)

INPUT: YB2 wing semi-span fraction

SUBROUTINES CALLED: None

Subroutine GETUP

PURPOSE: To perform double interpolation for fuselage upwash angle

METHOD: GETUP double-interpolates linearly for fuselage upwash angle in arrays TUPWC (upwash at basic fuselage incidence) and TUPWF (upwash per degree fuselage alpha).

USE: CALL GETUP (XPC, YFR, UPC, UPF)

INPUT: XPC per cent chord
YFR semi-span fraction
UPC, UFF upwash angles, radians

SUBROUTINES CALLED: None

Subroutine HUPWSH

PURPOSE: To compute wing/canard downwash

METHOD: HUPWSH is used to compute the local downwash acting along the fuselage centerline and in the plane of the horizontal tail. The fuselage asymmetric lifting pressure distribution (if any) is included in the wing lifting pressure definition.

The computed downwash is stored in arrays THWSHC (basic angle of attack) and THWSHF (per degree) for the horizontal tail; the arrays for the fuselage are BX (length fraction), BCC (downwash angle for basic angle of attack) and BCF (per degree).

USE: CALL HUPWSH

INPUT: Configuration geometry and pressure distributions in common blocks

SUBROUTINES CALLED: GGRID, GETR, DUBINT, TBLU1

Subroutine P91611

PURPOSE: To calculate local wing slopes for grid system

METHOD: Subroutine P91611 interpolates the wing camber surface definition for the streamwise slopes of the wing grid system. Interpolation is linear spanwise along constant per cent chord lines, followed by quadratic chordwise. The resultant slopes are stored in array TDBDX.

If wing twist tables or trailing edge flaps are input, the slopes are incremented by the appropriate slopes. Also, the grid element array containing the wing-fuselage intersection (INTN) is identified together with the corresponding fractional element (TNFR).

If the configuration angle of attack is not zero, as may be the case with limiting pressure calculations, all slopes are incremented by alpha.
If input WHUP = 1.0, the camber surface slopes are all zeroed. (This feature is used to generate the wing loading due to fuselage upwash only).

**USE:**

```
CALL P91611
```

**INPUT:**

Configuration geometry contained in common blocks

**SUBROUTINES CALLED:**

GGRID, TBLU1

Subroutine **SMØØTH**

**PURPOSE:**

To smooth wing pressure coefficients and sum wing area

**METHOD:**

SMØØTH is called by program WING after all wing pressure coefficients have been calculated, to remove irregularities in the calculated values. Either a 9 point or 3 point smoothing equation is applied, depending upon the wing pressure calculation technique used (discussed in the theory document, part 1). The wing area is also calculated, by summing the areas of the individual elements.

**USE:**

```
CALL SMØØTH (AREA9, SMØGØ)
```

**INPUT:**

Configuration data contained in common blocks /SMØØTH/, /INDEX/, and pressure data in /PCØEF/.

**AREA9**

(wing area of right hand wing) $\sqrt{M^2 - 1}$

**SMØGØ**

Smoothing technique code

0. = 9 term smoothing
1. = 3 term smoothing

**SUBROUTINES CALLED:**

None

Subroutine **WØNAC**

**PURPOSE:**

To calculate drag of wing lifting pressures acting on nacelle area distribution

**METHOD:**

WØNAC is used to compute the thrust or drag force due to the wing lifting pressures acting on the...
nacelle cross-sectional distribution. For this calculation, the lifting pressure is broken into upper and lower surface halves and the proper half used depending upon whether the nacelles are above or below the wing. The pressures are transferred aft along Mach lines from the wing to elemental frustrums describing the nacelle shape.

The drag increments for both the wing pressure distribution at basic incidence and per degree alpha are computed.

**USE:**

CALL WØNAC

**INPUT:**

Configuration geometry and wing pressure field contained in common blocks.

**SUBROUTINES CALLED:**

GGRID, TBLU1

---

Program PUTØUT

**PURPOSE:**

To print input data

**METHOD:**

PUTØUT is used to write the input and pertinent program data onto the output file.

