

XROTOR User Guide
Mark Drela
Harold Youngren

last update Nov 13, 2003

General Description

=====

XROTOR is an interactive program for the design and analysis of ducted and free-tip propellers and windmills. It consists of a collection of menu-driven routines which perform various useful functions such as:

- Design of minimum induced loss rotor (propeller or windmill)
- Prompted input of an arbitrary rotor geometry
- Interactive modification of a rotor geometry
- Twist optimization of an arbitrary rotor for minimum induced loss
- Analysis of a rotor with a wealth of choices of operating parameters
- Incoming slipstream effects (from an upstream propeller, viscous wake...)
- Multi-point parameter display
- Structural analysis and corrections for twist under load
- Acoustic analysis with dB noise footprint predictions
- Interpolation of geometry to radii of interest
- Plotting of geometry, aerodynamic parameters, and performance maps
- ...

Program Execution

=====

It is recommended that the X-window graphics generated by XROTOR be in reverse video for better visibility. This is the default. A black on white (like paper) background can be selected by setting the following shell variable:

```
% setenv XPLOT11_BACKGROUND white
```

XROTOR is simply run from the terminal, with up to two arguments:

```
% xrotor  
or % xrotor save_file  
or % xrotor save_file case_file
```

In any case, the following TOP LEVEL menu and prompt pop up:

```
QUIT   Exit program  
.AERO   Display or change airfoil characteristics  
NAME s Set or change case name  
  
ATMO r Set fluid properties from standard atmosphere  
VSOU r Set/change fluid speed of sound  
DENS r Set/change fluid density  
VISC r Set/change fluid viscosity
```

```

NACE f Specify nacelle geometry file
DUCT r Duct/free-tip option toggle
VRAT r Change duct velocity ratio

ARBI   Input arbitrary rotor geometry
.DESI  Design rotor geometry
.MODI  Modify rotor geometry
.OPER  Calculate off-design operating points
.BEND  Calculate structural loads and deflections
.NOIS  Calculate and plot acoustic signature
JMAP   Calculate Cp vs J operating map

INTE   Interpolate geometry to specified radii
SAVE f Save rotor to restart file
SAVO f Save rotor to old style XROTOR restart file
LOAD f Read rotor from restart file
WDEF f Write current settings to  xrotor.def  file.

VPUT f Save  slipstream velocity profiles
VGET f Read  slipstream velocity profiles
VCLR   Clear slipstream velocity profiles

HARD   Hardcopy current plot
WIND   Windmill/propeller plotting mode toggle
PLOP   Plotting options
DISP   Display current design point

```

```
XROTOR  c>
```

Specifying the `save_file` argument will automatically cause the `LOAD` command to be executed, which reads a previously-saved rotor. Specifying the `case_file` argument will also cause the case file to be read, which is equivalent to executing `CGET` from the `.OPER` menu.

In general, the commands preceded by a period place the user in another lower-level menu. Simply typing <Return> will cause an exit back up to the higher-level menu:

```
XROTOR  c>  desi
```

```
.DESI   c>  <Return>
```

```
XROTOR  c>
```

This convention is used throughout XROTOR.

All commands which are not preceded by a period are executed immediately, and the user is prompted for another command. The lowercase letters `i,r,f,s` following some commands indicate the type of argument(s) expected by the command:

```

i  integer
r  real
f  filename
s  character string

```

For example, to set Standard US Atmosphere fluid properties corresponding to 10 km flight altitude, one can issue the command

```
XROTOR  c>  atmo 10
```

Omitting the argument always results in a prompt. For example:

```
XROTOR  c>  atmo
```

```
flight altitude (km)  r>  10
```

Specifying an altitude of -1 sets the properties for water (special case).

Rotor Design

=====

The DESI command, which transfers to the design menu, produces the command prompt

```
.DESI  c>
```

Typing a " ? " will print out the DESI menu:

```
INPU      Input  design parameters... design rotor
.EDIT     Change design parameters... design rotor
N         Change number of radial points

VRTX     Toggle between Graded Mom. and Vortex Form.
FORM      Toggle between Graded Mom. and Potential Form.
WAKE      Toggle between rigid and self-deforming wake
ATMO r    Set fluid properties from standard atmosphere

DISP      Display current design point
TERS      Toggle between terse and verbose output
NAME s    Set or change case name

PLOT i    Plot various rotor parameters
ANNO      Annotate current plot
HARD      Hardcopy current plot
SIZE r    Change plot-object size
```

The design procedure allows calculation of a rotor chord and blade angle (c/R, beta) distributions to achieve a Minimum Induced Loss (MIL) circulation distribution. This can be either the

i) Betz-Prandtl distribution (Graded-Momentum Formulation),
or ii) Goldstein distribution (Potential Formulation),
depending on the state of the FORM toggle.

Design process

The design of a new rotor is typically begun with the INPU command, which prompts the user for all required design-parameter inputs, and then follows by displaying the input-modification menu.

