

A Guide to the Use of the Pressure Disk Rotor Model as Implemented in INS3D-UP

Mark S. Chaffin
ViGYAN, Inc.
Hampton, Virginia 23666

Summary

This is a guide for the use of the pressure disk rotor model that has been placed in the incompressible Navier-Stokes code INS3D-UP. The pressure disk rotor model approximates a helicopter rotor or propeller in a time averaged manner and is intended to simulate the effect of a rotor in forward flight on the fuselage or the effect of a propeller on other aerodynamic components. The model uses a modified actuator disk that allows the pressure jump across the disk to vary with radius and azimuth. The cyclic and collective blade pitch angles needed to achieve a specified thrust coefficient and zero moment about the hub are predicted. The method has been validated with experimentally measured mean induced inflow velocities as well as surface pressures on a generic fuselage. Overset grids, sometimes referred to as Chimera grids, are used to simplify the grid generation process. The pressure disk model is applied to a cylindrical grid which is embedded in the grid or grids used for the rest of the configuration. This document will outline the development of the method, and present input and results for a sample case.

Nomenclature

Note: Unless specified, the following variables are made nondimensional as follows: lengths are made nondimensional by radius, areas are made nondimensional by radius², velocities are made nondimensional by U_∞ , densities are made nondimensional by ρ_∞ , pressures are made nondimensional by $\rho_\infty U_\infty^2$, forces are made nondimensional by $\rho_\infty (U_\infty \text{radius})^2$, and torques are made nondimensional by $\rho_\infty U_\infty^2 \text{radius}^3$.

A_0	Collective pitch angle, radians
A_1, B_1	Cyclic pitch angles, radians
c	Rotor blade chord
C_T	Thrust Coefficient, $\frac{\text{Thrust}}{\rho \pi R^2 (V_{tip})^2}$
dA	Elemental area on rotor disk
dD	Sectional drag of rotor blade
dL	Sectional lift of rotor blade

dQ	Sectional torque of rotor blade
dr	Elemental span
dT	Sectional thrust of rotor blade
$d\psi$	Azimuthal spacing of grid, radians
N	Number of blades
\bar{n}	Unit vector normal to blade path
\bar{p}	Unit vector parallel to blade path
Δp	Pressure jump
R	Rotor radius
r	Radius location
\bar{r}	Radius vector
\tilde{r}	Radius unit vector
U_∞	Freestream velocity, ft/sec
V	Velocity magnitude relative to rotor blade
∇	Velocity vector
v_i	Induced component of velocity normal to the rotor disk
Δv	Tangential velocity jump
V_{tip}	Rotor blade tip velocity
V_ψ	Azimuthal component of velocity relative to rotor blade
V_n	Normal component of velocity relative to rotor blade
α_e	Effective angle of attack, radians
α_g	Geometric angle of attack, radians
α_i	Induced angle of attack, radians
α_t	Geometric blade twist, radians
λ_i	Mean induced inflow ratio, $\frac{v_i}{V_{tip}}$
μ	Advance ratio, $\frac{U_\infty}{V_{tip}}$
ρ	Local air density
ρ_∞	Free stream air density, slugs/ft ³
ψ	Azimuth, radians
$\bar{\Omega}$	Shaft rotation vector, radians/sec

Introduction

The flow field in which rotorcraft operate is dominated by vortical flow in the rotor wake and

viscous effects at the fuselage. Early analytical work focused on development of models to predict the performance of the isolated rotor without consideration of the effect of the fuselage. However, the fuselage can have a significant effect on the overall performance of the rotorcraft.

The goal of this code is to model the effect of the rotor on the fuselage and also give an indication of the flow field at the rotor. By accepting a time averaged solution for the rotor and wake, the complexity of the problem is reduced, while still yielding a time averaged rotor influence on the fuselage. With this method it is not necessary to generate grids for the individual rotor blades, reducing the cost in terms of operation and grid development and allowing quick changes in the rotor geometry to be made without generating new grids. This method will also provide the user with an estimate of the flow at the rotor and the rotor load distribution.

The method presented here uses overset, or Chimera, grids (Ref. 1) to allow the grids for the rotor disk and the fuselage to be generated independently. The incompressible Navier-Stokes code INS3D-UP (Ref. 2) is used to provide a flow solution. While the flow field relative to the rotor blade motion is certainly compressible, it is assumed that the primary effect of this is in the magnitude of the forces generated by the rotor blades. Thus, the effects of compressibility can be included when determining the forces generated by the rotor blades.

Pressure Disk Model

The method presented in this paper simulates the helicopter rotor by imposing boundary conditions at the surface swept by the rotor in space. The boundary conditions are applied on adjacent planes of a cylindrical grid. At this surface, a disk, two separate boundary conditions are imposed. The top of the rotor disk surface is modelled with an outflow boundary condition. On this plane, the pressure and velocities tangential to the disk are extrapolated from the flow field above the rotor. The velocity normal to the disk is specified to be equal to the normal velocity at the bottom of the disk to maintain continuity. The lower disk surface is modelled with an inflow boundary condition. On this plane, the pressure and tangential velocities are based on their values at the top of the disk and the normal component of velocity is extrapolated from below the disk. The effect of the rotor is imposed by the difference in pressure and tangential velocities between the upper and lower disk surfaces.

The first step in calculating the rotor model boundary condition is to determine the velocity of the flow relative to a rotor blade laying in the rotor disk plane. Once the flow at the disk is known, it is possible to determine the forces generated by a rotor blade at each point on the disk. Blade

element theory is used to compute the rotor forces.

The flow solver provides the velocity, \bar{V} , relative to the fixed frame axis at every point on the disk grid. Using Fig. 1 as a guide, it is possible to determine the velocity at point B on a rotor blade whose path coincides with the disk grid. Using the grid definition, the unit vector normal to the blade path, \bar{n} , can be determined. $\bar{V} \cdot \bar{n}$ is the velocity of the air in the direction normal to the blade path. \bar{r} is the vector from the shaft center to the point B . Because \bar{r} is perpendicular to \bar{n} , the vector parallel to the path of point B , \bar{p} , can be determined from \bar{r} and \bar{n} .

$$\bar{p} = \tilde{r} \times \bar{n}; \quad \tilde{r} = \frac{\bar{r}}{|\bar{r}|} \quad (1)$$

$\bar{V} \cdot \bar{p}$ is the component of the freestream velocity in the direction parallel to the blade path at point B . $(\bar{r} \times \bar{\Omega}) \cdot \bar{p}$ is the velocity due to the blade rotation in the direction parallel to the blade path at point B . The total velocity parallel to the blade path then becomes $\bar{V} \cdot \bar{p} + (\bar{r} \times \bar{\Omega}) \cdot \bar{p}$.