**USE:**

CALL ØVERLAY (SDA, 7, 3, 0)

**INPUT:**

Configuration geometry in common blocks

**SUBROUTINES CALLED:**

None

---

Program NACPF

**PURPOSE:**

To calculate pressure fields acting on the wing due to nacelles and asymmetric fuselage volume.

**METHOD:**

This overlay calculates the pressure fields due to nacelles and asymmetric fuselage volume acting on the wing. A schematic of the overlay is given in figure 2.6-3.

The program initially calculates the asymmetric fuselage pressure distribution by calling subroutines SEGRT and SPLIT, using the fuselage representation described in the theory document.
FIGURE 2.6.3.—PROGRAM NACPF SCHEMATIC
The nacelle pressure field is next calculated. A series of semi-span \( Y \) stations are selected, and the composite pressure signature due to all nacelles is computed; first for all nacelles below the wing, then for all nacelles above the wing. If there are nacelles both above and below the wing, a single nacelle lifting pressure definition is calculated in subroutine \( \text{COMBINE} \).

The basic program format is the same as the \( \text{NACPF} \) subroutine described in the near-field wave drag program, except that the nacelle and fuselage interference terms are not computed in the analysis program version. In addition, there is an optional feature in the analysis program version to permit calculation of the nacelle pressure field at a Mach number other than free stream (to account for local Mach number effects, using input \( \text{TMLC} \)). The resulting pressure field is afterwards referenced to free stream dynamic pressure.

The following subroutines associated with \( \text{NACPF} \) are the same as those previously described in the near-field wave drag program: \( \text{COMBINE} \), \( \text{CUTOFF} \), \( \text{GLANCE} \), \( \text{INTPLT} \), \( \text{MERGE} \), \( \text{NSETUP} \), \( \text{ORDER} \), \( \text{PSIG} \).

**USE:**

```
CALL \text{OVERLAY} (SDA, 7, 4, 0)
```

**INPUT:**

Configuration geometry, Mach number contained in .common blocks.

**SUBROUTINES CALLED:**

See schematic, figure 2.6-3.

Subroutine \( \text{BODY} \)

**PURPOSE:**

To calculate Whitham \( F(y) \) function for body of revolution

**METHOD:**

Subroutine \( \text{BODY} \) is the same as the previously described \( \text{BODY} \) subroutine in the near-field wave drag module, with two additional provisions:

1) It will compute the \( F(y) \) function using the "smooth body" form of the \( F(y) \) equation discussed in reference 4, in addition to the -Stieltjes integral equation. This provision is controlled by input \( \text{FYLB} \) in the calling statement. (\( \text{FYLB} \) less than zero uses smooth body equation).
2) Linear interpolation may be used (instead of quadratic) in fairing the body radius distribution. This provision is selected if FYLB is less than zero or greater than 9.0.

**USE:**

```
CALL BODY (TXB, TRB, TNXB, TNCUT, TTAU, TFTAU, JST0, FYLB, SWET)
```

**INPUT:**

- **TXB**, **TRB**, **TNXB**, **TNCUT**, **TTAU**, **TFTAU**, **JST0**, **FYLB**
  - **TXB**, **TRB** - Input body X stations and radius values, TNXB of each.
  - **TNCUT** - Number of body intervals in F(y) function
  - **TTAU**, **TFTAU** - Y and F(y) function calculated for body, JST0 values of each
  - **FYLB** - Calculation method code (see above)

**SUBROUTINES USED:**

```
TBLU1
```

---

**Subroutine SEGRT**

**PURPOSE:**

To set up area representations for below-wing and above wing fuselage areas.

