JavaProp – Users Guide

June 2009
Martin Hepperle

Contents

Symbols and Formulas

In the field of propellers and windmills a variety of definitions are used to describe operating points and performance. JAVAPROP follows mostly the traditional American notation as described in the following tables.
Note that in some publications coefficients of the same name (e.g. T_C) are used, which follow their own definitions – so be careful when comparing results.

Symbol	Description	Unit
D	diameter	m
D_{sp}	spinner or hub diameter	m
$R = \frac{D}{2}$	radius	m
$n = \frac{RPM}{60}$	rotations per second	1/s
P	power	W
T	thrust	N
v_∞	axial inflow speed (flight speed, wind speed)	m/s
$S = \pi \cdot R^2 = \pi \cdot \frac{D^2}{4}$	disc area	m^2
ρ	density of medium	kg/m^3
$\Omega = 2 \cdot \pi \cdot n$	angular speed	1/s

Description	Definition	Conversions
thrust coefficient (propeller)	$C_T = \frac{T}{\rho \cdot n^2 \cdot D^4}$	$= \frac{\pi}{8} \cdot T_C \cdot \left(\frac{v_\infty}{n \cdot D}\right)^2$ $= \frac{\pi}{8} \cdot \eta \cdot P_C \cdot \left(\frac{v_\infty}{n \cdot D}\right)^2$ $= \eta \cdot \frac{C_P}{\left(\frac{v_\infty}{n \cdot D}\right)}$
thrust coefficient (propeller)	$T_C = \frac{T}{\frac{\rho}{2} \cdot {v_\infty}^2 \cdot S}$	$= \frac{T}{\frac{\rho}{2} \cdot {v_\infty}^2 \cdot \pi \cdot R^2}$ $= \frac{8}{\pi} \cdot \frac{C_T}{\left(\frac{v_\infty}{n \cdot D}\right)^2}$ $= \frac{8}{\pi} \cdot \eta \cdot \frac{C_P}{\left(\frac{v_\infty}{n \cdot D}\right)^3}$
power coefficient (propeller)	$C_P = \frac{P}{\rho \cdot n^3 \cdot D^5}$	$= \frac{\pi}{8} \cdot P_C \cdot \left(\frac{v_\infty}{n \cdot D}\right)^3$ $= \frac{\pi}{8} \cdot \frac{1}{\eta} \cdot T_C \cdot \left(\frac{v_\infty}{n \cdot D}\right)^3$ $= \frac{1}{\eta} \cdot \frac{C_T}{\left(\frac{v_\infty}{n \cdot D}\right)}$ $= \left(\left(\frac{v_\infty}{n \cdot D}\right) \cdot \frac{1}{C_S}\right)^5$
power coefficient (propeller, wind turbine)	$P_C = \frac{P}{\frac{\rho}{2} \cdot {v_\infty}^3 \cdot S} = C_{P\ \text{wind turbine}}$	$= \frac{8 \cdot P}{\rho \cdot v_\infty^3 \cdot \pi \cdot D^2}$ $= \frac{P}{\frac{\rho}{2} \cdot {v_\infty}^3 \cdot \pi \cdot R^2}$ $= \frac{8}{\pi} \cdot \frac{C_P}{\left(\frac{v_\infty}{n \cdot D}\right)^3}$ $= \frac{8}{\pi} \cdot \frac{1}{\eta} \cdot \frac{C_T}{\left(\frac{v_\infty}{n \cdot D}\right)^2}$
efficiency (propeller)	$\eta = \frac{T \cdot v}{P}$	$= \frac{T_C}{P_C}$ $= \frac{C_T}{C_P} \cdot \left(\frac{v_\infty}{n \cdot D}\right)$
advance ratio (propeller)	$J = \frac{v_\infty}{n \cdot D}$	$= \pi \cdot \lambda$
advance ratio (propeller)	$\lambda = \frac{v_\infty}{\Omega \cdot R}$	$= \frac{1}{\pi} \cdot \frac{v_\infty}{n \cdot D}$
tip speed ratio (wind mill)	$X = \frac{\Omega \cdot R}{v_\infty}$	$= \frac{1}{\lambda}$

Propellers

How to design a propeller

JAVAPROP contains a powerful inverse design module. Inverse design means that you specify only a few basic parameters and JavaFoil produces a geometry which has the maximum efficiency for the selected design parameters. The beautiful thing is that JAVAPROP creates an optimum propeller with just 5 design parameters plus a selection of airfoil operating points along the radius. You can later modify this design to adapt to additional off-design conditions.

The following cards are relevant for the design of a propeller:

- Design,
- Airfoils,
- Options.

Parameters on the Design card

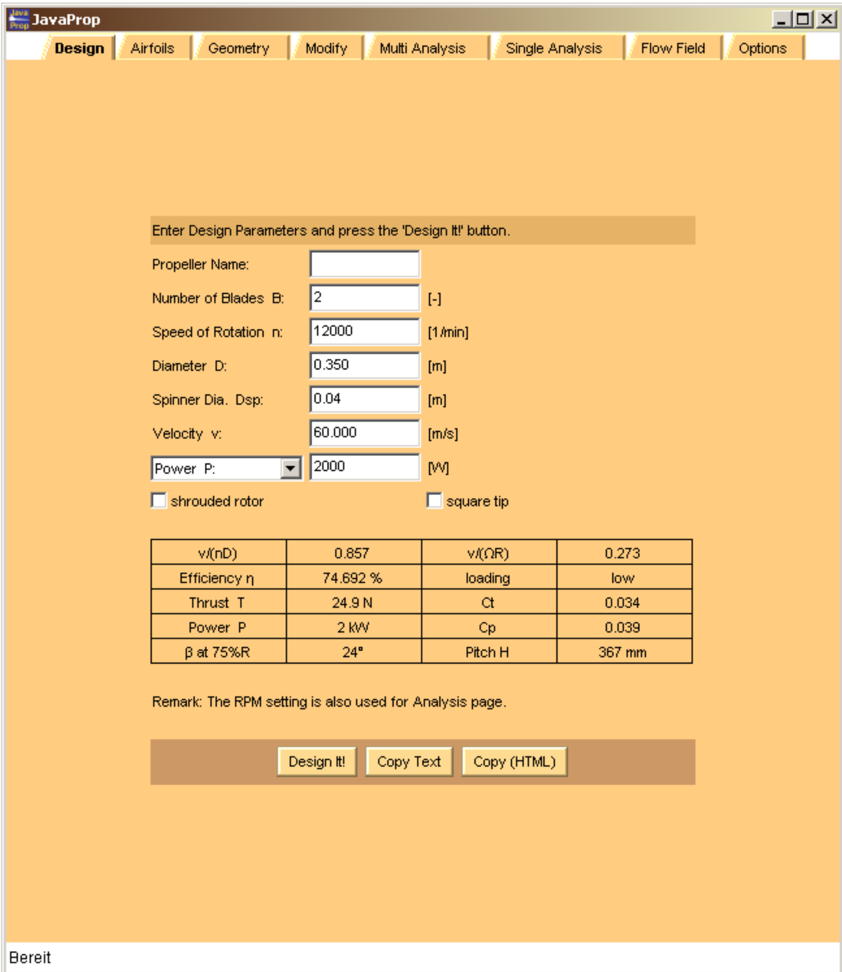


Figure 1: Design card after a design has been performed.

The design card holds most of the parameters which are required for a design.
It is possible to perform a design for either

- power (the propeller will consume the specified power),
- thrust (the propeller will produce the specified thrust), or
- torque (the propeller will consume the specified torque).

Shrouded propeller option

The thrust distribution along a propeller with free tips drops to zero at the tips. If a shroud is added to the propeller, this "tip loss" is suppressed. Note that this is only a crude approximation of the real flow field because the shroud itself affects the flow through the propeller and can create additional thrust, especially at low flight speeds. Also interaction between shroud shape and propeller would require a more complete model of such a system. A detailed modeling of such configurations is beyond the targeted capabilities of JAVAPROP.

