NASA Contractor Report 181701

UTRC Report R88-956977-14

PROGRAM USER'S MANUAL FOR AN UNSTEADY HELICOPTER ROTOR-FUSELAGE AERODYNAMIC ANALYSIS

Peter F. Lorber

United Technologies Research Center

East Hartford, CT 06108

August 1988

(NASA-CR-181701) PROGRAM USER'S MANUAL FOR AN UNSTEADY HELICOPTER ROTOR-FUSELAGE AERODYNAMIC ANALYSIS (United Technologies Research Center) 107 p CSCL 01A N88-28047 Unclas G3/02 0160871

Provided to the Rotorcraft Aerodynamics Office,
Army Aerostructures Directorate, USAARTA-AVSCOM

Under Contract NAS1-17469,
Analysis of Helicopter Aerodynamic Vibratory Excitations

Program User's Manual for an Unsteady Helicopter Rotor-Fuselage Aerodynamic Analysis

TABLE OF CONTENTS

PROGRAMS TO EXAMINE THE OUTPUT

INTRODUCTION

The Rotor-Fuselage Analysis is a method of calculating the aerodynamic interaction between a helicopter rotor and fuselage. The rotor blades are modelled by a lifting line, the rotor wake by a series of trailing vortex filaments, and the fuselage by source panels. The fuselage and rotor solutions are coupled through the velocities induced by each component on the other. Prescribed geometric rules are used to displace the rotor wake about the fuselage. An unsteady solution is obtained at each time step (rotor azimuthal position), accounting for the time-dependent changes in the wake structure, in the circulation strength of each segment of the wake, and in the strength of each of the fuselage source panels. The analysis may be coupled with an aeroelastic or rigid blade response program to determine rotor control values and airloads. The solution will predict the unsteady velocities, pressures, and airloads on the fuselage, together with the position, induced velocities, and airloads on the rotor. The rotor inflow program has been furhter enhanced to allow computation of first level main rotor - tail rotor interactions. Reference 1 describes the technical approach used for rotor-fuselage and main rotor - tail rotor interactions, illustrates typical results of the analysis, and presents a limited correlation with experimental data.

This manual describes the structure and operation of the computer programs that make up the Rotor-Fuselage Analysis, the programs that prepare the input, and the programs that display the output. A separate section of the manual is devoted to each set of programs. Complete interactive terminal sessions are presented for each of the input and output programs. The programs were originally written in FORTRAN/77 for use on Perkin-Elmer 32 bit super-minicomputers using the OS/32 operating system, but they have also been converted for use on Digital Equipment Corporation VAX machines using the VMS operating system, and for the Control Data Corporation 6600 mainframe using the NOS and NOS/VE operating systems. This manual was written using typical Perkin Elmer interactive terminal sessions. Sessions on other computers may be slightly different. The graphics output was originally designed for use with the Tektronix IGL package, but has been converted to the Precision Visuals DI3000 system, which is used in each of the three computer systems.

This work was sponsored by the Rotorcraft Aerodynamics Office of the Army Aerostructures Directorate, USAARTA-AVSCOM, and administered by the National Aeronautics and Space Administration at the Langley Research Center under Contract NAS1-17469.

ROTOR-FUSELAGE ANALYSIS PROGRAMS

The rotor-fuselage analysis consists of a coupled run of at least two programs, the rotor inflow program (**F389SR**) and the fuselage panel program (**WABAT**). A flow chart of the sequence of program calls used in the analysis is shown in Fig. 1a. If the blade response is to be included, either the rigid response program (**GRP**) or the aeroelastic program (**G400**) is added, as shown in Fig. 1b. A global convergence is determined using a separate program (**CONVRG**). The technical approach used in these programs has been described in Ref. 1; this section describes the program structure, the required input and output files, and the changes that have been made to existing programs in order to incorporate the rotor-fuselage coupling.

Fuselage Panel Program (WABAT)

The fuselage panel program computes the velocity and surface pressure on each of the fuselage panels, integrates over all panels to determine the airloads on the fuselage, and computes the fuselage induced velocity components at the rotor disc. The isolated fuselage version of **WABAT** is described in Refs. 2 and 3. The program has been extensively modified for coupled rotor-fuselage operation. The **WABAT** program is operated in several different modes for this application. The initial run of the program is used to compute the geometric fuselage panel on fuselage panel influence coefficients, and does not include any rotor influence. Later runs of **WABAT** include rotor induced velocities, and use the previously computed influence coefficients. The final **WABAT** run calculates the total velocity and pressure at each fuselage panel and integrates the pressure to determine the aerodynamic load on the fuselage. All **WABAT** runs calculate the fuselage induced velocity at the rotor disc.

The **WABAT** program also has the capability to add steady lifting wings to the basic nonlifting fuselage solution. The lifting wing subroutines in **WABAT** have been modified for improved efficiency and legibility. The combination of lifting wings and rotor/fuselage coupling has not yet been used, but there it should be functional.

Program structure

The structure of the **WABAT** program is illustrated by the following subroutine diagram. The program executes from the top to the bottom of the diagram, and calls subroutines from left to right. For example, main program **WABAT** first calls subroutine **INPUT**, which in turn calls subroutines **VCOUPL** and **PROT**. **PROT** then calls subroutine **CPCXR**. Routines that are new or have had their function substantially changed are identified by an asterisk.

```
WABAT*--- INPUT* --- VCOUPL*
      -          --- PROT    --- GPCXR
      -
      --- MLSM   --- GLSG
      -          --- MWING   --- GCSX
      -          -           --- CAMBER
      -          -           --- NTERP
      -          -
      -          --- INFLU   --- INVAL
      -          --- LALFA
      -          --- LSIMEQ
      -          --- WAC
      -          --- DPM
      -          --- PMI
      -          --- VRTXPL  --- CAMBER
      -
      --- ROTPNL --- GPCXR
      -
      --- NVPANL
      -
      --- DFILG
      -
      --- VECTOR --- PNLID
      -          --- CRMIV3*
      -
      --- INVPNL --- PNLID
      -          --- CRMIV3*
      -          --- REFLCT
      -
      --- DFILA
      --- VEL389*
      --- GCPANL --- TNFLU   --- VNFLU  --- REFLCT
      -          -           --- PNLSOR
      -          -           --- VTRANS --- CRMIV3*
      -          -           --- REFLCT
      -          -
      -          --- NORVEL*--- CRMIV3*
      -
      --- SOLVE
      --- DFILA
      --- CHKERR
      --- OUTPUT*--- LINE
      -          --- OFFBOD*--- ROTOR
      -          -          --- TNFLU --- VNFLU   --- REFLCT
      -          -          -         --- PNLSOR
      -          -          -         --- VTRANS  --- CRMIV3*
      -          -          -         --- REFLCT
      -          -          -
      -          -          --- HAR
      -          -
      -          --- UNBOUND*
      -          --- CLPLOT*
```

```
--- COUPLV*--- HAR
-           --- ROTFIL*
-
--- PCOORD*--- CTOTPP*
```

Description of WABAT subroutines

A brief description of each routine will now be given.

CAMBER - Computes camberline for lifting wing.

CHKERR - Checks for error in solution of coefficient matrix.

CLPLOT - Writes fuselage airload plotting file.

COUPLV - Computes fuselage-induced velocities at rotor plane.

CRMIV3 - Computes determinants and inverse matrix.

CTOTPP - Converts from fuselage to tip-path-plane coordinates.

DFILA - Reads or writes file of coefficient matrix and source strengths.

DFILG - Reads or writes file of geometry information.

DPM - Computes pressure mode coefficients for lifting wing.

GCPANL - Computes geometric influence coefficients at panels.

GCSX - Generates surface points for lifting wing.

GLSG - Generates geometry for lifting wing.

GPCXR - Generates coordinates for rotation.

HAR - Computes azimuthal harmonics in rotor plane and writes file.

INFLU - Computes influence of load functions on control points for lifting wing.

INPUT - Reads input control file and fuselage panel geometry file.

INVAL - Computes load integral for lifting wing.

INVPNL - Inverts panel coefficient matrix.

LALFA - Computes effective angle of attack for lifting wing.

LINE - Computes pressure along fuselage waterlines.

LSIMEQ - Solves equations for lifting wing using Gauss/Jordan reduction.

MLSM - Computes pressure distribution for lifting wing.

MWING - Generates geometry for lifting wing.

NORVEL - Computes flow velocity components at a panel.

NTERP - Interpolates.

NVPANL - Reads in panels that have imposed normal velocity.

OFFBOD - Computes velocities at off-body points.

OUTPUT - Controls program output, including plot files and printout.

PCOORD - Computes and writes out panel locations in tip-path-plane coordinates.

PMI - Computes chordwise pressure integral for lifting wing.

PNLID - Identifies panel location and type.

PNLSOR - Computes effect of source panel on control point.

PROT - Rotates about x or z axes, as required.

REFLCT - Computes coordinates of reflected panels.

ROTFIL - Writes out file for plotting panel representation of the rotor.

ROTOR - Computes location of off-body points on rotor disc.

ROTPNL - Rotates about x axis.

SOLVE - Solves source strength equations using Gauss/Seidel.

TNFLU - Computes geometric influence coefficients for all panels on control point.

UNBOUND- Computes unsteady aerodynamic terms for evaluation of fuselage surface pressure coefficients

VCOUPL - Reads rotor/fuselage coupling and rotor induced velocity input files.

VECTOR - Computes tangent vector, normal vector, and local coordinates for a panel.

VEL389 - Resolves components of rotor induced velocities at each panel at each time step.

VRTXPL - Computes end points of bound vortices and generated vortex panels for lifting wing.

VTRANS - Converts velocities to local panel coordinates.

WABAT - Main Program.

WAC - Computes airloads for lifting wing.

Input and Output Files

There are a large number of input and output files that are used by the rotor-fuselage analysis. The files are automatically opened in each program. *Job control language in the command files will reassign some of these file names* to those appropriate to a particular case. The input files are created by the input preprocessing program, **FWINPT**. The operation of **FWINPT** is described in the appropriate section. Many of the actual file names are of the form 'TYPExxxx', where 'TYPE' identifies the function of the file, and 'xxxx' indicates the particular run that the file pertains to. Upper case characters in the file name are mandatory characters, and do not change from run to run. Lower case characters in the file name are arbitrary, and may be changed for each run. The following list gives each **WABAT** logical unit number, the file name that is opened in the **WABAT** program, and a description of contents of the file.

Unit 4.	WPLOT	Output file containing airload time histories for plotting. (Formatted - 132 characters/line)
Unit 5.	WCON	Input control file. (Formatted - 80 char/line)
Unit 6.	WLIST	Output listing file. (Formatted - 132 char/line)
Unit 8.	GCA.DAT	Scratch file of geometric influence coefficients. (Unformatted)
Unit 9.	GCB.DAT	Scratch file of geometric influence coefficients. (Unformatted)
Unit 10.	GCN.DAT	Scratch file of geometric influence coefficients. (Unformatted)
Unit 11.	SOLTN.DAT	Scratch file of intermediate solutions for restart. (Unformatted)
Unit 12.	GEOM.DAT	Scratch file of panel geometries for restart. (Unformatted)
Unit 13.	SOLVE.DAT	Scratch file of matrix solution for restart. (Unformatted)
Unit 14.	PANVEL	Input file of rotor induced velocities at each fuselage panel. (Formatted - 132 char/line)
Unit 15.	HARMON	Output file of azimuthal harmonics of fuselage induced velocities at the rotor disc.
Unit 16.	PANELS	Input file of fuselage panel geometry. (Formatted - 132 char/line)
Unit 17.	TUFTS	Output file of velocity vector at each panel for plotting. (Formatted - 80 char/line)

Unit 18. RFCO	Input file of rotor/fuselage coupling parameters. (Formatted - 80 char/line)
Unit 19. XYZ389.DAT	Output file of panel coordinates in tip-path-plane (F389SR) coordinates. (Formatted - 80 char/line)
Unit 20. ROTORP	Output file of panel representations of the rotor blades for plotting. (Formatted - 80 char/line)
Unit 21. CVFP	Output file of velocity and pressure time histories at each panel for plotting. (Unformatted)
Unit 22. CIRCUL	Input file of rotor bound circulation distributions, and related F389SR parameters. (Formatted - 80 char/line)

Input Data File Formats

Because they are written automatically by the preprocessor, it is generally unnecessary to directly change any of the input data files. A summary of the input file format will however be included here for completeness.

The **input control file (unit 5)** was described in Ref. 3 for the isolated fuselage version of WABAT. Therefore, the parameters that do not change in the coupled rotor/fuselage version will only be summarized, and those used for options not used here will not be described. The file is organized as follows:

Line 1 - TITLE (A72 Format) - Identifier String

Line 2 - REFA, REFL (2F10) - Reference area, length for force and moment coefficients

Line 3 - ALPHA, BETA (2F10) - Fuselage pitch and yaw angles (deg; up,right +)

Line 4 - (10I5)

MONLY - 1 to only perform Multhopp analysis
0 to perform total analysis (standard)

MPRNT - 1 for printer plots of Multhopp analyses
2 for plots and Multhopp influence coefficients
0 for neither (standard)

NGPRNT - 1 to print panel unit vectors
2 to print panel coordinates in local coordinates
0 to print neither (standard)

NPRNT - 1 to print solution and constant matrix
2 to print panel influence coefficients for panels between NCM1 and NCM2
0 to print neither (standard)

NCM1 - 0 (standard) or see above

NCM2 - 0 (standard) or see above

MPRINT2 - 1 to print Multhopp loading functions
2 to print Multhopp wing geometry
3 to print wing camber computations
0 to print neither (standard)

ISUM - 0 for full printout of panel input and output
1 for reduced (summary) printout (standard)

LINE 5 -	NSAV1 (10I5)	- 0 to not file geometry data on units 8,9,10,11,12 -1 to write new geometry data files (1st run) +1 to read existing geometry data files (restart)
	NSAV2	- 0 to not file solution data on unit 11 (standard) -1 to write new solution to data file +1 to read existing solution data file
	LSAV	- 0 (not currently used)
	NTAPE	- 1 to store solution matrix on unit 13 0 to not store (standard)
	NFILE	- n record number for stored starting solution matrix 0 if none (standard)
	NRITE	- n record number for write solution matrix 0 if none (standard)
	LTM	- n logical unit for time trace output 0 if no time trace (standard)
	IROTOR	- 0 if isolated fuselage 1 if 1st iteration of rotor/fuselage 2 if final iteration 3 if intermediate iteration
	IRSYM	- 0 for rotor/fuselage solution with non-symmetric fuselage geometry 1 for rotor/fuselage solution with symmetric fuselage geometry (standard)

(Note: the isolated fuselage version of **WABAT** had panel coordinate data at this point. This version has all panel data on unit 14. The first five lines of unit 5 are read in by subroutine input. The remaining lines are read in later in the execution of **WABAT**)

Line 5 -	IROTAT (I5)	- Section number requiring panel rotation (Ref. 1) 0 if none (standard)
Line 6 -	ISUCT (I5)	- Panel number requiring fluid suction/ejection 0 if none (standard)
Line 7 -	XMOM YMOM ZMOM (3F10)	- (x,y,z) of moment center for aerodynamic pitching, rolling, yawing moments
Line 8 -	MWL (10I5)	- Number of waterlines (y=constant) to compute surface pressures along 0 if none (standard)

LBOD		- Number of body lines (angle=constant) to compute surface pressures along 0 if none (standard)
NOB		- Number of offbody points to compute velocities at or, Number of radial stations on rotor (NPSI>0) (must match **F389SR**)
NPSI		- Number of azimuthal positions on half rotor disc to compute velocities at (should match **F389SR**) 0 if none
IROTOR<2	KARD	- 1 to write velocity harmonics at rotor to unit 15 when (standard) 0 to not write harmonics

(Note: additional lines are required here if MWL or LBOD are nonzero, see Ref. 3. The original version of **WABAT** had rotor geometry information here if NPSI>0. This information is now contained in the rotor fuselage coupling file, unit 18)

The **input panel file (unit 16)** has a format that is unchanged from that used by the original version of **WABAT**:

Line 1 -	TYPE,TITLE (A4,A68)	- Type of fuselage section ('NL,S' means non-lifting, symmetric, and is standard)
Line 2 -	NOP (I10)	- Number of panels in this section (the maximum total number is 800, or 400 panels for a half-fuselage)
Line 3 -	XOF,YOF,ZOF (8F10)	- Offset of this fuselage section (0,0,0 standard)
	XROT,ZROT	- Rotation of this section about x,z axes (0,0 standard)
Line 4	I (I10,6F10)	- Panel number (arbitrary)
	X1,Y1,Z1	- Coordinates of 1st corner of 1st panel
	X2,Y2,Z2	- Coordinates of 2nd corner of 1st panel
Line 5	I (I10,6F10)	- Panel number (arbitrary)
	X3,Y3,Z3	- Coordinates of 3rd corner of 1st panel
	X4,Y4,Z4	- Coordinates of 4th corner of 1st panel

Lines 4 and 5 are repeated for each of the NOP panels of this section.

If there is more than one section, then the data for the next section follows, with the same format as Lines 1+.

The final line of the file contains the string 'END '.

The **input rotor induced velocity file (unit 14)** is written by program **F389SR**, and has the following format:

Line 1	NFPTS (1X,3I5)	- Number of field points, which must be equal to the number of fuselage panels
	N1	- Starting azimuthal harmonic for the velocity Fourier series (0 is standard)
	N2	- Ending harmonic for the Fourier series. Should be equal to the number of time steps, with a maximum value of 18.
Line 2	IP (1X,I5,3F12)	- Point (panel) number, from 1 to NFPTS
	XFP,YFP,ZFP	- Coordinates (in terms of the tip-path-plane) of the point (panel centroid). (XFP,YFP,ZFP) refer to (downstream, sideways, vertical).
Line 3+	A0 (9E14)	- Mean velocity (in terms of velocity/flight speed) The velocities are in tip-path-plane coordinates, with the x component velocity listed first. U(t) = A0 + A(1)cos(azimuth) + A(2)cos(2*azimuth) + ... + B(1)sin(azimuth) + B(2)cos(2*azimuth) + ...
	A(1)..A(N2)	- Fourier amplitudes, cosine terms
	B(1)..B(N2)	- Fourier amplitudes, sine terms

The mean velocity and harmonic amplitudes are repeated for the y and z components, and the point number and velocity data are repeated for each point.

The **rotor/fuselage coupling file (unit 18)** is written by the input preprocessor, and rewritten by **F389SR**. Its format is:

Line 1	NTIME (1X,I3,I4,A72)	- Number of time steps in the solution

	NPANL	- Number of panels in complete fuselage (<=800)
	RTITLE	- Case identification string
Line 2	IRSYM (2I5)	- 0 for non-symmetric fuselage 1 for symmetric fuselage (this parameter is also in the input control file)
	ICOUPV	- 0 to not write unsteady fuselage-induced velocities at rotor 1 to write velocities (standard)
Line 3	NBLADE (2I5)	- Number of blades
	NRADIAL	- Number of radial stations along blade (<=15) (should match value of NOB in input control file)
Line 4+	RAD(15) (8F10)	- Radial stations along blade (radius / tip radius)
Line 6	XR,YR,ZR (8F10)	- Rotor hub location in fuselage coordinates
	RRT	- Rotor radius in hub coordinates
Line 7	CONE (8F10)	- Blade coning angle (deg, + up)
	ASG	- Longitudinal shaft tilt (+ forward, deg)
	BSG	- Lateral shaft tilt (+ right, deg)
	ASF	- Longitudinal flapping (+ forward down, deg)
	BSF	- Lateral flapping (+ right down, deg)
Line 8	XMU (8F10)	- Advance ratio
	AZ1	- Initial blade 1 azimuth (0 deg standard)
	AZ2	- Final blade 1 azimuth (360 / NBLADE standard)
	ALPHA	- Fuselage pitch angle (+ nose up, deg)
	BETA	- Fuselage yaw angle (+ right, deg)

The input circulation file (unit 22) is written by F389SR, and contains the blade bound circulation strengths at each azimuthal position, and other related parameters:

Line 1	ITOT (2I5)	- Number of radial segments along blade (9 standard)
	JTOT	- Number of azimuthal stations for circulation
	' ***CIRCULATIONS AND PEAK CIRCULATIONS***'	
Line 2+	CIRC(1,1..JTOT) (10F7.2)	- Bound Circulations at radial station 1 (ft**2/sec)
	The circulations are repeated for each of the ITOT radial stations.	
Line 3+	PCIRC(1..JTOT)	- Maximum circulation at each azimuth station (ft**2/sec)
Line 4+	DPSI (1X,5E14.7)	- Azimuth increment (deg)
	MTPP	- Tip-path-plane advance ratio
	ARAT	- Blade aspect ratio
	OMR2	- Blade speed * radius (ft**2/sec)
	CTHR	- Geometric mean chord (ft)
	R	- Blade radius (ft)
	V	- Flight speed (ft/sec)
	VIMOM	- Momentum-induced downwash velocity (ft/sec)
Line 5+	RS(1..ITOT) (1X,E14.7)	- Blade radial stations (ft / R)
Line 6	'**END OF PUNCHED CARDS FOR CIRCULATIONS**'	

Output File Formats

The **output airload plotting file (unit 4)** is used by program **PLOTWAB** to create time histories of the aerodynamic forces and moments on the fuselage. Its format is as follows:

Line 1	NL (1X,3I5,1X,A8,1X,A8)	- Number of dependent variables (6)
	IX	- Number of values of 1st independent variable (0)
	IY	- Number of values of 2nd independent variable (set to the number of time steps for a complete rotor revolution)
	XLABEL	- String for 1st independent variable (' ')
	YLABEL	- String for 2nd independent variable ('Blade 1 Azimuth')
Line 2+	PSI(IY) (1X,10E13)	- Azimuth positions (deg)
Line 5	LLDS (1X,I4,1X,A16)	- Dependent variable number (1-6)
	CLABEL(LLDS)	- String for dependent variable number LLDS ('Lift, CL', etc.)
Line 6+	CLDS2(IY) (1X,10E13)	- Values of dependent variable number LLDS

Lines 5 and 6+ are repeated for each dependent variable.

The **output azimuthal harmonic file (unit 15)** contains Fourier series of the fuselage-induced velocities at the rotor disc. The velocities are scaled by the flight speed, and are expressed as the sum of the mean velocity and the cosine and sine harmonic amplitudes. The format is as follows:

Line 1	NH (2I3)	- Number of harmonics
	KK	- Velocity component number, ordered as: 2 - tangential 3 - radial 1 - axial
Line 2	A0 (E14)	- Mean velocity at first radial station

Line 3+ A(NH),B(NH) - Harmonic velocity amplitudes:
(5E14) A terms are cosine amplitudes,
B terms are sine amplitudes

Lines 2 and 3+ are repeated for each of the NRADIAL stations, and then repeated for each of the 3 velocity components.

The **output tuft velocity file (unit 17)** is used by program **PLOTWAB** to display a surface velocity vector at each panel at each time step. The velocities are scaled by the flight speed. The format is as follows:

Line 1 ' TUFT ' - Identifies file type
(A6)

Line 2 NTIME - Number of time steps
(1X,I3,I5,1X,F10,1X,F10,1X,I3)

NBOD - Number of panels in complete fuselage

AZA - Starting azimuth angle (0 deg)

AZB - Azimuthal increment (360/(NBLADE*NTIME) deg)

NBLADE - Number of Blades

Line 3 X,Y,Z - Panel centroid location in WABAT coordinates
(7F10)

VR - Velocity magnitude

VX,VY,VZ - Velocity components in the (x,y,z) directions

This line is repeated for each of the NBOD panels at the first time step. The lines are then repeated for the remaining time steps.

The **panel coordinate file for F389SR (unit 19)** contains the location of each panel centroid in tip-path-plane coordinates. These values determine where the rotor-induced velocities will be computed. It also contains the mean location of the upstream and downstream edges of each panel. This information is used in the close panel-vortex interaction model of **F389SR** to compute the average velocity along a panel. The format is as follows:

Line 1 IC - Numbers of panels in the complete fuselage

Line 2+ X,Y,Z - Panel centroid coordinates in tip-path-plane
(9F8) system

This line is repeated (3 coordinate triplets per line) for all IC panels.

Line ++ XUP,YUP,ZUP - Coordinates of upstream edge of panel
(9F8)

XDOWN,YDOWN,ZDOWN - Coordinates of downstream edge of panel

This line is repeated (3 coordinate triplets per line) for the upstream and downstream locations for all IC panels.

The **rotor panel file (unit 20)** contains a panel representation of the rotor blades' position at each time step. It is used by the fuselage and wake plotting program, **VABPLOT**. The format is as follows:

Line 1 ' ROTOR' - Identifies rotor panel file
(A6)

Line 2 NTIME - Number of time steps
(1X,I3,I5,1X,F10,1X,F10,1X,I3)

NP - Number of rotor panels (NBLADE*NRADIAL)

PSI(1) - Starting Azimuth angle (0 deg)

DELPSI - Azimuthal Increment (360 / (NBLADE*NTIME) deg)

NBLADE - Number of Blades

Line 3 NP - Number of Rotor Panels
(5X,I5)

Line 4 XOF,YOF,ZOF - Offset of the rotor (0,0,0)
(8F10)

XROT,ZROT - Rotation of the axes (0,0)

The next NP*2 lines give the coordinates of the panels at the first time step, in the same format as the fuselage panel file (unit 16). These lines are repeated for each time step.