B	2	number of blades
RT	1.5000	tip radius
RH	0.1000	hub radius
RW	0.0500	hub wake displacement body radius
V	10.0000	airspeed
A	---	advance ratio
R	200.0000	rpm
T	---	thrust
P	800.0000	power
CC	0.5000	lift coefficient (constant)
CL		linear lift coefficient (root,tip)
CX		lift coefficient (cursor input)
CR		read CL(r/R) from file
CW		write CL(r/R) to file

..EDIT Parameter, Value (or <Return>) c>

This echoes the user's inputs and allows any of them to be changed by specifying the parameter keywords and new values. For example

```
..EDIT Parameter, Value (or <Return>) c> B 3
..EDIT Parameter, Value (or <Return>) c> V 8
..EDIT Parameter, Value (or <Return>) c> <Return>
```

will change the blade number to 3, the airspeed to 8.0, and then redesign the rotor. This input-modification menu can be invoked repeatedly with the EDIT command. The intent here is to allow rapid modification of design parameters with immediate feedback on the resulting design.

The design parameters contain **two redundant pairs:**

```
advance ratio
rpm

thrust
power
```

Only one parameter in each pair can be prescribed. The remaining parameter is then a result of the design calculation.

All the **design parameters will retain their values for the length of the XROTOR session.** Hence the designed rotor can be analyzed in the other menus, and can then be further redesigned in DESI just by invoking EDIT again.

Rotor Modification

=====

The MODI command, which transfers to the modification menu, produces the command prompt

```
.MODI c>
```

Typing a " ? " will print out the MODI menu:

```

BLAD i  Set new number of blades
MODC    Modify chord distribution
MODB    Modify blade twist angle distribution
SCAL rr Scale current chords
TLIN r  Add linear blade twist (proportional to r/R)
RTIP r  Change tip radius (preserve chord/R)
RHUB r  Change hub radius
RWAK r  Change hub wake displacement body radius
RAKE r  Change blade rake angle
XPAX r  Change pitch-axis x/c

CLIP r  Clip current prop to new radius (preserve chord)

OPTI    Optimize blade twist angles for current planform

PLOT i  Plot various rotor parameters
ANNO    Annotate current plot
HARD    Hardcopy current plot
SIZE r  Change plot-object size

```

The **MODC** and **MODB** commands allow modification of the existing c/R and beta distributions via mouse cursor input. These can also be changed in more simple ways with the **SCAL** and **TLIN** commands.

The **RTIP**, **RHUB**, **RWAK** commands are used to change their corresponding radii. The c/r and beta distributions are preserved as closely as possible. The **XPAX** command changes the chordwise x/c location of the pitch axis. Currently this only has an effect on the plotted blade geometry.

The **OPTI** command is a hybrid between the Design and Modification operations. It will re-twist the rotor (the beta distribution) to achieve a MIL circulation while holding the current chord distribution fixed --- i.e. it "optimizes" the twist. This command must be used cautiously, since the MIL circulation will not be achievable if the blade stalls somewhere. Also, the "optimum" MIL distribution is not necessarily the best in an overall sense, since the rotor may be made worse at other operating points. This is the same situation as when a rectangular wing is "optimized" by twisting it to achieve an elliptical loading at one flight condition -- it becomes worse at most other lift coefficients. When the **OPTI** command is issued, it will list the changes to the current blade angle distribution, followed by a prompt for an underrelaxation factor.

```

Blade angle changes
i      r/R      d(beta)
1  0.106  -1.232
2  0.129  -1.359
3  0.165  -1.543
.
.
28 0.991  -3.163
29 0.997  -3.168
30 0.999  -3.171

```

Enter relaxation factor for blade angle changes: 1.000

The actual blade angle changes applied to the blade will be scaled by the specified relaxation factor. Usually, a value smaller than the default of 1.000 works best, since off-design operation is then less likely to be adversely affected.

Arbitrary-rotor input

=====

An arbitrary rotor geometry is input via the ARBI command at TOP LEVEL. This prompts the user for a number of operating and geometric parameters, and then prompts for the radial distribution of r/R , chord/ R , blade angle. The chord and angle distributions are splined radially and interpolated to the internal computational stations. All other XROTOR operations can then proceed as usual.

Rotor analysis

=====

The OPER command, which transfers to the operating menu, produces the command prompt

.OPER c>

Typing a " ? " will result in the rather extensive OPER menu:

ADVA r	Prescribe advance ratio
RPM r	Prescribe rpm
THRU r	Prescribe thrust
TORQ r	Prescribe torque
POWE r	Prescribe power
ASEQ rrr	Calculate case sequence of advance ratios
RSEQ rrr	Calculate case sequence of rpms
BSEQ rrr	Calculate case sequence of blade angles
VSEQ rrr	Calculate case sequence of speeds at fixed pitch
CLRC	Clear case accumulator
ADDC	Add current point point to case accumulator
CPUT f	Write current case accumulator to file
CGET f	Read cases from file
CASE i	Select case
ATMO r	Set fluid properties from standard atmosphere
VELO r	Set or change flight speed
ANGL r	Change collective blade angle
PVAR f	Enter and use engine rpm/power line
FORM	Toggle between Graded Mom. and Potential Form.
VRTX	Toggle between Graded Mom. and Vortex Form.
WAKE	Toggle between rigid and self-deforming wake

NAME s	Set or change case name
WRIT f	Write current operating point to disk file
DISP	Display current operating state
INIT	Initialize next analysis case
REIN	Re-initialize prop to known operating state
TERS	Toggle between terse and verbose output
PLOT i	Plot various rotor parameters
ANNO	Annotate plot
HARD	Hardcopy current plot
SIZE r	Change plot-object size

Hopefully, most of these commands are self-explanatory. An operating state can be specified by any one of the ADVA, RPM, THRU, TORQ, or POWE commands. Care must be taken when using the latter three commands, since the specified thrust, torque, or power may not be achievable by the current rotor, especially in the windmill mode.