Square tip option

The optimum design procedure creates blades with rounded tips. As this is not always practical the option "square tip" produces a tip with finite chord length by simple extrapolation of the last section.

Airfoils card

In addition to the basic parameters on the Design card, airfoils have to be selected and their operating point must be specified on the Airfoils card.
JAVAPROP comes with several built-in airfoil sections. For each section tables of lift and drag coefficients versus angle of attack are shown on the Airfoils card. For the design it is necessary to assign airfoil sections to four radial stations – JAVAPROP interpolates linearly between these design sections. You define lift- and drag coefficient by selecting a design angle of attack for each section. Note that the absolute maximum efficiency is obtained when the airfoil sections are operated at the individual maximum L/D. For real world propellers, which must also be useable at low speed off-design conditions, it is usually better to select angles of attack, which are lower than the point of maximum L/D. This is especially true for the inner sections towards the root, which see a large variation of angle of attack with forward speed.

How to use your own airfoil polars

JavaProp comes with a set of canned airfoil polars. These are sufficient for first steps and for understanding the main design parameters. Nevertheless, some users may want to add their own airfoil data. This is possible in case of a local installation by copying polar data files into the JAVAPROP installation directory. This option is not available when JAVAPROP is run via WEBSTART or as an APPLET in a browser, because these applications cannot access your computers file system for security reasons.

The polar data files must be named "af_#.EXT", where the # character represents a serial number and the extension EXT is either "afl" or "xml". JAVAPROP will search for files beginning with "af_1", trying the extension ".afl" first. If no matching file is found JAVAPROP tries the alternative file name ending in "xml". If either file was found it is read in and the airfoil index is incremented until no corresponding file is found (because no more files exist or the file names have a gap in their numbering).

The polar data files in "xml" format are in my standard XML format .These can be created with JAVAFOIL's Polar card and saving the polar with the extension ".xml". You can have as many data points in the regime from -180° to +180°, but it is usually sufficient to provide polars with a range from -45° to +45° in steps of 2.5 degrees. JAVAPROP adds data points at +/- 90° automatically if not supplied. In order to achieve realistic results it makes sense to select at best a NACA standard roughness and no perfect surface finish. Note that only the first configuration (Reynolds number) is read.

Polar data files in "afl" format are primitive text files with exactly 5 header lines, followed by data points describing the airfoil polars. The first line contains the airfoil name and will appear in the dropdown list boxes of the Airfoils card.

```
Tabulated Airfoil 1
This is an airfoil polar file for JavaProp. It can have up to 1000 data triples.
This format will be changed to my airfoil-polar-XML form in a future release.
---------------------------------------------------------
alpha         cl        cd        cm
-180.00000  0.00000  0.49786  -0.13940
-175.00000  0.19970  0.27181  -0.07611
... some polar points omitted...
178.00000  -0.08022  0.00005  -0.00001
179.00000  -0.04013  0.00001  -0.00000
180.00000  -0.00000  0.00000  -0.00000
```

Figure 2: Example of a tabulated airfoil polar data set. Some data points have been omitted for clarity.

It is recommended to use the XML format, the AFL format is only there for backwards compatibility.
Note that the polars must also include the stall delay effect due to 3D effects on the rotating blades. JAVAPROP does not modify the given polars for this effect because there is no general method to do so. Many stall delay models exist and each fits only a limited class of cases. The pitching moment coefficients are not used by JAVAPROP.

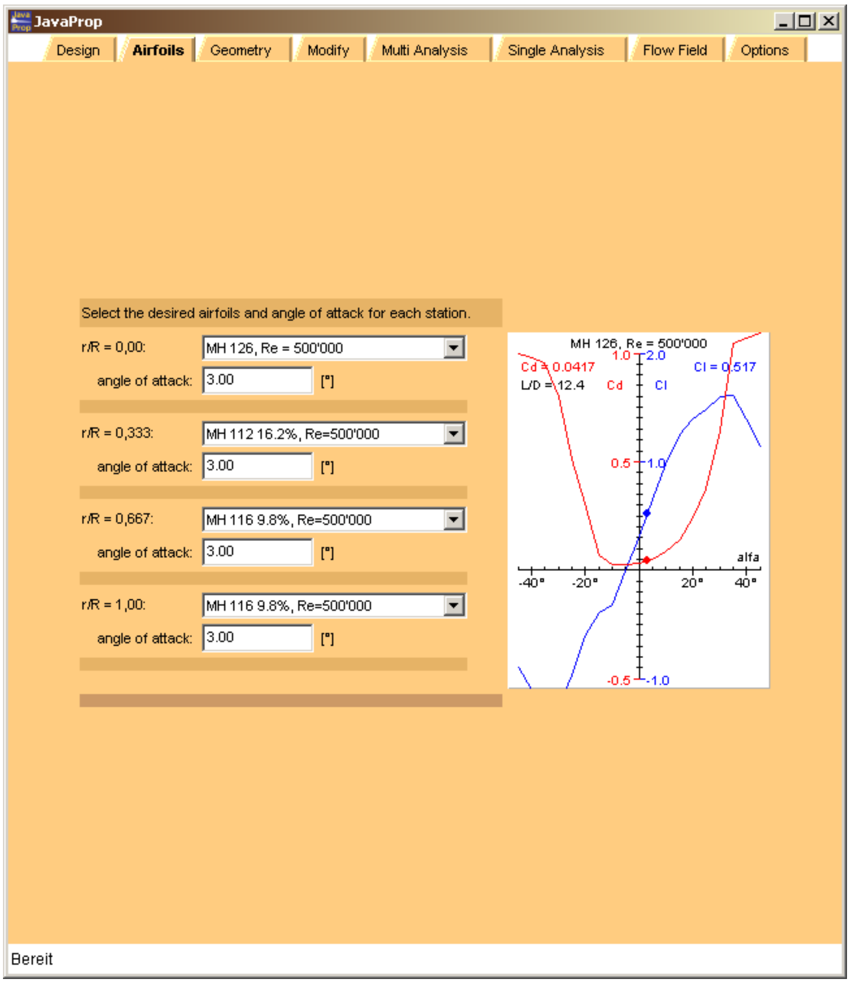


Figure 3: Airfoils card with lift and drag coefficients over the angle of attack.

Parameters on the Options card

Finally the density of the fluid from the Options card is used for the design. A propeller designed for a low density medium (e.g. high altitude) must have blades of a wider chord length than a design for a high density medium. This difference is also visible when a hydroprop is designed – the density of water is roughly 1000 higher than the density of air. Therefore a propeller for underwater operation would have blades of only 1/1000 the chord length of an aircraft propeller – if the diameter and the design lift coefficients were the same.

