Journal of Ship Production and Design, Vol. 32, No. 3, August 2016, pp. 1–15
http://dx.doi.org/10.5957/JSPD.32.3.150035

On the rotor lifting line wake model

Brenden Epps

Assistant Professor, Thayer School of Engineering, Dartmouth College, Hanover, NH, USA

Abstract

This article comments on the wake model used in moderately-loaded rotor lifting line theory for the preliminary design of propellers and horizontal-axis turbines. Mathematical analysis of the classical wake model reveals an inconsistency between the induced velocities numerically computed by the model versus those theoretically predicted by the model. An improved wake model is presented, which better agrees with theory than previous models and thus improves the numerical consistency and robustness of rotor lifting line design algorithms. The present wake model analytically relates the pitch of the trailing vortices to the pitch of the total inflow computed at the lifting line control points. For conciseness, the article focuses on the propeller case, although both propeller and horizontal-axis turbine examples are presented.

1 Introduction

Broadly speaking, methods for propeller design may be classified into two categories: *searching methods* or *mathematical optimization methods*. Searching methods attempt to find an optimized propeller geometry through trial and error. Typically, the propeller geometry is parameterized by the pitch, camber, and chord distributions, which are varied in either a systematic or random fashion to explore the design space. Alternatively, the geometry may be characterized by B-spline surface amplitudes, which would be varied in the searching algorithm. For each geometry, the performance is analyzed either by experiment or by numerical analysis.

Examples of searching methods applied to propeller design abound: Mertes and Heinke (2008) vary propeller pitch and camber, and analyze the resulting geometries using a lifting line model with lifting surface geometry corrections. Tamura et al (2010) also vary pitch and camber but analyze using a lifting surface method. Others use panel methods for performance analysis (Pashias and Turnock, 2003; Kim et al, 2009; Laurens et al, 2012, e.g.). Funeno (2009) use the RANSE method code `STAR-CD` for analysis and design of a ducted propeller.

Experimental searching methods involve the acquisition of a systematic series of experimental data, the most widely-known example being the Wangeningen B-Screw Series (Oosterveld and van Oossanen, 1975; Bernitsas et al, 1981). Isin (1987) interpolated these data to estimate the performance of a novel propeller design.

A propeller design may instead be optimized by way of solving a mathematical optimization problem. Maximizing propeller efficiency involves minimizing torque for a specified thrust (or maximizing thrust for a specified torque). This method is limited by the types of mathematical optimization problems that can be constructed. From an analytical point of view, the lifting line model is the only model simple enough to admit analytic design optimization. Thus, mathematical optimization methods are typically used with lifting line codes, which form the first step in the design process.

However, propeller design optimization involves more than simply determining the optimum circulation distribution to maximize efficiency: Duct geometry design, cavitation considerations, and unsteady loading all play a factor in final propeller design. Thus, a lifting line code is the first of many computational tools the designer might use. A propeller design workflow might be as follows:

1. ‘Lifting line method’ (e.g. `PLL`, `OpenProp`) used for preliminary design optimization of the circulation, chord length, and thickness distributions;

2. ‘Coupled lifting surface / Euler solver method’ (e.g. `PBD-MTFLOW` (Kerwin et al, 1994, 2006)) used for propeller blade design and analysis;

3. ‘Panel method’ (e.g. `MPUF-3A` (Lee, 1979; He, 2010), or unnamed code (Gaggero and Brizzolara, 2007)) or ‘RANSE method’ (e.g. `Star-CCM+` (CD-Adapco, 2008; Brizzolara et al, 2008)) used for final performance check and cavitation analysis.

For example, `PLL` and then `PBD-MTFLOW` were recently used to design an optimized propeller for the *Tethys* autonomous underwater vehicle for long range operations (Bellingham et al, 2010).

Lifting line design optimization is the important first step in this workflow, as the results of a lifting line code set design targets for the propeller blade design code, and so on. Lifting line codes for the design of free-running and wake-adapted propulsors have been available for many years (e.g. PLL (Coney, 1989), OpenProp (Epps et al, 2009a; Epps, 2010a; Epps et al, 2011) and LLOPT (Zan, 2008; Arán and Kinnas, 2013)). These prior codes are all based on a discretized vortex lattice lifting line model, in which the circulation is assumed constant over each segment of the lifting line, and each segment of the lifting line is surrounded by a horseshoe vortex.

Recent extension of the rotor lifting line model to the design of horizontal-axis turbines uncovered an inconsistency between the mathematical equations representing the lifting line wake model and the numerical implementation of these equations in all of these prior codes (Epps and Kimball, 2013b). This article investigates this inconsistency and proposes a novel wake model that ameliorates this problem.

This article is organized as follows: In §2, the propeller lifting line model is reviewed, with specific attention given to the wake model in §2.1. Section §2.1.2 presents an improved wake model, which more accurately agrees with the mathematical framework of moderately loaded lifting line theory than the classical wake model. Supporting analytical analysis is provided in §3, and illustrative examples are provided in §4. Appendix A reviews the details of propeller design optimization. In the interest of brevity, some variables are defined in the Nomenclature section at the end of this article and not in the body text.

2 Rotor lifting-line model

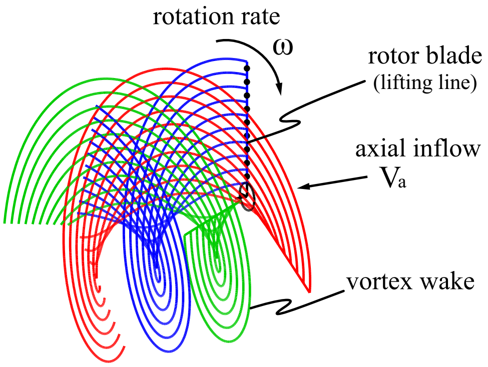

Figure 1: Classical rotor lifting line wake model, with control points on the key lifting line illustrated by '•'.

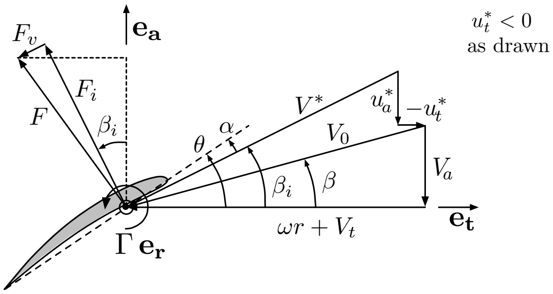

Figure 2: Propeller velocity-force diagram at radius r. Variables are defined in the Nomenclature table.

As illustrated in Figure 1, the rotor lifting line model is based upon the fundamental assumptions of moderately-loaded propeller theory: (i) the lifting lines have equal angular spacing and identical loading; (ii) the lifting lines are straight, radial lines; and (iii) each trailing vortex is assumed to be a helix with fixed radius and pitch, where the wake pitch angle $\beta_w(r_v)$ is related to the total inflow angle at the lifting line, $\beta_i(r_c)$.

Figure 2 illustrates the velocities and forces (per unit radius) on a 2D blade section: axial and tangential inflow velocities, V_a and V_t; induced velocities, u_a^* and u_t^*; and angular velocity ω. The total resultant inflow velocity has magnitude $V^* = \sqrt{(V_a + u_a^*)^2 + (\omega r + V_t + u_t^*)^2}$ and is oriented at pitch angle β_i, where

$$\tan \beta_i = \frac{V_a + u_a^*}{\omega r + V_t + u_t^*} \tag{2.1}$$

A review of the rotor lifting line model can be found in (Kerwin and Hadler, 2010) or (Epps and Kimball, 2013b).

2.1 Wake model

In general, the velocity induced by a vortex filament can be computed via the Biot-Savard Law:

$$\vec{u} = \frac{\bar{\Gamma}}{4\pi} \int \frac{d\vec{s} \times \vec{\rho}}{|\vec{\rho}|^3} \tag{2.2}$$

where $\bar{\Gamma}$ is the circulation about the vortex filament, $d\vec{s}$ is a segment of the filament, and $\vec{\rho}$ is the vector from the vortex segment to the field point at which the velocity is being calculated. Careful consideration of assumptions (*i*) and (*ii*) reveals that, by symmetry, the lifting lines do not contribute to the induced velocity. For a continuous circulation distribution, $\Gamma(r)$, the trailing vortex strength is

$$\bar{\Gamma}(r) = \left[-\frac{d\Gamma(r)}{dr} \right] dr \tag{2.3}$$

Integrating the influence of the trailing vortex shed from each radius, r_v, equation (2.2) can be recast as:

$$u_a^*(r_c) = \int_{R_h}^{R} \bar{u}_a(r_c, r_v, \beta_w) \left[-\frac{d\Gamma(r_v)}{dr_v} \right] dr_v \tag{2.4}$$

$$\bar{u}_a(r_c, r_v, \beta_w) = \frac{1}{4\pi} \sum_{k=1}^{\mathcal{Z}} \int \hat{e}_a \cdot \frac{d\vec{s} \times \vec{\rho}}{|\vec{\rho}|^3} \tag{2.5}$$

Similarly, u_t^* can be written in terms of Γ and $\bar{u}_t$. The sign convention in (2.4) has positive circulation $\bar{\Gamma}$ directed downstream, away from the lifting line (by right hand rule). The functions $\bar{u}_a$ and $\bar{u}_t$ account for the influence of a unit-strength trailing vortex shed from each of the $\mathcal{Z}$ blades. Thus, they only depend on the wake geometry (in particular, the wake pitch angle β_w). For constant-pitch, constant-radius helical vortex filaments, (2.5) can be evaluated analytically (Lerbs, 1952)

For $r_c < r_v$:

$$\begin{aligned} \bar{u}_a(r_c, r_v, \beta_w) &= \frac{\mathcal{Z}}{4\pi r_c} y - \frac{\mathcal{Z} S_1}{4\pi r_c} y \\ \bar{u}_t(r_c, r_v, \beta_w) &= \frac{\mathcal{Z} S_1}{4\pi r_c} \end{aligned} \tag{2.6a}$$

For $r_c > r_v$:

$$\begin{aligned} \bar{u}_a(r_c, r_v, \beta_w) &= -\frac{\mathcal{Z} S_2}{4\pi r_c} y \\ \bar{u}_t(r_c, r_v, \beta_w) &= \frac{\mathcal{Z}}{4\pi r_c} + \frac{\mathcal{Z} S_2}{4\pi r_c} \end{aligned} \tag{2.6b}$$

where

$$\begin{aligned} y &= \frac{r_c}{r_v \tan\beta_w} \\ y_0 &= \frac{1}{\tan\beta_w} \end{aligned} \tag{2.6c}$$

The first term in (2.6a) and (2.6b) is the circumferential average induced velocity, and the second term is the circumferentially-varying component, which is singular (i.e. $S_1 \to -\infty$ as $r_c \to r_v^-$ and $S_2 \to \infty$ as $r_c \to r_v^+$). These singular terms are given by (Lerbs, 1952) as an infinite series of Bessel functions, but they can instead be computed using the asymptotic formulae of (Wrench, 1957)[†]

$$\begin{aligned} S_1 &= -A \left[\frac{U}{1-U} + B \ln\left(1 + \frac{U}{1-U}\right) \right] \\ S_2 &= A \left[\frac{1}{U-1} - B \ln\left(1 + \frac{1}{U-1}\right) \right] \\ A &= \left(\frac{1+y_0^2}{1+y^2} \right)^{\frac{1}{4}} \\ B &= \frac{1}{24\mathcal{Z}} \left[\frac{9y_0^2+2}{(1+y_0^2)^{1.5}} + \frac{3y^2-2}{(1+y^2)^{1.5}} \right] \\ U &= \left[\frac{y_0\left(\sqrt{1+y^2}-1\right)}{y\left(\sqrt{1+y_0^2}-1\right)} \exp\left(\sqrt{1+y^2} - \sqrt{1+y_0^2}\right) \right]^{\mathcal{Z}} \end{aligned} \tag{2.6d}$$

For numerical implementation, equation (2.4) can either be formulated assuming $\Gamma(r)$ is a continuous function or it can be formulated assuming that $\Gamma(r)$ is piecewise constant across each segment of the lifting line.