Using the velocity components normal and parallel to the blade path, the induced angle of attack, α_i , can be determined.

$$\alpha_i = \text{atan}\left(\frac{\bar{V} \cdot \bar{n}}{\bar{V} \cdot \bar{p} + (\bar{r} \times \bar{\Omega}) \cdot \bar{p}}\right) \quad (2)$$

The effective angle of attack, α_e , at point B is

$$\alpha_e = \alpha_g - \alpha_i. \quad (3)$$

The relationship between α_g , α_e , and α_i is shown in Fig. 2. α_g is the geometric angle of attack and is determined by blade twist and the pitch of the blade. The pitch of the rotor blade relative to the shaft of the rotor is given by

$$\alpha_g = A_0 - A_1 \cos \psi - B_1 \sin \psi + \left(0.75 - \frac{r}{R}\right) \alpha_t. \quad (4)$$

Once the velocity relative to a blade element has been determined, the lift and drag generated by the blade element can be calculated. The grid used for the rotor is cylindrical and a planform view is shown in Fig. 3. Rotor geometry will specify that the rotor blade passing through the area dA has a chord of c . The grid definition will determine the value of the elemental span, or dr . With the effective angle of attack calculated, values for the lift and drag coefficients can be determined. These values could come from airfoil tables, other computational methods, or other simple empirical models. The user is given two options for calculating the blade forces. The first option is to set the lift coefficient to $2\pi\alpha_e$. With this option, once the lift coefficient reaches a user specified limit, it is not allowed to increase. While this is a rather simple stall model, it has

proven reasonably accurate. The drag coefficient is also determined with a simple empirical model defined as $0.006 + 0.0133c_l^2$. In the second option, the model uses airfoil tables to determine the forces. Because of the added expense of searching and interpolating the data in the airfoil tables, this option is slightly more expensive to use. If airfoil tables are to be used, they must be in the C81 format (Ref. 3). The variable **model** in the input for the rotor boundary condition controls this option. The amount of lift and drag acting on the section of the rotor blade passing over the area dA is

$$dL = \frac{1}{2}\rho V^2 c_l c dr \quad (5)$$

$$dD = \frac{1}{2}\rho V^2 c_d c dr. \quad (6)$$

In reality, the area, dA , would not see the effects of the rotor blade continuously. For this reason, the added pressure and velocity jumps are scaled by a time factor, $N\frac{d\psi}{2\pi}$, where N is the number of rotor blades and $d\psi$ is the grid azimuth resolution. This is the fraction of time a rotor blade spends over the area dA . With this time factor, dL , dD , and α_i , the forces can be resolved into the thrust and torque components.

$$dT = N\frac{d\psi}{2\pi} (dL \cos(\alpha_i) - dD \sin(\alpha_i)) \quad (7)$$

$$dQ = N\frac{d\psi}{2\pi} (dL \sin(\alpha_i) + dD \cos(\alpha_i)) \quad (8)$$

The pressure jump boundary condition then becomes

$$\Delta p = \frac{dT}{dA}. \quad (9)$$

The torque boundary condition is modelled as a jump in the tangential velocity. The magnitude of this jump is based on the induced velocity, v_i , and the ratio of torque to thrust. The induced velocity was calculated when the velocity relative to the rotor blade was determined.

$$\Delta v = v_i \frac{dQ}{dT \cdot r} \quad (10)$$

In practice, the torque boundary condition effect has been smaller than expected and in some cases its use has led to nonphysical solutions. Other, potentially more effective, ways of imposing the torque boundary condition are being studied. It is not implemented in the current version of the model, but it is discussed here for completeness.

Trim Loop

At user specified iterations, the model calculates the correct collective and cyclic pitch angles in order to achieve the desired thrust coefficient and eliminate moments about the hub. In the calculation of the moments, the flapping hinge offset is assumed to be zero. The trimming routine

assumes linear relationships between the thrust coefficient and the collective pitch angle, and between the hub moments and the cyclic pitch angles. A simple linear relationship is assumed to exist between the control angle and the error that is being minimized. Figure 4 shows the general scheme used to implement the trim loop. The first step is to calculate the pressure jump used for the boundary conditions. This can be integrated over the disk surface to determine the thrust coefficient and the flapping moments about the hub. Once this is done, the trim subroutines search for the proper pitch angles. The pressure jump is then recalculated using these new angles. This loop continues until the new angles match the angles from the previous loop, or a user specified number of iterations has passed. The Navier-Stokes solution is then advanced with the new angles. When several trim cycles pass with no change in the control angles, the rotor model is assumed to be converged. This does not necessarily mean that the overall flow field solution has converged.

Appendix A discusses the variables used by the rotor model boundary conditions contained in the file *'rotor.inp'*. Several of these variables are used to control the trimming operation. Initially, the collective and cyclic angles can be set by the user. If the user does not wish the model to trim the rotor, **itrmst** should be made sufficiently large so that the trimming routines are not called. It is not possible to change the collective and cyclic pitch angles at a restart. The only means of adjusting these angles on a restart is through the trim routines. The variable **iramp** is used to specify the number of iterations to ramp up the pressure jump. This can be used if the required pressure jump is so large it causes difficulty with the convergence of the code. The variable **iclcfrc** controls the intervals at which the rotor forces (C_T and hub moments) are calculated and printed. This is independent of the trimming operation, but does provide an insight to the status of the solution. The variable **inttrm** controls how often the trim subroutines are called. For difficult cases, **inttrm** should not be too small as it is best to allow the solution to “calm down” before trimming again. The variables **altol**, **bltol**, and **cttol** are the tolerances used for the trimming. If the trim routines determine the change in the hub moments and thrust coefficient are smaller than these values, the trim routines are exited. Setting **altol** and **bltol** to '-888' will prevent the cyclic pitch angles from being adjusted when the boundary condition is used to model a propeller. The variables **alinc**, **blinc** and **ctinc**, in degrees, are used to keep the trimming routines from diverging during the early attempts to trim by setting the maximum change in the control angles.

Flapping

The flapping motion of the rotor is not calculated by the code. However, if the flapping motion is known the code can account for the first two harmonics of flapping by adding the flapping velocity to the velocity normal to the blade path. Coning is handled by using a predetermined coned grid geometry.

Computational Method

As stated earlier, the pressure disk rotor model is implemented in version 1.8-19 of INS3D-UP. While this document is not intended to be a manual for INS3D, there are some specific steps that can be taken to increase the chances of success in the operation of the code with the rotor model. A sample input for the isolated rotor case is included in Appendix B. The most important variables to the operation of INS3D with the rotor model are **njswp**, **nkswp**, and **nlswp** in namelist **zonin**. **nkswp** should be set to zero, while typically, **njswp** and **nlswp** are set to a value of 5-10. A higher number of sweeps can aid in the convergence, but the cost of running the code is directly related to the number of sweeps performed. It is also important that **kpr** is set to 'T'. This variable controls the periodic boundary condition in the k direction. Finally, it is not possible to change the angle of attack during a restart.

Grids

The pressure disk model uses a cylindrical grid, an example of which is shown in Figure 5. The generation of this grid can be done in a relatively simple code. Using the jkl convention, the plane of the disk should lie on a constant j-plane. The k=1 plane is located on the 0° azimuth. Uniform spacing must be used for the distribution in the k direction. The k=k_{max} plane is located at the $\left(\frac{k_{\max} - 1}{k_{\max}}\right)360^\circ$ azimuth. The l direction is such that l=1 corresponds to the center of the grid and l=l_{max} is the outer cylinder of the grid. Note that the l=1 plane is a singularity. This requires the use of the axis boundary condition. The rotor model boundary conditions are written so that positive thrust will result in thrust in the -j direction. In other words, the j value at the top of the disk is less than the j value at the bottom of the disk.

As stated earlier, overset grids are used to simplify the grid generation process. With overset, or Chimera, grids, complex geometries are broken down into simpler component geometries. Individual grids can then be generated for these simpler shapes. Then the collection of grids can be combined to produce the final overset grid system.