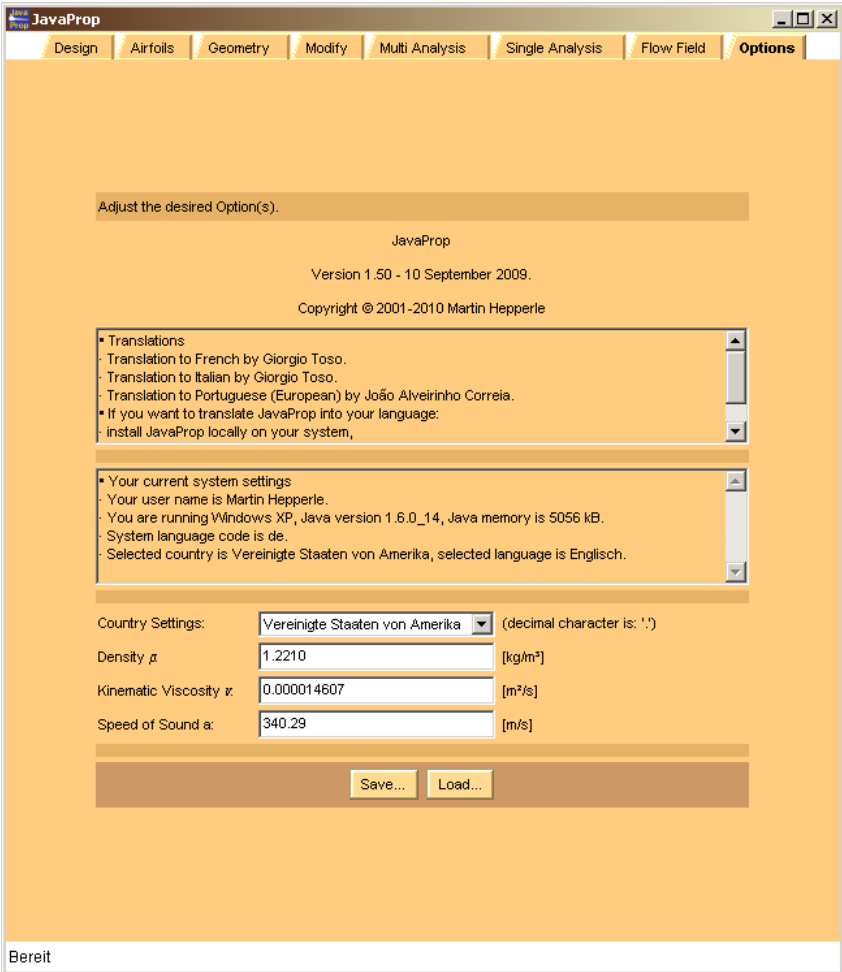


Figure 4: The Options card holds the density of the medium.

The Geometry card

This card (Figure 5) presents the geometry of the current propeller in form of a table and in graphical form. It also allows you to export the geometry in form of text files or as a 3D geometry in form of an AutoCAD DXF file. Note that this is intended for illustration purposes: while the geometry is exported correct, the resolution is probably too low for e.g. CAD machining.

Finally this card offers the option to import a given propeller geometry (Figure 6). The table must contain the planform as well as the blade angle in columns arranged “r/R”, “c/R”, “beta”. Note that the blade angle must be specified in degrees. An example data set can be produced by copying the current propeller in text format to the clipboard and then opening the “Geometry Import” form. This also allows for manual modifications of the current propeller. You can copy and paste your prepared data via your systems clipboard. During the import process, JAVAPROP tries to be smart and skips non-numeric data, but it is a good idea to keep to the proposed format.

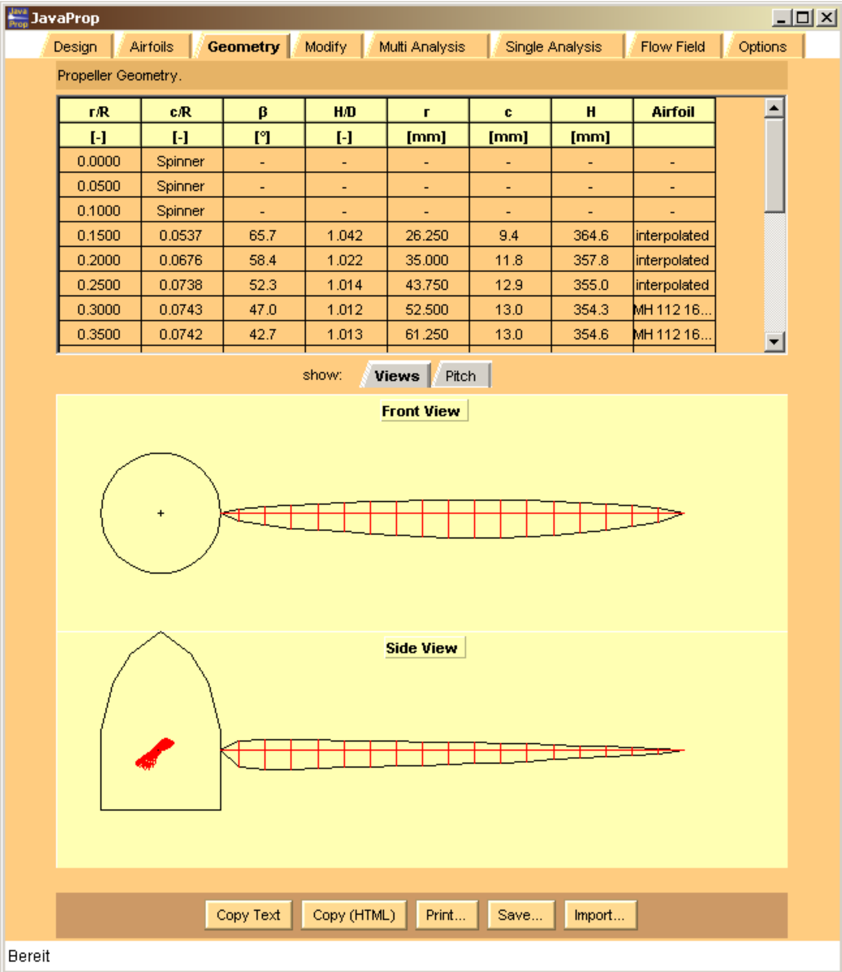


Figure 5: Geometry card with current propeller.

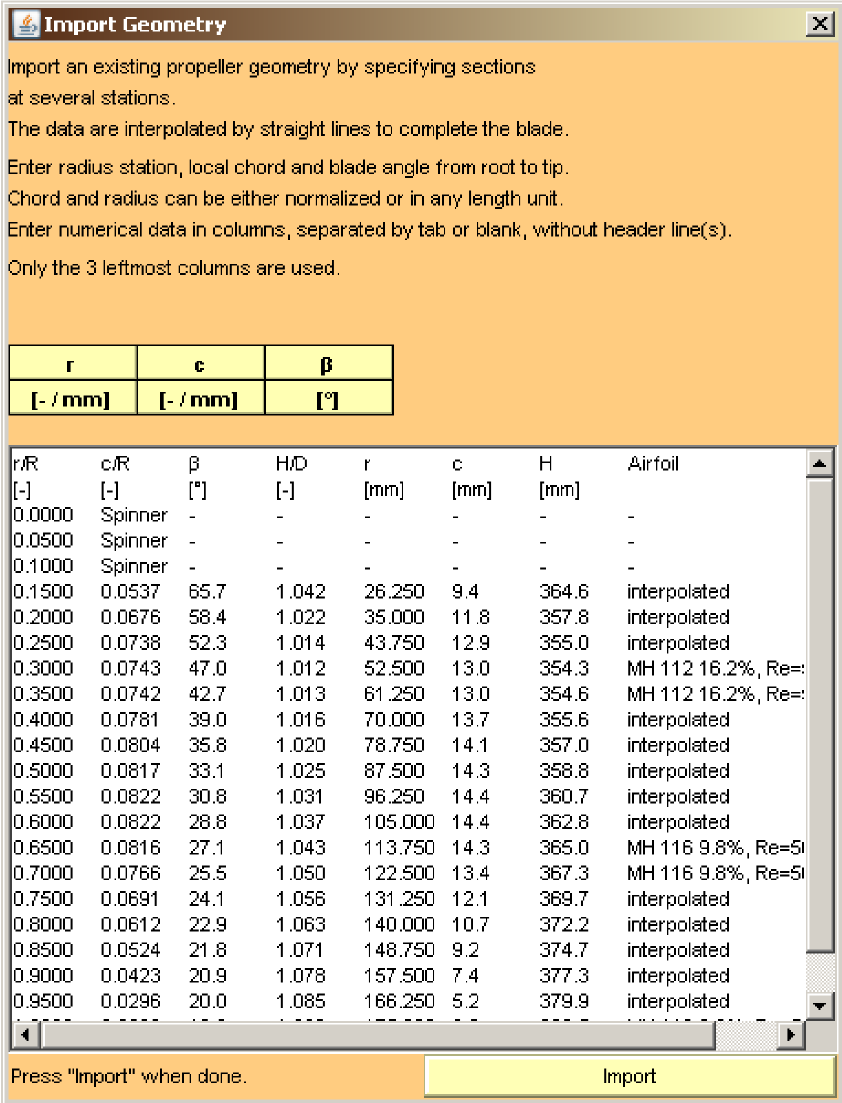


Figure 6: Geometry import form with example data.