A number of operating-point cases can be stored in the "case accumulator", which can be filled up on-by-one with the ADDC command, or with the **ASEQ, RSEQ, VSEQ, BSEQ** commands. These automatically sweep through one of the four corresponding parameters, appending to the case accumulator in the process. A case can then be quickly recomputed by specifying its index via the CASE command.

A few OPER and TOP LEVEL commands must be explained in more detail ...

... FORM command (mentioned earlier)

This allows the user to switch between two different methods of calculating induced velocities and induced losses. Both methods treat the rotor blades as lifting lines, and both assume that the disk loading is relatively low and hence the wake contraction and the wake self-deformation are small.

Graded Momentum Formulation.

This is the classical theory of propellers revived recently by E.E. Larrabee. **It relies on the Betz-Prandtl tip loss fudge factor which assumes that the rotor has a low advance ratio and/or many blades. As a result, this formulation is not suitable for advance ratios greater than about 0.5 . Its chief advantage is extreme computational economy.**

Potential Formulation.

This is a more modern approach which solves for the helically-symmetric potential flow about a rigid helicoidal wake and hence is valid for all blade numbers and advance ratios. It is an extension of Goldstein's 2 and 4 blade solution to all blade numbers and arbitrary radial load distributions. On the negative side, this approach is computationally more expensive than Graded Momentum, but still requires a fraction of a second per operating point on a RISC workstation. This method also treats finite radius rotor hubs and nacelle installations, which the Betz-Prandtl fudge factor cannot deal with properly. The method also readily accepts a tip casing boundary condition, which is enabled/disabled with the DUCT command at TOP LEVEL.

Vortex Formulation.

This uses a discrete vortex wake for induced velocity calculations. The velocity on the propeller lifting line is calculated from discrete line vortices on the rigid helicoidal wake surface that trail from the lifting line into the far-field downstream. This option was added for use with propellers with non-radial lifting lines (raked or swept blades). The discrete vortex approach is quite a bit slower than the Potential Formulation but can treat more general propeller geometries. Note that the wake is not relaxed to be force-free but trails at the wake advance ratio (calculated as part of the XROTOR solution). The Vortex Formulation is selected in OPER or DESI by the VRTX keyword.

... JMAP command

This produces data for a two-dimensional plot of C_p versus J , with contours of equal efficiency and blade angle. This is a compact way of summarizing propeller performance and is used by Hamilton Standard. The routine can run for some time, especially if the Potential Formulation is used. A crude plot with "ragged" contours can be generated fairly quickly, but much more runtime will be needed if a very high-quality plot is to result. The data produced by JMAP is written to an unformatted file specified by the user, and plotted later with the separate JPLOTT program.

Acoustic Calculations

Executing the NOIS command enters the .NOIS menu:

```
P   rrr Calculate acoustic p(t) at observer x,y,z
FOOT rr Calculate dB ground noise footprint
NTIM i  Change number of time samples
UNIT    Toggle coordinate unit  m,ft

AOC  r   Set constant blade cross-sectional area/c**2
AFIL f   Set blade cross-sectional area/c**2 from file

PLOT i  Plot various acoustic parameters
HARD    Hardcopy current plot
ANNO    Annotate plot
SIZE r   Change plot-object size
```

The P command calculates a pressure signature $p(t)$ at any point in space, over one blade-passing period. The position can be specified in either feet or meters, which is toggled with the UNIT command. The calculation method used is taken from the following two references.

G.Succi,
"Design of Quiet Efficient Propellers", SAE Paper 790584, 1979.
Note: Equation (16) in this reference is mistyped.

G.Succi, D.Munro, J.Zimmer,
"Experimental Verification of Propeller Noise Prediction", AIAA-80-0994.
Note: equations (8) and (9) in this reference are mistyped.

The method uses the retarded-time concept. A sphere collapsing towards the observer at the speed of sound "collects" acoustic signals from the blade thickness and the blade loading where and when it intersects the blades during its inward travel. The sum of all these signals then results in an instantaneous acoustic pressure p seen by the observer when the sphere collapses on him at some later time t . The method requires that the blade speed along the observer's line of sight be subsonic, because the blade must not run past the inward-collapsing sphere from behind. This invalidates the method.

If the relative blade-tip Mach is close to unity, then the resulting pressure signal will be very "spiky". The number of time samples per blade period may need to be increased with the NTIM command to maintain good resolution of the $p(t)$ signal.

One contribution to the acoustic pressure is due to the blade airfoil's cross-sectional area, which is set with the AOC command, and is specified as A/c^2 . For most airfoils, this is roughly related to the thickness/chord ratio:

$$A/c^2 \sim 0.7 t/c.$$

The AOC command sets a radially-constant value of A/c^2 , which should correspond to its value near the tip, since that's where most of the noise generation occurs. For more accuracy, an arbitrary A/c^2 distribution can be read in from a file via the AFIL command. This file has the format

```
r/R  A/c^2
r/R  A/c^2
.    .
.    .
```

where r/R increases outward along the blade.