Lerbs' (1952) *continuous formulation* employed a continuous circulation distribution $\Gamma(r)$, which was represented by a Fourier sine series. In this case, (2.4) is an improper integral, the principal value of which Lerbs determined analytically in terms of the Fourier coefficients. Implicit in his model is that the wake consisted of radially-continuous vortex sheets, faithfully representing physical reality. However, two disadvantages of Lerbs' theory exist: (1) the sine series representation required the circulation to be zero at the hub, which is not physically realistic[‡]; and (2) Lerbs' optimization equation ((A.9) herein) is not extendable to multi-component propulsors. Hence, a discretized formulation of (2.4) has been found to be advantageous.

In the discrete *vortex lattice lifting line formulation* (Kerwin, Coney, and Hsin, 1986), the lifting lines are discretized into M constant-strength panels. The induced velocities are computed at 'control points' on the key lifting line at radii $r_{c(m)}$ $(m{=}1 \ldots M)$, which typically

[†] Note that Wrench (1957) gives equations for $F_1 \equiv S_1/2Zy_0$ and $F_2 \equiv S_2/2Zy_0$, as opposed to the more convenient S_1 and S_2 notation adopted herein.

[‡] It is possible to model non-zero hub circulation by including half-sine terms in the Fourier expansion.

are located at the midpoint of each panel. The panel endpoints are denoted by radii $r_{v(m)}$ ($m{=}1 \ldots M{+}1$), so panel n spans radii $r_{v(n)}$ to $r_{v(n+1)}$ and contains $r_{c(n)}$. Let $\Gamma_{(m)} \equiv \Gamma(r_{c(m)})$, $u^*_{a(m)} \equiv u^*_a(r_{c(m)})$, and so on.

2.1.1 The classical wake model

In the *classical wake model* (Kerwin, Coney, and Hsin, 1986), one trailing vortex is shed from each panel endpoint; the trailing vortex at radius $r_{v(n)}$ and has strength

$$\bar{\Gamma}(r_{v(n)}) = \Gamma_{(n-1)} - \Gamma_{(n)} \tag{2.7}$$

where again the sign convention is that $\bar{\Gamma}$ is positive when pointing downstream. The induced velocities (2.4) are now computed by discrete summation

$$\begin{aligned} u^*_{a(m)} = \quad & \bar{u}_a\big(r_{c(m)}, r_{v(1)}, \beta_{w(1)}\big)\big[-\Gamma_{(1)}\big] \\ & + \sum_{n=2}^{M} \bar{u}_a\big(r_{c(m)}, r_{v(n)}, \beta_{w(n)}\big)\big[\Gamma_{(n-1)} - \Gamma_{(n)}\big] \\ & + \bar{u}_a\big(r_{c(m)}, r_{v(M+1)}, \beta_{w(M+1)}\big)\big[\Gamma_{(M+1)}\big] \end{aligned} \tag{2.8}$$

(and similarly for u^*_t). Thus, equation (2.4) in Lerbs' continuous formulation is replaced by equation (2.8) in Kerwin's discrete formulation. Equation (2.8) can be rewritten in terms of 'horseshoe influence functions' ($\bar{u}^*_a$ and $\bar{u}^*_t$) as follows:

$$u^*_{a(m)} = \sum_{n=1}^{M} \bar{u}^*_{a(m,n)}\,\Gamma_{(n)} \tag{2.9}$$

$$\begin{aligned} \bar{u}^*_{a(m,n)} = & \; \bar{u}_a\big(r_{c(m)}, r_{v(n+1)}, \beta_{w(n+1)}\big) \\ & - \bar{u}_a\big(r_{c(m)}, r_{v(n)}, \beta_{w(n)}\big) \end{aligned} \tag{2.10}$$

where $\bar{u}_a$ and $\bar{u}_t$ are still given by (2.6).

In Lerbs' continuous formulation, the wake pitch angle is set to the total inflow angle at the lifting line, $\beta_w(r) = \beta_i(r)$, but in the discretized formulation, $\beta_w(r_v)$, must be interpolated from $\beta_i(r_c)$. For example, `PLL` performs spline interpolation,

$$\tan\beta_v(r_{v(n)}) = \texttt{spline}(r_c, \tan\beta_i, r_{v(n)}),$$

and `OpenProp v2.4` performs 'piecewise cubic Hermite interpolating polynomial' interpolation,

$$\tan\beta_v(r_{v(n)}) = \texttt{pchip}(r_c, \tan\beta_i, r_{v(n)})$$

Consistent with Lerbs' approach, the wake pitch angle is then set to these interpolated values:

$$\tan\beta_{w(n)} = \tan\beta_v(r_{v(n)}) \tag{2.11}$$

The problem with the *classical wake model* is that equations (2.8) and (2.11) do not faithfully represent the model embodied in (2.4). For example, in Lerbs' continuous formulation, the circumferential average component of $\bar{u}_a(r_c, r_v{=}r_c, \beta_w(r_v{=}r_c))$ depends on the inflow angle at the control point $\beta_w(r_v{=}r_c) = \beta_i(r_c)$; however, in this discrete formulation, $\bar{u}^*_{a(m,m)}$ depends on the interpolated value at a different radius, namely $\beta_{w(m+1)} = \beta_v(r_{v(m+1)})$. This and other mathematical inconsistencies regarding the singular terms (S_1 and S_2) will be discussed in §3.

2.1.2 The present wake model

The *present wake model* allows for the possibility that two distinct trailing vortices are shed from each panel junction. In this model, each panel is surrounded by its own complete horseshoe vortex. This is consistent with the spirt of (2.10) but allows additional mathematical flexibility in defining the wake pitch angles. In the present wake model, the pitch of the two trailing vortices for each horseshoe vortex are analytically related to the inflow angle at the control point for that horseshoe. Sections §3 will show that this model faithfully represents the model embodied in (2.4).

In the present wake model, the induced velocities are still given by (2.9). However, the horseshoe influence functions ($\bar{u}^*_a$ and $\bar{u}^*_t$) are redefined. Let '0' and '1' denote the panel endpoints, and generalize (2.10) as follows:

$$\begin{aligned} \bar{u}^*_{a(m,n)} = & \; \bar{u}_a\big(r_{c(m)}, r_{v(n+1)}, \beta_{w(n,1)}\big) \\ & - \bar{u}_a\big(r_{c(m)}, r_{v(n)}, \beta_{w(n,0)}\big) \end{aligned} \tag{2.12}$$

In the present wake model, the wake pitch at the panel endpoints is set such that it is mathematically consistent with the pitch computed at the panel control point:

$$\begin{aligned} \tan\beta_{w(n,1)} &= r_{c(n)} \tan\beta_{i(n)} / r_{v(n+1)} \\ \tan\beta_{w(n,0)} &= r_{c(n)} \tan\beta_{i(n)} / r_{v(n)} \end{aligned} \tag{2.13}$$

In other words, the wake pitch, $2\pi r \tan\beta_w$, is assumed to be constant over the span of each vortex panel, with the value taken as that computed at the control point, $2\pi r_c \tan\beta_i$.

The *classical wake model* can also be represented in the notation of (2.12) by setting

$$\begin{aligned} \tan\beta_{w(n,0)} &= \tan\beta_{v(n)} \\ \tan\beta_{w(n,1)} &= \tan\beta_{v(n+1)} \end{aligned} \tag{2.14}$$

In summary, both vortex lattice lifting line methods can be represented by equations (2.9), (2.12), and (2.6). The *present wake model* further employs (2.13), whereas the *classical wake model* employs (2.14).

2.1.3 Illustration and discussion

The *classical wake model* and *present wake model* are graphically illustrated in Figures 3 and 4. The present wake model assumes each vortex panel is a constant-pitch vortex sheet, with the pitch taken as that computed at the control point. No interpolation is needed, and the wake pitch is analytically consistent with the computed pitch at the control points.

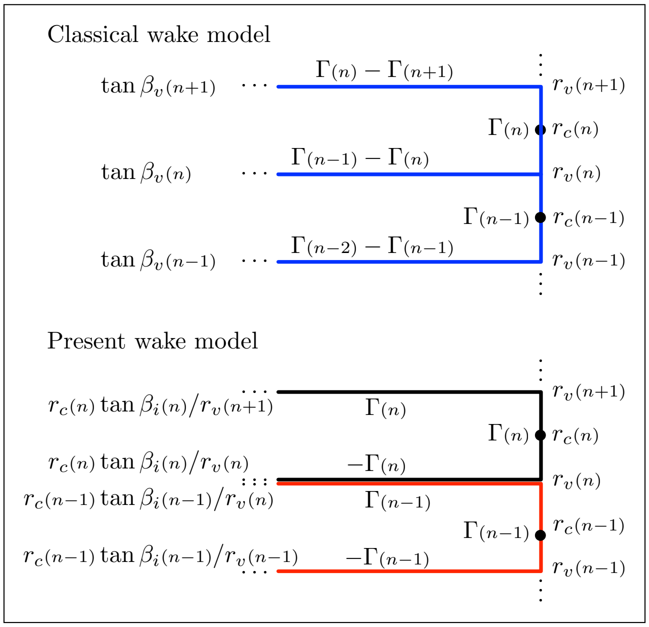

Figure 3: The classical wake model represents a single vortex sheet, whereas the present wake model represents each panel as a radially-constant-pitch vortex sheet (adjacent panels plotted in black and red for clarity).

In the classical wake model, one trailing vortex springs from the lifting line at each interior vortex point $r_v(n{=}2,\ldots,M)$, and the strength of this vortex is the difference in the circulation between adjacent vortex panels, $\Gamma(n{-}1) - \Gamma(n)$. In the present model, *two* vortices are shed at each interior vortex point: The vortex shed from $r_v(n)$ with pitch angle $r_c(n{-}1)\tan\beta_i(n{-}1)/r_v(n)$ has strength $\Gamma(n{-}1)$, and the vortex shed with pitch angle $r_c(n)\tan\beta_i(n)/r_v(n)$ has strength $-\Gamma(n)$. In the case that the wake truly is constant-pitch, $r_c(n{-}1)\tan\beta_i(n{-}1) = r_c(n)\tan\beta_i(n)$ and these two vortices are coincident, as in the classical model. In both models, one trailing vortex springs from each exterior vortex point ($r_v(n{=}1)$ and $r_v(n{=}M{+}1)$), with strengths $-\Gamma(1)$ and $\Gamma(M)$, respectively. Again, in the present model, the pitch of these tip vortices are analytically related to the pitch at the control points, whereas in the classical model, the pitch of these tip vortices is extrapolated. This extrapolation is known to cause numerical instability of the lifting line method.

