OpenProp v2.4 Theory Document

Brenden Epps[1]

December 15, 2010

1 Introduction

OpenProp is an open-source code suite that can be used for the design, analysis, and fabrication of optimized propellers and horizontal-axis turbines. The numerical model is based on propeller lifting line theory, which is used in parametric design codes employed by the U.S. Navy as well as commercial designers. OpenProp is written in Matlab M-code, which is widely used in academia and industry. OpenProp is designed to be a user-friendly tool that can be used by both propeller design professionals as well as novices to propeller design.

A team of researchers at MIT and Maine Maritime Academy have contributed to the current OpenProp code. OpenProp began in 2001 with the propeller code PVL developed by Kerwin (2007) as part of his MIT propeller design course notes. The first Matlab version of this code, MPVL, incorporated graphical user interfaces for parametric design and preliminary bladerow design (Chung, 2007). Geometry routines were later added which interfaced with the CAD program Rhino to generate a 3D printable propeller (D'Epagnier et al, 2007). These early codes were capable of designing propellers using a simple Lerb's criteria optimizer routine (Lerbs, 1952). Epps et al (2009b) implemented Coney's generalized propeller optimizer (Coney, 1989) and also created a turbine optimization routine. Epps et al (2009a) created an off-design analysis routine to predict the performance curve for a given propeller or turbine design. On- and off-design cavitation analysis capabilities were implemented by Flood (2009). Stubblefield (2008) extended the numerical model to handle the design of ducted propellers. Epps (2010) presented experimental data validating the off-design performance analysis feature for the propeller case, but his data showed that further development is required to accurately predict the off-design performance in the turbine case.

What follows is the theoretical foundation and numerical implementation of the OpenProp propeller/turbine design code suite. This text is taken from (Epps, 2010, ch. 7), and it draws from the theory presented in (Coney, 1989), (Kerwin, 2007), (Kerwin and Hadler, 2010), (Abbott and von Doenhoff, 1959), and (Carlton, 1994). In this document, all equations are given in dimensional terms, and their non-dimensionalized forms are given in table 1.

[1] B. P. Epps
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139
bepps@mit.edu

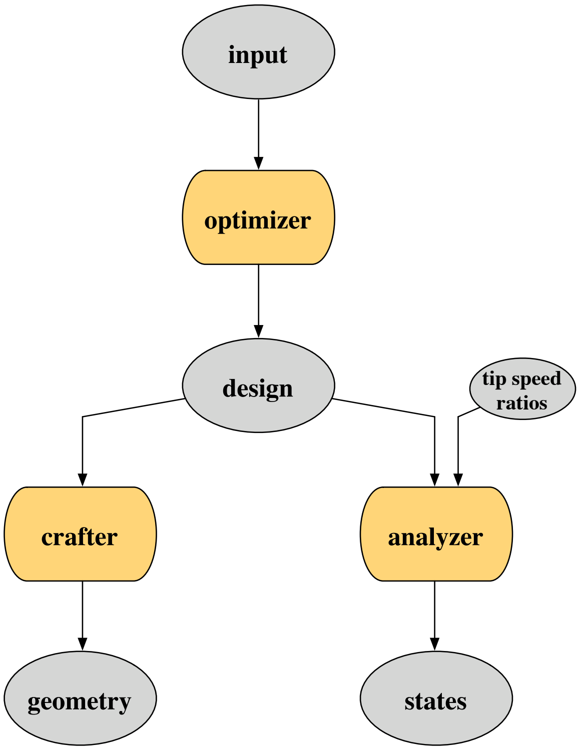


Figure 1: OpenProp information flow chart

OpenProp uses data structures to store the **input** parameters, **design**, **geometry**, and operating **states** of a propeller or turbine (bold indicates the names of the data structures). The data flow is illustrated in figure 1. For a parametric design study the **parinput** data (range of diameters, range of rotation rates, etc.) are defined by the user by running a short script. The graphical user interface (GUI) from OpenProp v1.0 was disabled in developing version 2.4, and a new GUI will be developed for future versions of the code suite. A Lerb's criterion design optimization routine is used to determine the optimum propeller design (and hence the best achievable efficiency) for each combination of input parameters, and these data are returned in the **paroutput** data structure. The user can then select the desired diameter, rotation rate, etc. for a detailed design and analysis of a single propeller.

For a single propeller design, the **input** data (diameter, rotation rate, etc.) are defined by the user by running a short script. The ***optimizer*** module determines the optimum propeller design, for the given inputs. The resulting propeller **design** can then be analyzed at off-design conditions (i.e. user-specified tip-speed ratios) in the ***analyzer*** to determine off-design operating **states**. The ***crafter*** can determine the 3D **geometry** and prepare rapid prototyping files for production of the propeller. What follows is a description of the propeller ***optimizer***, ***crafter***, and ***analyzer*** modules. The extension of OpenProp to the horizontal axis turbine case is also presented.

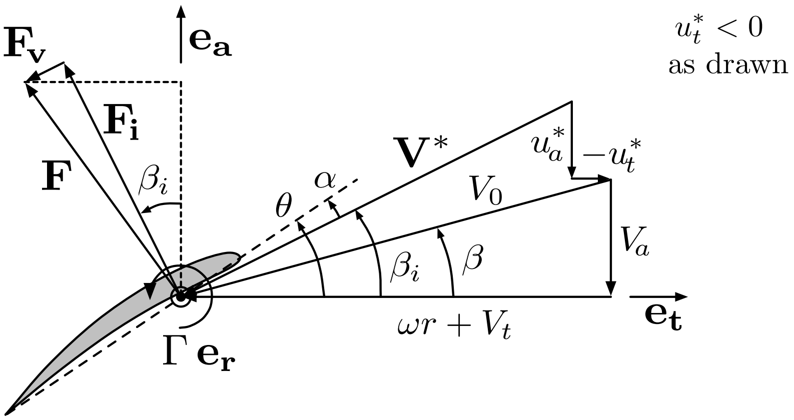


Figure 2: Propeller velocity/force diagram, as viewed from the tip towards the root of the blade. All velocities are relative to a stationary blade section at radius r.

2 Propeller lifting-line formulation

OpenProp is based on *moderately-loaded lifting line theory*, in which a propeller blade is represented by a lifting line, with trailing vorticity aligned to the local flow velocity (i.e. the vector sum of free-stream plus induced velocity). The induced velocities are computed using a vortex lattice, with helical trailing vortex filaments shed at discrete stations along the blade. The blade itself is modeled as discrete sections, having 2D section properties at each radius. Loads are computed by integrating the 2D section loads over the span of the blade.

The velocity/force diagram shown in figure 2 illustrates the velocities and forces (per unit span) on a 2D blade section in the axial $\mathbf{e_a}$ and tangential $\mathbf{e_t}$ directions. The propeller shaft rotates with angular velocity $\omega\,\mathbf{e_a}$, such that the apparent tangential (swirl) inflow at radius r is $-\omega r \mathbf{e_t}$. Also shown in figure 2 are the axial and tangential inflow velocities, $\mathbf{V_a} = -V_a\mathbf{e_a}$ and $\mathbf{V_t} = -V_t\mathbf{e_t}$; induced axial and tangential velocities, $\mathbf{u_a^*} = -u_a^*\mathbf{e_a}$ and $\mathbf{u_t^*} = -u_t^*\mathbf{e_t}$ (note that $u_t^* < 0$ during normal propeller operation, so $\mathbf{u_t^*}$ actually points in the $\mathbf{e_t}$ direction, as drawn); and the total resultant inflow velocity, $\mathbf{V}^*$, which has magnitude

$$V^* = \sqrt{(V_a + u_a^*)^2 + (\omega r + V_t + u_t^*)^2} \tag{2.1}$$

and is oriented at pitch angle,

$$\beta_i = \arctan\left(\frac{V_a + u_a^*}{\omega r + V_t + u_t^*}\right) \tag{2.2}$$

to the $\mathbf{e_t}$ axis. Also shown on figure 2 are the angle of attack, α; blade pitch angle $\theta = \alpha + \beta_i$; circulation, $\Gamma\,\mathbf{e_r}$; (inviscid) Kutta-Joukowski lift force, $\mathbf{F_i} = \rho\mathbf{V}^* \times (\Gamma\,\mathbf{e_r})$; and viscous drag force, $\mathbf{F_v}$, aligned with $\mathbf{V}^*$. Assuming the Z blades are identical, the total thrust and torque *on the* *propeller* are

$$\mathbf{T} = Z \int_{r_h}^{R} [F_i \cos\beta_i - F_v \sin\beta_i] dr \;\; (\hat{e}_a) \tag{2.3}$$

$$\mathbf{Q} = Z \int_{r_h}^{R} [F_i \sin\beta_i + F_v \cos\beta_i] r dr \;\; (-\hat{e}_a) \tag{2.4}$$

where $F_i = \rho V^* \Gamma$ and $F_v = \frac{1}{2}\rho (V^*)^2 C_D c$ are the magnitudes of the inviscid and viscous force per unit radius, ρ is the fluid density, C_D is the section drag coefficient, c is the section chord, and r_h and R are the radius of the hub and blade tip, respectively.

The power consumed by the propeller is the product of torque and angular velocity

$$P = Q\omega \tag{2.5}$$

where $P > 0$ indicates that power is being put into the fluid by the propeller (i.e. the torque resists the motion). The useful power produced by the propeller is TV_s where V_s is the ship speed (i.e. free-stream speed), so the efficiency of the propeller is

$$\eta = \frac{TV_s}{Q\omega} \tag{2.6}$$

2.1 Wake model

Following Kerwin (2007), OpenProp employs a standard propeller vortex lattice model to compute the axial and tangential induced velocities, $\{u_a^*, u_t^*\}$. In the vortex lattice formulation, a Z-bladed propeller is modeled as a single representative radial lifting line, partitioned into M panels. A horseshoe vortex filament with circulation $\Gamma(i)$ surrounds the i^{th} panel, consisting of helical trailing vortex filaments shed from the panel endpoints ($r_v(i)$ and $r_v(i+1)$) and the segment of the lifting line that spans the panel. The induced velocities are computed at *control points* on the lifting line at radial locations $r_c(m)$, $m = 1 \ldots M$, by summing the velocity induced by each horseshoe vortex

$$u_a^*(m) = \sum_{i=1}^{M} \Gamma(i)\, \bar{u}_a^*(m,i) \tag{2.7}$$

$$u_t^*(m) = \sum_{i=1}^{M} \Gamma(i)\, \bar{u}_t^*(m,i) \tag{2.8}$$

where $\bar{u}_a^*(m,i)$ and $\bar{u}_t^*(m,i)$ are the axial and tangential velocity induced at $r_c(m)$ by a unit-strength horseshoe vortex surrounding panel i. Since the lifting line itself does not contribute to the induced velocity,

$$\bar{u}^*_{a(m,i)} = \bar{u}_{a(m,i+1)} - \bar{u}_{a(m,i)} \tag{2.9}$$

$$\bar{u}^*_{t(m,i)} = \bar{u}_{t(m,i+1)} - \bar{u}_{t(m,i)} \tag{2.10}$$

where $\bar{u}_{a(m,i)}$ and $\bar{u}_{t(m,i)}$ are the axial and tangential velocities induced at $r_{c(m)}$ by a unit-strength constant-pitch constant-radius helical vortex shed from $r_{v(i)}$, with the circulation vector directed *downstream* (i.e. away from the lifting line) by right-hand rule. These are computed using the formulae by Wrench (1957):