**METHOD:**

The logic of the below-wing and above-wing fuselage area representations is described in the theory document (Part 1). Subroutine SEGRT computes these area distributions using the wing-body intersection definition found in INTSEC. The fuselage is considered circular in the calculation of the two area distributions, resulting in:

- **TRABV** - above-wing radius distribution
- **TRBLW** - below-wing radius distribution

Alternatively, SEGRT may use input values of fuselage areas (if SYMM = 2.0)

**USE:**

```
CALL SEGRT
```

**INPUT:**

Configuration geometry and wing-fuselage intersection definition contained in common blocks

**SUBROUTINES CALLED:**

None

77
Subroutine SPLIT

PURPOSE: To calculate asymmetric fuselage volume pressure field

METHOD: Subroutine SPLIT is used to calculate the fuselage pressure field acting on the wing, according to input SYMM. If SYMM=0., the configuration is considered to be mid-wing, and the input fuselage definition is used to calculate a symmetric (non-lifting) fuselage thickness pressure field. If SYMM>1.0, the above-wing and below-wing fuselage area representations obtained in SEGRT are used to calculate the respective pressure fields acting on the wing.

Pressure signatures due to the fuselage are calculated at the same X and Y stations used for the fuselage upwash field. The resultant fuselage volume pressure field is stored in common block/CPBASM/: 

RXUPW  per cent chord array
RYUPW  semi-span percentages
PABOVE  pressure coefficients above wing
PBELOW  pressure coefficients below wing

USE: CALL SPLIT

INPUT: Configuration geometry contained in common blocks.

SYMM  calculation code

SUBROUTINES CALLED: BODY, PSIG, TBLU1

Subroutine YSETUP

PURPOSE: To set up Y array for nacelle pressure field definition

METHOD: YSETUP sets up a series of Y stations located at each 5 per cent semi-span, plus extra stations located immediately inboard and outboard of each nacelle centerline. Subroutine ORDER is used to store the array (TYP) in monotonically increasing fashion.

USE: CALL YSETUP

78
INPUT: Configuration geometry contained in common blocks.

SUBROUTINES CALLED: ØRDER

Program FINISH

PURPOSE: To compute fuselage forces in presence of wing downwash field, add in contribution of horizontal tail, and write out complete configuration force coefficient summaries.

METHOD: The wing/canard lifting pressure distribution and force coefficients are computed in program WING. Summary data from WING are passed to FINISH by common blocks, where the fuselage contribution in the wing/canard downwash field is calculated (subroutine FUSCF), and the direct effects of the canard are added in. All coefficients are based on input reference geometry.

A calculation loop (D0 140) then calculates the contribution of the horizontal tail at various input incidences (in subroutine HTPART), and adds it to the wing-fuselage-canard data.

A summary of the configuration force coefficients, nacelle-on and nacelle-off, is then computed and printed. This includes lift (XCL), drag (ST\$T and SCDN) and pitching moment (CMA and CMB) coefficients. Corresponding coefficients for the flat wing configuration are printed for reference.

The streamwise lift distribution is then summed and printed (subroutine STRMW\$). If wing lifting pressure coefficients at specified lift coefficients were requested, these are then calculated and printed. Finally, the wing-canard spanwise lift distribution is printed.

USE: CALL ØVERLAY (SDA, 7, 5, 0)

INPUT: Configuration data in common blocks

SUBROUTINES CALLED: FUSCF, SUMIZ\$, HTPART, STRMW\$
Subroutine HTPART

**PURPOSE:**
To calculate the horizontal tail lifting pressure distribution and force coefficients

**METHOD:**
Subroutine HTPART is used to compute lifting pressure distributions on the tail in the presence of wing-canard downwash. The equations used are the same as those employed for the wing; the tail is broken into exposed and fuselage-carry-over portions, and fuselage upwash and wing downwash added to tail alpha for the purposes of computing lift coefficients.

The lifting pressure distributions are summed over the tail planform to get tail force coefficients, which are then added to the force coefficients of the rest of the configuration. These are passed back to FINISH in common blocks.

