A User’s Manual for ROTTILT Solver: Tiltrotor Fountain Flow Field Prediction

Hormoz Tadghighi
The Boeing Company, Mesa, Arizona

R. Ganesh Rajagopalan
Iowa State University, Ames, Iowa

January 1999
Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results ... even providing videos.

For more information about the NASA STI Program Office, see the following:


- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at (301) 621-0134

- Phone the NASA STI Help Desk at (301) 621-0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320
A User’s Manual for ROTTILT Solver: Tiltrotor Fountain Flow Field Prediction

Hormoz Tadghighi
The Boeing Company, Mesa, Arizona

R. Ganesh Rajagopalan
Iowa State University, Ames, Iowa
12 Appendices

A Sample Input File

B ROTTILT Input Geometry Scale Down or Up Program

C ROTTILT Grid Conversion into a 2D Gnuplot’s Format

D Geometry Section Cuts Generation Program

E ROTTILT-WOPWOP-TIN2 Hi-Res Airloads and Inflow Velocities Conversions

F Hess Write Format

G 2D Gnuplot Plotting Routine

H ROTTILT Input File for the XV15 Full Span Aircraft

I Hess Format Sample Data
## List of Figures

1. Schematic of Fountain Flow Recirculation .................................................. 1
2. Schematic of Rotor based Cartesian System ............................................... 3
3. A Schematic of Rotor Based Cylindrical System .......................................... 4
4. Schematic of Rotor based Cartesian System ............................................... 4
5. A Schematic of Rotor Based Cylindrical System .......................................... 6
6. A Schematic of Aerodynamic Forces on the Airfoil Section at "s" .................... 7
7. ROTTILT Solver Coupled with the Rotor(s) Trim Loop Flow Chart .................. 11
8. ROTTILT Input Geometry Preparation Process ........................................... 12
9. 3D View of the Section Cuts for the Semi-Span XV15 Aircraft ...................... 12
10. ROTTILT Internal Grid Generation Process ............................................. 12
11. XYZ Plane View of the Computational Grid of a Full-Span XV15 Aircraft ..... 13
12. XV15 Semi-Span Configuration in ROTTILT Computational Grid Domain .... 14
13. Three-block grid specification ........................................................................ 21
14. ROTTILT Coupling with WOPWOP and TIN2 Flow Chart ............................ 24
15. XV15 Isolated Hover Performance ................................................................ 25
16. Numerical Simulation of Full Span XV15 Configuration in Hover - ROTTILT Velocity Trace 26
17. Numerical Simulation of Full Span XV15 Configuration in Hover - ROTTILT Velocity Vectors 27
18. Azimuthal Variation of Rotor Mean Inflow for Full Span XV15 in Hover ........ 28
19. Radial Distribution of Local Inflow Velocity for Full Span XV15 in Hover - ROTTILT Solution 29
20. Comparison of Rotor Mean Inflow Velocity for a 15 percent Scaled JVX Rotor 29
21. Hover Acoustic Pressures for XV15 Aircraft: Along its Longitudinal Axis ...... 30
22. Global coordinates and boundary conditions .............................................. 39
List of Tables

1. List of Namelist Parameters in CNTRLPARMS ........................................ 10
2. List of Namelist Parameters in AdjustBox ........................................... 15
3. List of Namelist Parameters in Layer Control ....................................... 23
5. A2: Flow Field Print Control Flags .................................................... 37
6. A3: Solid Body Boundary Condition Control Flags .................................. 37
7. A4: Geometry and Flow Print Control Flags ......................................... 38
8. A5: Flow Numerical Iteration Control Variables ................................... 38
9. A6: Numerical Relaxation Control Variables ....................................... 39
10. A7: Flowfield Domain Boundary Condition Variables .............................. 40
11. A8: Flowfield Numerical Solution Directive Control Flags ....................... 40
12. A9: Computational Domain BC Setting Flags ...................................... 41
13. A10: Numerical Relaxation Control Variables ..................................... 41
14. A11: Rotor Airfoils Configuration Flags ............................................ 42
16. A13: Number of Rotor Configuration Control ....................................... 43
17. A14: Rotor Airfoil Identification Command ......................................... 43
18. A15: Rotor Blade Characteristics .................................................... 44
19. A16: Rotor Configuration ............................................................ 44
20. A17: Rotor Blade Aerodynamic Control Point Setting ............................ 44
21. A18: Rotor Blade Geometry and Operating Controls .............................. 45
22. A19: Rotor Blade Flap and Pitch Motions ......................................... 45
23. A20: Rotor Blade Geometry Definitions ............................................ 46
26. A23: Rotor C81 Table File(s) Designation ......................................... 47
27. A24: Aircraft Components Declaration ............................................. 47
28. A25: Aircraft Trim Conditions ....................................................... 47
Abstract

A CFD solver has been developed to provide the time averaged details of the fountain flow typical for tiltrotor aircraft in hover. This Navier-Stokes solver, designated as ROTTILT, assumes the three-dimensional fountain flowfield to be steady and incompressible. The details of the theoretical background are described fully in this manual. In order to enable the rotor trim solution in the presence of tiltrotor aircraft components such as wing, nacelle, and fuselage, the solver is coupled with a newly developed set of dynamic trim routines which are highly efficient in CPU and suitable for CFD analysis. The Cartesian computational grid technique utilized in ROTTILT provides the user with a unique capability for insertion or elimination of any components of the bodies considered for the tiltrotor aircraft configuration by the user. Their presence in the flow field domain can be controlled by a simple logical switch in the input file. Flow field associated with either a semi or full-span configuration can be computed through user options in the ROTTILT input file. Full details associated with the numerical solution implemented in the solver are presented along with the assumptions. A brief description pertaining to the preparation of input surface mesh topology is provided with the listing for all the preprocessor programs in the appendices. Detailed definition of all the input variables with their default values listed in the main input files are described with reference to the V22 aircraft. Limited validation results for the V22 aircraft in hover obtained from the coupled ROTTILT/WOPWOP program is furnished for completion.

A pre-processor program based on the graphics package GNU-PLOT3D is also provided to visualize the computational grid. Sample input files for the V22 aircraft containing the geometry coordinates associated with the wing, nacelle, and fuselage components are also included to enable the reproduction of the solutions given in this report.
Nomenclature

\( b \) = number of blades
\( c \) = rotor blade chord
\( C_l, C_d \) = sectional lift and drag coefficients, respectively
\( C_p \) = pressure coefficient \( \frac{P - P_\infty}{q_\infty} \)
\( D' \) = drag force per unit span at a section
\( \hat{e}_n, \hat{e}_\phi, \hat{e}_s \) = unit vectors in the \((n, \phi, s)\) system
\( \hat{e}_r, \hat{e}_\theta, \hat{e}_z \) = unit vectors in the cylindrical coordinate system
\( \vec{F} \) = resultant aerodynamic force on the blade, in the \((n, \phi, s)\) system
\( \vec{F}_r \) = resultant aerodynamic force on the blade, in the \((X, Y, Z)\) system
\( \hat{I}, \hat{J}, \hat{K} \) = unit vectors in the \((X, Y, Z)\) system
\( L' \) = lift force per unit span at a section
\( M_1, M_2, M_3 \) = transformation matrices
\( n, \phi, s \) = coordinate system attached to the blade
\( p \) = static pressure
\( q_\infty \) = free-stream dynamic pressure
\( R \) = radius of the rotor
\( \vec{R} \) = position vector of a point on the blade span, in the \((r, \theta, z)\) system
\( s \) = generalized coordinate measured along the blade span
\( S_X, S_Y, S_Z \) = source terms in the discretized momentum equation
\( S'_X, S'_Y, S'_Z \) = source terms in the momentum equation
\( \vec{V}_{rel} \) = flow velocity relative to the blade
\( V_\infty \) = freestream velocity
\( \vec{V} \) = absolute velocity of the flow with respect to the global coordinates
\( X, Y, Z \) = reference frame attached to the computational domain
\( \alpha \) = blade angle of attack
\( \alpha_{TD} \) = tip-path-plane angle of attack
\( \delta \) = blade deflection out of the rotor plane
\( \xi, \eta, \zeta \) = rotor-based Cartesian coordinates
\( \theta_r \) = blade pitch at the root
\( \theta_s \) = blade pitch at any section
\( \theta_{tw} \) = linear rate of twist
\( \mu \) = fluid viscosity
\( \rho \) = fluid density
\( \psi \) = azimuth angle
\( \Omega \) = rotational velocity of the rotor
\( \vec{\zeta} \) = vorticity vector \( \nabla \times \vec{V} \)
\( \| \vec{\zeta} \| \) = magnitude of vorticity \( \sqrt{\zeta_X^2 + \zeta_Y^2 + \zeta_Z^2} \)
\( \zeta_X, \zeta_Y, \zeta_Z \) = components of \( \vec{\zeta} \)
1 Introduction

Air traffic congestion at major airports throughout the world is quickly reaching the saturation point. The economic and political difficulties of constructing new major airports in heavily populated areas are enormous. Smaller commuter aircraft account for approximately 30 percent of airport usage while carrying only 5 percent of the passengers. A civil tiltrotor transport, operating in a National Airspace System tailored to permit vertiport access independent of airport control, would allow a significant increase in airport passenger movements. This would also minimize the requirement for new or large expansions to existing airports. The high noise levels generated by the tiltrotors and particularly during long periods in a terminal area could raise the ire community surrounding a vertiport. For tiltrotor transport to be a viable option the flow field and noise associated with terminal area operations must be understood.

The tiltrotor flow field in hover has been observed experimentally to possess many interesting features [Ref. 1]. Because the proprotor is nominally one wing chord length above the wing, the flow fields induced by the wing and the rotor are closely coupled. The flow field associated with the wing is largely unsteady and turbulent, and separated beneath the wing. As shown in Figure [1], the inboard-moving spanwise flow on the upper surface from both wing panels meets at the vehicle center-line and is redirected upward, and then downward through the rotors creating a recirculation pattern referred to as fountain flow.

![Fountain Flow Region](image)

Figure 1: Schematic of Fountain Flow Recirculation

A CFD solver developed by the second author designated as ROTTILT solver [2] utilizes an efficient numerical technique using Navier-Stokes equations to simulate the flowfield associated with the tiltrotor aircraft in hover. The current version of the ROTTILT is equipped with an automated non-body fitted computational grid generation capability. The presence of the fountain flow for the tiltrotor in hover is significant in contributing to the proprotor unsteady airloads, download or vertical drag, and its impulsive noise characteristics in hover.

The numerical prediction of all the flow features associated with the fountain flow still lies beyond the state-of-the-art. However, to render the problem tractable, the rotor is simplified in our analysis (see [Ref. 2]). A non-body conforming grid topology coupled with the ROTTILT solver offers a unique flexibility by accommodating a complex geometry such as semi-span/full-span tiltrotor configuration using a single grid block topology without sacrificing the accuracy of the solution sought. Using the grid control capabilities in the ROTTILT solver, we are able to employ a fine grid near the rotor disc plane. This preserves the accuracy of the numerical solution of the complex flow field around the rotor blade which is crucial for both performance and acoustic analysis. A non-body fitted computational grid is used for the wing/fuselage/nacelle to enhance
the run time efficiency and hence reduce the dynamic memory requirement of the solver, but it also reduces the accuracy of the flow field around the bodies (i.e., wing, fuselage, nacelle) to a lower order. Since our primary goal of this case is not to perform accurate download calculations, the accuracy of the analysis in capturing the fountain flow effects is not degraded.

The aim of this user’s manual is to provide instructions on utilizing the ROTTILT solver for applications to the tiltrotor aircraft configuration in hover. There are several preprocessors developed as stand-alone programs. Their utilization together with a public domain graphic package GNUPLOT is used to generate an accurate surface mesh which is required as input to the ROTTILT solver. Listings of these programs are given in the appendices. Step-by-Step procedure on input preparation to the ROTTILT for computational grid as well as ROTTILT’s general input files are documented in this report. To decrease the complexity of the procedure for running the solver accurately, schematic flow charts are provided as a guide to generate input files. For completeness, formulations used in the ROTTILT are presented in full. However, this user manual is intended for the users who are familiar with CFD flow analyses pertaining to rotor-body flow interferences. Understanding the steps associated with the ROTTILT solver is vital for grid generation and input file preparation where many control flags are set to default values and should not be changed without a full understanding of the changes. Program listings of the post-processors developed for coupling of the ROTTILT with WOPWOP [Ref. 3] program are also included in the appendices of this report. Since the rotor mean downwash velocities are employed for turbulence ingestion noise prediction, a control flag for extracting the rotor inflow information at user specified planes is also added to the solver. As a basis for the validation of the solver, the results of the computed flow field and the associated aerodynamic and acoustic characteristics for the V22 and XV15 aircrafts respectively are presented.

2 Flow Field Solution

With the assumptions that the fluid density and viscosity are constant, the governing equations are based on steady, incompressible, laminar Navier-Stokes formulations. In the Cartesian coordinates system, the governing equations (i.e. continuity and momentum equations) can be written as:

Continuity:

\[
\frac{\partial u}{\partial X} + \frac{\partial v}{\partial Y} + \frac{\partial w}{\partial Z} = 0
\]  

(1)

X momentum :

\[
\rho \left( u \frac{\partial u}{\partial X} + v \frac{\partial u}{\partial Y} + w \frac{\partial u}{\partial Z} \right) = \mu \left( \frac{\partial^2 u}{\partial X^2} + \frac{\partial^2 u}{\partial Y^2} + \frac{\partial^2 u}{\partial Z^2} \right) - \frac{\partial p}{\partial X} + S'_X
\]

(2)

Y momentum :

\[
\rho \left( u \frac{\partial v}{\partial X} + v \frac{\partial v}{\partial Y} + w \frac{\partial v}{\partial Z} \right) = \mu \left( \frac{\partial^2 v}{\partial X^2} + \frac{\partial^2 v}{\partial Y^2} + \frac{\partial^2 v}{\partial Z^2} \right) - \frac{\partial p}{\partial Y} + S'_Y
\]

(3)

Z momentum :

\[
\rho \left( u \frac{\partial w}{\partial X} + v \frac{\partial w}{\partial Y} + w \frac{\partial w}{\partial Z} \right) = \mu \left( \frac{\partial^2 w}{\partial X^2} + \frac{\partial^2 w}{\partial Y^2} + \frac{\partial^2 w}{\partial Z^2} \right) - \frac{\partial p}{\partial Z} + S'_Z
\]

(4)

The above equations are solved using a finite-volume based method known as SIMPLER. Detailed information regarding the the principles of SIMPLER technique are given in Ref. 4.

The source terms \(S'_X, S'_Y, S'_Z\) added to the momentum equations are due to the rotor induced forces per unit volume acting at the cells which are encompassing the rotor plane. It is through these terms that the rotor’s influence is introduced into the flow field. In effect, the rotor is represented as a a distribution of momentum sources acting in the rotor designated computational plane.
3 Rotor Modeling

In determining where the rotor's influence on the flow field is felt, the locations of the rotor physical plane in conjunction with the corresponding momentum equation source terms should be determined accurately. Therefore, the description of the rotor geometry is required (i.e. radius, chord, hub radius, etc.). To model the rotor in a Cartesian computational grid topology, four coordinate systems are employed. A brief description of each coordinate system as well as expressions for mutual transformations are presented in the following sections.

3.1 Coordinates of the computational domain

The governing equations are solved in the global $(X,Y,Z)$ Cartesian coordinate system. $\hat{I}$, $\hat{J}$ and $\hat{K}$ are the unit vectors in the coordinate system. The center of the rotor is at $(X_c, Y_c, Z_c)$ with respect to this system and its axis of rotation is along the vector $\vec{\omega}$ where

$$\vec{\Omega} = \Omega_1 \hat{I} + \Omega_2 \hat{J} + \Omega_3 \hat{K} \quad (5)$$

and $|\vec{\Omega}| = \Omega$, the rotational speed in radians per second.

![Figure 2: Schematic of Rotor based Cartesian System.](image)

3.2 Rotor based Cartesian system

It is convenient to have the computational coordinates in the direction parallel and normal to the freestream velocity. The rotor blade orientation is arbitrary with reference to the freestream. Therefore a Cartesian coordinate system $(\xi, \eta, \zeta)$ which has its origin at the center of the rotor and the $\xi$ axis in the direction opposite to the rotational velocity $\vec{\Omega}$ has been defined. As shown in Fig. 2, the $\xi$ axis is perpendicular to the plane of rotation while the $\eta$ and $\zeta$ axes lie in the plane. Euler angle method is employed to establish a relation between this system and the computational coordinate system. This results in an orthogonal transformation. Using this transformation which includes the shift of origin, the transformation from the rotor-based to the computational coordinates can be written as
\[
\begin{bmatrix}
\xi \\
\eta \\
\zeta
\end{bmatrix} = 
\begin{bmatrix}
\cos B & \sin A \sin B & -\cos A \sin B \\
0 & \cos A & \sin A \\
\sin B \sin A & -\sin A \cos B & \cos A \cos B
\end{bmatrix}
\begin{bmatrix}
X - X_c \\
Y - Y_c \\
Z - Z_c
\end{bmatrix}
= M_1 \begin{bmatrix}
X - X_c \\
Y - Y_c \\
Z - Z_c
\end{bmatrix}
\tag{6a}
\]

where \( A \) and \( B \) are two angles which describe the orientation of the rotor with respect to the computational coordinate system Ref [5].