Figure 6 shows an overset grid system for an isolated rotor case. Two grids were used to form this system. The first grid is the cylindrical grid used for the rotor model. The second grid is a simple cartesian grid that serves to move the far field boundaries sufficiently far from the rotor disk. The cartesian grid also makes applying the far field boundary conditions easier. PEGSUS 4.0 (Ref. 4) was used to combine the individual grids into the single overset grid system. It is assumed that the user of this code has experience with PEGSUS.

A problem arises when using PEGSUS with the cylindrical grid used for the rotor model. Due to the manner in which INS3D-UP treats periodic boundary conditions, the cylindrical grid is not completely closed, *i.e.*, a slice is removed. When PEGSUS performs a search to determine where to cut holes in the outer cartesian grid, it will fail to cut holes at the missing slice. It is important to remember that the gap in the cylindrical grid is not seen by INS3D-UP. Therefore, something must be done to cut holes in the cartesian grid. This problem is solved by adding an additional k plane to complete the cylindrical grid and cut holes in the cartesian grid. In other words, on this new cylindrical grid, the $k=1$ and the $k=k_{max}$ lines coincided. PEGSUS is then run using this new cylindrical grid. The file containing the interpolation coefficients is then searched to find instances where the k_{max} grid index is used. In these cases, the k_{max} value is replaced with 1. The new cylindrical grid is then replaced with the old cylindrical grid when the final grid file is generated.

The cylindrical rotor disk grid should be generated with the experimentally determined, or estimated value of coning. During the validation of the method, although several cases requiring slightly different coning angles were used, only one coning angle was used. Experience has shown that the effect of coning on the solution is small and having a grid that is 1 degree off in coning makes no noticeable difference in the calculation.

The sample isolated rotor grid system uses a cylindrical rotor disk grid with 58 points in the vertical direction, 50 points in the circumferential direction, and 68 radial points. The outer cartesian grid has 65 points in the streamwise direction, 65 points in the lateral direction and 50 points in the vertical direction. Spacing in the vertical direction is approximately $0.0003R$ at the disk surface. Geometric spacing is used above and below the rotor in the vertical direction. A cosine distribution is used for the radial spacing on the disk with geometric spacing used off the disk. The total number of points was 408,450 and 16 million words of memory were required for the sample isolated rotor computation.

Implementation of Boundary Conditions

The boundary conditions added to INS3D-UP by the pressure disk model are 101 and 102. Boundary condition 101 is used for the top of the disk, 102 is used for the bottom of the disk. The top and bottom planes of the disk are separated by a plane with a boundary condition of -1 (iblanke). This is an INS3D-UP requirement. Because the first plane in the l direction of the cylindrical grid is singular, the axis boundary condition (71) must be used on this plane. The boundary condition file *bcmain.dat* for the isolated rotor test case is shown below. The rotor boundary conditions must be listed last in *bcmain.dat*. This does not mean that the grid used for the rotor boundary conditions has to be the last grid in the system. INS3D allows the boundary conditions to be listed in any order. Appendix C shows a sample of the input file *rotor.inp* used by the rotor model boundary condition.

```

c*****
c  boundary conditions for isolated rotor
c-----
c  ibcval      zone      jbeg  jend      kbeg  kend      lbeg  lend
c-----
c  cartesian grid boundary conditions
c      31      1      65   65      1   65      1   50
c  rotor grid boundary conditions
c      71      2      1   58      1   50      1   1
c      101     2      28   28      1   50      9   50
c      -1      2      29   29      1   50      9   50
c      102     2      30   30      1   50      9   50
c*****

```

Sample Results

The pressure disk method was tested with the rotor alone as well as with the addition of a generic fuselage at five advance ratios, μ , ranging from 0.15 to 0.40. Results of these tests were compared with experimental induced inflow measurements and trimmed control angles (Refs. 5-9). A more extensive discussion of these results can be found in Ref. 10. The trimmed control angles for the isolated rotor test case with both airfoil models are compared with the experimental values for the 0.15 advance ratio case reported by Ghee and Elliot (Ref. 11) in Table 1.

Table 1: Computed Trim Conditions

Advance Ratio, μ	Experiment	Isolated Rotor w/ $2\pi\alpha$ model	Isolated Rotor w/airfoil tables
0.15	$A_0=6.55^\circ$	$A_0=6.76^\circ$	$A_0=5.76^\circ$
	$A_J=-1.39^\circ$	$A_J=-2.27^\circ$	$A_J=-2.24^\circ$
	$B_J=1.99^\circ$	$B_J=2.42^\circ$	$B_J=1.80^\circ$

Code Output

Two types of information about the rotor is included in the standard output from INS3D-UP. The calculated thrust coefficients and the lateral and longitudinal hub moments are printed every **iclcfrc** iterations. This information is useful for determining the rate of convergence of the trimming routines. Typically, **iclcfrc** is set equal to **inttrm**. The other information also relates to the trim routines. At every **inttrm** iterations, the code prints the number of iterations needed by the trim routines, as well as the values for A_0 , A_1 , and B_1 . At the start of the calculation, it is not unusual for the number of trim iterations to be equal to **itloop**. As the thrust coefficient approaches the value specified in **cthrust**, the number of iterations should decrease, eventually becoming 0. A sample of the standard output is included in Appendix D.

The code will also give information at the disk. This information is contained in the file *fort.35*. This file is printed every **np3d** iterations and at the end of each run. The file is formatted for use with the Tecplot®(Ref. 12) plotting software, but the format is easily translatable for use with other plotting software. The file contains the following information for each grid point on the disk: the x and y coordinates, the sectional lift and drag coefficients, angle of attack, induced angle of attack, mean induced inflow ratio at the $j=j_{inflow}$ plane, the pressure jump across the disk and the two components of velocity. The velocities, V_ψ and V_n , are made nondimensional by the freestream velocity. The dimensional sectional lift can be calculated as

$$L = \frac{1}{2} c_l \left(V_\psi^2 + V_n^2 \right) chord \cdot \rho_\infty \cdot U_\infty^2 \quad , \quad (11)$$

where c_l is the lift coefficient, *chord* is the physical chord of the rotor blade, ρ_∞ is the free stream density, and U_∞ is the free stream velocity.

Computational Expense

The number of iterations required for the rotor model to converge depends on the conditions which are being modelled. The basic scheme for using the rotor model is to start with a small amount of collective and no cyclic controls. After some time, about 10-30 iterations depending on the severity of the case, the trim routines are called and the collective will start to increase to achieve the desired thrust coefficient. The trim routines will also adjust the cyclic controls to try to maintain zero moments at the hub. The amount of change during a single trim cycle is limited for stability reasons. Experience shows that this method arrives at the desired conditions more quickly than starting at an initial guess of the final conditions. This is partly due to the inability of

the code to handle abrupt changes in the rotor boundary conditions from one iteration to the next. Higher advance ratios typically converge fairly quickly. This is caused by the increase in average pressure jump required for lower advance ratio cases. The average pressure jump is approximately equal to C_t/μ^2 , therefore as the advance ratio decreases, the average pressure jump increases, and the rotor boundary condition has a greater impact on the flow field and in most cases slows the convergence. Table 2 shows the number of iterations required for a typical isolated rotor calculation. The speed of the code on a Cray C-90 was approximately 160×10^{-6} seconds/iteration/grid point.