For $r_{c(m)} < r_{v(i)}$:

$$\begin{aligned}\bar{u}_{a(m,i)} &= \frac{Z}{4\pi r_c}(y - 2Zyy_0F_1)\\ \bar{u}_{t(m,i)} &= \frac{Z^2}{2\pi r_c}(y_0F_1)\end{aligned}$$

For $r_{c(m)} > r_{v(i)}$:

$$\begin{aligned}\bar{u}_{a(m,i)} &= -\frac{Z^2}{2\pi r_c}(yy_0F_2)\\ \bar{u}_{t(m,i)} &= \frac{Z}{4\pi r_c}(1 + 2Zy_0F_2)\end{aligned}$$

where

$$\begin{aligned}F_1 &\approx \frac{-1}{2Zy_0}\left(\frac{1+y_0^2}{1+y^2}\right)^{\frac{1}{4}}\left\{\frac{U}{1-U} + \frac{1}{24Z}\left[\frac{9y_0^2+2}{(1+y_0^2)^{1.5}} + \frac{3y^2-2}{(1+y^2)^{1.5}}\right]\ln\left|1+\frac{U}{1-U}\right|\right\}\\ F_2 &\approx \frac{1}{2Zy_0}\left(\frac{1+y_0^2}{1+y^2}\right)^{\frac{1}{4}}\left\{\frac{1}{U-1} - \frac{1}{24Z}\left[\frac{9y_0^2+2}{(1+y_0^2)^{1.5}} + \frac{3y^2-2}{(1+y^2)^{1.5}}\right]\ln\left|1+\frac{1}{U-1}\right|\right\}\\ U &= \left(\frac{y_0\left(\sqrt{1+y^2}-1\right)}{y\left(\sqrt{1+y_0^2}-1\right)}\exp\left(\sqrt{1+y^2}-\sqrt{1+y_0^2}\right)\right)^Z\\ y &= \frac{r_c}{r_v\tan\beta_w}\\ y_0 &= \frac{1}{\tan\beta_w}\end{aligned}$$

and β_w is the pitch angle of the helical vortices in the wake. Consistent with *moderately-loaded lifting line theory*, we set $\beta_w = \beta_i$ in order to ‘align’ the wake with the local flow at the blade (Kerwin, 2007).

2.2 Hubs and ducts

A hub of radius r_h is modeled as an image vortex lattice. The image trailing vortex filaments have equal and opposite strength as the real trailing vortex filaments; they are stationed at radii

$$r_{im} = \frac{r_h^2}{r_v} \tag{2.11}$$

and have pitch angle

$$\tan[\beta_i^{im}] = \frac{r_v(1) \cdot \tan[\beta_i^v(1)]}{r_{im}} \tag{2.12}$$

The image vorticity is shed through the trailing surface of the hub and rolls up into a *hub vortex* of radius, r_o, and the drag due to the hub vortex is

$$\mathbf{D_h} = \frac{\rho Z^2}{16\pi}\left[\ln\left(\frac{r_h}{r_o}\right) + 3\right][\Gamma(1)]^2\,(-\mathbf{e_a}) \tag{2.13}$$

In OpenProp the default hub radius is $\frac{r_h}{r_o} = 0.5$.

The influence of a duct is also modeled using an image vortex lattice with equal and opposite strength of the real vortex lattice. For a duct radius r_d, the image radii are $r_{im} = \frac{r_d^2}{r_v}$, and the image pitch angles are defined by $\tan[\beta_i^{im}] = \frac{r_v(Mp) \cdot \tan[\beta_i^v(Mp)]}{r_{im}}$.

In addition, a duct endowed with circulation will induce axial velocity at the lifting line. The circulation about the duct Γ_d is represented by a series of N_d axisymmetric vortex rings of radius r_d equispaced along the duct at axial locations $x_d(n)$ for $n = 1 \ldots N_d$. The circulation distribution is taken to be that of a NACA a=0.8 foil, such that the n^{th} vortex ring has circulation $\Gamma_d \cdot \bar{\Gamma}_d(n)$, and the total circulation of the duct is $\Gamma_d = \Gamma_d \cdot \sum_{n=1}^{N_d} \bar{\Gamma}_d(n)$. Thus, $\bar{\Gamma}_d$ represents the circulation distribution for a unit-strength duct. The axial velocity induced at $r_c(m)$ by a unit-strength duct is

$$\bar{u}^*_{a,d}(m) = \sum_{n=1}^{N_d} \bar{\Gamma}_d(n)\bar{u}_{a,d}(m,n) \tag{2.14}$$

where $\bar{u}_{a,d}(m,n)$ is the axial velocity induced at at $(x = 0,\ r_c(m))$ by a unit-strength vortex ring at $x = x_d(n)$, which is given in terms of complete elliptic integrals in (Stubblefield, 2008, p. 22). Thus, the axial velocity induced at $r_c(m)$ by duct with circulation Γ_d is

$$u^*_{a,d}(m) = \Gamma_d \cdot \bar{u}^*_{a,d}(m) \tag{2.15}$$

In the case of a hub or duct (or both), the horseshoe influence functions $(\bar{u}^*_a,\ \bar{u}^*_t)$ are modified to include the influence of the image vortex lattices, and the axial induced velocity (2.7) is redefined to include the flow induced by the duct circulation

$$u_a^*(m) = \sum_{i=1}^{M} \Gamma(i)\, \bar{u}_a^*(m,i) \;\; + \;\; \Gamma_d\, \bar{u}_{a,d}^*(m) \tag{2.16}$$

$$u_t^*(m) = \sum_{i=1}^{M} \Gamma(i)\, \bar{u}_t^*(m,i) \tag{2.17}$$

Note that the tangential induced velocity (2.8) is unaffected by the presence of a constant-radius duct. This model breaks down in the case of a converging duct, since by conservation of angular momentum, the tangential (swirl) velocity must increase as the flow contracts. A more sophisticated model must be used to accurately model the flow through a converging-ducted propulsor.

The thrust produced by the duct can be computed in terms of the axial and radial *circumferential mean velocities* induced on the duct by the propeller, u_a^d and u_r^d, as follows

$$T_d = 2\pi r_d \cdot \sum_{n=1}^{N_d} \left(\rho \Big[- u_r^d(n) \Big] \Gamma_d\, \bar{\Gamma}_d(n) - \tfrac{1}{2}\rho \Big[V_a^d + u_a^d(n) \Big]^2 C_{D,d} \frac{c_d}{N_d} \right) \tag{2.18}$$

where $u_r^d(n)$ is the radial velocity induced at $(x_d(n), r_d)$ by the propeller, $C_{D,d}$ is the two-dimensional section drag coefficient, and c_d is the chord length of the duct. Note that this equation is written incorrectly in (Coney, 1989) and (Stubblefield, 2008). The circumferential mean velocities are also written incorrectly in (Coney, 1989) and (Stubblefield, 2008) but can be found in (Hough, 1964).

3 Propeller design optimization

The performance of a propeller can be computed given the circulation distribution, Γ, and flow parameters $\{V^*$, β_i, u_a^*, u_t^*, $\bar{u}_a^*$, $\bar{u}_t^*\}$. These all must be self-consistent for the state to be physically realistic. That is, equations {(2.1), (2.2), (2.16), (2.17), (2.9), (2.10)} must all hold, given Γ. Thus, propeller design optimization reduces to finding the optimum circulation distribution.

Following Coney (1989), the propeller optimization problem is to find the set of M circulations of the vortex lattice panels that produce the least torque

$$Q = \rho Z \sum_{m=1}^{M} \left\{ [V_a + u_a^*]\Gamma + \tfrac{1}{2} V^* C_D c [\omega r_c + V_t + u_t^*] \right\} r_c \triangle r_v \tag{3.1}$$

for a specified thrust, T_s,

$$\begin{aligned} T = \rho Z \sum_{m=1}^{M} & \left\{ [\omega r_c + V_t + u_t^*]\Gamma - \tfrac{1}{2} V^* C_D c [V_a + u_a^*] \right\} \triangle r_v \\ & - \text{Hflag} \cdot \tfrac{\rho Z^2}{16\pi} \left[\ln\left(\tfrac{r_h}{r_o}\right) + 3 \right] [\Gamma(1)]^2 = T_s \end{aligned} \tag{3.2}$$

where Hflag is set to 1 to model a hub or 0 for no hub. Here, $\{\rho, Z, \omega\}$ are constants and $\{\Gamma$, u_a^*, u_t^*, V^*, c, V_a, V_t, C_D, r_c, $\triangle r_v\}$ are evaluated at $r_c(m)$ in the summation. In the case of a ducted propeller optimization, the propeller only provides a portion of the total required thrust, T_r. The *thrust ratio* is defined as

$$\tau = \frac{T_s}{T_r} = \frac{\text{propeller thrust}}{\text{total thrust}} \tag{3.3}$$

such that the thrust required of the duct is $T_d = (1 - \tau)T_r$ and the total thrust is $T_r = T_s + T_d$. In the case of no duct, $\tau = 1$, $T_d = 0$, and $T_s = T_r$.

To solve this optimization problem, Coney (1989) employs the method of the Lagrange multiplier from variational calculus. He forms an auxiliary function, $H = Q + \lambda_1(T - T_s)$, where λ_1 is the unknown Lagrange multiplier that introduces the thrust constraint (3.2). Clearly, if $T = T_s$, then a minimum H coincides with a minimum Q. To find this minimum, the derivatives with respect to the unknowns are set to zero

$$\frac{\partial H}{\partial \Gamma(i)} = 0 \quad \text{for} \quad i = 1 \dots M \tag{3.4}$$

$$\frac{\partial H}{\partial \lambda_1} = 0 \tag{3.5}$$

which is a system of $M+1$ equations for as many unknowns $\{\Gamma(i{=}1 \dots M),\ \lambda_1\}$. This non-linear system of equations is solved iteratively until convergence of the optimized circulation distribution, Γ, and flow parameters $\{V^*,\ \beta_i,\ u_a^*,\ u_t^*,\ \bar{u}_a^*,\ \bar{u}_t^*\}$.

The section chord length, c, can also be optimized . Equation (3.1) shows that minimizing c minimizes the parasitic torque due to viscous drag. However, since c is related to the loading by $C_L = \frac{\Gamma}{\frac{1}{2}(V^*)c}$, where C_L is the section lift coefficient, the chord cannot be made arbitrarily small. If a maximum allowable lift coefficient is chosen, (typically, $0.1 < C_{L_{\max}} < 0.5$), then the “optimum" chord is

$$\hat{c} = \frac{|\Gamma|}{\frac{1}{2}(V^*)C_{L_{\max}}} \tag{3.6}$$

In OpenProp v2.4, the *expanded area ratio* EAR can also be specified to be EAR_{spec}. Expanded area ratio is defined as

$$EAR = \mathcal{Z}\int_{r_h}^{R} \hat{c}(r)\, dr \tag{3.7}$$

If the specification is given, then the chord length distribution is scaled as follows

$$c(r) = \hat{c}(r)\frac{EAR_{spec}}{EAR} \tag{3.8}$$

Otherwise, the chord length distribution is set from equation (3.6), as $c(r) = \hat{c}(r)$.

