
Using Blade Element Momentum Methods with Gradient-Based Design Optimization

Andrew Ning ([ORCID](#))

Abstract Blade element momentum methods are widely used for initial aerodynamic analysis of propellers and wind turbines. A wide variety of correction methods exist, but common to all variations, a pair of residuals are converged to ensure compatibility between the two theories. This paper shows how to rearrange the sequence of calculations reducing to a single residual. This yields the significant advantage that convergence can be guaranteed and to machine precision. Both of these considerations are particularly important for gradient-based optimization where a wide variety of atypical inputs may be explored, and where tight convergence is necessary for accurate derivative computation. On a moderate-sized example optimization problem we show over an order of magnitude increase in optimization speed, with no changes to the physics. This is done by using the single residual form, providing numerically exact gradients using algorithmic differentiation with an adjoint, and by leveraging sparsity in the Jacobian using graph coloring techniques. Finally, we demonstrate a revised formulation for cases when no inflow exists in one of the directions (e.g., a hovering rotor or a parked rotor). These new residuals allow for robust convergence in optimization applications, avoiding the occasional numerical difficulties that exist with the standard formulation.

Keywords BEM · blade · gradients · Jacobian

1 Introduction

Blade element momentum (BEM) theory is a common methodology for initial aerodynamic analysis and de-

sign of rotors, especially for propellers and wind turbines. The basic theory is well known and described in many textbooks [21, 47, 27, 9]. The method is of lower fidelity than mesh-based computational fluid dynamics, similar in fidelity to vortex lattice methods used for fixed wings [34], and is widely used for initial design and for pedagogical purposes. Numerous extensions have been added to the basic theory making it remarkably useful for industrial applications including hub/tip losses [60, 75, 73, 66, 6], turbulent wake region models [22, 74, 8, 9, 59, 61], wake expansion and swirl corrections [44, 15, 30], skewed inflow corrections [22, 11, 16, 45, 57, 56, 55, 69, 9], and on the airfoil side, rotational corrections [68, 13, 17, 18, 43, 4], high angle of attack extrapolations [72, 50], dynamic stall models to extend the method to unsteady loading situations [71, 41, 40, 54, 26, 39], etc. The importance and relative impact of these corrections has been demonstrated in propeller [48] and wind turbine applications [65].

Beyond analysis, there has been increased interest in using BEM for optimization application. Some of the difficulty lies in reliably solving the BEM equations. A variety of papers have demonstrated some of the convergence issues with the BEM equations and some ways to detect or improve the convergence, for example through relaxation methods [49, 46, 70]. Additionally, the efficiency of propeller/turbine optimization with the BEM equations has improved, often with multiple disciplines such as structures, dynamics, fatigue, and acoustics. Many studies have used gradient-free methods like genetic algorithms [58, 31, 64], genetic algorithms combined with topology optimization [2], and particle swarm methods [10], as well as sequential [28], hybrid [25, 5], and multilevel methods [38]. The difficulties of providing accurate gradients has been discussed [38], with many gradient-based approaches using finite differencing [15, 38, 51]. To allow for accurate gradients

A. Ning
EB 360, Brigham Young University, UT, 84604
Tel.: 801-422-1815
E-mail: aning@byu.edu

the complex step approach has been used [37]. More recently, OpenMDAO [24] has enabled approaches to more easily provide system-level gradients by breaking a program into smaller components for which analytic gradients can be supplied. The methodology from an early version of this paper was implemented in OpenMDAO for this purpose [32],¹ and that implementation has enabled several propeller optimization studies at NASA [33, 29].