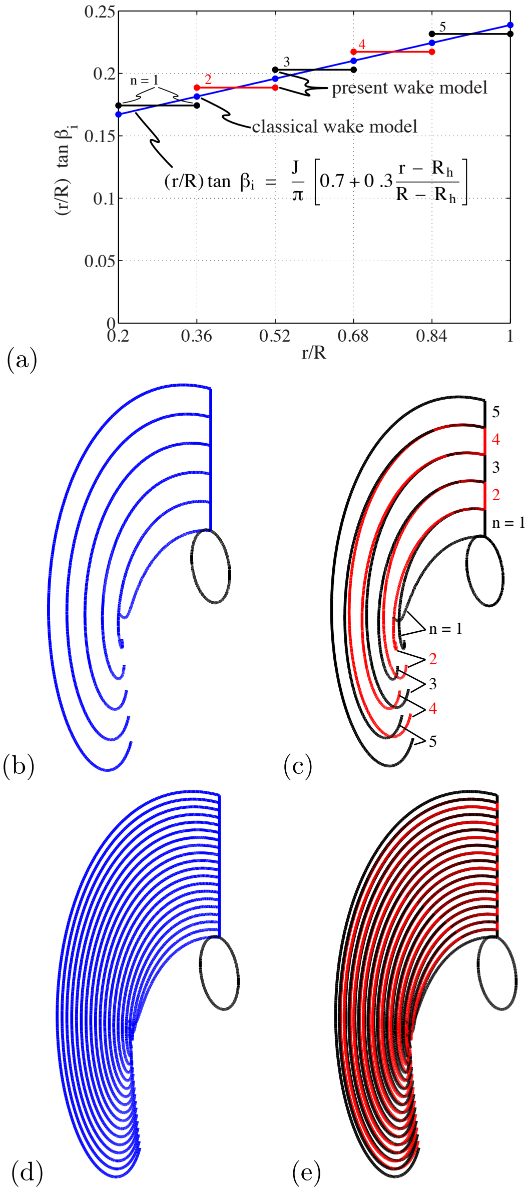

Figure 4: Example illustration of the key blade for the *classical* and *present* wake models: (a) Example linear wake pitch distribution (blue line), with the values used by each model marked by a '●'; (b) *Classical* wake model, with pitch corresponding to blue '●' in (a); (c) *Present* wake model (with adjacent panels plotted in black and red for clarity), with pitch corresponding to black '●' and red '●' in (a) ; (d) *Classical* wake model, with 19 vortex panels; (e) *Present* wake model, with 19 vortex panels.

Figure 4 further illustrates the two wake models. In this example, the wake pitch is chosen to vary linearly over the span, with a nominal advance coefficient achieved at the tip of $J = 0.75$:

$$(r/R)\tan\beta_i(r) = \frac{J}{\pi}\left[0.7 + 0.3\frac{r - R_h}{R - R_h}\right] \qquad (2.15)$$

Figure 4a shows this linear pitch distribution. Figure 4b illustrates the *classical* wake model with $M = 5$ vortex panels; since $r\tan\beta_i$ is linear in this example, evaluating $r_v \tan\beta_w$ in the classical model (2.14) simply reduces to evaluating (2.15) at each $r = r_v$, and these values are shown as '●' Figure 4a. Figure 4c illustrates the *present* wake model with $M = 5$ vortex panels, plotted in alternating colors for clarity. In the present model (2.13), $r\tan\beta_i$ is assumed constant across each panel, with the value computed by evaluating (2.15) at each $r = r_c$ (i.e. at the panel midpoint), and the corresponding $r_v \tan\beta_w$ values are shown as '●' and '●' in Figure 4a.

Figures 4d and 4e show the *classical* and *present* wake models with $M = 19$ vortex panels. With a small panel width, $\delta r_v = (R - R_h)/M$, the *present* wake model appears visually to represent a continuous vortex sheet, as in the classical model. In fact, the jump in wake pitch between adjacent panels is $\frac{\partial[r\tan\beta_i]}{\partial r}\delta r_v$, so as the number of panels increases, this discontinuity vanishes as $1/M$, and the wake appears as a smooth vortex sheet.

In reality, the near wake is a continuous vortex sheet, as opposed to a discrete lattice of horseshoe vortices. In modeling this continuous sheet with a discrete number of horseshoes, there is no physical reason to require a single vortex shed from the lifting line at any particular radius: Each constant-strength horseshoe satisfies Helmholtz' Laws independent of the other horseshoes. By setting pitch analytically and allowing the possibility of free vortices with differing pitch as described, the math embodied in the Lerbs/Wrench equations works out to be theoretically consistent with itself.

Free-running optimized propellers have $r\tan\beta_i$ nearly constant. In fact, the Betz optimum propeller is one for which $r\tan\beta_i = \text{constant}$, in which case the *classical* and *present* wake models coincide.

Figure 5 shows the induced velocities predicted by the *classical* and *present* wake models, for the $M = 5$ panel example shown in Figure 4. The velocities shown are those induced by the horseshoe panel spanning radii $r_v/R = 0.52$ to 0.68; for reference, the panel endpoint radii are marked by '×'. The control point radii are at the midpanel locations. This example shows that the choice of wake model has a small but noticeable effect on the predicted induced velocities, the contrast being larger for smaller radii. In particular, the axial induced velocities predicted by the classical model are larger in magnitude (more negative) for low radii than those predicted by the present wake model. For $\mathcal{Z} = 50$ (approximating $\mathcal{Z} = \infty$), the axial induced velocities are correctly evaluated to be zero in the present model and erroneously nonzero in the classical model. The tangential induced velocities also show slight model differences, although not as pronounced as the axial induced velocities.

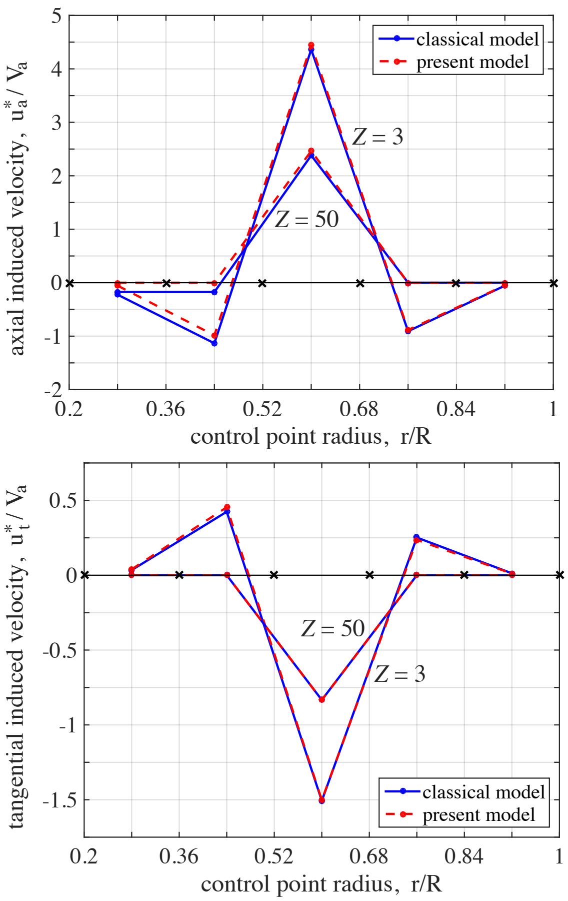

Figure 5: Normalized velocity (u_a^*/V_a, u_t^*/V_a) induced along a lifting line at discrete control point radii (r/R) by a set of $\mathcal{Z}$ horseshoes of strength $\Gamma = 2\pi R V_a/\mathcal{Z}$ spanning radii $r_v/R = 0.52$ to 0.68. The wake pitch is given in equation (2.15). '×' denote horseshoe endpoint radii r_v/R.

3 Analytical analysis

3.1 Infinite blade number case

In the *classical* wake model, the wake pitch angle is interpolated from the inflow angle at the control points. Using these interpolated values, the wake pitch angle is set via equation (2.14). This model implicitly assumes that there is one trailing vortex shed from each r_v location, as illustrated in Figure 1. Although this is physically realistic, it leads to a mathematical inconsistency between the intended wake pitch angle at a control point, $\beta_i(r_{c(n)})$, and the induction factors for the panel surrounding that point, $\bar{u}^*_{a(n,n)}$ and $\bar{u}^*_{t(n,n)}$. This inconsistency is most easily revealed by examining the case of an infinite-bladed rotor.

Careful examination of equations (2.9), (2.12), and (2.6) shows that for a large number of blades, the horseshoe influence matrices are nearly diagonal. That is, for a large number of blades, one can reasonably make the approximations

$$\begin{aligned}
\bar{u}^*_a(m,n) &\approx 0 \qquad (\text{for } m \neq n)\\
\bar{u}^*_t(m,n) &\approx 0 \qquad (\text{for } m \neq n)\\
u^*_a(n) &\approx \bar{u}^*_a(n,n)\,\Gamma(n)\\
u^*_t(n) &\approx \bar{u}^*_t(n,n)\,\Gamma(n)
\end{aligned} \tag{3.1}$$

In the limit of an infinite-bladed rotor, the singular terms S_1 and S_2 become zero, equations (3.1) become exact, and the induced velocities are simply the circumferential average velocities. Practically, $\mathcal{Z} = 50$ blades yields these results to $O(10^{-5})$ for any reasonably fine discretization (say $M > 10$ panels).

In the infinite-bladed case, equations (2.9), (2.12), and (2.6), along with the *classical wake model* (2.14) yields

$$\begin{aligned}
u^*_a(r_c(n)) = \bar{u}^*_a(n,n)\,\Gamma(n) &= \frac{\mathcal{Z}}{4\pi r_v(n+1)\tan\beta_v(n+1)}\,\Gamma(n)\\
u^*_t(r_c(n)) = \bar{u}^*_t(n,n)\,\Gamma(n) &= -\frac{\mathcal{Z}}{4\pi r_c(n)}\,\Gamma(n)
\end{aligned} \tag{3.2}$$

The actual circumferential average velocities at the propeller disk are as follows:

$$\begin{aligned}
\tilde{u}_a(r_c) &= \frac{\mathcal{Z}}{4\pi r_c \tan\beta_i(r_c)}\Gamma(r_c)\\
\tilde{u}_t(r_c) &= -\frac{\mathcal{Z}}{4\pi r_c}\Gamma(r_c)
\end{aligned} \tag{3.3}$$

(Hough and Ordway, 1964).

Comparing (3.2) and (3.3), it is evident that the circumferential average axial induced velocity predicted by the *classical wake model* is inconsistent with the analytically-derived circumferential mean velocity in (3.3) for all but the special case:

$$r_v(n+1)\tan\beta_v(n+1) = r_c(n)\tan\beta_i(n)$$

which is the case of a constant-pitch vortex sheet.

Using the present wake model (2.13), the influence functions in the infinite-bladed case are now analytically consistent with (3.3):

$$\begin{aligned}
u^*_a(r_c(n)) = \bar{u}^*_a(n,n)\,\Gamma(n) &= \frac{\mathcal{Z}}{4\pi r_c(n)\tan\beta_i(n)}\,\Gamma(n)\\
u^*_t(r_c(n)) = \bar{u}^*_t(n,n)\,\Gamma(n) &= -\frac{\mathcal{Z}}{4\pi r_c(n)}\,\Gamma(n)
\end{aligned} \tag{3.4}$$

The present wake model enforces this consistency by assuming that each horseshoe panel is a constant-pitch vortex sheet.