A useful attribute associated with orthogonal transformations is that the inverse of the transformation matrix is its transpose. Thus we can write the inverse transformation from \((\xi, \eta, \zeta)\) to \((X, Y, Z)\) as

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} = \begin{bmatrix}
X_c \\
Y_c \\
Z_c
\end{bmatrix} + M_1^T \begin{bmatrix}
\xi \\
\eta \\
\zeta
\end{bmatrix}
\tag{6b}
\]

The unit vectors in the two systems are also related by the matrix \( M_1 \).

Figure 3: A Schematic of Rotor Based Cylindrical System

Figure 4: Schematic of Rotor based Cartesian System.
3.3 Rotor based cylindrical polar system

Further a cylindrical coordinate system \((r, \phi, z)\) as shown in Figs. 3 and 4 is defined which provides the necessary transformation from the Cartesian system to the blade coordinate system. The unit vectors in this system are related to those in the \((\xi, \eta, \zeta)\) system by the following matrix relation.

\[
\begin{bmatrix}
\hat{e}_r \\
\hat{e}_\phi \\
\hat{e}_z
\end{bmatrix} = \begin{bmatrix}
0 & \cos \phi & \sin \phi \\
0 & -\sin \phi & \cos \phi \\
1 & 0 & 0
\end{bmatrix} \begin{bmatrix}
\hat{e}_\xi \\
\hat{e}_\eta \\
\hat{e}_\zeta
\end{bmatrix} = M_2 \begin{bmatrix}
\hat{e}_\xi \\
\hat{e}_\eta \\
\hat{e}_\zeta
\end{bmatrix}.
\]

(7a)

The inverse relation can once again be obtained by taking the transpose of \(M_2\) such that

\[
\begin{bmatrix}
\hat{e}_\xi \\
\hat{e}_\eta \\
\hat{e}_\zeta
\end{bmatrix} = M_2^T \begin{bmatrix}
\hat{e}_r \\
\hat{e}_\phi \\
\hat{e}_z
\end{bmatrix}
\]

(7b)

3.4 Coordinate system for blade deflection

To model elastic blade deformation, an additional coordinate system is required \((n, \phi, s)\), where \(s\) is in the spanwise direction of the blade (i.e., \(s\) is the location of the aerodynamic center of the airfoil sections). A line sketch in Fig. 5 depicts a curved blade. The direction \(\hat{e}_\phi\) is the same as in the previous system and \(\hat{e}_n\) is defined to complete the right handed system. Thus the \((n, s)\) axes always lie in the \(r-z\) plane and, when \(\phi = 0\), the \(n\) axis opposes \(z\) while the \(s\) axis coincides with \(r\). The transformation between this and the cylindrical system can be written as

\[
\begin{bmatrix}
\hat{e}_n \\
\hat{e}_\phi \\
\hat{e}_s
\end{bmatrix} = \begin{bmatrix}
\sin \delta & 0 & -\cos \delta \\
0 & 1 & 0 \\
\cos \delta & 0 & \sin \delta
\end{bmatrix} \begin{bmatrix}
\hat{e}_r \\
\hat{e}_\phi \\
\hat{e}_z
\end{bmatrix} = M_3 \begin{bmatrix}
\hat{e}_r \\
\hat{e}_\phi \\
\hat{e}_z
\end{bmatrix}
\]

(8a)

and inversely

\[
\begin{bmatrix}
\hat{e}_r \\
\hat{e}_\phi \\
\hat{e}_z
\end{bmatrix} = M_3^T \begin{bmatrix}
\hat{e}_n \\
\hat{e}_\phi \\
\hat{e}_s
\end{bmatrix}
\]

(8b)

Given the distribution of the deflection along the blade span, the following equation of the curved blade can be easily derived

\[
\vec{R}(s) = \hat{e}_r \int_0^s \cos \delta(s) \, ds + \hat{e}_s \int_0^s \sin \delta(s) \, ds
\]

(9)

3.5 Rotor discretization

The rotor blades are discretized into spanwise elements. Blade properties such as chord length, out of plane deflection, twist, thickness and the airfoil section characteristics at the control point of each element are assumed to be constant across the length of the element. The control points for the blade segments should prescribe a circular path which is aligned with the blade aerodynamic center. Therefore, it is important
Figure 5: A Schematic of Rotor Based Cylindrical System

to locate the grid cells of the computational domain which are in the path of the rotor circular motion. Since the three-dimensional computational grid is oriented arbitrarily with respect to this circle, a general algorithm has been developed for this purpose. Details can be found in Ref. 6.

3.6 Calculation of rotor forces

Let the fluid velocity at any point $s$ on the blade element at an angular position $\phi$ be

$$\vec{V} = u\hat{I} + v\hat{J} + w\hat{K}. \tag{10}$$

Using equations (6a), (7a) and (8a) the same can be written in the $(n, \phi, s)$ system as

$$\vec{V} = v_n \hat{e}_n + v_\phi \hat{e}_\phi + v_s \hat{e}_s \tag{11}$$

where

$$\begin{bmatrix} v_n \\ v_\phi \\ v_s \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \cdot \begin{bmatrix} M_3 \\ M_2 \\ M_1 \end{bmatrix}.$$

The blade has a velocity due to its rotation which can be written in the $(n, \phi, s)$ system as

$$\vec{V}_{bl} = M_3 \left( M_2 M_1 \vec{\Omega} \times \vec{R}(s) \right) \tag{12}$$

where $\vec{\Omega}$ is defined in Equation (5) and $\vec{R}(s)$ is the position vector of the point on the blade under consideration. Hence the flow velocity relative to the blade, $\vec{V}_{rel} = v_n \hat{e}_n + v_\phi \hat{e}_\phi + v_s \hat{e}_s$ is given by

$$\vec{V}_{rel} = \vec{V} - \vec{V}_{bl} = M_3 M_2 M_1 \vec{V} - M_3 \left( M_2 M_1 \vec{\Omega} \times \vec{R}(s) \right) \tag{13}$$

In order to determine the aerodynamic forces on the airfoil section at $s$, we need only the component of $\vec{V}_{rel}$ in the plane normal to $\hat{e}_s$. The angle made by this component with the $\hat{e}_\phi$ direction is given by (see Fig. 6)

$$\beta = \arctan\left( -\frac{v_n'}{v_\phi'} \right) \tag{14}$$

If the blade has an angle of twist $\theta_s$ with respect to the plane of rotation then, from Fig. 6, the effective angle of attack seen by 2D airfoil section is

$$\alpha = \theta_s - \beta. \tag{15}$$
Figure 6: A Schematic of Aerodynamic Forces on the Airfoil Section at "s"

Knowing the angle of attack $\alpha$ and the components of the relative flow velocity experienced by the airfoil section, we can find the sectional aerodynamic force coefficients $C_l$ and $C_d$ from the classical 2D blade element formulations. The lift and drag forces on the blade element of length $ds$ can then be found from

\[
L' = \frac{1}{2} \rho V_n^2 C_{l} c ds \tag{16a}
\]

\[
D' = \frac{1}{2} \rho V_n^2 C_{d} c ds \tag{16b}
\]

where $c$ is the blade chord-length and $V_n^2 = V_n'^2 + V_\varphi'^2$. The lift and drag forces act perpendicular and parallel, respectively, to the relative velocity vector. Resolving these forces in the $e_\varphi$ and $e_n$ directions, we have

\[
f_n = L' \cos \beta - D' \sin \beta \tag{17a}
\]

\[
f_\varphi = L' \sin \beta + D' \cos \beta \tag{17b}
\]

Also, since there are no aerodynamic forces along the span,

\[
f_s = 0.
\]

Thus the resultant aerodynamic force on the blade segment, $\vec{f} = (f_n, f_\varphi, f_s)$, can be computed for the $(n, \varphi, s)$ system. The corresponding force vector in the $(X, Y, Z)$ system, $\vec{F}$, can be found by using the inverse transformation relations (8b), (7b) and (6b) as

\[
\vec{F} = M_1^T M_2^T M_3^T \vec{f}. \tag{18}
\]

The instantaneous force acting on the fluid element at the $(s, \varphi)$ location is, then, $-\vec{F}$. Since the blade actually spends a finite fraction of its total revolution time passing through this control volume, the time averaged source terms $\vec{S} = (S_X, S_Y, S_Z)$ should be added to the discretized momentum conservation equations at the control volume which is the result of $-\vec{F}$ multiplied by its fractional time value, i.e.,

\[
\vec{S} = \frac{b \Delta \varphi}{2\pi} (-\vec{F}) \tag{19}
\]

where $b$ is the number of blades and $\Delta \varphi$ is the angular distance through which the blade traverses in passing through the specific control volume.
4 ROTTILT Program Architecture

In order to eliminate any computational errors associated with the coupling of the ROTTILT solver with external rotor trim program such as CAMRAD, trim loop has been added to the solver. For the user’s specified Ct/sigma, the collective pitch angle is iteratively perturbed. As shown in Fig.(7), a final flow solution for any given tiltrotor configuration is obtained when both the flowfield and rotor trim are converged simultaneously. The rotor trim condition is controlled by the user’s input variables which are furnished in the rotor trim namelist section of solver main input file. Details regarding their definitions are given in the following sections. Integration of the rotor trim loop into the ROTTILT has increased the run CPU time of the solver by only 5 percent which is negligible compared with the overall run time of the solver. This is due to the implementation of the trim loop as a sub-iteration to the solver main flow field iteration loop, (see Fig. (7)).

5 Aircraft Body Components Section Cuts Generation

Conventionally, the aircraft body geometry is assumed to be obtained by the user from a CADDY system (e.g. Boeing’s UG system). A translator is required to convert the CADDY’s design part into a surface mesh format such as HESS which is ascii and it is generally considered to be a standard CFD file format. A sample file containing the HESS format is given in Appendix (I). Using the HESS file containing the surface mesh geometry coordinates for the user specified components (i.e. fuselage, wing, and nacelle), a preprocessor SPLINE can be employed to generate the section cuts which are required as input to the ROTTILT grid generation program. As an option, SPLINE program provides equally the same number of graphic files for the user checks on the section cuts computed by the SPLINE program. Public domain graphic program GNUPLOT, which is 3D and interactive is then used to visualize the cuts before proceeding to the ROTTILT grid generation state. As a successful completion of the geometry manipulation process, one file per component is generated with the component’s name and an extension " .dat " (e.g. Wing.dat, Fuselage.dat, etc). These files are ascii and hence they are not machine dependent which make them easily portable to any computer platform. Overall architecture of the process flow chart is shown in Fig. (8). Also, a typical 3D plot of the XV15 semi-span section cuts generated from this process coupled with the GNUPLOT graphic can be seen in Fig. 9. Using GNUPLOT graphic control commands the displayed aircraft configuration can be scrutinized for quality of the section cuts.

6 Grid Generation

The Grid generation program lays out the computational grid over a tiltrotor’s various components. The user needs to specify the information related to the surface geometries of the bodies in a data file. Input file format for tiltrotor bodies such as Rotor, Wing, Nacelle and Fuselage components are fully described in this documentation. It important to mention that the input lines that begin with the " # " (Number Sign ) as the first character indicates a comment statement (i.e. inactive input).

The input file provides the following control parameters which are required as input to the Grid Generation program:

- Executive control parameters for the grid generation
- Rotor and its Dimensions
- Rotor center location with respect to user specified common axis employed for all other components.
- Number of grid points to be generated on the rotor disk
 Existence of the aircraft components like Wing, Nacelle(s) and Fuselage.

For each one of the existing bodies
  – Component name file containing the body points.
  – Number of grid cells required on the body (in I and J directions)
    The grid generation program will use the user specified cell density as a guidance to generate the actual computational grid.
  – The extent of the bounding box for each component of the aircraft.

User specified grid layer file name containing the grid density specification.

6.1 ROTTILT Internal Grid Generation and Quality assessment

Since Computational grid generation is an integral part of the ROTTILT solver, it is advisable to ensure the quality of the grid in terms of its density and distribution per component following the process depicted in the Fig. (10). Therefore, with setting the flag “StpAfrtGD” to “true”, the computational grids generated as output for each aircraft body component are written separately into their respective part names appended with an extension of “.out” in a 2D format. Postprocessing program Conv.f can be used to convert the grid files into graphic postscript format which can then be visualized using “GOSTVIE”, as shown in Fig. (11). Three view angles of the computational grid is provided to the user namely, XY, XZ, and YZ planes. If the grid distribution is not acceptable as a whole or partially, after adjusting the grid control parameters in the “.lay” file, a new set of grids can then be obtained through an iterative process with ROTTILT before initiating the solver. As the flow field solution around the rotor domain is of primary importance here, the user should ensure that the grid distribution in the rotor region is uniform as shown in Fig.(11). Abrute force increase in the grid density will naturally increases the CPU run time significantly. User experience with the solver is therefore essential in balancing the solution accuracy versus grid density for the ROTTILT solver. Furthermore, the ROTTILT generates an output grid file in FAST format, as shown in Fig. (12). The various grid block for each aircraft components with their density are clearly depicted. It is important to mention that the computational grid for each block overlaps the neighboring block thus creating regions of high density grid. With the user’s experience, the fringe regions between the blocks can be tailored in such a fashion to improve the grid quality in the rotor domain without increasing the total number of grid points.

7 Namelist File CNTRLPARMS

Control parameters are provided as Namelist parameters and they begin with $CNTRLPARMS and ends with $END. The keywords $CNTRLPARMS and $END must start at the second character in the line.

In Table [1], the actual namelist parameters are specified between the $CNTRLPARMS and $END. Each one of the parameters need to be given on a separate line. If any of the parameters is not provided then the default values would be used.

In Table [2], the Bounding Box Adjustment may be done beyond the bounding box of all bodies in the I, J, and K directions. Setting to TRUE will allow the grid generation program to have no constraint on extending the boundaries of bounding boxes beyond the specified maximum and minimum of the box. For example, the grid over the wing should not extend beyond Y-min and hence AdjstByndYL should be FALSE.
Table 1: List of Namelist Parameters in CNTRLPARMS

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Value</th>
<th>Description (Meaning of TRUE)</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBGlvl</td>
<td>Integer [1 to 9]</td>
<td>Print the values of fields</td>
<td>0</td>
</tr>
<tr>
<td>DBGpltLvl</td>
<td>Integer</td>
<td>Used in debugging</td>
<td>0</td>
</tr>
<tr>
<td>FrstNacl</td>
<td>TRUE/False</td>
<td>First Nacelle</td>
<td>False</td>
</tr>
<tr>
<td>ScndNacl</td>
<td>TRUE/False</td>
<td>Second Nacelle</td>
<td>False</td>
</tr>
<tr>
<td>WingBody</td>
<td>TRUE/False</td>
<td>Wing</td>
<td>False</td>
</tr>
<tr>
<td>Fuselage</td>
<td>TRUE/False</td>
<td>Fuselage</td>
<td>False</td>
</tr>
<tr>
<td>ScndRtr</td>
<td>TRUE/False</td>
<td>Second Rotor</td>
<td>False</td>
</tr>
<tr>
<td>FullSpan</td>
<td>TRUE/False</td>
<td>Full span</td>
<td>False</td>
</tr>
<tr>
<td>StpAftrGd</td>
<td>TRUE/False</td>
<td>Stop execution after Grid Generation</td>
<td>False</td>
</tr>
<tr>
<td>BothWings</td>
<td>TRUE/False</td>
<td>Both Wings exist</td>
<td>False</td>
</tr>
<tr>
<td>ExtndWing</td>
<td>TRUE/False</td>
<td>Extend Wing to the center of the Fuselage</td>
<td>False</td>
</tr>
</tbody>
</table>

(Useful when there is no fuselage)
Figure 7: ROTTILT Solver Coupled with the Rotor(s) Trim Loop Flow Chart
Figure 8: ROTTILT Input Geometry Preparation Process

Figure 9: 3D View of the Section Cuts for the Semi-Span XV15 Aircraft

Figure 10: ROTTILT Internal Grid Generation Process
Figure 11: XYZ Plane View of the Computational Grid of a Full-Span XV15 Aircraft
Figure 12: XV15 Semi-Span Configuration in ROTTILT Computational Grid Domain
Table 2: List of Namelist Parameters in AdjustBox

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Value</th>
<th>Description (Meaning of TRUE)</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdjstByndXL</td>
<td>TRUE/False</td>
<td>Low value in X direction</td>
<td>TRUE</td>
</tr>
<tr>
<td>AdjstByndXH</td>
<td>TRUE/False</td>
<td>High value in X direction</td>
<td>TRUE</td>
</tr>
<tr>
<td>AdjstByndYL</td>
<td>TRUE/False</td>
<td>Low value in Y direction</td>
<td>TRUE</td>
</tr>
<tr>
<td>AdjstByndYH</td>
<td>TRUE/False</td>
<td>High value in Y direction</td>
<td>TRUE</td>
</tr>
<tr>
<td>AdjstByndZL</td>
<td>TRUE/False</td>
<td>Low value in Z direction</td>
<td>TRUE</td>
</tr>
<tr>
<td>AdjstByndZH</td>
<td>TRUE/False</td>
<td>High value in Z direction</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

A typical example for the namelist file ”CNTRLPARMS” is given below:

Example:

$CNTRLPARMS
DBGlvl = 1
DBGpltLvl = 1
WingBody = .TRUE.
FrstNacl = .TRUE.
Fuselage = .TRUE.
AdjstByndYL = .FALSE.
SEND

7.1 Rotor Related inputs

In a multi rotor configuration, the rotor information is provided through the first rotor input parameters only. For example, in the case of a dual rotor configuration, the second rotor is simply treated as a mirror image to the first rotor. The following sections explain the details of the input specifications for the rotor.