The **velocity and pressure time history output file (unit 21)** is an unformatted file that is read by program **CVFPLT** and used to plot conditions at a single panel or along a line of panels. The file is organized in a series of logical records:

Record 1 NTIME - Number of time steps

NBOD - Number of panels in complete fuselage

AZA - Starting azimuth (0 deg)

	AZB	- Azimuthal increment (deg)
	NBLADE	- Number of blades
	XMU	- Advance ratio
	RRT	- Rotor radius (WABAT coordinates)
	XR,YR,ZR	- Rotor hub location (WABAT coordinates)
	ASF	- Longitudinal flapping
	BSF	- Lateral flapping
	ASG	- Longitudinal shaft tilt
	BSG	- Lateral shaft tilt
	(XCC(I),YCC(I),ZCC(I),I=1,NBOD)	- Panel centroid locations (WABAT coordinates)
RECORD 2	ITIME	- Time step number (1-ITIME)
	CP(NBOD)	- Pressure coefficient at each panel
	VX(NBOD)	- Total x component velocity at each panel (scaled by flight speed)
	VY(NBOD)	- Total y component velocity
	VZ(NBOD)	- Total z component velocity
	CX(NBOD)	- Force component in x direction
	CY(NBOD)	- Force component in y direction
	CZ(NBOD)	- Force component in z direction
	VXR(NBOD)	- Rotor induced velocity in x direction (scaled by flight speed)
	VXY(NBOD)	- Rotor induced velocity in y direction
	VXZ(NBOD)	- Rotor induced velocity in z direction

This record is repeated for each time step.

Job Control Files for WABAT

There are different job control files for each computer system. For the Perkin Elmer two job control files used for **WABAT** during a coupled rotor/fuselage run. The first, 'WABAT.CSS,' is used for the initial run of the **WABAT** task. The scratch files (units 8, 9, 10, 11, 12, and 13), output files (units 4, 6, 15, 17, 19, 20, and 21), and one input file (unit 14) are allocated at this time. All files are assigned, and the program started. The second job control file, 'WABAT2.CSS,' is used for restarts of the **WABAT** task. No files are allocated since the stored geometric influence coefficients are reused. The file assignments are the same as for the initial **WABAT** run.

For VAX/VMS, there is a single job control file for **WABAT**, 'WABAT.COM'. This file defines which file is associated with each of the logical file names that will be opened during execution of **WABAT**. The scratch and output files are opened using TYPE='UNKNOWN', which will only create a new file if an existing file could not be found. Therefore repeat runs may use the same job control language as used during the initial run:

```
@WABAT Input control file name, Input panel file name, Run ID string
```

The 'Run ID string' is a four character string that identifies the run.

For NOS/VE on the CDC, all input, output, and scratch files must be created on the temporary '$LOCAL' catalog. The input files must first be copied from the permanent '$USER' catalog to the $LOCAL catalog. This copying process changes the specific file names on $USER to the generic names on $LOCAL, which correspond to the files that are opened during execution of **WABAT**. The procedure that is used is described in the final section of this report.

Rotor Inflow Program (F389SR)

The rotor inflow program (Ref.4) computes the distribution of rotor wake induced velocities based on a prescribed wake model. Section operating conditions are prescribed from blade motion and pitch angle data determined by a blade response analysis (**GRP** or **G400**) and from the fuselage-induced velocity determined by **WABAT**. Blade loading and bound circulation distributions are computed based upon section lift coefficient data. The strength of the trailing vortex elements is determined from the spanwise variation in bound circulation. The rotor-induced velocities are computed at the rotor and at external locations such as the fuselage panels.

There are four types of **F389SR** runs that are used during a coupled rotor-fuselage case. The first **F389SR** run computes the wake geometry and stores the geometric influence coefficients for the induced velocities induced by the rotor and wake on the rotor. The wake geometry is modified to include the displacement effects of the fuselage. The first run also calculates the blade bound circulation distribution based upon an initial estimate of the fuselage-induced velocities and blade response. The second **F389SR** run computes the geometric influence coefficients for the velocities induced by the rotor and wake on the fuselage. The induced velocities at the fuselage panels are then calculated using the initial values for the bound circulation strengths. After the fuselage and blade response programs have updated the induced velocities and blade operating conditions, the third **F389SR** run computes a new bound circulation strength. The fourth **F389SR** run recomputes the rotor and wake induced velocities at the fuselage. Both the third and fourth **F389SR** runs use the precomputed geometric influence coefficients.

The **F389SR** program provides the primary interface between the other programs in the rotor/fuselage analysis. Both the fuselage program (**WABAT**) and the blade response program (**GRP** or **G400**) interact only with **F389SR**. The interaction occurs through the numerous data files that are written by each of the programs. The files and their functions will be described below.

Program Structure

The structure of the **F389SR** program is illustrated below for the coupled rotor-fuselage version. The program executes from top to bottom and calls subroutines from left to right. New or functionally changed routines are marked with an asterisk.

```
F389SRPG --- SECONE --- DATAIN*--- LOAD1*
       -         -             --- LINKUP*--- SETTAB
       -         -                        --- INTER1
       -         --- PRINIT*
       -
       --- SECTWO --- LAERO
       -         --- AFDATA --- CLOAD  --- CLCHEK --- CLDUMP
```

```
-          --- CALATT
-          --- CALBET --- HARCAL
-          --- BLDXYZ
-          --- CLASWK
-          --- UAERO  --- INTER1
-          -
-          --- BCIN   --- SETTAB
-          -          --- RDHARM --- HARCAL
-          -          -          --- INTER1
-          -          -
-          -          --- INTPO
-          -
-          --- VTHETA --- SETTAB
-          -          --- HARCAL
-          -          --- INTPO
-          -
-          --- AERPAR --- BLIN4
-          -
-          --- READW1 --- SETTAB
-          -          --- INTER1
-          -
-          --- PRINT2 --- INTPO
-          -
-          --- READFP*--- TAILLC*
-                     --- FPCOOR*
-
--- SECTRE --- SECFOR*--- RFCWRT*
           -          --- DISWAK
           -          --- TIPWAK
           -          --- GWAKE2 --- GWCOEF --- LINEAR
           -          --- PCOORD
           -          --- FUSFIL*--- SEGCHK*--- BODRAD*--- JNTERP*
           -                                --- RESEG* --- JNTERP*
           -
           --- SECFIV --- GCJACK*--- GCCAL* --- VORPNC*
                      -                     --- WRTGC
                      -
                      --- SOLVEM --- INTMAT --- PRTGC
                      -          -          --- SETTAB
                      -          -          --- RDHARM
                      -          -          --- INTPO
                      -          -
                      -          --- SIEDEL --- CFUNCT
                      -          -          --- VICAL
                      -          -
                      -          --- VICAL
                      -
                      --- CIRPRT*--- INTPO
                      -          --- CONPLT
                      -          --- PLTFIL*
                      -
                      --- INTVEL --- SETTAB
```

```
-            --- INTER1
-            --- INTPO
-
--- ANALYS --- INTER1
-            --- HARM1
-            --- WRHARM
-            --- PCHHAR
-
--- CIRPCH --- READIJ
-            --- INTPO
-
--- FPVICL*--- HARMW*
             --- HARM1
```

Additionally, routines RREC, WREC, SREC, SKIPFL, and REW are used to manipulate the scratch data files, and are called from many of the other routines.

Description of F389SR Routines

The following is a brief description of each routine used in F389SR.

AERPAR - Computes blade element aerodynamic quantities required for circulation analysis.

AFCIP - Used in the utility airfoil data package.

AFDATA - Reads real airfoil data tables.

ANALYZE - Controls harmonic analysis of the inflow solution and associated output options.

BCIN - Reads in fuselage induced velocities at the rotor disc.

BLDXYZ - Computes blade coordinates, shaft angle quantities, and other wake-related parameters.

BLIN4 - Performs bi-variant interpolation.

BODRAD - Determines whether a point is inside or outside the fuselage.

CALATT - Calculates rotor attitude and transformation quantities.

CALBET - Calculates rotor blade position from input flapping response data.

CFUNCT - Calculates correction terms for non-linear solution.

CIRPCH - Writes and reads blade bound circulation file.

CIRPRT - Computes and writes out angle of attack, stall indicator, circulation, and inflow velocities.

CLASWK - Computes coordinates of classical undistorted helical wake.

CLCHEK - Checks input airfoil data format.

CLDUMP - Writes out input airfoil data if format is incorrect.

CLOAD - Reads in real airfoil data.

CONPLT - Writes printer plots of contours around rotor disc.

DATAIN - Controls input of variable data and computes control parameters.

DISWAK - Controls input of wake geometry from external file.

F389PG - Main Program.

FPCOOR - Computes field point coordinates (for tail rotor?).

FPVICL - Calculates induced velocities at field points.

FUSFIL - Displaces wake filaments about the fuselage.

GCCAL - Calculates geometric influence coefficients for a single point.

GCJACK - Controls influence coefficient computation.

GWAKE2 - Calculates tip vortex locations from generalized wake model.

GWCOEF - Calculates generalized wake model coefficients.

HARCAL - Evaluates particular value of a quantity from the harmonic series.

HARM1 - Computes harmonic (Fourier) series of a function.

HARMW - Computes harmonic series for velocities at fuselage.

INTER1 - Performs linear interpolation.

INTMAT - Reads and sets up matrix coefficients for solution.

INTPO - Prints out two-dimensional arrays as function of radius and azimuth.

INTVEL - Interpolates and prints out inflow velocities.

JNTERP - Interpolates for wake displacement routines.

LAERO - Reads in linearized airfoil data.

LINEAR - Performs linear interpolation or extrapolation.

LINKUP - Reads rotor response data files written by GRP or G400.

LOAD1 - Reads variable input data in 'loader' format.

PCHHAR - Writes harmonic series.

PCOORD - Prints wake geometry coordinates.

PLTFIL - Writes plotting file of circulations and induced velocities.

PRINIT - Prints variable input data and description.

PRINT2 - Prints blade element aerodynamic quantities.

PRTGC - Prints geometric influence coefficients.

RDHARM - Reads harmonic series input data.

READFP - Reads field point locations.

READIJ - Controls zeroing of circulation on the rotor disc.

READW1 - Reads non-induced velocities from external file.

RESEG - Resegments a section of a wake filament.

REW - Rewinds a file.

RFCWRT - Reads and rewrites the rotor/fuselage coupling file.

RREC - Reads a data array.

SECFIV - Obtains circulation solution and analyzes results.

SECFOR - Sets up wake geometry and calculates influence coefficients.

SECONE - Reads input control data.

SECTRE - Calls SECFOR and SECFIV.

SECTWO - Reads input data and sets up aerodynamic parameters.

SEGCHK - Checks if wake segment is inside or outside of fuselage.

SETTAB - Sets up table for interpolation.

SIEDEL - Solves circulation matrix using Gauss-Siedel iterations.

SKPFIL - Skips ahead on tape.

SOLVEM - Controls the circulation solution.

SREC - Reads record from tape.

TAILLC - Reads in and computes tail field point coordinates.

TIPWAK - Reads tip wake geometry and overwrites classical wake coordinates.

UAERO - Reads linearized airfoil data from blade response analysis.

UNINT - Interpolates to maintain slope continuity between adjacent intervals.

VICAL - Calculates induced velocity using circulation solution and geometric influence coefficients.

VORPNC - Computes influence coefficients for close vortex/panel interaction.

VTHETA - Reads input blade element pitch distribution from blade response analysis.

WREC - Writes record to tape.

WRHARM - Writes harmonic series.

WRTGC - Writes geometric influence coefficients to scratch file.

Input, Output, and Scratch Files

Program **F389SR** uses files for input, output, scratch storage, and for coupling with **WABAT**, **G400**, and **GRP**. The file name conventions are explained in the description of the **WABAT** program.

Unit 1.	TEMP1	Scratch file.
Unit 2.	TEMP2	Scratch file.
Unit 3.	TEMP3	Scratch file.
Unit 4.	DEBUGF	Output file for optional debug printout.
Unit 5.	FCON	Input control and airfoil data file. (Formatted - 80 characters/line)
Unit 6.	FLIST	Output listing file. (Formatted - 132 char/line)
Unit 7.	CIRCUL	Input/Output bound circulation file. (Formatted - 80 char/line)
Unit 8.	XYZ389	Input file of fuselage panel coordinates. (Formatted - 80 char/line)
Unit 12.	TEMP12	Scratch file.
Unit 13.	G400W1S	Input file of non-induced axial velocities from **G400** program. (Formatted - 80 char/line)
Unit 14.	VINFLOW	Output file of induced velocities at rotor for **G400** or **GRP**. (Formatted - 80 char/line)
Unit 15.	HARMON	Input file of harmonic series of fuselage induced velocities at rotor from **WABAT**. (Formatted - 80 char/line)
Unit 16.	GFCOUP	Input control file from **G400** or **GRP**. The format for a **G400** file differs from that for a **GRP** file.
Unit 17.	N17	Input control file from **GRP**. Not presently used.
Unit 18.	SOLVEGC	Scratch file of rotor-on-rotor influence coefficients.
Unit 19.	CIRC	Output circulation listing file. Not used.
Unit 20.	RFCO	Input/Output rotor fuselage coupling file. (Formatted - 80 char/line)

Unit 21.	PANVEL	Output file of rotor velocities at fuselage. (Formatted - 132 char/line)
Unit 23.	LINEARA	Input file of airfoil data from GRP or G400. (Formatted - 80 char/line)
Unit 24.	WDP	Input wake displacement body file. (Formatted - 132 char/line)
Unit 25.	TEMP25	Scratch file.
Unit 26.	WAKE	Output wake filament plotting file. (Formatted - 40 char/line)
Unit 27.	FPLT	Output circulation and velocity plotting file. (Formatted - 132 char/line)
Unit 28.	TEMP28	Scratch file.
Unit 29.	FIELDGC	Scratch file of rotor-on-fuselage influence coefficients.
Unit 30.	TEMP30	Scratch file of wake geometry.

Input Data File Formats

In the rotor/fuselage coupling mode, all input files are written automatically by either the preprocessor program (**FWINPT**), the fuselage program (**WABAT**), or by a blade response program (**GRP** or **G400**). The file formats are as follows:

The **input control file (unit 5)** contains an array of the basic variables used to define the conditions and control the operation of F389SR. It is normally written by **FWINPT**, but may be edited by hand. The information in this file may be overwritten by the **GRP** or **G400** coupling data on unit 16. The file format is referred to as 'LOADER format', and each line is as follows:

```
LL  L   DATA(L)    DATA(L+1)  ... DATA(L+LL-1)
(I2,I4,5E12 Format)
```

where: LL is the number of data items in the line
L is the starting location number
DATA are the values at each location

Two ending lines are usually written after the LOADER data on unit 5:

```
-1   99-0.10000E+01
-1   99-0.10000E+01
```

If no blade response program is to be used, linearized airfoil data is added to unit 5 following these lines. The format is described in Ref. 4.

The **input circulation file (unit 7)** is written during the first and third runs of **F389SR** and read back in during the second and fourth runs. It contains the blade bound circulations (in ft*ft/sec) at each radial position and azimuthal station. The contetnts and format of this file are described in the section of **WABAT**.

The **input fuselage panel coordinate file (unit 8)** is written by the **WABAT** program. Its format is described in the section on **WABAT**.

The **input file of non-induced axial velocities (unit 13)** is only used if coupling with **G400** is selected. It is written by **G400**, and contains the axial velocity caused by rigid blade motion and by blade torsional and bending flexibility. The velocity is given at each radial station and at each azimuthal position.

The **input file of fuselage induced velocities at the rotor disc (unit 15)** is written by **WABAT**. Its format is described in the section on **WABAT**.

A **second input control file (unit 16)** is read in if a blade response program is being used. If **GRP** is being used, the control file is in LOADER format, and overwrites the array elements read in from unit 5 at locations 1-3, 5-85, 87, 163-184, 204, 218, and 363-381. These values set the flight speed, rotor tip speed, sound speed, rotor radius, number of blades, hinge offset, blade chord, number of radial stations, wake transport velocity, rotor shaft angle, blade collective, twist, cyclic, lag-angle, coning, flapping harmonics, pitch-flap coupling, blade sweep, and blade thickness. The remaining data read in from unit 5 remain unchanged.

If **G400** is being used, the control file (unit 16) is in (5G12) format, and directly overwrites the values for the flight speed, tip speed, sound velocity, rotor radius, number of blades, blade chord, hinge offset, radial stations, wake transport velocity, shaft angle, collective, twist, cyclic, lag-angle, flapping harmonics, coning, pitch-flap coupling, blade sweep, and blade thickness.

The **rotor/fuselage coupling file (unit 20)** is read in and rewritten by F389SR. This file and its format are described in the section on **WABAT**. Program **F389SR** uses the values of the rotor radius and hub location from this file to determine the relative rotor/fuselage position for the purpose of displacing the wake about the fuselage. The file is rewritten to update the values of the rotor coning, flapping, and shaft tilt angles, and the fuselage pitch and yaw angles to the current values computed by the blade response program. This provides the link between the blade response programs and the fuselage program.

The **input file of airfoil data (unit 23)** is written by the blade response program and supersedes any airfoil data read in from unit 5.

The **input wake displacement body data file (unit 24)** represents the shape about which the wake will be displaced. This file is written by the interactive input program **WCROSS**, and is described in the section on that program.

Output File Formats.

The **output file for debug printout (unit 4)** is only used when certain write statements in **F389SR** are activated. It needs to be assigned to a file only in such cases.

The **output listing file (unit 6)** contains the basic **F389SR** printout of conditions and results. Its contents are described in Ref. 4.

The **output rotor induced velocity file (unit 14)** contains the harmonic series of the velocities induced by the rotor, wake, and fuselage at the rotor disc. This file is written if a blade response program is being used. The velocities are normalized by the flight velocity. For each velocity component (radial, tangential, and axial) the file contains the number of harmonics, the velocity component number, and the harmonic series for the velocity at each radial station:

Line 1 - NHARM (2I3) - Number of harmonics (<=18)

INH - Velocity component number (1=axial, 2=tangential, 3=radial)

Line 2 - A0 (E14.6) - Mean velocity

Line 3+ - A(1)...A(NHARM) (5E14.6) - Cosine terms of harmonic series

Line 4+ - B(1)...B(NHARM) - Sine terms of harmonic series

The **output rotor-induced velocity at the fuselage file (unit 21) is** written by the second and fourth runs of **F389SR**. It is described in the section on **WABAT**.

The output wake filament plotting file (unit 26) is written if the proper **F389SR** control variable is selected (see **FWINPT** documentation). It is used by plotting program **WABPLOT** to define the wake geometry. The wake coordinates are in the tip-path-plane system, and are given for each time step of the solution. The format is as follows:

Line 1 - ' WAKE ' (A6) - Identifies wake file

Line 2+ - BJTOT (8I5) - Number of time steps

	NWPT(1)...NWPT(BJTOT)	- Number of wake segments at each time step
	ISTNO	- Number of radial stations
Line 3 -	LTIME (10I5)	- Time step (1...BJTOT)
	IIB	- Blade number
	KK	- Filament number at this time step
	MPTS	- Number of segments at this filament
	ISTAT	- Radial station number (1...ISTNO)
Line 4	MM (5X,I5,3F10.5)	- Segment number along filament (top to bottom)
	X,Y,Z	- Coordinates of segment endpoint

Line 4 is repeated for each segment along a filament. Lines 3 and 4+ are repeated for each filament at each blade, at each time step. This file can obviously become quite long.

The **output circulation and velocity plotting file (unit 27) is used by** plotting program **PLOTWAB** to generate plots of the radial and azimuthal distributions of bound circulation, blade element angle of attack, and rotor and fuselage induced velocity components at the rotor. The format of this type of file is described in the section on **WABAT**, and the contents of the file in the section on **PLOTWAB**.

Scratch File Lengths

The **F389SR** scratch files on units 1, 2, 3, 12, 25, 28, and 30 are used only within a single F389SR run. The scratch files for the geometric influence coefficients for the rotor-on-rotor velocities (unit 18) and for the rotor-on-fuselage velocities (unit 29) must be saved between the F389SR runs for a given rotor/fuselage case. The largest storage is required for unit 29. Its size in bytes is equal to:

(# Radial Sta) x (# Azimuthal Sta) x (# Time Steps) x
(# Fuselage Panels) x (3 coordinates) x (4 Bytes/value)

For a typical four-bladed rotor case, with 9 radial stations, a 7.5 deg azimuthal spacing (implying 12 time steps and 48 azimuthal stations), and a fuselage with 428 panels, the scratch file will be 26.6 MBytes in length.

Job Control Files for F389SR

For the Perkin-Elmer, there is a single job control file for **F389SR**, 'F389R.CSS'. It allocates the single run scratch files (units 1, 2, 3, 12, 25, 28, and 30), assigns all files, and starts the program. The saved scratch files (units 18 and 29), the listing file (unit 6), and the output files (units 7, 14, 26, and 27) must be allocated before the first **F389SR** run.

For VAX/VMS, the **F389SR** job control file is 'F389SR.COM'. It functions in a similar manner to the **WABAT** job control file, and is started using:

```
@F389SR Input control file, Input wake displacement file,
        Run identification string
```

The procedure for executing **F389SR** using NOS/VE is described at the end of the report.

Aeroelastic Blade Response Program (G400)

The aeroelastic blade response program determines the time history of the dynamic response of the rotor system. This program is described in detail in Ref. 5. Only one modification (described below) was made to G400 for use with the coupled rotor/fuselage analysis. The coupling between **G400** and **F389SR** is otherwise unchanged. There is no direct link between G400 and **WABAT**. The description of **G400** will be limited to how it is used during a coupled rotor/fuselage analysis. Further information on the methodology, the assumptions, and a detailed listing of the input and output variables is given in Ref. 5.

Program **G400** is run in two ways during a rotor/fuselage coupled analysis. The initial run computes the blade response without rotor and wake induced inflow velocities from **F389SR**. Follow-on **G400** runs include the inflow velocities from **F389SR**, and utilize several data files written during the first **G400** run to reduce computational time.

The following files are used during a **G400** run:

Unit 1.	G400U01	Scratch file. (Formatted)
Unit 5.	G400CON	Input control file. (Formatted - 80 char/line)
Unit 6.	G400LIS	Output listing file. (Formatted - 132 char/line)
Unit 12.	G400U12	Scratch file. (Formatted)
Unit 13.	G400W1S	Output W1 velocity file for **F389SR**. (Formatted - 132 char/line)
Unit 14.	VINFLOW	Input inflow velocity file from **F389SR**. (Formatted - 80 char/line)
Unit 16.	GFCOUP	Output coupling parameter file for **F389SR**. (Formatted - 132 char/line)
Unit 23.	LINEARA	Output airfoil data file for **F389SR**. (Formatted - 132 char/line)
Unit 24.	G400U24	Scratch file. (Formatted)
Unit 26.	G400U26	Scratch file. (Formatted)

The input control file (unit 5) is in a modified 'LOADER' format, and is described in Ref. 5, and summarized in the section of this documentation that describes the input preprocessor, **FWINPT**. The first two lines of the control file contain title and initialization parameters, and should be:

Line 1 ' 1'

Line 2 ' -4-4-4 Title String for this run'

These lines are followed by the two-dimensional airfoil data in 'CLOAD' format. This data will be transferred to **F389SR** through unit 23. The 'LOADER' format control data is listed after the airfoil data. The following values should be set for proper coupling with **F389SR**:

A(3) =	0.97	Tip loss factor
A(60) =	7.5 deg?	Azimuth increment for **F389SR**
A(65) =	0	Do not read variable inflow data
A(66) =	0	Do not use variable inflow data
S(23) =	1	Write out blade response for **F389SR**
S(26) =	0	Not used
S(32) =	0	Not used
S(33) =	1	Mode of data transfer to **F389SR**
S(34) =	1	Create/use initial condition file

The other values in the input control file should be set appropriately (Ref. 5). As discussed in the section on **F389SR**, they will overwrite the values selected for the **F389SR** input data file. After the 'LOADER' format data, there is additional **G400** input, as described in Ref. 5.

The output files for **F389SR** (units 13, 16, and 23) were discussed in the section on **F389SR**.

Repeat runs of **G400** use a different input control file, 'G4002ND'. This file contains only parameters that define the solution process and coupling. All information on structural and flight conditions is read in from the scratch files that were written during the first **G400** run. The control file has the following values:

A(3) =	1.0	Tip loss factor
A(29) =	0.0	Inflow ratio
A(65) =	1.0	Read variable inflow data from **F389SR**
A(66) =	1.0	Use variable inflow data from **F389SR**
S(23) =	1.0	Write blade response for **F389SR**
S(33) =	1.0	Mode of data transfer to **F389SR**
S(52) =	1.0	Analyze and write spanwise airload harmonics
S(60) =	1.0	Multiple of blade number for hub motion analysis

The only modification that was made to the **G400** program for the rotor/fuselage analysis is concerned with this restart capability. Subroutine 'RESETQ' was modified so that the values of A(65) and A(66) that have been read in from unit 5 are not overwritten by the values that were saved on unit 26 from the initial **G400** run. Without this modification, it would not be possible to start with a no inflow **G400** run and then go to a repeat run that includes variable inflow from **F389SR**.

The scratch data files for **G400** must be preserved between the runs of a rotor/fuselage case.

For the Perkin-Elmer, there are separate job control files for the initial and repeat runs of **G400**. File 'G4001ST' controls the initial run. The scratch files (units 1, 12, 24, and 26), and output files (units 6, 13, 14, 16, and 23) are allocated. All files are assigned and the program is started. File 'G4002ND' is used for restarting **G400**. This file allocates a listing file (unit 6), assigns all files, and starts **G400**.

For the VAX, the **G400** job control file is 'G400.COM'. For the initial run of **G400**, the command line is:

@G400 Input control file name (G4001ST.DAT?), FIRST

The string 'FIRST' indicates that this is the initial run, and causes all of the **G400** scratch files to be deleted and reopened. For subsequent **G400** runs the command line is:

@G400 Input control file name (G4002ND.DAT?), SECOND

Using any string except 'FIRST' retains the **G400** scratch files.

The procedures for executing **G400** using NOS/VE are described at the end of the report.

Rigid Blade Response Program (GRP)

The rigid blade response program (Ref. 6) computes rotor performance and airloads. It provides information to the **F389SR** analysis on the required control angles and rigid blade flapping. The coupling with **F389SR** is very similar to the coupling of **G400** with **F389SR**. Program **GRP** is initially run with no variable inflow, and later restarted using variable inflow data from **F389SR**. This program is not well documented, and only the information required to set up and run a case for this application will be given here.