As will be shown in §3.2, (3.4) represents the circumferential average component of the horseshoe influence functions and is analytically consistent with (3.2) *for any number of blades.*

3.2 Finite blade number case

We have already shown that the *classical* wake model does not yield the correct circumferential average velocities for infinite blade number. We now consider the behavior of the *classical* and *present* wake models for finite blade number. Recall the notation that '0' and '1' denote the endpoints of panel n at $r_v(n)$ and $r_v(n+1)$, respectively. For convenience, define the notation

$$\begin{aligned}
y^{(0)} &\equiv y(r_c(m), r_v(n), \beta_w(n,0))\\
y^{(1)} &\equiv y(r_c(m), r_v(n+1), \beta_w(n,1))\\
S^{(0)}_k &\equiv S_k(r_c(m), r_v(n), \beta_w(n,0))\\
S^{(1)}_k &\equiv S_k(r_c(m), r_v(n+1), \beta_w(n,1))
\end{aligned}$$

where y and S_k, k=1,2, are auxiliary variables defined in (2.6c) and (2.6d), respectively.

The key difference between the two models is in the calculation of the auxiliary variable y in (2.6c), which is reproduced here:

$$y = \frac{r_c}{r_v \tan\beta_w} \tag{2.6c}$$

In the *classical* wake model

$$y^{(0)} = \frac{r_c(m)}{r_v(n)\tan\beta_v(n)} \neq \frac{r_c(m)}{r_v(n+1)\tan\beta_v(n+1)} = y^{(1)} \tag{3.5}$$

whereas in the *present* wake model

$$y^{(0)} = y^{(1)} = \frac{r_c(m)}{r_c(n)\tan\beta_i(n)} = y \tag{3.6}$$

To reiterate, for the *classical* model $y^{(0)} \neq y^{(1)}$, whereas for the *present* model $y^{(0)} = y^{(1)} = y$. This small inequality in (3.5) causes large errors in computing $\bar{u}^*_a(m,n)$ and $\bar{u}^*_t(m,n)$.

Consider computing $\bar{u}^*_a(m,n)$ and $\bar{u}^*_t(m,n)$ for two of the three possible cases: (A) $r_c(m) < r_v(n) < r_v(n+1)$; (B) $r_v(n) < r_c(n) < r_v(n+1)$; (C) $r_v(n) < r_v(n+1) < r_c(m)$. Since the conclusions drawn from Case C are similar to Case A, the details of Case C are left to the reader.

First consider Case A: $r_c(m) < r_v(n) < r_v(n+1)$. In this case, $\bar{u}^*_a(m,n)$ and $\bar{u}^*_t(m,n)$ are found by inserting (2.6a) into (2.12) twice as follows:

$$\begin{aligned}
\bar{u}^*_a(m,n) = & \left[\frac{\mathcal{Z}}{4\pi r_c}y^{(1)} - \frac{\mathcal{Z}S^{(1)}_1}{4\pi r_c}y^{(1)}\right]\\
& - \left[\frac{\mathcal{Z}}{4\pi r_c}y^{(0)} - \frac{\mathcal{Z}S^{(0)}_1}{4\pi r_c}y^{(0)}\right]\\
= & \frac{\mathcal{Z}}{4\pi r_c}\left[y^{(1)} - y^{(0)}\right] - \frac{\mathcal{Z}}{4\pi r_c}\left[S^{(1)}_1 y^{(1)} - S^{(0)}_1 y^{(0)}\right]
\end{aligned} \tag{3.7a}$$

$$\bar{u}^*_{t\,(m,n)} = \frac{\mathcal{Z}}{4\pi r_c}\left[S_1^{(1)} - S_1^{(0)}\right] \qquad (3.7\text{b})$$

As discussed in §3.1, the first term in (3.7a) is the circumferential average induced velocity, which should be zero according to (3.3). The second term is the difference of two large numbers, $S_1^{(1)}y^{(1)}$ and $S_1^{(0)}y^{(0)}$, which must be computed accurately for the difference to be small, as it should be. In the *classical* wake model, $y^{(0)} \neq y^{(1)}$, so the circumferential average velocity in (3.7a) is erroneously non-zero. Also, the discrepancy between $y^{(0)}$ and $y^{(1)}$ may cause errors in the difference $\left[S_1^{(1)}y^{(1)} - S_1^{(0)}y^{(0)}\right]$, thus leading to additional errors in $\bar{u}^*_{a\,(m,n)}$. By contrast, the *present* model has $y^{(0)} = y^{(1)} = y$, so the circumferential average velocity in (3.7a) is zero, as desired. Additionally, $\left[S_1^{(1)}y^{(1)} - S_1^{(0)}y^{(0)}\right] = \left[S_1^{(1)} - S_1^{(0)}\right] y$, and this computation is accurate as well.

Now consider Case B: $r_{v\,(n)} < r_{c\,(n)} < r_{v\,(n+1)}$.
In this case, $\bar{u}^*_{a\,(n,n)}$ and $\bar{u}^*_{t\,(n,n)}$ are found by inserting (2.6a) and (2.6b) into (2.12) as follows:

$$\begin{aligned} \bar{u}^*_{a\,(n,n)} = & \left[\frac{\mathcal{Z}}{4\pi r_c}y^{(1)} - \frac{\mathcal{Z}S_1^{(1)}}{4\pi r_c}y^{(1)}\right] \\ & - \left[-\frac{\mathcal{Z}S_2^{(0)}}{4\pi r_c}y^{(0)}\right] \\ = & \frac{\mathcal{Z}}{4\pi r_c}y^{(1)} - \frac{\mathcal{Z}}{4\pi r_c}\left[S_1^{(1)}y^{(1)} - S_2^{(0)}y^{(0)}\right] \qquad (3.8\text{a}) \end{aligned}$$

$$\bar{u}^*_{t\,(n,n)} = -\frac{\mathcal{Z}}{4\pi r_c} + \frac{\mathcal{Z}}{4\pi r_c}\left[S_1^{(1)} - S_2^{(0)}\right] \qquad (3.8\text{b})$$

Again, the first term in (3.8) is the circumferential average velocity, which should agree with (3.3) regardless of the number of blades. In the *classical* wake model,

$$y^{(1)} \neq \frac{r_{c\,(n)}}{r_{c\,(n)}\tan\beta_{i\,(n)}}$$

so the axial circumferential average velocity does not agree with (3.3) *for any arbitrary number of blades*. By contrast, the *present* model correctly yields $y^{(0)} = y^{(1)} = y$ $(= 1/\tan\beta_{i\,(n)}$ in Case B), which leads to

$$-\frac{\bar{u}^*_{t\,(n,n)}}{\bar{u}^*_{a\,(n,n)}} = \tan\beta_{i\,(n)}\ ,$$

in agreement with (3.3). As with Case A above, the singular terms are only accurately calculated with the *present wake model*.

This analysis shows that the present wake model more accurately agrees with the analytic representation of the rotor lifting line model.

4 Illustrative examples

The reader is assumed to be familiar with the lifting line design optimization problem, which is discussed at length in (Kerwin, Coney, and Hsin, 1986) or (Epps and Kimball, 2013b). For details, see also Appendix A.

4.1 Summary of codes used in this study

In order to systematically assess the wake model proposed in §2, the author produced four versions of a lifting line code (`OpenProp v3.1`) with one change made between each version. In this way, controlled numerical experiments were performed. Table 1 summarizes the codes used in this study: Both `PLL` and `OpenProp v2.4.4` employ the classical wake model and a numerical method based on a linearized system of equations. Four combinations of wake model (classical or present) and numerical method (linear system or Newton solver) are implemented in `OpenProp v3.1`, which can thus be used to perform controlled studies of the effect of wake model and/or numerical solution method. The only difference between <`OpenProp v3.1` with the *present wake model*> and <`OpenProp v3.1` with the *classical wake model*> is the equation used to evaluate β_w, (2.13) and (2.14) respectively.

As expected, the numerical method has no effect on the final solution. Three illustrative examples are given in this section based on DTRC propeller 4119: open-water case (§4.2), wake-adapted case (§4.3), and bollard pull case (§4.4). In all cases where both the linear solver and Newton solver converged, the final results were within the tolerance of the iterative scheme (considered converged when $|G_{current} - G_{last}|/G_{current} < 10^{-4}$). For example, in the open water case, the differences between the *linear system solver* and *Newton solver* results were on the order of 10^{-7} for G, 10^{-6} for each of u_a^*/V_s, u_t^*/V_s, and $(r/R)\tan\beta_i$, and 10^{-7} for each of K_T, $10K_Q$, and η. Since the choice of solver had no impact on the final results, the following three sections compare the results of the codes employing the linear system solver.

Table 1: Summary of design optimization methods. 'LL' stands for 'Lifting Line'. The `LL-Linear` and `LL-Newton` methods are described in further detail in (Epps and Kimball, 2013b), and the `OpenProp` source code is available online at (Epps and Kimball, 2013a).

Name	Wake model	Numerical method
`PLL`	Classical	Linear system
`OpenProp v2.4.4`	Classical	Linear system
`OpenProp v3.1`	Classical	Linear system
`OpenProp v3.1`	Classical	Newton solver
`OpenProp v3.1 LL-Linear`	Present	Linear system
`OpenProp v3.1 LL-Newton`	Present	Newton solver
Betz condition code	n/a	n/a

4.2 Propeller 4119 open water case

David Taylor propeller 4119 (designed by Denny (1968)) is a free-running propeller with a nearly ideal circulation distribution, fair blades, and moderate thickness. Its performance has been well characterized by many workers (Jessup, 1989, e.g.). Here, we design a 4119 replica, given the propeller 4119 design specifications: $\mathcal{Z} = 3$, $J_s = 0.833$, $K_T = 0.15$, $V_a/V_s = 1$, $V_t = 0$, and $M = 40$. Viscous forces are considered, with $C_D = 0.008$ and the blade outline given in Table 2.

Table 2: DTMB Propeller 4119 chord distribution.

r/R	0.2000	0.3000	0.4000	0.5000	0.6000
c/D	0.3200	0.3625	0.4048	0.4392	0.4610
r/R	0.7000	0.8000	0.9000	0.9500	1.0000
c/D	0.4622	0.4347	0.3613	0.2775	0.0020

The optimized circulation distribution is plotted in Figure 6. There is very good agreement between all optimization methods, validating the wake model developed herein and the new numerical methods implemented in `OpenProp v3.1`. They also agree with the experimental data of (Jessup, 1989) and the coupled lifting-surface/RANS analysis of (Kimball, 2001), which indicates that both the magnitude and distribution of circulation are correct.

As shown in Figure 6, the classical wake model (embodied in `PLL` and `OpenProp v2.4.4`) and the present wake model (embodied in `OpenProp v3.1`) yield nearly identical design optimization results for a moderately-loaded propeller in uniform inflow. This is expected, since the optimized wake pitch $r \tan \beta_i$ is nearly constant (see Figure 6c); in the case that the wake pitch is constant, the classical wake model and present wake model are identical.

Table 3 shows that the performance predictions (K_T, K_Q, and η) are also nearly identical, regardless of wake model or solver, as expected. These design optimizations converged in 6 to 7 iterations for all four codes, with no appreciable difference in run time.

Table 3: Open-water design case results.