7.1.1 First Rotor Dimensions and Location

Example:

#  
# Rotor Radius, Half Thickness , Rotor Center (X,Y,Z) Coordinate  
12.5, 0.3, 26.8478, 17.227, 18.400  
#  

Explanation:

Rotor Radius in feet  
Half Thickness (i.e half of the height of the computational box which is encompassing the rotor) is specified as a ratio of the Rotor Radius  
(i.e actual Half thickness (half Z-width ) by Rotor Radius)
This info is used only for grid generation (to specify # of grids in K dir)
X coordinate for Rotor Hub Center
Y coordinate for Rotor Hub Center
Z coordinate for Rotor Hub Center

**First Rotor - No of Grids**
The approximate number of grid points required on the First Rotor in I, J, K directions respectively need to be provided as follows:

Example:

```
# Rotor I-Grids, J-Grids and K-Grids
30, 14, 2
#
```

**First Rotor - Bounding Box Adjustment**
The bounding box for the Rotor is initially set up by adding and subtracting the rotor radius from the three coordinates of the Rotor center. The size of this bounding box can be adjusted, using similar procedure and convention as the body bounding box adjustments explained previously.

Bounding Box Adjustments for rotor are specified as a ratio of the rotor diameter.
So if the user wants the bounding box to be

- extended then a Positive Ratio needs to be set
- reduced then a Negative Ratio needs to be set

**7.1.2 Second Rotor**

If the Second Rotor exists then all the above inputs specified for the First Rotor need to be given for the Second rotor also.

**7.2 Inputs Related to Other Bodies**
The other bodies that may be defined are Wing, Nacelles and Fuselage. This chapter explains the user specifications for the grid over these bodies.

**7.2.1 Wing**

If wing exists (WingBody = TRUE) then the wing data is given as follows

_Wing - Body Points File_
The Wing points are specified in a separate file and the name of the file is provided as follows:

Example:

```
# Wing File Name (File Containing Wing Points)
'BdyPts/Wing.dat'
#
```

**Wing - No of Grids**

The approximate number of grid cells required on the Wing in I, J, K directions respectively need to be provided as follows:

Example:

```
#
# Number of I-Grids, J-Grids and K-Grids for Wing
 12, 12, 12
#
```

**Wing - Bounding Box Adjustment**

The Adjustment Ratios for the Bounding Box around the Wing follows the same notation and convention as for the bounding box adjustments for the body and rotor explained previously.

Example:

```
#
# Adjust Wing BBox (MinX, MaxX, MinY, MaxY, MinZ, MaxZ) by Ratios
 0.06, 0.18, -0.045136, -0.02639252, 0.45, 0.27
#
```

7.2.2 First Nacelle

If First Nacelle exists (FrstNacl = TRUE ) then the First Nacelle data is given as follows

**First Nacelle - Body Points File**

The Nacelle points are specified in a separate file and the name of the file is specified as follows

Example:

```
#
# Nacelle File Name (File Containing Nacelle Points)
```
'#BdyPts/Nacelle.dat' (with "BdyPts" being the path)

#

**First Nacelle - No of Grids**

The approximate number of grid cells required on the First Nacelle in I, J, K directions respectively need to be provided as follows:

**Example:**

```plaintext
# Number of I-Grids, J-Grids and K-Grids for Nacelle

1, 7, 14

#
```

**First Nacelle - Bounding Box Adjustment**

The Adjustment Ratios for the Bounding Box around the Nacelle are specified as per the notation explained in Section 7.3;
**Example:**

```plaintext
# Adjust Nacelle-1 BBox (MinX, MaxX, MinY, MaxY, MinZ, MaxZ)
# by Ratios (before Grid Gen. on Bodys):
  0.0, 0.0, 0.0, 0.0, -0.45, -0.13
```

### 7.2.3 Second Nacelle

If Second Nacelle exists (ScndNacl = TRUE) then the Second Nacelle data is given just like the First Nacelle.

### 7.2.4 Fuselage

If Fuselage exists (Fuselage = TRUE) then the Fuselage data is given as follows:

**Fuselage - Body Points File**

The Fuselage points are specified in a separate file and the name of file is specified as follows:

**Example:**

```plaintext
#
# Fuselage File Name (File Containing Fuselage Points),
# Translated Fuselage Pts. Out File Name
  'BdyPts/Fuselage.dat'
#
```

**Fuselage - No of Grids**

The approximate number of grid cells required on the Fuselage in I, J, K directions respectively need to be provided as follows:

**Example:**

```plaintext
# Number of I-Grids, J-Grids and K-Grids for Fuselage
  20, 5, 11
```
Example:

```
# Adjust Fuselage BBox (MinX, MaxX, MinY, MaxY, MinZ, MaxZ)
# by Ratios (before Grid Gen. on Bodys):
   0.15, 0.15, 0.0, 0.0, 0.0, -0.37
```

7.3  **Bounding Box Adjustment Notation**

The bounding box for a body is initially set based on the geometry of the body. A bounding box bounds the body completely. Adjusting the bounding box allows a clearance between the geometry and the surrounding box.

This Bounding Box may be adjusted using the input ratios provided by the user. The adjustment in a direction is provided as a ratio of the total dimension in that direction.

Example:

```
# Adjust Rotor-1 BBox (MinX, MaxX, MinY, MaxY, MinZ, MaxZ) by Ratios
   0.15, 0.15, -0.5662888, 0.15, -0.82, 0.07
```

The above 6 entries correspond to Bounding Box Adjustment Ratio for the following:

- X minimum
- X maximum
- Y minimum
- Y maximum
- Z minimum
- Z maximum

In the above example, the original bounding box at the starting X location (X minimum) is enlarged by the size of 0.15 times the total dimension of the body. A negative ratio simply means that the original bounding box is reduced instead of being enlarged.

So if the user wants the bounding box to be

- extended, then a Positive Ratio needs to be set
- reduced, then a Negative Ratio needs to be set
The grid specification of the total computational domain follows a three-block arrangement, which are designated as Block#1, Block#2, and Block#3, as given in Figure (13). Block#2 contains the rotor and the bodies, and its grid specification is given automatically by the program. Grids outside the rotor-body block up to the boundaries of the computational domain are specified by the user, by providing the layer information of Block#1 and Block#3 in the three coordinate directions. The grid generation program needs this layer information which is specified in a separate file.

Figure 13: Three-block grid specification.

In the current input file, the name of the layer information file is specified as follows:

Example:

```
# Layer Information File
   'Layer.dat'
#
```

The layer information for Block#1 and Block#3 in the “Layer.dat” file must follow the following format:

Example:

```
# Block#1
# Layer Definition for Grid before body (from South-West-Bottom to Body Start)
# #
# # NO OF LAYERS (X-grid)
   2
# # LAYER# RLEN CLEN NCEL RATX
#   1    14.0  12.5  7  -1.65
   2    1.0  12.5  4  -1.05
#```

21
# NO OF LAYERS (Y-grid)
0

# LAYER# RLEN CLEN NCEL RATX
#
#
# NO OF LAYERS (Z-grid)
4

# LAYER# RLEN CLEN NCEL RATX
#
1 24.0 12.5 4 -1.3
2 5.92 12.5 4 -1.4
3 1.0 12.5 3 -1.9
4 2.0 1.0 3 -1.2

# # Block#3
#
# Definition for Grid after body (from Body End to North-East-Top)
#
# NO OF LAYERS (X-grid)
2

# LAYER # RLEN CLEN NCEL RATX
#
1 1.0 12.5 4 1.05
2 14.0 12.5 7 1.65

# # NO OF LAYERS (Y-grid)
2

# LAYER # RLEN CLEN NCEL RATX
#
1 1.0 12.5 -5 1.15
2 14.0 12.5 -7 1.65

# # NO OF LAYERS (Z-grid)
2

# LAYER # RLEN CLEN NCEL RATX
#
1 1.0 12.5 3 1.85
2 9.0 12.5 4 1.6

The parameters in the layer information file are explained in the following table:
Table 3: List of Namelist Parameters in Layer Control

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO OF LAYERS</td>
<td>Integer</td>
<td>Number of layer in a particular coordinate direction</td>
</tr>
<tr>
<td>LAYER</td>
<td>Integer</td>
<td>Layer number</td>
</tr>
<tr>
<td>RLEN</td>
<td>Real</td>
<td>Specified length</td>
</tr>
<tr>
<td>CLEN</td>
<td>Real</td>
<td>Characteristic length</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RLEN times CLEN gives the true length</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CLEN=1 means RLEN is the true length</td>
</tr>
<tr>
<td>NCEL</td>
<td>Integer</td>
<td>Number of grid cell</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Negative number means equally spaced grid</td>
</tr>
<tr>
<td>RATX</td>
<td>Real</td>
<td>Negative value means that the next cell width ($CW_{i+1}$) is smaller than the previous value ($CW_i$) and the ratio $CW_i/CW_{i+1} = 1/\text{ABS}(\text{RATX})$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Positive value means that the next cell width is greater than the previous value and the ratio $CW_{i+1}/CW_i = \text{ABS}(\text{RATX})$</td>
</tr>
</tbody>
</table>
9 Coupling of WOPWOP/TIN2 and ROTTILT

The acoustic analysis of the tiltrotor in hover is performed using the acoustic prediction code WOPWOP [3]. Presently, the noise sources associated with the fountain flow effects are classified into the tonal noise (i.e, deterministic) as well as the low frequency broadband noise (non-deterministic), as discussed in Ref [9]. Current version of the ROTTILT computed high resolution airloads and mean rotor inflow downwash which are employed as input to WOPWOP and TIN2 programs. No comparative studies have yet been made to determine their relative contributions to the overall noise in the presence of fountain flow effects. However, it is important to mention that an accurate low frequency noise computation resulting from the fountain flow is strongly dependent on the partial inflow turbulence induced by the flow recirculation. The prediction of the noise sources due to the inflow of turbulence is currently beyond the scope of the CFD solver presented here, but the utilization of a newly developed empirical formulation(s) based on well defined experimental data for the turbulence characteristics associated with this phenomena would be an attractive candidate for low frequency broadband noise analysis for tiltrotor in hover. Further, the TIN2 program inflow distortion matrix employs the rotor inflow information over the whole rotor plane which can be computed using ROTTILT solver. For WOPWOP, the high resolution airloads (e.g. per 0.5 degs) computed by ROTTILT solver are used for the noise analysis A postprocessing code WOPPOST (Appendix E) is used to convert the computed ROTTILT airloads into a suitable format for the WOPWOP. Compact version of the WOPWOP program has been utilized. The accuracy of the computed high resolution airloads (order of 0.5 degs.) required for prediction of impulsive noise is closely examined in terms of the grid clustering in the vicinity of proprotor plane. Fig. 14 depicts a flow chart for the process used in coupling the ROTTILT with the WOPWOP and TIN2 programs.

10 Validations

In order to accurately model the rotor in the ROTTILT-v2 solver, we have conducted a number of computational runs for the isolated XV15 proprotor. Aircraft components such as wing, fuselage, and nacelle were removed from the computational domain before performing additional rotor grid refinements. We have computed the figure-of-merit for the XV15 isolated rotor for a range of $C_{t}/\sigma$ from 0.06 to 0.17 which covers the full spectrum of the measured data presented in the reference [10]. Figure (15) depicts the correlation with the measured data which is in very good agreement over a wide range of flight conditions. As shown, an important feature associated with the computed results is the capturing of the rotor stall characteristics observed in the experimental data. The study made here has a significant value in establishing the computational accuracy of the solver before the inclusion of more aircraft components where an accurate assessment of the computed flow could be much more cumbersome.
To better understand and evaluate the predictions, the XV15 full-span surface geometry and the flow field are graphically represented. Because of the lack of detailed flow measurement for XV15 aircraft, no attempt is made to justify the accuracy of the computed flow features captured in our CFD simulation of the fountain flow effects. However, it is encouraging to observe that the fountain flow feature associated with the tiltrotor is physical.

Figs. (16 and 17) show the flow field computed by ROTTILT in a constant x-plane which intersects the nacelle and the wing laterally. As shown, the proprotor wake flow features are analyzed using velocity traces (Fig. 16) and velocity vector field (Fig. 17). The fountain flow region on the rotor retreating side (above the wing ) has been captured and its extent is clearly defined. Figure 16 shows three distinct regions which can listed as: 1) fountain flow region on the rotor retreating side over the wing upper surface area; 2) typical wake contraction region on the rotor advancing side (clean side); and 3) a stagnated flow region below the wing near the wing fuselage juncture. Furthermore, the rotor downwash flow over the wing region has been split into two distinguishable domains with most of the flow from the wing midspan to the fuselage centerline entrained into the fountain flow region. Slight asymmetry associated with the recirculation region of the fountain between the two rotors is attributed to disparity of the rotor trim forces computed for each proprotor by the ROTTILT solver. A difference of 4 lbs in rotor thrust is obtained from the flow solution for the XV15 proprotor trim analysis after 1200 iterations.

In order to investigate the effects of the fountain flow on the rotor performance, we have presented the results for the rotor inflow velocity variation in terms of rotor azimuth, as shown in Fig (18). In the comparison, the rotor mean inflow velocities for the XV15 isolated rotor case where the aircraft components, fuselage, wing, and nacelle have been removed from the computation domain are compared with the corresponding results from the semi-span XV15 aircraft. Two distinct trends for the inflow velocity are noticeable for the cases considered here where a significant deviation from the mean values is computed in the fountain.
flow region (between $\psi = 230$ degs. and $\psi = 300$ degs.). Consequently, the computed rotor lift as a function of radial stations is affected, as shown in Fig (19). Most importantly, from aerodynamic and acoustic perspectives, the blade airloads have been altered near the tip region for the rotor blade, for example, at $\psi = 100$ degs. versus $\psi = 280$ degs.

Bell’s hover/inflow turbulence data is employed for ROTTILT inflow validation. Their inflow measurement was made at 7 ins above the rotor on the advancing side (no fountain flow) and the retreating side (above the wing region). Bell’s model rotor is a semi-span tiltrotor 15 percent scale of the V22 aircraft with JVX proprotor configuration. In order to correlate the ROTTILT mean inflow computation with that of the measured data the solver with the JVX rotor and semi-span aircraft components was run for up to 1200 flow iterations. The data at the inflow plane corresponding to 7 ins above the rotor plane were extracted from the flow computed by ROTTILT. Figure (20) depicts the comparison of the computed inflow with the measured data. Only the w-component (axial) is presented here since it is considered to be more important parameter than the other two inflow components (namely u and v) to the accuracy of the noise prediction. As shown, most of the salient features of the measured rotor inflow have been captured in the prediction. For example, in the fountain flow region, the computations show a sharp radial variation of the inflow near the tip region changing from a negative value (upwash) outboard of the tip to a large positive (downwash) value which remains virtually constant over most of of the blade span, similar to features observed in the experimental data, see Figure (1). However, the intensity of the dowwash regions ahead and aft of the fountain regions are somewhat underpredicted. In addition, the radial extent of the inflow gradient is limited to a smaller region near the blade tip than that observed in the measured data. One plausible explanation is the lack of accurate modeling of the flow on the plane of symmetry in the absence of boundary turbulence boundary layer model in the ROTTILT solver. Overall, the correlation with the experimental data is considered to be encouraging with most of the important flow features over rotor plane captured in the flow field computational results.
Acoustic validation was performed using the flight test data for the XV15 aircraft which presented in Conner's AHS-RaeS Ref[12]. The correlation is made with acoustic time histories only. The XV15 acoustic characteristics are examined by considering the noise radiation pattern at two observer locations corresponding to $\phi = 45$ degress and $\phi = 135$ degress, see Figure (21). The noise characteristics of tiltrotor aircraft can be assessed accurately by these two observers. Moreover, from the flight test data, it is clear that the longitudinal variations of the acoustic characteristics associated with the XV15 aircraft are more dominant than their lateral variations.

Figure(21) depicts measured and predicted acoustic time histories corresponding to the two observer locations. Acoustic analysis was performed using WOPWOP program. The predicted results (microphone: $\phi = 45$ degress) contain negative peak of -20 dyne-per-sqcm amplitude which are not clearly detectable in the experimental data. These peaks are associated with the impulsive loading noise and not the thickness noise component due to the observer location. In the Figure (21), also depicted are the acoustic pressure results for the microphone located at $\phi = 135$ degress. Overall the comparison with the measured data for this microphone is considered to be encouraging. As shown, the general pulse width has been predicted correctly, whereas the peak-to-peak amplitude is under predicted by 50 percent. In summary, the CFD model ROTTILT coupled with the linearized acoustic model WOPWOP have been effective for aero/acoustic analysis of the tiltrotor in hover. The additional noise source associated with the fountain flow effects which
Figure 18: Azimuthal Variation of Rotor Mean Inflow for Full Span XV15 in Hover

is due to turbulence ingestion is not included in the computed results presented in this report.
Figure 19: Radial Distribution of Local Inflow Velocity for Full Span XV15 in Hover - ROTTILT Solution

Figure 20: Comparison of Rotor Mean Inflow Velocity for a 15 percent Scaled JVX Rotor
Figure 21: Hover Acoustic Pressures for XV15 Aircraft: Along its Longitudinal Axis
11 References


12 Appendices
A Sample Input File

An example input file is provided below, followed by the explanation for each input field-line.

Note: The comment lines are required in the input file.