Turning our attention back to equations (3.4) and (3.5), we can now evaluate the required partial derivatives of $\Gamma, \lambda_1, u_a^*, u_t^*,$ and V^* with respect to $\Gamma(i)$ and λ_1:

$$\frac{\partial\Gamma(m)}{\partial\Gamma(i)} = \begin{cases} 0 & {\scriptstyle (m\neq i)} \\ 1 & {\scriptstyle (m=i)} \end{cases} \quad , \quad \frac{\partial\lambda_1}{\partial\lambda_1} = 1 \tag{3.9}$$

$$\frac{\partial u_a^*(m)}{\partial\Gamma(i)} = \bar{u}_a^*(m,i) \quad , \quad \frac{\partial u_t^*(m)}{\partial\Gamma(i)} = \bar{u}_t^*(m,i) \tag{3.10}$$

$$\begin{aligned} \frac{\partial V^*(m)}{\partial\Gamma(i)} &= \tfrac{1}{2}(V^*)^{-1}\left(\begin{array}{l} 2(V_a + u_a^*)\frac{\partial u_a^*(m)}{\partial\Gamma(i)} + \\ 2(\omega r_c + V_t + u_t^*)\frac{\partial u_t^*(m)}{\partial\Gamma(i)} \end{array}\right) \\ &= \sin(\beta_i(m))\ \bar{u}_a^*(m,i) + \cos(\beta_i(m))\ \bar{u}_t^*(m,i) \end{aligned} \tag{3.11}$$

All other partial derivatives are zero or are ignored. In OpenProp v2.4, since expanded area ratio may be specified, chord length is no longer directly related to circulation, so the partial derivative $\frac{\partial c(m)}{\partial\Gamma(i)}$ is ignored.

The system of equations $\{(3.4), (3.5)\}$ is non-linear, so the following iterative approach is used to solve them. During each solution iteration, flow parameters $\left\{u_a^*, u_t^*, \bar{u}_a^*, \bar{u}_t^*, V^*, \frac{\partial V^*}{\partial \Gamma}, \lambda_1\right\}$ are frozen in order to linearize $\{(3.4), (3.5)\}$. The linear system of equations, with the linearized unknowns marked as $\{\breve{\Gamma}, \breve{\lambda}_1\}$, is as follows

$$\begin{aligned}
\frac{\partial H}{\partial \Gamma(i)} = \quad & \rho Z \sum_{m=1}^{M} \breve{\Gamma}(m) \cdot [\bar{u}_a^*(m,i) r_c(m) \triangle r_v(m) + \bar{u}_a^*(i,m) r_c(i) \triangle r_v(i)] \\
& + \rho Z V_a(i) r_c(i) \triangle r_v(i) \\
& + \rho Z \sum_{m=1}^{M} \tfrac{1}{2} C_D \frac{\partial V^*(m)}{\partial \Gamma(i)} c(m) [\omega r_c(m) + V_t(m) + u_t^*(m)] r_c(m) \triangle r_v(m) \\
& + \rho Z \sum_{m=1}^{M} \tfrac{1}{2} C_D V^*(m) c(m) [\bar{u}_t^*(m,i)] r_c(m) \triangle r_v(m) \\
& + \rho Z \lambda_1 \sum_{m=1}^{M} \breve{\Gamma}(m) \cdot [\bar{u}_t^*(m,i) \triangle r_v(m) + \bar{u}_t^*(i,m) \triangle r_v(i)] \\
& + \rho Z \breve{\lambda}_1 [\omega r_c(i) + V_t(i)] \triangle r_v(i) \\
& - \rho Z \breve{\lambda}_1 \sum_{m=1}^{M} \tfrac{1}{2} C_D \frac{\partial V^*(m)}{\partial \Gamma(i)} c(m) \big[V_a(m) + u_a^*(m) \big] \triangle r_v \\
& - \rho Z \breve{\lambda}_1 \sum_{m=1}^{M} \tfrac{1}{2} C_D V^*(m) c(m) [\bar{u}_a^*(m,i)] \triangle r_v \\
& - \text{Hflag} \cdot \frac{\partial \Gamma(1)}{\partial \Gamma(i)} \cdot \lambda_1 \frac{\rho Z^2}{8\pi} \left[\ln \left(\frac{r_h}{r_o} \right) + 3 \right] \breve{\Gamma}(1) \\
= 0 \ \text{ for } \ & i = 1 \ldots M
\end{aligned} \tag{3.12}$$

$$\begin{aligned}
\frac{\partial H}{\partial \lambda_1} = \quad & \rho Z \sum_{m=1}^{M} \breve{\Gamma}(m) \cdot [\omega\, r_c(m) + V_t(m) + u_t^*(m)] \triangle r_v(m) \\
& - \rho Z \sum_{m=1}^{M} \tfrac{1}{2} C_D V^*(m) c(m) [V_a(m) + u_a^*(m)] \triangle r_v(m) \\
& - \text{Hflag} \cdot \frac{\rho Z^2}{16\pi} \left[\ln \left(\frac{r_h}{r_o} \right) + 3 \right] \Gamma(1) \cdot \breve{\Gamma}(1) \\
& - T_s \\
= 0 &
\end{aligned} \tag{3.13}$$

The system $\{(3.12),\ (3.13)\}$ is solved for the now linear $\{\breve{\Gamma}, \breve{\lambda}_1\}$, the circulation and Lagrange multiplier are updated ($\Gamma = \breve{\Gamma}$, $\lambda_1 = \breve{\lambda}_1$), and the new Γ is used to update the flow parameters. Coney (1989) describes a 'wake alignment procedure' for updating the flow parameters, whereby he iteratively updates: (1) the induced velocities $\{u_a^*, u_t^*\}$ via $\{(2.16),\ (2.17)\}$; (2) the inflow angle β_i via (2.2); and (3) the horseshoe influence functions $\{\bar{u}_a^*, \bar{u}_t^*\}$ via $\{(2.9),\ (2.10)\}$, and iteration continues until convergence of these flow parameters. Given the now-aligned wake, he then updates the remaining flow parameters $\left\{V^*, \frac{\partial V^*}{\partial \Gamma}, c\right\}$ and continues the main iterative loop, finding the next guess for Γ. This wake alignment procedure is time-consuming and tenuous, because it is prone to crash if the induced velocities do not vary smoothly over the span.

A slightly different optimization procedure was implemented by Epps et al (2009b) in OpenProp. This implementation still solves $\{(3.12),\ (3.13)\}$ for a guess for $\{\breve{\Gamma}, \breve{\lambda}_1\}$, updates the circulation and Lagrange multiplier ($\Gamma = \breve{\Gamma}$, $\lambda_1 = \breve{\lambda}_1$), and uses the new Γ to update the flow parameters. However, in this procedure, the wake is not iteratively "aligned"; instead, one new guess is made for the wake flow parameters (in particular β_i), and the main iterative loop continues to find the next guess for Γ. Therefore, each iteration of the main loop involves updating Γ via $\{(3.12),\ (3.13)\}$. The critical step in the procedure is that $\{u_a^*, u_t^*\}$ are updated via $\{(2.16),\ (2.17)\}$ and then "repaired" by smoothing the velocities at the blade root and tip. This minor smoothing is critical to enable the entire system of equations to converge, because the alignment of the wake and the horseshoe influence functions which are fed into the next solution iteration are very sensitive to irregularities in the induced velocities. This smoothing is reasonable in the vortex-lattice model, since it introduces no more error than ignoring hub or tip vortex roll-up, or other flow features. Using these smooth induced velocities, the remaining flow parameters $\left\{\beta_i, \bar{u}_a^*, \bar{u}_t^*, V^*, \frac{\partial V^*}{\partial \Gamma}, c\right\}$ are updated via $\{(2.2),\ (2.9),\ (2.10),\ (2.1),\ (3.11),\ (3.6)\}$. This process is repeated until convergence of all of the flow parameters, yielding an optimized circulation distribution and a physically-realistic design operating state. Initial values of $\left\{\beta_i, V^*, \frac{\partial V^*}{\partial \Gamma}\right\}$ are computed with $\{u_a^*, u_t^*\} = 0$. The Lagrange multiplier is initialized at $\lambda_1 = -R$, and the section chord lengths at $c \approx 0$. If chord-length optimization is not desired, then the chord is set to the input value during the optimization process.

4 Propeller geometry

Once the design operating state of the propeller/turbine is known, the geometry can be determined to give such performance. The 3D geometry is built from given 2D section profiles that are scaled and rotated according to the design lift coefficient, chord length, and inflow angle $\{C_{L_0}, c, \beta_{i_0}\}$.

A given 2D section profile includes camber and thickness normalized by the chord, $\{\widetilde{f}/c, \widetilde{t}/c\}$, ideal angle of attack, $\widetilde{\alpha}_I$, and ideal lift coefficient, $\widetilde{C}_{L_I}$. Note that $\{\widetilde{f}, \widetilde{\alpha}_I, \widetilde{C}_{L_I}\}$ scale linearly with the maximum camber, $\widetilde{f}_0$ (Abbott and von Doenhoff, 1959). The section lift coefficient is given in terms of the geometry by $C_L = 2\pi(\alpha - \alpha_I) + C_{L_I}$ for $|\alpha - \alpha_I| \ll |\alpha - \alpha_I|_{\text{stall}}$, and the stall model is described in Section 5. In the geometry module, the angle of attack of each blade section is set to the ideal angle of attack ($\alpha = \alpha_I$) to prevent leading edge flow separation and/or cavitation. The lift coefficient then becomes the ideal lift coefficient ($C_L = C_{L_I}$). In order to achieve the desired lift coefficient, C_{L_0}, the given $\widetilde{C}_{L_I}$ is scaled by scaling the section camber. Thus, the desired lift coefficient and section geometry is

$$\{C_L, f_0, f, \alpha_I\} = \frac{C_{L_0}}{\widetilde{C}_{L_I}} \cdot \{\widetilde{C}_{L_I}, \widetilde{f}_0, \widetilde{f}, \widetilde{\alpha}_I\} \tag{4.1}$$

The pitch angle of the blade section is then fixed at

$$\theta = \alpha_I + \beta_{i_0} \tag{4.2}$$

With this computed blade 2D section geometry, OpenProp can then form the full 3D propeller geometry and export files for rapid prototyping of physical parts.