Name	`OpenProp v2.4.4`	`OpenProp v3.1`	`OpenProp v3.1`	`OpenProp v3.1`
wake:	classical	classical	present	present
solver:	linear	linear	linear	Newton
J_s	0.833	0.833	0.833	0.833
K_T	0.15	0.15	0.15	0.15
$10K_Q$	0.2825	0.2827	0.2829	0.2829
η	0.7041	0.7033	0.7030	0.7030

As a check of convergence versus radial discretization, two codes were run for various number of panels: $M = 8$, 12, 16, 20, and 40. The (`OpenProp v3.1`, classical wake model, linear solver) code and the (`OpenProp v3.1`, present wake model, linear solver) code exhibited similar numerical convergence versus panel number for both the circulation distribution and efficiency. The change in efficiency between $M = 16$ and 20 panels was 0.0001, and the root mean square difference in circulation distribution was almost nil. These convergence results were similar for both this open-water case and the following wake-adapted case (§4.3).

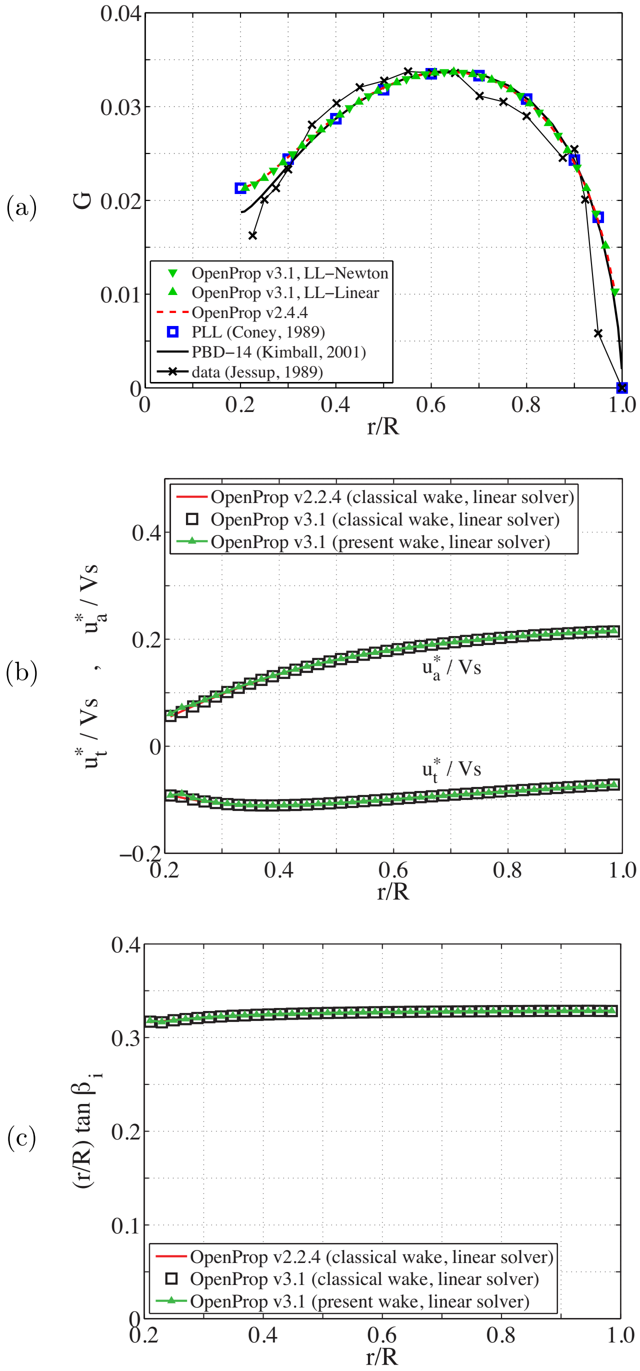

Figure 6: Optimized circulation distribution for a DTMB propeller 4119 replica, $G = \Gamma/(2\pi R V_s)$.

4.3 Wake adapted case

Consider now a notional wake-adapted propeller design. The assumed wake profile $V_a(r)$ is that from (Laskos, 2010) and is representative of a single-screw ship (see Figure 7). For consistency with the open water case, the propeller 4119 design specifications are carried over from §4.2; the only difference between this wake-adapted design and the open-water design is the assumed axial inflow $V_a(r)$.

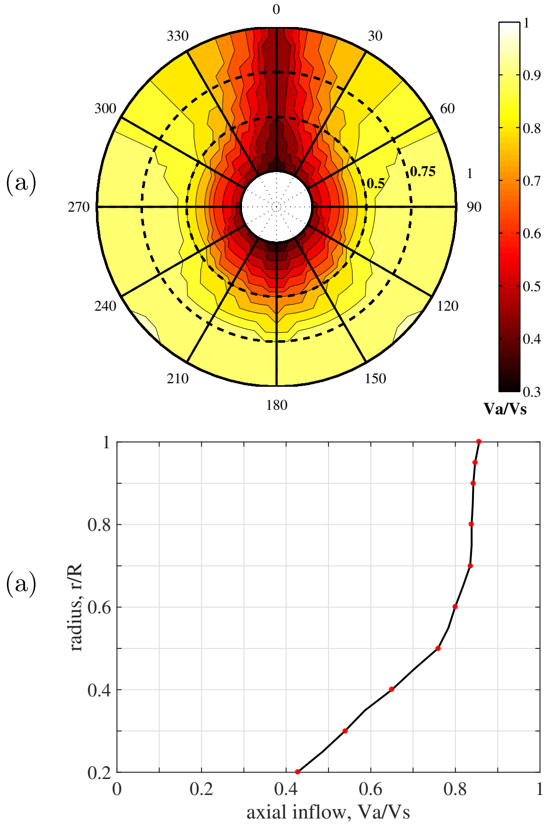

Figure 7: Axial inflow used for the wake adapted design case (wake data from (Laskos, 2010)): (a) polar plot $V_a(r, \theta)$; (b) circumferential-average $V_a(r)$.

Table 4: Wake-adapted design case converged results.

Name wake: solver:	OpenProp v3.1 classical linear	OpenProp v3.1 present linear
J_s	0.833	0.833
J_a	0.6527	0.6527
K_T	0.15	0.15
$10K_Q$	0.2393	0.2408
η	0.6512	0.6471

Table 4 and Figure 8 shows the converged design results. Since the linear solver and Newton solver yielded nearly identical results, only the linear solver results are shown.

Figure 8c shows that the wake pitch varies radially between $0.2 < r/R < 0.6$ (and is approximately constant for $r/R > 0.6$). As a result, the "optimized" circulation distribution and induced velocities are different depending on choice of wake model. The predicted efficiencies given in Table 4 are within 0.004, which is within the uncertainty of the lifting line model (and most experiments!).

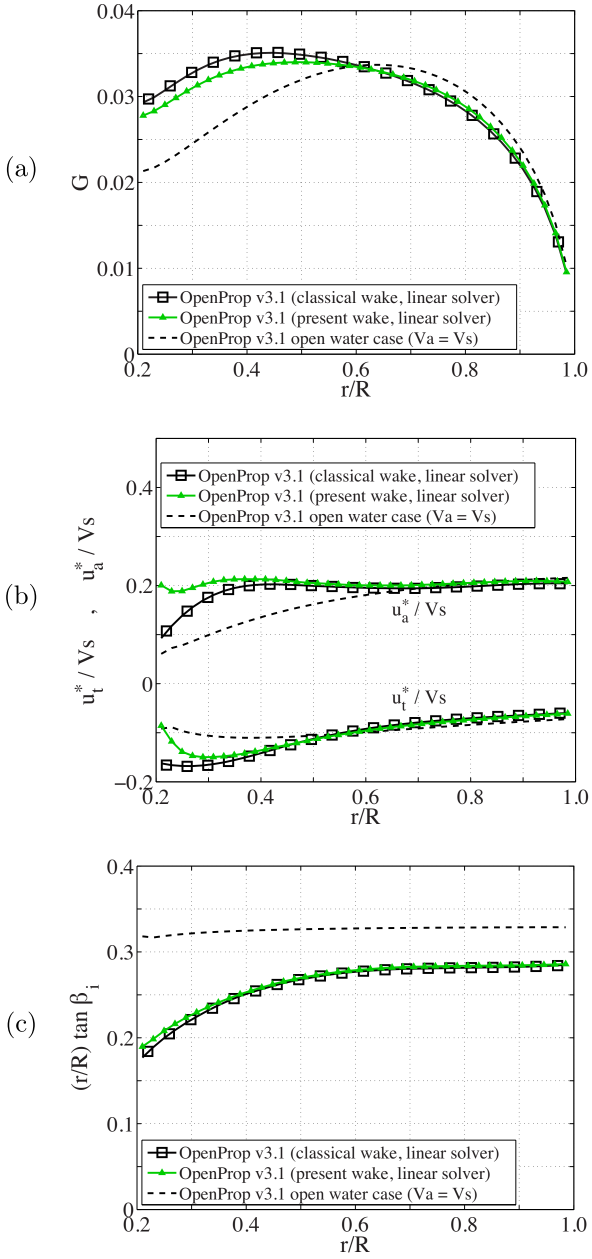

Figure 8: Wake adapted design case: (a) circulation, G; (b) induced velocities, u_a^*/V_s and u_t^*/V_s; and (c) wake pitch $(r/R)\tan\beta_i$.

The results in Figure 8 prompt the question: *In the limit of infinite span-wise panels, should the present and classical method (numerically) converge to the same result?* This question is difficult to answer, because the discretized lifting line model is only valid for finite panel number, M, by definition. Moreover, any discrete lifting line design optimization code will crash for a large enough M. The reason is that the numerics are very sensitive to round off error. Recall, computation of the horseshoe influence functions in (3.7) or (3.8) involves taking the difference of two very large numbers, S_1 and S_2. As M grows, so to do S_1 and S_2, and for large enough M, the computer can not accurately compute their difference. Unfortunately, the critical M is somewhat low, say 20-40 panels. In Figure 8, the classical wake model results are shown with $M = 20$ panels (and that code crashes with $M = 40$). Similarly, the present wake model results are shown with $M = 40$ panels (and that code crashes with $M = 80$). Thus, it is impossible to see numerically what would happen with, say, $M = 1000$ panels.

One might instead ask: *Are each of the curves in Figure 8 converged with panel number, M?* To answer this question, consider the following spatial resolution convergence results. Let $G_{c,20}$ be the circulation distribution $G(r)$ for the classical model with $M = 20$, and let $G_{p,20}$ be that with the present wake model. Let `rms` be the root mean square and `pchip` be the piecewise cubic Hermite interpolating polynomial interpolation function. Some numerical results are as follows: $\texttt{rms}(G_{c,20} - G_{p,20})/\texttt{rms}(G_{p,20}) = 0.0327$, and $\texttt{rms}(G_{p,20} - \texttt{pchip}(r_{p,40}, G_{p,40}, r_{p,20}))/\texttt{rms}(G_{p,20}) = 0.0032$. These calculations show that the difference between the classical and present wake models (3.27%) is about 10 times larger than the difference between present wake model with $M = 20$ and 40 panels (0.32%). Thus, it is reasonable to conclude that the results in Figure 8 are converged with panel number and that the difference between the circulation distributions is a real effect of the choice of wake model.

In a design optimization problem resulting in a constant wake pitch, $r \tan \beta_i$ (e.g. the open-water case in §4.2), the present and classical wake models yield identical results (see Figure 7). In design optimization problems resulting in a radially-varying wake pitch (e.g. this wake-adapted case), the present and classical wake models yield different horseshoe influence functions. Consequently, the results from the present and classical codes are different (see Figure 8). In particular, Figure 8 shows that the circulation and induced velocities differ appreciably over the part of the span where the wake pitch varies appreciably.