Two main challenges with performing optimization using the BEM equations are addressed in this paper. First, as highlighted above, solving the residual equations is not always reliable. Fixed point iteration, Newton’s method, and its variants are susceptible to various well known convergence difficulties, or if multiple solutions exist, the solver may jump between solutions as the design is varied. While such scenarios are rare in analysis, they are more common during an optimization as the exploratory nature of the algorithms creates intermediate designs with sometimes extreme or ill-suited combinations of design parameters. For gradient-free optimization, occasional convergence failures or jumps in the solution space are not problematic, but for gradient-based methods these issues can cause the optimizer to terminate prematurely or produce gradients that are wildly inaccurate. As optimization problems increase in dimensionality, gradient-based methods become more desirable, prompting a greater need to allow for robust convergence and exact derivatives.

For a basic BEM implementation, the author developed a reformulated approach that reduces the solution process to a single residual and is provably convergent [52]. This prior derivation only accommodated certain correction methods. The present papers builds on this idea in a more general manner. The methodology no longer depends on preselected brackets, which allows for the use of any BEM formulation or extension (e.g., tip-loss, turbulent wake), as long as the computation of the induction factors can be decoupled, as is typical. Furthermore, we consider all possible inflow angles, large axial inductions, turbines or propellers, and cases with zero inflow from one of the directions (e.g., a hovering rotor, or a parked rotor). For the latter case a new formulation is derived that permits robust convergence and derivatives even for cases with exactly zero inflow, and the new residuals are shown to fill a singularity in the standard equations, unifying the solution space.

The second challenge addressed in this paper is providing accurate gradients, which can be difficult or is sometimes thought to be unnecessary for these lower-fidelity approaches. The tight convergence of the method-

ology discussed in this paper makes it easier to provide accurate gradients. We use a dual-number based algorithmic differentiation approach, which is straightforward to apply. We also show the impact of combining AD with analytic sensitivity methods around the solver, and demonstrate how leveraging sparsity in the Jacobian can greatly speed up the computational of derivatives in common BEM optimization scenarios. The significance of these considerations is demonstrated with an example optimization application.

The next two sections describe the theory. The fundamental BEM theory is well understood and available from many sources. We do generalize some aspects of the theory to allow wider exploration and discuss some details helpful for optimization. However, if the reader already has basic familiarity with the methodology and is mainly interested in the results, these two sections can safely be skipped. The details are relevant to those interested in implementation and for educational purposes.

Following the theory sections we discuss different ways to converge the residuals and highlight an efficient solution process that is guaranteed to converge. Next, we discuss derivative computation and explore how leveraging sparsity is often beneficial when computing derivatives for optimization problems involving BEM theory. These pieces are put together in an optimization example demonstrating the benefits afforded by the considerations discussed in this paper. Finally, the sometimes numerically problematic cases where one of the inflow velocities is zero (e.g., hover or a parked rotor) are explored with comparisons to experimental data and a demonstration of the improved robustness.

2 Theory: Standard Operation

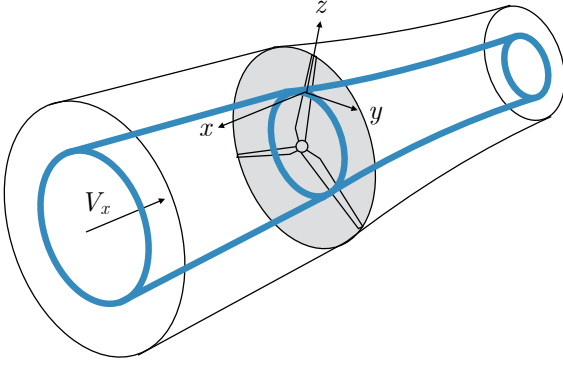
Blade element momentum theory is derived in many sources. We do not repeat all of those details, but rather focus on generalizing inputs, and noting some details that are important so that the output is smooth and amenable to gradient-based optimization. Through most of the document we will use the conventions for propellers, but the same equations are equally applicable to wind turbines as will be highlighted throughout.