```
 hxv15.dat

..DEBUG..ITEROP..ITERCL..LUNWR..LUNRS..LUNFL..LUNPL..LUNTB..LUNCT
 FALSE,  20 ,  180 ,  6 ,  8 ,  9 , 10 , 17 , 18

..DETAIL ... U ... V ... P ... W ... str Fun ... GAM ... RO ...
 TRUE, TRUE, TRUE, TRUE, TRUE, FALSE, FALSE, FALSE

..LROCON .. LMUCON .. LSOLID .. LDATPR .. LUGRID ...
 TRUE, TRUE, TRUE, FALSE, TRUE

..LGEOMP .. LUCOFP .. LVCOFP .. LPCOFP .. LTOCPP .. LPLOT .. LFILDP .. LPLT3D
 FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, TRUE, TRUE

..MAX..ITERATIONS .. ITEMOD .. NU .. NV .. NW .. NP .. NT
 200 ,  10 ,  2 ,  2 ,  2 ,  6 , 75

..RHO .. 0.002377  .. MU .. 3.719E-07  .. RELAX-X .. RELAX-Y .. RELAX-Z
 0.0023700 , 3.719E-07 , 0.05D0 , 0.05D0 , 0.05D0

..UINF .. VINF .. WINF .. PINF .. TINF .. REAL-GAS-CON .. GAMMA
 0.0d0 , 0.0D0, -1.0D0, 2116 , 419.0 , 1718.0D0 , 1.4D0

..LINI .. OUTI .. LINJ .. OUTJ .. Link .. OUTK .. LIP .. LJP .. LKP
 TRUE, TRUE, TRUE, TRUE, TRUE, FALSE, TRUE, TRUE, TRUE

..IPREF .. JPREF .. KPREF .. WINI .. WOUTI .. WINJ .. WOUTJ .. WINK .. WOUTK
 2 , 64 , 59 , FALSE, FALSE, FALSE, FALSE, FALSE, FALSE

..DRELXU .. DRELXV .. DRELXW .. DRELXT .. DRELXP .. MODREL .. RELMAX
 0.01 , 0.01 , 0.01 , 0.01 , 0.00 , 50 , 0.2

..LCSI .. LSRSEC .. LSMSEC .. LCLCD .. LCLCOR .. LROTBOD ..
 TRUE, FALSE, FALSE, TRUE, TRUE, FALSE

..Velocity check .. UCK .. JUCK .. KWCK .. XUCK .. YVCK .. ZVCK
 26 , 22 , 17 , 0.0 , 0.0 , 0.0

..Number of Rotor s .. DATA SET REFERENCE =14 NASA(TM X-952) .. IBOUND
 1 , 4

..Name of airfoil used ..
 MODIFIED NACA_0012

****************************************************************************** Data for rotor # 1 ******************************************************************************