4.4 Bollard pull case

We now consider the bollard pull design condition (zero advance speed, $V_a(r) = 0$). This example is purely illustrative, since it is well accepted that the *moderately-loaded lifting line model* does not apply in the case of a heavily-loaded propeller at bollard pull. In the heavily-loaded case, the propeller wake is known to contract considerably and roll up, two effects that are not captured in the moderately-loaded lifting line model discussed herein. In the extreme case of zero advance speed, $V_a(r) = 0$, codes employing the classical wake model break down due to numerical instabilities, but codes employing the present wake model converge. This is due to the treatment of the singular terms in equation (2.6), as discussed in §3.

The propeller 4119 example is again used for consistency with §4.2 and §4.3. Since we had set $V_s = 1$ m/s and $D = 1$ m in the open water case (§4.2), the only modification to the design input was setting the axial inflow to zero[§] , $V_a/V_s = 0$.

For this example, six codes (all five `OpenProp` codes and the Betz condition code) were each run twice (once with each of the initialization methods (A.17) or (A.18)). Regardless of the initialization method, the end result was the same: `OpenProp v2.4.4` crashed, as it employs the unstable 'classical' wake model. In addition, both `OpenProp v3.1` codes employing the 'classical' wake model crashed as well. However, both `OpenProp v3.1` codes employing the 'present' wake model successfully converged. To reiterate, the only difference between `OpenProp v3.1` crashing or converging was the wake model used to compute $\bar{u}_a^*$ and $\bar{u}_t^*$. The Betz condition explicitly sets $r \tan \beta_i$ = constant, so the classical wake model and present wake model are mathematically identical in that code.

Table 5: Bollard pull design case converged results.

Name	`OpenProp v3.1`	`OpenProp v3.1`	`Betz`
	'LL-Linear'	'LL-Newton'	
wake:	present	present	n/a
solver:	linear	Newton	n/a
J_s	0.833	0.833	0.833
J_a	0	0	0
K_T	0.15	0.15	0.15
$10K_Q$	0.1116	0.1116	0.1123
η	0	0	0
$\mathcal{QF}$	0.6611	0.6611	0.6568
iterations	49	17	5

The results of the three converged codes are shown in Table 5 and Figure 9. The results were considered converged when the fractional change in non-dimensional circulation between iterations was less than 10^{-4}: $(|G_{current} - G_{last}|/G_{current} < 10^{-4})$. The use of the

[§] Typically, the 'reference speed' V_s used for non-dimensionalization is the ship speed (as indicated by the underscore 's'). However, in the bollard pull case, the ship speed is zero by definition. Numerically, one can model the zero-ship-speed case by setting $V_s = 1$ m/s and $V_a/V_s = 0$, where V_a is the propeller advance speed.

bollard pull initialization (A.18) in lieu of open-water initialization (A.17) had no effect on either the numerical stability of any code, nor on the final converged result (up to the fourth significant digit). Likewise, the numerical method used ('linear system' or 'Newton solver') did not affect the converged result (up to the fourth significant digit).

Figure 9 shows the converged results for both the open water ($V_a/V_s = 1$) and bollard pull ($V_a/V_s = 0$) cases. The two solution methods, 'LL-Linear' and 'LL-Newton', converged to the same circulation, induced velocities, and wake pitch.

The optimized circulation distribution is shown in Figure 9a. In comparison to the open water case, the bollard pull case shows a flatter, more broad circulation distribution, with circulation carried further out towards the propeller tip. This is consistent with actuator disk theory, which prescribes constant circulation. Comparing the induced velocities shown in Figure 9b, we find that the tangential induced velocity is similar in magnitude between the open water ('- -') and bollard pull ('-●-') cases, but there is less tangential induced velocity near the tip for the bollard pull case. This can be understood as follows: since the thrust scales by $\int(\omega r + u_t^*)\Gamma\, dr$, so an increase of Γ near the tip must accompany a decrease of u_t^* to maintain the same thrust.

Another quantity of interest, shown in Figure 9c, is $(r/R)\tan(\beta_i)$. According to the Betz condition, this should be constant, such that the wake forms a constant-pitch helical vortex sheet. Interestingly, $(r/R)\tan\beta_i \approx$ *constant* for both the open water and bollard pull cases.

The nature of the optimization algorithm was further investigated by considering a hub-less propeller. Assuming no hub present, the 'LL-Newton' code was re-run; the resulting induced velocities and wake pitch were nearly identical to those when a hub was present, as shown in Figures 9(b) and 9(c). However, the hubless circulation distribution tended to zero at the hub ($r/R = 0.2$), contrary to that of the hubbed propeller. Thus, this hubless circulation distribution was simply what was needed to produce the same induced velocities as the hubbed case. This demonstrates that the propeller optimization equations (A.14) are actually optimizing the induced velocity, not the circulation. Circulation is just a convenient choice of independent variable for the derivation and solution of the mathematical optimization equations.

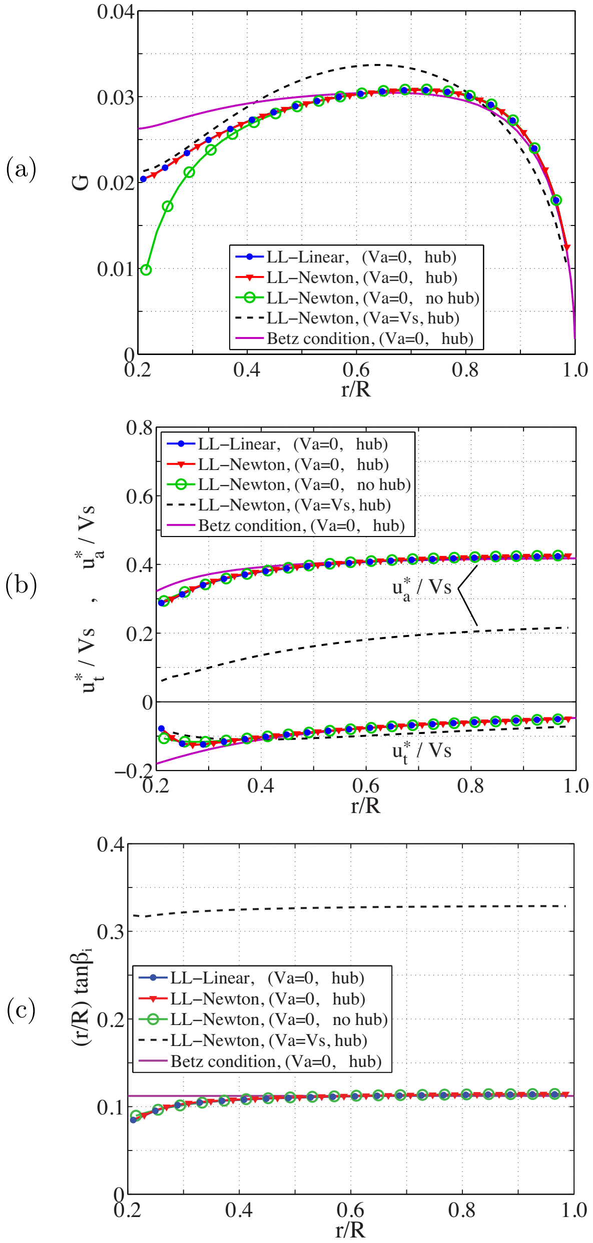

Figure 9: Bollard pull design case: (a) circulation, G; (b) induced velocities, u_a^*/V_s and u_t^*/V_s; and (c) wake pitch $(r/R)\tan\beta_i$.

4.5 Horizontal-axis turbine design

Epps and Kimball (2013b) created a unified rotor lifting line theory for the design of either propellers or horizontal-axis turbines. In that work, a parametric design study was performed in which several turbines were optimized to maximize power extraction at each of a range of tip-speed ratios. The study considered both infinite-bladed ($\mathcal{Z} = 100$) and finite-bladed ($\mathcal{Z} = 3$) cases, assuming uniform inflow ($V_a/V_s = 1$, $V_t = 0$), inviscid flow ($C_D = 0$), $M = 80$ panels, and a blade root radius of $R_{hub}/R = 0.005$. No hub image was used.

Figure 10 shows power coefficient versus tip-speed ratio for each of the optimized turbines. Momentum theory is shown as the solid line, and the well-known Betz Limit of $C_P = 16/27 \approx 0.5926$ is reached in the limit of infinite tip speed ratio, λ. The off-design performance of the propellers designed for $\lambda = 2$, 5, and 8 is also shown. As expected, for any given tip speed ratio (e.g. $\lambda = 3$), the off-design performance curves (solid lines) never exceed the performance of the turbine optimized for that tip speed ratio (dashed lines). For Figure 10, the present wake model was used.

If the classical wake model were used, then the off-design performance curves (solid lines) would cross the optimized design frontier (dashed lines). In other words, the code would claim that some turbine operating at some off-design λ could have a larger power coefficient than the turbine specifically optimized for that λ. This violates the premise of the design optimization method, which is shown in Figure 10 to be in agreement with momentum theory. This self-inconsistency is a direct result of the wake model. In contrast, the present wake model yields self-consistent design optimization and off-design performance analysis.

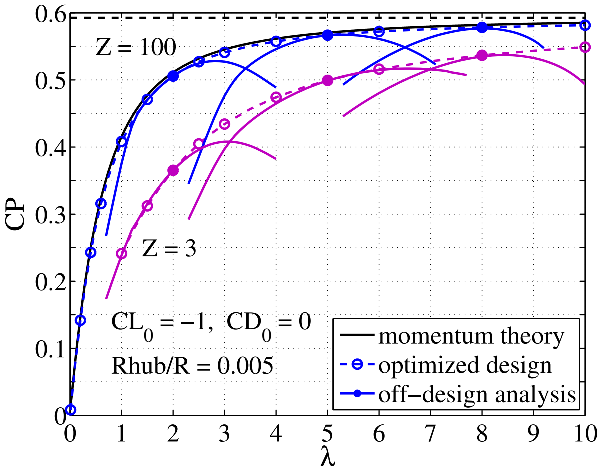

Figure 10: Power coefficient C_P of several turbines at their design point '- -○- -', as well as the off-design performance of selected turbines '–●–'. Tip speed ratio is defined as $\lambda = \omega R/V_s$.

5 Summary

This paper reconsiders the classical wake model typically used in moderately-loaded propeller lifting line theory (§2). Through analytical analysis, it is shown that the classical wake model is theoretically and numerically self-inconsistent (§3). Further analysis shows that a self-consistent wake model can be created, by relating the pitch of the trailing vortex filaments to the pitch of the total inflow computed at the lifting line control points; this leads to the present wake model described in (§2.1.2). Illustrative examples (§4) show that the 'classical' and 'present' wake models yield nearly identical optimized propellers when the wake pitch is nearly constant over the span, as expected. However, for the case of a non-uniform inflow, the 'classical' and 'present' wake models produce different optimized circulation distributions, with the present wake model suggesting a small reduction in hub circulation; efficiency for these two designs is within 0.004, making it difficult to distinguish which design might perform better. In the extreme example of the bollard pull design case (zero forward speed), the classical wake model causes the propeller optimization code to crash, while the present wake model provides enhanced numerical stability leading to an optimum bollard pull thruster design. Horizontal-axis turbine designs are presented, showing that the present wake model leads to a self-consistent framework for design optimization and off-design performance analysis, whereas the classical wake model produces self-inconsistencies in the model. In summary, the present wake model is theoretically self-consistent and numerically more stable than the classical wake model, thus enabling propeller design over a wider range of scenarios and enabling application of rotor lifting line theory to horizontal axis turbine design.