2.1 Linear Momentum Balance

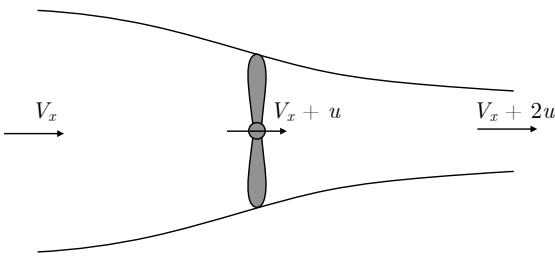
The methodology applies a mass and momentum balance to a streamtube passing through the rotor disk. Rather than using one streamtube for the entire rotor disk, it uses multiple infinitesimally thin annulus streamtubes, passing through a specific radial location

¹ The resulting OpenBEMT code has since been folded back into the CCBlade repository to consolidate efforts.

of the rotor (Fig. 1a). We denote the axial direction as x , positive in the designed direction of flight (or upwind facing direction of a wind turbine). The inflow velocity from the perspective of the propeller is in the opposite direction, which we call V_x rather than V_∞ as the inflow velocity will in general vary between radial stations because of wind, blade motion, etc. A positive thrust is in the $+x$ direction, and a positive torque is about the $+x$ axis. The positive y direction is a tangential direction in the designed rotation direction. The inflow opposes this direction and is called V_y . For a simple inflow condition $V_y = \Omega r$, but again we allow for general inflow including reversals. By the right hand rule the positive z direction is nominally from hub to tip of the blade. For blades with coning and sweep we would need to distinguish between the rotor and local coordinate systems, using for example the conventions in [53].



(a) The blue annulus streamtube is used as our control volume in the mass and momentum balances. The coordinate system is also shown.



(b) Depiction of the induced velocity at the rotor disk and in the farfield.

Fig. 1: Streamtube for mass and momentum balances.

Use of a mass and momentum balance, combined with either an inviscid energy balance or Bernoulli's equations applied separately before and after the rotor disk, yields the result that the velocity at the disk is halfway between the upstream and downstream velocity. The same relationship can be derived for a lifting

wing showing that the downwash at the wing is half of the downwash in the farfield.

With this relationship we relate the velocities upstream, at the disk, and downstream in terms of an unknown induced velocity u (Fig. 1b). The induced velocity is in the opposite direction of the force. So for a propeller the induced velocity is downstream, whereas for a turbine the induced velocity subtracts from the axial velocity.

By convention, one typically nondimensionalizes u as follows, where V_d is the velocity at the rotor disk:

$$\begin{aligned} V_d &= V_x + u \\ &= V_x \left(1 + \frac{u}{V_x} \right) \\ &= V_x (1 + a) \end{aligned} \quad (1)$$

The quantity a is called the axial induction factor.

The mass and momentum balance yields the following expression for the thrust coefficient on the rotor, which is positive in the $+x$ direction.

$$C_T = 4a(1 + a) \quad (2)$$

The coefficient was normalized by the dynamic pressure of the axial inflow ($q_x = \frac{1}{2}\rho V_x^2$) and the annular cross sectional area at the disk ($A_d = 2\pi r dr$). This is not a standard nondimensionalization in the propeller or rotorcraft communities, but is the convention used in the wind energy community and is used in this case just for convenience as it produces a simpler output. It doesn't matter what normalization is used for this force coefficient as it will cancel out when equating to blade element theory. For a turbine, the axial induction factor will be negative, and thus will produce drag. While it actually is a drag force, it is still commonly called thrust in the wind energy community by convention.

Hub and Tip Losses: The basic momentum theory ignores the hub and tip vortices that affect the induced velocity. Various correct methods exist, and any can be used. In this paper we use the simple analytical expression developed by Prandtl [21]:

$$\begin{aligned} f_{tip} &= \frac{B}{2} \left(\frac{R_{tip} - r}{r |\sin \phi|} \right) \\ F_{tip} &= \frac{2}{\pi} \arccos(\exp(-f_{tip})) \\ f_{hub} &= \frac{B}{2} \left(\frac{r - R_{hub}}{R_{hub} |\sin \phi|} \right) \\ F_{hub} &= \frac{2}{\pi} \arccos(\exp(-f_{hub})) \\ F &= F_{tip} F_{hub} \end{aligned} \quad (3)$$