Table 2: Convergence for isolated rotor case

Advance Ratio, μ	Iterations	Cray-90 hours	Memory (Mwds)
0.15	180	3.25	16

Acknowledgments

This research was supported, in part, by the National Aeronautics and Space Administration under Contract No. NAS1-19672.

References

1. Benek, J.A., Steger, J.L., Dougherty, F.C., Buning, P.G., "Chimera: A Grid-Embedding Technique," Arnold Engineering Development Center TR 85-64, April 1986.
2. Rogers, S.E., Kwak, D., Kiris, C., "Steady and Unsteady Solutions of the Incompressible Navier-Stokes Equations," *AIAA Journal*, Volume 29, Number 4, April 1991, Pages 603-610.
3. Van Gaasbeek, J. R., "Rotorcraft Flight Simulation Computer Program C81 with Datamap Interface, Volume 1- User's Manual," USAAVRADCOM-TR-80-D-38A, October 1981.
4. Suhs, N.E., Tramel, R.W., "Pegsus 4.0 User's Manual," Arnold Engineering Development Center TR-91-8, November 1991.
5. Elliott, J.W., Althoff, S.L., Sailey, R.H., "Inflow Measurement Made With a Laser Velocimeter on a Helicopter Model in Forward Flight, Volume 1, Rectangular Planform Blades at an Advance Ratio of 0.15," NASA TM 100541, AVSCOM TM 88-B-004, April 1988.
6. Elliott, J.W., Althoff, S.L., Sailey, R.H., "Inflow Measurement Made With a Laser Velocimeter on a Helicopter Model in Forward Flight, Volume 2, Rectangular Planform Blades at an Advance Ratio of 0.23," NASA TM 100542, AVSCOM TM 88-B-005, April 1988.
7. Elliott, J.W., Althoff, S.L., Sailey, R.H., "Inflow Measurement Made With a Laser Velocimeter on a Helicopter Model in Forward Flight, Volume 3, Rectangular Planform Blades at an Advance Ratio of 0.30," NASA TM 100543, AVSCOM TM 88-B-006, April 1988.

8. Hoad, D.R., Althoff, S.L., Elliott, J.W., Sailey, R.H., "Inflow Measurement Made With a Laser Velocimeter on a Helicopter Model in Forward Flight, Volume 7, Rectangular Planform Blades at an Advance Ratio of 0.35," NASA TM 101598, AVSCOM TM 89-B-001, April 1989.
9. Hoad, D.R., Althoff, S.L., Elliott, J.W., Sailey, R.H., "Inflow Measurement Made With a Laser Velocimeter on a Helicopter Model in Forward Flight, Volume 7, Rectangular Planform Blades at an Advance Ratio of 0.40," NASA TM 101599, AVSCOM TM 89-B-002, April 1989.
10. Chaffin, M. S., and Berry, J. D., "Navier-Stokes Simulation of a Rotor Using a Distributed Pressure Disk Method," 51th Annual Forum Proceedings, Volume 1, American Helicopter Soc., May 1995, pp. 112-136.
11. Ghee, T. A., and Elliott, J. W., "A Study of the Rotor Wake of a Small-Scale Rotor Model in Forward Flight Using Laser Light Sheet Flow Visualization with Comparisons to Analytical Methods," 48th Annual Forum Proceedings, American Helicopter Soc., June 1992.
12. Amtec Engineering, Inc., "Tecplot Version 6 User's Manual", Copyright 1988-1993, P.O. Box 3633, Bellevue, WA 98008-3633.

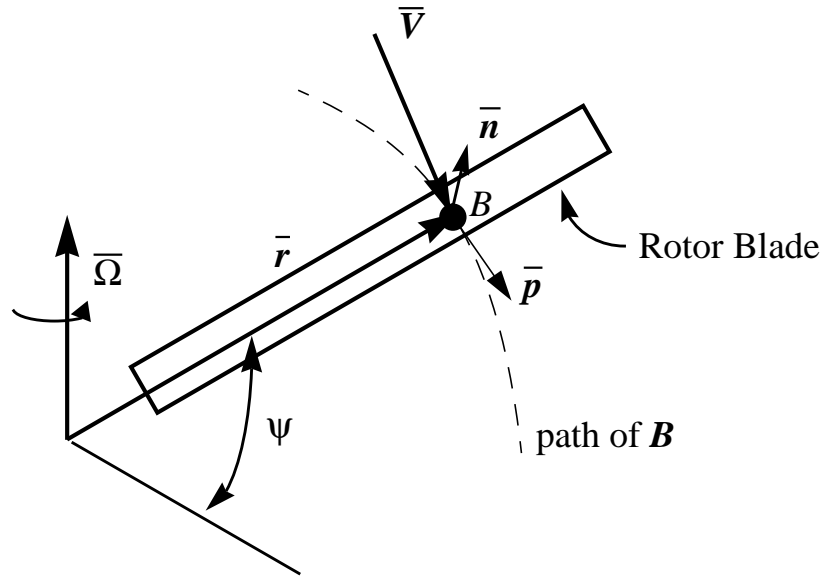


Figure 1. Rotor blade.

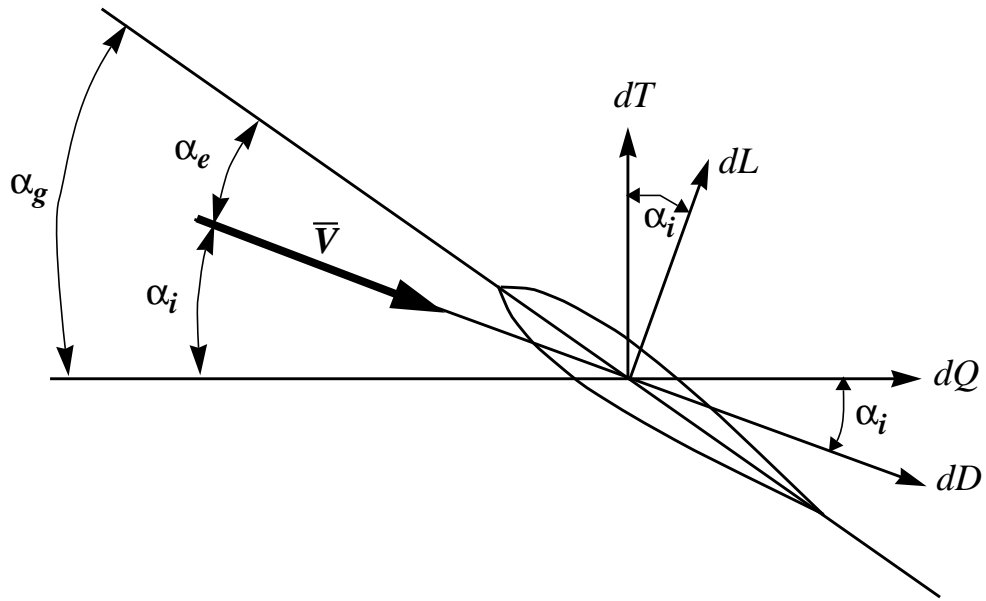


Figure 2. Rotor section with relative velocities and forces.