**USE:**
CALL HTPART (NH, REفار)

**INPUT:**

NH  Tail incidence loop index
REفار Reference area in program units

**SUBROUTINES CALLED:**
GGRID, GETR, DUBINT, TBLU1

Subroutine STRMW9

**PURPOSE:**
To sum and print streamwise lift distribution

**METHOD:**
In STRMW9, the streamwise lift distributions due to wing/canard, nacelles, fuselage, and horizontal tail are summed together and printed. The presentation is in fraction of total lift coefficient, so that the final number printed for the complete configuration is 1.0.

**USE:**
CALL STRMW9 (REفار, SCLN9)

**INPUT:**

REفار Reference area in program units
SCLN9 Total lift coefficient

**SUBROUTINES CALLED:**
TBLU1
Subroutine SUMIB

PURPOSE:  To summarize and print configuration force coefficients

METHOD:  The configuration force coefficients, including the interference drag terms, are summarized for program FINISH in subroutine SUMIB.

USE:  CALL SUMIB(N)

INPUT:  

N  Index variable to identify printout series

1= WING
2= WING + FUSELAGE
3= WING + FUSELAGE + CANARD

SUBROUTINES CALLED:  None

Subroutine FUSCF

PURPOSE:  To calculate force coefficients of fuselage in presence of wing downwash field

METHOD:  FUSCF repeats the slender body fuselage lift calculations of program FUSLGE, adding the wing downwash field computed in HUPWSH to the basic fuselage angle of attack. The fuselage is broken into segments and the equations given in the theory document (Part 1) are used to calculate the fuselage lift. Fuselage drag is computed by applying the fuselage lift distribution to the local mean-line slopes.

Lift for the fuselage at basic incidence and the corresponding incremental (flat) fuselage at one degree angle of attack, plus interference drag terms, are all calculated. The results are stored in common blocks /BØDSØL/ and /BPND/:

TBDLC  basic incidence streamwise lift
TBDLF  flat body streamwise lift (1° alpha)
CLCND, CLFND  lift coefficients of basic and flat solutions
CDCND, CDFND  drag coefficients
CMCND, CMBFND  pitching moment coefficients

Data for the isolated fuselage solution (no wing downwash) are also printed for reference.
USE: CALL FUSCF

INPUT: Configuration geometry contained in common blocks. Isolated fuselage force data in /B0DSFL/.

SUBROUTINES CALLED: TBLU1

Program LOAD9

PURPOSE: To read input data for lift analysis module

METHOD: Program LOAD9 reads input data (interface tape written by analysis subprogram of geometry module).

USE: CALL OVERLAY (SDA, 7, 8, 0)

INPUT: See user's manual

SUBROUTINES CALLED: None

Program TRNSFM9

PURPOSE: To convert input data to program units and set up grid system

METHOD: The wing, canard, and horizontal tail planforms are represented in the analysis program as a set of rectilinear elements, as described in the theory document (Part 1). Given the number of semi-span element rows (FNN) used to define the right-hand wing, TRNSFM interpolates the wing planform definition at the mid point of each semi-span row for leading edge and trailing edge X-value. These X values are converted to program scale (by means of RATIQ = y wing tip * BETA/FNN) and stored in arrays TXLE and TXTE, with XLEG and XTEG defining the wing centerline grid points.

Storage of wing pressure coefficients and surface slopes uses a space-conserving technique: the leading edge of a spanwise row is stored immediately after the trailing edge of the adjacent inboard row. The index array for the row storage is JEW, which gives the number of elements stored.
ahead of the leading edge of a selected row. The same storage arrangement is used for canard and horizontal tail, based on indices JEC and JEH.