The following files are used during a **GRP** run as part of a rotor/fuselage analysis:

Unit 5.	GRPCON	Input control file (Formatted - 80 char/line)
Unit 6.	GRPLIST	Output listing file (Formatted - 80 char/line)
Unit 14.	VINFLOW	Input inflow velocity file from **F389SR** (Formatted - 80 char/line)
Unit 16.	GFCOUP	Output coupling parameter file for **F389SR** (Formatted - 80 char/line)
Unit 17.	GRPU17	Scratch file
Unit 18.	GRPU18	Scratch file
Unit 21.	GRPU21	Scratch file
Unit 22.	GRPU22	Scratch file
Unit 23.	LINEARA	Output airfoil data file for **F389SR** (Formatted - 132 char/line)
Unit 24.	GRPU24	Scratch file
Unit 25.	GRPU25	Scratch file

On the Perkin-Elmer, the input control file (unit 5) is a concatenation of several files that contain header titles, airfoil data, and control variables:

HEADER.DTA - Header list information

NACA0012.DTA - Airfoil data

GRPLOD.DTA - Control and condition information

CONSTANT.DTA - Constant information

On the VAX, a single input control file contains all of the above sections.

On either system, the input values for rotor lift and drag are set from prompts by the input pre-processor (**FWINPT**). The values of flight speed, tip speed, rotor radius, sound speed, number of blades, and shaft angle are set automatically by **FWINPT** to match the values in **F389SR** and **WABAT**. All other values in the **GRP** input control file must be set using the editor. hand.

The output files for **F389SR** are described in the section on **F389SR**. Program **GRP** has been modified for this application to write loader format control data for **F389SR** to unit 16, rather than control data to both units 16 and 17. This change to subroutine 'LOADF' improves the logic and legibility of the coupling between **GRP** and **F389SR**.

The restart run of **GRP** uses a different control file on unit 5 (OPTIONV). This file has only two variables:

Location 117 - LAML = 1 - Load in variable inflow

Location 118 - UVL = 2 - Use variable inflow

All other input data is read in from a scratch file written during the initial **GRP** run. The scratch files must therefore be saved between runs of **GRP**.

On the Perkin-Elmer, there are two job control files for running **GRP**. The first,'GRP1ST.CSS', allocates the output and scratch files (units 6, 14, 16, 17, 18, 21, 22, 23, 24, and 25), assigns them, and starts **GRP**. The second job control file, 'GRP2ND.DTA', does not allocate any files, and assigns the substitute control file, 'OPTIONV.DTA,' instead of 'TEMP.DTA'.

On the VAX, the job control file is 'GRP.COM'. The command line to run **GRP** for the first time is:

@GRP Input control file (GRP1ST.DAT?), FIRST

For further **GRP** runs, the command line is:

@GRP Input control file (OPTIONV.DAT?), SECOND

Program **GRP** has not been installed on NOS/VE.

Global Convergence Program (CONVRG)

The global convergence of the rotor/fuselage analysis is determined from the rotor and wake induced velocities at the fuselage panels. Program **CONVRG** compares the induced velocity file written by the latest iteration of **F389SR**, 'PANVEL,' with a file written during the previous iteration of **F389SR**. This file was renamed as 'PANOLD'. The mean velocity and the azimuthal harmonic amplitudes at each panel are compared, and the largest difference in either the mean velocity or the harmonic amplitude is used as the convergence parameter. If the maximum change between iterations is less than the pre-selected limit, the solution is judged to be converged. If not, another global iteration is performed.

Program **CONVRG** uses four input and output files:

Unit 4 - CONLIM	Input/output convergence limits and history (Formatted - 80 char/line)
Unit 6 - CONLIST	Output listing file (Formatted - 132 char/line)
Unit 7 - PANVEL	Input rotor induced velocity file #1
Unit 8 - PANOLD	Input rotor induced velocity file #2

The input convergence limit file (unit 4) is originally written by the input preprocessor, **FWINPT**, and is rewritten by **CONVRG** after each global iteration. The file is one line long:

Line 1	NUMBIT (* Format)	Number of last iteration
	MAXIT	Maximum number of iterations allowed
	DVLIM	Maximum velocity change for convergence, (velocity / flight velocity)
	DVLAST	Maximum velocity change at the last iteration

Variables MAXIT and DVLIM do not change; the others are reset by **CONVRG.**

The output listing file (unit 6) describes the convergence history of the rotor/fuselage analysis.

The input induced velocity files (units 7 and 8) are described in the section on **WABAT**, and are unchanged by **CONVRG**. The file names should be renamed so that the previous iteration is saved as 'PANOLD' before 'PANVEL' is re-allocated by the next iteration of **F389SR**.

The convergence information is indicated by the end-of-task (EOT) code of **CONVRG**. An EOT code of 0 means the solution has converged. An EOT code of 1 means that the solution has not converged, but the maximum number of iteration has not yet been exceeded. An EOT code of 2 means that the solution has not converged, and the maximum number of allowed iterations has been exceeded. An EOT code of 3 means that there has been an error in one of the data files. The job control program for the global iteration must take appropriate action based upon the EOT code returned by **CONVRG**. Another option is to just use CONVRG to determine write out the convergence, and to always run the rotor-fuselage analysis for 5 to 8 iterations, since the later global iterations add only slightly to the cost of the complete solution. This avoids having to pass EOT information to the job control program. This simpler approach is used for the VAX/VMS and NOS/VE versions.

For VAX/VMS, the control line for **CONVRG** is:

```
@CONVRG runnum
```

The parameter 'runnum' is the four character string that identifies the run.

For NOS/VE, execution of **CONVRG** is described at the end of this report.

PROGRAMS TO CREATE INPUT FILES

There are several forms of input required for the programs used in this analysis. The input for each of the major programs (**F389SR**, **WABAT**, etc.) have been described in the sections devoted to each individual program. This section deals with three programs that are used to generate input data files: **WABGEN**, **WCROSS**, and **FWINPT**. The first, **WABGEN**, generates the fuselage panel files based upon input cross section descriptions. The second, **WCROSS**, creates a simplified representation of the fuselage that is used in displacing the rotor wake. The final program, **FWINPT**, is used to create most of the additional input data and job control files that are required. **FWINPT** must be run for each individual case, while **WABGEN** and **WCROSS** are only run when a new fuselage geometry is required.

Fuselage Panel Generation (WABGEN)

The fuselage panel generation program, **WABGEN**, is a slightly modified version of the Automated Paneling Technique (APT) described in Ref. 3. In brief, the fuselage cross section at each longitudinal station is defined using a series of straight or curved segments. Rectangular and triangular panels are generated with linear dimensions that are between the specified minimum and maximum lengths. The output consists of a file containing the coordinates of the corners of each panel. This output file may be plotted using program **WABPLOT**, and provides the input geometry for the **WABAT** analysis. A listing file is also generated by **WABGEN** to document the process and describe any errors encountered. The following is a sample of an interactive **WAGBEN** session:

```
*WABGEN
THIS IS THE FUSELAGE PANEL GENERATING PROGRAM FOR WABAT
ENTER THE FILE NAME FOR THE INPUT CROSS SECTION DATA (THE "APT" FILE)
 XXXXXXXXXXXXXXXXXXXXXXXX
>APTFILE.DAT
NEXT ENTER THE OUTPUT FUSELAGE PANEL FILE NAME
 XXXXXXXXXXXXXXXXXXXXXXXX
>PANFILE.DAT
FINALLY ENTER THE OUTPUT LISTING FILE NAME
 XXXXXXXXXXXXXXXXXXXXXXXX
>APTPAN.LIS

           HOLD IT. I'M THINKING
**END OF INPUT DATA - FILE HAS BEEN REWOUND
USE PROGRAM WABPLOT TO PLOT THE PANELS GENERATED HERE
STOP
```

The input 'APT' data file uses logical unit 14, the output panel file is created on unit 16, and the listing file is on unit 17. Units 5 and 6 are used for the interactive terminal. A detailed description of the input 'APT' file is given in Ref. 3.

Wake Displacement Body Generation (WCROSS)

When used in the coupled rotor/fuselage mode, the rotor inflow program, **F389SR**, must displace the wake filaments about the fuselage. The fuselage geometry is passed to **F389SR** by an input data file called the wake displacement body file. The name of this file is typically of the form 'WDPxxx.DTA'. The wake displacement body defines the fuselage using a series of cross section planes at specified stations (typically 8 to 15) along the longitudinal axis. Radii are defined at equal angular increments (typically 15 to 20 deg) at each cross section. This format allows **F389SR** to efficiently determine whether each wake filament would pass inside or outside of the fuselage.

An interactive graphics program, **WCROSS**, was written to take a standard **WABAT** fuselage panel file (as created by **WABGEN**) and generate the wake displacement body file. The program is begun by selecting the longitudinal stations (coordinate z in Fig. 2) where cross sections will be defined. All **WABAT** panel corners within a specified distance (delta z) of the station are displayed on the screen. A first approximation to these points is made using either a best-fit ellipse or a Fourier series. The approximation may be improved by using keyboard commands to increase or decrease the radius at each angle. When a satisfactory approximation has been produced, the data is stored and the program moves on to the next longitudinal station. The wake displacement file is written at the conclusion of the process. As an option, a panel file that illustrates the geometry of the wake displacement body may also be created. This panel file can be plotted by **WABPLOT** to verify that an accurate representation has been obtained. The **WCROSS** program may also be started using an existing wake displacement body file if only a small number of changes are required. The wake displacement body file needs to be changed only when the fuselage geometry is changed. The same file is used for all rotor configurations and flow conditions.

A sample terminal session using **WCROSS** will now be described. The program is started and the panel file created by **WABGEN** is selected:

```
*WCROSS
THIS IS THE WAKE DISPLACEMENT BODY GENERATOR
ENTER THE FILENAME FOR THE INPUT BODY PANELS
THIS INPUT FILE IS GENERATED BY WABGEN
 XXXXXXXXXXXXXXXXXXXXXX
>S76PAN.DTA
```

The program now asks whether to begin with an existing displacement body:

```
START WITH EXISTING DISPLACEMENT BODY? (Y/N)
>Y
```

A 'Y' response prompts for a the name of an existing wake displacement body file:

```
ENTER FILE NAME FOR INPUT DISPLACEMENT BODY
 XXXXXXXXXXXXXXXXXXXXXX
>WDPS76.DTA
```

A 'N' response prompts for the number of longitudinal stations and radii to be used for the new wake displacement body:

```
NOW SET UP A NEW WAKE DISPLACEMENT BODY...
FIRST ENTER THE # OF CROSS SECTIONS (Z DIRECTION),
AND THE ANGULAR INCREMENT TO DEFINE LOCAL RADIUS (X/Y PLANE)
TYPICALLY ABOUT 15 STATIONS AND 18 DEG
>11, 18
```

For either option, the longitudinal stations (z coordinate, Fig. 2) are now selected or verified:

```
CHECK AND/OR CHANGE THE Z STATIONS FOR DEFINING THE CROSS SECTIONS
(BODY NOSE IS AT Z =     19.0, TAIL AT Z =     450.0)
STATION    1    Z = 19.0,     DELTA Z =  1.0
STATION    2    Z = 35.0,     DELTA Z =  1.0
...
STATION   11    Z =450.0,     DELTA Z =  1.0
ENTER  -1 TO START OVER AND GET ALL NEW Z S
        0 IF CURRENT VALUES ARE CORRECT
        N TO ONLY CHANGE VALUE OF Z(N)
>2
```

The response of '2' means that only the second z value needs to be changed. A response of '0' means that all z values are correct, and a response of '-1' means that no z values are correct, and all must be re-entered. The values of delta z are listed for reference only, they can only be changed later in the session. The program next prompts for the correct value of z(2):

```
Z(2) =  35.0  ENTER NEW VALUE
>30
STATION    1    Z =  19.0       DELTA Z = 1.0
STATION    2    Z =  30.0       DELTA Z = 1.0
...
STATION   11    Z = 450.0       DELTA Z = 1.0
ENTER  -1 TO START OVER AND GET ALL NEW Z S
        0 IF CURRENT VALUES ARE CORRECT
        N TO ONLY CHANGE VALUE OF Z(N)
>0
```

The program now begins to march through all of the z stations to set up the appropriate radii:

```
NOW SETTING UP RADII FOR STATION     1 AT  Z=  19.0
SEARCHING FOR ALL BODY PANEL CORNERS WITHIN DELTA Z =   1.0
THE PROGRAM HAS FOUND ONLY ONE POINT IN THIS RANGE:
X=         0.0,   Y=     67.5,  Z=    19.0
```

```
THIS MAY BE CORRECT IF THIS IS THE BODY NOSE OR TAIL
IS THIS OK? (Y/N)
>Y
```

At the first and last radial stations the fuselage should be closed, so that only one point will be found. If this message is typed at any other station, the response should be 'N', and the increment, DELTA Z, increased. In this case the 'Y' answer means that the program will go on to the next radial station.

```
NOW SETTING UP RADII FOR STATION     2 AT Z=   30.0
SEARCHING FOR ALL BODY PANEL CORNERS WITHIN DELTA Z =  1.0
CLEAR THE SCREEN AND TYPE <CR>
```

Since at least two panel corners were found between z=29 and z=31, a screen plot of their locations is now generated, as shown in Fig. 3. The 'x' marks indicate the corners, and the solid line is the current representation of the radii in the wake displacement body file. (If a new file is being started, there will initially be no solid line.) The following options are now available:

```
ENTER: 0-QUIT, 1-NEW DEL Z, 2-FIT ELLIPSE, 3-FOURIER FIT,
       4-MANUAL FIT, 5-REFRESH SCREEN, 6-NEXT Z
>2
```

A response of '0' keeps the radii at the current values and skips all further longitudinal stations. A response of '1' generates a prompt for a new value of DELTA Z and then redraws the cross section. A response of '2' computes a best fit ellipse (in the least-squares sense), and draws this curve (as shown in Fig. 3). A response of '3' generates a prompt for the number of terms in the Fourier series representation of the input data, and then draws the resulting curve. The Fourier series fit has generally been found to not give very good results, and its use is not recommended. The manual fit (response of '4') will be described below. A response of '5' clears and redraws the screen. A response of '6' stores the current results for this longitudinal station and moves on to the next station.

The panels at station 2 of this example are approximated quite well by the ellipse, and do not require manual adjustment. At a later station the fuselage has a more complex cross section, and an elliptical fit is unsatisfactory, as shown in Fig. 4a. A manual fit (response '4') is required. Figures 4b and C show the screen display during a manual fit. The solid lines represent the radius vectors, and the dotted lines the body surface. The radius vector is adjusted by responding to the following prompt:

```
THETA= 54.0- K=+1%, J=-1%, L=+10%, H=-10%, R=REFRESH, F=FORWARD, B=BACKUP
>F
```

THETA=54.0 indicates the particular angle (with respect to the negative y axis) of the radius now being changed. Responses 'K' through 'H' change this radius by the appropriate percentage of its current value. The radius vector is redrawn, with an 'x' at its tip. Response 'R' clears and redraws the screen. Response 'F' stores the value of the radius at the current value of THETA and moves on to change the radius at the next value of THETA. Response 'B' moves back to change the radius at the previous value of THETA. Since the panel input data file from **WABGEN** indicates that this particular fuselage is symmetric about the y axis, a symmetric wake displacement body will also be generated. Therefore, once the radius at THETA=90.0 has been defined, a matching half-body is created for the rest of the cross section, as shown in Fig. 4d, followed by this prompt:

```
ENTER:  0-QUIT, 1-NEW DEL Z, 2-FIT ELLIPSE, 3-FOURIER FIT,
        4-MANUAL FIT, 5-REFRESH SCREEN, 6-NEXT Z
>6
```

Response '6' causes the program to move on to the next longitudinal station. After a response of '0' to this prompt, or after all longitudinal stations have been set up, the output files are selected:

```
WRITE OUT A NEW WAKE DISPLACEMENT FILE? (Y/N)
>Y
```

If the session began with an existing wake displacement file, the next prompt is:

```
USE THE EXISTING FILE: WDPS76.DTA ? (Y/N)
>N
ENTER THE FILE NAME TO WRITE TO:
 XXXXXXXXXXXXXXXXXXXXXXX
>WDPS76B.DTA
```

A 'Y' response would have overwritten the existing file (WDPS76.DTA). The next prompt is:

```
WRITE OUT A PANEL FILE FOR WABPLOT? (Y/N)
>Y
ENTER THE FILE NAME TO WRITE TO:
 XXXXXXXXXXXXXXXXXXXXXXX
>WCPS76.DTA

STOP
```

This response writes a panel representation of the wake displacement body to a new file, 'WCPS76.DTA'. This file may then be plotted using **WABPLOT**, as shown in Fig. 5. The simplification of this fuselage model is obvious upon comparison with the original body shown in Fig. 2.

The input panel file for program **WCROSS** is on logical unit 16, the output file is on unit 7, the panel representation of the wake displacement body is on unit 8, and the interactive terminal is on units 5 and 6.

Input Preprocessing Program (FWINPT)

The input and control files used by the component programs of the rotor/fuselage analysis are created by an interactive program, **FWINPT**. This program provides the opportunity to change most (but not all) of the parameters that describe the physical conditions for each case and the options used in the numerical analysis. The first stage of this process is to start **FWINPT** and select the name of the file that contains a set of names of the input, output, and control files:

```
*FWINPT
THIS IS FWINPT - CREATES ROTOR/FUSELAGE INPUT & CONTROL FILES
ENTER THE NAME FOR THE INPUT FILE LIST
XXXXXXXXXXXXXXXXXXXXXX
>COPFIL.DTA
```

This file contains the name and description of the 25 input, control, and output files that may be required:

CONTROL.CSS	CONTROL FILE
WABTCON1.DTA	STARTING WABAT FILE
WABTCON2.DTA	MIDDLE WABAT FILE
WABTCON3.DTA	FINAL WABAT FILE
RFCOXXXX.DTA	ROTOR/FUSELAGE COUPLING FILE
TUFTXXXX.DTA	TUFT OUTPUT FILE
ROTPXXXX.DTA	ROTOR PANEL OUTPUT FILE
WPLTXXXX.DTA	WABAT PLOTTING OUTPUT FILE
PANEL.DTA	FUSELAGE PANEL FILE
HARM1.DTA	INITIAL VELOCITIES AT ROTOR
HARM2.DTA	REPEAT VELOCITIES AT ROTOR
F389CON1.DTA	FIRST F389 FILE
F389CON2.DTA	SECOND F389 FILE
F389CON3.DTA	THIRD F389 FILE
F389CON4.DTA	FOURTH F389 FILE
FILPXXXX.DTA	WAKE FILAMENT OUTPUT FILE
FPLTXXXX.DTA	F389 PLOTTING OUTPUT FILE
CNVRXXXX.DTA	GLOBAL CONVERGENCE FILE
WDPFILE.DTA	WAKE DISPLACEMENT BODY FILE
F389CONF.DTA	F389 FILAMENT ONLY FILE
GRPLOD.DTA	LOADER INPUT FOR GRP
WABTCON4.DTA	WABAT FILE FOR GRP/G400 RUN ONLY
G4001ST.DTA	FIRST G400 FILE
G4002ND.DTA	SECOND G400 FILE
WABAT.JOB	BATCH JOB CONTROL FILE

The names of these files may be changed during the **FWINPT** session. The 'XXXX' represents a 4 character string that identifies the files for a particular run. Which of the 25 files are used depends upon the case being run. Further details on the use and contents of these files are given below.

The next prompt from **FWINPT** asks whether the name for the rotor/fuselage coupling file will be changed from the original value:

```
READ ROT/FUS COUPLING FILE FROM   RFCOXXXX.DTA   OK? (Y/N)
>N
ENTER FILE TO USE INSTEAD
 XXXXXXXXXXXXXXXXX
 RFCORUN1.DTA
READ ROT/FUS COUPLING FILE FROM   RFCORUN1.DTA  OK? (Y/N)
>Y
-- 9/86 - TEST ELLIPSE FOR F389/WABAT/G400 COUPLING --
```

This file is used by **WABAT** and **F389SR**, and contains such parameters as the number of time steps, the position of the rotor relative to the fuselage, the tip path plane attitude, and the fuselage pitch and yaw angles. The (optional) title string is typed out to identify the type of fuselage.

The 4 character run identification string is entered next. This string is primarily used to distinguish the output files and prevent overwriting.

```
ENTER 4 CHARACTERS TO BE USED TO IDENTIFY THIS RUN
>RUN2
THIS RUN WILL BE IDENTIFIED BY RUN2,  OK?
>Y
```

Changes may now be made in several of the parameters in the rotor coupling file:

```
ROTOR HUB IN WABAT COORDINATES IS AT X,Y,Z= 0.0,  157.0, 200.0
ROTOR RADIUS IS  264.0  OK?
>Y
SHAFT ANGLE (FORWARD TILT NEGATIVE) IS -5.0
CONING ANGLE IS 4.50   OK?
>Y
```

The rotor hub location and radius are entered in whatever coordinates have been used to define the **WABAT** fuselage panel file. The shaft tilt is the angle of the shaft relative to the longitudinal axis of the fuselage. The coning angle will be reset if a blade motion analysis program (**G400** or **GRP**) is included in the solution process. The other parameters in the rotor coupling file will be changed automatically later in the program to correspond to values in the other files.

The next prompt in the **FWINPT** session concerns reading **WABAT** input data from a previous run:

```
READ 1ST WABAT CONTROL FILE FROM  WABTCON1.DTA  OK?
>Y
```

This file is a modified version of the **WABAT** input file described in Ref. 3. Several indices were added for this coupled mode of operation and to optionally reduce the amount of printout that is generated. These new indices are set automatically by **FWINPT**. The program now prompts for changes to some of the other **WABAT** input parameters:

```
REFERENCE AREA, LENGTH FOR BODY ARE 218956.0, 450.0
POINT TO CALCULATE MOMENTS ABOUT, (X,Y,Z)=  0.0,  157.0,  200.0  OK?
>Y
BODY PITCH, YAW ANGLES (DEG) ARE  3.88,  0.0  OK?
>Y
THERE ARE 0 PRESSURE WATERLINES
THERE ARE 0 BODY LINES AT 0.0 :
 OK?
>Y
```

The body reference area and length are used by **WABAT** to compute the nondimensional aerodynamic forces and moments on the fuselage, and are in the **WABAT** panel coordinate system. The point to compute the moments about is also in this coordinate system. In this case this point is the rotor hub. The body pitch and yaw angles are with respect to the freestream (flight) velocity vector. The pitch angle is automatically reset during a **G400** or **GRP** coupled run. The pressure waterlines and body lines are used by **WABAT** to control the printout (Ref. 3), and are not usually used during an coupled run.

The next prompt from **FWINPT** asks for the name of a previous **F389SR** input data file:

```
READ 1ST F389 INPUT FILE FROM F389CON1.DTA    OK?
>Y
```

This file is in the same ('loader') format as the **F389SR** input files described in Ref. 4, but several additional parameters have been added for fuselage coupled operation. These will be described below. The azimuthal and radial increments are now verified:

```
AZIMUTH INCREMENTS FOR F389 CIRCULATION, FIELD SOLUTIONS ARE 15.0, 7.5
ARE THESE OK?
>Y
THE 9 RADIAL INFLOW STATIONS FOR F389 ARE
0.1750 0.3250 0.4750 0.6250 0.7500 0.8500 0.9250 0.9650 0.9900
ARE THESE OK?
>Y
THE 15 RADIAL STATIONS FOR WABAT/GRP ARE:
0.0737 0.1186 0.2126 0.3150 0.4150 0.5150 0.6150 0.7150 0.8075 0.8600
0.8925 0.9225 0.9450 0.9650 0.9925
ARE THESE OK?
>Y
```

The azimuth increments (in deg.) define the time resolution of the solution. The first value is used by **F389SR** to compute the rotor bound circulation distribution. In most cases an azimuth increment of 15 deg is sufficient. The second value is used by **F389SR** to compute the velocities induced at the fuselage and by **VABAT** to compute the fuselage pressures and the velocities induced at the rotor. This value should generally be not greater than 7.5 deg. The minimum value allowed by this version of F389 is 2 deg. / Number of Blades. Smaller values provide higher time resolution, especially at larger advance ratios, but greatly increase the run-time of the analysis and the temporary storage requirements. This increase is proportional to the inverse square of the azimuth increment.

The first set of radial stations (in r/R) are used for the **F389SR** solution. The nine stations given above generally give acceptable results. A maximum of 15 stations may be used. The second set of radial stations is used by **VABAT** to compute the fuselage induced velocities at the rotor disc, and by **GRP** to perform the blade motion analysis. It has a maximum value of 25. **G400** uses a separate set of radial stations which are not changed by **FVINPT**.

The next stage of the **FVINPT** session deals with the blade motion analyses, **GRP** or **G400**:

```
LINK WITH GRP?
>N
LINK WITH G400?
>N
```

These two options cannot be selected together. If the **GRP** option is selected, or if no blade motion option is selected, **FVINPT** now prompts for a check of the rotor geometry and flight speed:

```
THIS IS A 4.0 BLADED ROTOR WITH DIAMETER 25.0 FT -?  (Y/N)
>Y
FLIGHT SPEED= 62.2KTS, TIP SPEED=700.0FPS => MU=0.1500 --? (Y/N)
>N
ENTER CORRECT V (KTS), OMEGA R (FPS)
>124.5, 700.
FLIGHT SPEED=124.5KTS, TIP SPEED=700.0FPS => MU=0.3000 --?  (Y/N)
>Y
```

If a **GRP** coupled run was selected, **FVINPT** now prompts for the **GRP** loader input file:

```
READ GRP LOADER FILE FROM   GRPLOD.DTA   OK?
>Y
THE 15 RADIAL STATIONS FOR WABAT/GRP ARE:
0.0737 0.1185 0.2126 0.3150 0.4150 0.5150 0.6150 0.7150 0.8075 0.8600
0.8925 0.9225 0.9450 0.9650 0.9925

ROTOR LIFT = 18000.0 LBS,  DRAG = 300 LBS - ? (Y/N)
>Y
```

The radial stations typed at this point have been read in from the **GRP** input file. They replace the values read in from the **F389SR** input file (as were displayed earlier by **FWINPT**). The rotor lift and drag are verified next. The entire array of **GRP** input parameters are now typed out:

```
NON ZERO VALUES OF LOADER INPUT FOR GRP:
5   1 0.70000E+03 0.25000E+02 0.11164E+04 0.00000E+00 0.40000E+01
5   6 0.62230E+02 0.46580E+02 0.36120E-01 0.47800E-01 0.10950E+00
...
5 796 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.20000E+01
THESE VALUES MAY BE CHANGED BY EDITING GRPLOD.DTA
REWRITE THE GRP LOADER FILE?
>Y
```

The format of this list (and of the **GRP**, **G400**, and **F389SR** input data files) has as the first item in each line the number of data array elements in the line, as the second item the starting array element number, and finally the values of the (up to 5) array elements. With the exception of the items specifically prompted for by **FWINPT**, all changes to the **GRP** loader array must be made by editing the **GRP** loader file. The **GRP** loader parameters are given in Ref. 7. If the rewrite option is selected, **FWINPT** will automatically change the following **GRP** loader parameters to maintain compatibility with **F389SR** and **WABAT**: tip speed, rotor radius, sound speed, number of blades, flight velocity, and shaft angle.