Nomenclature

Symbol	Description
α	angle of attack (rad)
α_i	ideal angle of attack (rad)
β	free-stream inflow angle (rad)
β_i	total inflow angle (rad)
β_w	wake pitch angle (rad)
ρ	fluid density (kg/m^3)
θ	section pitch angle (rad)
$\omega = 2\pi n$	rotation speed (rad/s)
Γ	circulation (m^2/s)
c	section chord length (m)
n	rotation speed (rev/s)
r	radial coordinate (m)
r_c	control point radius (m)
r_v	vortex point radius (m)
u_a^*	axial induced velocity (m/s)
u_t^*	tangential induced velocity (m/s)
$\bar{u}_a^*$	axial horseshoe influence function (1/m)
$\bar{u}_t^*$	tangential horseshoe influence function (1/m)
$\tilde{u}_a$	circumferential average axial induced velocity (m/s)
$\tilde{u}_t$	circumferential average tangential induced velocity (m/s)
$\widetilde{V}_a$	volumetric mean inflow velocity (m/s)
$D = 2R$	rotor diameter (m)
$F_i = \rho V^* \Gamma$	inviscid lift force per unit span (N/m)
$F_v = \frac{1}{2}\rho (V^*)^2 C_D c$	viscous drag force per unit span (N/m)
Q	torque (N-m)
R	rotor radius (m)
R_h	hub radius (m)
T	thrust (N)
V_a	axial inflow speed (m/s)
V_s	ship/wind speed (m/s)
V_t	tangential inflow speed (m/s)
V^*	total inflow speed (m/s)
$\eta = \dfrac{T\widetilde{V}_a}{2\pi n Q}$	wake-adapted efficiency
C_D	section drag coefficient
$C_L = \dfrac{2\,\Gamma}{V^* c}$	section lift coefficient
$G = \dfrac{\Gamma}{2\pi R V_s}$	circulation
$J_s = \dfrac{V_s}{nD}$	advance coefficient
$J_a = \dfrac{\widetilde{V}_a}{nD}$	wake-adapted adv. coeff.
$K_Q = \dfrac{Q}{\rho n^2 D^5}$	torque coefficient
$K_T = \dfrac{T}{\rho n^2 D^4}$	thrust coefficient
M	number of panels
$\mathcal{QF}$	quality factor
$\mathcal{Z}$	number of blades

Acknowledgments

The author would like to thank the anonymous reviewers for their constructive suggestions that helped improve this manuscript.

A Rotor lifting-line design optimization

A.1 Loads, efficiency, and quality factor

The *thrust* and *torque* acting on the rotor are

$$T = \mathcal{Z} \int_{R_h}^{R} [F_i \cos\beta_i - F_v \sin\beta_i] dr \qquad (\mathbf{e_a}) \tag{A.1}$$

$$Q = \mathcal{Z} \int_{R_h}^{R} [F_i \sin\beta_i + F_v \cos\beta_i] r dr \qquad (-\mathbf{e_a}) \tag{A.2}$$

where R_h and R are the hub and tip radii.

The *efficiency* of a wake-adapted propeller is defined as

$$\eta = \frac{T\widetilde{V}_a}{Q\omega} = \frac{J_a}{2\pi}\frac{K_T}{K_Q} \tag{A.3}$$

where

$$\widetilde{V}_a = \frac{1}{\pi R^2 - \pi R_h^2} \int_{R_h}^{R} V_a(r)\, 2\pi r\, dr \tag{A.4}$$

is the volumetric mean inflow velocity (which is related to the average wake fraction). In the case of bollard pull, $V_a = 0$, $\widetilde{V}_a = 0$, and thus $\eta = 0$, so another metric must be used to quantify propeller performance. That metric is the quality factor.

The *quality factor* is defined as the ratio of the propeller efficiency to the theoretically-achievable ideal efficiency:

$$\mathcal{QF} = \frac{\eta}{\eta_i} \tag{A.5}$$

(Woud and Stapersma, 2008). The ideal efficiency of an actuator disk in uniform inflow of speed $V_a = \widetilde{V}_a$ is

$$\eta_i = \frac{2}{1 + \sqrt{1 + C_{T_a}}} = \frac{2}{1 + \sqrt{1 + \dfrac{8K_T}{\pi J_a^2}}} \tag{A.6}$$

where $C_{T_a} = T/(\frac{1}{2}\rho \widetilde{V}_a^2 \pi R^2)$ is the thrust coefficient based on the volumetric mean inflow velocity. Combining (A.3) and (A.6) with (A.5), the quality factor can be expressed as

$$\mathcal{QF} = \frac{1}{2\pi}\frac{K_T}{K_Q}\frac{J_a + \sqrt{J_a^2 + \dfrac{8K_T}{\pi}}}{2} \tag{A.7}$$

In the case of bollard pull ($J_a = 0$), the quality factor remains non-zero

$$\mathcal{QF}(J_a = 0) = \frac{1}{2\pi}\frac{K_T}{K_Q}\frac{\sqrt{\dfrac{8K_T}{\pi}}}{2} \tag{A.8}$$

Thus, the quality factor is used to quantify the performance of a propeller at zero or near-zero speeds of advance.

A.2 Propeller design optimization

As reviewed in (Epps and Kimball, 2013b), the fundamental propeller design problem is to find the optimum circulation distribution, Γ, for a given inflow (V_a, V_t) and blade properties ($\mathcal{Z}$, R, ω, c, C_D), such that efficiency (or quality factor) is maximized for a prescribed thrust. Since a wake-alignment procedure can be used to determine $\{V^*, u_a^*, u_t^*, \beta_i, \bar{u}_a^*, \bar{u}_t^*\}$ and ultimately $\{T, Q, \eta, \mathcal{QF}\}$ for a given Γ, this design optimization problem is to simply find the optimum Γ distribution. Additional features may be added, such as chord length optimization, which adds a layer of complexity to the optimization problem but does not change the fundamental design problem. Also, systematic evaluation of the design space can be performed by varying ($\mathcal{Z}$, R, ω, c, C_D) and solving the mathematical optimization problem for each case.

As part of this investigation, several Γ optimization strategies were studied, including a random searching method, a method enforcing the *Lerbs criterion*, a method enforcing the *Betz condition*, and a *Lagrange multiplier method.*

A.2.1 Searching methods

Perhaps the most straightforward optimization method is to consider candidate circulation distributions that can be formed by a cubic B-spline, which are defined by a small set of B-spline amplitudes. The B-spline amplitudes are adjusted iteratively, and given the resulting circulation distribution, the wake is aligned and efficiency (or quality factor) evaluated. Adjustments that increase efficiency are kept, while adjustments that decrease efficiency are discarded. As it turns out, this method results in a physically-unrealistic, tip-loaded circulation distribution (with little circulation over most of the span and a large loading near the tip), which has been previously observed by other workers (Kerwin et al, 1986; Brockett and Korpus, 1986, e.g.).

A.2.2 Lerbs Criterion

The *Lerbs criterion* prescribes that the optimum propeller satisfies (Lerbs, 1952):

$$\frac{\tan\beta(r)}{\tan\beta_i(r)} = \text{constant} \cdot \sqrt{\frac{V_a(r)}{V_s}} \tag{A.9}$$

Since $\tan\beta = V_a/(\omega r + V_t)$, (A.9) is equivalent to

$$\tan\beta_i = \frac{V_a/(\omega r + V_t)}{\text{constant} \cdot \sqrt{V_a/V_s}} \sim \frac{\sqrt{V_a}}{\omega r + V_t}$$

This method fails in the bollard pull case ($V_a = 0$), since it prescribes $\tan\beta_i = 0$, which requires zero induced velocities and thus zero circulation. Strictly speaking, the Lerbs criterion (A.9) is invalid at bollard pull anyway, since it assumes a lightly loaded propeller $u_a^* \ll V_s$, which is not true in the bollard pull case.

A.2.3 Betz condition

In the case of uniform inflow ($V_a(r) = \text{constant}$), the 'Lerbs criterion' reduces to the 'Betz condition'

$$r \tan \beta_i(r) = \text{constant} \tag{A.10}$$

Strictly speaking, the Betz condition is only valid for *lightly loaded propellers*, but nevertheless, consider (A.10) in the limit $V_a \to 0$. Definition (2.1) can be rearranged as follows (in the case $V_a = V_t = 0$)

$$u_a^* - u_t^* \tan \beta_i = (\omega r) \tan \beta_i \tag{A.11}$$

Taking right hand side as constant by (A.10), and upon inserting (2.9), we arrive at a system of optimization equations

$$\sum_{n=1}^{M} (\bar{u}_a^*{}_{(m,n)} - \bar{u}_t^*{}_{(m,n)} \tan \beta_{i(m)})\, \Gamma_{(n)} = \text{constant} \tag{A.12}$$

for ${\scriptstyle m=1\ldots M}$. This forms a system of M equations for the M unknown $\Gamma_{(m)}$, which can be solved for any given value of the constant on the right hand side (which determines the thrust loading). The solver can iteratively choose the constant, solve (A.12) and 'align the wake' by updating $\{u_a^*, u_t^*, \beta_i\}$ until all state variables have converged to some specified tolerance and the thrust matches the prescribed thrust. Although this method technically is not proven to yield optimized propellers at bollard pull, it serves as a baseline reference for the Lagrange multiplier method.

A.2.4 Lagrange multiplier method

The Lagrange multiplier method described herein follows that of (Kerwin, Coney, and Hsin, 1986), which was previously implemented in codes such as `PLL` (Coney, 1989) and `OpenProp v2.4.4` (Epps, 2010b). The problem is to find the set of M vortex panel circulation values that produce the least torque for a specified thrust, $T = T_s$. The method is to form an auxiliary function, $H = Q + \lambda_1(T - T_s)$, where λ_1 is a Lagrange multiplier, and to find the optimum Γ by setting the partial derivatives of H to zero

$$\frac{\partial H}{\partial \Gamma_{(i)}} = 0, \quad \frac{\partial H}{\partial \lambda_1} = 0 \tag{A.13}$$

In the inviscid case, this system of equations reduces to

$$\begin{aligned}
&\sum_m \begin{pmatrix} \bar{u}_a^*{}_{(m,i)}\, r_{c(m)} \triangle r_{v(m)} \\ + \bar{u}_a^*{}_{(i,m)}\, r_{c(i)} \triangle r_{v(i)} \\ + \lambda_1 \bar{u}_t^*{}_{(m,i)}\, \triangle r_{v(m)} \\ + \lambda_1 \bar{u}_t^*{}_{(i,m)}\, \triangle r_{v(i)} \end{pmatrix} \Gamma_{(m)} \\
&\quad + \Big([\omega r_{c(i)} + V_{t(i)}] \triangle r_{v(i)} \Big) \lambda_1 \\
&\qquad = -V_{a(i)} r_{c(i)} \triangle r_{v(i)} \qquad \text{for} \;\; i = 1 \ldots M
\end{aligned}$$

$$\sum_m \Big([\omega\, r_{c(m)} + V_{t(m)} + u_t^*{}_{(m)}] \triangle r_{v(m)} \Big) \Gamma_{(m)} = T_s / (\rho \mathcal{Z}) \tag{A.14}$$

This is a system of $M + 1$ non-linear equations for as many unknowns $\{\Gamma_{(m=1\ldots M)},\ \lambda_1\}$, which can be solved iteratively. Each iteration consists of solving (A.14) to update $\{\Gamma,\ \lambda_1\}$ and then updating $\{u_a^*,\ u_t^*,\ \tan\beta_i,\ \bar{u}_a^*,\ \bar{u}_t^*\}$ to 'align the wake'. Iteration finishes with a converged set of values that describe the design state.