5 Off-design performance analysis

This section details the analysis of a propeller operating at an off-design (OD) *advance coefficient*

$$J_{s,OD} = \frac{V_s}{n_{OD} D} = \frac{\pi V_s}{\omega_{OD} R} \tag{5.1}$$

Following Epps (2010), an off-design operating state is defined by the rotation rate, ω_{OD}, and unknown flow parameters $\{V^*, \alpha, C_L, \Gamma, u_a^*, u_t^*, \beta_i, \bar{u}_a^*, \bar{u}_t^*, \Gamma_d\}$, which all must be self-consistent for the state to be physically realistic. To proceed, we need equations for the angle of attack, α, lift coefficient, C_L, circulation, Γ, and duct circulation Γ_d. In the analyzer, the pitch angle, θ, of each blade section is fixed, so the net angle of attack is

$$\alpha - \alpha_I = \beta_{i_0} - \beta_i \tag{5.2}$$

The circulation is computed from the 2D lift coefficient, which is given in terms of the loading by

$$C_L = \frac{2\Gamma}{V^* c} \tag{5.3}$$

The 2D section lift and drag coefficients are shown in figure 3 and given in closed form by equations

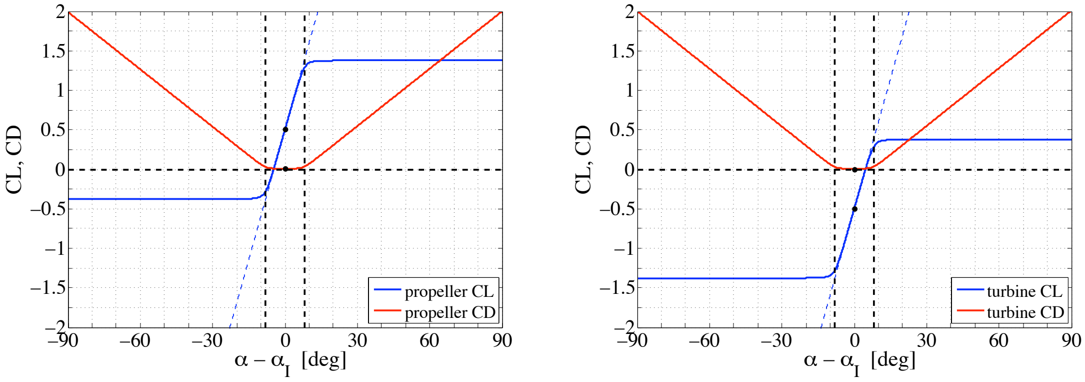
$$\begin{aligned} C_L = C_{L,0} &+ \tfrac{dC_L}{d\alpha}\Delta\alpha \\ &- \tfrac{dC_L}{d\alpha}(\Delta\alpha - \Delta\alpha_{\text{stall}}) \cdot F(\Delta\alpha - \Delta\alpha_{\text{stall}}) \\ &+ \tfrac{dC_L}{d\alpha}(-\Delta\alpha - \Delta\alpha_{\text{stall}}) \cdot F(-\Delta\alpha - \Delta\alpha_{\text{stall}}) \end{aligned} \tag{5.4}$$

$$\begin{aligned} C_D = C_{D,0} & \\ &+ A \cdot (\Delta\alpha - \Delta\alpha_{\text{stall}}) \cdot F(\Delta\alpha - \Delta\alpha_{\text{stall}}) \\ &+ A \cdot (-\Delta\alpha - \Delta\alpha_{\text{stall}}) \cdot F(-\Delta\alpha - \Delta\alpha_{\text{stall}}) \\ &- 2A \cdot (-\Delta\alpha_{\text{stall}}) \cdot F(-\Delta\alpha_{\text{stall}}) \end{aligned} \tag{5.5}$$

where the auxiliary function $F(x) = \frac{\arctan(Bx)}{\pi} + \frac{1}{2}$ has limits $F(x \to -\infty) \to 0$ and $F(x \to \infty) \to 1$. Here: $\Delta\alpha = \alpha - \alpha_I$ [rad]; $\Delta\alpha_{\text{stall}} = 8\frac{\pi}{180}$ [rad] is the default OpenProp stall angle; $B = 20$ is the default OpenProp stall sharpness parameter; $A = \frac{2 - C_{D,0}}{\frac{\pi}{2} - \Delta\alpha_{\text{stall}}}$ is drag coefficient post-stall slope; and $\frac{dC_L}{d\alpha} = 2\pi$ is default OpenProp lift curve slope, which is consistent with linear foil theory. These values are used in all calculations unless specifically noted otherwise. Thus $C_L \approx C_{L,0} + 2\pi(\alpha - \alpha_I)$ before stall and approximately constant post stall. The drag coefficient is approximately constant until stall and then rises to the canonical value of 2 when the inflow is normal to the blade. This type of model is used in ASWING (Drela, 1999). Equations (5.4) and (5.5) offer the flexibility to change the stall angle, lift curve slope, and drag coefficient to more accurately model foil sections of moderate thickness to chord ratios.

If a duct is present, a system of equations analogous to (5.2), (5.4), (5.3) must also hold true for the duct circulation, as will be discussed.

The operating states of a propeller or turbine for each given ω_{OD} are computed as follows. An operating state is defined by ω_{OD} and unknown flow parameters $\{V^*, \alpha, C_L, \Gamma, u_a^*, u_t^*, \beta_i, \bar{u}_a^*, \bar{u}_t^*, \Gamma_d\}$, which all must be self-consistent for the state to be physically-realistic. That is, equations $\{$(2.1), (5.2), (5.4), (5.3), (2.16), (2.17), (2.2), (2.9), (2.10), (5.10)$\}$ must all hold, given ω_{OD}. Since



(a) (b)

Figure 3: Lift coefficient, C_L, and drag coefficient, C_D, versus net angle of attack, $\alpha - \alpha_I$, for the (a) propeller and (b) turbine cases, with $\frac{dC_L}{d\alpha} = 2\pi$ and on-design specifications $C_{L_0} = 0.5$ and $C_{D_0} = 0.05$. The vertical dashed lines at $|\alpha - \alpha_I|_{\text{stall}} = \pm 8$ deg indicate the stall angle of attack.

there are M vortex panels, there are $7M + 2M^2 + 1$ unknowns and a system of $7M + 2M^2 + 1$ non-linear equations that govern the state of the system. This system is solved in OpenProp using an approach similar to a Newton solver.

Since the $7M + 2M^2 + 1$ equations are coupled through the parameters $\{\beta_i, \bar{u}_a^*, \bar{u}_t^*, \Gamma_d\}$, they can be decoupled by considering two state vectors: $\mathbf{X} = \{V^*, \alpha, C_L, \Gamma, u_a^*, u_t^*\}^\top$ and $\mathbf{Y} = \{\beta_i, \bar{u}_a^*, \bar{u}_t^*, \Gamma_d\}$. During each solution iteration, state vector $\mathbf{X}$ is updated, and then $\mathbf{Y}$ is updated; this process repeats until convergence of the entire system.

Consider state vector $\mathbf{X}$: It consists of M sets of 6 state variables, one set per vortex panel. The 6 variables for each vortex panel are coupled to one another, but not to the other variables in $\mathbf{X}$. Thus, $\mathbf{X}$ can be partitioned into M state vectors, $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}^\top$, where $\mathbf{x}_m = \{V^*, \alpha, C_L, \Gamma, u_a^*, u_t^*\}^\top$ with each variable evaluated at $r_{c(m)}$. Each of these state vectors can be updated independently.

Each vortex panel state vector, $\mathbf{x}_m$, is updated using a Newton solver. Define the residual vector for the m^{th} panel as

$$\mathbf{R}_m = \begin{bmatrix} V^* & - & \sqrt{(V_a + u_a^*)^2 + (\omega_{OD} r_c + V_t + u_t^*)^2} \\ \alpha & - & (\alpha_I + \beta_{i0} - \beta_i) \\ C_L & - & C_L(\alpha) \\ \Gamma & - & (\frac{1}{2} C_L V^* c) \\ u_a^* & - & \Big([\bar{u}_a^*] \cdot [\Gamma] + \Gamma_d \bar{u}_{a,d}^*\Big) \\ u_t^* & - & \Big([\bar{u}_t^*] \cdot [\Gamma]\Big) \end{bmatrix} \tag{5.6}$$

where each variable is evaluated at $r_{c(m)}$. In order to drive the residuals to zero, the desired change in the state vector, $\mathbf{dx}_m$, is found by solving the matrix equation

$$0 = \mathbf{R}_m + \mathbf{J}_m \cdot \mathbf{dx}_m$$

where non-zero the elements of the Jacobian matrix, $\mathbf{J}_{m(i,j)} = \frac{\partial \mathbf{R}_m(i)}{\partial \mathbf{x}_m(j)}$, are

$$\begin{aligned}
\mathbf{J}_{m(i,i)} &= \frac{\partial R_{V^*}}{\partial V^*} = \frac{\partial R_\alpha}{\partial \alpha} = \frac{\partial R_{C_L}}{\partial C_L} = \frac{\partial R_\Gamma}{\partial \Gamma} = \frac{\partial R_{u_a^*}}{\partial u_a^*} = \frac{\partial R_{u_t^*}}{\partial u_t^*} = 1 \quad (i = 1 \dots 6) \\
\mathbf{J}_{m(1,5)} &= \frac{\partial R_{V^*}}{\partial u_a^*} = -\frac{V_a + u_a^*}{\sqrt{(V_a + u_a^*)^2 + (\omega_{OD} r_c + V_t + u_t^*)^2}} \\
\mathbf{J}_{m(1,6)} &= \frac{\partial R_{V^*}}{\partial u_t^*} = -\frac{\omega_{OD} r + V_t + u_t^*}{\sqrt{(V_a + u_a^*)^2 + (\omega_{OD} r_c + V_t + u_t^*)^2}} \\
\mathbf{J}_{m(2,5)} &= \frac{\partial R_\alpha}{\partial u_a^*} = \frac{\partial R_\alpha}{\partial \beta_i} \cdot \frac{\partial \beta_i}{\partial \tan(\beta_i)} \cdot \frac{\partial \tan(\beta_i)}{\partial u_a^*} = \frac{1}{1 + \tan^2(\beta_i)} \cdot \frac{1}{\omega_{OD} r_c + V_t + u_t^*} \\
\mathbf{J}_{m(2,6)} &= \frac{\partial R_\alpha}{\partial u_t^*} = \frac{\partial R_\alpha}{\partial \beta_i} \cdot \frac{\partial \beta_i}{\partial \tan(\beta_i)} \cdot \frac{\partial \tan(\beta_i)}{\partial u_t^*} = \frac{1}{1 + \tan^2(\beta_i)} \cdot \frac{-\tan(\beta_i)}{\omega_{OD} r_c + V_t + u_t^*} \\
\mathbf{J}_{m(3,2)} &= \frac{\partial R_{C_L}}{\partial \alpha} = -\frac{dC_L(\alpha)}{d\alpha} \\
\mathbf{J}_{m(4,1)} &= \frac{\partial R_\Gamma}{\partial V^*} = -\tfrac{1}{2} C_L c \\
\mathbf{J}_{m(4,3)} &= \frac{\partial R_\Gamma}{\partial C_L} = -\tfrac{1}{2} V^* c \\
\mathbf{J}_{m(5,4)} &= \frac{\partial R_{u_a^*}}{\partial \Gamma} = -\bar{u}^*_{a(m,m)} \\
\mathbf{J}_{m(6,4)} &= \frac{\partial R_{u_t^*}}{\partial \Gamma} = -\bar{u}^*_{t(m,m)} \\
\mathbf{J}_{m(5,2)} &= \frac{\partial R_{u_a^*}}{\partial \alpha} = \frac{\partial R_{u_a^*}}{\partial \beta_i} \cdot \frac{\partial \beta_i}{\partial \alpha} = \sum_{j=1}^{M} \Gamma_{(j)} \frac{\partial \bar{u}^*_{a(m,j)}}{\partial \beta_{i(m)}} \\
\mathbf{J}_{m(6,2)} &= \frac{\partial R_{u_t^*}}{\partial \alpha} = \frac{\partial R_{u_t^*}}{\partial \beta_i} \cdot \frac{\partial \beta_i}{\partial \alpha} = \sum_{j=1}^{M} \Gamma_{(j)} \frac{\partial \bar{u}^*_{t(m,j)}}{\partial \beta_{i(m)}}
\end{aligned}$$

where the flow parameters are evaluated at $r_{c(m)}$ unless explicitly stated. All other terms are zero or are ignored.