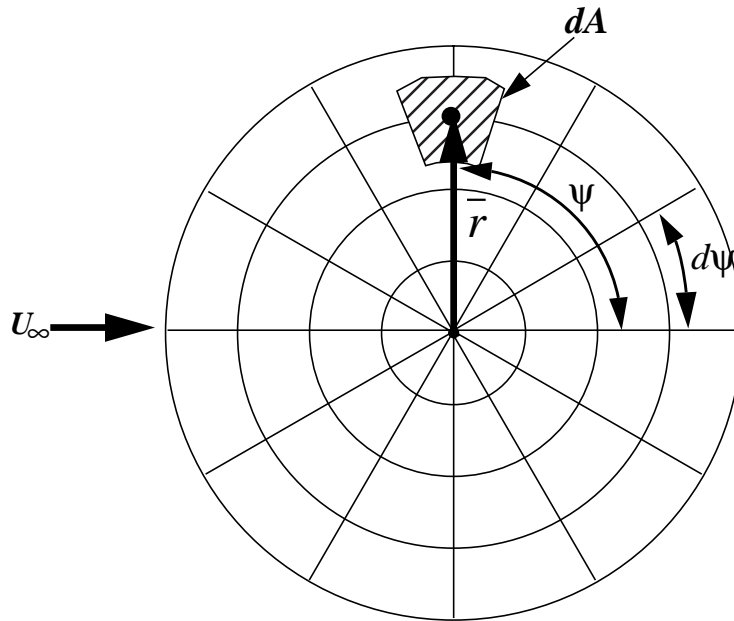


Figure 3. Top view of disk grid at disk surface.

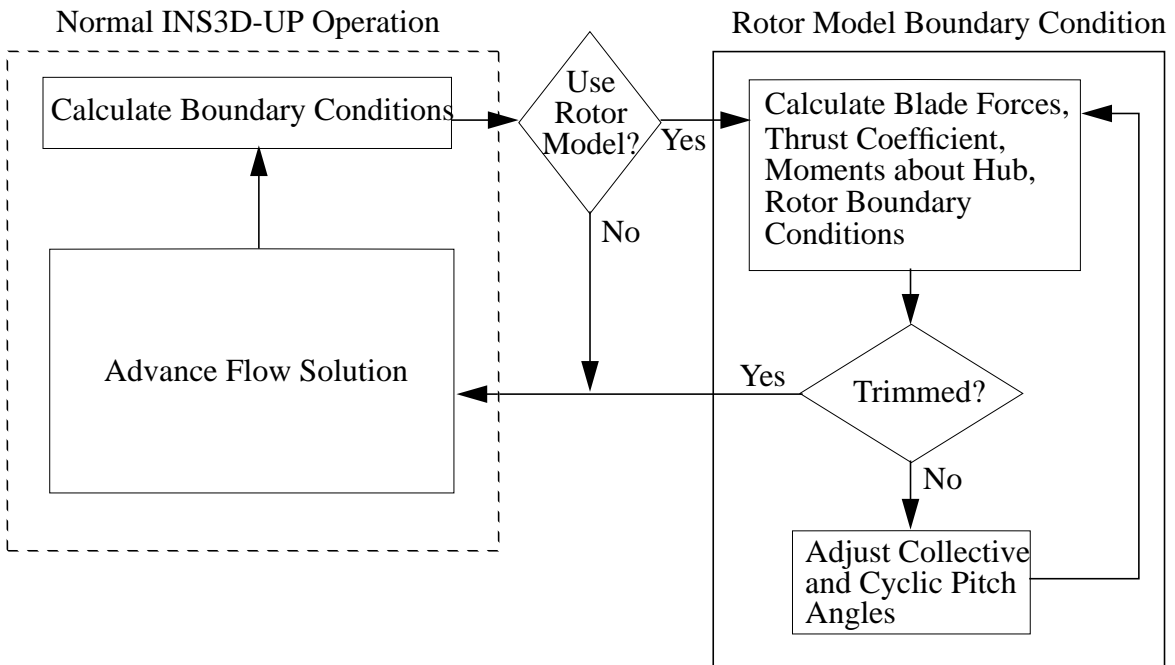


Figure 4. Implementation of rotor model and trim loop.

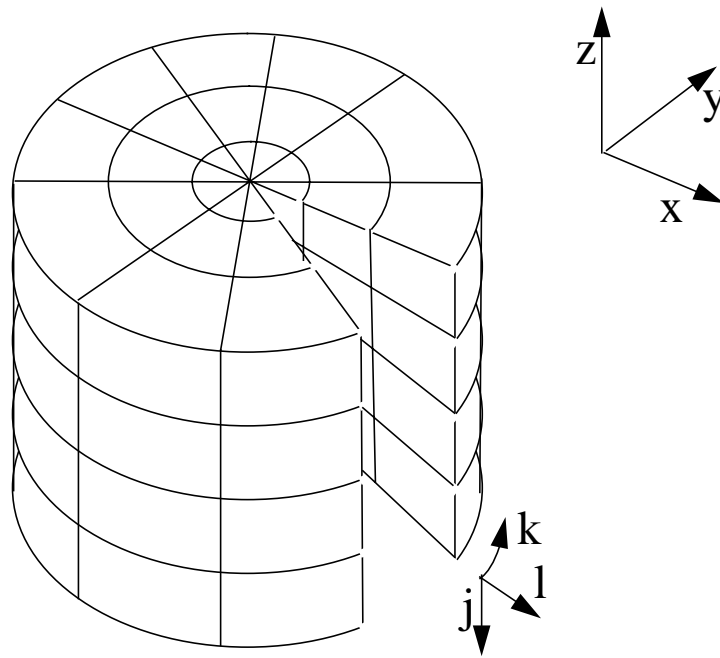


Figure 5. Cylindrical rotor grid topology.

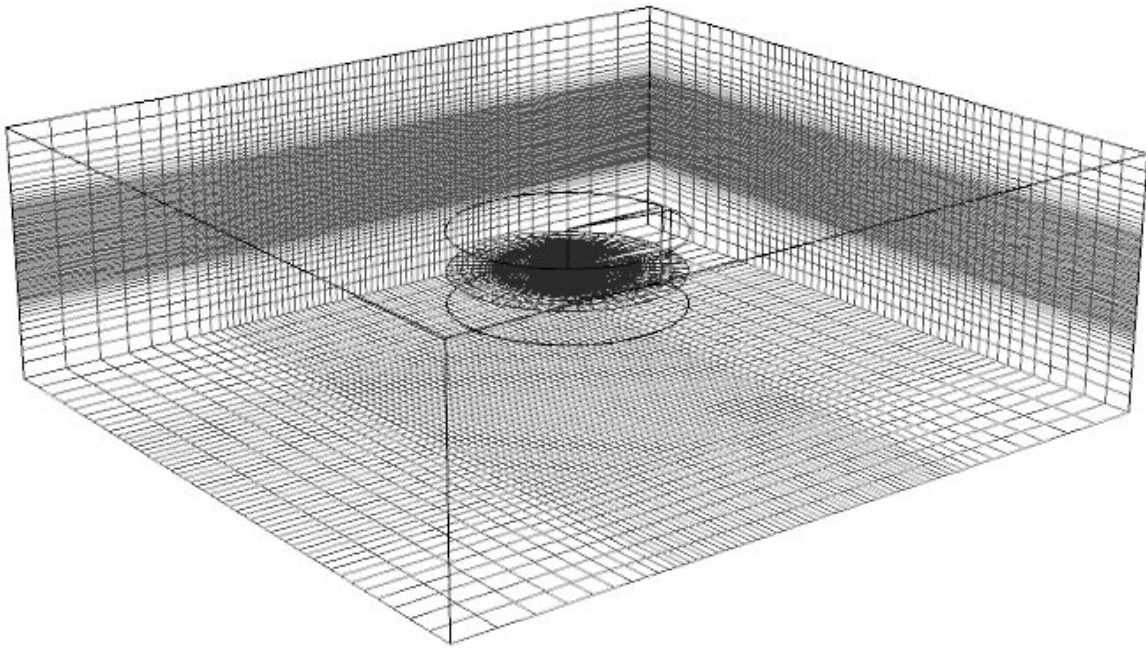


Figure 6. Isolated rotor embedded grid.