The Modify card

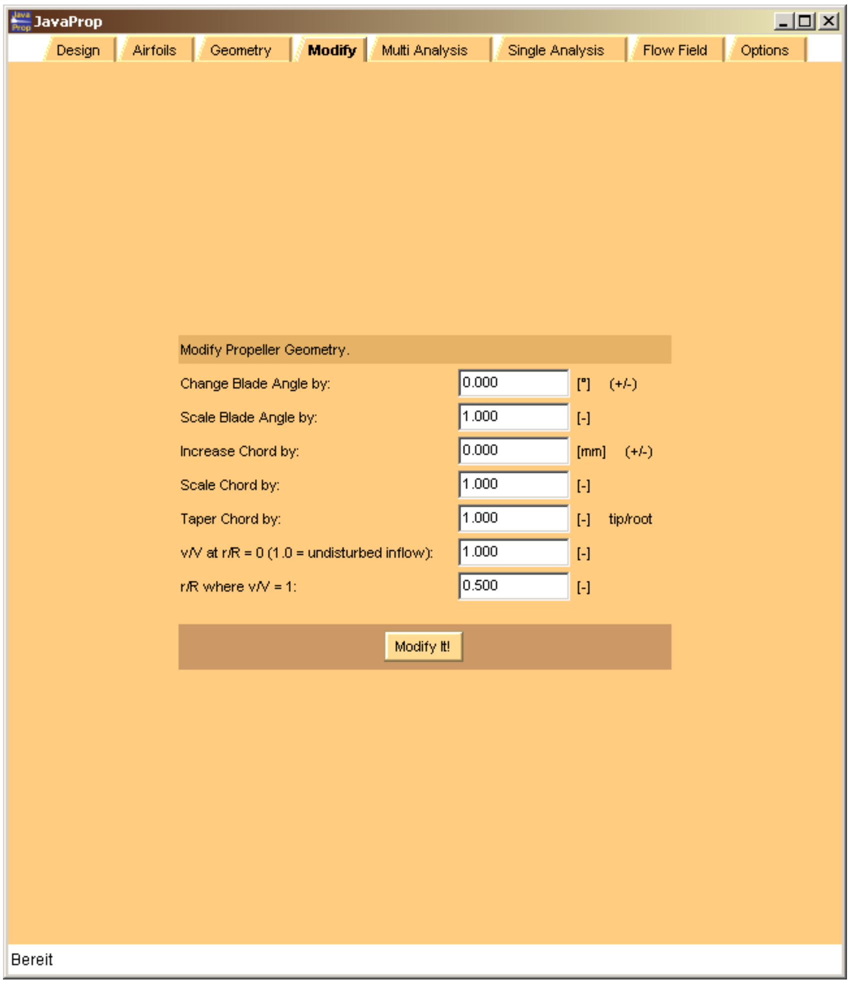


Figure 7: Modifications of the blade geometry can be performed using the Modify card.

How to analyze a propeller

JAVAPROP can analyze propellers at arbitrary operating points. The propellers can be created by the design module of JAVAPROP or by importing a given geometry. There are two cards available for analysis:

- Multi-Analysis,
- Single-Analysis.

Both cards differ in their analysis range and in the level of detail of their output.

Usage of the Multi-Analysis card

The Multi-Analysis card is used to analyze the propeller over its complete useable operating range from static operation up to the beginning of the windmilling regime at high speeds. The output of this card consists of the global propeller data like thrust, power or efficiency versus advance speed.
The coefficients shown on this card are generally applicable performance parameters. n the other hand the absolute values like thrust or power are calculated using these coefficients plus additional data taken from the

- design card (diameter, and one of n, P, T or Q),
- options card (density).

You can change any of these values and perform an additional analysis to study their effect. As long as you do not modify the geometry, the coefficients will always be the same, but the absolute values will change.

The four different cases for the calculation of the absolute values represent:

- n=given – constant speed propeller, P, T, Q vary with air speed,
- P=given – n is adjusted so that the propeller consumes the given power,
- T=given – n is adjusted so that the propeller produces the given thrust,
- Q=given – n is adjusted so that the propeller consumes the given torque.

In order to analyze for example a constant speed propeller at different speeds of rotation n, you would just change the value of n on the design card and then perform and additional Multi-Analysis. Careful: do not perform a new design on the Design card – this would create a new blade shape.

Note that none of these cases represents a propeller operating on a given engine because for simplicity no engine performance curve model is used in JAVAPROP. While the “constant speed” (n=give) is a realistic operating procedure, the constant P, T or Q methods are somewhat artificial, but can be used to get an overview of the basic characteristics. Note that all these dimensional results are obtained from the single set of thrust and power coefficients versus advance ratio.

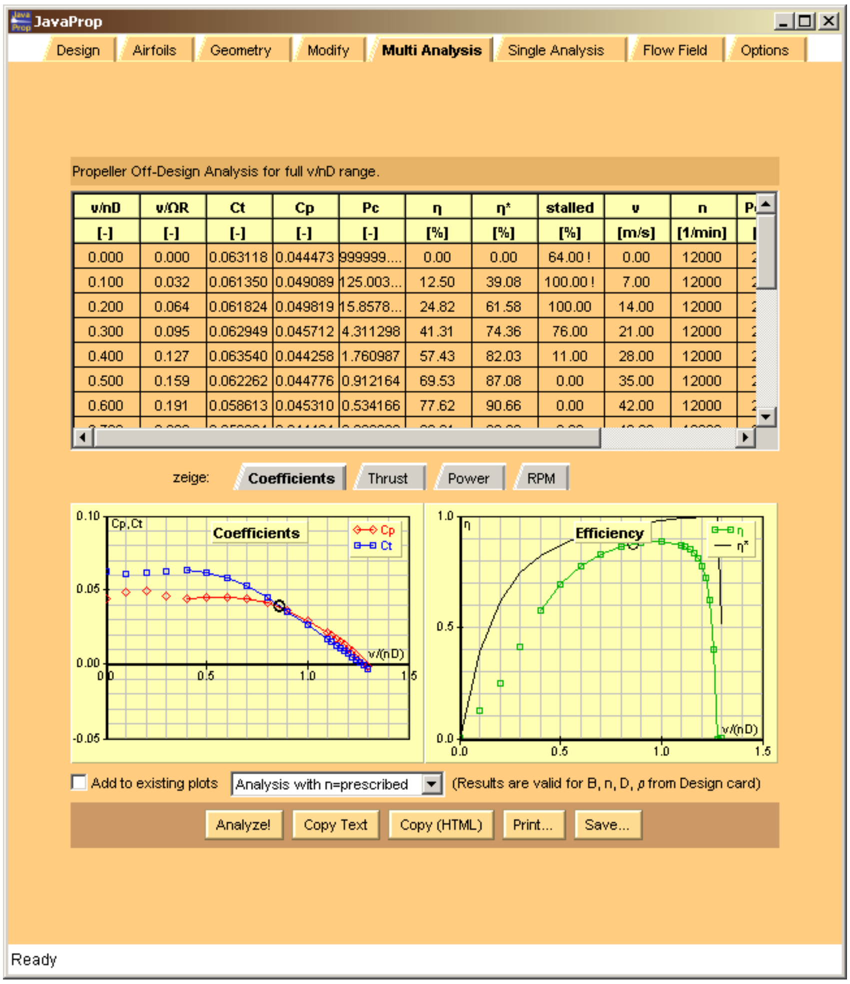
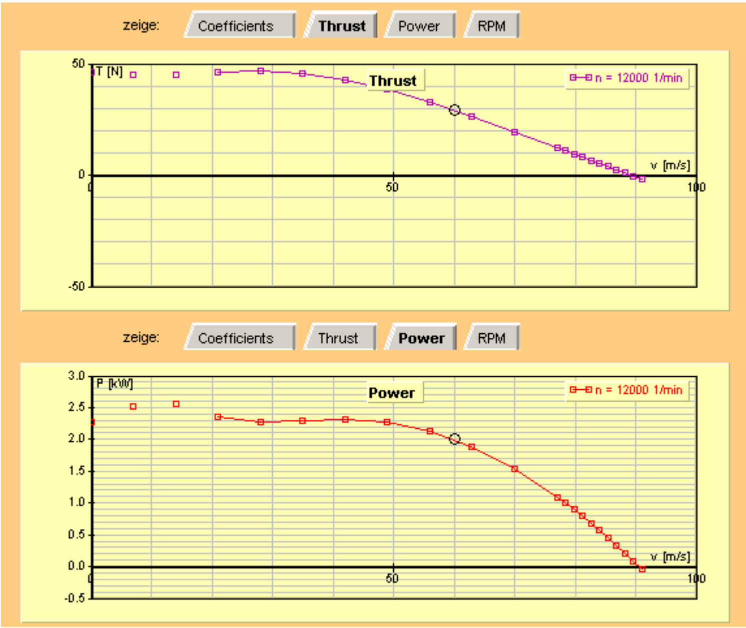