..Airfoil .. # blades .. Press dist(r/R) ..
 1 , 3.0D0 , 0.250D0

Tip speed .. Rotor radius .. Hub radius (r/R) .. Hinge Offse (r/R) .. clock rot ..
```

Number Of Reference Rotor radius(r/R)max has been set to 45f........

Reference Rotor radius(r/R)---------------------------------------------
.100, .125, .150, .175, .200, .225, .240, .250, .260, .275, .290, .300, .310, .325, .340, .350, .375, .400, .425, .475, .500, .525, .575, .600, .625, .650, .675, .700, .750, .775, .800, .825, .850, .875, .900, .915, .925, .940, .950, .960, .970, .980, .990, .995

CL cor(r/R)_Ref Twist___# Blade Div___Azi cor___# azi loc___conv fac___
0.9D0 , -0.0D0 , 100 , 0.0D0 , 36 , 0.083333333
# harmonic pit___# harmonic flap__FLAPPING__flap geo corr
2 , 3 , FALSE , FALSE

-Harmonic_pit..coefficients-Positive series-
0, 13.35D0, 0.000D0
1, -0.5D0, 0.300D0
-Harmonic_flapping.coefficients-Positive series-
0 0.0000 0.0000
1 0.0000 0.0000
2 0.0000 0.0000

# data_points_______________________________________________________
21

nn___r/R___deflec___Chord/RAD___CL des___T/chrd___Twist___
1, .1353, 0.0D0, .13780, 0.0E0, 0.12E0, 29.500
2, .2035, 0.0D0, .13780, 0.0E0, 0.12E0, 25.500
3, .2700, 0.0D0, .13780, 0.0E0, 0.12E0, 20.750
4, .3250, 0.0D0, .13470, 0.0E0, 0.12E0, 18.625
5, .3750, 0.0D0, .13470, 0.0E0, 0.12E0, 15.875
6, .4250, 0.0D0, .13470, 0.0E0, 0.12E0, 13.500
7, .4750, 0.0D0, .13470, 0.0E0, 0.12E0, 11.250
8, .5250, 0.0D0, .13470, 0.0E0, 0.12E0, 9.000
9, .5750, 0.0D0, .13470, 0.0E0, 0.12E0, 6.875
10, .6250, 0.0D0, .13470, 0.0E0, 0.12E0, 4.875
11, .6750, 0.0D0, .13370, 0.0E0, 0.12E0, 2.813
12, .7250, 0.0D0, .12330, 0.0E0, 0.12E0, 0.944
13, .7750, 0.0D0, .10990, 0.0E0, 0.12E0, -0.833
14, .8200, 0.0D0, .10220, 0.0E0, 0.12E0, -2.354
15, .8600, 0.0D0, .09710, 0.0E0, 0.12E0, -3.771
16, .9000, 0.0D0, .09080, 0.0E0, 0.12E0, -5.216
17, .9350, 0.0D0, 0.08330, 0.0E0, 0.12E0, -6.500
18, .9600, 0.0D0, 0.07170, 0.0E0, 0.12E0, -7.438
19, .9750, 0.0D0, 0.05730, 0.0E0, 0.12E0, -8.000
20, .9850, 0.0D0, 0.04770, 0.0E0, 0.12E0, -8.350
21, .9950, 0.0D0, 0.03810, 0.0E0, 0.12E0, -8.700

__no of rad stations for airfoil table____ DATA IS DIFF for non c81__
5

__station at which airfoil table is considered___
0.00, 0.24, 0.92, 0.95, 1.00

__logical unit for the airfoil tables___
28

__The name of the file corresponding to lun = 28___
64-x08.c81

__The name of the file corresponding to lun = 28___
64-x12.c81

__The name of the file corresponding to lun = 28___
64-x18.c81

__The name of the file corresponding to lun = 28___
64-x25.c81

LWING__LNACEL__LV22BD
TRUE, TRUE, TRUE

LTRIM__idtriml__ctreq__DELCOLM__TRIMTOL
TRUE, 10, 0.013d0, 2d0, 0.00001

In the subsequent pages, the meaning of the field in each line, the type of the variable, and the possible values are explained.

__DEBUG_ITEROP_ITERCL_LUNWR_LUNRS_LUNFL_LUNPL_LUNTB_LUNCT__
FALSE, 20, 180, 6, 8, 9, 10, 17, 18

This line has 9 fields and the values need to be separated by comma. The user is advised not to change these values in this line except the value for field DEBUG. Each field is explained below.
Table 4: A1: Input/Output Print Control Parameters

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEBUG</td>
<td>True/False</td>
<td>Whether Debug information is required</td>
</tr>
<tr>
<td>ITEROP</td>
<td>Integer</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>ITERCL</td>
<td>Integer</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>LUNWR</td>
<td>Integer</td>
<td>Logical Unit for standard output</td>
</tr>
<tr>
<td>LUNRS</td>
<td>Integer</td>
<td>Logical Unit for Restart files</td>
</tr>
<tr>
<td>LUNFL</td>
<td>Integer</td>
<td>Logical Unit for field information</td>
</tr>
<tr>
<td>LUNPL</td>
<td>Integer</td>
<td>Logical Unit for plot files</td>
</tr>
<tr>
<td>LUNTB</td>
<td>Integer</td>
<td>Logical Unit for files</td>
</tr>
<tr>
<td>LUNCT</td>
<td>Integer</td>
<td>Logical Unit for files</td>
</tr>
</tbody>
</table>
This line has 8 fields and the values need to be separated by comma. The fields in this line, allows the user to selectively print the fields, to assist in debugging. This is generally set to FALSE otherwise the output would be very HUGE.

Table 5: A2: Flow Field Print Control Flags

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DETAIL</td>
<td>TRUE/FALSE</td>
<td>If FALSE none of the following values will be printed</td>
</tr>
<tr>
<td>U</td>
<td>TRUE/FALSE</td>
<td>If TRUE the values of U velocity will be printed</td>
</tr>
<tr>
<td>V</td>
<td>TRUE/FALSE</td>
<td>If TRUE the values of V velocity will be printed</td>
</tr>
<tr>
<td>P</td>
<td>TRUE/FALSE</td>
<td>If TRUE the values of P Pressure will be printed</td>
</tr>
<tr>
<td>W</td>
<td>TRUE/FALSE</td>
<td>If TRUE the values of W velocity will be printed</td>
</tr>
<tr>
<td>str Fun</td>
<td>TRUE/FALSE</td>
<td>Not Applicable - Do not Change</td>
</tr>
<tr>
<td>GAM</td>
<td>TRUE/FALSE</td>
<td>If TRUE the values of Viscosity will be printed</td>
</tr>
<tr>
<td>RO</td>
<td>TRUE/FALSE</td>
<td>If TRUE the values of Density will be printed</td>
</tr>
</tbody>
</table>

Table 6: A3: Solid Body Boundary Condition Control Flags

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LROCON</td>
<td>TRUE/FALSE</td>
<td>If TRUE Density is a constant</td>
</tr>
<tr>
<td>LMUCON</td>
<td>TRUE/FALSE</td>
<td>If TRUE Viscosity is a constant</td>
</tr>
<tr>
<td>LSOLID</td>
<td>TRUE/FALSE</td>
<td>If FALSE there is no body</td>
</tr>
<tr>
<td>LDATPR</td>
<td>TRUE/FALSE</td>
<td>For example, in the case of an isolated rotor, LSOLID should be FALSE and for the tilt rotor with a wing, the LSOLID is TRUE as there is a solid body in the field</td>
</tr>
<tr>
<td>LUGRID</td>
<td>TRUE/FALSE</td>
<td>Whether User defined special Grid information has been provided to specify the geometry</td>
</tr>
</tbody>
</table>

Note:

- Make sure to set LROCON and LMUCON to TRUE as this version provides for constant density and constant viscosity.
- LUGRID has a specialised use; Make sure to set LUGRID to TRUE.
Table 7: A4: Geometry and Flow Print Control Flags

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LGEOMP</td>
<td>TRUE/FALSE</td>
<td>Debug Prints Geometry related parameters</td>
</tr>
<tr>
<td>LUCOFP</td>
<td>TRUE/FALSE</td>
<td>Debug Prints U Coefficient</td>
</tr>
<tr>
<td>LVCOFP</td>
<td>TRUE/FALSE</td>
<td>Debug Prints V Coefficient</td>
</tr>
<tr>
<td>LWCOFP</td>
<td>TRUE/FALSE</td>
<td>Debug Prints W Coefficient</td>
</tr>
<tr>
<td>LPCOFP</td>
<td>TRUE/FALSE</td>
<td>Debug Prints P Coefficient</td>
</tr>
<tr>
<td>LTOCOFP</td>
<td>TRUE/FALSE</td>
<td>Debug Prints T Coefficient</td>
</tr>
<tr>
<td>LPLOTP</td>
<td>TRUE/FALSE</td>
<td>Debug plotting grid information</td>
</tr>
<tr>
<td>LFILDP</td>
<td>TRUE/FALSE</td>
<td>ASCII output of (X,Y,Z) of the grid</td>
</tr>
<tr>
<td>LPLT3D</td>
<td>TRUE/FALSE</td>
<td>If TRUE prints output in PLOT3D format for usage with FAST in two files in binary format The fort.23 and fort.24 files are the grid and the Q-file respectively.</td>
</tr>
</tbody>
</table>

MAX_ITERATIONS, ITEMOD, NU, NV, NW, NP, NT

200, 10, 2, 2, 2, 6, 75

The meanings of the above fields are explained in the following table.

Table 8: A5: Flow Numerical Iteration Control Variables

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX_ITERATIONS</td>
<td>Integer</td>
<td>Maximum Number of Iterations</td>
</tr>
<tr>
<td>ITEMOD</td>
<td>Integer</td>
<td>Iteration Modulus - After every ITEMOD iterations, restart information will be written to restart file</td>
</tr>
<tr>
<td>NU</td>
<td>Integer</td>
<td>Number of U Velocity sweeps for each solver iteration (2 is found to be optimal )</td>
</tr>
<tr>
<td>NV</td>
<td>Integer</td>
<td>Number of V Velocity sweeps for each solver iteration (2 is found to be optimal )</td>
</tr>
<tr>
<td>NW</td>
<td>Integer</td>
<td>Number of W Velocity sweeps for each solver iteration (2 is found to be optimal )</td>
</tr>
<tr>
<td>NP</td>
<td>Integer</td>
<td>Number of P Pressure sweeps for each solver iteration (6 is found to be optimal )</td>
</tr>
<tr>
<td>NT</td>
<td>Integer</td>
<td>Do not Change</td>
</tr>
</tbody>
</table>

It is used to determine the number of iterations required to stabilize the flowfield
In this line, the fluid density $\rho$ and viscosity $\mu$, and the numerical relaxation parameters (for each of the momentum equation in all the three coordinate directions) are entered. Lower Relaxation will result in better stability while higher relaxation may converge faster; suggested initial value for relaxations is 0.05D0. The User will have the flexibility to increase the initial relaxation values (entered in the above line) periodically after a User's specified number of iterations, and also limit the maximum values of these parameters by entering the appropriate fields in subsequent lines.

| Table 9: A6: Numerical Relaxation Control Variables |
|---------------------------------|----------|----------|
| Parameter Name | Values Type | Description |
| RHO | Real | Density |
| MU | Real | Kinematic Viscosity |
| RELAX-X | Real | Relaxation in X direction |
| RELAX-Y | Real | Relaxation in Y direction |
| RELAX-Z | Real | Relaxation in Z direction |

In the above line, the freestream values for the velocity components, pressure, and temperature are entered. For hover calculations, $U_{\text{INF}}$ and $V_{\text{INF}}$ are set to zero and $W_{\text{INF}}$ is set to -1 ft/s at the top boundary. This program is designed for a hovering rotor. The computational domain is shown in the following figure with the global (X,Y,Z) coordinate system. The boundaries of the domain are at distances considered to be numerically infinite. For a hovering rotor, a far upstream velocity of -1 ft/sec is necessary (at Z-max).

Figure 22: Global coordinates and boundary conditions.
Table 10: A7: Flowfield Domain Boundary Condition Variables

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UINF</td>
<td>Real</td>
<td>U Velocity at Infinity</td>
</tr>
<tr>
<td>VINF</td>
<td>Real</td>
<td>V Velocity at Infinity</td>
</tr>
<tr>
<td>WINF</td>
<td>Real</td>
<td>W Velocity at Infinity</td>
</tr>
<tr>
<td>PINF</td>
<td>Real</td>
<td>Pressure at Infinity</td>
</tr>
<tr>
<td>REAL-GAS-CON</td>
<td>Real</td>
<td>Gas Constant (R)</td>
</tr>
<tr>
<td>GAMMA</td>
<td>Real</td>
<td>Ratio of specific heats</td>
</tr>
</tbody>
</table>

..LINI____OUTI___LINJ___LOUTJ___LINK__LOUTK___LIP____LJP____LKP__...
TRUE, TRUE, TRUE, TRUE, TRUE, FALSE, TRUE, TRUE, TRUE

The first six logical flags in the above allow the User to treat the respective boundaries as symmetric/outflow or specified boundaries. The ‘IN’ (preceeding I, J, or K) indicates the boundary plane at X, Y, or Z values smaller than the one which are preceeded by the ‘OUT’. There must be at least one specified boundary and one outflow boundary.

For the hover case, the top boundary is specified with a velocity of -1 ft/s. Hence, ‘LOUTK’ is FALSE. The rest of the boundaries are allowed to float and hence all the corresponding logical parameter values are set to TRUE. At a boundary where the velocity is specified and is not allowed to change, this variable should be set to FALSE.

The fields LIP, LJP, and LKP are logical variables which allow the user to selectively choose directions in which the flow solver has to be inverted. For example, if LIP is set to FALSE, then the inversion routine solves only in the j- and k-directions and not in the i-direction. It is recommended that all values are set to TRUE for the hovering rotor.

Table 11: A8: Flowfield Numerical Solution Directive Control Flags

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINI</td>
<td>TRUE/FALSE</td>
<td>Set to TRUE if X-min is a symmetry or outflow Boundary</td>
</tr>
<tr>
<td>LOUTI</td>
<td>TRUE/FALSE</td>
<td>Set to TRUE if X-max is a symmetry or outflow Boundary</td>
</tr>
<tr>
<td>LINJ</td>
<td>TRUE/FALSE</td>
<td>Set to TRUE if Y-min is a symmetry or outflow Boundary</td>
</tr>
<tr>
<td>LOUTJ</td>
<td>TRUE/FALSE</td>
<td>Set to TRUE if Y-max is a symmetry or outflow Boundary</td>
</tr>
<tr>
<td>LINK</td>
<td>TRUE/FALSE</td>
<td>Set to TRUE if Z-min is a symmetry or outflow Boundary</td>
</tr>
<tr>
<td>LOUTK</td>
<td>TRUE/FALSE</td>
<td>Set to TRUE if Z-max is a symmetry or outflow Boundary</td>
</tr>
<tr>
<td>LIP</td>
<td>TRUE/FALSE</td>
<td>Set to TRUE to solve in I direction</td>
</tr>
<tr>
<td>LJP</td>
<td>TRUE/FALSE</td>
<td>Set to TRUE to solve in J direction</td>
</tr>
<tr>
<td>LKP</td>
<td>TRUE/FALSE</td>
<td>Set to TRUE to solve in K direction</td>
</tr>
</tbody>
</table>
The meanings of the above fields are adequately explained in the following table.

### Table 12: A9: Computational Domain BC Setting Flags

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPREF</td>
<td>Integer</td>
<td>All pressure calculations are relative to the pressure at a point. The coordinate indexes of this point are identified by (IPREF, JPREF, KPREF)</td>
</tr>
<tr>
<td>JPREF</td>
<td>Integer</td>
<td>Set to TRUE if the Xmin-plane is a wall</td>
</tr>
<tr>
<td>KPREF</td>
<td>Integer</td>
<td>Set to TRUE if the Xmax-plane is a wall</td>
</tr>
<tr>
<td>WINI</td>
<td>TRUE/FALSE</td>
<td>Set to TRUE if the Ymin-plane is a wall</td>
</tr>
<tr>
<td>WOUTI</td>
<td>TRUE/FALSE</td>
<td>Set to TRUE if the Ymax-plane is a wall</td>
</tr>
<tr>
<td>WINJ</td>
<td>TRUE/FALSE</td>
<td>Set to TRUE if the Zmin-plane is a wall</td>
</tr>
<tr>
<td>WOUTJ</td>
<td>TRUE/FALSE</td>
<td>Set to TRUE if the Zmax-plane is a wall</td>
</tr>
</tbody>
</table>

### Table 13: A10: Numerical Relaxation Control Variables

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRELXU</td>
<td>Real</td>
<td>Relaxation increment for solving the u-velocity</td>
</tr>
<tr>
<td>DRELXV</td>
<td>Real</td>
<td>Relaxation increment for solving the v-velocity</td>
</tr>
<tr>
<td>DRELXW</td>
<td>Real</td>
<td>Relaxation increment for solving the w-velocity</td>
</tr>
<tr>
<td>DRELXT</td>
<td>Real</td>
<td>Relaxation increment for solving the temperature</td>
</tr>
<tr>
<td>DRELXP</td>
<td>Real</td>
<td>Relaxation increment for solving the pressure</td>
</tr>
<tr>
<td>MODREL</td>
<td>Real</td>
<td>Number of iterations between two successive changes in the relaxations</td>
</tr>
<tr>
<td>RELMAX</td>
<td>Real</td>
<td>Upper limit of all the relaxation values</td>
</tr>
</tbody>
</table>

The meanings of the above fields are adequately explained in the following table.
Table 14: A11: Rotor Airfoils Configuration Flags