Immediately after the $p(t)$ signal is generated, it is Fourier-decomposed, and the dB values of the individual harmonic components are displayed. These can also be plotted with PLOT 2. The Fourier decomposition is defined as follows.

$$p(t) = \text{Real} \left[\sum_{k=1}^{\infty} C_k \exp(-ik\omega t) \right] \quad ; \quad \omega = \text{blade-passing radian frequency}$$

$k = \text{number of blades}$
 $\times \text{prop shaft speed}$

The individual dB value for each component is defined relative to 20 microPascals:

$$\text{dB}(k) = 20 \log_{10} \left(\frac{\sqrt{1/2} |C_k|}{20 \times 10^{-6}} \right)$$

The total dB level uses the r.m.s. pressure:

$$\text{total dB} = 20 \log_{10} \left(\frac{p_{\text{rms}}}{20 \times 10^{-6}} \right)$$

where

$$p_{rms} = \frac{1}{2\pi} \sqrt{\frac{1}{2} \sum_{k=1}^{\infty} |C_k|^2}$$

The FOOT command calculates the total dB level at each point of a rectangular grid at a specified distance under the aircraft, thus forming ground noise-level contours.

Higher-Order Corrections

The light-loading assumption states that a trailing vortex sheet remains in the path of the rotor blade which created it. In XROTOR, the inviscid efficiency of the rotor η_i is used to estimate the self-induced motion of the vortex sheets to yield a "wake advance ratio" L_w which differs from the rotor advance ratio (this is the core of Theodorsen's theory).

$$L_w = (V/WR) / \eta_i$$

This "wake advance ratio" is then used to define the vortex sheet geometry which is an important parameter in the calculation of induced velocities. The net result is more realistic behavior at high disk loadings. A similar technique was employed by David Munroe in his propeller noise PhD work at MIT around 1980. In XROTOR, the correction is an option which can be enabled/disabled with the WAKE command from the OPER or the DESI menus, and is employed by both the Graded Momentum and Potential Formulations. Note that this is not a true "free wake" model since the wake will deform itself in pitch, but not in contraction or in roll-up (which for sanity's sake is thoroughly neglected here). Incidentally, the wake self-deformation makes the overall calculation method perfectly consistent with the actuator disk model for arbitrarily large disk loading in the limit of zero advance ratio (and zero blade profile drag, of course). The fixed-wake model does not have this property.

Incoming Slipstream Effects

XROTOR has the capability to approximately model the effects of an upstream or downstream propeller. In a counter-rotating configuration, for example, the downstream prop will effectively see a freestream with additional axial and rotational velocity components. **The upstream propeller will only see an additional axial component, and only if the two props are sufficiently close (within a few radii).** The two propellers need not have the same radius.

To analyze a typical counter-rotating configuration, for example, one might begin by designing or analyzing the front propeller assuming it is unaffected by the rear one. The VPUT command will then write out the circumferential averages of its far-downstream slipstream velocity components to a named file. The format for a slipstream file is

```
R(1)  VAXIAL(1)  VTANG(1)
```

```

R(2)  VAXIAL(2)  VTANG(2)
      .          .          .
      .          .          .
R(N)  VAXIAL(N)  VTANG(N)

```

with R in meters, and V in m/s. VAXIAL is positive for downstream velocities, and VTANG is positive in the prop rotation direction. This file can be subsequently read in with the VGET command before the rear propeller is designed or analyzed. **What velocities the rear propeller actually "sees" depends on its axial location and rotation direction relative to the front propeller, and whether or not the propellers are inside a duct.** The user will be asked to specify the weights to apply to the input axial and tangential velocity components. Appropriate weights are:

	Vaxial	Vtang.
For calculating front prop which is far ahead of rear prop:	0.0	0.0
For calculating front prop which is just ahead of rear prop:	0.5	0.0
For calculating rear prop which is just behind front prop:	0.5	+/-1.0
For calculating rear prop which is far behind front prop:	1.0	+/-1.0

The tangential velocity weight is +1.0 for co-rotating propellers, and -1.0 for counter-rotating propellers. If both propellers are in a duct, then all the Vaxial weights should be 0.5. This incidentally will also account for the additional induction velocity set up by the props inside the duct.

After the aft propeller is designed, _its_ slipstream velocity components can be written out with VPUT to another named file, and this can be read in with VGET for a subsequent re-design of the front propeller, with the appropriate velocity weights chosen from the table above (typically Vaxial = 0.5, Vtang = 0.0)

If desired, the rear propeller can be subsequently redesigned with the slipstream from the new front prop, ad infinitum. Fortunately, if each propeller is designed for a specific thrust or power, this iteration will converge fairly quickly. For preliminary design, only redesigning the front propeller once will typically suffice.

If a rotor is designed and/or analyzed with a non-zero slipstream velocity profiles and then saved with the SAVE command from TOP LEVEL, the slipstream profiles are saved with it. When the rotor is read back in later using LOAD, the original slipstream information is read in as well, and will overwrite any slipstream information already present (appropriate messages are printed). Hence, it is not necessary to load in the original slipstream file before the loaded rotor is analyzed.

It should be pointed out that a slipstream profile file can be generated by any means available to the user. It does not necessarily have to be generated with VPUT. The V distributions are splined to the radial stations of the prop being analyzed.

Nacelle Corrections

=====

If the rotor hub is embedded in a nacelle or fuselage, or even if the rotor is operating in the vicinity of a nacelle, the thrust, power absorbed, and overall efficiency can be substantially different than if

the presence of the nacelle is ignored. Also, the optimum (minimum induced loss) radial circulation distribution can be significantly altered by the presence of the nacelle. The calculation of the thrust is further complicated by the nacelle's perturbation flowfield changing the thrust on the rotor, and the rotor's pressure field causing a thrust force on the nacelle. What really counts is the sum of the rotor and nacelle thrust (or drag). In XROTOR, the inviscid thrust, power, and ideal efficiency are calculated for both the rotor alone and for the rotor/nacelle combination. To summarize...