The factor $\text{RATI}0$ is also used to scale the canard and horizontal tail arrays, as follows:

<table>
<thead>
<tr>
<th></th>
<th>Canard</th>
<th>Tail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leading edge array</td>
<td>TCXLE</td>
<td>THXLE</td>
</tr>
<tr>
<td>Trailing edge array</td>
<td>TCXTE</td>
<td>THXTE</td>
</tr>
<tr>
<td>Centerline leading edge</td>
<td>CXLE0</td>
<td>HXTE0</td>
</tr>
<tr>
<td>Centerline trailing edge</td>
<td>CXTE0</td>
<td>HXTE0</td>
</tr>
</tbody>
</table>

The maximum size of the pressure coefficient arrays for the right hand wing is 2500 for the wing, 200 for the canard, and 500 for the horizontal tail. Also, the maximum $X$ dimension of the configuration, to the most aft point on the wing, is 205 (program units). If any of these dimensions are exceeded, TRNSFM rescales the program units by reducing $\text{FN}0\text{N}$.

TRNSFM is also used to identify the grid elements associated with trailing edge flaps. Since the flap edges will usually not coincide with element edges, an approximate element array is used to represent the flaps and the input flap deflection angles are altered such that the product of flap area times deflection is the same for the input and the approximate program definition. The input flap area is computed in subroutine PHLAP.

**USE:**

```
call overlay (SDA, 7, 9, 0)
```

**INPUT:**

Configuration geometry read in program LOAD9

**SUBROUTINES CALLED:**

PHLAP, GGRID, TBLU1
Page Intentionally Left Blank
3.0 REFERENCES


Page Intentionally Left Blank
APPENDIX A

INTERACTIVE GRAPHICS SUBROUTINES

The graphics subroutines in the design and analysis program are described in this appendix, excepting the LRC standard CRT software routines.

Three general purpose subroutines are described first, followed by the subroutines associated with the individual modules:

Subroutine CScale

**Purpose:** Computes a plot origin and scale factor given an array of values.

**Method:** If the values in the input array are not equal, LRC subroutine ASCALE is called to compute an origin and scale factor. If the values in the input array are equal, the origin is set to that value -.5, and the scale factor is set to 2.0/length over which the data is plotted.

**Use:**

```
CALL CSCALE (ARRAY, S, N, K, DV)
```

**Input:**

- **ARRAY** = Array of data to be scaled.
- **S** = Length over which data will be plotted.
- **N** = Number of values in ARRAY.
- **K** = Interleave factor. (1=all points).
- **DY** = Number of divisions per inch of paper.

**Output:**

- **ARRAY (N*K+1)** = Plot origin
- **ARRAY (N*K+1+K)** = Plot scale factor

**Subroutines Called:**

ASCALE

Subroutine SHOW

**Entry SHOW**

To display an array of floating point values on the CDC250.

**Method:** LRC subroutines NOTATE and NUMBER are called to display a variable name, equal sign and a number of values on the same line. When the display line is complete, the vertical coordinate of the line is decremented by a preset value. Alternate entry point SHOW I is used for integer values.

**Use:**

```
CALL SHOW (A, N, BCD, NC)
CALL SHOWI (L, N, BCD, NC)
```
INPUT:  
A = Floating point variable or array  
L = Integer variable or array  
N = Number of values to display  
BCD = Hollerith label preceding the first value  
NC = Number of characters in BCD

COMMON/PBLOK/  
YORG = Y coordinate of display line  
VORG = X coordinate of first value to display. If values to be displayed are negative, the negative sign will be positioned at VORG-CS, VORG+VDEL-CS and VORG+VDEL+VDEL-CS  
VDEL = Vertical distance between lines  
VDEL = Horizontal distance between values  
CH = Character height  
CS = Spacing between characters (6/7*CH)  
THETA = Angle of display for label and values  
ND = Number of decimal places to display

OUTPUT:  
COMMON/PBLOK/  
XORG = Coordinate of first character in BCD

SUBROUTINES CALLED:  
NOTATE, NUMBER

EXAMPLE:  
R(1) = 1.25   -3.79   5.86  
R(4) = 6.73   9.50   -3.17

Subroutine SHOW3

PURPOSE:  
To display a label and three floating point variables on the CDC 250.

METHOD:  
LRC subroutines NOTATE and NUMBER are called to display a hollerith label and three values on the same display line. When the line has been displayed, the vertical coordinate of the line is decremented by a preset value.