The state vector for the next iteration, then, is $\mathbf{x}_m^{\text{next}} = \mathbf{x}_m^{\text{current}} + \mathbf{dx}_m$. By solving one Newton iteration for each of the $m = 1, \dots, M$ vortex panels, state vector $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_M\}^\top$ is updated.

Given the new $\mathbf{X}$ values, $\mathbf{Y}$ is updated: β_i is updated via (2.2), and then $\{\bar{u}_a^*, \bar{u}_t^*\}$ are updated via $\{$(2.9), (2.10)$\}$. In the next solution iteration, these new values of $\mathbf{Y}$ are used to update $\mathbf{X}$, and so on. Since the solution scheme updates both $\mathbf{X}$ and $\mathbf{Y}$ in each iteration, it accounts for the coupled interaction between all $7M + 2M^2 + 1$ unknown flow parameters and converges on a physically-realistic operating state of the system.

For off-design analysis of a ducted propeller, the duct circulation Γ_d is unknown and is estimated in a manner similar to that used to update the propeller blade circulation. After each iteration of the Newton solver, the circumferential mean velocities are evaluated at the quarter chord position of the duct. Since the propeller plane ($x = 0$) is located at the midchord of the duct, the quarter chord is at $x = -c_d/4$. Thus, the desired velocities are $\tilde{u}_a^d = u_a^d(x = -c_d/4, r = r_d)$ and $\tilde{u}_r^d = u_r^d(x = -c_d/4, r = r_d)$. The total inflow speed at the duct quarter chord is

$$\tilde{V}^* = \sqrt{(V_a^d + \tilde{u}_a^d)^2 + (\tilde{u}_r^d)^2} \tag{5.7}$$

The inflow angle at the quarter chord is

$$\tilde{\beta}_i = \arctan\left(\frac{-\tilde{u}_r^d}{V_a^d + \tilde{u}_a^d}\right) \tag{5.8}$$

The 2D lift coefficient (i.e. lift per unit circumference) for the duct becomes

$$C_{L,d} = \tfrac{dC_L}{d\alpha} \cdot (\tilde{\beta}_{i_0} - \tilde{\beta}_i) + C_{L,d0} \tag{5.9}$$

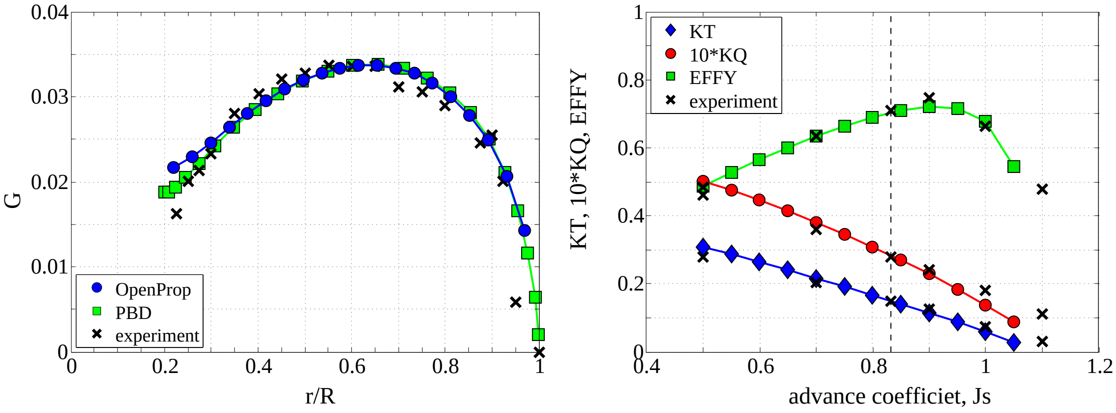
where $\tilde{\beta}_{i,0}$ and $C_{L,d0}$ are the inflow angle and lift coefficient on design, respectively, and $\frac{dC_L}{d\alpha} = 2\pi$ is the lift curve slope. The duct circulation then is

$$\Gamma_d = \tfrac{1}{2}\tilde{V}^* c_d C_{L,d} \tag{5.10}$$

Since all of these equations can be computed explicitly from any given guess of state vector $\mathbf{X}$, this procedure to estimate Γ_d can be done between Newton solver iterations. Given the new guess for Γ_d, the axial induced velocity at the propeller u_a^* will change, as shown in residual equation (5.6). Since the new value of Γ_d is used in the next Newton solver iteration, the coupled interaction between Γ_d and the remaining unknown flow parameters is accounted for.

The system is said to converge when all $6M$ elements of $\mathbf{X}$ have converged. Since β_i is directly related to α and $\bar{u}_a^*$ and $\bar{u}_t^*$ are functions of β_i, once α converges, this implies that $\mathbf{Y}$ has converged as well. For each operating state, the analyzer computes the propeller/turbine thrust, torque, and power coefficients and efficiency.

The OpenProp analyzer was validated with *U.S. Navy propeller 4119.* Figure 4a shows the circulation distribution of an OpenProp-designed version of 4119, showing good agreement with U.S. Navy code PBD and experimental data from (Black, 1997). Figure 4b also shows good agreement between the off-design performance curve predicted by OpenProp and experimental data from (Black, 1997), thus validating the performance analysis method presented herein.



(a)

(b)

Figure 4: (a) Design circulation distribution for U.S. Navy propeller 4119, and (b) off-design performance curves for propeller 4119. OpenProp results agree with PBD code solution and experimental data from (Black, 1997).

The performance analyzer also enables cavitation analysis, which requires the blade surface pressure distribution. The pressure distribution is computed in OpenProp using either of two 2D foil solvers that require the lift coefficient distribution. The lift coefficient distribution is found in the analyzer for each off-design operating state. Peterson (2008) developed a cavitation analysis module using the open-source code XFOIL (Drela, 1989). Chung (2007) implemented a simpler 2D vortex lattice code that can also be used as the 2D foil solver engine. The pressure distribution results are incorporated in a module that generates Brockett diagrams for a given blade design and off-design operating state. Using the Brockett diagram, the thickness ratio can be chosen to give adequate on-design cavitation margin and off-design angle of attack margin. The 2D solvers can also be used to analyze the blade pressure coefficient distributions for determining cavitation margin and location by comparing the pressure coefficient to the local cavitation number of the section.

5.1 Example propeller off-design analysis

As an illustrative example of off-design performance analysis, I now replicate the propellers designed in Coney (1989, p. 28-31). In this exercise, several propellers are designed to give the same thrust coefficient, $C_T = 0.512$, for a range of design advance coefficients

$$J_s = \frac{V_s}{nD} = \frac{\pi V_s}{\omega R}$$

Each is a hubless, five-bladed propeller with a diameter $D = 1$ m, hub diameter $D_{\text{hub}} = 0.2$ m, and ship speed $V_s = 1$ m/s. The chord lengths are optimized for each propeller, with $C_{L,\max} = 0.2$, and

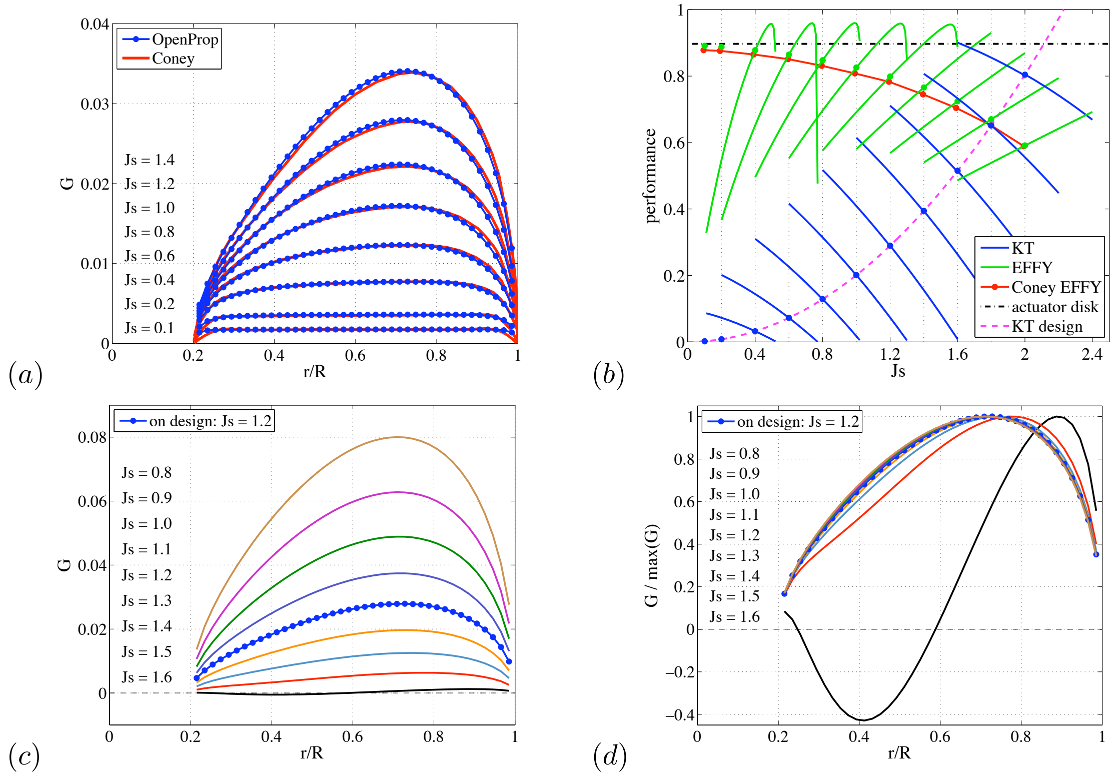


Figure 5: Example 5-bladed propellers from Coney (1989): (a) on-design circulation distributions for propellers designed for selected advance coefficients, (b) off-design performance of these propellers, (c) off-design circulation distributions for the propeller with on-design advance coefficient $J_s = 1.2$: (d) off-design circulation distributions each normalized by its maximum value.

viscous effects are ignored.

Each of the circulation distributions in figure 5a were optimized to give the same thrust, for the prescribed advance coefficient. The distributions I computed using OpenProp agree well with those computed by Coney; minor disagreement is expected, since Coney did not align the wake to the local flow (i.e. he computed the wake influence functions with the wake aligned to the undisturbed flow, which is acceptable for lightly-loaded propellers such as these). The on-design efficiencies computed by Coney also agree well with those I calculate, as shown in figure 5b.