				Ttotal	
				Twake	
		Tvisc			
				Tinv	
				Tnacelle	
				Ptotal	
				Pwake	
		Pvisc			
				Pinv	

In the actual calculations, The total thrust and power are computed by

$$\begin{aligned} T_{\text{total}} &= T_{\text{visc}} + T_{\text{wake}} \\ P_{\text{total}} &= P_{\text{visc}} + P_{\text{wake}} \end{aligned}$$

with T_{wake} and P_{wake} being calculated on an inviscid "equivalent rotor", which has no nacelle and which has the same far vortex wake as the real rotor. T_{visc} and P_{visc} are calculated by direct integration of the blade profile drag on the real rotor.

The nacelle thrust (or drag) is calculated indirectly by

$$\begin{aligned} T_{\text{nacelle}} &= T_{\text{wake}} - T_{\text{inv}} \\ P_{\text{nacelle}} &= P_{\text{wake}} - P_{\text{inv}} = 0 \quad \text{always} \end{aligned}$$

with T_{inv} and P_{inv} being calculated by direct integration of the blade profile lift forces on the real rotor. It is reassuring that P_{nacelle} always turns out to vanish, since a fixed axisymmetric nacelle cannot possibly deliver power to or from the flow.

The radial circulation distribution of the "equivalent rotor", which is needed to calculate its thrust and power, is determined using mass conservation and Kelvin's circulation theorems. Just behind a rotor, the circumferential circulation around a streamtube of some radius is simply the total bound circulation on the rotor blades at that radius,

and this circulation cannot change downstream (Kelvin's theorem). The streamtube's radius, on the other hand, will change downstream as the average velocity in the streamtube changes and/or the cross-sectional area of any nacelle in the center changes. The contraction and velocity are related by mass continuity requirements. Specifically, if we define

r = real rotor radial coordinate
 $U(r)$ = real rotor freestream + average induced axial velocity
 $u(r)$ = perturbation velocity induced by nacelle at rotor
 r' = equivalent rotor radial coordinate
 $U'(r')$ = equivalent rotor freestream + average induced axial velocity

then the radial location of the streamtube at the real rotor (r) and its location at the equivalent rotor (r') are related by the differential mass continuity equation:

$$(U+u) r dr = U' r' dr'$$

If this relation is satisfied, the real-rotor and equivalent-rotor streamtubes from r and r' will coincide in the far wake and have the same axial velocity since their mass flow will be the same. If they also have the same circulation, their tangential velocity will be the same, and thus the far wakes from the real and equivalent rotors will be the same.

The "correspondence" function $r'(r)$ is obtained by numerically integrating the ODE above. To do this requires that the initial value of $r'(r)$ corresponding to the blade root at $r = r_0$ be specified, say $r'(r_0) = r_0'$. The best value for this equivalent rotor (and far wake) "core radius" r_0' is the radius of the displacement body of the nacelle far downstream. The displacement body cross-sectional area far downstream A_{inf} is just the drag area of the nacelle, and is related to the viscous drag D on the nacelle and the freestream dynamic pressure q by

$$D = q A_{inf} \quad \text{and} \quad A_{inf} = \pi (r_0')^2$$

Once the function $r'(r)$ and thus $r(r')$ is determined, the bound circulation distribution on the equivalent rotor is then simply $G'(r') = G[r(r')]$, where $G[r]$ is the bound circulation on the real rotor at radius r .

A minor simplification is made in the integration of the ODE above. Instead of using the average axial induced velocity, the local induced velocity at the propeller is used instead. This is a reasonable approximation, especially at the inner radii where the streamtubes have the greatest relative radius changes.

The nacelle perturbation velocity $u(r)$ which appears in the ODE can in principle be determined by any potential flow calculation method. In XROTOR, it is determined by a simple model where the nacelle (assumed axisymmetric) is replaced by a distribution of sources and sinks along its centerline. The source strength m is then related to the nacelle cross-sectional area distribution $A(z)$ (more precisely, $A(z)$ should be the cross-sectional area of the viscous displacement body of the nacelle, since this is what the outer potential flow really "sees").

$$dA$$

$$m(z) = U \frac{d}{dz}$$

where z is the axial coordinate increasing downstream. The perturbation velocity is then calculated by integrating over this source distribution:

$$u(r) = \frac{1}{4\pi} \int \frac{m(\bar{z}) [\bar{z} - z]}{[(\bar{z} - z)^2 + r^2]^{3/2}} d\bar{z}$$

where the rotor is located at the axial location $z = \bar{z}$. The nacelle area distribution $A(z)$ can be loaded from a file into XROTOR via the NACE command at TOP LEVEL. The file must have one of two forms, specifying the area $A(z)$ directly, or indirectly via the radius $r(z)$

\bar{z}	Ainf		\bar{z}	-r0'
z1	A1		z1	-r1
z2	A2		z2	-r2
z3	A3	OR	z3	-r3
.	.		.	.
.	.		.	.
.	.		.	.
zn	An		zn	-rn

where the axial coordinates $z1, z2, z3, \dots$ (in meters) must be monotonically increasing downstream. A negative number is assumed to be a body radius, and is then immediately used to determine the corresponding area:

$$A = \pi r^2$$

Ainf is the nacelle profile-drag area D/q , and the corresponding radius $r0'$ is then the far wake displacement radius described earlier.