USE:  
CALL SHOW3 (V1, V2, V3, BCD)

INPUT:  
V1 = First variable to display  
V2 = Second variable to display  
V3 = Third variable to display  
BCD = Hollerith label (10 character maximum)

COMMON/PBLOK/  
XORG = X coordinate of label

88
YORG = Y coordinate of display line
VORG = X coordinate of \( V_1 \)
VDEL = X distance between variables
  If negative, the variables are displayed at \( VORG-CS, VORG+VDEL-CS \)
  and \( VORG+VDEL+VDEL-CS \)
YDEL = Y distance between display lines.
CH = Character height
CS = Spacing between characters \((6/7*CH)\)
THETA = Angle of display for label and variables
ND = Number of decimal places to display.

SUBROUTINES Called:  NOTATE, NUMBER

EXAMPLE:

NAC1  -57.254  9.000  5.790
NAC2   69.874  16.785 -9.103

GEOMETRY MODULE

Configuration geometry is displayed and/or edited by program DISGEØM, a secondary overlay in the geometry module.

Program DISGEØM

PURPOSE: To display and/or edit configuration geometry
METHOD: A description of the display capability is presented in Appendix A of the user's manual.
USE: CALL OVERLAY (SDA, 5, 5, 0)

Subroutine ALTER

PURPOSE: To alter the wing camber surface shape to match a new trailing edge definition.
METHOD: The camber definition at each input airfoil is rotated about the airfoil leading edge point until the trailing edge point coincides with the new trailing edge definition. The new camber surface definition is stored in temporary array BBØRD

USE: CALL ALTER (8TE)
INPUT:  \[ \text{ZTE} = \text{Array of new trailing edge values.} \]

**COMMON/TEMP/**

\[
\text{LECODE} = +, \text{ZTE array consists of camber values} \\
\text{LECODE} = -, \text{ZTE array consists of camber values} \\
\text{+ the Z value of the leading edge}\]

**COMMON/WING/**

\[
\text{TBOARD} = \text{Original camber definition}\]

OUTPUT:  \[ \text{COMMON/TEMP/} \]

\[
\text{BBOARD} = \text{Altered camber definition}\]

Subroutine PLTSIB

**PURPOSE:**

To compute the configuration minimum and maximum X and Y coordinates, and the fuselage minimum and maximum \( \beta \) coordinates. A plot scale factor is also output.

**METHOD:**

Each configuration component is analyzed as to its coordinate values. The minimum and maximum values are stored in common. If the range of data in the X direction is greater than that in Y, SCALE is computed as:

\[
\text{SCALE} = \frac{\text{XMAX}-\text{XMIN}}{10.0}
\]

If the range of data in the Y direction is greater than that in X, SCALE is computed as:

\[
\text{SCALE} = \frac{\text{YMAX}-\text{YMIN}}{7.0}
\]

**USE:**

\[ \text{CALL PLTSIB} (\text{SCALE}) \]

**INPUT:**

All configuration geometry in COMMON.

**OUTPUT:**

\[ \text{SCALE} = \text{Plot scale factor} \]

**COMMON/\text{GV11/}**

\[
\text{XMIN} = \text{Minimum X value of configuration} \\
\text{XMAX} = \text{Maximum X value of configuration} \\
\text{YMIN} = \text{Minimum Y value of configuration} \\
\text{YMAX} = \text{Maximum Y value of configuration} \\
\text{BMIN} = \text{Minimum } \beta \text{ value of fuselage} \\
\text{BMAX} = \text{Maximum } \beta \text{ value of fuselage} \]
FAR-FIELD WAVE DRAG MODULE

The graphics subroutine (DIS080) in the far-field wave drag module is located in program PUT, overlay (3,6).

Subroutine PUT

PURPOSE: To display area plots and drag summary

METHOD: The CRT is used to display far-field wave drag module results as described in the user's manual, Appendix A.