Figure 8: The Multi-Analysis card produces global propeller coefficients over a range of operating conditions.



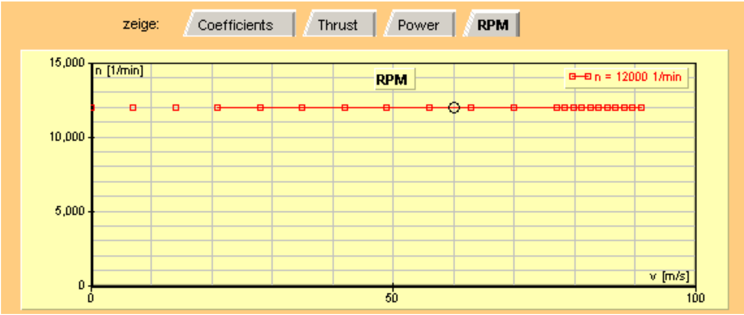


Figure 9: The individual graphs on the Multi-Analysis card present thrust, power and RPM versus flight speed for the selected operating mode.

Usage of the Single-Analysis card

The Single-Analysis card is used to analyze the propeller at a single, arbitrary operating point. This point is specified by the flight speed v, rotational speed n and diameter D on the Design card, which define an advance ratio

The output of the Single Analysis card is more detailed than that of the Multi-Analysis card. It consists of distribution of aerodynamic data along the radius of the blade and includes coefficients related to structural loads (shear force and bending moment) as well.

Propeller

r/R	relative radial station
α	angle of attack in degrees
C_l	lift coefficient
C_d	drag coefficient
L/D	ratio lift to drag
Re	Reynolds number
Ma	Mach number
a	axial induction factor (axial velocity through the propeller is $v_\infty \cdot (1+a)$)
a'	tangential induction factor(tangential velocity at the propeller is $(\Omega \cdot r \cdot (1-a'))$)
C_x	tangential (in-plane) force coefficient for a blade element $C_x = \frac{F_x}{\frac{\rho_\infty}{2} \cdot v_{eff}^2 \cdot c \cdot dr}$
C_y	thrust force coefficient for a blade element) $C_y = \frac{F_y}{\frac{\rho_\infty}{2} \cdot v_{eff}^2 \cdot c \cdot dr}$
δ	swirl angle at propeller in degrees
δ_{ff}	swirl angle far behind the propeller in degrees (integrated tip to root)
C_{Qx}	tangential (in-plane) shear force coefficient (integrated tip to root)
C_{Mx}	tangential (in-plane) bending moment coefficient (integrated tip to root)
C_{Qy}	normal shear force coefficient (integrated tip to root)
C_{My}	normal bending moment coefficient (integrated tip to root)

Table 1: Description of the tabular results on the Single Analysis card.

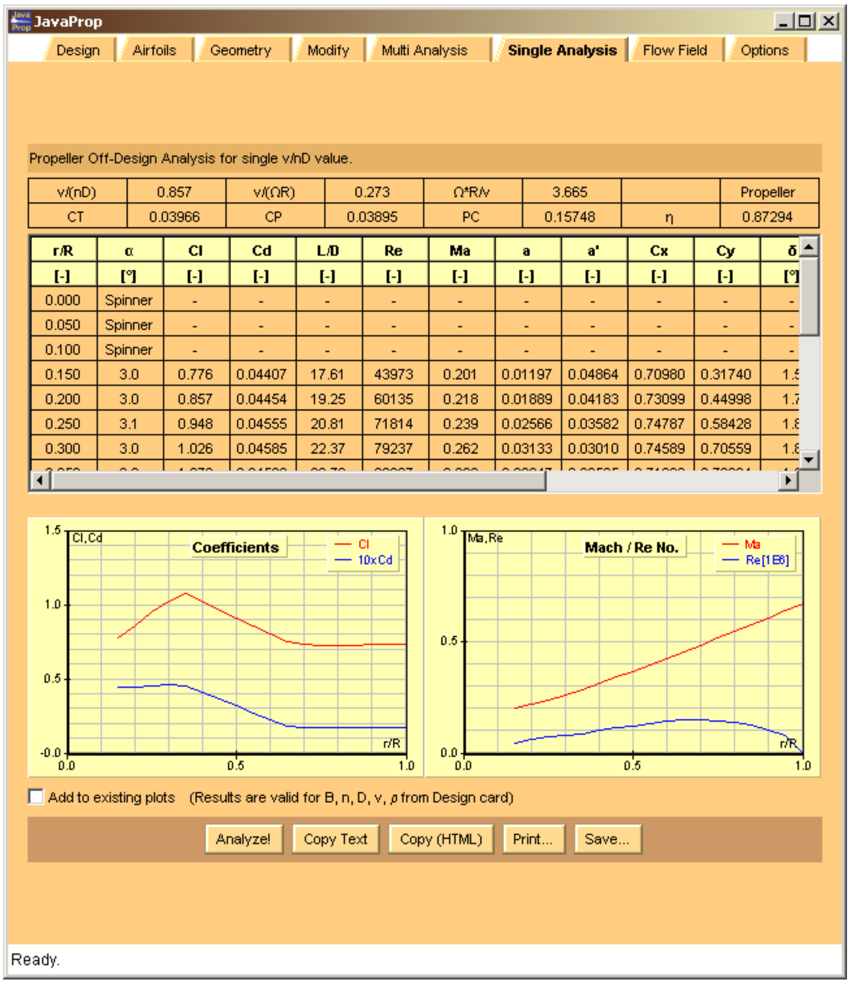


Figure 10: The Single-Analysis card produces detailed propeller data for a single operating condition.

Definition of shear force and bending moment coefficients

In order to assess loads on propeller blades the local aerodynamic forces represented by the local coefficients C_x and C_y are transformed into global coefficients.
These coefficients are defined similar to the thrust and torque coefficients of the propeller.
The shear force and bending moments are integrated from tip to root.