Using my off-design performance analyzer, I computed the performance of each of these propellers for a range of advance coefficients. I show in figure 5b the off-design efficiency, EFFY, and thrust coefficient, KT, of all these propellers; the torque coefficient is omitted for clarity. The black dash-dotted line represents the efficiency of an *actuator disc* producing a thrust coefficient of $C_T = 0.512$, which is

$$EFFY = \frac{2}{1+\sqrt{1+C_T}} = 0.8970 \tag{5.11}$$

Propellers designed for advance coefficients approaching zero approach the actuator disk efficiency, since the rotation rate approaches infinity in this limit, and the blades lose their identity and 'become' the actuator disk. The magenta dashed line represents the required thrust coefficient, for a given advance ratio, since

$$K_T = \frac{\pi}{8} C_T \cdot J_s^2 \tag{5.12}$$

by definition. Each of the propellers meets this thrust requirement on-design, and they produce larger K_T for smaller off-design J_s, and visa versa. This change in loading is due to the change in net angle of attack: Referring to the propeller velocity/force diagram, figure 2, recall the raw inflow angle is defined as

$$\tan\beta = \frac{V_s}{\omega r} = \frac{J_s}{\pi} \cdot \frac{R}{r} \tag{5.13}$$

For small off-design J_s, the apparent tangential inflow due to propeller rotation is larger, and the inflow angle is smaller. This corresponds to an increased angle of attack (since the blade pitch is fixed) and, therefore, increased loading. Figure 5c shows the load distribution for several off-design advance coefficients for the propeller designed for $J_s = 1.2$; loading increases as J_s decreases, and visa versa. For larger off-design J_s, this corresponds to larger inflow angles, reduced angle of attack, and reduced loading.

Consideration of equation (5.13) reveals that a change in advance coefficient should affect the innermost blade sections more than the outermost sections, since $\frac{R}{r}$ increases with decreasing radial position r. Thus, the off-design circulation distribution should shift inwards for lower advance coefficients (i.e. higher rotation rates) and shift outwards for higher advance coefficients. This is demonstrated in figure 5, which shows the off-design circulation distributions, each normalized by its maximum value. This effect is minimal for low advance coefficients but is quite dramatic for higher advance coefficients, which tend to unload the root more than the tip, shifting the circulation distribution outboard.

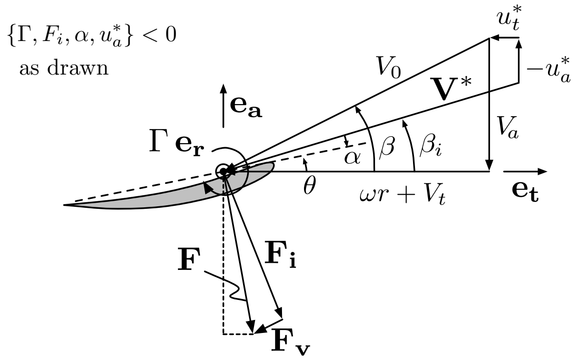


Figure 6: Turbine velocity/force diagram, as viewed from the tip towards the root of the blade. All velocities are relative to a stationary blade section at radius r.

6 Turbine lifting-line representation

This section demonstrates that a turbine can be represented in the propeller lifting line formulation of Section 2 simply by allowing a negative circulation, $\Gamma < 0$, and other associated sign changes. If $\Gamma < 0$, then $\{F_i = \rho V^* \Gamma, C_L, u_a^*, u_t^*, f_0, \alpha\} < 0$ as well, via equations $\{(5.3), (2.16), (2.17), (4.1)\}$.

Figure 6 shows the turbine velocity/force diagram, with $\{\Gamma, F_i, f_0, \alpha, u_a^*\} < 0$ and $u_t^* > 0$ as drawn. Since $u_a^* < 0$, $\mathbf{u_a^*}$ points in the $\mathbf{e_a}$ direction (as drawn). In this case, the turbine still rotates with angular velocity $\omega\mathbf{e_a}$, but the direction of the circulation is reversed (as drawn). This amounts to $|\Gamma|(-\mathbf{e_r}) = \Gamma\,\mathbf{e_r}$ with $\Gamma < 0$.

With, $\{\Gamma, F_i\} < 0$ but F_v always positive, the thrust and torque acting *on the turbine* are

$$\mathbf{T} = Z\int_{r_h}^{R} [|F_i|\cos\beta_i + F_v \sin\beta_i]dr \;\; (-\mathbf{e_a}) \qquad \text{(as drawn)}$$

$$= Z\int_{r_h}^{R} [F_i\cos\beta_i - F_v \sin\beta_i]dr \;\; (\mathbf{e_a}) \qquad \text{(eqn. 2.3)}$$

$$\mathbf{Q} = Z\int_{r_h}^{R} [|F_i|\sin\beta_i - F_v \cos\beta_i]rdr \quad (\mathbf{e_a}) \qquad \text{(as drawn)}$$

$$= Z\int_{r_h}^{R} [F_i\sin\beta_i + F_v \cos\beta_i]rdr \quad (-\mathbf{e_a}) \qquad \text{(eqn. 2.4)}$$

The fluid dynamic power of the turbine acting *on the fluid* is still

$$P = Q\omega \qquad \text{(eqn. 2.5)}$$

but since $Q < 0$ for the turbine case, $P < 0$, indicating that power is being extracted from the fluid by the turbine.

The geometry of a turbine is also correctly handled when $\Gamma < 0$. In this case, the 2D section lift coefficient

$$C_L = \frac{F_i}{\frac{1}{2}\rho(V^*)^2 c} = \frac{2\Gamma}{(V^*)c} \tag{5.3}$$

is also negative, and this carries through to negative values of the camber and angle of attack via

$$\{C_L, f_0, f, \alpha_I\} = \frac{C_{L_0}}{\widetilde{C}_{L_I}} \cdot \{\widetilde{C}_{L_I}, \widetilde{f}_0, \widetilde{f}, \widetilde{\alpha}_I\} \tag{4.1}$$

where C_{L_0} is the on-design lift coefficient, which is negative.

In summary, the thrust, torque, and power are correctly predicted by equations (2.3), (2.4), and (2.5) when $\Gamma < 0$ for the turbine. Furthermore, since $\{u_a^*, u_t^*, C_L, f_0, \alpha\}$ depend linearly on the circulation, these parameters are also correctly handled when $\Gamma < 0$. Therefore, the same lifting line code can be used for both the propeller and turbine cases!

7 Turbine design optimization

7.1 Simple turbine optimization scheme

One might formulate the turbine optimization problem as follows: Find the set of M circulations of the vortex lattice panels that produce the least torque (i.e. the most negative torque, giving the largest power extraction at the specified rotation rate). In other words, solve the propeller optimization problem, {(3.4),(3.5)}, with no thrust constraint. For simplicity, consider the inviscid flow case, $C_D = 0$. In this formulation, the system of equations for minimizing torque (3.12) becomes:

$$0 = \frac{\partial Q}{\partial \Gamma(i)} = \rho Z \sum_{m=1}^{M} \Gamma(m) \cdot \begin{pmatrix} \bar{u}_a^*(m,i) r_c(m) \triangle r_v(m) + \\ \bar{u}_a^*(i,m) r_c(i) \triangle r_v(i) \end{pmatrix}$$
$$+ \rho Z V_a(i) r_c(i) \triangle r_v(i) \qquad \text{(for } i = 1 \ldots M) \tag{7.1}$$

Figure 7a shows that this scheme does **not** yield the largest power extraction possible (i.e. this scheme does not reproduce actuator disc theory). In this figure, a series of turbines were “optimized" by solving (7.1), and the power coefficients of these turbines are plotted against their tip-speed ratios. The theoretical maximum power extraction at these tip-speed ratios is given by actuator-disc-with-swirl-and-viscous-losses (*ADS*) theory (?), which is shown as a solid black line in figure 7a. Clearly, “optimization" by solving equations (7.1) does not reproduce actuator-disc-with-swirl theory. However, turbines may be designed to replicate *ADS* theory using an ADS-based optimizer, as discussed in the following section. In this example, $C_D = 0$, $Z = 80$, and the axial

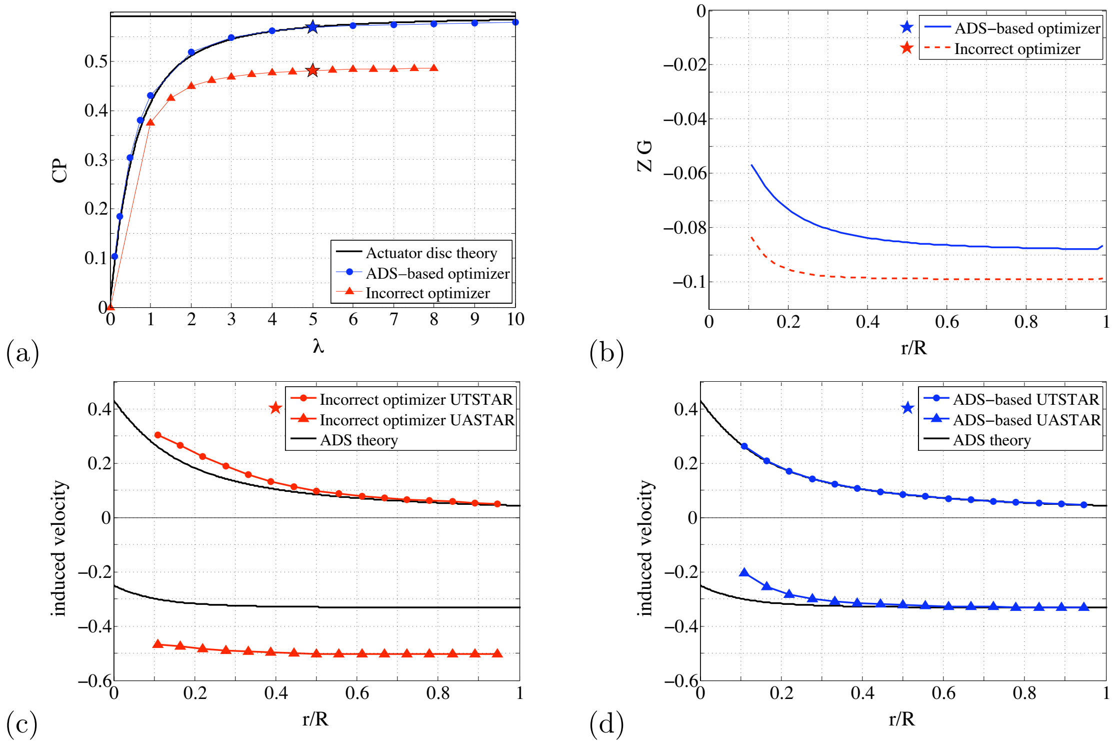


Figure 7: (a) Power coefficient, C_P, versus tip speed ratio, $\lambda = \frac{\omega R}{V_s}$, for turbines “optimized" by solving the system of equations (7.1). Optimization by solving equations (7.1) does not reproduce actuator-disc-with-swirl theory (black line), whereas using the actuator-disk-with-swirl-based optimizer (7.2) does. Here, $C_D = 0$ and $Z = 80$. (b) Circulation $G = \frac{\Gamma}{2\pi R V_s}$ versus radius for the turbines optimized for $\lambda = 5$. (c) Induced velocities $\{\frac{u_a^*}{V_s}, \frac{u_t^*}{V_s}\}$ for the simple optimizer (7.1). (d) Induced velocities $\{\frac{u_a^*}{V_s}, \frac{u_t^*}{V_s}\}$ for the actuator-disk-based optimizer (7.2).

inflow is $\frac{V_a}{V_s} = 1$ for all blade sections.