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LC81</td>
<td>TRUE/FALSE</td>
<td>Whether the Airfoil is of type c81</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If TRUE indicates that the airfoil is of type c81</td>
</tr>
<tr>
<td>LSRSEC</td>
<td>TRUE/FALSE</td>
<td>If TRUE indicates that the entire rotor is represented by one airfoil table</td>
</tr>
<tr>
<td>LSMSEC</td>
<td>TRUE/FALSE</td>
<td>If TRUE the airfoil tables have only one Mach number entry</td>
</tr>
<tr>
<td>LCLCD</td>
<td>TRUE/FALSE</td>
<td>Inactive and do not change</td>
</tr>
<tr>
<td>LCLCOR</td>
<td>TRUE/FALSE</td>
<td>Inactive and do not change</td>
</tr>
<tr>
<td>LROTBOD</td>
<td>TRUE/FALSE</td>
<td>Inactive and do not change</td>
</tr>
</tbody>
</table>
The meanings of the above fields are explained in the following table.

Table 15: A12: Flowfield Solution Directive Monitoring Variables

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IUCK</td>
<td>Integer</td>
<td>I Grid Point where we monitor the velocity</td>
</tr>
<tr>
<td>JVCK</td>
<td>Integer</td>
<td>J Grid Point where we monitor the velocity</td>
</tr>
<tr>
<td>KWCK</td>
<td>Integer</td>
<td>K Grid Point where we monitor the velocity</td>
</tr>
<tr>
<td>XUCK</td>
<td>Real</td>
<td>X coordinate where we need to print some value /* NOT USED */</td>
</tr>
<tr>
<td>YVCK</td>
<td>Real</td>
<td>Y coordinate where we need to print some value /* NOT USED */</td>
</tr>
<tr>
<td>ZWCK</td>
<td>Real</td>
<td>Y coordinate where we need to print some value /* NOT USED */</td>
</tr>
</tbody>
</table>

Table 16: A13: Number of Rotor Configuration Control

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Rotors</td>
<td>Integer</td>
<td>Number of Rotors</td>
</tr>
<tr>
<td>IBOUND</td>
<td>Number</td>
<td>4 is used for representing the boundary conditions for a hovering rotor</td>
</tr>
</tbody>
</table>

Table 17: A14: Rotor Airfoil Identification Command

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of Airfoil Used</td>
<td>Character</td>
<td>Name of Airfoil Used</td>
</tr>
</tbody>
</table>

---

"...Velocity check...IUCK...JVCK...KWCK...XUCK...YVCK...ZWCK...
26, 22, 17, 0.0, 0.0, 0.0...

The meanings of the above fields are explained in the following table.

Table 15: A12: Flowfield Solution Directive Monitoring Variables

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IUCK</td>
<td>Integer</td>
<td>I Grid Point where we monitor the velocity</td>
</tr>
<tr>
<td>JVCK</td>
<td>Integer</td>
<td>J Grid Point where we monitor the velocity</td>
</tr>
<tr>
<td>KWCK</td>
<td>Integer</td>
<td>K Grid Point where we monitor the velocity</td>
</tr>
<tr>
<td>XUCK</td>
<td>Real</td>
<td>X coordinate where we need to print some value /* NOT USED */</td>
</tr>
<tr>
<td>YVCK</td>
<td>Real</td>
<td>Y coordinate where we need to print some value /* NOT USED */</td>
</tr>
<tr>
<td>ZWCK</td>
<td>Real</td>
<td>Y coordinate where we need to print some value /* NOT USED */</td>
</tr>
</tbody>
</table>

...Number_of_Rotors......DATA_SET_REFERENCE = 14 NASA(TM X-952)...IBOUND
1, 4...

Table 16: A13: Number of Rotor Configuration Control

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Rotors</td>
<td>Integer</td>
<td>Number of Rotors</td>
</tr>
<tr>
<td>IBOUND</td>
<td>Number</td>
<td>4 is used for representing the boundary conditions for a hovering rotor</td>
</tr>
</tbody>
</table>

...Name of Airfoil Used...
MODIFIED NACA.0012...

Table 17: A14: Rotor Airfoil Identification Command

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of Airfoil Used</td>
<td>Character</td>
<td>Name of Airfoil Used</td>
</tr>
</tbody>
</table>
************ Data for rotor # 1 ************

Example:

_Airfoil___#_blades___Press dist(r/R)___

1 , 3.0D0 , 0.250D0

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Airfoils</td>
<td>Integer</td>
<td>Number of Airfoils</td>
</tr>
<tr>
<td>Number of Blades</td>
<td>Real</td>
<td>Number of Blades</td>
</tr>
<tr>
<td>r/R</td>
<td>Real</td>
<td>A constant used in tip correction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Do not change</td>
</tr>
</tbody>
</table>

Tip speed___Rotor radius_Hub radius(r/R)_Hinge Offse(r/R)_clock rot___
723.12D0 , 12.5 , 0.1353 , 0.0357 , FALSE

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tip Speed</td>
<td>Real</td>
<td>Linear velocity of the tip</td>
</tr>
<tr>
<td>Rotor Radius</td>
<td>Real</td>
<td>Rotor Radius</td>
</tr>
<tr>
<td>Hub Radius</td>
<td>Real</td>
<td>Hub Radius</td>
</tr>
<tr>
<td>Hinge Offset</td>
<td>Real</td>
<td>Hinge Offset</td>
</tr>
<tr>
<td>clock rot</td>
<td>TRUE/FALSE</td>
<td>Clock rotation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If TRUE implies that the rotor is spinning clockwise</td>
</tr>
</tbody>
</table>

Number Of Reference Rotor radius(r/R) (max has been set to 45)_______

44

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Reference Rotor radius(r/R)</td>
<td>Integer</td>
<td>Number of Reference Rotor radius(r/R)</td>
</tr>
</tbody>
</table>

Reference Rotor radius(r/R)-----------------------------------------------

.100, .125, .150, .175,
.200, .225, .240, .250, .260, .275, .290, .300, .310, .325,
.340, .350, .375, .400, .425, .475, .500, .525, .575, .600,
.625, .650, .675, .700, .750, .775, .800, .825, .850, .875,
.900, .915, .925, .940, .950, .960, .970, .980, .990, .995
Table 21: A18: Rotor Blade Geometry and Operating Controls

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CL cor(r/R)</td>
<td>Real</td>
<td>Tip correction starts at this radius</td>
</tr>
<tr>
<td>Ref Twist</td>
<td>Real</td>
<td>Reference Twist</td>
</tr>
<tr>
<td># Blade Div</td>
<td>Number</td>
<td>Number of Blade Divisions</td>
</tr>
<tr>
<td>Azi cor</td>
<td>Real</td>
<td>Azimuthal Corrections</td>
</tr>
<tr>
<td># azi loc</td>
<td>Number</td>
<td>Number of azimuth locations at which output is desired</td>
</tr>
<tr>
<td>conv fac</td>
<td>Real</td>
<td>Inactive and do not change</td>
</tr>
</tbody>
</table>

Table 22: A19: Rotor Blade Flap and Pitch Motions

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td># harmonic pit</td>
<td>Number</td>
<td>Number of entries in the harmonic pitch table</td>
</tr>
<tr>
<td># harmonic flap</td>
<td>Number</td>
<td>Number of entries in the harmonic flap table</td>
</tr>
<tr>
<td>FLAPPING</td>
<td>TRUE/FALSE</td>
<td>If TRUE the rotor is allowed to flap</td>
</tr>
<tr>
<td>flap geo corr</td>
<td>TRUE/FALSE</td>
<td>If TRUE flap Geometry correction is made</td>
</tr>
</tbody>
</table>

The following table contains as many entries as they are stated for the harmonic pitch. In the above example, two entries are specified for the harmonic pitch table. The first value ($a_o$) refers to the collective pitch. All the values are given in radians and the series is assumed to be positive of the form:

$$\theta = a_o + \sum_{n=1}^{N} a_n \cos^n \psi + \sum_{n=1}^{N} b_n \sin^n \psi.$$  \hspace{1cm} (1)

-Harmonic_pitching_coefficients-Positive series-
0, \hspace{0.5cm} 13.35, \hspace{0.5cm} 0.000
1, \hspace{0.5cm} -0.50, \hspace{0.5cm} 0.300

Similarly, three entries are specified for the harmonic flap table in the example above. The first value ($c_o$) refers to the cone and the other terms belong to the positive series. All values are given in radians and take the form:

$$\beta = c_o + \sum_{n=1}^{N} c_n \cos^n \psi + \sum_{n=1}^{N} d_n \sin^n \psi.$$  \hspace{1cm} (2)

-Harmonic_flapping_coefficients-Positive series-
0 \hspace{0.5cm} 0.0000 \hspace{0.5cm} 0.0000
1 \hspace{0.5cm} 0.0000 \hspace{0.5cm} 0.0000
2 \hspace{0.5cm} 0.0000 \hspace{0.5cm} 0.0000
# data_points: 21

nn, r/R, deflec, Chord/RAD, CL des, T/chrd, Twist

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td># data_points</td>
<td>Integer</td>
<td>Number of data entries in the table that describes the geometric parameters of the rotor blade</td>
</tr>
<tr>
<td>nn</td>
<td>Integer</td>
<td>The table entry</td>
</tr>
<tr>
<td>r/R</td>
<td>Real</td>
<td>Non-dimensionalized radius</td>
</tr>
<tr>
<td>deflec</td>
<td>Real</td>
<td>Out of plane deflection in degrees</td>
</tr>
<tr>
<td>Chord/RAD</td>
<td>Real</td>
<td>Chord non-dimensionalized by radius</td>
</tr>
<tr>
<td>CL des</td>
<td>Real</td>
<td>Inactive</td>
</tr>
<tr>
<td>T/chrd</td>
<td>Real</td>
<td>Inactive</td>
</tr>
<tr>
<td>Twist</td>
<td>Real</td>
<td>Geometric twist of the rotor</td>
</tr>
</tbody>
</table>

__station at which airfoil table is considered__
0.00, 0.24, 0.92, 0.95, 1.00

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>no of rad stations for airfoil table</td>
<td>Integer</td>
<td>Number of radial stations for airfoil table</td>
</tr>
</tbody>
</table>

__logical unit for the airfoil tables__
28

__The name of the file corresponding to lun = 28__
64-x08.c81

__The name of the file corresponding to lun = 28__
64-x08.c81
Table 26: A23: Rotor C81 Table File(s) Designation

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>logical unit for the airfoil tables</td>
<td>Integer</td>
<td>Logical unit for the airfoil tables</td>
</tr>
<tr>
<td>Name of the lun = 28 file</td>
<td>Integer</td>
<td>The name of the file containing Airfoil tables; one such line must be provided for each Airfoil table</td>
</tr>
</tbody>
</table>

.LWING_LNACEL_LV22BD
TRUE, TRUE, TRUE

Table 27: A24: Aircraft Components Declaration

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWING</td>
<td>TRUE/FALSE</td>
<td>If TRUE wing is present</td>
</tr>
<tr>
<td>LNACEL</td>
<td>TRUE/FALSE</td>
<td>If TRUE nacelle is present</td>
</tr>
<tr>
<td>LV22BD</td>
<td>TRUE/FALSE</td>
<td>If TRUE fuselage is present</td>
</tr>
</tbody>
</table>

.LTRIM_idtriml_ctreq_DELCOLM_TRIMTOL
TRUE, 10, 0.013, 2.0, 0.00001

Table 28: A25: Aircraft Trim Conditions

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Values Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTRIM</td>
<td>TRUE/FALSE</td>
<td>If TRUE the rotor is trimmed for a given thrust</td>
</tr>
<tr>
<td>idtriml</td>
<td>Integer</td>
<td>Number of iterations between trim</td>
</tr>
<tr>
<td>ctreq</td>
<td>Real</td>
<td>Non-dimensional thrust required</td>
</tr>
<tr>
<td>DELCOLM</td>
<td>Real</td>
<td>Maximum collective change permitted during trim</td>
</tr>
<tr>
<td>TRIMTOL</td>
<td>Real</td>
<td>Trim tolerance for ( \theta ) collective</td>
</tr>
</tbody>
</table>
B  ROTTLILT Input Geometry Scale Down or Up Program

* -*-FORTRAN-***
* *************************************************************************
* * Developed by Hormoz Tadghighi * *
* * Boeing Helicopter Company * *
* * Mesa, Arizona * *
* * Output: Scale Geometry from Full Scale to Model Scale * *
* *************************************************************************
* *
*****************************************************************************
*****************************************************************************

Program Main
parameter (mi = 300, mj = 300, mk = 300)
dimension xf(mi),yf(mi,mj),zf(mi,mj)
dimension xwu(mi,mj),ywmi(mi),zwu(mi,mj)
dimension xwl(mi,mj),ywmi(mi),zwl(mi,mj)
dimension xn(mi,mj),yn(mi,mj),zn(mi)
c Input Files
c open(10, file='fsl.OUT',status='unknown')
open(11, file='wng.OUT',status='unknown')
open(12, file='ncl.OUT',status='unknown')
c Output Files
c open(14, file='fsl-plot.OUT',status='unknown')
open(15, file='wngupper-plot.OUT',status='unknown')
open(17, file='wnglower-plot.OUT',status='unknown')
open(16, file='ncl-plot.OUT',status='unknown')
c c Read
c c Fuselage Sections
c
read (10,*)imax1,jmax1
c do k = 1, imax1
read(10,*)xf(k)
do j = 1, jmax1
read(10,*)yf(j,k),zf(j,k)
end do
done
close (10)
c c

48
c Wing Sections

c
    read (11,*)imx2,jmax2
    c
    do k = 1, imax2
        read(11,*)yw(k)
        do j = 1, jmax2
            read(11,*)xw(j,k),zw(j,k)
        end do
    end do
    c
    do k = 1, imax4
        read(11,*)yw1(k)
        do j = 1, jmax4
            read(11,*)xw1(j,k),zw1(j,k)
            delta = zw1(j,K) - zw(j,k)
            zw1(j,k) = zw(j,k) + delta
        end do
    end do
    close(11)
    c
    c Nacelle Sections
    c
    read (12,*)imx3,jmax3
    c
    do k = 1, imax3
        read(12,*)zn(k)
        do j = 1, jmax3
            read(12,*)xn(j,k),yn(j,k)
        end do
    end do
    close (12)
    c
    c Write Output Files - Fuselage
    c
    do k = 1,imax1
        do j = 1,jmax1
            write(14,1001)xf(k),yf(j,k),zf(j,k)
        end do
    end do
    c
    c
    c Write Output Files - wing
    c
    do k = 1,imax2
        do j = 1,jmax2
            write(15,1001)xw(j,k),yw(k),zw(j,k)
        end do
    end do

49
end do

do k = 1,imax4
do j = 1,jmax4
write(17,1001)xwl(j,k),ywl(k),zwl(j,k)
end do
end do

C
C
C Write Output Files - nacelle
C
do k = 1,imax3
do j = 1,jmax3
write(16,1001)xn(j,k),yn(j,k),zn(k)
end do
end do

1001 format(2x,f14.5,1x,f14.7,1x,f14.7)

C
close (14)
close (15)
close (16)
stop
end

c------Input Namelist------------------------
&INPUTS
NB = 3,
CMEAN = 1.7225,
RHO = 1.234,
RADIUS = 3.81,
VTIP = 220.41,
OMEGA = 57.84,
FC = 30.,
FCC = 10.,
&END
C ROTTILT Grid Conversion into a 2D Gnuplot’s Format

* --FORTRAN-- *
* ************************************************************************* *
* * * Developed by Hormoz Tadghighi * *
* * * Boeing Helicopter Company * *
* * * Mesa, Arizona * *
* * * Grid Generated by ROTTILT Solver * *
* * ************************************************************************* *
* *
*****************************************************************************

Program Main
integer lu, luu, N, M, N1, N2, N3, M1, M2, M3
real*8 X(10000,10),Y(10000,10)
lu = 39
open (39, file='grid-2d-xyz.dat', status='unknown')
open (40, file='xygrid.out', status='unknown')
open (41, file='xzgrid.out', status='unknown')
open (42, file='yzgrid.out', status='unknown')
do i = 1,3
   Read (lu,15) N, M
   if(i .eq. 1)N1 = N
   if(i .eq. 1)M1 = M
   if(i .eq. 2)N2 = N
   if(i .eq. 2)M2 = M
   if(i .eq. 3)N3 = N
   if(i .eq. 3)M3 = M
   do k = i, M
      do j = 1, N
         read (lu,20) X(j,i), Y(k,i)
      end do
   end do
end do
read(lu,101)
end do
luu = 39
do i = 1,3
   luu = luu + 1
   if(i .eq. 1)N = N1
   if(i .eq. 1)M = M1
   if(i .eq. 2)N = N2
   if(i .eq. 2)M = M2
   if(i .eq. 3)N = N3
   if(i .eq. 3)M = M3
   write (luu,15) N, M
   do k = 1, M

51
do j = 1, N
    write (luu,20) X(j,i), Y(k,i)
end do
end do
end do

15 format (I25, I25)
20 format(1X,2(1X,F20.10))
101 format(4a20)

close (39)
close (40)
close (41)
close (42)

stop
end
D Geometry Section Cuts Generation Program