A.3 Solution methods

Equations (A.14) can be solved in a number of ways. Coney (1989) linearizes (A.14) into a linearized system of the form $[A] \cdot [\Gamma; \lambda_1] = [B]$, as suggested by the presentation of (A.14). He then iteratively solves for $\{\Gamma,\ \lambda_1\}$ and then 'aligns the wake' by updating the other parameters $\{u_a^*,\ u_t^*,\ \tan\beta_i,\ \bar{u}_a^*,\ \bar{u}_t^*\}$. The new values of these parameters are then fed back into (A.14), and the process is repeated. Let us call this numerical method the 'linear system' solver. This method is used in MIT code `PLL`, which employs the classical wake model.

This 'linear system' solution procedure is tenuous, since it is prone to crash if the induced velocities do not vary smoothly over the span. Epps et al (2009b) overcame this difficulty by smoothing the induced velocities at the blade root and tip between each solver iteration. His method is coded in `OpenProp v2.4.4`, which employs the classical wake model and the linear system solver.

For the purposes of this investigation, it was desired to create an algorithm that did not require the smoothing employed in `OpenProp v2.4.4`. As such, `OpenProp v3.1`, does *not* smooth the induced velocities between solver iterations. Thus, the only difference between `OpenProp v2.4.4` and <`OpenProp v3.1` with the *classical wake model* and *linear system solver*> is in whether or not u_a^* and u_t^* are smoothed before evaluating $\{\tan\beta_{i(r)},\ \bar{u}_a^*,\ \bar{u}_t^*\}$.

The main difficulty with the 'linear system' solution procedure is that equations (A.14) are unstable and depend very sensitively on $\bar{u}_a^*$ and $\bar{u}_t^*$, which are very sensitive to the inflow angle β_i. If $\tan\beta_{i(r)}$ is not smooth in r, $\bar{u}_a^*$ and $\bar{u}_t^*$ amplify this irregularity, (A.14) results in a $\Gamma_{(r)}$ distribution that is not smooth in r, and the procedure breaks down: An irregular $\Gamma_{(r)}$ yields irregular $u_a^*{}_{(r)}$ and $u_t^*{}_{(r)}$, which yields an irregular $\tan\beta_{i(r)}$, and so on until the code crashes.

A more robust approach for solving (A.14) is to use a *Newton solver*. In the `OpenProp v3.1` Newton solver formulation, $\{\Gamma,\ u_a^*,\ u_t^*,\ \tan\beta_i,\ \lambda_1\}$ are taken as the vector of unknowns updated by the Newton solver, and $\{\bar{u}_a^*,\ \bar{u}_t^*\}$ are updated between solver iterations (Epps and Kimball, 2013b). As with the 'linear system' method, the 'Newton solver' method can be used with either wake model.

A.4 Initialization of the numerical procedure

These optimization methods require initialization of flow parameters $\{V^*, u_a^*, u_t^*, \beta_i\}$ and horseshoe influence functions $\{\bar{u}_a^*, \bar{u}_t^*\}$ before equation (A.12) or (A.14) can be solved. A simple initialization method, which works well for lightly-loaded propellers ($u_a^* \ll V_a$), is to ignore the induced velocities and initialize as follows $\{u_a^* = 0, u_t^* = 0\}$ from which $V^* = V_0$ and $\beta_i = \beta$. However, in the bollard pull design case, $V_a = 0$, and thus $\beta = 0$, rendering this method unusable (since $\bar{u}_a^* \to \infty$ as $\beta_i \to 0$).

For any advance speed V_a, a better estimate of u_a^* can be obtained from actuator disk theory. The result is well known and will not be re-derived here. From Bernoulli's equation, the thrust produced by the actuator disk is (Kerwin and Hadler, 2010, eqn. (4.15))

$$T = \rho u_w \left(V_a + \frac{u_w}{2} \right) \pi R^2 \qquad \text{(A.15)}$$

where u_w is the axial perturbation velocity far downstream of the disk, which is related to the induced velocity at the disk u_a^* by $u_w = 2\,u_a^*$. Thus, after non-dimensionalizing by the 'reference speed' V_s, we have

$$C_T = \frac{\rho(2u_a^*)\,(V_a + u_a^*)\,\pi R^2}{\frac{1}{2}\rho V_s^2 \pi R^2} = 4\frac{u_a^*}{V_s}\left(\frac{V_a}{V_s} + \frac{u_a^*}{V_s} \right) \qquad \text{(A.16)}$$

In the *open-water case* ($V_a = V_s$), equation (A.16) can be solved by quadratic formula to yield the initialization

$$\frac{u_a^*}{V_s} = \frac{-1 + \sqrt{1 + C_T}}{2} \qquad \text{(A.17a)}$$

$$\frac{u_t^*}{V_s} = 0 \qquad \text{(A.17b)}$$

where the tangential induced velocity is initialized to zero, since it is assumed small ($u_t^* \ll V_a$).

In the *bollard pull case* ($V_a = 0$), equation (A.16) simplifies to

$$\frac{u_a^*}{V_s} = \frac{\sqrt{C_T}}{2} \qquad \text{(A.18a)}$$

$$\frac{u_t^*}{V_s} = 0 \qquad \text{(A.18b)}$$

where V_s is an arbitrary reference speed (say, 1 m/s). The tangential induced velocity is again initialized to zero, such that the wake pitch is initialized as constant

$$r \tan \beta_i = r \frac{u_a^*}{\omega r + u_t^*} \to \frac{u_a^*}{\omega} = \text{constant}$$

With C_T given as one of the design specifications (or computed from T_s, V_s, and R), equation (A.17) or (A.18) can be used to initialize u_a^* and u_t^*. In both cases, $\{V^*, \tan \beta_i, \bar{u}_a^*, \bar{u}_t^*\}$ are then initialized by definition.

References

Arán DM, Kinnas S (2013) Optimization and preliminary design of marine current turbines. In: Proceedings of the 18th Offshore Symposium, SNAME, Houston, TX

Bellingham JG, Zhang Y, Kerwin JE, Erikson J, Hobson B, Kieft B, Godin M, McEwen R, Hoover T, Paul J, Hamilton A, Franklin J, Banka A (2010) Efficient propulsion for the tethys long-range autonomous underwater vehicle. In: IEEE/OES Autonomous Underwater Vehicles (AUV), Monterey, CA

Bernitsas M, Ray D, Kinley P (1981) Kt, kq, and efficiency curves for the wageningen b-series propellers. Tech. rep., University of Michigan

Brizzolara S, Villa D, Gaggero S (2008) A systematic comparison between rans and panel methods for propeller analysis. In: 8th International Conference on Hydrodynamics (ICHD2008), Nantes, France

Brockett T, Korpus R (1986) Parametric evaluation of the lifting-line model for conventional and preswirl propulsors. In: Proceesings International Symposium on Propeller and Cavitation, Wuxi, China

CD-Adapco (2008) Starccm+, v. 2.10 users manual. Tech. rep., http://www.cd-adapco.com/products/star-ccm-plus

Coney WB (1989) A method for the design of a class of optimum marine propulsors. PhD dissertation, Massachusetts Institute of Technology, Cambridge, MA

Denny SB (1968) Cavitation and open-water performance tests of a series of propellers designed by lifting-surface methods. Tech. rep., Naval Ship Research and Development Center

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

Epps B (2010b) Openprop v2.4 theory document. Tech. rep., MIT

Epps B, Kimball R (2013a) OpenProp v3: Open-source software for the design and analysis of marine propellers and horizontal-axis turbines. URL http://engineering.dartmouth.edu/epps/openprop

Epps B, Kimball R (2013b) Unified rotor lifting line theory. Journal of Ship Research 57(4)

Epps B, Víquez O, Chryssostomidis C (2011) Dual-operating-point blade optimization for high-speed propellers. In: 11th International Conference on Fast Sea Transportation (FAST 2011), Honolulu, Hawaii

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: Proc. 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, Williamsburg, VA

Funeno I (2009) Hydrodynamic optimal design of ducted azimuth thrusters. In: First International Symposium on Marine Propulsors, Trondheim, Norway

Gaggero S, Brizzolara S (2007) Exact modelling of trailing vorticity in panel method for marine propeller. In: Proceedings of ICMRT, Naples, Italy

He L (2010) Numerical simulation of unsteady rotor/stator interaction and application to propeller/rudder combination. PhD thesis, The University of Texas at Austin, Austin

Hough GR, Ordway DE (1964) The Generalized Actuator Disk. Tech. Rep. TAR-TR 6401, Therm Advance Research

Isin Y (1987) Practical bollard-pull estimation. Marine Technology and SNAME News 24(3)

Jessup S (1989) An Experimental Investigation of Viscous Aspects of Propeller Blade Flow. PhD thesis, The Catholic University of America

Kerwin J, Hadler J (2010) Principles of Naval Architecture: Propulsion. SNAME, Jersey City, NJ

Kerwin J, Keenan D, Black S, Diggs J (1994) A coupled viscous/potential flow design method for wake-adapted, multi-stage, ducted propulsors. SNAME Transactions 102:23–56

Kerwin J, Michael T, Neely S (2006) Improved algorithms for the design/analysis of multi-component complex propulsors. In: Proc. 11th SNAME Propellers/Shafting Symposium, Williamsburg, VA

Kerwin JE, Coney WB, Hsin CY (1986) Optimum circulation distributions for single and multi-component propulsors. In: Twenty-First American Towing Tank Conference, Washington, DC, pp 53 – 62

Kim YC, Kim TW, Pyo S, Suh JC (2009) Design of propeller geometry using streamline-adapted blade sections. Journal of Marine Science and Technology 14:161–170

Kimball R (2001) Experimental investigations and numerical modelling of a mixed flow marine waterjet. PhD thesis, MIT

Laskos D (2010) Design and cavitation performance of contra-rotating propellers. Master's thesis, MIT

Laurens JM, Moyne S, Deniset F (2012) A bem method for the hydrodynamic analysis of fishing boats propulsive systems. In: Second International Symposium on Fishing Vessel Energy Efficiency, Vigo, Spain

Lee C (1979) Prediction of steady and unsteady performance of marine propeller with or without cavitation by numerical lifting surface theory. PhD thesis, MIT

Lerbs H (1952) Moderately loaded propellers with a finite number of blades and an arbitrary distribution of circulation. SNAME Transactions 60

Mertes P, Heinke HJ (2008) Aspects of the design procedure for propellers providing maximum bollard pull. In: ITS, 2

Oosterveld MWC, van Oossanen P (1975) Further computer-analyzed data of the wageningen b-screw series. International Shipbuilding Progress 22

Pashias C, Turnock S (2003) Hydrodynamic design of a bi-directional rim-driven ducted thruster suitable for underwater vehicles. Tech. Rep. Ship Science Report No. 128, University of Southampton

Tamura Y, Nanke Y, Matsuura M, Taketani T, Kimura K, Ishii N (2010) Development of a high performance ducted propeller. In: ITS, 7

Woud H, Stapersma D (2008) Design of Propulsion and Electric Power Generation Systems. IMarEST

Wrench JW (1957) The calculation of propeller induction factors. Tech. Rep. 1116, David Taylor Model Basin

Zan K (2008) A study of optimum circulation distributions for wind turbines. Tech. Rep. Report No UT-OE 08-3, Ocean Engineering Group, The University of Texas at Austin