The question is: why does the system of equations (7.1) under-perform actuator disk theory? Figures 7b, 7c, and 7d show the reason for the under-performance of the (7.1) scheme. Optimizer (7.1) produces turbines that induce axial velocity $\frac{u_a^*}{V_s} = \frac{u_a^*}{V_a} \approx -\frac{1}{2}$ (as shown in figure 7c), whereas actuator disk theory prescribes $\frac{u_a^*}{V_a} = -\frac{1}{3}$ for maximum power extraction. Equations (7.1) do not yield turbines that extract as much power from the flow as *ADS* theory predicts, because solving (7.1) yields a circulation distribution which induces too much axial induced velocity, thereby reducing the flow rate through the turbine more than it should, resulting in less power available for extraction. For comparison, my ADS-based optimizer (7.2) produces turbines that induce axial velocity $\frac{u_a^*}{V_s} \approx -\frac{1}{3}$ (as shown in figure 7d), which is why the power produced by these turbines replicates that of *ADS* theory.

To deduce the under-performance of (7.1) mathematically, note that the horseshoe influence matrices $\{\bar{u}_a^*, \bar{u}_t^*\}$ are dominated by their diagonal terms. To the leading order, the influence functions and induced velocities behave like

$$\bar{u}^*_{a(m,i)} \approx \begin{cases} 0 & (m \neq i) \\ \bar{u}^*_{a(i,i)} & (m = i) \end{cases}$$

$$u^*_{a(i)} \approx \Gamma_{(i)} \bar{u}^*_{a(i,i)}$$

With these approximations, it is evident that the system of equations (7.1) behaves like M independent equations ($i = 1 \dots M$)

$$\begin{aligned} 0 = \rho Z \cdot \Gamma_{(i)} \cdot [2\, \bar{u}^*_{a(i,i)} r_{c(i)} \triangle r_{v(i)}] \\ + \rho Z V_{a(i)} r_{c(i)} \triangle r_{v(i)} \end{aligned}$$

which are each satisfied when

$$u^*_{a(i)} = -\tfrac{1}{2} V_{a(i)}$$

This result is consistent with the example induced velocity distribution shown in figure 7c, as discussed above. In short, the simple turbine optimization scheme (7.1) under-performs actuator disk theory, because solving (7.1) yields a circulation distribution which induces too much axial velocity. Physically, this results in a smaller flow rate through the turbine than actuator disk theory suggests and too little power available for extraction.

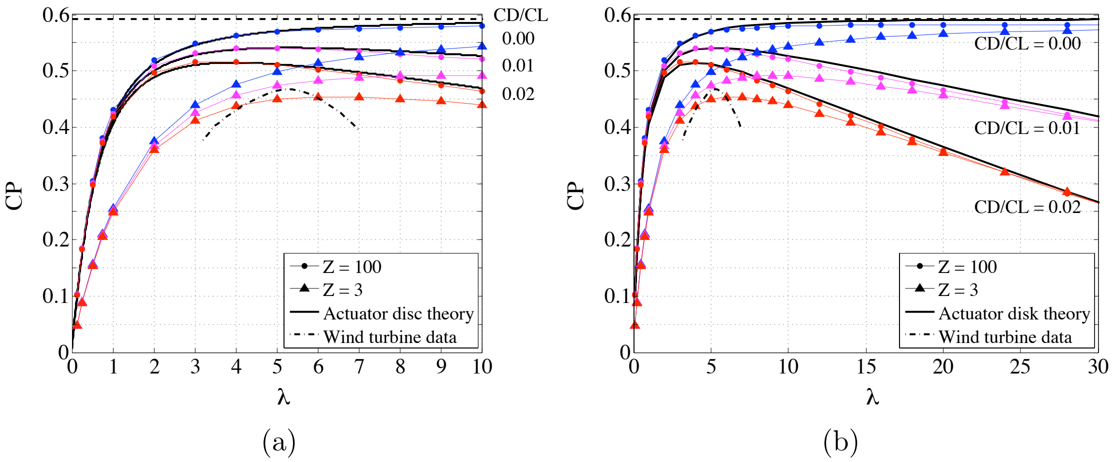
7.2 Improved, actuator-disk-based optimization scheme

Epps et al (2009b) created an actuator-disk-based turbine optimization procedure, which is formulated as follows. The procedure is a vortex-lattice adaptation of actuator-disc-with-swirl-and-viscous-losses (ADS) theory (Stewart, 1976). During the design optimization, flow parameters $\{\Gamma,\, u^*_a,\, u^*_t,\, \bar{u}^*_a,\, \bar{u}^*_t,\, \beta_i\}$ must be self consistent to define a physically-realistic operating state of the turbine. That is, equations {(2.16), (2.17), (2.9), (2.10), (2.2)} must hold, given Γ.

In the present optimization scheme, the tangential induced velocity is set to the actuator disc with swirl (ADS) value

$$u^*_t \equiv u^*_{t,ADS} \tag{7.2}$$

The remaining flow parameters $\{\Gamma, u^*_a, \bar{u}^*_a, \bar{u}^*_t, \beta_i\}$ are determined iteratively. Initially setting $u^*_a = u^*_{a,ADS}$ allows one to start a loop that computes β_i via (2.2), then $\{\bar{u}^*_a, \bar{u}^*_t\}$ via {(2.9), (2.10)}. Then,



(a) (b)

Figure 8: (a) Power coefficient, $C_P = P/\frac{1}{2}\rho V_\infty^3 \pi R^2$, versus tip speed ratio, $\lambda = \frac{\omega R}{V_\infty}$, for optimized turbines. The C_P of turbines designed with 100 blades agrees quite well with actuator-disc-with-swirl-and-viscous-losses theory (Stewart, 1976), as shown for three C_D/C_L ratios. Performance data of 3-bladed wind turbines in service, digitized from (Kahn 2006), is also shown for reference. (b) Power coefficients of 3-bladed and 100-bladed turbines converge for high tip speed ratios ($\lambda > 25$), as expected.

the circulation distribution can then be determined by solving the matrix equation

$$[\bar{u}_t^*] \cdot [\Gamma] = [u_{t,ADS}^*]$$

for Γ. Finally, u_a^* is computed via (2.16), and the loop restarts. Iteration continues until every state variable has converged.

The performance of several turbines optimized using this scheme is shown in figure 8. Using this scheme (7.2), one can reproduce the C_P vs. λ performance curves from ADS theory (Stewart, 1976), as shown by the (essentially infinite-bladed) $Z = 100$ curves in figure 8a. An additional check that this scheme works correctly, which is shown in figure 8b, is that for very high tip speed ratios ($\lambda > 25$), each of the $Z = 3$ curves asymptotes to its corresponding $Z = 100$ curve, as expected.

7.3 Chord length optimization

During the circulation optimization procedure, the chord, c, can chosen in order to restrict the lift coefficient to a given maximum allowable absolute value, $C_{L_{\max}}$, such that

$$C_L = C_{L_{\max}} \cdot \frac{\Gamma}{|\Gamma|} \tag{7.3}$$

$$c = \frac{2|\Gamma|}{(V^*) C_{L_{\max}}} \tag{7.4}$$

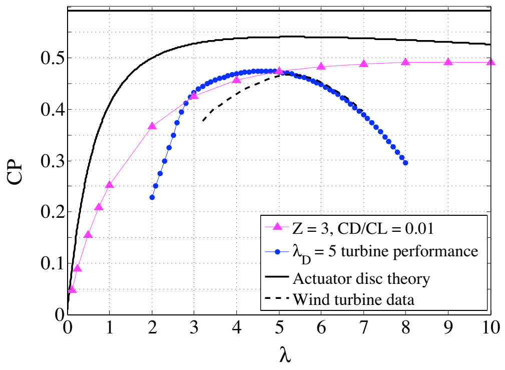


Figure 9: Power coefficient, C_P, versus off-design tip speed ratio, λ, for a turbine designed to operate at $\lambda_D = 5$, with specifications $C_D = 0.01$ and $Z = 3$.

8 Off-design performance analysis

The same off-design performance analysis method presented in Section 5 can be used for propellers as well as turbines. The Newton solver implemented in OpenProp was in fact written for the turbine case, where the operating state is characterized by an off-design (OD) *tip-speed ratio*,

$$\lambda_{OD} = \frac{\omega_{OD} R}{V_s} = \frac{\pi}{J_{s,OD}} \tag{8.1}$$

and unknown flow parameters $\{V^*, \alpha, C_L, \Gamma, u_a^*, u_t^*, \beta_i, \bar{u}_a^*, \bar{u}_t^*\}$. Since the same governing equations apply to the propeller and turbine cases, the same code can be used for either. It should be noted that at the time of this publication, the ducted design or off-design analysis features were not validated for the turbine case.

An example off-design analysis is presented in figure 9. For reference, the ADS performance frontier, industry wind turbine data, and the performance of ADS-based optimized turbines with $C_D/C_L = 0.01$ and $Z = 3$ are reproduced from figure 8. The off-design performance is shown for the turbine designed to operate at $\lambda_D = 5$. The performance predicted by the analyzer ('●') agrees with the performance predicted by the optimizer ('▲') at $\lambda = 5$, and the performance for higher tip speed ratios compares quite favorably with wind turbine industry performance data from (Kahn 2006). For $\lambda < 3$, the power coefficient drops precipitously, as the net angle of attack drops below the specified stall angle (-8°) at many blade sections and the blade stalls. For $3 < \lambda < 5$, the turbine optimized for $\lambda_D = 5$ outperforms the ADS-based performance frontier. That is, the $\lambda_D = 5$ turbine ('●') outperforms the 'optimized' turbines at $\lambda = 4$ and 3 ('▲'), indicating that the ADS-based optimization method does not truly find the best configurations possible. Reformulating the turbine optimization method is one focus of ongoing work.