USE

CALL DIS080 (S, B, BØ, C, RC, N)

INPUT:

S       Array of X values
B       Array of fuselage areas
BØ      Array of optimum fuselage areas
C       Array of overall configuration areas
RC      Array of restrained areas
N       Number of values in input arrays

NEAR-FIELD WAVE DRAG MODULE

The graphics program (DISP916) is called from the primary level of the near-field wave drag overlay.

Program DISP916

PURPOSE: To display near-field wave drag module results

METHOD: DISP916 displays summary results described in the user's manual, Appendix A

USE:

CALL OVERLAY (SDA, 6, 3, 0)

INPUT:

Summary data in common blocks

SKIN FRICTION DRAG MODULE

The graphics subroutine (DISP158) in the skin friction module is called from subroutine DRAG.
Subroutine DISP158

**PURPOSE:** To display skin friction module results

**METHOD:** DISP158 displays summary results described in the user's manual, Appendix A.

**USE:** CALL DISP158

**INPUT:** Summary data in common blocks

**WING DESIGN MODULE**

Display and editing in the wing design module is performed by three graphics programs called from the primary overlay.

**Program BUCKETP**

**PURPOSE:** To display drag-due-to-lift bucket plot

**METHOD:** BUCKETP is used to display the $K$ versus $C_{mo}$ plot plus design point solutions described in Appendix A of the user's manual.

**USE:** CALL OVERLAY (SDA, 10, 4, 0)

**INPUT:** Data in common block/BUCKET/

**Program EDBUCK**

**PURPOSE:** To permit editing of design point variables

**METHOD:** EDBUCK allows user to edit wing design program variables $C_{mo}$, $CL_{DIN}$, $RESTART$, and $CNSTR(1)$ through $CNSTR(4)$. In addition it allows the user to execute next design case or calculating edited design point, as described in Appendix A of the user's manual.

**USE:** CALL OVERLAY (SDA, 10, 5, 0)

**Program STOPPOP**

**PURPOSE:** To permit termination of wing design program cases

**METHOD:** STOPPOP allows user to exit from a series of wing design cases, as described in Appendix A of the
user's manual.

USE: CALL OVERLAY (SDA, 10, 6, 0)

LIFT ANALYSIS MODULE

The graphics displays in the analysis module consist of two overlays called from the primary overlay, plus two subroutines called from program FINISH.

Program DISTWST

PURPOSE: To display wing twist and permit editing of twist and several execution codes.

METHOD: DISTWST displays and permits editing of wing twist array, plus canard angles of attack, SYMM, WHUP, and ANYBOD. The display presentation is described in Appendix A of the user's manual.

USE: CALL OVERLAY (SDA, 7, 6, 0)

INPUT: Input variables in common blocks

Program DISUPWS

PURPOSE: To display fuselage upwash and wing pressure coefficient data

METHOD: DISUPWS provides display options for calculated fuselage upwash or wing pressure coefficient data, as described in Appendix A of the user's manual.

USE: CALL OVERLAY (SDA, 7, 7, 0)

INPUT: Data in common blocks/UPWSH/and/CPBUPW1/.

Subroutine DISTAB

PURPOSE: To display analysis module force coefficient summary and permit editing of horizontal tail angle of attack.

METHOD: DISTAB provides display of analysis program results as described in Appendix A of the user's manual.

USE: CALL DISTAB (NH, HTALP)
INPUT:  

N  NH  Horizontal tail angle of attack
H  loop index

T  HTALP  edited tail alpha

Subroutine EXL00P

PURPOSE:  
To display and permit editing of variables within DO loops.

METHOD:  
EXL00P displays current values of Mach number, configuration alpha, and canard alpha. It then permits editing of the next value to change within the cycle, as described in Appendix A of the user's manual.

USE:  
CALL EXL00P

INPUT:  
Configuration data in common blocks
"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—National Aeronautics and Space Act of 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons. Also includes conference proceedings with either limited or unlimited distribution.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include final reports of major projects, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546