Appendix A

Input for INS3D Rotor Model

The following variables are used to describe the geometry of the rotor as well as the conditions for which the rotor is being run. The input is in namelist **rtrstf** in the file *rotor.inp*. Default values are shown in [].

rmu advance ratio [0.4]
aao collective (A_0) (degrees) [0.1]
aaa1 A_1 cyclic (degrees) [0.0]
ab1 B_1 cyclic (degrees) [0.0]
 aa0, **aaa1**, and **ab1** can not be changed at restarts
 note convention: $\theta = \mathbf{aao} - \mathbf{aaa1} * \cos(\psi) - \mathbf{ab1} * \sin(\psi)$
fa1 first harmonic cosine flapping (degrees) [0.0]
fb1 first harmonic sine flapping (degrees) [0.0]
fa2 second harmonic cosine flapping (degrees) [0.0]
fb2 second harmonic sine flapping (degrees) [0.0]
 note convention: flapping = coning + $\mathbf{fa1} * \cos(\psi) + \mathbf{fb1} * \sin(\psi) + \dots$
btwst amount of linear twist in blade (degrees) [-8.0]
aspr aspect ratio of blade, (blade radius/blade chord) [13.0]
bldnum number of blades [4.0]
cdmax upper limit allowed for c_d (used when **model** = 1) [0.3]
clmax upper limit allowed for c_l (used when **model** = 1) [1.2]
clmin lower limit allowed for c_l (used when **model** = 1) [-1.2]
iramp ramp up pressure jump for first **iramp** iterations [10]
iclcfrc interval to calculate and print rotor forces [5]
itrnst iteration to start trim calculations [10]
inttrm interval for trim calculations [5]
altol tolerance for longitudinal trim [0.0001]
alinc maximum change in A_1 during trimming (degrees) [0.1]
bltol tolerance for lateral trim [0.0001]
blinc maximum change in B_1 during trimming (degrees) [0.1]
cttol tolerance for thrust coefficient trim [0.000001]
ctinc maximum change in A_0 during trimming (degrees) [0.1]
itloop maximum number of trimming iterations [10]

`cthrust` desired thrust coefficient [0.0064]

`jinflw` j plane at which the mean induced inflow ratio is to be calculated [15]

`model` model used for sectional loads, 1 = $2\pi\alpha$, 2 = airfoil tables [1]

Notes:

1. Typically, **iramp** is set to 10 to give the flow a chance to develop before trimming starts. Use an initial guess of 0.1 for **aa0** and 0.0 for **aaa1** and **ab1**. Doing this and using the default limits on the trim loops allows the solution to gradually develop and usually results in quicker convergence than making initial guesses for **aa0**, **aaa1**, and **ab1**.
2. Limits on `cl` are to approximate stall and give more reasonable results.
3. Make sure **itrmst** is greater than **iramp**.
4. If trim loops have trouble converging, decrease **alinc**, **blinc**, and **ctinc** and/or increase **inttrm**.

Appendix B

An input file to INS3D for the isolated rotor case is show below. Note that `iover` is set to “1” indicating that an overset grid is being used. Also notice that the namelist `zonin` is repeated for each grid. The angle of attack is set in namelist `geomn`. It is not possible to change the angle of attack during a restart.

```
$datain
  ntmax=500,
  iover=1,
  iss=1,
  istart=0,
  iturb=2,
  ivis=3,
  beta=100.0,
  dtau=1.e24,
  dt=1.e24,
  reynum=2.035e6,
$end
$zonein
  kpr= f,
  iaxsym=111,
  njswp=0,
  nkswp=0,
  nlswp=5,
  cdiss=1.0,
$end
$zonein
  kpr= t,
  iaxsym=111,
  njswp=5,
  nkswp=0,
  nlswp=5,
  cdiss=1.0,
$end
$geomn
  alpha=-3.00,
$end
```

Appendix C

Below is the input for a typical run of the rotor model boundary conditions. This information is contained in the file *rotor.inp*.

```
$rtrstf
  rmu = 0.15,
  aspr = 13.0,
  cthrust = 0.0064
  iramp = 1,
  itrmst = 5,
  aao = 0.1,
  aaa1 = 0.0,
  abl = 0.0,
  altol = 0.0001, alinc = 0.1,
  bltol = 0.0001, blinc = 0.1,
  cttol = 0.00001, ctinc = 0.1,
  inttrm = 5,
  model = 2,
  jinflw = 12,
$end
```

Appendix D

Below is output from a typical run of INS3D-UP using the rotor model boundary conditions. The parts of the output that are due to the use of the rotor model are shown in bold font. Comments are shown in {}.

```
*****
      INS3D-UP Version 1.80-19
      06/06/95 13:45:45   on sabre
*****

-----
                        Grid Information

Opening file named: xyz.dat
This is an unformatted, multiple zone grid file with      an iblank array.
nz =      1 jmax =      65 kmax =      65 lmax =      50
nz =      2 jmax =      58 kmax =      50 lmax =      68
total number of grid points = 408450
largest single dimension   =      68
-----

Rotor input
&RTRSTF  RMU = 0.15,  AAO = 0.1,  AAA1 = 0.,  AB1 = 0.,  FA1 = 0.,  FB1 = 0.,
FA2 = 0.,  FB2 = 0.,
  BTWST = -8.,  ASPR = 13.,  BLDNUM = 4.,  CDMAX = 0.3,  CLMAX = 1.2,  CLMIN =
-1.2,
  IRAMP = 1,  ICLCFRC = 5,  ITRMST = 5,  INTTRM = 5,  A1TOL = 1.0E-4,  B1TOL =
1.0E-4,
  A1INC = 0.1,  B1INC = 0.1,  CTTOL = 1.0E-5,  CTINC = 0.1,
  ITLOOP = 10,  CTHRUST = 6.4E-3,  JINFLW = 12,  MODEL = 2,  &END

{The above input shows that the minimum ramping is being used, the collective is starting out
small, and the model will start trimming on the fifth iteration.}

&DATAIN  NITER = 1,  NP3D = 100,  NQFILE = 100,  NSUB = 1,  NTMAX = 50,  IGRID
= 0,  IINTRP = 0,  INGRID = 1000,  IOVER = 1,
  IPATCH = 0,  ISS = 1,  ISTART = 0,  ISUBIO = 1,  ITURB = 2,  IVIS = 3,  BETA
= 100.,  DTAU = 1.0E+24,
  DT = 1.0E+24,  EIGEPS = 0.1,  EPSCON = 1.0E-4,  PIN = 1.5,  POUT = 1.,
  REYNUM = 2035000.,  &END
&ZONEIN  JMAX = 3,  KMAX = 3,  LMAX = 3,  KPR = F,  IAXSYM = 111,  IFLXODR =
3,  NJSWP = 0,  NKSWP = 0,  NLSWP = 5,  NTJSWP = 0,
  NTKSWP = 0,  NTLSWP = 2,  CDISS = 1.,  &END
&ZONEIN  JMAX = 3,  KMAX = 3,  LMAX = 3,  KPR = T,  IAXSYM = 111,  IFLXODR =
3,  NJSWP = 5,  NKSWP = 0,  NLSWP = 5,  NTJSWP = 0,
  NTKSWP = 0,  NTLSWP = 2,  CDISS = 1.,  &END
Airfoil table read from file naca0012.c81

Allocated      14.296 million words of memory.

&GEOMIN  ALPHA = -3.,  REFAREA = 1.,  &END
```