Shear force due to out-of-plane axial force (thrust)

$$Q_y = C_{Q,y} \cdot \rho \cdot n^2 \cdot D^4$$

Shear force due to in-plane tangential force (torque/r)

$$Q_x = C_{Q,x} \cdot \rho \cdot n^2 \cdot D^4$$

Bending moment due to out-of-plane axial force (thrust)

$$M_y = C_{M,y} \cdot \rho \cdot n^2 \cdot D^5$$

Bending moment due to in-plane tangential force (torque/r)

$$M_x = C_{M,x} \cdot \rho \cdot n^2 \cdot D^5$$

Some simple validation checks

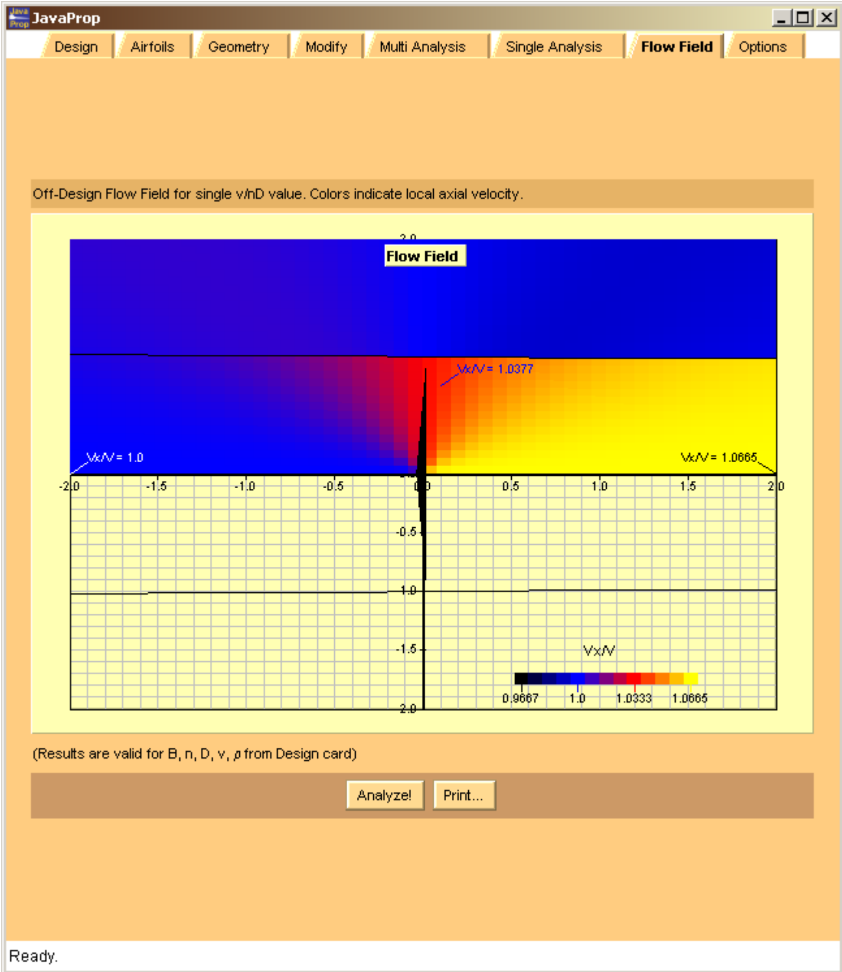
A quick validity check is the axial shear force coefficient at the root must be equal to the thrust coefficient divided by the number of blades;

$$Q_y = \frac{C_T}{n_{blades}}$$

Another plausibility check is the center of thrust of each blade which is representing the thrust of the blade by a single force acting at

$$\frac{r}{R} = 2 \cdot \frac{C_{M\,y}(0)}{C_{Q\,y}(0)}.$$

Most propellers have their thrust force located between 60 and 70% of the radius.



Windmills

While originally being designed as a propeller design and analysis tool, JAVAPROP can also be used for windmills. Some differences must be considered, though. Figure 11 shows the general power and thrust curves of a general rotor, covering a wide speed range. In the case of propellers, only the left hand side of the graph is of interest, for windmills it is the right hand side. The transition between propeller and windmill state are fluent. Any fixed pitch propeller running at constant speed of rotation will eventually reach the windmilling state. When the air speed is increased further, it will act as a windmill, albeit a relatively poor one. This is because the airfoils on a windmill operate a negative lift and hence must be applied “upside down”. Note also, that between the propeller and the windmill regimes there is a small range of advance ratios where the propeller already produces drag but still consumes power. This is a not very useful condition as the propeller merely creates entropy (heat). This effect is cause by friction and induced losses due to the radial lift distribution and cannot be avoided. Luckily this is only a very narrow band.

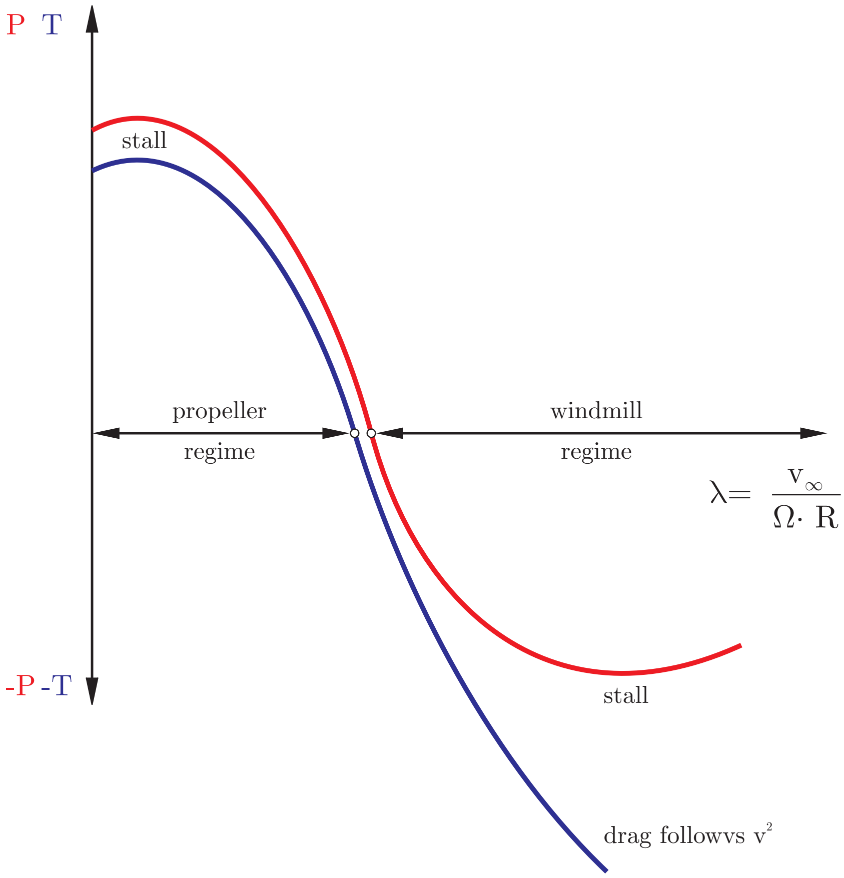


Figure 11: General operating characteristics of propellers and windmills, plotted versus advance ratio.

For the aerodynamic design and analysis of windmills the methods used for propellers can be applied. The analysis routines are the same, while the optimum design method is different, because the figure of merit of a windmill is different from the efficiency of a propeller. For a propeller the figure of merit is how much thrust can be generated for a given input power. The efficiency of a windmill can be expressed in how much energy is extracted from the mass of air passing through the rotor disc in relation to the amount of energy contained in this stream of air. The drag (negative thrust) acting on the tower is of no primary interest, only the amount of power extracted.

Because the performance characteristics of a windmill start where the operating range of the propeller ends, this windmilling regime does not start at a wind speed of zero. There is a required minimum wind speed at which the windmill can start to turn. This has some implications on the design parameters, as a design for a too low advance ratio $v/(n \cdot D)$ would not work.
The design method implemented in JavaProp is based on publications by Prandtl, Betz and Glauert in the 1930s. It neglects friction forces, which can lead to a seemingly successful design which will not work when analyzed afterwards because the analysis takes friction into account. In this case you should move the design advance ratio to a higher value, e.g. by reducing the speed of rotation n. Figure 12 gives an overview of typical operating parameters depending on the windmill size.