If a **G400** coupled run was selected, **FWINPT** prompts for the **G400** input file and displays the **G400** input data arrays:

```
READ 1ST G400 LOADER DATA FROM   G4001ST.DTA ?  (Y/N)
>Y
NONZERO VALUES OF A VECTOR ARE
5   1 0.19200E-02 0.11164E+04 0.97000E+00 0.00000E+00 0.30000E+01
...
5 366-0.25679E-02 0.52340E-02-0.23519E-03-0.12313E-02-0.12463E-02
NONZERO VALUES OF R VECTOR ARE
5   1 0.30000E-01 0.31800E-01 0.22000E-01 0.96400E-01 0.90200E-01
...
NONZERO VALUES OF G VECTOR ARE
5   1 0.70000E+03 0.25000E+02 0.00000E+00 0.46580E-01 0.15000E+02
...
NONZERO VALUES OF S VECTOR ARE
5   1 0.10000E+01 0.00000E+02 0.00000E+00 0.00000E+00 0.00000E+00
...
FLIGHT SPEED =  62.13 KTS - TIP SPEED = 700.0 FPS => MU = 0.149822
ROTOR LIFT = 10000.0 LBS - PROPULSIVE FORCE = 250.0 LBS
THE 15 RADIAL STATIONS FOR WABAT/G400 ARE:
0.0616 0.0925 0.1194 0.1786 0.2719 0.3620 0.4519 0.5289 0.5930 0.6696
0.7491 0.8109 0.8913 0.9493 0.9850
AZIMUTH INCREMENT FOR CIRC SOLUTION = 15.0
CHANGE THESE VALUES BY EDITING  G4001ST.DTA
IS THIS THE CORRECT G400 FILE TO USE?  (Y/N)
>Y
```

The **G400** input data files are described in detail in Ref. 5. In brief, they consist of a series of arrays (identified by 'A', 'R', etc.) that have a format similar to the **F389SR** and **GRP** loader arrays. There are two **G400** input files used during a coupled rotor/fuselage analysis: G4001ST.DTA for the initial run, and G4002ND.DTA for a restart during the global iteration. G4001ST.DTA contains most of the parameters that are changed between runs. **FVINPT** does not change any of these parameters or rewrite either of the files. Because of the complexity of **G400**, the files may only be changed by directly editing the files using Ref. 5 as a guide. Several parameters used by **F389SR** and **WABAT** are, however, changed by **FVINPT** to conform to the values read in from the **G400** file: flight velocity, tip speed, sound velocity, azimuth increment for the **F389SR** circulation solution, rotor radius, number of blades, and the number and location of the blade radial stations used for computing the induced velocities at the rotor (by **WABAT** and by **G400**). These parameters are typed out by **FVINPT** for verification. If these parameters are incorrect, a response of 'N' to the final question above allows a different **G400** input file to be read in.

The **F389SR** input data is reviewed during the next part of an **FVINPT** session. The data was read in from the 1st **F389SR** input file, as selected previously.

```
THE CURRENT NONZERO VALUES OF F389 LOADER INPUT ARE:
5   1 0.62130E+02 0.70000E+03 0.11164E+04 0.15000E+02 0.25000E+02
...
5 386 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.20000E+01
5 391 0.00000E+01 0.30000E+01 0.10000E+00 0.50000E-01 0.00000E+00
5 396-0.75000E+02 0.10000E+01 0.15000E+00 0.00000E+00 0.00000E+00
ENTER (N, VALUE) TO CHANGE LOCATION N TO VALUE,
ENTER (0,0) IF ALL ARE OK,  ENTER (-N,0) TO TYPE LOCATION N,
ENTER (-999,0) TO TYPE EVERYTHING
>-392,0
LOCATION 392 VALUE=3.0
ENTER (N, VALUE) TO CHANGE LOCATION N TO VALUE,
ENTER (0,0) IF ALL ARE OK, ENTER (-N,0) TO TYPE LOCATION N,
ENTER (-999,0) TO TYPE EVERYTHING
>0,0
```

As shown, any individual **F389SR** data item may be verified or modified using **FVINPT**. The meaning of each item is printed during each **F389SR** run, and the values used in the original, uncoupled version of **F389SR** are described in Ref. 4. The following additions, changes, and comments apply to the modified version of **F389SR** used for rotor/fuselage coupling:

LOC 88 - XLINK - Option to couple with **G400**. This value must only be set equal to 0 for no **G400** coupling or 1 for **G400** coupling. A value of 2 will cause the **F389SR** program to fail.

LOC 206 - W1OPT - Option for W1 input. This must be set to 1 for G400 coupling.

LOC 210 - PUNCH - 'Card Punch' option for induced velocities. This must be set to -1 for **G400** coupling ('punch' harmonics of induced axial velocities)

(Note that **F389SR** locations 88, 218, 223, 187, 206, and 210 are automatically set by **FWINPT** to the proper values for **GRP** or **G400** coupled runs.)

LOC 390 - WCOUPL - Option to write a file containing induced velocities at the fuselage panels. This is the main rotor fuselage coupling switch. A value of 0 means no file is written, a value of 1 writes out velocities at each time step (azimuth value), and a value of 2 writes out velocities in a Fourier series in azimuth. The present coupling with **WABAT** requires a value of 2 at this location.

LOC 391 - FILPLT - Option to write a plot file for rotor data. This file (FPLTXXXX.DTA, logical unit 27) contains blade circulation, angle of attack, rotor induced velocities, and fuselage induced velocities, on the rotor disc at each azimuthal and radial station. This file may be read by **PLOTWAB** and then displayed as a two dimensional, contour, or carpet plot. A value of 0 disables this option, and a value of 1 causes the file to be written.

LOC 392 - FILDO - Option for displacing the wake about the fuselage. This option controls the wake displacement. A value of 0 causes **F389SR** to ignore the fuselage as far as the wake geometry is concerned. A value of 1 causes **F389SR** to ignore the fuselage, but to write an output plotting file describing the wake geometry (FIPLXXXX.DTA, logical unit 26). This file can be read in by program **WABPLOT**. A value of 2 causes **F389SR** to displace the wake about the fuselage (as described by the wake displacement file - WDPFILE.DTA, logical unit 24) and to also write out the plotting file. A value of 3 causes **F389SR** to displace the wake, but not to write the plotting file. A value of 4 causes **F389SR** to displace the wake and to write the plotting file, but to cease execution before any circulations or velocities are computed. A value of 5 causes **F389SR** to not displace the wake, to write the plotting file, and then to cease execution. Options 4 and 5 are used in conjunction with the input control file F389CONF.DTA to quickly produce a plotting file of the wake.

LOC 393 - DELTAF - Wake filament offset distance. This value is used to set the minimum distance that the wake filaments are offset from the fuselage. All filaments that would intersect or approach the fuselage will be pushed away from the surface by this distance.. This option is only used if LOC 392, FILDO, is equal to 2, 3, or 4. The displacement distance is expressed in terms of a fraction of the fuselage radius at the longitudinal location of the filament. The standard value is 0.10 (10% of the local body radius). This is also the default value, and is automatically used by **F389SR** if the input DELTAF is set less than 0.

LOC 394 - PANSIZ - Minimum filament length for resegmenting. This value determines the minimum length of wake filament that will be resegmented. It is only used if LOC 392, FILDO, is equal to 2, 3, or 4. It is expressed in units of the rotor radius. All filaments with segments of length greater than this value that are displaced around the fuselage will be divided up into 10 sub-segments for a smoother approximation of the wake geometry. This value should be set approximately equal to the minimum fuselage panel size. A typical value is 0.10 (10% of the rotor radius). This is also the default value, and is automatically used by **F389SR** if the input PANSIZ is set less than 0.

LOC 395 - CIRSTR - Option to change the wake circulation strength for filaments that are displaced about the fuselage. It is used only if LOC 392, FILDO, is equal to 2, 3, or 4. A value of 0 causes **F389SR** to keep the wake circulation constant for all filaments displaced about the fuselage. A value of +1 causes **F389SR** to increase the circulation associated with filaments that are displaced about the fuselage, proportional to the increase in length of each segment caused by filament stretching. A value of +2 causes **F389SR** to decrease the circulation associated with such filaments, inversely proportional to the increase in segment length. A value of -1 causes the circulation associated with such filaments to be set to zero. This option allows the effects of filament stretching and dissipation to be crudely modeled. The standard option is to keep the circulation constant, CIRSTR = 0.

LOC 396 - THETIL - The angle at which the filament shifts from passing above the fuselage to passing below the fuselage. It is used only if LOC 392, FILDO, is equal to 2, 3, or 4. The angle is measured from a horizontal line passing through the center of the fuselage cross section at the longitudinal location of each filament, and is expressed in degrees. A positive angle is above this line, a negative angle below it. Wake filaments that would intersect the fuselage surface at an angle greater than THETIL are displaced above the fuselage, those that would intersect at an angle below THETIL are displaced below the fuselage. This option models the splitting of the wake filament after it is stretched around the fuselage. The standard value is THETIL = -75 deg. This implies that the filaments are stretched around 85% of the fuselage before the snap over to the passing below the fuselage.

LOC 397 - FARNER - Near fuselage / far fuselage wake option. This option allows the influence of various components of the wake to be removed from the solution. It is only used if LOC 392, FILDO, is equal to 2, 3, or 4. A value of 0 causes **F389SR** to use the entire wake structure and the blade bound circulation in computing the solution. This is the standard value. A value of -1 causes all wake segments that are far from the fuselage (defined as those that pass outside of the smallest rectangular box that can enclose the entire fuselage) to be neglected. A value of +1 causes all wake wake segments that are near to the fuselage (i.e. those within this box) to be neglected. A value of +2 causes the bound circulation associated with the rotor blades to be neglected. A value of +3 causes the entire wake to be neglected, and the solution computed using only the bound circulation. This option is primarily used to study the contribution of various wake regions, and it should be set to a value of 0 for any realistic case.

LOC 398 - RCLOSE - Close vortex / fuselage panel interaction distance. This parameter sets the maximum distance between a wake filament and a fuselage panel at which the close interaction model is applied. The velocity induced at the fuselage panel by a wake filament that is at a distance greater than RCLOSE is evaluated only at the centroid of the panel. The velocity induced by a wake filament that is nearer than RCLOSE is computed at five points spaced across the panel. This model is required to reduce the numerical vortex / panel resonance effects. The value of RCLOSE is expressed in terms of the rotor radius, and is typically set to 0.15 (15% of the rotor radius). A value of RCLOSE less than 0 causes the close interaction model to never be used.

Once all of the **F389SR** input values have been set properly, the next stage of the **FWINPT** session is to rewrite the five separate **F389SR** input files:

```
REWRITE F389 FILES?
>Y
```

If the response to this prompt is 'N' the **F389SR** input files will not be updated and will not reflect any changes made during this **FWINPT** session.

```
WRITE  F389CON1.DTA   FOR THE FIRST F389 FILE?
>Y
```

This file is used to control **F389SR** during computation of the initial circulation solution. The geometric influence coefficients for the rotor on rotor velocities are computed and stored in temporary disc files. The values of the input data array used in this file are identical to those used during the **FWINPT** session.

```
VORTEX CORE RADII: CIRC SOLN TIP= 0.0060  INBD= 0.0120
                   FIELD TIP    = 0.0100  INBD= 0.2000
ARE THESE OK?
>N
ENTER NEW FIELD TIP, INBD RADII (R/ROTOR R)
>0.015, 0.2000
VORTEX CORE RADII: CIRC SOLN TIP= 0.0060  INBD= 0.0120
                   FIELD TIP    = 0.0150  INBD= 0.2000
ARE THESE OK?
>Y
```

F389SR defines two vortex filament core radii, one for the tip vortex (location 202 - RCORE), and one for the filaments that represent the inboard sheet (location 203 - RCOREI). For the **F389SR** circulation solution, the tip core radius is typically set to 10% of the blade chord (about .6% of the rotor radius for this sample case), and the inboard core radius is set to a somewhat larger value, since these filament are meant to represent a continuous sheet rather than a discrete vortex. Different values of the radii should be used to compute the velocities induced at the fuselage (the 'field solution'). The tip vortex core radius is typically doubled to represent vortex diffusion, while the inboard core radius is greatly increased to avoid creating spurious peaks in the fuselage pressures. This prompt allows each of these values to be verified or changed.

The file names for the other F389SR input files are now verified, and the files written out:

```
WRITE  F389CON2.DTA   FOR THE 2ND F389 FILE?
>Y
WRITE  F389CON3.DTA   FOR THE 3RD F389 FILE?
>Y
WRITE  F389CON4.DTA   FOR THE 4TH F389 FILE?
>Y
WRITE  F389CONF.DTA   FOR THE F389 FILAMENT FILE?
>Y
```

The second **F389SR** file is used to control **F389SR** during the initial field computation of the velocities at each fuselage panel. This step computes and stores the geometric influence coefficients for the rotor on fuselage velocities. **FWINPT** automatically makes several changes to the values stored in the first **F389SR** file ('F389CON1.DTA') to create the second **F389SR** file ('F389CON2.DTA'): The tip and inboard vortex core radii (locations 202 and 203) are changed, as described above, the azimuth increment (location 4) is changed to the value selected for the field solution, and solution mode option (location 224) is changed to a value of 1 to indicate field point induced velocity mode.

The third **F389SR** file is used to control **F389SR** during repeat computations of the rotor circulation solution. The geometric influence coefficients are retrieved from the disc files. **FWINPT** automatically changes the value of the value of the geometric influence coefficient save option (location 294) to 1 to select this mode.

The fourth **F389SR** file is used to control **F389SR** during repeat computations of the induced velocities at the rotor panels. **FWINPT** changes the geometric influence coefficient save option (location 294) to 1 to cause **F389SR** to retrieve these coefficients from the disc files. The other values in this input file are identical to those in the second **F389SR** file.

The **F389SR** filament file is used to control **F389SR** when it is run to only write out a wake filament plotting file. Location 392 is set automatically to a value of 4 to perform this function, as described above.

Additional information is added after the loader data arrays in the first and third **F389SR** input files. This information is a line by line image of the airfoil data lines that were originally read in from the first **F389SR** file. The format of this airfoil data is given in Ref. 4. This airfoil data is ignored when **F389SR** is being run coupled to **GRP** or **G400**.

The next stage of the **FWINPT** session is to verify the fuselage panel file, and read from it the number of fuselage panels:

```
BODY PANEL FILE IS   PANEL.DAT      OK?
>Y
```

The **WABAT** input control files are verified next:

```
REWRITE WABAT FILES?
>Y
```

A response of 'N' prevents updating of the **WABAT** input files.

```
WRITE 1ST WABAT FILE TO   WABTCON1.DTA    OK?
>Y
PREVIOUS TITLE WAS
-- 9/86 - TEST ELLIPSE FOR F389/G400/WABAT COUPLING --
OK?
>Y
```

The first **WABAT** file is used for the initial solution for steady flow with no rotor influence. The geometric influence coefficients for the fuselage on fuselage velocities are computed and stored on disc. **FWINPT** automatically sets the proper control parameters for this function, and sets the reference quantities and flow angles to the values selected during this **FWINPT** session.

The remaining **WABAT** input files are verified and written next:

```
WRITE 2ND WABAT FILE TO   WABTCON2.DTA    OK?
>Y
WRITE 3RD WABAT FILE TO   WABTCON3.DTA    OK?
>Y
WRITE 4TH WABAT FILE TO   WABTCON4.DTA    OK?
>Y
```

The second **WABAT** file is used for the intermediate solutions. The influence coefficients are read from disc, a complete unsteady solution is found, including rotor induced velocities, but the fuselage pressures and integrated loads are not computed.

The third **WABAT** file is used for the final solution. A complete unsteady solution is found, and the final fuselage pressures and airloads are computed and written out.

The fourth **WABAT** file is only used when a **GRP** or **G400** coupled run is made. This file causes **WABAT** to use previously stored geometric influence coefficients, but to compute only a steady solution with no rotor influence.

The next prompt verifies the wake displacement file name:

```
WAKE DISPLACEMENT BODY FILE IS   WDPFILE.DTA   OK?
>Y
```

The rotor fuselage coupling file name is verified, and the file is updated:

```
WRITE ROT/FUS COUPLING FILE TO RFCORUN2.DTA   OK?
>Y
```

The file name is automatically set to 'RFCOxxxx.DTA', with xxxx set to the 4 character run identification string set at the beginning of the **FWINPT** session.

FWINPT next opens the previous global convergence file ('CNVRxxxx.DTA'), and verifies the limits:

```
THE CONVERGENCE LIMIT ON THE ROTOR INDUCED VELOCITY IS  0.00050
THE MAXIMUM NUMBER OF GLOBAL ITERATIONS IS 8
ARE THESE OK?
>Y
```

The convergence file is used by program **CONVRG** to set the criterion for a globally converged solution and to set the maximum permissible number of global iterations **(F389/WABAT/GRP-G400)**. The convergence limit refers to the maximum allowed change in the mean rotor induced velocity at any fuselage panel or in the amplitude of any harmonic of this velocity, expressed as a fraction of the flight velocity. The solution is terminated after the maximum number of iterations, regardless of convergence. A new convergence file is written out next, with name 'CNVRxxxx.DTA', where the 'xxxx' refers to the run identification string.

The input and output file list is given next:

```
INPUT/OUTPUT FILES WILL BE:
1 CONTROL.CSS      CONTROL FILE
2 WABTCON1.DTA     STARTING WABAT FILE
...
25WABAT.JOB        BATCH JOB CONTROL FILE
ENTER FILE # TO CHANGE, ENTER 0 IF ALL OK
>0
```

It is not a generally good idea to change these files at this point in **FWINPT**. The file names should be changed individually at the appropriate point in the session.

The job control file is created now. The syntax of this file is heavily dependent on the particular computer system used. No user input is required or allowed during an **FWINPT** session. Modification to the **FWINPT** program to generate job control language for a different computer system should follow the general guidelines given here, and should be based upon the detailed commands used in the **FWINPT** program. There are three different variations of the job control file, corresponding to running just **F389SR** and **WABAT**, running **F389SR**, **WABAT**, and **G400**, and running **F389SR**, **WABAT**, and **GRP**. An **FWINPT** session produces the following for a PERKIN ELMER computer:

```
THE JOB CONTROL FILE WILL BE WRITTEN AS:
SET LOG; XAL LIST:RLOGRUN2.LST,IN,80
XAL SCRH:FIELDGC,DTA,IN,256
...
* 1ST WABAT RUN - STEADY, NO ROTOR, GUESS AT POSIT
WABAT WABTCON1.DTA,    PANEL.DTA,    RUN2
* 1ST G400/F389 LOOP: INITIAL G400 W NO F389 INFLOW
G4001ST G4001ST.DTA,  RUN2
F389R   F389CON1.DTA,   WDPFILE.DTA,  RUN2
...
* FINAL WABAT RUN AFTER CONVERGENCE
WABAT2  WABTCON3.DTA, PANEL.DTA,    RUN2
...
$EXIT
IS THIS OK?
>Y
```

A response of 'N' to this prompt merely types out the job control file again; no changes can be made at this stage. The filename for this file is now verified:

```
REWRITE JOB CONTROL FILE  CONTROL.CSS  ? (Y /DIFFERENT FILE / NO)
>Y
```

A response of 'Y' rewrites the existing file. A response of 'N' prompts for a new file name. A response of 'N' does not update any job control file.

A batch job file is often required. This file is also computer system dependent. The filename is verified now, and the file written out:

```
REWRITE BATCH FILE  WABAT.JOB   ? (Y / DIFFERENT / NO)
>Y
```

The final prompt from **FWINPT** asks whether the the list of input and output file names should be rewritten:

```
REWRITE LIST OF FILE NAMES ON COPFIL.DTA  ? (Y / DIFFERENT / NO)
>Y

STOP
```

This concludes the **FWINPT** session. Executing the job control file either directly or through the batch job file will begin execution of the run.

PROGRAMS TO EXAMINE THE OUTPUT

The output of the rotor-fuselage analysis consists of a printout generated by each analysis program (**F389SR**, **WABAT**, **G400**, **AND GRP**), and a series of plotting files generated by **F389SR** and **WABAT**. There are three programs that plot this output: **WABPLOT**, **PLOTWAB**, and **CVFPLT**. **WABPLOT** generates a view of the three-dimensional fuselage, rotor, wake filaments, and fuselage surface velocity vectors. **PLOTWAB** produces plots of the time histories of the fuselage airloads, and of the azimuthal and radial distributions of the bound circulation, the induced velocities, and the blade angle of attack at the rotor. **CVFPLT** produces plots of the time histories of the pressures and the induced velocities at individual fuselage panels, and of the pressures or velocities along a fuselage body line.

Fuselage, Rotor, and Wake Plotting (WABPLOT)

WABPLOT is an expanded version of the fuselage plotting package written for the **WABAT** program, as described in Ref. 3. The program has been modified to add plots of the rotor and the wake filaments, and to draw plots at each time step of the unsteady analysis. The package is designed to use subroutine calls to the Tektronix IGL graphics library and to produce plots on Tektronix 4010 compatible devices. The graphics calls have been segregated from the remainder of the program, so that it is relatively easy to substitute another graphics system (such as DI3000) for the Tektronix IGL. The use of the program is now illustrated for a typical terminal session. The Tektronix IGL version uses a single terminal for both interactive commands and graphics display.

```
*WABPLOT

WELCOME TO WABPLOT - WHAT DO YOU WANT TO DO?
ENTER 1 TO GET NEW INPUT FILE NAMES
ENTER 2 TO KEEP USING THE SAME NAMES
ENTER 0 TO QUIT PROGRAM
>1
```

A response of '1' must be used upon starting the program, or when a new case is to be plotted.

```
ENTER FILENAME FOR THE WABAT FUSELAGE PANELS
 XXXXXXXXXXXXXXXXXXXXX
>ELL06P.DTA
```

The fuselage panel file name refers to the panel file generated by WABGEN and used by **WABAT**. The file 'ELL06P.DTA' represents an ellipsoid with a particular panel spacing, and will be used during this example. This file is opened by **WABPLOT** on logical unit 16. An error in this file will cause the message ' ** NO PANEL DATA ON UNIT 16 **' to be generated.

ENTER 4 CHARACTER RUN ID STRING
XXXX
>RUN2

The run identification string corresponds to that selected in **FWINPT**, and serves to specify the following files:

The fuselage velocity tuft file, "TUFTxxxx.DTA". This file is written by **VABAT**, and contains three components of the surface velocity at each fuselage panel at each time step of the unsteady solution. **VABPLOT** opens this file on logical unit 16. If the file does not exist, the message 'TUFT FILE TUFTxxxx.DTA DOES NOT EXIST' is typed and no file is opened. Attempting to plot velocity tufts with no open file will cause an error.

The rotor panel file, "ROTPxxxx.DTA". This file is also written by **VABAT**, and contains a panel representation of the rotor blades. The rotor panel geometry does not necessarily depict the actual blade geometry, since it is always rectangular, and has an arbitrary chord. The rotor radius, shaft angle, coning, and flapping are correct, and the panels represent the radial stations used by the **VABAT/GRP** portion of the analysis. The file will be opened on logical unit 18. If this file does not exist, the message 'ROTOR PANEL FILE ROTPxxxx.DTA DOES NOT EXIST' is typed and no file is opened. Attempting to plot the rotor with no open file will cause an error.

The wake filament file, "FILPxxxx.DTA". This file is written by **F389SR**, and contains the coordinates of the endpoints of each segment of each wake filament used in the rotor inflow analysis. The wake structure is given for each time step of the analysis. The file is frequently quite long. The file will be opened on logical unit 19. If this file does not exist, the message 'WAKE FILAMENT FILE FILPxxxx.DTA DOES NOT EXIST' is typed, and no file is opened. Attempting to plot wake filaments with no open file will cause an error.

The rotor/fuselage coupling file, "RFCOxxxx.DTA". This file is originally written by **FWINPT**, but may be rewritten by **F389SR** when the rotor geometry has been changed by a blade motion program (**G400** or **GRP**). This file contains the blade coning and flapping angles, and the relative position of the rotor and fuselage (and other parameters not of interest here). This file will be opened on logical unit 20. If this file does not exist, the message 'ROTOR/FUSELAGE COUPLING FILE RFCOxxxx.DTA DOES NOT EXIST' is typed, and no file is opened. Attempting to plot anything but the fuselage with no open coupling file will cause an error.

In this example, the wake filament file was not found. Therefore, all components except the wake (fuselage, rotor, and velocity tufts) can be plotted without error.