Boundary Conditions:

```
Number of BC regions read in from bcmain.dat =      5
  Outflow boundary      ibcval = 31 nz=      1 j=      65      65 k=      1      65 l=
1      50
  axial singularity      ibcval = 71 nz=      2 j=      1      58 k=      1      50 l=
1      1
Actuator disk inflow ibcval = 101 nz=      2 j=      28      28 k=      1      50 l=
```

```

9 50
  Hole region          ibcval = -1 nz= 2 j= 29 29 k= 1 50 l=
9 50
  Actuator disk outflo ibcval = 102 nz= 2 j= 30 30 k= 1 50 l=
9 50

```

{NOTE: the rotor model boundary conditions **must** be in the above order and **must** be the last boundary conditions specified.}

Total of 0 no-slip walls processed.

```

Allocated      111680 words for bcozne arrays.
Read in overlap zonal info, nz, ibpnts:      1 9792
Read in overlap zonal info, nz, ibpnts:      2 1376
-----

```

```

Test of overlaid zonal interface reports a max error of 0.3464E-13 for base
zone 1
Test of overlaid zonal interface reports a max error of 0.6566E-11 for base
zone 2

```

{Make sure the above numbers shown in italics are small. They represent the errors caused by the interpolation of grid information. This is easy to overlook, but very important.}

```

Zone: 1 Max Jacobian: 0.14972E+04 Min Jacobian: 0.12762E+02
Zone: 2 Max Jacobian: 0.36905E+08 Min Jacobian: 0.17840E+03

```

Allocated 1.634 Mwords for turbulence model.

Baldwin-Barth Tubulence Model
Version 1.27

```

karman constant = 0.41
cmu              = 9.E-2
cle              = 1.2
c2e              = 2.
sigma e          = 0.700416666667
aplus1           = 26.
aplus2           = 10.
dt model         = 1.E+24
re_t inf         = 1.
Transition locations:
  nz,jtranlo,jtranup = 1, 2*0
  nz,jtranlo,jtranup = 2, 2*0
Line-relaxation sweeps:
  nz,ntjssp,ntksp,ntlsp = 1, 2*0, 2
  nz,ntjssp,ntksp,ntlsp = 2, 2*0, 2
Walls:

```

Distance function was read from file distfun.dat.

```

resmax0 for zone 1 : 6.76634930277088E-22
resmax0 for zone 2 : 9.876532829657071E-9

```

```

nt   resmax   divmax   j   k   l   nz   tures   cd   cl
cdp  clp
---  ---
---  ---

1   0.988E-08  0.988E-10  31  27  50  2   0.883E-17  0.000E+00
0.000E+00  0.000E+00  0.000E+00
2   0.353E+04  0.353E+02  30  46  51  2   0.916E-04  0.000E+00
0.000E+00  0.000E+00  0.000E+00
3   0.322E+04  0.322E+02  29  29  51  2   0.101E-03  0.000E+00
0.000E+00  0.000E+00  0.000E+00
4   0.240E+04  0.240E+02  32  48  46  2   0.931E-04  0.000E+00
0.000E+00  0.000E+00  0.000E+00
-7.562800129194934E-4, -1.242615678921699E-2, 4.478379790770992E-2
right left front back
-0.272E-01 -0.396E-01 -0.176E-01 -0.624E-01
starting trim loop
starting trim loop
10 trim iterations
A0 =1.099999999999987 A1 =-0.3861963347996582 B1 =0.2735704129524734
5   0.227E+04  0.227E+02  31  27  50  2   0.111E-03  0.000E+00
0.000E+00  0.000E+00  0.000E+00

```

{ After the 5th iteration, the calculated C_T is -7.56×10^{-4} , the trim loops are started and after 10 passes through the trim loops, the collective has been increased to try to correct for the low C_T . Note that the cyclic angles have also changed. Had **CTINC** not been set to a small number, A_0 might have become too large during the early trimming attempts and the solution might have become unstable. }

```

6   0.265E+04  0.265E+02  31  29  50  2   0.147E-03  0.000E+00
0.000E+00  0.000E+00  0.000E+00
7   0.288E+04  0.288E+02  31  29  50  2   0.194E-03  0.000E+00
0.000E+00  0.000E+00  0.000E+00
8   0.308E+04  0.308E+02  29  48  51  2   0.216E-03  0.000E+00
0.000E+00  0.000E+00  0.000E+00
9   0.250E+04  0.250E+02  29  46  51  2   0.209E-03  0.000E+00
0.000E+00  0.000E+00  0.000E+00
5.230469709086374E-4, -5.173697433143054E-3, 4.058687632199098E-2
right left front back
0.144E-01 0.922E-02 0.239E-01 -0.167E-01
10 trim iterations
A0 =1.989999999999938 A1 =-0.7501364027702522 B1 =0.2835252502170498
10  0.185E+04  0.185E+02  29  44  51  2   0.230E-03  0.000E+00
0.000E+00  0.000E+00  0.000E+00
11  0.155E+04  0.155E+02  31  28  50  2   0.249E-03  0.000E+00
0.000E+00  0.000E+00  0.000E+00
12  0.187E+04  0.187E+02  29  29  51  2   0.444E-03  0.000E+00
0.000E+00  0.000E+00  0.000E+00
13  0.218E+04  0.218E+02  29  29  51  2   0.553E-03  0.000E+00
0.000E+00  0.000E+00  0.000E+00
14  0.191E+04  0.191E+02  29  31  51  2   0.715E-03  0.000E+00
0.000E+00  0.000E+00  0.000E+00
1.508858882341035E-3, -1.588902451108298E-2, 3.13936377128674E-2
right left front back
0.506E-01 0.347E-01 0.523E-01 0.209E-01
10 trim iterations
A0 =2.879999999999882 A1 =-1.027095750198953 B1 =0.3014633315327107
15  0.207E+04  0.207E+02  31  29  50  2   0.695E-03  0.000E+00
0.000E+00  0.000E+00  0.000E+00
16  0.196E+04  0.196E+02  31  29  50  2   0.647E-03  0.000E+00
0.000E+00  0.000E+00  0.000E+00
17  0.234E+04  0.234E+02  29  33  51  2   0.122E-02  0.000E+00
0.000E+00  0.000E+00  0.000E+00