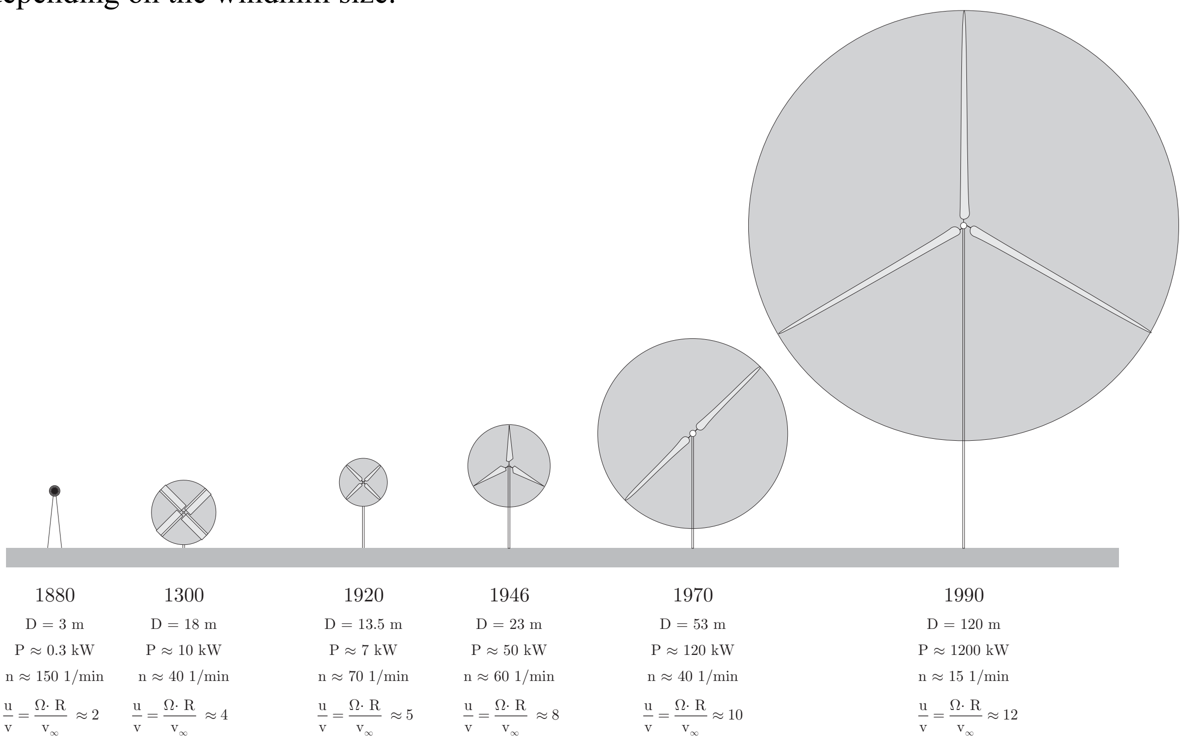


Figure 12: Typical operating parameters of different windmills types and sizes.

Note that JAVAPROP sticks to its propeller roots in maintaining the usual propeller coefficients and plots. This leads to windmills having negative values for torque and power as well as thrust. This sign change indicates that power and torque are delivered, not consumed and that the thrust is actually a drag force, acting on the tower. Also the graphs of coefficients versus advance ratio are different from the common graphing of coefficients versus the tip speed ratio X, which is the reciprocal of the advance ratio λ, i.e. $X = 1/\lambda = \Omega \cdot R / v_\infty$. Keeping the propeller conventions is not too inconvenient though, as the graphs versus $v_\infty/(n \cdot D)$ still display the behavior versus wind speed for a constant speed of rotation. Note especially, theta the power coefficient C_P as commonly used for windmills is not identical to the power coefficient C_P of propellers, but to the coefficient P_C.

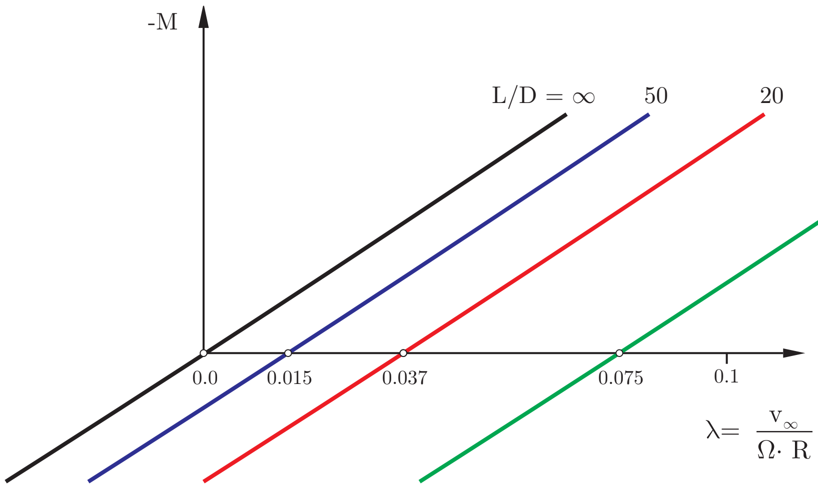


Figure 13: Effect of airfoil L/D ratio on the start speed of a windmill.

How to design a wind turbine

JavaProp can design an optimum wind turbine for a given wind speed, speed of rotation and diameter. In order to specify the design of a wind turbine, you just enter a negative value for the power on the Design card. The power value itself is not used, only its sign is checked. Everything else is identical to the propeller design, the data on the Airfoils and the Options cards are used for the design.
The design follows the method of Glauert and therefore takes swirl losses into account but neglects friction losses. JavaProp determines the efficiency of a windmill as the ratio of the power coefficient P_C for windmills) to the power coefficient P_C^* which represents the maximum power which could be extracted from the stream tube passing through the rotor. Swirl losses become very large when the tip speed ratio X is considerably lower than 1.0, i.e. the wind turbine is turning too slow. At high tip speed ratios the power coefficient approaches the limit derived by Betz for zero swirl, i.e. $P_C = 16/27$.

One main difference between the geometry of a wind turbine and a propeller is the orientation of the airfoil sections. JavaProp automatically turned the airfoils upside down when a wind turbine is designed and also maintains these inverted airfoils for the analysis. In terms of geometry you will note that e.g. in the DXF output of the Geometry card all airfoils are arranged with an upside down orientation.

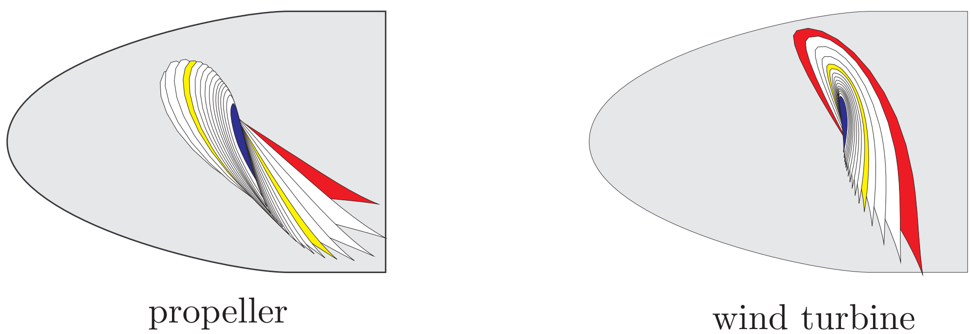


Figure 14: Orientation of airfoil sections on a propeller and on a wind turbine.