```fortran
Program Spline
  Call Tiltrotor
  Stop
End Program Spline

Subroutine Tiltrotor
  Implicit Integer(I-N)
  Implicit Real*4(A-H,O-Z)
  Character*60 Dir,Fname,Outname
  Dimension Pt1(200),Pt2(200),Pt3(200),Ptnew1(5000)
  Dimension Ptnew2(5000),Ptnew3(5000),Index(200)
  Dimension Rho(200),Tau(200),S(200)
  Print *, 'What File Do You Want To Manipulate?'
  Read(*,'(A30)') Fname
  Print *, 'How Many Cross-Sections Do You Want?'
  Read *, Nxsect
  Print *, 'In What Planes Are Your Cross-Sections Located?'
  Print *, '<xdir>, <ydir>, or <zdir>'
  Read(*,'(A4)') Dir
  Open(1,File=Fname,Status='Old')

C The Following Line Reads In The Header That Should Be
C Included In The File Read The Number Of Lines Of Data
C (NLines) And The Number Of Data Points Per Line (Max).
```

53
read(1,*) max, nlines

nx = 0
nz = 0
do 50 line = 1,nlines

    if (dir.eq.'xdir') then
        do 10 lninc = 1,max
            read(1,*) pt3(lninc),pt1(lninc),pt2(lninc)
        10        continue
    end if

    if (dir.eq.'ydir') then
        do 11 lninc = 1,max
            read(1,*) pt1(lninc),pt3(lninc),pt2(lninc)
        11        continue
    end if

    if (dir.eq.'zdir') then
        do 12 lninc = 1,max
            read(1,*) pt1(lninc),pt2(lninc),pt3(lninc)
        12        continue
    end if

    step = (pt3(max)-pt3(1))/real(nxsect-1)
do 20 lninc = 1,nxsect
        ptnew3(lninc) = pt3(1)+(lninc-1)*step
  20    continue

    if (line.eq.1) then
        step = (pt3(max)-pt3(1))/real(nxsect-1)
do 20 lninc = 1,nxsect
        ptnew3(lninc) = pt3(1)+(lninc-1)*step
  20    continue
end if

c create the new x - coordinates along the line
call spcoef(max,pt3,pt1,s,index,rho,tau,nxsect)
do 30 lninc = 1,nxsect
    nx = nx+1
    ptnew1(nx) = spline(max,pt3,pt1,s,index,ptnew3(lninc))
  30    continue

c create the new z - coordinates along the line
call spcoef(max,pt3,pt2,s,index,rho,tau,nxsect)
do 40 lninc = 1,nxsect
    nz = nz+1
    ptnew2(nz) = spline(max,pt3,pt2,s,index,ptnew3(lninc))
  40    continue

50  continue

close(1)
c TRANSFORM THE DATA TO READABLE OUTPUT TO THE DATA FILE IN THE
  c FORM OR Y LOCATION OF THE CROSS SECTION AND THE X AND Z

54
COORDINATES WITHIN THE SECTION.

print *, 'WHAT IS THE NAME OF THE OUTPUT FILE?'
read(*,'(a30)') outname
open(2,file=outname,status='unknown')
write(2,70) nxsect,nlines

format(1x,2(i3,3x))

ncount = 0
do 90 lninc = 1,nxsect
   write(2,100) ptnew3(lninc)
   do 80 line = 1,nlines
      write(2,110) ptnew1(lninc+nxsect*(line-1)),ptnew2(lninc + nxsect*(line-1))
   80    continue
90    continue

format(1x,f10.5)
format(1x,2(f10.5,4x))

close(2)
end

--------------------------------------------------------------------------
subroutine spcoef(n,xn,fn,s,index,rho,tau,npts)
implicit real*4 (a-h,o-z)
dimension xn(n),fn(n),s(n),index(n),rho(npts),tau(npts)
nml=n-1
do 1 i=1,n
   index(i)=i
1   index(i)=i
   ip1=i+1
   do 2 j=ip1,n
      ii=index(i)
      ij=index(j)
      if(xn(ii).le.xn(ij)) go to 2
      itemp=index(i)
      index(i)=index(j)
      index(j)=itemp
2    continue
3    continue
   nm2=n-2
   rho(2)=0.0e+00
   tau(2)=0.0e+00
   do 4 i=2,nm1
      iim1=index(i-1)
      ii=index(i)
      iip1=index(i+1)
      him1=xn(ii)-xn(iim1)
      hi=xn(iip1)-xn(ii)
      temp=(him1/hi)*(rho(i)+2.0e+00)+2.0e+00
4    continue
rho(i+1) = -1.0e+00 / temp

d = 6.0 * ((fn(iip1) - fn(ii)) / hi - (fn(ii) - fn(iim1)) / him1) / hi

4 tau(i+1) = (d - him1 * tau(i) / hi) / temp

s(i) = 0.0e+00
s(n) = 0.0e+00

do 5 i = 1, nm2
ib = n - i
5 s(ib) = rho(ib+1) * s(ib+1) + tau(ib+1)

return
end

function spline(n, xn, fn, s, index, x)

implicit real*4 (a-h, o-z)
dimension xn(n), fn(n), s(n), index(n)

ii = index(1)
if (x.ge.xn(ii)) go to 1
i2 = index(2)
h1 = xn(i2) - xn(ii)
spline = fn(ii) + (x-xn(ii)) * ((fn(i2) - fn(ii)) / h1 - h1*s(2) / 6.0)
return

1 in = index(n)
if (x.le.xn(in)) go to 2
inl = index(n-l)
hnml = xn(in) - xn(inml)
spline = fn(in) + (x-xn(in)) * ((fn(in) - fn(inml)) / hnml + hnml*s(n-l) / 6.0)
return

2 do 3 i = 2, n
ii = index(i)
if (x.le.xn(ii)) go to 4
continue
3
l = i - 1
ilp1 = index(l+1)
a = xn(ilp1) - x
b = x - xn(il)
hl = xn(ilp1) - xn(il)
spline = a*s(l) * (a**2 / hl - hl1) / 6.0 + b*s(l+1) * (b**2 / hl - hl1) / 6.0 + (a*fn(il) + b*fn(ilp1)) / hl
return
end
E  ROTTILT-WOPWOP-TIN2 Hi-Res Airloads and Inflow Velocities Conversions

*    -*FORTRAN-*-
*      **********************************************
*      * Developed by Hormoz Tadghighi     *
*      * Boeing Helicopter Company        *
*      * Mesa, Arizona                   *
*      * Output: Conversion of ROTTILT Performance *
*      * Output file for TIN2 & WOPWOP     *
*      * Programs & 2D Plots              *
*      **********************************************
*    *
*
program main
    implicit double precision (a-h,o-z)
    logical acou,plot,flow
    integer npsi,nrad
    dimension rad(361),azim(361),cl(361,361),
      > cd(361,361),slift(361,361),sdrag(361,361),
      > sload(361,361),thrust(361,361),torque(361,361),
      > alpha(361,361),beta(361,361),twist(361,361),
      > vr(1000,361),vtheta(361,361),vz(361,361),
      > FZ(361,361),FN(361,361),fact(361),parm(361),
      > dum(361),t(361),df(361),d(361),VZZ(361),
      > r1(3),r2(3),theta1(3),theta2(3)
    data acou '/true'/
    data plot '/false'/
    data flow '/true'/
C*************************************************************
    NAMELIST/INPUTS/ nb, cmean, rho, radius, vtip, omega, fc, fcc
C*************************************************************
OPEN(5,FILE='Conv.nam', status='unknown',
  > FORM='FORMATTED')
OPEN(8,FILE='rot-az-perf.dat', status='unknown',
  > FORM='FORMATTED')
OPEN(12,FILE='xvI5.1ds', status='unknown',
  > FORM='FORMATTED')
OPEN(13,FILE='Vz.inflow', status='unknown',
  > FORM='FORMATTED')
OPEN(14,FILE='Vz.plot', status='unknown',
  > FORM='FORMATTED')

pi = 4.*atan(1.)
DTR = 0.01745
c--------------set rotor sub-domain --------------
c sub domain 1
    r1(1) = 0.1
    r2(1) = 0.3
    theta1(1) = 0.
    theta2(1) = 360.
c sub domain 2
    r1(2) = 0.3
    r2(2) = 0.6
    theta1(2) = 0.
    theta2(2) = 360.
c sub domain 3
    r1(3) = 0.6
    r2(3) = 0.1
    theta1(3) = 0.
    theta2(3) = 360.
c--------------input read
C
C..Read in the input and write it back out
C
    READ (5, INPUTS)
    WRITE (6, INPUTS)
C
    read(8,1006)
    read(8,1006)
    read(8,1006)
        read(8,1001)L, nrad , npsi
    write(6,*)L,nrad , npsi

    do 2001 i=1,nrad-1
    read(8,1002)
    read(8,1003)rad1,rad(i)
    read(8,1004)
    do 2002 j=2,npsi+1

        if(i .lt. 36)
            > read(8,1005)azim(j),cl(i,j),cd(i,j),slift(i,j),sdrag(i,j),
            > sload(i,j),thrust(i,j),torque(i,j),alpha(i,j),beta(i,j),
            > twist(i,j),vr(i,j),vtheta(i,j),vz(i,j)

        if(i .ge. 36)
            > read(8,1007)azim(j),cl(i,j),cd(i,j),slift(i,j),sdrag(i,j),
            > sload(i,j),thrust(i,j),torque(i,j),alpha(i,j),beta(i,j),
            > twist(i,j),vr(i,j),vtheta(i,j),vz(i,j),dummy

    2002 continue
    2001 continue

C--------------------------------------------------
c_ perform Smoothing
C
    nd = npsi+1
    azim(1) =0.0
    dt = azim(2) - azim(1)
fc = fc/360.
fcc = fcc/360.

write(6,*) 'Sample interval = ', DT
DELF = 360/(azim(npsi+1)-azim(l))
write(6,*) 'Frequency resolution = ', DELF, ' per rev'
FMAX = DELF*(npsi-1)/2
write(6,*) 'Max. frequency = ', FMAX, ' per rev'
do i=1,nrad-1
   cl(i,l) = cl(i,npsi+1)
   cd(i,l) = cd(i,npsi+1)
   slift(i,l) = slift(i,npsi+1)
   sdrag(i,l) = sdrag(i,npsi+1)
   sload(i,l) = sload(i,npsi+1)
   thrust(i,l) = thrust(i,npsi+1)
   torque(i,l) = torque(i,npsi+1)
   alpha(i,l) = alpha(i,npsi+1)
   beta(i,l) = beta(i,npsi+1)
   twist(i,l) = twist(i,npsi+1)
   vr(i,l) = vr(i,npsi+1)
   vtheta(i,l) = vtheta(i,npsi+1)
   vz(i,l) = vz(i,npsi+1)
endo
c........ Sec Lift
do i = 1, nrad-1
do j = 1, npsi+1
   t(j) = azim(j)
dum(j) = cl(i,j)
endo
comment call smooth (nd,dt,fc,dum,df)
do j = 1, npsi+1
   cl(i,j) = df(j)
endo
c....Plot
   open(1,file='try1.out',status='new')
do I=1,nd
   write(I,*)T(I),DUM(I),DF(I)
endo
close(1)
call SYSTEM('gnuplot try1_plot.gpt')
call SYSTEM('rm try1.out')
c....End

c........ Sec Lift
do i = 1, nrad-1
do j = 1, npsi+1
   t(j) = azim(j)
dum(j) = cd(i,j)
endo
comment call smooth (nd,dt,fc,dum,df)
do j = 1, npsi+1
cd(i,j) = df(j)
enddo
enddo
c....Plot
ch    open(1,file='try1.out',status='new')
ch    do l=1,nd
ch        write(l,*),T(I),DUM(I),DF(I)
ch    end do
ch    close(i)
ch    call SYSTEM('gnuplot try1_plot.gpt')
ch    call SYSTEM('rm try1.out')
C....End

c........ Sec Lift
do i = 1, nrad-1
    do j = 1, npsi+1
        t(j) = azim(j)
        dum(j) = slift(i,j)
    enddo
    call smooth (nd,dt,fc,dum,df)
    do j = 1, npsi+1
        slift(i,j) = df(j)
    enddo
c....Plot
ch    open(1,file='try1.out',status='new')
ch    do Ii=1,nd
ch        write(Ii,*),T(Ii),DUM(Ii),DF(Ii)
ch    end do
ch    close(I)
ch    call SYSTEM('gnuplot try1_plot.gpt')
ch    call SYSTEM('rm try1.out')
C....End
    enddo

c........ Sec Lift
do i = 1, nrad-1
    do j = 1, npsi+1
        t(j) = azim(j)
        dum(j) = sdrag(i,j)
    enddo
    call smooth (nd,dt,fc,dum,df)
    do j = 1, npsi+1
        sdrag(i,j) = df(j)
    enddo
c....Plot
ch    open(1,file='try1.out',status='new')
ch    do II=1,nd
ch        write(II,*),T(II),DUM(II),DF(II)
ch    end do
ch    close(II)
ch    call SYSTEM('gnuplot try1_plot.gpt')
ch    call SYSTEM('rm try1.out')
C....End
c........... Sec Lift
do i = 1, nrad-1
do j = 1, npsi+1
   t(j) = azim(j)
dum(j) = sload(i,j)
endo
call smooth (nd,dt,fc,dum,df)
do j = 1, npsi+1
   sload(i,j) = df(j)
endo
do i = 1, nrad-1
do j = 1, npsi+i
   t(j) = azim(j)
dum(j) = thrust(i,j)
endo
call smooth (nd,dt,fcc,dum,df)
do j = 1, npsi+i
   thrust(i,j) = df(j)
endo

c........... Sec Lift
do i = 1, nrad-1
do j = 1, npsi+1
   t(j) = azim(j)
dum(j) = torque(i,j)
endo
call smooth (nd,dt,fcc,dum,df)
do j = 1, npsi+1
   torque(i,j) = df(j)
endo

c........... Sec Lift
do i = 1, nrad-1
do j = 1, npsi+1
   t(j) = azim(j)
dum(j) = thrust(i,j)
endo
call smooth (nd,dt,fcc,dum,df)
do j = 1, npsi+i
   thrust(i,j) = df(j)
endo
call smooth (nd,dt,fcc,dum,df)
do j = 1, npsi+i
   torque(i,j) = df(j)
endo

c....Plot
ch   open(1, file='tryl.out', status='new')
ch   do I=1,nd
ch      write(I,*) T(I), DUM(I), DF(I)
ch   end do
ch   close(1)
ch   call SYSTEM('gnuplot tryl_plot.gpt')
ch   call SYSTEM('rm tryl.out')
C....End
c........ Sec Lift
do i = 1, nrad-1
  do j = 1, npsi+1
    t(j) = azim(j)
    dum(j) = alpha(i,j)
  enddo
  call smooth (nd,dt,fc,dum,df)
  do j = 1, npsi+1
    alpha(i,j) = df(j)
  enddo
enddo

c........ Sec Lift
do i = 1, nrad-1
  do j = 1, npsi+1
    t(j) = azim(j)
    dum(j) = beta(i,j)
  enddo
  call smooth (nd,dt,fc,dum,df)
  do j = 1, npsi+1
    beta(i,j) = df(j)
  enddo
enddo

c....Plot
  open(1,file='tryl.out',status='new')
  do l=1,nd
    write(l,*)T(1),DUM(1),DF(1)
  end do
  close(1)
  call SYSTEM('gnuplot tryl_plot.gpt')
  call SYSTEM('rm tryl.out')
C....End

c........ Sec Lift
do i = 1, nrad-1
  do j = 1, npsi+1
    t(j) = azim(j)
    dum(j) = twist(i,j)
  enddo
  call smooth (nd,dt,fc,dum,df)
  do j = 1, npsi+1
    twist(i,j) = df(j)
  enddo
enddo

c....Plot
  open(1,file='tryl.out',status='new')
  do l=1,nd
    write(l,*)T(1),DUM(1),DF(1)
  end do
  close(1)
  call SYSTEM('gnuplot tryl_plot.gpt')
  call SYSTEM('rm tryl.out')
C....End
c........ Sec Lift
do i = 1, nrad-1
do j = 1, npsi+1
   t(j) = azim(j)
dum(j) = vr(i,j)
enddo
   call smooth (nd,dt,fc,dum,df)
enddo
do j = 1, npsi+1
   vr(i,j) = dum(j)
enddo

C....Plot
ch      open(l,file='try1.out',status='new')
ch      do l=1,nd
ch         write(l,*)T(l),DUM(l),DF(l)
ch      end do
ch      close(1)
ch      call SYSTEM('gnuplot try1_plot.gpt')
ch      call SYSTEM('rm try1.out')
C....End

--------- Sec Lift
do i = 1, nrad-1
do j = 1, npsi+1
   t(j) = azim(j)
dum(j) = vtheta(i,j)
enddo
   call smooth (nd,dt,fc,dum,df)
enddo
do j = 1, npsi+1
   vtheta(i,j) = df(j)
enddo

C....Plot
ch      open(l,file='try1.out',status='new')
ch      do l=1,nd
ch         write(l,*)T(l),DUM(l),DF(l)
ch      end do
ch      close(1)
ch      call SYSTEM('gnuplot try1_plot.gpt')
ch      call SYSTEM('rm try1.out')
C....End

--------- Sec Lift
do i = 1, nrad-1
do j = 1, npsi+1
   t(j) = azim(j)
dum(j) = vz(i,j)
enddo
   call smooth (nd,dt,fc,dum,df)
enddo
do j = 1, npsi+1
   vz(i,j) = df(j)
enddo

C....Plot
do 561 i=1,nrad-1  
conv = 0.5 * rho * (radius*omega*rad(i))**2.  
convl = 48.22 * radius * rad(i)  
c_______FORCE CONVERSION FOR WOPWOP______________  
do 562 j=1,npsi+1  
alphi = DTR*alpha(i,j)  
c lift and drag  
c dl=slift(i,j)  
c dd=sdrag(i,j)  
dl=thrust(i,j)  
dd=torque(i,j)  
c thrust and torque  
c FZ(I,J) = 4.*conv1*(dl*cos(-alphi)-dd*sin(-alphi))  
c FN(I,J) = 4.*conv1*(dl*sin(-alphi)+dd*cos(-alphi))  
  
  
562    CONTINUE  
c....Plot  
ch     open(1,file='tryl.out',status='new')  
ch     do jj=1,nd  
ch     write(1,*)azim(jj),fz(i,jj),fn(i,jj)  
ch     end do  
ch     close(1)  
ch     call SYSTEM('gnuplot tryl_plot.gpt')  
ch     call SYSTEM('rm tryl.out')  
C....End  
561    CONTINUE  
C  
C ENSURE PERIODICITY  
DO I = i, NRAD-I  
FZ(I,1) = FZ(I,NPSI+1)  
FN(I,1) = FN(I,NPSI+1)  
ENDDO  
C  
c********** acoustic output ****************
if( acou ) then  
write(12,3001) npsi+1,nrad-1  
write(12,3002)(rad(i), i=1,nrad-1)
do 401 j=1,npsi+1
write(12,3003)(FZ(i,j),i=1,nrad-1)
401 continue
do 402 j=1,npsi+1
write(12,3003)(FN(i,j),i=1,nrad-1)
402 continue
endif
c********** Inflow output ****************
if( flow ) then
   write(13,*)'radial station = 0.1 to 0.3'
   write(13,5001)NRAD, NPSI+1
   write(13,5002)r1(1),r2(1),theta1(1),theta2(1)
write(13,3001) npsi+1
   do i = 1, npsi+1
      VzSum = 0.0
      do j = 1, 12
         VzSum = VzSum + VZ(j,i)
      enddo
      Vzz(i) = VzSum/12.0
   enddo
write(13,3003)(VZZ(i),i=1,npsi+1)
   write(14,*)'radial station = 0.1 to 0.3'
   do k = 1,npsi+1
      write(14,3004)azim(k), VZZ(k)
   enddo
c+++++++++++++++++++++++++++++++++
write(13,*)'radial station = 0.3 to 0.6'
write(13,5001)NRAD, NPSI+1
write(13,5002)r1(2),r2(2),theta1(2),theta2(2)
   do i = 1, npsi+1
      VzSum = 0.0
      do j = 8, 24
         VzSum = VzSum + VZ(j,i)
      enddo
      Vzz(i) = VzSum/16.0
   enddo
write(13,3003)(VZZ(i),i=1,npsi+1)
   write(14,*)'radial station = 0.3 to 0.6'
   do k = 1,npsi+1
      write(14,3004)azim(k), VZZ(k)
   enddo
c+++++++++++++++++++++++++++++++++
write(13,*)'radial station = 0.6 to 1.0'
write(13,5001)NRAD, NPSI+1
write(13,5002)r1(3),r2(3),theta1(3),theta2(3)
   do i = 1, npsi+1
      VzSum = 0.0
      do j = 20, nrad-1
         VzSum = VzSum + VZ(j,i)
      enddo
      Vzz(i) = VzSum/27.0
   enddo
write(13,3003)(VZZ(i),i=1,npsi+1)
   write(14,*)'radial station = 0.6 to 1.0'
   do k = 1,npsi+1
      write(14,3004)azim(k), VZZ(k)
   enddo
c+++++++++++++++++++++++++++++++++
write(13,*)'radial station = 0.1 to 0.3'
write(13,5001)NRAD, NPSI+1
write(13,5002)r1(1),r2(1),theta1(1),theta2(1)
write(13,3003)(VZZ(i),i=1,npsi+1)
write(14,*)'radial station = 0.1 to 0.3'
write(14,3004)azim(k), VZZ(k)
enddo
65
do k = l,npsi+l
  write(14,3004)azim(k), VZZ(k)
enddo
endif

C------------------------------------------------------------------------
1001  format(1x,'rotor # ', i2,/','no of rad__no of azi'
   $ ,/i3,7x,'',i3)
1002  FORMAT(5x,'___radius_________r/R___')
1003  FORMAT(3x,1Pe12.4,2x,1Pe12.4)
1004  FORMAT(1x,' AZI',' s cl   s cd',6X,
>  'S LIFT',6X,'S DRAG',6X,'S LOAD',
>  '6x','Thrust',5x,'Torq Fr',ix,'Alpha',2x,'Beta'
>  ',(twist',3X,' vr','vtheta',3x,' vz'/,125('-'))
1005  FORMAT(1x,f6.1,(' ',f5.2,' ',f6.4),5(' ',e11.4),7(' ',f6.1))
1006  FORMAT(20A4)

C................................................................
5001 format(1x,2i4)
5002 format(2x,f10.4,2x,f10.4,2x,f10.4,2x,f10.4)
3001 format(1x,2i4)
3002 format(1x,5f7.4)
3003 format(1x,5F16.8)
3004 format(2(2x,fi6.8))
4001 format(1x,6f7.4)
4002 format(1x,6f12.7)
  close(5)
  close(8)
  close(9)
  close(12)
  close(13)
stop
end

C subroutine SMOOTH(N,DT,FC,D,DF)
C Subroutine to smooth out artificial wiggles in the data caused by
C insufficient grid resolution. The routine uses FFT to filter out
C frequencies beyond a specified cutoff frequency.
C Arguments:
C N=number of points in array.
C DT=time interval between data points.
C FC=cutoff frequency, specified in units of 1/(time units).
C D=array of input data points.
C DF=output array of filtered points.
C Note that the work array WSAVE must be dimensioned at least 2*N+15.
C This routine must be compiled along with the FFTPACK library.
implicit double precision (A-H,O-Z)
dimension D(*),DF(*),WSAVE(1000)
do I=1,N
  DF(I)=D(I)
end do
DELF=1.DO/((N-1)*DT)
NFC=int(FC/DELF)+1
call RFFTI(N,WSAVE)
call RFFTF(N,DF,WSAVE)
NUP=N
if(mod(N,2) .gt. 0)NUP=N-1
do I=1+2*NFC,NUP
   DF(I)=0.DO
end do
call RFFTBN(N,DF,WSAVE)
do I=1,N
   DF(I)=DF(I)/N
end do
return
den
F Hess Write Format

*C******** HESS FORMAT FILE
*********** INPUT FILE FROM CADDY PROGRAM

open(11,file='Blade.hess',status='unknown',form='formatted')