It must be pointed out that even if the nacelle perturbation velocities are negligible, the difference between the displacement body radii at the rotor and far downstream can still give a non-zero nacelle thrust and also change the optimum circulation distribution. Therefore, XROTOR allows the user to set these radii independent of the nacelle geometry. These radii are input as design parameters for the DESI minimum induced loss design routine. The far wake displacement body radius can also be changed independently at any time with the RWAK command from the .MODI menu.

If a rotor is designed and/or analyzed after loading a nacelle area distribution file and then saved with the SAVE command from TOP LEVEL, the perturbation velocities and far wake displacement body radius are saved with it. When the rotor is read back in later using LOAD, this information is read in as well. It is not necessary to load in the original nacelle area file or to set the displacement body radius before the loaded rotor is analyzed.

If a counter-rotating configuration on a nacelle is analyzed, the nacelle effects are simply superimposed on the incoming-slipstream effects described in the previous section.

Ducted Rotors

=====

XROTOR has the capability to approximately model a rotor embedded within a duct. This option uses the equivalent prop concept described above for nacelle effects to model increased (or decreased) massflow through the prop due to induction from a duct.

The critical parameter for a ducted rotor is the area (or 1/velocity ratio) A_{exit}/A_{prop} . XROTOR assumes that a duct is an aerodynamically "long" duct in that the induced axial velocity at the disk has the normal $0.5 \cdot V_{slipstream}$ induced axial velocity and an additional induced axial velocity due to the duct. This additional induced axial velocity at the disk is a function of the duct area ratio, i.e. if $A_{exit}/A_{prop} = 1.0$ the additional induced axial velocity is $0.5 \cdot V_{slipstream}$.

Note that the apparent freestream is also scaled by the duct area ratio so that $A_{exit}/A_{prop} > 1.0$ will increase the apparent inflow velocity at the propeller disk.

The tangential (swirl) velocity is not affected by the presence of a duct.

The "equivalent" prop for ducted rotors is normally larger than the physical prop. This produces the additional thrust due to the duct effect. In XROTOR this will result in higher overall thrust and a "Nacelle thrust" which in this case is the duct thrust (calculated as the difference between the thrust of the real prop and its "equivalent" prop).

The design condition for a ducted rotor is not the constant induced efficiency condition (MIL condition) used for open rotors, a constant radial circulation condition (so-called free-vortex condition which gives no radial flow) is used for ducted rotors.

Structural Loads and Deflections

=====

The BEND command from TOP LEVEL gives the structural-routine prompt:

```
.BEND  c>
```

Typing a " ? " will result in the BEND menu being displayed:

```
READ f Read in blade structural properties
EVAL  Evaluate structural loads and deflections
CLR   Clear all structural deflections

DEFL  Set new twist = static + structural twist
REST  Set new twist = static twist
SETS  Set static twist = current - structural twist

WRIT f Write structural solution to disk file
```

```

PLOT i Plot structural parameters
ANNO  Annotate plot
HARD  Hardcopy current plot
SIZE r Change plot-object size

```

A 3-axis nonlinear slender beam model is used to model the structural behavior of the blade. Before any structural calculations can be performed, it is necessary to define the blade's structural properties via the READ command, which reads a file of the following format:

```

Name-label
Case-label
Variable-label
R(1) EIXX(1) EIYY(1) EA(1) GJ(1) EK(1) M(1) MXX(1) XOCG(1) XOSC(1) RST(1)
R(2) EIXX(2) EIYY(2) EA(2) GJ(2) EK(2) M(2) MXX(2) XOCG(2) XOSC(2) RST(2)
.
.
.
R(N) EIXX(N) EIYY(N) EA(N) GJ(N) EK(N) M(N) MXX(N) XOCG(N) XOSC(N) RST(N)

```

The first three lines are arbitrary alphanumeric labels which are ignored. The following lines give the structural properties along the radius, (dimensioned in SI units!):

```

R      radius  (must monotonically increase)
EIXX   in-plane stiffness
EIYY   out-of-plane stiffness
EA      extensional stiffness
GJ      torsional stiffness
EK      extensional/torsional stiffness  (torsion per extension-strain)
M       mass density / length
MXX     pitch axis inertia / length
XOCG    x/c of section CG
XOSC    x/c of section shear center (structural axis)
RST     structural radius for strain evaluation

```

These variables are splined to the computational stations.

The EVAL command will then calculate and display the structural solution for the current operating point. The display has the following format.

i	r/R	u/R	w/R	t (deg)	Mz (N-m)	Mx (N-m)	T (N-m)	P (N)
1	0.041	0.000	0.000	0.00	127.79	182.19	-103.77	25684.4
4	0.184	0.000	0.000	-0.22	94.93	108.46	-103.34	24878.7
7	0.335	-0.001	0.001	-0.32	65.59	54.51	-91.82	22061.5
.								
.								

i	r/R	Ex /1e3	Ez /1e3	Ey /1e3	E _{max} /1e3	g /1e3
1	0.041	0.243	-0.367	0.695	1.135	-0.505
4	0.184	0.103	-0.261	0.748	1.028	-0.179


```
7  0.335   0.108  -0.200   0.853   1.080  -0.103
.
.
```

The axis definitions are:

```
X  aft along prop rotation axis
Y  radial along blade
Z  perpendicular to blade:    X x Y = Z
```

u/R and w/R are the deflections in the X and Z directions, and t is the torsional twist (positive in the increasing incidence direction). Mz and Mx are the bending moments about the Z and X axes, while T is the moment about the radial Y axis (i.e. torsion). P is the tensile load.