As the airfoils are automatically turned upside down, the design must be performed with a positive lift coefficient, typically close to the maximum L/D of each airfoil section. Note that the design may lead to unrealistic shapes, which will not lead to realistic results in subsequent analysis. This is usually the case when the advance ratio is too far off reality. Compared to

propeller design, the useful range of advance ratios is smaller, so that some experimentation ay be required. Typical advance ratios $v/(n \cdot D)$ for windmills are in the order of $v_\infty/(n \cdot D) = 0.5 0.6$ to $v_\infty/(n \cdot D) = 1.5$. It is also recommended to use more sophisticated airfoils, not for example the flat plate.
The design method can be used to produce a first geometry, which is later refined and modified to suit additional requirements, like a desired maximum chord length.

Large Windmill		
diameter D	120	m
spinner $D_{spinner}$	4	m
speed of rotation n	12	1/min
velocity v	10	m/s
number of blades B	3	
airfoils		
r/R=0.0	MH 126	α=14°
r/R=0.333	MH 112	α=9°
r/R=0.667	MH 116	α=6°
r/R=1.0	MH 116	α=5°

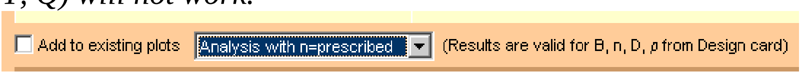
Small Windmill		
diameter D	0.35	m
spinner $D_{spinner}$	0.05	m
speed of rotation n	1600	1/min
velocity v	7	m/s
number of blades B	2	
airfoils		
r/R=0.0	E 193	α=8°
r/R=0.333	E 193	α=7°
r/R=0.667	E 193	α=6°
r/R=1.0	E 193	α=3°

Table 2: Example cases: "Large Windmill" and "Small Windmill".

How to analyze a wind turbine

The single and multiple operating point analyses of JAVAPROP are performed exactly in the same way as the propeller analysis. At times, the analysis of wind turbines can be a bit more sensitive – it is recommended to use a spinner to blank the inner most sections and to use reasonable airfoil sections, not the flat plate.
For simplicity, the coefficients are also the usual propeller related coefficients, i.e. the power coefficient C_P is different from the power coefficient for windmills which is named P_C according to propeller terminology.
Note also, that many parameters, like the power coefficient or the axial and circumferential induction factors of a wind turbine are output as a negative value because it delivers power instead of the propeller which absorbs power.
For comparison with wind turbine codes you should compare the coefficient P_C versus the reciprocal of the advance ratio λ, which equals the tip speed ratio X as used for wind turbines.

Note that the analysis of wind turbines in JavaProp must be performed for a constant value of n (which can be specified on the Design card) – the other options (constant P, T, Q) will not work.



Validation of JavaProp

JAVAPROP is a rather simple but nevertheless useful tool. The underlying blade-element-momentum theory allows for the design and analysis of typical propellers as long as

- the number of blades is small (say below 10) so that no strong interaction due to thickness occurs,
- three-dimensional effects are small (no winglets), and
- compressible flow effects are small ($M_{tip} < 1.0$).

To compare the results of JAVAPROP with experimental data, a set of test results for a propeller according to design 5868-9 of the Navy Bureau of Aeronautics was selected. The propeller geometry as well as the test data can be found in NACA Report 594. The propeller used airfoils of the Clark Y type and had 3 blades. Data shown in Figure 16 are for the configuration “Nose 6, Propeller C”. The blade geometry according to the NACA report has been imported into JAVAPROP via the geometry card. The blade angle was adjusted to match the given angles at 75% of the radius and the results produced by the Multi-Analysis card were collected in an Excel spreadsheet. No further tweaking was performed.
The comparison shows that JAVAPROP predicts the general performance characteristics in the typical “linear” operating range quite well. For this example, thrust and power are somewhat under-predicted, indicating that possibly the zero lift angle of the Clark Y airfoil in JAVAPROP might me too low. A more likely explanation however, is that the blade angle of the NACA tests refers to the flat underside of the blade while JAVAPROP uses the x-axis of the airfoil section for reference. Unfortunately the NACA reports do not give a clear indication, how exactly the blade angle was measured. in case of a Clark Y airfoil having 12% thickness, the difference amounts to about 2°.

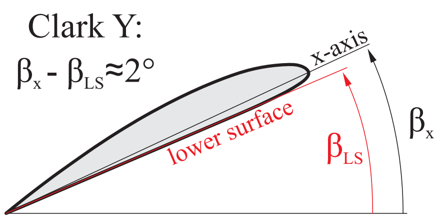


Figure 15: Possible reference lines for blade angle measurement.

Similar levels of the efficiency indicate that the lift to drag ratio of the Clark Y airfoil model in JAVAFOIL corresponds well to the tests.
Large deviations occur in the regions towards the left, where the propeller stalls. Here the flow is largely separated, three dimensional, unsteady and also depending on the external flow field (e.g. crosswind, wind tunnel interference). Such flow regimes are beyond the assumptions of the underling theory so that no good match can be expected here. It should be noted, that the experimental data show considerable scatter and irregular behavior in this regime too.

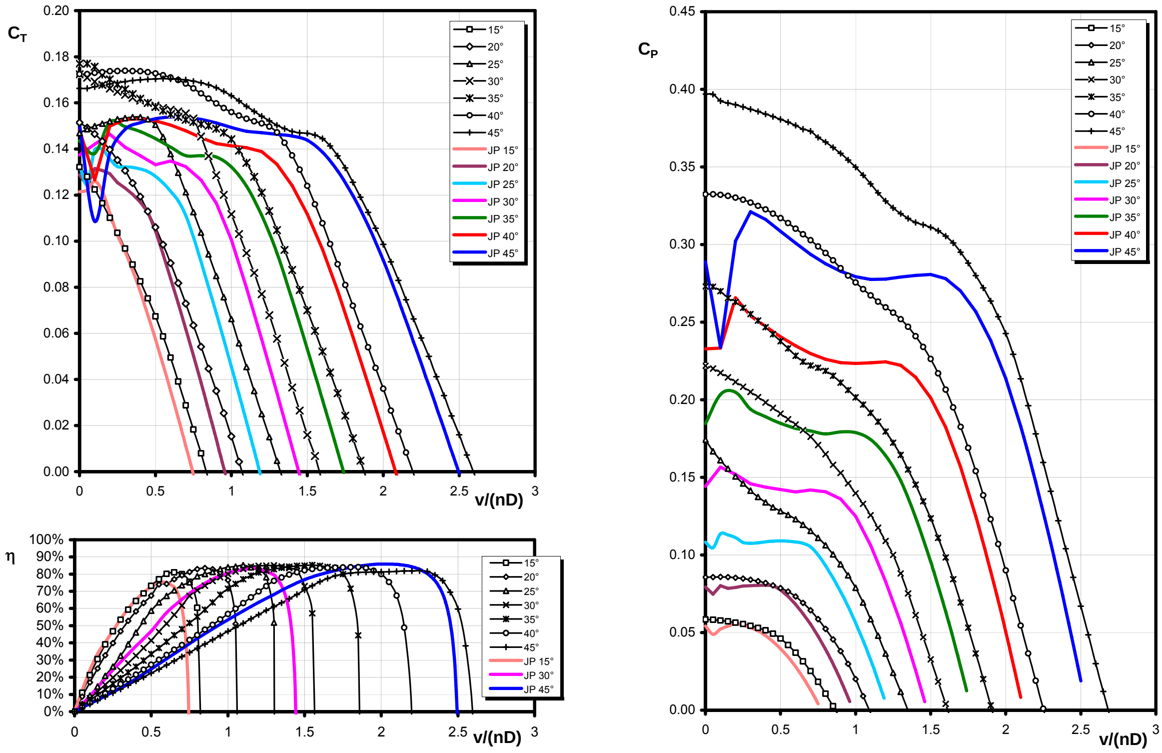


Figure 16: Comparison of data predicted by JAVAPROP with experimental data.