The next prompt from **WABPLOT** refers to how the fuselage will be drawn:

```
DRAW BOTH SIDES, ADVANCING ONLY (MIRROR IMAGE)
     OR RETREATING ONLY? (B/A/R)
>B
```

The standard response to this prompt is 'B', to plot both sides. The half body plots (advancing or retreating blade side) are primarily used to provide a direct comparison of the results on each side.

```
REMOVE HIDDEN LINES? (Y/N/BODY ONLY)
>B
```

WABPLOT has a primitive hidden line capability for the fuselage and for the wake filaments. Fuselage panels are considered to be hidden when the outward normal vector from the panel is directed away from the viewer. Wake filament segments are considered to be hidden when the vector from the rotor hub to the segment is pointed away from the viewer. A response of 'Y' removes both hidden fuselage panels and hidden wake filaments. A response of 'N' plots all panels and all segments. A response of 'B' removes all hidden fuselage panels, but plots all wake segments.

```
DRAW VELOCITY TUFT VECTORS? (Y/N)
>Y
THERE ARE TUFT DATA FOR  12 TIME STEPS
STARTING AT AZIMUTH= 0.000 DEG, IN INCREMENTS OF  7.500 DEG
```

A response of 'Y' will cause the velocity tuft file to be read in and an arrow to be drawn on each fuselage panel representing the direction and magnitude of the velocity at the surface. A single horizontal arrow in the upper left corner of the plot indicates the freestream (flight) velocity. The first line of a tuft file generated by the version of **WABAT** used here is 'TUFT'. Files beginning with anything else are assumed to have been created by the original version of **WABAT**, and hence to contain data at only one time step.

```
DRAW THE ROTOR? (Y/N)
>Y
```

A response of 'Y' will cause the rotor panel file to be read in and the rotor to be drawn. The first line of this file must be 'ROTOR'. If not, the error message 'ERROR IN WOPADO - ROTOR FILE STARTS WITH xxxxx' is typed, and no rotor is drawn. If the number of blades and the number of time steps in the rotor file do not match the values read from the tuft file, an error message, 'ERROR IN WOPADO - ROTOR & TUFT FILES ARE INCONSISTENT' is typed.

```
DRAW WAKE FILAMENTS? (Y/N)
>Y
```

A response of 'Y' will cause the wake file to be read in, and the filaments to be drawn. The first line of this file must be 'WAKE'. If not, the error message 'ERROR IN WOPADO - WAKE FILAMENT FILE BEGINS WITH xxxxx' is typed, and no wake is drawn. If the number of time steps in the wake file do not match those in the rotor file, the error message 'ERROR IN WOPADO - WAKE TIME STEPS (xxx) DONT MATCH' is typed.

```
WHICH FILAMENTS? ALL 10, TIP, HUB, OR SINGLE (A/T/H/S)
>A
```

The response to this prompt determines which of the radial wake filaments will be drawn. A response of 'A' draws all filaments. A response of 'T' draws only the tip (outboard) filament. A response of 'H' draws only the hub (inboard) filament. A response of 'S' generates a prompt asking which single filament is to be drawn:

```
WHICH RADIAL STATION? (1->10)
>6
```

This response will cause only the filament from the sixth radial station to be drawn.

```
FROM WHICH BLADES?  (0 FOR ALL, N FOR BLADE N ONLY)
>0
```

The response to this prompt determines whether the wake filaments from all blades will be drawn or whether the filament from a single blade will be drawn. A response of '0' draws filaments from all blades. A response of any number 'n' will draw only the filaments from blade number n. Blade 1 is usually the blade that is over the fuselage tail at 0 azimuth.

```
ENTER AZIMUTH ANGLE TO USE (0.000, 7.500, ..., 82.500)
>15.0
```

This prompt determines which rotor azimuth position will be displayed. At zero azimuth the first blade is pointed downstream. The azimuth that is displayed is rounded to the nearest time step:

```
THIS IS TIME STEP  3
```

The azimuth is reset if the requested value is out of range. For example, a response of '187.5' will generate an error message 'AZIMUTH OUT OF RANGE - RESET TO 7.5 DEG'.

Once these options have been selected, the tuft, rotor, and wake filament data will be read in for the appropriate time step. If the maximum velocity at any tuft at this time step is greater than five times the freestream velocity, an error message is typed, and the tuft length should be scaled down:

```
MAX TUFT LENGTH = 15.6 ENTER LENGTH TO DIVIDE TUFTS BY
>4.0
```

WABPLOT may require a long period of time to read in the wake filament files. This is happening while the following message is visible:

```
NOW READING IN THE WAKE FILAMENT FILE
```

At this point, **WABPLOT** prompts for changes in the viewing orientation and scale. A complete description of the available options has been given in Ref. 3. Only a brief description of the most common options is presented here.

```
        PRESENT POSITIONS ARE..
  1..YAW ANGLE....      30.0            2..ROLL ANGLE...      0.0
  3..PITCH ANGLE..       0.0            4..SIGHT PLANE..  10000.0
  5..SCALE FACTOR.       1.0            6..LONG OFFSET..      0.0
  7..B.L. OFFSET        0.00            8..ELEV ANGLE...      0.0
  ENTER INDEX NO. & VALUE IF A CHANGE IS DESIRED (I1,G12.5 FORMAT)
  ENTER 0 TO CONTINUE
>5,1.2
>0
```

The YAW ANGLE should be set to 0 for a front view, 90. for a retreating blade side view, 180. to a tail view, and 270. for an advancing side view. The ROLL ANGLE should be set to 0. for a side view, 90. for a top view, and -90. for a bottom view. The PITCH ANGLE should be set to 0. to show a horizontal fuselage. A SCALE FACTOR of 1 will generally produce a view of the complete fuselage and rotor, with most of the wake visible. Increasing the SCALE FACTOR zooms in on a portion of the plot.

```
        PRESENT VIEW CENTER IS AT
  1...HORIZONTAL        0.0    AND 2...VERTICAL       0.0
  HORIZONTAL SHIFT RIGHT IS +.   VERTICAL SHIFT UP IS +.
  ENTER INDEX NO. & NEW VALUE  (I1,G12.5 FORMAT)
  ENTER 0 TO CONTINUE
>0
```

The VIEW CENTER is typically used together with the SCALE FACTOR to perform the zoom function. Otherwise it is not used.

```
        PRESENT OBJECT BOUNDARIES ARE...
  1..STATION MIN....   -119.2235            2..STATION MAX.... 630.5149
  3..BUTT LINE MIN..   -318.4338            4..BUTT LINE MAX.. 318.4338
  5..WATER LINE MIN.    -27.6889            6..WATER LINE MAX. 213.1932
  ENTER INDEX NO. & VALUE IF CHANGE IS DESIRED (I1,G12.5 FORMAT)
  ENTER 0 TO CONTINUE
  >0
```

The OBJECT BOUNDARIES establish the viewing window (in WABAT fuselage coordinates). Anything outside these boundaries is ignored. If only the fuselage is to be drawn, the boundaries include only it. If a rotor or wake is also to be drawn, the boundaries are automatically expanded. The STATION MAX and MIN refer to the longitudinal coordinate (z), the BUTT LINE MAX and MIN to the horizontal coordinate (x), and the WATER LINE MAX and MIN to the vertical coordinate (y). The only change that normally need to be made for this application is to increase the STATION MAX and decrease the WATER LINE MIN to capture the complete wake structure.

Following this response, the graphics screen in cleared, and the plot is generated, as shown in Fig. 6. The symbol in the lower right corner represents the number of blades and the current azimuthal position. WABPLOT now waits for a <CR> to clear the screen and prompt for the next plot:

```
ENTER 0 TO RETURN, 1 TO CHANGE VIEW ANGLE,
2 TO CHANGE TIME,  3 TO CHANGE THIS PLOT,  4 TO CHANGE MORE
>0
```

A response of '0' is used to return to the main menu and plot a new case or to exit the program. A response of '1' causes a prompt to change the PRESENT POSITION and the VIEW CENTER, and then redraws the current plot. A response of '2' causes a prompt for a new azimuth angle, and then redraws the current plot. A response of '3' returns to the 'DRAW BOTH SIDES?' prompt, and allows all of the previously set plotting parameters to be changed. A response of '4' causes prompts for additional options:

```
RESET SIZE?
>N
```

A response of 'Y' to this prompt is similar to the 'CHANGE VIEW ANGLE' option above, but adds the ability to reset the OBJECT BOUNDARIES. A response of 'N' moves on to:

```
EXPLODED VIEW?
>N
```

A response of 'Y' will generate additional prompts concerning the option to separate out the various fuselage components (nacelle, pylon, etc.). This option is not normally used for rotor/fuselage plots. If both responses are 'N', **WABPLOT** reinitializes all viewing angles and boundaries to the default settings and then generates the 'DRAW BOTH SIDES?' prompt.

Fuselage Airloads and Rotor Inflow Plotting (PLOTWAB)

PLOTWAB is used to plot two forms of output files, one of which is created by **WABAT**, and the other by **F389SR**. The operation of **PLOTWAB** and the various type of plots it can create will now be described. The actual plotting is done through subroutine 'QPLOT', which has a Tektronix IGL and a Precision Visuals DI3000 version. The description here is for the DI3000 version, which uses separate terminals for the graphics and for the interactive commands.

```
*PLOTWAB
ENTER NAME OF FILE TO PLOT:
 XXXXXXXXXXXXXXXXXXXXXX
>WPLTRUN2.DTA
```

Plotting files written by **WABAT** have the form 'WPLTxxxx.DTA'. Files written by **F389SR** have the form 'FPLTxxxx.DTA'. The file is opened on logical unit 14. Units 15 and 16 are used for the interactive terminal commands. A session for a **WABAT** file has fewer options and will be discussed first.

```
RESCALE THE DATA?
>Y
ENTER HOW TO RESCALE:
    1 TO CONVERT FORCES & MOMENTS TO F/THRUST, M/LT
    2 TO CONVERT F&M TO F/QS, M/QSL
    3 TO CONVERT F&M TO LB, FT-LB
    4 TO MULTIPLY BY SPECIFIED CONSTANTS
>1
```

The **WABAT** file contains the time histories of the aerodynamic forces and moments on the fuselage in coefficient form. The three forces (lift, drag, and side force) have been normalized by dividing by the dynamic pressure based upon the freestream velocity and by the reference area that was specified during the **FWINPT** session. The three moments have been normalized by additionally dividing by the reference length specified in **FWINPT**, and are about the moment center defined in **FWINPT**. A response of 'N' to the rescale option will leave the results in these original terms.

There are four RESCALE options. The first, and most commonly used, is '1'. This option renormalizes the forces by dividing by the rotor thrust, and renormalizes the moments by dividing by the rotor thrust and the reference length. For straight and level flight the force coefficients are therefore approximately equivalent to accelerations in g's. This option requires that the rotor thrust coefficient and advance ratio be entered. The advance ratio was specified in **FWINPT**, and the thrust coefficient is approximately computed by **F389SR**, (it is more accurately computed by **GRP** or **G400**). In either case, the thrust coefficient must be read from the printout. This option assumes that the reference area specified in **FWINPT** was the rotor disc area. The prompt for this option is:

```
ENTER ADVANCE RATIO, THRUST COEFFICIENT
>0.15, 0.007
```

The second RESCALE option allows the reference area, length, and dynamic pressure to be changed. The prompt is:

```
ENTER NEW,OLD REFERENCE AREAS, LENGTHS, QS
>218956,218956, 450,1, 1,1
```

The third RESCALE option restores the forces and moments to dimensional units, by multiplying by the quantities used to nondimensionalize:

```
ENTER REFERENCE AREA, LENGTH, VELOCITY IN FT, SEC
>1620, 37.5, 200
```

The fourth RESCALE option provides for an arbitrary multiplication factor for each of the forces and moments:

```
1 - Lift, CL                  ENTER MULT FACTOR
>0.5
2 - Drag, CD                  ENTER MULT FACTOR
>0.9
...
```

The minima and maxima of the data are now computed, and a prompt issued for the selection of which variables to plot:

```
THERE ARE  6 VARIABLES, ENTER # TO PLOT, WHICH
      1  Lift, CL
         MEAN=  -.0380  MIN=  -0.0476  MAX=  -0.0324
      2  Drag, CD
         MEAN=  0.0011  MIN=  -0.0056  MAX=   0.0081
      ...
      6  Rolling Moment, Cl
         MEAN= -0.0044  MIN=  -0.0063  MAX=  -0.0025
>3, 1,2,3
```

The first item in the response list is the number of curves to put on the plot, the next items are the list of curve numbers to plot.

The labels to be used with the plot are selected now:

```
NEW X-Y AXIS LABELS?
>Y

X AXIS:
 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
>Blade Azimuth, deg
Y AXIS:
 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
>Force / Thrust
```

```
ADD SIDE LABELS TO THE PLOT? (Y/N/SAME/DEFAULT)
>Y
ENTER LABEL OR STOP
>Run 2
ENTER LABEL OR STOP
>Sample Case
ENTER LABEL OR STOP
>STOP
```

The SIDE LABELS are written at a selected location on the plot. A response to the prompt of 'Y' allows arbitrary labels to be entered. A response of 'N' will generate no labels. A response of 'S' will reuse any previously selected labels. A response of 'D' will create a three line default label indicating the thrust coefficient, advance ratio, and the date the plot was made.

```
DRAW MEAN VALUES ON PLOT?
>Y
```

A response of 'Y' will draw lines along the left edge of the plot indicating the mean value of each curve.

```
DATA LIMITS: XMIN=0.0, XMAX=360.0, YMIN=-.035, YMAX=0.05
ENTER PLOTTING LIMITS: (XMIN,XMAX,YMIN,YMAX)
>0 360 -.06 .04
```

The minimum and maximum values for the data for all of the selected curves are computed and the values to use for the plot are selected.

Subroutine QPLOT is now called to draw the plot; the following two prompts are from the DI3000 version of QPLOT:

```
ENTER X,Y FOR LABEL; X,Y FOR CURVE ID
>.2,.9,   .6,.9
```

This prompt asks for the locations to place the SIDE LABEL and the CURVE ID strings. The x,y locations are given in fractions of the full scale axis values. Therefore (0,0) is the lower left corner, and (1,1) is the upper right corner.

```
USE A NEW TITLE?
>Y
XXXXXXXXXXXXXXXXXXXXXXX ENTER TITLE XXXXXXXXXXXXXXXXXXXXXXXXXXXX
Test Case for Plotting Program
```

The TITLE is written across the top of the plot. A sample plot for a case similar to that in this example is shown in Fig. 7a. After the plot has been created, **PLOTWAB** prompts for any additions or changes:

```
ENTER: 0 TO REDRAW THIS PLOT
       1 TO ADD A CURVE FROM THIS SET
       2 TO ADD A DIFFERENT CURVE
       3 TO START A NEW PLOT
       4 TO QUIT THE PROGRAM
>0
```

A response of '0' will prompt for changes and redraw the current plot:

```
LINE TYPE? +N=LINE, 0=MARKS, -N=BOTH
>1
```

This prompt concerns changes to the type of line drawn for the first curve in the plot. A LINE TYPE of '1' is a solid line, other line types are different forms of dashed line. A LINE TYPE of '0' draws 'X' marks at each data point. A LINE TYPE of '-1' draws 'X' marks and a solid line. A response of '1' is the standard.

```
ARROW HEADS?
>N
```

A response of 'Y' will draw a series of arrow heads on the curves.

```
DIFFERENT INDEPENDENT VARIABLE?
>N
```

A response of 'Y' will prompt for reselection of the plotting variables. Otherwise **PLOTWAB** will prompt for any changes in the plot labels of scale factors, and the redraw the plot.

A response of '1' to the prompt immediately following the plot will add a single curve from the current data file to the plot:

```
WHICH CURVE # (1->6)
>4
```

This response will add the fourth curve (Pitching Moment in this case).

A response of '2' to the prompt that follows the plot will add a curve from a different data file to the current plot:

```
ADD AN IDENTICAL CURVE FROM ANOTHER FILE?
>Y
```

A response of 'Y' will add a curve from a different file that represents the same quantity as was last plotted:

```
ENTER NAME OF FILE TO PLOT:
 XXXXXXXXXXXXXXXXXXXXXX
>WPLTRUN1.DTA
RESCALE THE DATA?
```

```
>Y
...
```

A response of 'N' will also ask for a curve number to be specified for the additional plot.

A response of '3' to the prompt that follows the plot will start over to create an entirely new plot:

```
USE THE SAME FILE?
>N
ENTER NAME OF FILE TO PLOT:
...
```

A response of '4' will terminate this session of PLOTWAB.

The changes in the terminal session when an **F389SR** output file is read in to **PLOTWAB** will be discussed now.

```
*PLOTWAB
ENTER NAME OF FILE TO PLOT:
 XXXXXXXXXXXXXXXXXXXXXX
>FPLTRUN2.DTA
THE DEPENDENT VARIABLES ARE Azimuth AND Radius  ENTER
    1 TO PLOT VS Radius
    2 TO PLOT VS Azimuth
    3 TO MAKE A CONTOUR PLOT
    5 TO RESTART
>1
```

The data in the **F389SR** plotting file is a function of two independent variables, azimuth and radius. This causes **PLOTWAB** to offer three plot options. A response of '1' will lead to a two-dimensional plot versus Radius at a single azimuth:

```
WHICH Azimuth? - 0.0, 15.0, 30.0, 45.0, 60.0, 75.0, 90.0, 105.0, 120.0,
135.0, 150.0, 165.0, 180.0, 195.0, 210.0, 225.0, 240.0, 255.0, 270.0,
285.0, 300.0, 315.0, 330.0, 345.0, 360.0
>30
```

A similar prompt is issued after response '2', which leads to a two-dimensional plot versus azimuth at a single radius:

```
WHICH Radius? - 0.1750, 0.3250, 0.4750, 0.6250, 0.7500, 0.8500, 0.9250,
0.9650, 0.9900
>0.9250
```

In either case the remainder of the **PLOTWAB** session is similar to a session for plotting a **WABAT** file. There are eight possible plot variables:

```
1 Bound Circulation (ft2/sec)
2 Radial Velocity, Rotor Induced (ft/sec)
3 Tangential Velocity, Rotor Induced (ft/sec)
4 Axial Velocity, Rotor Induced (ft/sec)
5 Blade Angle of Attack (deg)
6 Radial Velocity, Body Induced (ft/sec)
7 Tangential Velocity, Body Induced (ft/sec)
8 Axial Velocity, Body Induced (ft/sec)
```

The rotor induced velocities are computed by **F389SR** as the direct result of the effects of the rotor and wake vorticity. The body induced velocities are computed by **WABAT** from the fuselage panel source strengths. Because of the interaction between the rotor and fuselage and the global iteration procedure, neither velocity component is a pure effect of either the rotor or fuselage. Figure 7b shows a typical **PLOTWAB** graph of **F389SR** data.

A response of '3' to the plot type prompt will create a contour plot of one of the eight dependent variables as a function of azimuth and radius. The dependent variable is selected first:

```
1   Bound Circulation
...
8   Axial Velocity       Body Induced
WHICH COMPONENT IS TO BE PLOTTED?
>1

 INPUT TITLE
>Bound Circulation for case RUN 1 - Test plot

 CLEAR PAGE, HIT RETURN
```

A contour plot is now drawn, as shown in Fig. 7c.

Fuselage Panel Time Histories and Bodyline Plotting (CVFPLT)

CVFPLT is used to plot the pressures and induced velocities on the fuselage. The data is read in from an input file that was written by **WABAT**. **CVFPLT** can produce two types of plots: time histories at an individual fuselage panel, and bodylines: the distribution of the pressure or velocity along a line on the fuselage surface. As an option, data from an additional input file can be added to the plots of the **WABAT** results. This may be used to compare computational and experimental data. **CVFPLT** uses the same plotting subroutine (QPLOT) and DI3000 or IGL calls as used by **PLOTWAB**.

```
*CVFPLT
ENTER RUN ID FOR THIS CASE:
 XXXX
>RUN1
CVFPLT NOW READING IN CVF DATA FILE
```

The CVF DATA FILE is named 'CVFPxxxx.DTA', where 'xxxx' refers to the run identification string selected in **FWINPT**. This is an unformatted file composed of several logical records. The first record contains header information and panel centroid locations, and each additional record contains the velocities, forces, and pressures at each panel for a given time step. The data file is opened on logical unit 4.

```
CVF FILE READ IN.   THERE ARE-
 12 TIME STEPS AT AZIMUTHS OF  0.00,  7.500,... 82.50
 WITH 428 PANELS AND 4  BLADES
 THE ADVANCE RATIO IS   0.150
 PLOT EXPERIMENTAL DATA IN ADDITION TO WABAT DATA? (Y/N)
 >Y
 ENTER FILENAME FOR EXPT DATA
  XXXXXXXXXXXXXXXXXXXXXX
 >TESTRUN1.DTA
```

The experimental data file has the same format as a **PLOTWAB** plotting file. The first line contains the number of dependent variables, a '0', and the length of the array of independent variables, in a (1X,3I5) format. The next line contains the independent variable array, in a (1X,I4,10E13) format. The independent variable is either the azimuth position (in deg) or the distance along a bodyline (in **WABAT** coordinates). For each dependent variable there is a line containing the number of the variable and its name, in a (1X,I4,1X,C40) format. This line is followed by the dependent variable array, in a (1X,I4,10E13) format.

```
MAIN MENU: ENTER-
0 TO QUIT PROGRAM
1 TO PLOT TIME HISTORY
2 TO PLOT ALONG BODY LINE
>1
```

```
ENTER WHAT DATA TO PLOT:
1 FOR CP
2 FOR NET VELOCITY (ROTOR INDUCED + BODY INDUCED)
3 FOR FORCE INCREMENT ON A PANEL
4 FOR VELOCITY INDUCED BY ROTOR ONLY
>2
```

A response of '1' selects the pressure coefficient, defined as surface pressure minus freestream static pressure, divided by the freestream dynamic pressure. A response of '2' selects the total velocity at the panel, divided by the freestream velocity. A response of '3' selects the force on the panel, divided by the freestream dynamic pressure. A response of '4' selects the only velocity induced by the rotor (as calculated by **F389SR**). If the velocity or force is to be plotted, the directional component must be specified:

```
ENTER WHICH DIRECTIONAL COMPONENT TO PLOT:
1 FOR MAGNITUDE
2 FOR X (ADVANCING BLADE SIDE = +)
3 FOR Y (VERTICAL = +)
4 FOR Z (DOWNSTREAM = +)
5 FOR ALL THREE COMPONENTS
>5
```

If an experimental data file was read in, the variable to plot must be selected:

```
PLOT WHICH EXPT CURVE # ? (1-2) - 0 TO PLOT NONE
1  PANEL 1
2  PANEL 17
>2
```

There are two ways to select which panel to plot data for: by panel number, or by searching for the panel closest to a specified position:

```
ENTER PANEL # TO PLOT (1-> 428) OR ENTER 0 TO SEARCH FOR (X,Y,Z)
>0
PANEL MIN/MAX:     -49.51 < X <     49.41  (+ADV SIDE)
                    51.13 < Y <    148.90  (+VERTICAL)
                    36.75 < Z <    463.21  (+DOWNSTREAM)
ENTER X,Y,Z, COORDINATES TO FIND CLOSEST PANEL TO -
>0 148 300
THE 3 CLOSEST PANELS ARE
 130 X=    6.877,  Y=     147.625,  Z=     301.838
 344 X=   -6.877,  Y=     147.625,  Z=     301.838
 119 X=    6.981,  Y=     148.464,  Z=     281.090
ENTER WHICH PANEL TO PLOT DATA FOR
>130
```

If 'CP' is to be plotted, a correction must be applied to account for the total pressure rise induced by the rotor. The output file from **VABAT** only accounts for the changes in velocity from the freestream value, not for the total pressure rise. The correction increases the total pressure in the region of the fuselage that lies in the wake of the rotor. This region may be defined in two ways: by specifying a wake skew angle and assuming that the total pressure jump is confined to a skewed cylinder emanating from the rotor, or by specifying two streamwise coordinates and assuming that the total pressure jump occurs only between these two values of z:

```
TOTAL PRESSURE CORRECTION - SKEW ANGLE OR Z START,FINISH? (S/Z)
>S
ENTER WAKE SKEW ANGLE (DEG), DELTA CP INCREMENT
>30., 1.2
```

The skew angle may be determined from plots of the wake geometry (made by **WABPLOT**), and the total pressure increment can be estimated (from actuator disc theory) as 2 times the thrust coefficient divided by the square of the advance ratio. If the 'Z' option had been chosen the prompt would have been:

```
ENTER Z START, FINISH FOR WAKE ON BODY (BODY UNITS), DELTA CP TO ADD
>120.  370.
```

At this stage in **CVFPLT**, the plotting subroutine, QPLOT, is called. The prompts from this routine depend on whether it is the Tektronix IGL or the DI3000 version. The DI3000 prompts for QPLOT are described in the section on **PLOTWAB**. A typical plot of the three components of the velocity at a panel is shown in Fig. 9a. Once the plot is complete the next prompt is:

```
ENTER - 0 TO GO BACK TO MAIN MENU
        1 TO REDRAW THIS PLOT
        2 TO DRAW A NEW TIME HISTORY
>0
```

A response of '1' allows the total pressure correction and QPLOT selections to be changed. A response of '2' allows any new time history plot to be made. A response of '0' returns to the main menu:

```
MAIN MENU: ENTER -
0 TO QUIT PROGRAM
1 TO PLOT TIME HISTORY
2 TO PLOT ALONG BODY LINE
>2
```

The BODYLINE option first asks for the selection of quantity to plot, directional component, and whether to add an experimental data curve. These questions are the same as for a time history plot. Bodylines are defined by selecting an angle from the body center and tolerance about this angle. All panels with centroids within this range will contribute to the plot. The dependent variable on the plot is the streamwise coordinate (z).

```
ENTER Y VALUE AT AXIS (0?), ANGLE OF BODY LINE (0=TOP, 90=ADV SIDE)
ANGULAR TOLERANCE (10 DEG?)
>0., 45., 5.
SEARCHING FOR ALL PANELS WITHIN 5.0 DEG
OF BODY ANGLE = 10.0 DEG ABOUT  (X,Y) = (0.,  0.)
J=   1 PHI=     1.44 (X,Y,Z)= (    2.82     112.31     36.75)
J=   2 PHI=     4.10 (X,Y,Z)= (    7.88     109.87     36.75)
...
J=  47 PHI=     7.93 (X,Y,Z)= (    6.86      52.40    301.83)
ARE THESE OK? (1=YES, 2=TRY AGAIN, 3=GIVE UP)
>1
```

A response of '1' means that the indicated panels are acceptable. A response of '2' allows the ANGLE OF BODY LINE and ANGULAR TOLERANCE to be changed. A response of '3' returns to the main menu.