The strains Ex and Ez are those due to bending in the X and Z directions, while Ey is the strain due to extension in the Y direction. Emax is the maximum strain calculated by $Emax = \sqrt{Ex^2 + Ez^2} + Ey$. g is the shear strain due to twist t. Ex, Ez, and g are evaluated at the local radius input via the RST array described above.

The structural twist t can be added to the static blade geometry angles by issuing the DEFL command (with the lifting line model, only the twist is significant). One can subsequently go to OPER to recalculate the aero solution, and then back to BEND to recalculate the structural solution and add the new twist, etc. Normally, this iteration is not necessary unless the twists are very severe.

The DEFL command can always be reversed with REST command, which resets the blade twists back to the static values. The SETS command subtracts the twists from the static blade angles themselves, and is not reversible as such. It is used to set that static twist which under load will give the current twist.

Output
=====

All output goes directly to the terminal screen. In addition, the same output can be also written to a disk file by the WRIT command. The first time WRIT is issued, the user will be prompted for a file name, which will then be used for the disk output. If that file already exists on disk, the output can be appended to it.

In addition to the primary output file, a save file with all the attributes of the current rotor (including the airfoil characteristics and operating parameters) can be created at any time with the SAVE command. The user is asked for a file name each time. The LOAD command is used to read the save file later.

XROTOR uses the Xplot11 (modified Versaplot) plot package, which drives X-terminals and can generate a PostScript file of the current window plot at any time. The hardcopy is not affected by the reverse-video selection described earlier.

The default PostScript output is black & white, but color PostScript can be selected by setting the IDEVRP variable appropriately in SUBROUTINE INIT (in xrotor.f). Color PostScript can also be selected at runtime from the PLOP menu at TOP LEVEL.

Start-up default file =====

XROTOR tries to read most of its control parameters from the xrotor.def file at startup. If this file is not found, then hardwired defaults set in SUBROUTINE SETDEF are used instead.

Many of the control parameters can be altered during runtime. A new xrotor.def file containing them can be written with the WDEF command, so that they will be automatically restored when XROTOR is started again.

Units and Non-dimensionalization =====

XROTOR uses SI units for all dimensioned input and output. However, all internal calculations are performed with variables normalized with the following quantities:

R tip radius
V freestream speed
rho freestream density

Some of the output uses conventional normalization definitions:

$$T_c = T / (0.5 \rho V^2 \pi R^2)$$

$$P_c = P / (0.5 \rho V^3 \pi R^2)$$

$$C_t = T / (\rho D^4 n^2) \quad ; \quad D = 2R$$

$$C_p = T / (\rho D^5 n^3) \quad ; \quad n = \text{revs. per second}$$

$$J = V / (n D) = \pi V / (W R) \quad ; \quad W = 2 \pi n \text{ rad/s}$$

$$\eta = T_c / P_c = J C_t / C_p \quad (\text{total efficiency})$$

The circulation "Gamma" is displayed on some of the plots in the Betz-Prandtl canonical form:

$$G = B (\text{Gamma}/VR) / (2 \pi Lw)$$

where Lw is the wake advance ratio, related to the geometric advance ratio V/wR and the inviscid efficiency η_i by

$$\eta_i Lw = V/(wR)$$

In the actuator disk limit η_i is the ideal (Froude) efficiency, and G approaches unity. Hence, G shows how close the real prop disk loading is to this ideal limit.

Since all internal calculations are done in non-dimensional variables, if all input is in some other consistent set of units, all output will be in the same set of units. Only the SI unit labels on the formatted output will be wrong. A modification to SUBROUTINE ATMO will also be necessary to convert its SI output to whatever units are desired.

Blade Section Properties

=====

Blade section aero properties are set in the AERO menu. This allows the user to specify one or more sets of aerodynamic data for the blade on "aero sections". Each aero section corresponds to a radial station on the blade. If only one aero section is present all radial stations on the blade will have the same characteristics, if two or more sections are input linear interpolation is used to define aero properties for radial stations in between the aero sections.

.AERO c>

Typing a " ? " will result in the AERO menu being displayed:

```
DISP  Display airfoil characteristics
NEW   Create a new aero section
DEL   Delete an aero section
EDIT  Edit section aero data
READ  Read airfoil characteristics from disk file
WRIT  Write airfoil characteristics to disk file
PLOT  Plot airfoil characteristics
ANNO  Annotate current plot
HARD  Hardcopy current plot
```

Defined aerodynamic sections:

N	r/R	CLmax	CLmin	CDmin	Mcrit	REexp	REref
1	0.0000	2.0000	-1.5000	0.00700	0.6200	-0.2000	0.2000E+07
2	0.6000	1.9000	-1.4000	0.00550	0.7200	-0.2000	0.8000E+07
3	0.8000	1.8000	-1.2000	0.00550	0.7600	-0.2000	0.6000E+07
4	1.0000	1.8000	-1.2000	0.00550	0.7700	-0.2000	0.5000E+07