The angle ϕ is the inflow angle shown later in Fig. 5. We have added the absolute values around $\sin \phi$ because our generalizations permit both positive and negative inflow angles and the loss functions must always be positive (between 0 and 1). This hub/tip-loss factor is applied directly to the thrust and torque.

$$C_T = 4a(1 + a)F \quad (4)$$

Large Induction Factors: The velocity in the wake from the momentum balance is $V_w = V_x(1 + 2a)$. For a turbine, a could drop below -0.5, and from that equation the wake velocities would be predicted to reverse direction (this consideration isn't important for propeller analysis). This reversal is non-physical, as the real flow entrains momentum in the wake through turbulence. Empirical data is needed to determine the behavior as a approaches -0.5 and beyond. Notional behavior of the thrust coefficient with large (negative) induction factors is seen in Fig. 2.

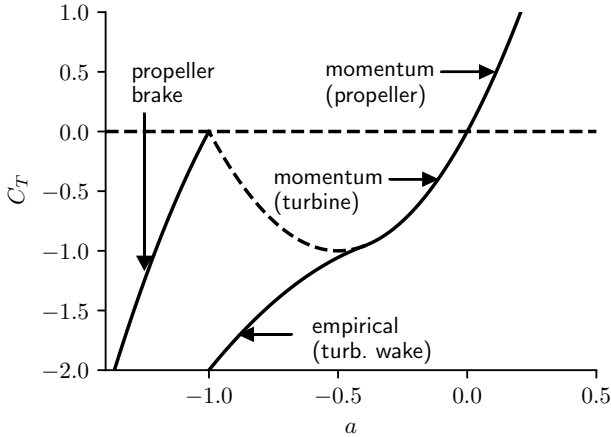


Fig. 2: Thrust coefficient as a function of axial induction factor.

Various extension methods exist for the turbulent wake region. A common simple method is the quadratic fit from Glauert [22]. However, the Glauert correction does not maintain continuity when the tip/hub loss corrections are included. Instead, we use a small modification of Glauert's method developed by Buhl (but written in propeller coordinates)[8]:

$$C_T = \left(4F - \frac{50}{9}\right)a^2 + \left(4F - \frac{40}{9}\right)a - \frac{8}{9} \quad (5)$$

for $-1 \leq a \leq -0.4$

Additional considerations may be needed for induction factors less than -1, the propeller brake region.

The current expression Eq. 4 predicts thrust for induction factors less than -1. However, repeating the momentum balance shows that the force changes signs, in other words it still acts as a drag device.

$$C_T = -4a(1 + a)F \quad (6)$$

For this case the rotor behaves like a propeller (requiring power input) but with a large negative pitch so that the thrust is reversed allowing the rotor to act like an aerodynamic brake.

Other Wind Directions: The above derivation is independent of the tangential, or in-plane velocities. If the axial wind direction is reversed ($V_x < 0$), then the typical direction and thus sign for thrust and u change as well. A flow reversal may happen locally over parts of a propeller, and for a wind turbine can occur across the entire rotor. In this case the equations would predict positive thrust to the right, but that is a negative thrust in our coordinate system. Thus, we need to multiply the thrust equation by $\text{sign}(V_x)$.

Summary:

$$C_T = \text{sign}(V_x) \begin{cases} 4a(1 + a)F & a \geq -0.4 \\ (4F - 50/9)a^2 + (4F - 40/9)a - 8/9 & -1 < a < -0.4 \\ -4a(1 + a)F & a \leq -1 \end{cases} \quad (7)$$

The latter two cases are generally not needed for propellers, and the thrust is positive in the $+x$ direction.