```

```

18 0.267E+04 0.267E+02 29 35 51 2 0.763E-03 0.000E+00
0.000E+00 0.000E+00 0.000E+00
19 0.250E+04 0.250E+02 29 35 51 2 0.656E-03 0.000E+00
0.000E+00 0.000E+00 0.000E+00
2.595043126406765E-3, -2.437389306736159E-2, 1.943479592517771E-2
right left front back
0.864E-01 0.620E-01 0.864E-01 0.669E-01
10 trim iterations
A0 =3.76999999999984 A1 =-1.199683126443517 B1 =0.6282236783746846
20 0.232E+04 0.232E+02 29 35 51 2 0.510E-03 0.000E+00
0.000E+00 0.000E+00 0.000E+00
21 0.215E+04 0.215E+02 31 33 50 2 0.367E-03 0.000E+00
0.000E+00 0.000E+00 0.000E+00
22 0.336E+04 0.336E+02 29 37 51 2 0.459E-03 0.000E+00
0.000E+00 0.000E+00 0.000E+00
23 0.332E+04 0.332E+02 29 37 51 2 0.345E-03 0.000E+00
0.000E+00 0.000E+00 0.000E+00
24 0.279E+04 0.279E+02 31 35 50 2 0.393E-03 0.000E+00
0.000E+00 0.000E+00 0.000E+00
3.658183123326649E-3, -1.495913391743819E-2, 1.626192560556116E-2
right left front back
0.113E+00 0.979E-01 0.125E+00 0.109E+00
10 trim iterations
A0 =4.659999999999798 A1 =-1.357818367685731 B1 =0.6576795697164393
25 0.282E+04 0.282E+02 31 35 50 2 0.418E-03 0.000E+00
0.000E+00 0.000E+00 0.000E+00

nt resmax divmax j k l nz tures cd cl
cdp resmax divmax j k l nz tures cd cl
clp
---
26 0.274E+04 0.274E+02 31 37 50 2 0.447E-03 0.000E+00
0.000E+00 0.000E+00 0.000E+00
27 0.255E+04 0.255E+02 31 37 50 2 0.651E-03 0.000E+00
0.000E+00 0.000E+00 0.000E+00
28 0.196E+04 0.196E+02 31 37 50 2 0.663E-03 0.000E+00
0.000E+00 0.000E+00 0.000E+00
29 0.171E+04 0.171E+02 26 12 38 2 0.701E-03 0.000E+00
0.000E+00 0.000E+00 0.000E+00
4.817764003763158E-3, -3.107120529424989E-2, 2.069634084709637E-2
right left front back
0.157E+00 0.126E+00 0.170E+00 0.150E+00
10 trim iterations
A0 =5.45100106978208 A1 =-1.560228468702284 B1 =1.128201772374972
30 0.183E+04 0.183E+02 26 12 38 2 0.803E-03 0.000E+00
0.000E+00 0.000E+00 0.000E+00

```

{ After 30 iterations, the trim routines are still requiring the preset limit of 10 iterations. This means that they are not able to trim the rotor. However, note that the C_T is now 0.00482, the moments about the hub are under control, and the residual (`resmax`) is generally decreasing. }

```

31 0.179E+04 0.179E+02 26 12 38 2 0.967E-03 0.000E+00
0.000E+00 0.000E+00 0.000E+00
32 0.201E+04 0.201E+02 29 39 51 2 0.115E-02 0.000E+00
0.000E+00 0.000E+00 0.000E+00
33 0.188E+04 0.188E+02 29 40 51 2 0.132E-02 0.000E+00
0.000E+00 0.000E+00 0.000E+00
34 0.226E+04 0.226E+02 29 37 51 2 0.129E-02 0.000E+00
0.000E+00 0.000E+00 0.000E+00
5.753871723164522E-3, -1.513962421256565E-2, 2.29005854214348E-2
right left front back
0.178E+00 0.163E+00 0.208E+00 0.185E+00
10 trim iterations
A0 =5.799686119039791 A1 =-1.765531638217212 B1 =1.372259070130418

```

```

35 0.361E+04 0.361E+02 29 37 51 2 0.124E-02 0.000E+00
0.000E+00 0.000E+00 0.000E+00
36 0.451E+04 0.451E+02 29 37 51 2 0.132E-02 0.000E+00
0.000E+00 0.000E+00 0.000E+00
37 0.462E+04 0.462E+02 29 37 51 2 0.150E-02 0.000E+00
0.000E+00 0.000E+00 0.000E+00
38 0.410E+04 0.410E+02 29 38 51 2 0.189E-02 0.000E+00
0.000E+00 0.000E+00 0.000E+00
39 0.417E+04 0.417E+02 29 38 51 2 0.231E-02 0.000E+00
0.000E+00 0.000E+00 0.000E+00
6.153947664546494E-3, -1.176517428103984E-2, 2.120075657348508E-2
right left front back
0.188E+00 0.176E+00 0.223E+00 0.202E+00
10 trim iterations
A0 =5.946114109135635 A1 =-1.965499082583975 B1 =1.526974903672006
40 0.339E+04 0.339E+02 29 38 51 2 0.288E-02 0.000E+00
0.000E+00 0.000E+00 0.000E+00
41 0.372E+04 0.372E+02 29 39 51 2 0.369E-02 0.000E+00
0.000E+00 0.000E+00 0.000E+00
42 0.401E+04 0.401E+02 29 39 51 2 0.515E-02 0.000E+00
0.000E+00 0.000E+00 0.000E+00
43 0.404E+04 0.404E+02 29 39 51 2 0.657E-02 0.000E+00
0.000E+00 0.000E+00 0.000E+00
44 0.396E+04 0.396E+02 29 39 51 2 0.848E-02 0.000E+00
0.000E+00 0.000E+00 0.000E+00
6.360627238996691E-3, -8.764058811065567E-3, 1.653152708428962E-2
right left front back
0.193E+00 0.184E+00 0.228E+00 0.211E+00
8 trim iterations
A0 =5.983084257049086 A1 =-2.120740322853678 B1 =1.61160819233671
45 0.327E+04 0.327E+02 29 39 51 2 0.105E-01 0.000E+00
0.000E+00 0.000E+00 0.000E+00

```

{ After 45 iterations, the trim routines are able to trim the rotor in less than the 10 iteration limit set at the input. The values for C_T and hub moment shown above were calculated before the trim routines were called. }

```

46 0.228E+04 0.228E+02 29 39 51 2 0.128E-01 0.000E+00
0.000E+00 0.000E+00 0.000E+00
47 0.200E+04 0.200E+02 29 40 51 2 0.157E-01 0.000E+00
0.000E+00 0.000E+00 0.000E+00
48 0.262E+04 0.262E+02 29 40 51 2 0.186E-01 0.000E+00
0.000E+00 0.000E+00 0.000E+00
49 0.335E+04 0.335E+02 29 40 51 2 0.214E-01 0.000E+00
0.000E+00 0.000E+00 0.000E+00
6.458302075405431E-3, -6.877644627512502E-3, 1.158253065751679E-2
right left front back
0.195E+00 0.188E+00 0.228E+00 0.217E+00
8 trim iterations
A0 =5.969781957344964 A1 =-2.226282329385924 B1 =1.66012949647596
50 0.394E+04 0.394E+02 29 40 51 2 0.238E-01 0.000E+00
0.000E+00 0.000E+00 0.000E+00

```

```

*****
<<<<Restart file has been written>>>>
*****

```

{ The restart file *fort.20* also contains the information needed to restart the rotor solution. There are no special flags that need to be set in the rotor input to handle a restart. }

```

CPU time: Total= 2.241E+03; sec/iter= 4.439E+01; sec/iter/point= 108.7E-06

```