Data can be plotted either for individual time steps or for the time mean:

```
PLOT HOW MANY TIME STEPS?
(1->  12 - OR 0 FOR MEAN)
>3
ENTER AZIMUTHS TO PLOT (DEG)
> 0, 30, 45
```

The total pressure correction is selected next, in the same manner as for a time history plot. A call to the QPLOT subroutine draws the plot, as shown in Fig. 9b for a typical bodyline. **CVFPLT** then returns to the main menu.

Printed Output

There is no new form of printout generated by the rotor/fuselage coupling program. The printout generated by the component programs (**WABAT**, **F389SR**, **GRP**, and **G400**) are maintained virtually unchanged. The **WABAT** printout is not generally useful in analysis of unsteady results, since it contains a very lengthy tabulation of many quantities at each time step for each global iteration. Some additional statements have been added to aid in debugging any problems that may be encountered. In general the three plotting programs, **WABPLOT, PLOTWAB**, and **CVFPLT** are sufficient to display the **WABAT** results.

The **F389SR** printout is described in Ref. 4. It has not been changed for this application. It is still quite useful in documenting the case and in displaying many parameters that are not directly plotted.

The **G400** and **GRP** printouts are also unchanged for this application. They remain the primary means of examining the blade motion, aeroelastic, and rotor performance results.

OPERATING USING THE LANGLEY CDC NOS AND NOS/VE SYSTEM

Execution of the rotor-fuselage analysis on the NASA Langley CDC system (Ref. 8) is complicated by the requirements to run the analysis using the virtual memory capability of NOS/VE on the 'A' machine (a CYBER 180-860) and to produce DI-3000 graphics using NOS on the 'D' machine (a CYBER 170-855). The analysis programs (**WABAT**, **F389SR**, **G400**, and **CONVRG**) and the input pre-processing program (**FWINPT**) are executed using NOS/VE on the 'A' machine, while the other programs (**WABGEN**, **WCROSS**, **WABPLOT**, **PLOTWAB**, and **CVFPLT**) are run using NOS on the 'D' machine.

The pre-processing program (**FWINPT**) is run on NOS/VE by moving to the $LOCAL catalog and executing a command procedure that transfers the input files from the permanent ($USER) catalog to the $LOCAL catalog, and running the compiled **FWINPT** object code ($USER.OFWINPT):

```
SETWC $LOCAL
$USER.RUNFWIN
```

FWINPT will now operate as described starting on page 46 of this report. The default input file names are contained in $USER.COPFIL, which is automatically copied from $USER to $LOCAL by the procedure RUNFWIN. Also copied over are the default input files used by **FWINPT**: WABTCN1, F389CN1, F389CN2, G4001ST, RFCOUPL, CNVPARM, and PANELS. Any additional input files used must be transferred to $LOCAL before executing **FWINPT**.

Upon completion of **FWINPT**, the complete set of input files may be copied back to the $USER catalog by executing another command procedure:

```
$USER.FWCOPY
```

The rotor-fuselage analysis is run in the batch mode by submitting the batch control procedure that was written by **FWINPT**:

```
SUBJ F=$USER.BATCHRF, SO=$USER.BATLIST
```

BATCHRF is the default control procedure name, and BATLIST is the output listing that describes the operation of the job. If an error occurs, the latest program output listings will be written to $USER.ERRWABT, $USER.ERRF389, $USER.ERRG400, and $USER.ERRCNVR. Note that the compiled object codes must exist on the permanent catalog: $USER.OWABAT, $USER.OF389SR, $USER.OCONVRG, AND $USER.OG400. They may be compiled by typing:

```
FTN I=WABAT,B=OWABAT,OPT=HIGH
FTN I=F389SR,B=OF389SR,OPT=HIGH
FTN I=G400688,B=OG400,OPT=HIGH
FTN I=CONVRG,B=OCONVRG,OPT=HIGH
```

If the batch job executes sucessfully, the final program output listings will be on $USER.WABLIST, $USER.F389LIS, and $USER.G400LIS. Program output listings from earlier phases of the job may be on $USER.WABLI1, $USER.F389L1C,

$USER.F389L2F, etc, as indicated by the control procedure (BATCHRF). The convergence history will be on $USER.CONLIST.

The default plotting output files are:

$USER.WABPLTF	(**WABAT** output for **PLOTWAB**)
$USER.F389PLT	(**F389SR** output for **PLOTWAB**)
$USER.CVFPLT	(**WABAT** output for **CVFPLT**)
$USER.PANELS	(Fuselage panel geometry for **WABPLOT**)
$USER.ROTOPAN	(Rotor panel geometry)
$USER.WAKE	(Wake geometry)
$USER.TUFTS	(Fuselage velocity vectors)
$USER.RFCOUPL	(Coupling Parameter file)

These files are transferred from NOS/VE to NOS permanent files on the 'A' machine by executing a command procedure:

```
$USER.NOSPLOT
```

In order to create NOS indirect access permanent files, files WABPLTF, F389PLT, etc must already exist on the 'A' machine. If not, direct access NOS files will be created.

To plot the results, log out of NOS/VE, log in to NOS on the 'A' machine, and transfer the plotting files to common storage (CPF), using command procedure MOVEPLT. (All NOS command procedures are contained in file PROCFIL, stored on CPF. Each is executed by typing -Procedure Name.)

```
FILESET,IA=LPF
GET,PROCFIL,ST=CPF
-MOVEPLT
```

Now log off of the 'A' machine, and switch to NOS on the 'D' machine. The plotting files are copied from CPF to local files using command procedure GETPLOT:

```
FILESET,IA=CPF
GET,PROCFIL
-GETPLOT
REWIND,*
```

The local NOS files on the 'D' machine have slightly different default names:

TUFTX, FILPX, ROTPX, and RFCOX for the **WABPLT** files - answer 'X' to the run number prompt, or rename by replacing the 'X' with a three character run identification string to save results from a case.

CVFPX for the CVFPLT file.

The output plotting programs are executed by calling the appropriate command procedure from PROCFIL:

-RWABPLT	to execute **WABPLOT**, or
-RPLTWAB	to execute **PLOTWAB**, or
-RCVFPLT	to execute **CVFPLT**.

To set up a new fuselage model, programs **WABGEN** and **WCROSS** are run using NOS on the 'D' machine. Again the appropriate command proceures are in PROCFIL, and all input files must be local:

-RWABGEN	to execute **WABGEN**, or
-RWCROSS	to execute **WCROSS**.

OPERATING F389SR TO COMPUTE MAIN-TAIL ROTOR INTERACTIONS

The rotor inflow program (**F389SR**) has been modified to allow computation of coupled main rotor - tail rotor interactions. The coupling is performed by computing the velocities that are induced by each rotor at the disc of the other rotor, and adding these velocities to the inflow that is used to compute the circulation solutions. Two assumptions are made in this first level analysis: the wake geometries of each rotor are not affected by the other rotor, and the rotors are operating at speeds which are integral multiples of each other. This technique is discussed in Appendix A of Ref. 1.

The following sections describe the coupling procedure, indicate the option controls that have been added to the F389SR input control file (unit 5), and give an example of the Job Control commands that are used with the VAX/VMS operating system.

The output of F389SR in this mode is identical to that described previously (page 31). In particular, the rotor bound circulation, the blade angles of attack, and the induced velocity components may be plotted using program PLOTWAB (page 65). Output listings are also generated (page 75 and Ref. 4).

Coupling Procedure.-The coupling of the main rotor and tail rotor is performed by using the existing features of the code with the modifications noted above to minimize the amount of input data the the user must provide. All coupling is done using externally defined data files and the appropriate system commands (job control language, JCL). The sequence of program execution is described below along with the key data that must be stored as input for subsequent executions for the code. The following abbreviations are used, MR for main rotor, TR for tail rotor, GC for geometric influence coefficients.

1) Calculate Isolated MR Circulation Solution
   Save MR Circulation Solution GC
   Save MR Circulation Solution

2) Calculate Induced Velocity Of MR on TR Field Points
   Read MR Circulation Solution
   Save MR on TR Field Point GC
   Save Harmonics of MR Induced Velocity At TR

3) Calculate Isolated TR Circulation Solution
   Save TR Circulation Solution GC

4) Calculate TR Circulation with MR Influence
   Read Harmonics of MR Induced Velocity At TR
   Read TR Circulation Solution GC
   Save TR Circulation Solution

5) Calculate Induced Velocity Of TR on MR Field Points
   Read TR Circulation Solution
   Save TR on MR Field Point GC
   Save Harmonics of TR Induced Velocity At MR

6) Calculate MR Circulation with TR Influence
   Read Harmonics of TR Induced Velocity At MR
   Read MR Circulation Solution GC
   Save MR Circulation Solution

7) Calculate Induced Velocity Of MR on TR Field Points
   Read MR Circulation Solution
   Read MR on TR Field Point GC
   Save Harmonics of MR Induced Velocity At TR

8) Calculate TR Circulation with MR Influence
   Read Harmonics of MR Induced Velocity At TR
   Read TR Circulation Solution GC
   Save TR Circulation Solution

9) Repeat Steps 5 to 8 until convergence is obtained

Input Options and Job Control

The specific input option flags required for this coupling procedure are locations 217,224,225,226,and 294 of the LOADER input data. A brief description of these inputs is provided.

Location	Name	Description
217	OPVIBC	option for input of additional velocities at the rotor blades
224	EPOPT	option to calculate field point velocities
225	TAILVI	option for calculation of induced velocities at tail
226	FPHARM	option for calculation of harmonics of induced velocities
294	GCSAVE	option to save and read wake GC's

An example of the JCL used on a Digital Equipment Corporation VMS operating system is shown below.

```
$!*****************************************************
$!    COUPLED MAIN ROTOR / TAIL ROTOR INTERACTION JCL  *
$!*****************************************************
$ INQUIRE TAG "CASE IDENTIFIER"
$!-------------
$!  FIRST PASS
$!-------------
$!
$!  create master output file
$!
$SET DEF [.DATA]
$create 'TAG'MAIN_tail.lst
$
$!
$! calculate isolated MAIN rotor circulation
$!
$DEFINE FOR002 'TAG'MAIN_CIRC_ISO.GCM
$DEFINE FOR005 [-]MAIN_CIRC_ISO.INP
```

```
$!
$!  217 = 0.0
$!  224 = 0.0
$!  225 = 0.0
$!  226 = 0.0
$!  294 = 1.0   save gc's
$!
$DEFINE FOR006 'TAG'MAIN_CIRC_ISO.LST
$DEFINE FOR007 'TAG'MAIN_CIRC_ISO.GAM
$DEFINE FOR018 'TAG'MAIN_CIRC_SOL.GCSV
$DEFINE FOR027 'TAG'MAIN_CIRC_ISO.PGAM
$DEFINE FOR028 'TAG'MAIN_CIRC_ISO.GCS
$DEFINE FOR030 'TAG'MAIN_CIRC_ISO.WAKE
$RUN [-]F389SR
$DEASSIGN FOR002
$DEASSIGN FOR005
$DEASSIGN FOR006
$DEASSIGN FOR007
$DEASSIGN FOR018
$DEASSIGN FOR027
$DEASSIGN FOR028
$DEASSIGN FOR030
$!
$DEL 'TAG'MAIN_CIRC_ISO.GCM;*
$DEL 'TAG'MAIN_CIRC_ISO.GCS;*
$DEL 'TAG'MAIN_CIRC_ISO.WAKE;*
$!
$!  append lst file
$!
$append 'TAG'MAIN_circ_iso.lst 'TAG'MAIN_tail.lst
$DEL 'TAG'MAIN_CIRC_ISO.LST;*
$!
$! calculate induced velocity at TAIL rotor due to MAIN rotor
$!
$DEFINE FOR003 'TAG'MAIN_TAIL_IV.HARM
$DEFINE FOR005 [-]MAIN_TAIL_FP1.INP
$!
$!  217 = 0.0
$!  224 = 1.0  calc induced velocities at field points
$!  225 = 2.0  calc at another rotor disk
$!  226 = 2.0  calc harmonics of induced velocities at field points
$!  294 = 1.0  save gc's
```

```
$!
$DEFINE FOR006 'TAG'MAIN_TAIL_FP.LST
$DEFINE FOR007 'TAG'MAIN_CIRC_ISO.GAM
$DEFINE FOR027 'TAG'MAIN_CIRC_ISO.PGAM
$DEFINE FOR029 'TAG'MAIN_TAIL_FP.GCSV
$define FOR030 'TAG'MAIN_TAIL_FP.WAKE
$RUN [-]F389SR
$DEASSIGN FOR003
$DEASSIGN FOR005
$DEASSIGN FOR006
$DEASSIGN FOR007
$DEASSIGN FOR027
$DEASSIGN FOR029
$DEASSIGN FOR030
$!
$DEL  'TAG'MAIN_CIRC_ISO.GAM;*
$DEL  'TAG'MAIN_TAIL_FP.WAKE;*
$!
$!  append lst file
$!
$append 'TAG'MAIN_tail_fp.lst 'TAG'MAIN_tail.lst
$DEL 'TAG'MAIN_TAIL_FP.LST;*
$!
$! calculate isolated TAIL rotor circulation
$!
$DEFINE FOR002 'TAG'TAIL_CIRC_ISO.GCM
$DEFINE FOR005 [-]TAIL_CIRC_ISO.INP
$!
$!  217 = 0.0
$!  224 = 0.0
$!  225 = 0.0
$!  226 = 0.0
$!  294 = 1.0  save gc's
$!
$DEFINE FOR006 'TAG'TAIL_CIRC_ISO.LST
$DEFINE FOR007 'TAG'TAIL_CIRC_ISO.GAM
$DEFINE FOR018 'TAG'TAIL_CIRC_SOL.GCSV
$DEFINE FOR027 'TAG'TAIL_CIRC_ISO.PGAM
$DEFINE FOR028 'TAG'TAIL_CIRC_ISO.GCS
$DEFINE FOR030 'TAG'TAIL_CIRC_ISO.WAKE
$RUN [-]F389SR
$DEASSIGN FOR002
$DEASSIGN FOR005
$DEASSIGN FOR006
$DEASSIGN FOR007
```

```
$DEASSIGN FOR018
$DEASSIGN FOR027
$DEASSIGN FOR028
$DEASSIGN FOR030
$!
$DEL 'TAG'TAIL_CIRC_ISO.GCM;*
$DEL 'TAG'TAIL_CIRC_ISO.GCS;*
$DEL 'TAG'TAIL_CIRC_ISO.GAM;*
$DEL 'TAG'TAIL_CIRC_ISO.WAKE;*
$!
$!  append 1st file
$!
$append 'TAG'TAIL_circ_iso.1st 'TAG'MAIN_tail.1st
$DEL 'TAG'TAIL_CIRC_ISO.LST;*
$!
$! calculate TAIL rotor circulation with MAIN rotor inflow
$!
$DEFINE FOR002 'TAG'TAIL_CIRC.GCM
$DEFINE FOR005 [-]TAIL_CIRC.INP
$!
$!  217 = 1.0  include induced velocities from another rotor
$!  224 = 0.0
$!  225 = 0.0
$!  226 = 0.0
$!  294 = 2.0  read gc's
$!
$DEFINE FOR006 'TAG'TAIL_CIRC.LST
$DEFINE FOR007 'TAG'TAIL_CIRC.GAM
$DEFINE FOR015 'TAG'MAIN_TAIL_IV.HARM
$DEFINE FOR018 'TAG'TAIL_CIRC_SOL.GCSV
$DEFINE FOR027 'TAG'TAIL_CIRC.PGAM
$DEFINE FOR028 'TAG'TAIL_CIRC.GCS
$RUN [-]F389SR
$DEASSIGN FOR002
$DEASSIGN FOR005
$DEASSIGN FOR006
$DEASSIGN FOR007
$DEASSIGN FOR015
$DEASSIGN FOR018
$DEASSIGN FOR027
$DEASSIGN FOR028
$!
$DEL 'TAG'TAIL_CIRC.GCM;*
$DEL 'TAG'TAIL_CIRC.GCS;*
```

```
$!
$!  append lst file
$!
$append 'TAG'TAIL_circc.lst 'TAG'MAIN_tail.lst
$DEL 'TAG'TAIL_CIRC.LST;*
$!
$! calculate induced velocity at MAIN rotor due to TAIL rotor
$!
$DEFINE FOR003 'TAG'TAIL_MAIN_IV.HARM
$DEFINE FOR005 [-]TAIL_MAIN_FP1.INP
$!
$!  217 = 0.0
$!  224 = 1.0  calc induced velocities at field points
$!  225 = 2.0  calc at another rotor disk
$!  226 = 2.0  calc harmonics of induced velocities at field points
$!  294 = 1.0  save gc's
$!
$DEFINE FOR006 'TAG'TAIL_MAIN_FP.LST
$DEFINE FOR007 'TAG'TAIL_CIRC.GAM
$DEFINE FOR027 'TAG'TAIL_CIRC.PGAM
$DEFINE FOR029 'TAG'TAIL_MAIN_FP.GCSV
$DEFINE FOR030 'TAG'TAIL_MAIN_FP.WAKE
$RUN [-]F389SR
$DEASSIGN FOR003
$DEASSIGN FOR005
$DEASSIGN FOR006
$DEASSIGN FOR007
$DEASSIGN FOR027
$DEASSIGN FOR029
$DEASSIGN FOR030
$!
$DEL 'TAG'TAIL_CIRC.GAM;*
$DEL 'TAG'TAIL_MAIN_FP.WAKE;*
$!
$!  append lst file
$!
$append 'TAG'TAIL_main_fp.lst 'TAG'MAIN_tail.lst
$DEL 'TAG'TAIL_CIRC_FP.LST;*
$!
$!
```

```
$!-------------
$!  SECOND PASS
$!-------------
$!
$! calculate MAIN ROTOR circulation with TAIL ROTOR inflow
$!
$DEFINE FOR002 'TAG'MAIN_CIRC.GCM
$DEFINE FOR005 [-]MAIN_CIRC.INP
$!
$!  217 = 1.0  include induced velocities from another rotor
$!  224 = 0.0
$!  225 = 0.0
$!  226 = 0.0
$!  294 = 2.0  read gc's
$!
DEFINE FOR006 'TAG'MAIN_CIRC.LST
$DEFINE FOR007 'TAG'MAIN_CIRC.GAM
$DEFINE FOR015 'TAG'TAIL_MAIN_IV.HARM
$DEFINE FOR018 'TAG'MAIN_CIRC_SOL.GCSV
$DEFINE FOR028 'TAG'MAIN_CIRC.GCS
$DEFINE FOR027 'TAG'MAIN_CIRC.PGAM
$RUN [-]F389SR
$DEASSIGN FOR002
$DEASSIGN FOR005
$DEASSIGN FOR006
$DEASSIGN FOR007
$DEASSIGN FOR015
$DEASSIGN FOR018
$DEASSIGN FOR027
$DEASSIGN FOR028
$!
$DEL 'TAG'MAIN_CIRC.GCM;*
$DEL 'TAG'MAIN_CIRC.GCS;*
$!
$!  append lst file
$!
$append 'TAG'MAIN_circ.lst 'TAG'MAIN_tail.lst
$DEL 'TAG'MAIN_CIRC.LST;*
$!
$! calculate induced velocity at TAIL ROTOR due to MAIN ROTOR
$!
$DEFINE FOR003 'TAG'MAIN_TAIL_IV.HARM
$DEFINE FOR005 [-]MAIN_TAIL_FP2.INP
```

```
$!
$!  217 = 0.0
$!  224 = 1.0  calc induced velocities at field points
$!  225 = 2.0  calc at another rotor disk
$!  226 = 2.0  calc harmonics of induced velocities at field points
$!  294 = 2.0  read gc's
$!
$DEFINE FOR006 'TAG'MAIN_TAIL_FP.LST
$DEFINE FOR007 'TAG'MAIN_CIRC.GAM
$DEFINE FOR027 'TAG'MAIN_CIRC.PGAM
$DEFINE FOR029 'TAG'MAIN_TAIL_FP.GCSV
$RUN [-]F389SR
$DEASSIGN FOR003
$DEASSIGN FOR005
$DEASSIGN FOR006
$DEASSIGN FOR007
$DEASSIGN FOR027
$DEASSIGN FOR029
$!
$DEL 'TAG'MAIN_CIRC.GAM;*
$!
$!  append lst file
$!
$append 'TAG'MAIN_tail_fp.lst 'TAG'MAIN_tail.lst
$DEL 'TAG'MAIN_TAIL_FP.LST;*
$!
$! calculate TAIL ROTOR circulation with MAIN ROTOR inflow
$!
$DEFINE FOR002 'TAG'TAIL_CIRC.GCM
$DEFINE FOR005 [-]TAIL_CIRC.INP
$!
$!  217 = 1.0  include induced velocities from another rotor
$!  224 = 0.0
$!  225 = 0.0
$!  226 = 0.0
$!  294 = 2.0  read gc's
$!
$DEFINE FOR006 'TAG'TAIL_CIRC.LST
$DEFINE FOR007 'TAG'TAIL_CIRC.GAM
$DEFINE FOR015 'TAG'MAIN_TAIL_IV.HARM
$DEFINE FOR018 'TAG'TAIL_CIRC_SOL.GCSV
$DEFINE FOR027 'TAG'TAIL_CIRC.PGAM
$DEFINE FOR028 'TAG'TAIL_CIRC.GCS
$RUN [-]F389SR
```

```
$DEASSIGN FOR002
$DEASSIGN FOR005
$DEASSIGN FOR006
$DEASSIGN FOR007
$DEASSIGN FOR015
$DEASSIGN FOR018
$DEASSIGN FOR027
$DEASSIGN FOR028
$!
$DEL 'TAG'TAIL_CIRC.GCM;*
$DEL 'TAG'TAIL_CIRC.GCS;*
$!
$!  append lst file
$!
$append 'TAG'TAIL_circ.lst 'TAG'MAIN_tail.lst
$DEL 'TAG'TAIL_CIRC.LST;*
$!
$! calculate induced velocity at MAIN ROTOR due to TAIL ROTOR
$!
$DEFINE FOR003 'TAG'TAIL_MAIN_IV.HARM
$DEFINE FOR005 [-]TAIL_MAIN_FP2.INP
$!
$!  217 = 0.0
$!  224 = 1.0  calc induced velocities at field points
$!  225 = 2.0  calc at another rotor disk
$!  226 = 2.0  calc harmonics of induced velocities at field points
$!  294 = 2.0  read gc's
$!
$DEFINE FOR006 'TAG'TAIL_MAIN_FP.LST
$DEFINE FOR007 'TAG'TAIL_CIRC.GAM
$DEFINE FOR027 'TAG'TAIL-CIRC.PGAM
$DEFINE FOR029 'TAG'TAIL_MAIN_FP.GCSV
$RUN [-]F389SR
$DEASSIGN FOR003
$DEASSIGN FOR005
$DEASSIGN FOR006
$DEASSIGN FOR007
$DEASSIGN FOR027
$DEASSIGN FOR029
$!
$DEL 'TAG'TAIL_CIRC.GAM;*
$!
$!  append lst file
$!
$append 'TAG'TAIL_main_fp.lst 'TAG'MAIN_tail.lst
$DEL 'TAG'TAIL_MAIN_FP.LST;*
$!
```

```
$!-------------
$!  THIRD PASS
$!-------------
$!
$! calculate MAIN ROTOR circulation with TAIL ROTOR inflow
$!
$DEFINE FOR002 'TAG'MAIN_CIRC.GCM
$DEFINE FOR005 [-]MAIN_CIRC.INP
$DEFINE FOR006 'TAG'MAIN_CIRC.LST
$DEFINE FOR007 'TAG'MAIN_CIRC.GAM
$DEFINE FOR015 'TAG'TAIL_MAIN_IV.HARM
$DEFINE FOR018 'TAG'MAIN_CIRC_SOL.GCSV
$DEFINE FOR028 'TAG'MAIN_CIRC.GCS
$DEFINE FOR027 'TAG'MAIN_CIRC.PGAM
$RUN [-]F389SR
$DEASSIGN FOR002
$DEASSIGN FOR005
$DEASSIGN FOR006
$DEASSIGN FOR007
$DEASSIGN FOR015
$DEASSIGN FOR018
$DEASSIGN FOR027
$DEASSIGN FOR028
$!
$DEL 'TAG'MAIN_CIRC.GCM;*
$DEL 'TAG'MAIN_CIRC.GCS;*
$!
$!  append lst file
$!
$append 'TAG'MAIN_circ.lst 'TAG'MAIN_tail.lst
$DEL 'TAG'MAIN_CIRC.LST;*
$!
$! calculate induced velocity at TAIL ROTOR due to MAIN ROTOR
$!
$DEFINE FOR003 'TAG'MAIN_TAIL_IV.HARM
$DEFINE FOR005 [-]MAIN_TAIL_FP2.INP
$DEFINE FOR006 'TAG'MAIN_TAIL_FP.LST
$DEFINE FOR007 'TAG'MAIN_CIRC.GAM
$DEFINE FOR027 'TAG'MAIN_CIRC.PGAM
$DEFINE FOR029 'TAG'MAIN_TAIL_FP.GCSV
$RUN [-]F389SR
$DEASSIGN FOR003
$DEASSIGN FOR005
$DEASSIGN FOR006
$DEASSIGN FOR007
$DEASSIGN FOR027
$DEASSIGN FOR029
```

```
$!
$DEL 'TAG'MAIN_CIRC.GAM;*
$!
$!  append lst file
$!
$append 'TAG'MAIN_tail_fp.lst 'TAG'MAIN_tail.lst
$DEL 'TAG'MAIN_TAIL_FP.LST;*
$!
$! calculate TAIL ROTOR circulation with MAIN ROTOR inflow
$!
$DEFINE FOR002 'TAG'TAIL_CIRC.GCM
$DEFINE FOR005 [-]TAIL_CIRC.INP
$DEFINE FOR006 'TAG'TAIL_CIRC.LST
$DEFINE FOR007 'TAG'TAIL_CIRC.GAM
$DEFINE FOR015 'TAG'MAIN_TAIL_IV.HARM
$DEFINE FOR018 'TAG'TAIL_CIRC_SOL.GCSV
$DEFINE FOR027 'TAG'TAIL_CIRC.PGAM
$DEFINE FOR028 'TAG'TAIL_CIRC.GCS
$RUN [-]F389SR
$DEASSIGN FOR002
$DEASSIGN FOR005
$DEASSIGN FOR006
$DEASSIGN FOR007
$DEASSIGN FOR015
$DEASSIGN FOR018
$DEASSIGN FOR027
$DEASSIGN FOR028
$!
$DEL 'TAG'TAIL_CIRC.GCM;*
$DEL 'TAG'TAIL_CIRC.GCS;*
$!
$!  append lst file
$!
$append 'TAG'TAIL_circ.lst 'TAG'MAIN_tail.lst
$DEL 'TAG'TAIL_CIRC.LST;*
$!
$! calculate induced velocity at MAIN ROTOR due to TAIL ROTOR
$!
$DEFINE FOR003 'TAG'TAIL_MAIN_IV.HARM
$DEFINE FOR005 [-]TAILMMAIN_FP2.INP
$DEFINE FOR006 'TAG'TAIL_MAIN_FP.LST
$DEFINE FOR007 'TAG'TAIL_CIRC.GAM
$DEFINE FOR027 'TAG'TAIL_CIRC.PGAM
$DEFINE FOR029 'TAG'TAIL_MAIN_FP.GCSV
$RUN [-]F389SR
```

```
$DEASSIGN FOR003
$DEASSIGN FOR005
$DEASSIGN FOR006
$DEASSIGN FOR007
$DEASSIGN FOR027
$DEASSIGN FOR029
$!
$DEL 'TAG'TAIL_CIRC.GAM;*
$!
$!  append lst file
$!
$append 'TAG'TAIL_main_fp.lst 'TAG'MAIN_tail.lst
$DEL 'TAG'TAIL_MAIN_FP.LST;*
$!
$!
$!  FINAL MAIN ROTOR CIRC PASS
$!
$! calculate MAIN ROTOR circulation with TAIL ROTOR inflow
$!
$DEFINE FOR002 'TAG'MAIN_CIRC.GCM
$DEFINE FOR005 [-]MAIN_CIRC.INP
$DEFINE FOR006 'TAG'MAIN_CIRC.LST
$DEFINE FOR007 'TAG'MAIN_CIRC.GAM
$DEFINE FOR015 'TAG'TAIL_MAIN_IV.HARM
$DEFINE FOR018 'TAG'MAIN_CIRC_SOL.GCSV
$DEFINE FOR028 'TAG'MAIN_CIRC.GCS
$DEFINE FOR027 'TAG'MAIN_CIRC.PGAM
$RUN [-]F389SR
$DEASSIGN FOR002
$DEASSIGN FOR005
$DEASSIGN FOR006
$DEASSIGN FOR007
$DEASSIGN FOR015
$DEASSIGN FOR018
$DEASSIGN FOR027
$DEASSIGN FOR028
$!
$DEL 'TAG'MAIN_CIRC.GCM;*
$DEL 'TAG'MAIN_CIRC.GCS;*
$!
$!  append lst file
$!
$append 'TAG'MAIN_circ.lst 'TAG'MAIN_tail.lst
$DEL 'TAG'MAIN_CIRC.LST;*
$!
$SET DEF [-]
```

REFERENCES

1. Lorber, P.F., and Egolf, T.A., "An Unsteady Helicopter Rotor - Fuselage Interaction Analysis," UTRC Report R88-956977-15, August 1988. (to be published as a NASA Contractor Report)

2. Sheehy, T.W., and Clark, D.R., "A Method for Predicting Helicopter Hub Drag," USAAMRDL-TR-75-48, Jan. 1976.

3. Studwell, R.E., "User's Manual for the Automated Paneling Technique (APT) and the Wing Body Aerodynamic Technique (WABAT) Programs," NASA CR-165895, June 1982.