In this example the rotor blade is described by four aero sections at radial stations (r/R) of 0.0 (root), 0.6, 0.8 and 1.0 (tip). The listing shows some of the aerodynamic parameters used in the functional model of the aero properties which gives CL, CD, CM as a function of local angle of attack, Mach

number, and Reynolds number. The variables for each aero section may be displayed or altered with the EDIT command which displays this for section #1.

```
Sect# = 1 r/R = 0.0000
=====
1) Zero-lift alpha (deg): 0.00      7) Minimum Cd      : 0.0070
2) d(Cl)/d(alpha)      : 6.280      8) Cl at minimum Cd : 0.150
3) d(Cl)/d(alpha)@stall : 0.100      9) d(Cd)/d(Cl**2)   : 0.0040
4) Maximum Cl          : 2.00        10) Reference Re number : 2000000.
5) Minimum Cl          : -1.50       11) Re scaling exponent : -0.2000
6) Cl increment to stall: 0.200      12) Cm              : -0.100
                                   13) Mcrit              : 0.620
=====
```

.EDIT c>

Note that the user may change the lift curve slope (at M=0, Prandtl-Glauert scaling is used at higher Mach numbers), the post-stall lift slope, the CLmax, the CLmin and the delta CL for the stall transition region. Other inputs set the CDmin, CL for CDmin and the quadratic drag dependence on CL. The drag is scaled by a Reynold's number scaling based on a reference Reynold's number and a scaling exponent. In the unstalled region, CD has a quadratic dependence on CL and a power-law dependence on Reynolds number as follows.

$$CD = \left| \frac{CD_o + b (CL - CL_o)^2}{(Re/Re_{ref})^f} \right|$$

where

CD_o = minimum drag coefficient ; Fortran name: (CDMIN)

CL_o = CL at which CD = CD_o (CLDMIN)

b = quadratic CD(CL) coefficient d(CD)/d(CL**2) (DCDCL2)

Re_{ref} = Reynolds Number at which CD formula applies (REREF)

f = Reynolds Number scaling exponent. (REXP)

Typically:

f = -0.1 to -0.2 for high-Re turbulent flow Re > 2M

f = -0.5 to -1.5 for "low-Re" regime Re ~ 200K..800K

f = -0.3 to -0.5 for mostly-laminar airfoils at Re < 100K

Obviously, the particular choice of f is strongly case-dependent, especially in the intermediate "low-Re" regime.

A simple separated drag model is used to add pressure drag in regions outside the linear lift range. The crude stall model is used to smoothly merge the unstalled and fully stalled lift curve portions. A crude treatment is justified on the grounds that a stalled rotor blade airfoil will have

properties vastly different than those of a stalled a 2-D section due to the presence of powerful centrifugal and Coriolis forces on the separating boundary layer fluid.

A Prandtl-Glauert compressibility correction is employed by multiplying the incompressible CL values above by the factor $1/\sqrt{1 - M^2}$, where M is the local Mach number. Further compressibility effects are included through a simple drag rise model that adds a drag term of the form

$$\Delta C_D = K (Mach - M_{crit})^n$$

where K is a scaling constant and n is an exponent (near 3 for this model). This model comes from correlations from drag rise behavior for representative NACA airfoils, its use for more exotic airfoils should be checked by the user to ensure the behavior is representative of their drag rise characteristics.

As part of the transonic drag rise model the CLmax and CLmin are reduced as the Mach number is raised towards Mcrit. This mimics the behavior in test data but the details may not be correct for all airfoils, again the user is encouraged to plot the aero characteristics (using the PLOT command under AERO) to check the behavior with Mach and alpha.

The PLOT command in the .AERO menu displays each aero section properties for $Re = R_{REF}$ and several Mach numbers.

Windmills
=====

Although XROTOR input and output is biased towards propellers, windmill design and analysis can be performed with nearly equal facility. The main thing to remember is that a windmill is a propeller with NEGATIVE blade CL's, thrust, torque, power, and also a reciprocated efficiency. About the only complication comes into the section properties CL(alpha) and CD(alpha). Since the section CL values are negative in the propeller sense, the blade airfoils are always "upside down" in a windmill. The curves must be reflected about the origin. Specifically, the following changes must be made:

a	---->	- a
o		o
CL	---->	- CL
o		o
CL	---->	- CL
min		max
CL	---->	- CL
max		min

The REFL command from the AERO menu will perform these reflections at runtime.

Of course, these can be made the default by altering the XROTOR.DEF start-up file appropriately. When a windmill is being designed, negative CL values must still be specified along with the negative thrust or power.

Caveat

=====

The code is written with many safeguards against unintended crashes. However, it is always easy to input data which will result in a non-unique or even impossible analysis or design problem. For instance, specifying power for a given windmill will in general have two solutions, one on each side of the power peak. Specifying a power above or near the peak will cause convergence failure or maybe even a program crash. Specifying thrust (tower load) instead, may help in such a situation. Specifying a thrust, torque, or power which will stall all or part of the blades sometimes leads to trouble, while specifying an advance ratio is generally safe whether or not the blades stall.

The self-deforming wake algorithm can be touchy if a high windmill disk loading is combined with very low advance ratios. In such a case, some pathological non-physical situation such as reverse far-slipstream velocity might occur. The mathematical model is incapable of handling such flows, especially if they don't exist in the real world. One can always disable the self-deforming wake algorithm to get a more robust code, although accuracy might suffer as a result.

The discrete vortex wake model was added to permit analysis of non-planar propellers but the XROTOR code does not (at present) support geometric input for more general propellers. There is an undocumented RAKE command that allows a blade to be raked upstream or downstream out of the radial plane. For such cases the VRTX option should be used to properly calculate induced velocities.