9 Using OpenProp for ship resistance and powering calculations

This section reviews ship resistance and powering and outlines how to use OpenProp to perform such calculations. Assume that bare ship resistance data is given, R_s, tabulated versus ship speed, V_s. The *effective power* required to propel the ship at speed V_s then is

$$P_E \equiv R_s V_s \tag{9.1}$$

The power delivered to the propeller at ship speed V_s is called the *delivered power*,

$$P_D \equiv (2\pi n) Q \tag{9.2}$$

where n and Q are the rotation rate (rev/s) and torque at the propeller. The ratio of the effective power to the delivered power is known as the *quasi-propulsive coefficient*

$$\eta_D \equiv \frac{P_E}{P_D} \tag{9.3}$$

A propeller operating in the wake of a ship sees a distribution of (circumferential average) axial inflow, $V_a(r)$, different from (and often less than) the ship speed. The *volumetric mean inflow velocity* is defined as

$$\bar{V_A} \equiv \frac{1}{\pi R^2 - \pi R_{\text{hub}}^2} \int_{R_{\text{hub}}}^{R} 2\pi r \, V_a(r) \, dr \tag{9.4}$$

Moreover, the action of the propeller alters the pressure field about the ship, often requiring propeller thrust T greater than the bare ship resistance R_s to maintain the same ship speed, V_s. Combining these two effects, the *thrust power* is defined as

$$P_T \equiv T \bar{V_A} \tag{9.5}$$

Thus, the efficiency of the propeller operating behind the ship is

$$\eta_B \equiv \frac{P_T}{P_D} = \frac{T \bar{V_A}}{(2\pi n) Q} = \frac{J_s}{2\pi} \frac{K_T}{K_Q} \frac{\bar{V_A}}{V_s} \tag{9.6}$$

This is the efficiency computed by OpenProp. The off-design analysis code in OpenProp calculates K_T, K_Q, and η_B versus *advance coefficient* J_s, defined as.

$$J_s = \frac{V_s}{nD} \tag{9.7}$$

For clarity and completeness, several other definitions are now given. The *advance coefficient* can also be defined using the volumetric mean inflow speed, as

$$J_A = \frac{\bar{V}_A}{nD} \tag{9.8}$$

In *open water*, the propeller sees uniform axial inflow, $V_a(r) = \bar{V}_A$. A propeller operating in open water with the same speed $\bar{V}_A$ and advance coefficient J_A (i.e. the same rotation rate n) as the wake inflow case, will produce thrust T_o and torque Q_o and thus have *open-water efficiency* of

$$\eta_o \equiv \frac{T_o \bar{V}_A}{(2\pi n) Q_o} \tag{9.9}$$

The *relative-rotative efficiency* is defined as the ratio of the *behind-ship efficiency* to the *open-water efficiency* with the propeller operating at the same advance coefficient J_A,

$$\eta_R \equiv \frac{\eta_B}{\eta_o} \tag{9.10}$$

The *Taylor wake fraction*, w, is defined as $(1 - w) = \frac{V_a(r)}{V_s}$. The volumetric mean wake fraction w_v is calculated in a similar manner as equation (9.4).

$$(1 - w_v) = \frac{\bar{V}_A}{V_s} \tag{9.11}$$

Given a wake profile $V_a(r)$, OpenProp computes VMIV $\equiv \frac{\bar{V}_A}{V_s}$ and stores it in the design data structure. The *thrust deduction factor* relates the propeller thrust T required to drive the ship at speed V_s to the bare hull ship resistance at that speed, R_s, via

$$(1 - t) = \frac{R_s}{T} \tag{9.12}$$

Building on these definitions, the *hull efficiency* is defined as

$$\eta_H \equiv \frac{P_E}{P_T} = \frac{R_s V_s}{T \bar{V}_A} = \frac{(1 - t)}{(1 - w_v)} \tag{9.13}$$

Thus, the *quasi-propulsive coefficient* can be written as

$$\eta_D = \eta_H \eta_B = \eta_H \eta_R \eta_o \tag{9.14}$$

The *shaft transmission efficiency* (typically assumed to be 1.0) is in general

$$\eta_S = \frac{P_D}{P_S} \tag{9.15}$$

where P_S is the shaft power output from the engine (typically taken as the power required of the ship's engine). Combining the *quasi-propulsive coefficient* and the *shaft transmission efficiency*, one arrives at the *overall propulsive efficiency*

$$\eta_P \equiv \frac{P_E}{P_S} = \eta_D \eta_S = \eta_H \eta_B \eta_S \tag{9.16}$$

9.1 Computing powering requirements

The following procedure can be used to compute ship powering requirements using OpenProp.

1. Given the following:
   - ship resistance data, R_s, tabulated versus ship speed, V_s,
   - propeller design, with wake profile $V_a(r)$ or wake fraction w_v,
   - thrust-deduction factor, t,
   - shaft transmission efficiency, η_S.
2. Use Analyze.m to find the (wake-adapted) propeller performance curve:

$$\{C_T^*, K_T^*, K_Q^*, \eta_B^*, \ldots\} \quad \text{versus} \quad J_s^* \tag{9.17}$$

   (where * indicates that the data is tabulated versus these particular J_s^*.)
3. Find required propeller thrust (for each ship speed):

$$T = \frac{R_s}{(1-t)} \tag{9.18}$$

4. Find required thrust coefficient (for each ship speed):

$$C_T = \frac{T}{\frac{1}{2}\rho V_s^2 \pi R^2} \tag{9.19}$$

5. Interpolate to find the advance coefficient (for each ship speed):

$$J_s = \text{interpolate}(C_T^*, J_s^*, C_T) \tag{9.20}$$

6. Interpolate to find the propeller behind-hull efficiency (for each ship speed):

$$\eta_B = \text{interpolate}(J_s^*, \eta_B^*, J_s) \tag{9.21}$$

7. Interpolate to find torque coefficient (for each ship speed):

$$K_Q = \text{interpolate}(J_s^*, K_Q^*, J_s) \tag{9.22}$$

8. Find required rotation rate (for each ship speed):

$$n = \frac{V_s}{J_s D} \tag{9.23}$$

9. Find required torque (for each ship speed):

$$Q = K_Q \rho n^2 D^5 \tag{9.24}$$

10. Find delivered power (for each ship speed):

$$P_D = (2\pi n) Q \tag{9.25}$$

11. Find the engine shaft power (for each ship speed):

$$P_S = \frac{P_D}{\eta_S} \tag{9.26}$$

12. Find the hull efficiency (same for all ship speeds):

$$\eta_H = \frac{R_s V_s}{T \bar{V}_A} \tag{9.27}$$

13. Find the overall propulsive efficiency (for each ship speed):

$$\eta_P = \eta_D \eta_S = \eta_H \eta_B \eta_S \tag{9.28}$$

10 Non-dimensional parameters

Table 1 summarizes the non-dimensional form of the flow- and performance parameters used herein.

OpenProp	Herein	Description
Vs	V_s	ship speed (free-stream speed) [m/s]
R	R	propeller radius [m]
D	D	propeller diameter [m]
n	n	rotation rate [rev/s] ($\omega = 2\pi n$)
Rhub_oR	r_h/R	normalized hub radius
RC	r_c/R	normalized control point radius
DR	$\triangle r_v/R$	normalized difference in vortex radii
CoD	c/D	normalized section chord
VAC	V_a/V_s	normalized axial inflow velocity
VTC	V_t/V_s	normalized tangential inflow velocity
UASTAR	u_a^*/V_s	normalized induced axial velocity
UTSTAR	u_t^*/V_s	normalized induced tangential velocity
UAHIF	$2\pi R \cdot \bar{u}_a^*$	normalized axial horseshoe influence function
UTHIF	$2\pi R \cdot \bar{u}_t^*$	normalized tangential horseshoe influence function
G	$\Gamma/(2\pi R V_s)$	normalized circulation
VSTAR	V^*/V_s	normalized total inflow speed
dVdG	$2\pi R \cdot \frac{\partial V^*}{\partial \Gamma}$	normalized $\frac{\partial V^*}{\partial \Gamma}$
dVdW	$\frac{\partial V^*}{\partial \omega}/R$	normalized $\frac{\partial V^*}{\partial \omega}$
LM	λ_1/R	normalized Lagrange multiplier
CT	$C_T = \frac{T}{\frac{1}{2}\rho V_s^2 \pi R^2}$	thrust coefficient based on ship speed
CQ	$C_Q = \frac{Q}{\frac{1}{2}\rho V_s^2 \pi R^3}$	torque coefficient based on ship speed
CP	$C_P = \frac{Q\omega}{\frac{1}{2}\rho V_s^3 (\pi R^2)}$	power coefficient ($C_P = \frac{\omega R}{V_s} C_Q = \lambda C_Q = \frac{C_Q \pi}{J_s}$)
KT	$K_T = \frac{T}{\rho n^2 D^4}$	thrust coefficient based on blade tip speed
KQ	$K_Q = \frac{Q}{\rho n^2 D^5}$	torque coefficient based on blade tip speed
Js	$J_s = \frac{V_s}{nD} = \frac{\pi V_s}{\omega R}$	advance coefficient
L	$\lambda = \frac{\omega R}{V_s} = \frac{\pi}{J_s}$	tip-speed ratio

Table 1: Table of non-dimensional flow parameters in OpenProp.

References

Abbott IH, von Doenhoff AE (1959) Theory of Wing Sections Dover.

Anderson JD (2007) Fundamentals of Aerodynamics 4th edn. McGraw Hill.

Black SD (1997) Integrated lifting surface / Navier-Stokes design and analysis methods for marine propulsors PhD thesis, MIT.

Carlton JS (1994) Marine Propellers and Propulsion Butterworth-Heinemann.

Chung HL (2007) An enhanced propeller design program based on propeller vortex lattice lifting line theory Master's thesis, MIT.

Coney WB (1989) A Method for the Design of a Class of Optimum Marine Propulsors Ph.D. thesis, MIT.

D'Epagnier KP, Chung HL, Stanway MJ, Kimball RW (2007) "An open source parametric propeller design tool," In: Oceans.

Drela M (1989) "XFOIL: An analysis and design system for low Reynolds number airfoils," In: Low Reynolds Number Aerodynamics: Proceedings for the Conference Notre Dame, Springer-Verlag.

Drela M (1999) "Integrated simulation model for preliminary aerodynamic, structural, and control-law design of aircraft," In: AIAA SDM Conference 99-1394.

Epps BP, Chalfant JS, Kimball RW, Techet AH, Flood K, Chryssostomidis C (2009a) "OpenProp: An open-source parametric design and analysis tool for propellers," In: Grand Challenges in Modeling and Simulation (GCMS09) Istanbul, Turkey.

Epps BP, Stanway MJ, Kimball RW (2009b) "OpenProp: An open-source design tool for propellers and turbines," In: SNAME Propellers and Shafting.

Epps BP (2010) An impulse framework for hydrodynamic force analysis: fish propulsion, water entry of spheres, and marine propellers Ph.D. thesis, MIT.

Flood K (2009) Propeller performance analysis using lifting line theory Master's thesis, MIT.

Hough GR and Ordway DE (1964) "The Generalized Actuator Disk", TAR-TR 6401, Therm Advanced Research, Inc.

Kerwin JE (2007) Hydrofoils and propellers MIT course 2.23 notes.

Kerwin JE and Hadler JB (2010) Principles of Naval Architecture: Propulsion SNAME, to appear.

Lerbs H (1952) “Moderately loaded propellers with a finite number of blades and an arbitrary distribution of circulation," In: SNAME Transactions vol 60.

Newman J (1977) Marine Hydrodynamics MIT Press.

Peterson CJ (2008) Minimum pressure envelope cavitation analysis using two-dimensional panel method Master’s thesis, MIT.

Saffman P (1995) Vortex Dynamics Cambridge University Press.

Stubblefield JM (2008) Numerically based ducted propeller design using vortex lattice lifting line theory Master’s thesis, MIT.

Theodorsen T (1935) “General theory of aerodynamic instibility and the mechanism of flutter," Technical Report TR490, NACA.

Wrench JW (1957) “The calculation of propeller induction factors," Technical Report 1116, David Taylor Model Basin.