4. Egolf, T.A., and Landgrebe, A.J., "User's Manual for the UTRC Prescribed Wake Rotor Inflow and Flow Field Prediction Analysis," UTRC81-2, August 1981, supplied to the NASA Langley Structures Laboratory, Army Research and Development Laboratory, (AVRADCOM) under Contract NAS1-16058.

5. Bielawa, R.L., "Extended Aeroelastic Analysis for Helicopter Rotors with Prescribed Hub Motion and Blade Appended Pendular Vibration Absorbers," NASA CR-172455, Dec. 1984.

6. Loiselle, J.W., "Generalized Helicopter Rotor Performance Predictions," Master's Thesis, Naval Postgraduate School, Monterey CA, Sep. 1977.

7. Studwell, R.E., "Set-Up and Input Requirements for the Sikorsky GRP Program," letter to T.A. Egolf, Nov. 1974.

8. Shoosmith, J.N., ed., "Introduction to the use of the Langley Central Scientific Computing Complex," Central Scientific Computing Complex Document A-1c, August 1986.

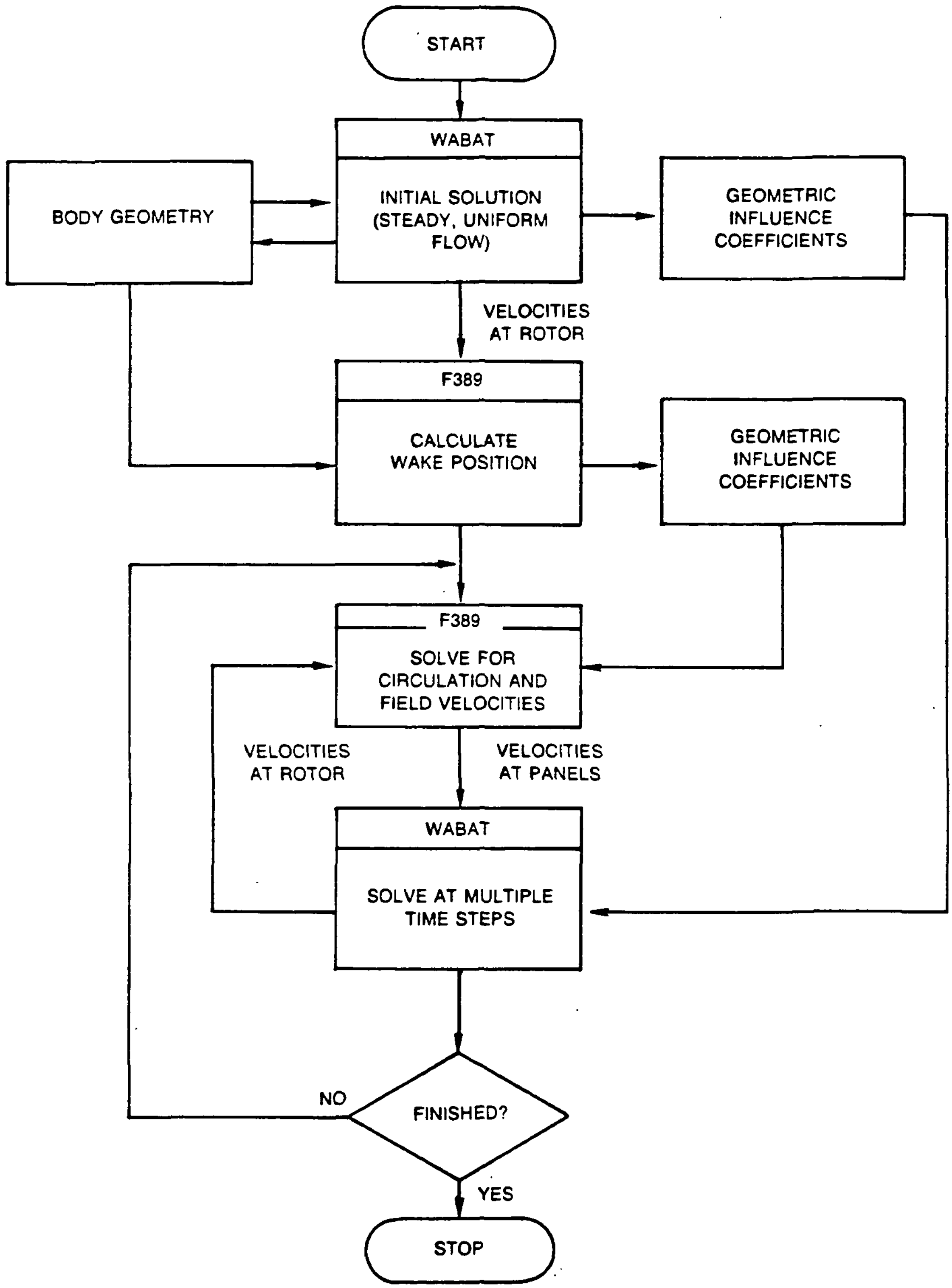


Figure 1. Program calling sequence for rotor-fuselage analysis.

a) Prescribed rotor position.

87—11—47—2

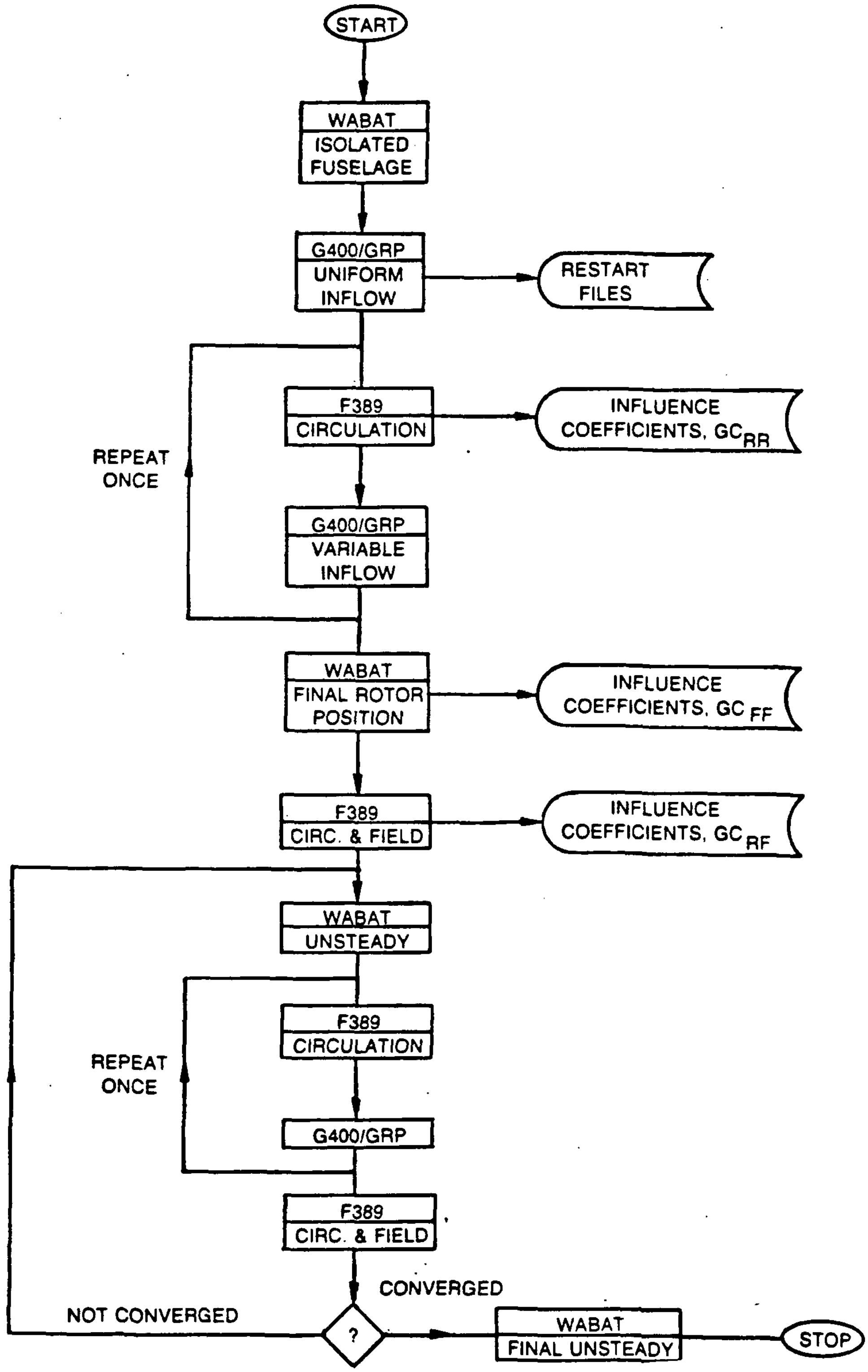


Figure 1. Program calling sequence (concluded).
b) With a blade response program.

87-1?-47-21

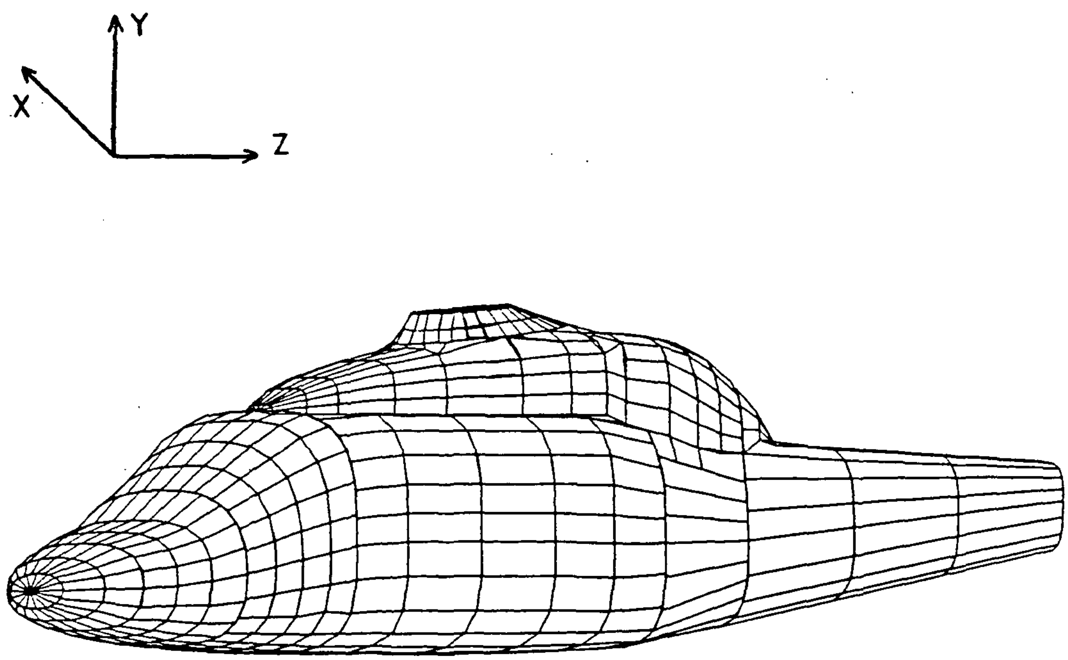


Figure 2. Fuselage panel model and coordinates for the wake displacement body.

C-2

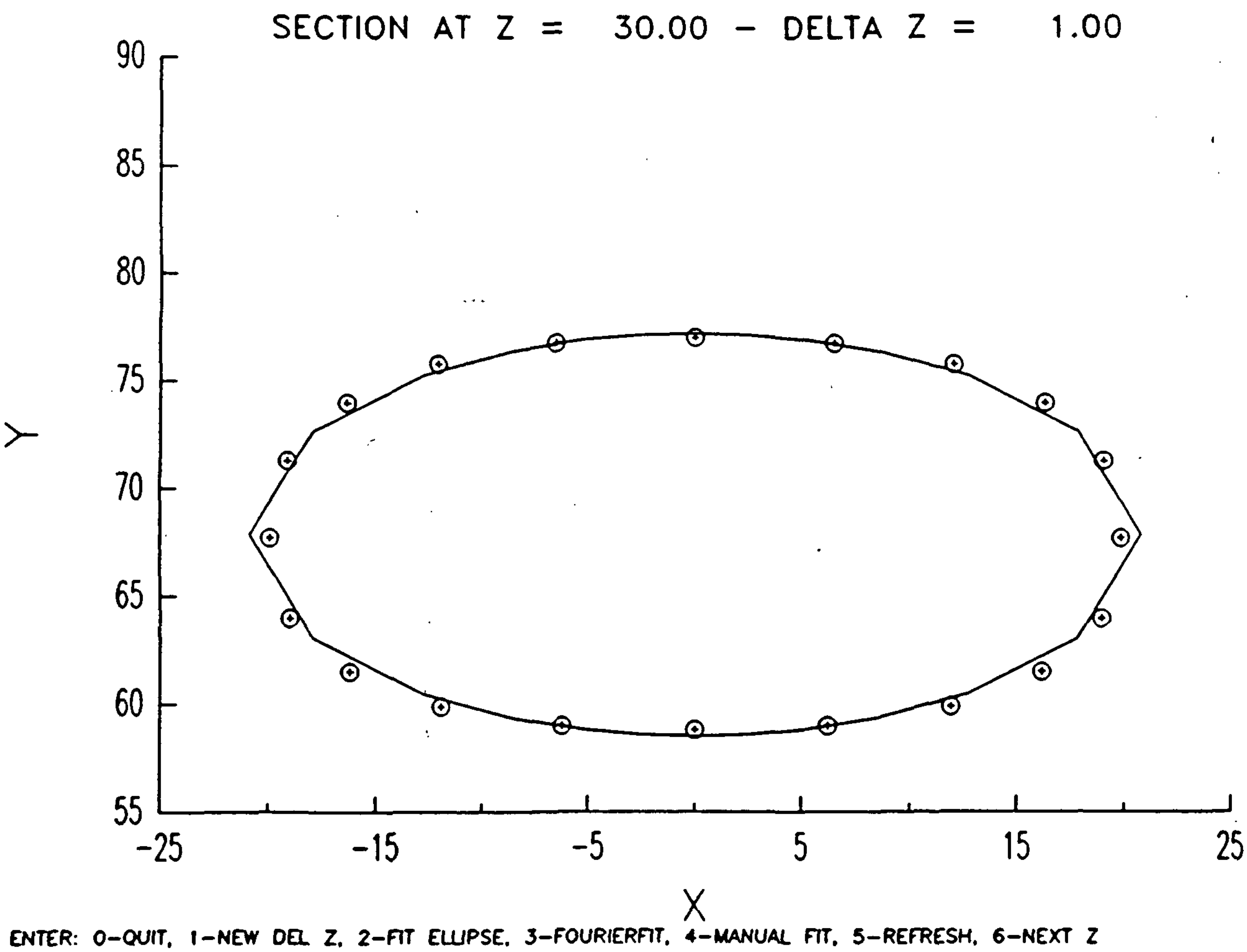
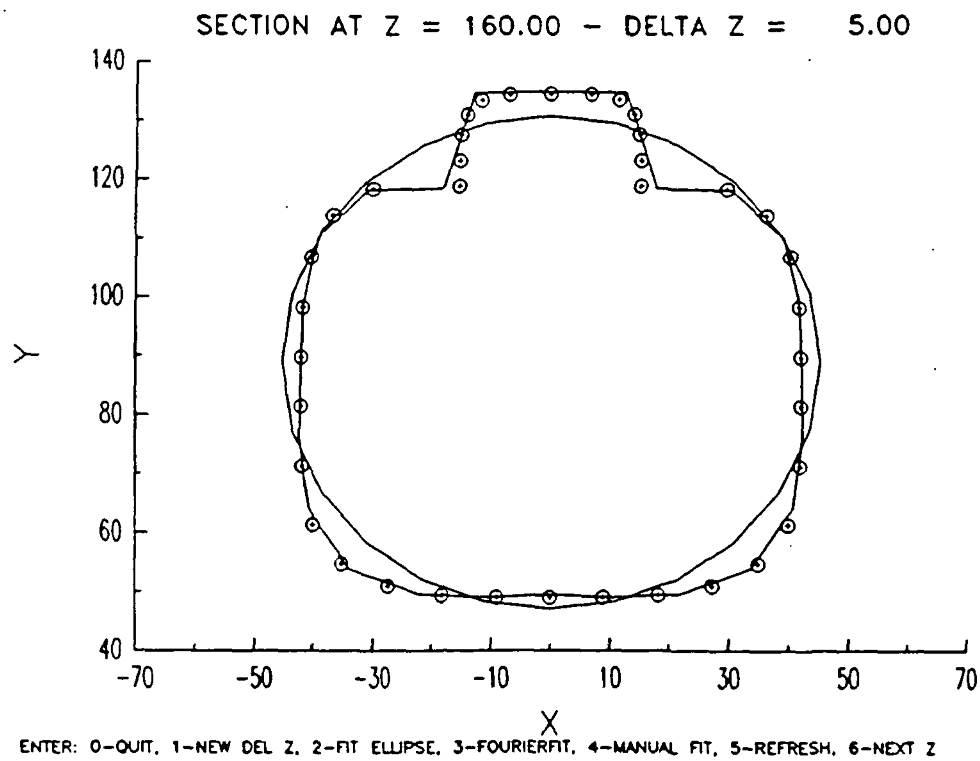
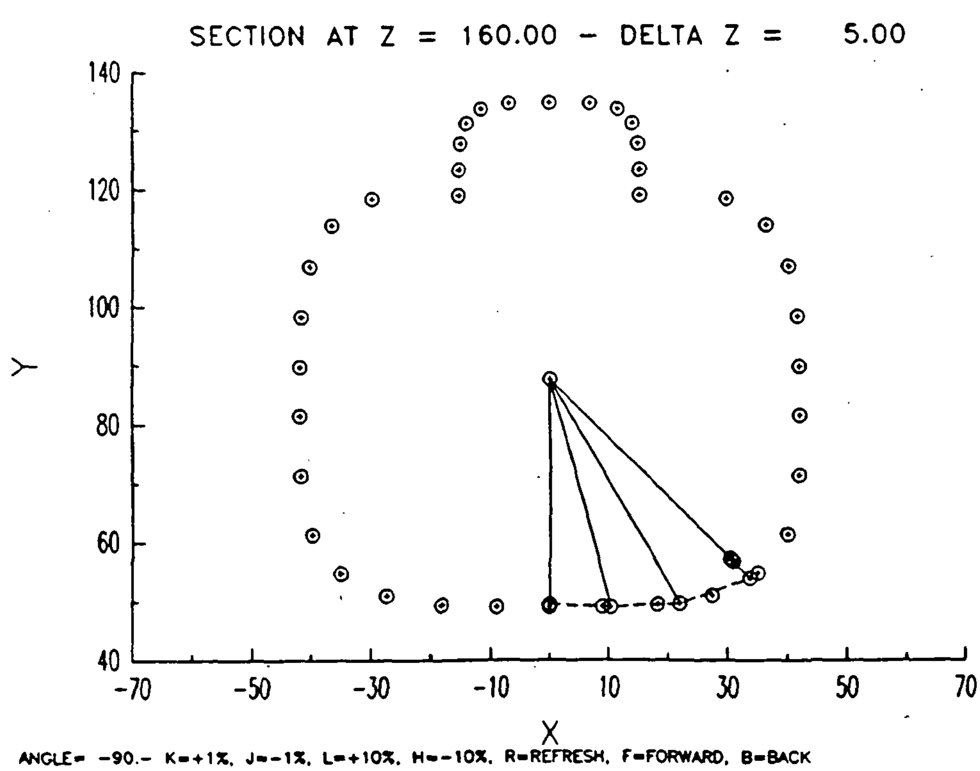


Figure 3. Terminal display: Fuselage panel corners and an elliptical fit.

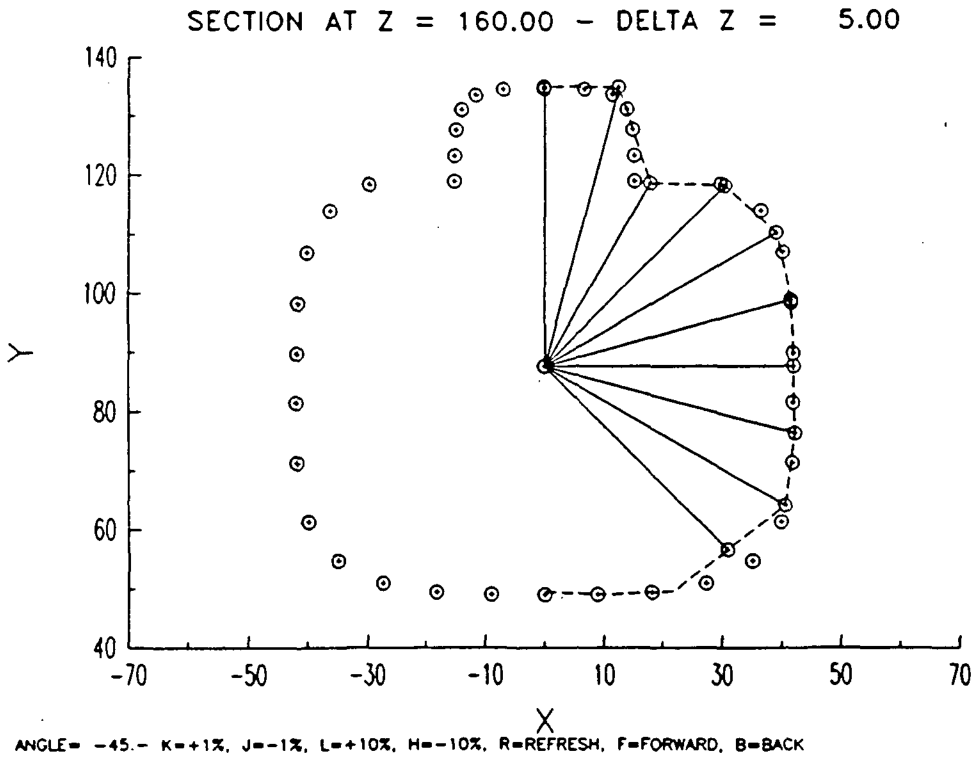


a) Fuselage panel corners and elliptical fit.