D0 4002 K = 2,km
D0 4002 J = 1,jm
D0 4002 I = 16,im
ICOLR = 0
IFLG = 0
IF (I .EQ. 16)THEN
IFLG = 1
IF (J .EQ. 1)IFLG=2
ENDIF
XH(I,J,K) = X(I,J,K)
YH(I,J,K) = YYY(I,J,K)
ZH(I,J,K) = Z(I,J,K)
WRITE(11,1012)XH(I,J,K),YH(I,J,K),ZH(I,J,K),IFLG,ICOLR
4002 CONTINUE
1012 FORMAT (3F10.6, 2I1)
G 2D Gnuplot Plotting Routine

* **FORTRAN**
* *********************************************************
* * Developed by Hormoz Tadghighi *
* * Boeing Helicopter Company *
* * Mesa, Arizona *
* * Output: Conversion of ROTTILT grid file *
* * into Gnuplot format *
* *********************************************************

program PLOTXY
real XG(500),YG(500),XOBJ(500,20),YOBJ(500,20)
real XGI(500),YGI(500)
integer NPOBJ(20)
character filnam*80,CH*3,XLBL*6,YLBL*6,TLBL*8,STR*80,BLANK*80

do I=1,80
  BLANK(I:I)=' '
end do
c Read grid point data file.
write(6,*')' Enter name of grid file'
read(5,'(a)')filnam
open(1,filnam,status='old')
continue
read(1,'(a)')STR
if(STR .eq. BLANK)goto 5
read(STR,*)NXG,NYG
do J=1,NYG
  YGI(J)=float(J)
do I=1,NXG
  read(1,'(a)')STR
  if(STR .eq. BLANK)then
    read(STR,*)XG(I),YG(J)
    XGI(I)=float(I)
  end if
end do
end do
c Read object contour data and find min/max extent of object dimensions.
write(6,*')' Enter name of object file'
read(5,'(a)')filnam
open(1,filnam,status='old')
read(1,*)NOBJ
XOBJMIN=1.E+30
XOBJMAX=-1.E+30
YOBJMIN=1.E+30
YOBJMAX=-1.E+30
do I=1,NOBJ
   read(I,*)NP0BJ(I)
do J=1,NP0BJ(I)
   read(I,'(a)')STR
   if(STR .ne. BLANK)then
      read(STR,*)N,XOBJ(J,I),YOBJ(J,I)
      if(XOBJ(J,I) .gt. XOBJMAX)XOBJMAX=XOBJ(J,I)
      if(XOBJ(J,I) .lt. XOBJMIN)XOBJMIN=XOBJ(J,I)
      if(YOBJ(J,I) .gt. YOBJMAX)YOBJMAX=YOBJ(J,I)
      if(YOBJ(J,I) .lt. YOBJMIN)YOBJMIN=YOBJ(J,I)
   end if
end do
dend do
close(i)
write(6,*)' Enter 1 for XY, 2 for YZ, 3 for XZ'
read(5,*)IXYZ
if(IXYZ .eq. 1)then
   XLBL='X AXIS'
   YLBL='Y AXIS'
   TLBL='XY PLANE'
else if(IXYZ .eq. 2)then
   XLBL='Y AXIS'
   YLBL='Z AXIS'
   TLBL='YZ PLANE'
else if(IXYZ .eq. 3)then
   XLBL='X AXIS'
   YLBL='Z AXIS'
   TLBL='XZ PLANE'
end if
XWi=XOBJMIN-0.2*(XOBJMAX-XOBJMIN)
XW2=XOBJMAX+0.2*(XOBJMAX-XOBJMIN)
YW1=YOBJMIN-0.2*(YOBJMAX-YOBJMIN)
YW2=YOBJMAX+0.2*(YOBJMAX-YOBJMIN)
if(XW1 .lt. XG(1))XW1=XG(1)
if(XW2 .gt. XG(NXG))XW2=XG(NXG)
if(YW1 .lt. YG(1))YW1=YG(1)
if(YW2 .gt. YG(NYG))YW2=YG(NYG)
IOUT=1
   c Start plotting segment (IOUT=1 for X plot, 2 for PS plot).
   10 continue
   if(IOUT .eq. 1)then
      ISTAT=PGBEG(0,'/xwindow',I,I)
   else
      ISTAT=PGBEG(0,'/ps',I,I)
   end if
   c Start new page.
call PGPAGE
   c Set viewport and window.
call PGVP(0.0,1.0,0.0,1.0)
call PGVP(1,VPX1,VPX2,VPY1,VPY2)
D=min(VPX2-VPX1,VPY2-VPY1)/40
VPX1=VPX1+8*D
VPX2=VPX2-2*D
VPY1=VPY1+8*D
VPY2=VPY2-8*D
call PGVSIZ(VPX1,VPX2,VPY1,VPY2)
call PGWNAD(XW1,XW2,YW1,YW2)
c c Draw and label frame around viewport.
call PGBOX('bcn',0.0,0,'bcn',0.0,0)
call PGLAB(XLBL,YLBL,TLBL)
c c Draw horizontal grid lines.
IY=0
IY1=1
IY2=NYG
do I=1, NYG
  if(IY .eq. 1 .and. YG(I) .gt. YW2)then
    IY=0
  end if
  if(YG(I) .ge. YW1 .and. YG(I) .le. YW2)then
    if(IY .eq. 0)then
      IY=I
    end if
    call PGMOVE(XG(I), YG(I))
    call PGDRAW(XG(NXG),YG(I))
  end if
end do
c c Draw vertical grid lines.
IX=0
IX1=1
IX2=NXG
do I=1, NXG
  if(IX .eq. 1 .and. XG(I) .gt. XW2)then
    IX=0
  end if
  if(XG(I) .ge. XW1 .and. XG(I) .le. XW2)then
    if(IX .eq. 0)then
      IX=I
    end if
    call PGMOVE(XG(I), YG(I))
    call PGDRAW(XG(I),YG(NYG))
  end if
end do
c c Draw objects with interiors filled with background color.
do I=1, NOBJ
call PGSLS(1)
call PGSW(10)
call PGLINE(NOBJ(I),XOBJ(I,I),YOBJ(I,I))
call PGSCI(0)
call PGPOLY(NOBJ(I),XOBJ(I,I),YOBJ(I,I))
call PGSCI(1)
end do
call PGSLW(1)
c Reset viewport and window to maximum extent to permit drawing second set of axes.
    call PGQVP(0,XVMIN,XVMAX,YVMIN,YVMAX)
call PGQWIN(XVMIN,XVMAX,YVMIN,YVMAX)
BVX=(XVMAX-XVMIN)/(XVMAX-XVMIN)
BVY=(YVMAX-YVMIN)/(YVMAX-YVMIN)
AVX=XVMIN-BVX*XVMIN
AVY=YVMIN-BVY*YVMIN
XW1N=AVX
XW2N=AVX+BVX
YW1N=AVY
YW2N=AVY+BVY
call PGSVIP(0.0,1.0,0.0,1.0)
call PGSWIN(XW1N,XW2N,YW1N,YW2N)
call PGSCII(0.75)
c Draw horizontal secondary axis.
    DEL=0.1*(YW2N-YW1N)
call PGMOVE(XG(IXI),YW1-DEL)
call PGDRAW(XG(IX2),YW1-DEL)
do I=IXI,IX2
call PGMOVE(XG(I),YW1-DEL)
call PGDRAW(XG(I),YW1-0.875*DEL)
if(MOD(I,20).eq.0)then
    if(I .lt. 10)then
        write(CH(1:1),'(i1)')I
        LCH=1
    else if(I .ge. 10.and. I .lt. 100)then
        write(CH(1:2),'(i2)')I
        LCH=2
    else if(I .ge. 100)then
        write(CH(1:3),'(i3)')I
        LCH=3
    end if
    call PGPTXT(XG(I),YW1-1.2*DEL,0.0,0.5,CH(I:LCH))
end if
end do
c Draw vertical secondary axis.
    DEL=0.08*(XW2N-XW1N)
call PGMOVE(XW1-DEL,YG(IY1))
call PGDRAW(XW1-DEL,YG(IY2))
do I=IY1,IY2
call PGMOVE(XW1-DEL,YG(I))
call PGDRAW(XW1-0.9*DEL,YG(I))
if(MOD(I,20).eq.0)then
    if(I .lt. 10)then
        write(CH(1:1),'(i1)')I
        LCH=1
    else if(I .ge. 10.and. I .lt. 100)then
        write(CH(1:2),'(i2)')I
        LCH=2
    else if(I .ge. 100)then
        write(CH(1:3),'(i3)')I
        LCH=3
    end if
call PGPTXT(XW1-1.05*DEL,YG(I),90.0,0.5,CH(I:LCH))
end if
end do
call PGEND
if(IOUT .eq. 1)then
  write(6,*)’ Enter 0 to quit, 1 for Postscript file’
  read(5,*)I
  if(I .eq. 1)then
    IOUT=2
    goto 10
  end if
else
  write(6,*)’ Postscript output is in file pgplot.ps’
end if
stop
end
H  ROTTTILT Input File for the XV15 Full Span Aircraft

*****General Input File for the XV15 Full Span Aircraft**************

_DEBUG_ITEROP_ITEROPL_LUNWRL_LUNWR_LUNSN_LUNPL_LUNTB_LUNCT_
false , 200 , 740 , 6 , 8 , 9 , 10 , 17 , 18
_DETAL__U____V____P____W____str Fun____GAM____RO____
TRUE , TRUE , TRUE , TRUE , TRUE , FALSE , FALSE , FALSE
_LROCON_LMUOCN_LSOILD_LDATPR____LUGRID_ 
TRUE , TRUE , TRUE , TRUE , TRUE
_LGEDMP__LUOCFP__LVOCFP__LWOCFP__LPCDFP__LTCDFP__LMLTOPF__LFIDFPL_PLT3D
FALSE , FALSE , FALSE , FALSE , FALSE , TRUE , FALSE , TRUE
_MAX_ITERATIONS__ITEMOD____NU____NW____NP____NT____
850 , 10 , 2 , 2 , 2 , 6 , 75
__RHO_ (0.002377) __MU (3.719E-07) __RELAX-X__RELAX-Y____RELAX-Z__
0.0023700 , 3.719E-07 , 0.10 , 0.10 , 0.10
_--UINF____VINF____WINF____PINF____TINF____REAL-GAS-CON____GAMMA
0.0d0 , 0.0d0 , -1.0d0 , 2116 , 419.0d0 , 1719.0d0 , 1.4d0
__LINI__LOUTI__LINJ__LOUTJ__LIP__LJP__LKP__
TRUE , TRUE , TRUE , TRUE , TRUE , TRUE , TRUE , TRUE
__IPREF__JPREF__KPREF__WINI__WOUTI__WINJ__WOUTJ__WINK__WOUTK
2 , 64 , 59 , FALSE , FALSE , FALSE , FALSE , FALSE , FALSE
___DRELXU__DRELXV__DRELXW__DRELXT__DRELXP__MODREL_RELMAX
0.05 , 0.05 , 0.05 , 0.01 , 0.00 , 0.50
_LCS8__LCSRSEC__LCSMSEC__LCLCD__LCLCOR__LROTBOD
true , false , true , true , true
Velocity check__IUCK____JUCK____KWCK____XUCK____YVCK____ZWCK
26 , 22 , 17 , 0.0 , 0.0 , 0.0
_Number_of_Rotors____DATA SET REFERENCE =14 NASA(TM X-952)__IBOUND
2 , 4
Name of airfoil used
MODIFIED VM SECTIONS

*************** Data for rotor # 1 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
__Airfoil___# blades___Press dist(r/R)__
1 , 3.000 , 0.250d0
Tip speed_Rotor radius_Hub radius(r/R)_Hinge Offse(r/R)_clock rot_
771.12d0 , 12.5 , 0.1353 , 0.0357 , FALSE
Number Of Reference Rotor radius(r/R){max has been set to 15}f_______
44
Reference Rotor radius(r/R)---------------------------------------------
.100 , .125 , .150 , .175,
.200 , .225 , .240 , .250 , .260 , .275 , .290 , .300 , .310 , .325,
.340 , .350 , .375 , .400 , .425 , .475 , .500 , .525 , .575 , .600,
.625 , .650 , .675 , .700 , .750 , .775 , .800 , .825 , .850 , .875,
.900 , .915 , .925 , .940 , .950 , .960 , .970 , .980 , .990 , .995
CL cor(r/R)__Ref Twist__# Blade Div__Azl cor__# azi loc__conv fac__
0.9d0 , 0.0d0 , 100 , 0.0d0 , 720 , 0.083333333
# harmonic pit__# harmonic flap_FLAPPING__flap geo corr
2 , 3 , FALSE , FALSE
<< Harmonic_pitching_coefficients--Positive series-->>
0 , 8.175100 , 0.000000
1 , 0.000000
74
<<<Harmonic_flapping_coefficients--Positive series-->>>
0 0.0000 0.0000
1 0.0000 0.0000
2 0.0000 0.0000
# data_points
21

nn ..... r/R ....... Chord/RAD .... CL deflec___T/chrd___Twist____
1  .1353, 0.00D, .13780, 0.00E, 0.12E0,  29.500
2  .2035, 0.00D, .13780, 0.00E, 0.12E0,  25.500
3  .2700, 0.00D, .13780, 0.00E, 0.12E0,  20.750
4  .3250, 0.00D, .13470, 0.00E, 0.12E0,  18.625
5  .3750, 0.00D, .13470, 0.00E, 0.12E0,  15.875
6  .4250, 0.00D, .13470, 0.00E, 0.12E0,  13.500
7  .4750, 0.00D, .13470, 0.00E, 0.12E0,  11.250
8  .5250, 0.00D, .13470, 0.00E, 0.12E0,   9.000
9  .5750, 0.00D, .13470, 0.00E, 0.12E0,   6.875
10 .6250, 0.00D, .13470, 0.00E, 0.12E0,   4.875
11 .6750, 0.00D, .13370, 0.00E, 0.12E0,   2.813
12 .7250, 0.00D, .12330, 0.00E, 0.12E0,   0.944
13 .7750, 0.00D, .10990, 0.00E, 0.12E0,  -0.833
14 .8200, 0.00D, .10220, 0.00E, 0.12E0,  -2.354
15 .8600, 0.00D, .09710, 0.00E, 0.12E0,  -3.771
16 .9000, 0.00D, .09080, 0.00E, 0.12E0,  -5.216
17 .9350, 0.00D, .08330, 0.00E, 0.12E0,  -6.500
18 .9600, 0.00D, .07170, 0.00E, 0.12E0,  -7.438
19 .9750, 0.00D, .05730, 0.00E, 0.12E0,  -8.000
20 .9850, 0.00D, .04770, 0.00E, 0.12E0,  -8.350
21 .9950, 0.00D, .03810, 0.00E, 0.12E0,  -8.700

*************** Data for rotor # 2 *****************

_Airfoil___# blades___Press dist(r/R)___
1  ,  3.00D, 0.250D0

Tip speed_Rotor radius_Hub radius(r/R)_Hinge Offse(r/R)_clock rot__
771.12D0, 12.5 , 0.1353 , 0.0357 , TRUE

Number Of Reference Rotor radius(r/R){max has been set to 15}f______
44
Reference Rotor radius(r/R)___________________________________________
.100, .125, .150, .175,
.200, .225, .240, .250, .260, .275, .290, .300, .310, .325,
.340, .350, .375, .400, .425, .475, .500, .525, .575, .600,
.625, .650, .675, .700, .750, .775, .800, .825, .850, .875,
.900, .915, .925, .940, .950, .960, .970, .980, .990, .995

CL cor(r/R)___Ref Twist__# Blade Div___Azi cor___# azi loc___conv fac___
0.9D, -0.0D, 100, 0.0D0 , 720 , 0.083333333

# harmonic pit___# harmonic flap_FLAPPING__flap geo corr
2 , 3 , FALSE , FALSE

<<<Harmonic_pitching_coefficients--Positive series-->>>
0 8.1751D0, 0.000D0
1 0.000D0, 0.000D0

<<<Harmonic_flapping_coefficients--Positive series-->>>
0 0.0000 0.0000
1 0.0000 0.0000
2 0.0000 0.0000

# data_points
21
nn\_\_r/R\_\_deflec\_\_Chord/RAD\_\_CL\_\_\_\_T/chrd\_\_\_\_Twist\_\_\_

1. .1353, 0.0D0, .13780, 0.0E0, 0.12E0, 29.500
2. .2035, 0.0D0, .13780, 0.0E0, 0.12E0, 25.500
3. .2700, 0.0D0, .13780, 0.0E0, 0.12E0, 20.750
4. .3250, 0.0D0, .13470, 0.0E0, 0.12E0, 18.625
5. .3750, 0.0D0, .13470, 0.0E0, 0.12E0, 15.875
6. .4250, 0.0D0, .13470, 0.0E0, 0.12E0, 13.500
7. .4750, 0.0D0, .13470, 0.0E0, 0.12E0, 11.250
8. .5250, 0.0D0, .13470, 0.0E0, 0.12E0, 9.000
9. .5750, 0.0D0, .13470, 0.0E0, 0.12E0, 6.875
10. .6250, 0.0D0, .13470, 0.0E0, 0.12E0, 4.875
11. .6750, 0.0D0, .13370, 0.0E0, 0.12E0, 2.813
12. .7250, 0.0D0, .12330, 0.0E0, 0.12E0, 0.944
13. .7750, 0.0D0, .10990, 0.0E0, 0.12E0, -0.833
14. .8200, 0.0D0, .10220, 0.0E0, 0.12E0, -2.354
15. .8600, 0.0D0, .09710, 0.0E0, 0.12E0, -3.771
16. .9000, 0.0D0, .09080, 0.0E0, 0.12E0, -5.216
17. .9350, 0.0D0, .08330, 0.0E0, 0.12E0, -6.500
18. .9600, 0.0D0, .07170, 0.0E0, 0.12E0, -7.438
19. .9750, 0.0D0, .05730, 0.0E0, 0.12E0, -8.000
20. .9850, 0.0D0, .04770, 0.0E0, 0.12E0, -8.350
21. .9950, 0.0D0, .03810, 0.0E0, 0.12E0, -8.700

__no of rad stations for airfoil table_____ DATA IS DIFF for non c81__
6
__station at which airfoil table is considered___
0.00, 0.17, 0.3, 0.9, 0.95, 1.00
__logical unit for the airfoil tableS___
28
__The name of the file corresponding to lun = 28___
newcuff.c81
__The name of the file corresponding to lun = 28___
airtilt4.c81
__The name of the file corresponding to lun = 28___
airtilt4.c81
__The name of the file corresponding to lun = 28___
airtilt2.c81
__The name of the file corresponding to lun = 28___
airtilt3.c81
LWING\_\_LNACEL\_\_LV22BD
true, true, true
LTRIM\_\_idtrim\_\_ctreq\_\_DELCOLM\_\_TRIMTOL
true, 10, 0.0107420d0, 2d0, 0.000001

76
## Hess Format Sample Data

*********** Sample Hess format file for the XV15 Fuselage***********

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>168</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td>.11964832E+02</td>
<td>-.35352356E-08</td>
<td>.84840379E+01</td>
</tr>
<tr>
<td>.1207737E+02</td>
<td>-.35352356E-08</td>
<td>.83474809E+01</td>
</tr>
<tr>
<td>.12231729E+02</td>
<td>-.35352356E-08</td>
<td>.82237974E+01</td>
</tr>
<tr>
<td>.12411149E+02</td>
<td>-.35352356E-08</td>
<td>.81239555E+01</td>
</tr>
<tr>
<td>.12592197E+02</td>
<td>-.35352356E-08</td>
<td>.80379855E+01</td>
</tr>
<tr>
<td>.12790360E+02</td>
<td>-.35352356E-08</td>
<td>.79569148E+01</td>
</tr>
<tr>
<td>.12988524E+02</td>
<td>-.35352356E-08</td>
<td>.78974983E+01</td>
</tr>
<tr>
<td>.13192248E+02</td>
<td>-.35352356E-08</td>
<td>.78379662E+01</td>
</tr>
<tr>
<td>.13399462E+02</td>
<td>-.35352356E-08</td>
<td>.77829827E+01</td>
</tr>
<tr>
<td>.13607160E+02</td>
<td>-.35352356E-08</td>
<td>.77362173E+01</td>
</tr>
<tr>
<td>.13820726E+02</td>
<td>-.35352356E-08</td>
<td>.76899111E+01</td>
</tr>
<tr>
<td>.14034295E+02</td>
<td>-.35352356E-08</td>
<td>.76466233E+01</td>
</tr>
<tr>
<td>.14249661E+02</td>
<td>-.35352356E-08</td>
<td>.76060746E+01</td>
</tr>
<tr>
<td>.14466899E+02</td>
<td>-.35352356E-08</td>
<td>.75664590E+01</td>
</tr>
<tr>
<td>.14684138E+02</td>
<td>-.35352356E-08</td>
<td>.75301205E+01</td>
</tr>
<tr>
<td>.14904112E+02</td>
<td>-.35352356E-08</td>
<td>.74962837E+01</td>
</tr>
<tr>
<td>.15123381E+02</td>
<td>-.35352356E-08</td>
<td>.74632632E+01</td>
</tr>
<tr>
<td>.15340245E+02</td>
<td>-.35352356E-08</td>
<td>.74329212E+01</td>
</tr>
<tr>
<td>.1556381E+02</td>
<td>-.35352356E-08</td>
<td>.74050202E+01</td>
</tr>
<tr>
<td>.15771334E+02</td>
<td>-.35352356E-08</td>
<td>.73803484E+01</td>
</tr>
<tr>
<td>.15981738E+02</td>
<td>-.35352356E-08</td>
<td>.73672958E+01</td>
</tr>
<tr>
<td>.1619504E+02</td>
<td>-.35352356E-08</td>
<td>.73691177E+01</td>
</tr>
<tr>
<td>.1639601E+02</td>
<td>-.35352356E-08</td>
<td>.73709327E+01</td>
</tr>
<tr>
<td>.1660363E+02</td>
<td>-.35352356E-08</td>
<td>.73727495E+01</td>
</tr>
<tr>
<td>.16811307E+02</td>
<td>-.35352356E-08</td>
<td>.73745700E+01</td>
</tr>
<tr>
<td>.17019323E+02</td>
<td>-.35352356E-08</td>
<td>.73763940E+01</td>
</tr>
<tr>
<td>.17227781E+02</td>
<td>-.35352356E-08</td>
<td>.73782219E+01</td>
</tr>
<tr>
<td>.17436751E+02</td>
<td>-.35352356E-08</td>
<td>.73800542E+01</td>
</tr>
<tr>
<td>.17643641E+02</td>
<td>-.35352356E-08</td>
<td>.73818684E+01</td>
</tr>
<tr>
<td>.17851419E+02</td>
<td>-.35352356E-08</td>
<td>.73836903E+01</td>
</tr>
<tr>
<td>.18060514E+02</td>
<td>-.35352356E-08</td>
<td>.73855238E+01</td>
</tr>
<tr>
<td>.18271396E+02</td>
<td>-.35352356E-08</td>
<td>.73873729E+01</td>
</tr>
<tr>
<td>.18509707E+02</td>
<td>-.35352356E-08</td>
<td>.73894625E+01</td>
</tr>
<tr>
<td>.18750294E+02</td>
<td>-.35352356E-08</td>
<td>.73915712E+01</td>
</tr>
<tr>
<td>.18992769E+02</td>
<td>-.35352357E-08</td>
<td>.73936983E+01</td>
</tr>
<tr>
<td>.19237920E+02</td>
<td>-.35352356E-08</td>
<td>.73958479E+01</td>
</tr>
<tr>
<td>.20696137E+02</td>
<td>-.35352356E-08</td>
<td>.74086344E+01</td>
</tr>
<tr>
<td>.20904720E+02</td>
<td>-.35352356E-08</td>
<td>.74104634E+01</td>
</tr>
<tr>
<td>.21113045E+02</td>
<td>-.35352356E-08</td>
<td>.74122901E+01</td>
</tr>
<tr>
<td>.21321396E+02</td>
<td>-.35352356E-08</td>
<td>.74141171E+01</td>
</tr>
<tr>
<td>.21529756E+02</td>
<td>-.35352356E-08</td>
<td>.74159441E+01</td>
</tr>
<tr>
<td>.21738120E+02</td>
<td>-.35352356E-08</td>
<td>.74177711E+01</td>
</tr>
<tr>
<td>.22015920E+02</td>
<td>-.35352356E-08</td>
<td>.74202070E+01</td>
</tr>
</tbody>
</table>
A CFD solver has been developed to provide the time averaged details of the fountain flow typical for tiltrotor aircraft in hover. This Navier-Stokes solver, designated as ROTTILT, assumes the 3-D fountain flowfield to be steady and incompressible. The theoretical background is described in this manual. In order to enable the rotor trim solution in the presence of tiltrotor aircraft components such as wing, nacelle, and fuselage, the solver is coupled with a set of trim routines which are highly efficient in CPU and suitable for CFD analysis. The Cartesian grid technique utilized provides the user with a unique capability for insertion or elimination of any components of the bodies considered for a given tiltrotor aircraft configuration. The flowfield associated with either a semi or full-span configuration can be computed through user options in the ROTTILT input file. Full details associated with the numerical solution implemented in ROTTILT and assumptions are presented. A description of input surface mesh topology is provided in the appendices along with a listing of all preprocessor programs. Input variable definitions and default values are provided for the V22 aircraft. Limited predicted results using the coupled ROTTILT/WOPWOP program for the V22 in hover are made and compared with measurement. To visualize the V22 aircraft and predictions, a preprocessor graphics program GNU-PLOT3D was used. This program is described and example graphic results presented.