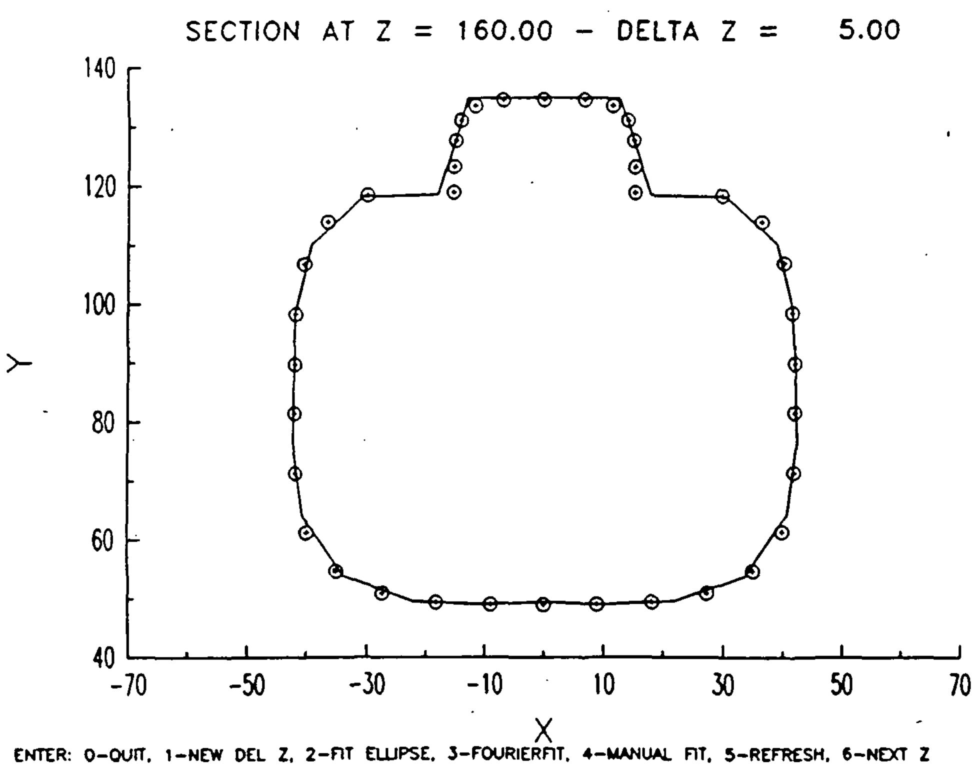


b) Start of the manual fit.

Figure 4. Terminal display: Example of a manual fit.



c) Continuation of the manual fit.



d) Competed manual fit.

Figure 4. Terminal display: Example of a manual fit (concluded).

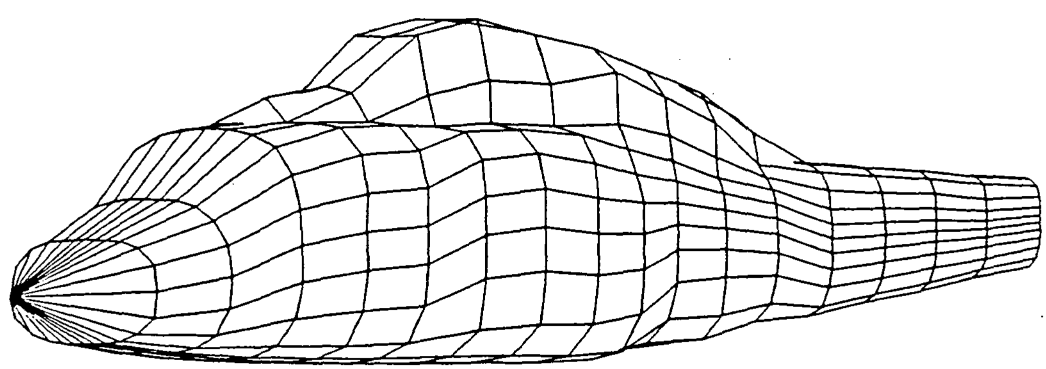
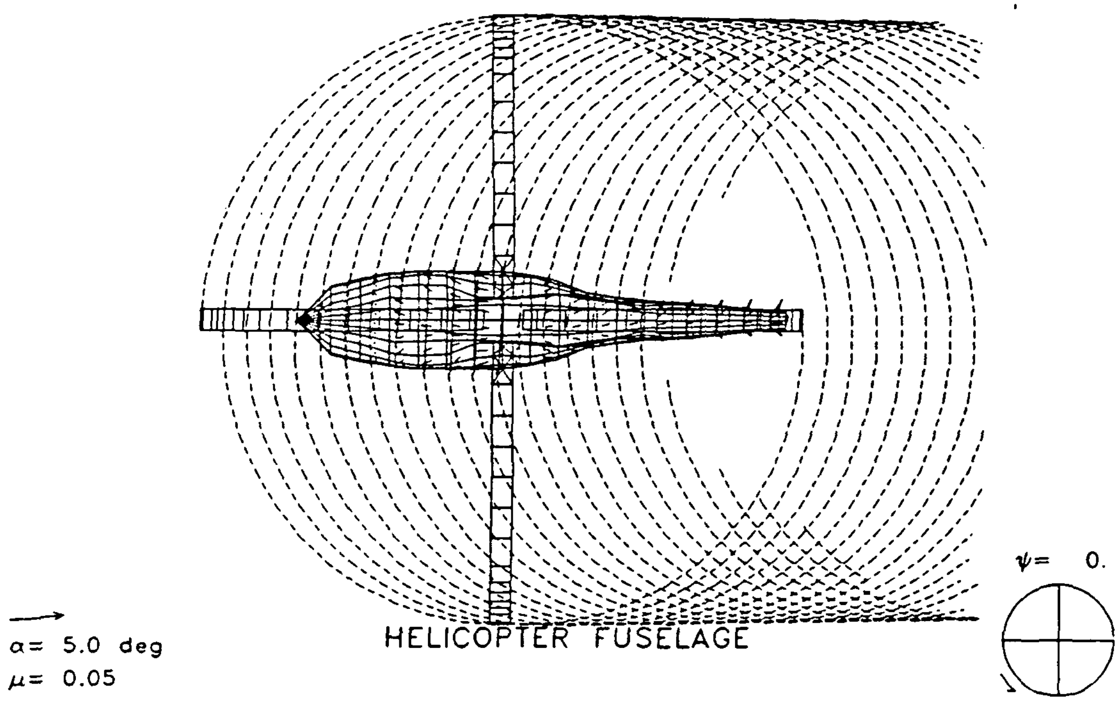


Figure 5. Panel representation of the wake displacement body.



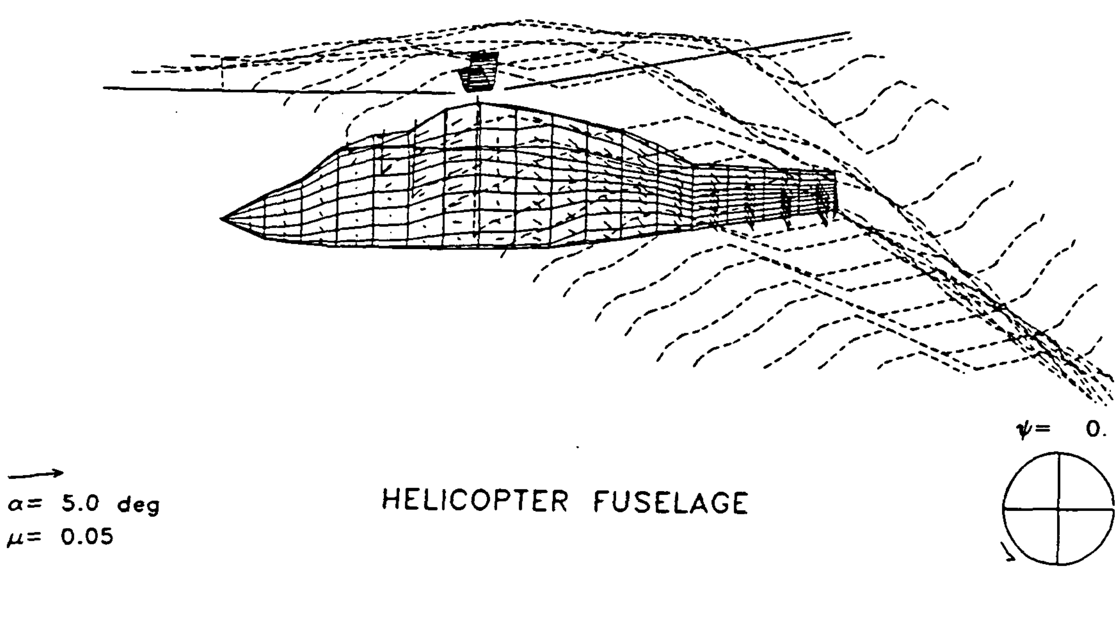
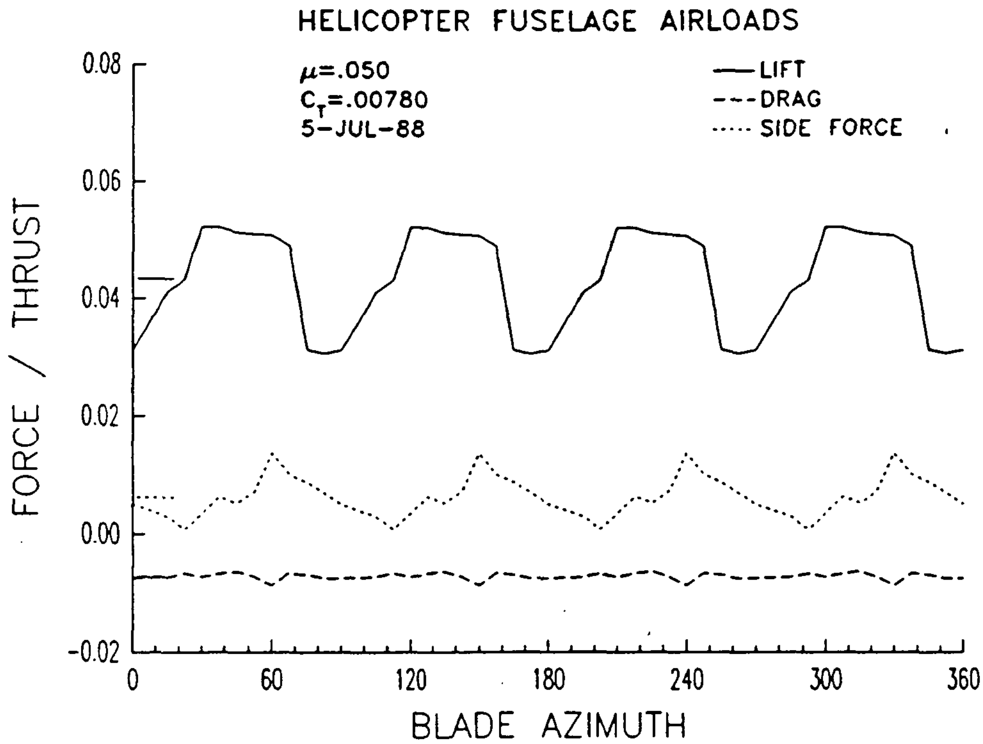
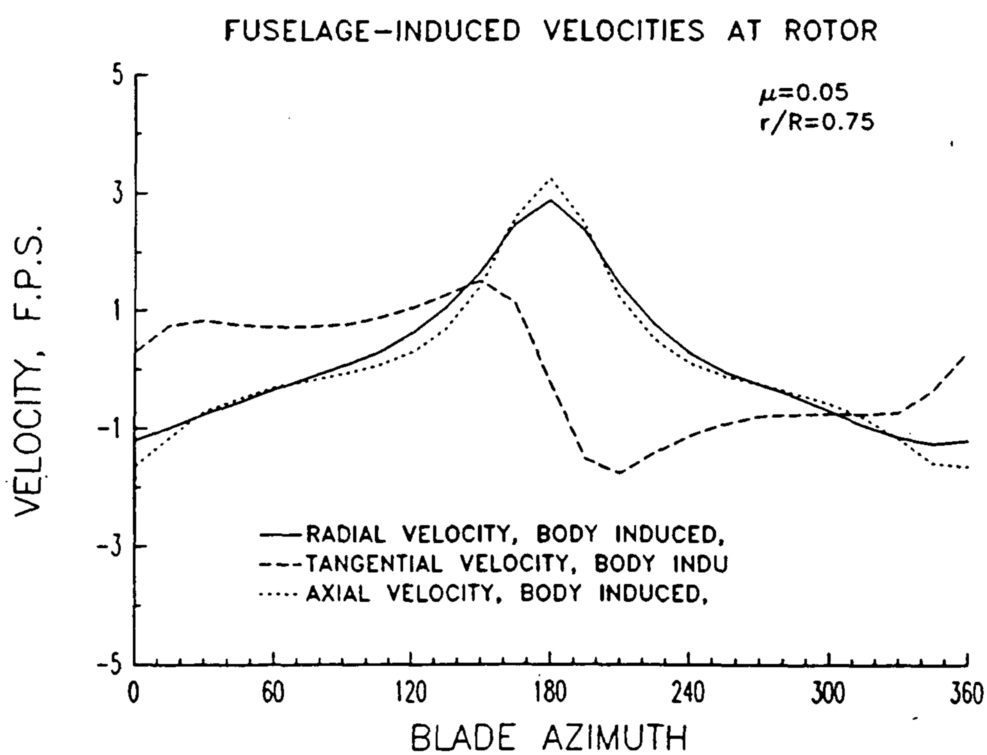


Figure 6. Example of the output of WABPLOT.

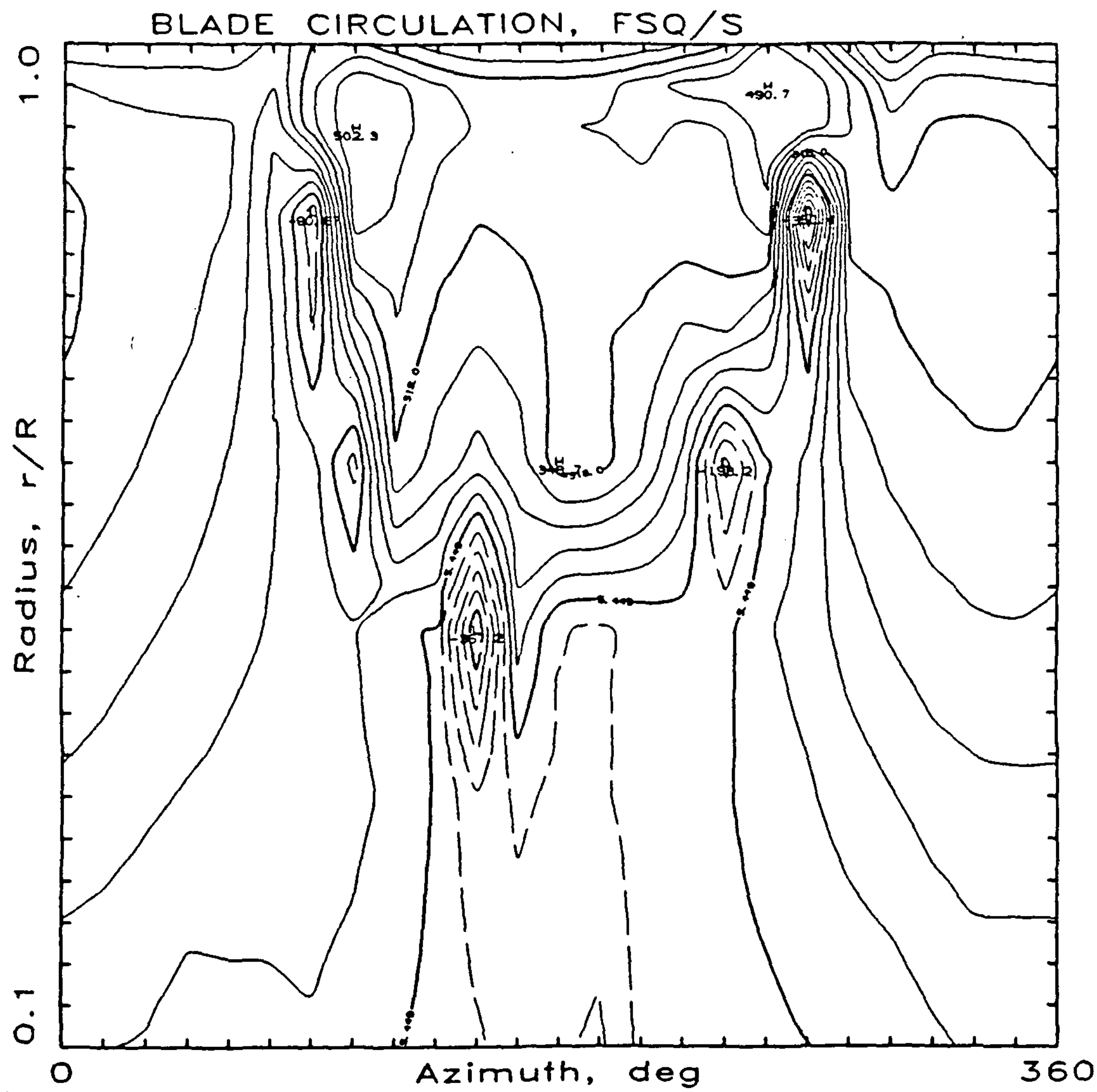


a) Aerodynamic forces on the fuselage.



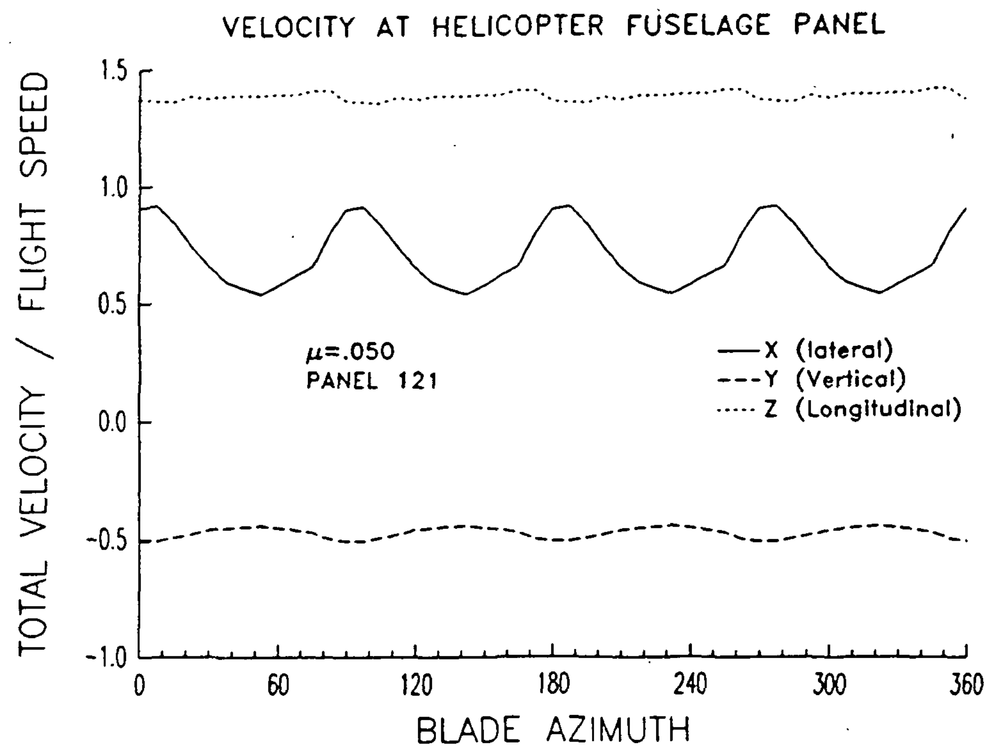
b) Fuselage-induced velocities at the rotor.

Figure 7. Example of the output of **PLOTWAB**.

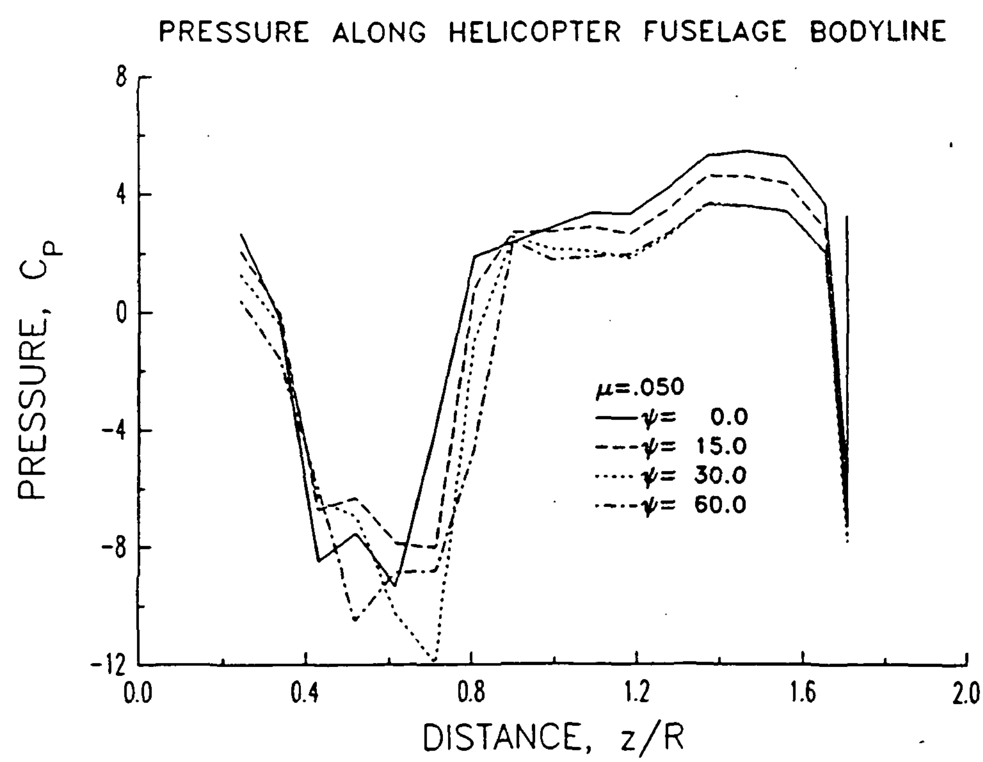


c) Contour plot of the blade bound circulation.

Figure 7. Example of the output of PLOTWAB (concluded).



a) Time history of the velocity at a fuselage panel.



b) Pressure along a fuselage bodyline.

Figure 8. Example of the output of **CVFPLT**.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No. NASA CR-181701	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Program User's Manual for an Unsteady Helicopter Rotor-Fuselage Aerodynamic Analysis		5. Report Date August 1988
		6. Performing Organization Code
7. Author(s) Peter F. Lorber		8. Performing Organization Report No. R88-956977-14
		10. Work Unit No. 505-61-51-10
9. Performing Organization Name and Address United Technologies Research Center East Hartford, CT 06108		11. Contract or Grant No. NAS1-17469
		13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Langley Research Center Hampton, VA 23665-5225		14. Sponsoring Agency Code

15. Supplementary Notes

Langley Technical Monitor: John C. Wilson

16. Abstract

The Rotor-Fuselage Analysis is a method of calculating the aerodynamic interaction between a helicopter rotor and fuselage. This manual describes the structure and operation of the computer programs that make up the Rotor-Fuselage Analysis, the programs that prepare the input and the programs that display the output.

17. Key Words (Suggested by Author(s)) Helicopter Aerodynamics	18. Distribution Statement Unclassified - Unlimited Subject Category 02

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of pages 106	22. Price

NASA FORM 1626 OCT 86

PREPARATION OF THE REPORT DOCUMENTATION PAGE

The last page of a report facing the third cover is the Report Documentation Page, RDP. Information presented on this page is used in announcing and cataloging reports as well as preparing the cover and title page. Thus it is important that the information be correct. Instructions for filling in each block of the form are as follows:

Block 1. Report No. NASA report series number, if preassigned.

Block 2. Government Accession No. Leave blank.

Block 3. Recipient's Catalog No. Reserved for use by each report recipient.

Block 4. Title and Subtitle. Typed in caps and lower case with dash or period separating subtitle from title.

Block 5. Report Date. Approximate month and year the report will be published.

Block 6. Performing Organization Code. Leave blank.

Block 7. Author(s). Provide full names exactly as they are to appear on the title page. If applicable, the word editor should follow a name.

Block 8. Performing Organization Report No. NASA installation report control number and, if desired, the non-NASA performing organization report control number.

Block 9. Performing Organization Name and Address. Provide affiliation (NASA program office, NASA installation, or contractor name) of authors.

Block 10. Work Unit No. Provide Research and Technology Objectives and Plans (RTOP) number.

Block 11. Contract or Grant No. Provide when applicable.

Block 12. Sponsoring Agency Name and Address. National Aeronautics and Space Administration, Washington, D.C. 20546-0001. If contractor report, add NASA installation or HQ program office.

Block 13. Type of Report and Period Covered. NASA formal report series; for Contractor Report also list type (interim, final) and period covered when applicable.

Block 14. Sponsoring Agency Code. Leave blank.

Block 15. Supplementary Notes. Information not included elsewhere: affiliation of authors if additional space is required for block 9, notice of work sponsored by another agency, monitor of contract, information about supplements (film, data tapes, etc.), meeting site and date for presented papers, journal to which an article has been submitted, note of a report made from a thesis, appendix by author other than shown in block 7.

Block 16. Abstract. The abstract should be informative rather than descriptive and should state the objectives of the investigation, the methods employed (e.g., simulation, experiment, or remote sensing), the results obtained, and the conclusions reached.

Block 17. Key Words. Identifying words or phrases to be used in cataloging the report.

Block 18. Distribution Statement. Indicate whether report is available to public or not. If not to be controlled, use "Unclassified-Unlimited." If controlled availability is required, list the category approved on the Document Availability Authorization Form (see NHB 2200.2, Form FF427). Also specify subject category (see "Table of Contents" in a current issue of STAR), in which report is to be distributed.

Block 19. Security Classification (of this report). Self-explanatory.

Block 20. Security Classification (of this page). Self-explanatory.

Block 21. No. of Pages. Count front matter pages beginning with iii, text pages including internal blank pages, and the RDP, but not the title page or the back of the title page.

Block 22. Price Code. If block 18 shows "Unclassified-Unlimited," provide the NTIS price code (see "NTIS Price Schedules" in a current issue of STAR) and at the bottom of the form add either "For sale by the National Technical Information Service, Springfield, VA 22161-2171" or "For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402-0001," whichever is appropriate.