2.2 Angular Momentum Balance

Similar to the linear momentum case where an induced velocity is produced in opposition to the force on the rotor, a swirl velocity is induced in the opposite direction of the torque on the rotor. The rotational velocity change occurs much more rapidly. Conservation of momentum yields the same result as the linear case, where the induced swirl velocity at the disk is halfway between its upstream and downstream values. The induced rotational velocity is 0 upwind of the rotor, v in the plane of the rotor, and $2v$ downstream of the rotor. The tangential induction factor is defined as $a' = v/V_y$, where v is the induced velocity in the y direction and V_y is the inflow in the y direction.

An angular momentum balance across a given control volume can be expressed as:

$$\int_S (\vec{r} \times \vec{V}) \dot{m} = \sum \vec{r} \times \vec{F} \quad (8)$$

We use a disk-shaped control volume that surrounds the rotor disk, and assume no axial component of velocity exists on the sides of the control volume. We are then interested in only the inflow and outflow velocity vectors into the control volume. Figure 3 uses an ground-centered inertial control volume, rather than a blade-centric control volume to show the velocity triangles. This type of reference frame and orientation is more commonly used in turbomachinery analysis, and is convenient for this particular analysis. In this figure, v_w is the y-component of external wind. The axial component is somewhat larger than $V_x a$ as this end of the control volume is past the disk, however its exact magnitude is irrelevant as it does not contribute a net moment regardless of its value.

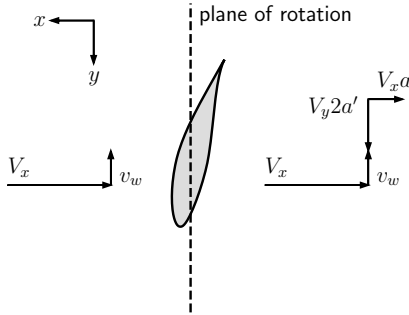


Fig. 3: Velocity vectors for angular momentum balance.

Applying conservation of angular momentum yields (where we define positive torque on the rotor as positive in the $+x$ direction):

$$\begin{aligned} (rv_w)(-\dot{m}) + r(v_w - V_y 2a')\dot{m} &= -Q \\ rV_y 2a'\dot{m} &= Q \end{aligned} \quad (9)$$

Notice, that the external wind component cancels out, assuming a relatively small disk, but apparent wind from blade motion would still be included in V_y . Using the results from the previous section:

$$\dot{m} = \rho V_d A_d = \rho V_x (1 + a) A_d \quad (10)$$

Thus:

$$Q = 2rV_y a' \rho V_x (1 + a) A_d \quad (11)$$

As we did for thrust, we normalize to form the torque coefficient.

$$C_Q = \frac{Q}{\frac{1}{2} \rho V_x^2 A_d r} \quad (12)$$

$$= 4a'(1 + a) \frac{V_y}{V_x} \quad (13)$$

The expression is equally applicable to turbines. For a propeller both a' and a are negative. This reverses the direction of the torque (power is extracted rather than required as an input).

Hub/Tip Losses: The torque correction is the same approach as is used for thrust, where the hub/tip loss factor F is multiplied against the torque.

Other Wind Directions: If the direction of V_x reverses, then nothing in the above derivation changes. We note that only component of velocity that matters is the induced velocity: $V_y 2a'$. If V_x reverses, then the induced velocity $V_y 2a'$ switches to the other side of the airfoil, but still points in the same direction. Because the direction of the velocity is the same, the $\vec{r} \times \vec{V}$ term retains the same sign, and the mass flow is still on the “out” side of the control volume and also retains the same sign. However, because V_x appears in our formula, and has switched signs itself (along with u), we need to flip the sign so that the overall sign for Q remains unchanged. This can be accomplished by taking the absolute value of V_x .

Changes in V_y don't matter per se, as long as the sign for a' doesn't change. A change in sign for a' is taken care of automatically as it appears in the equation, so no additional change is needed.

Summary:

$$C_Q = 4a'(1 + a) \frac{V_y}{|V_x|} F \quad (14)$$

The positive direction of torque for this derivation is in the $+x$ direction.

2.3 Blade Element Theory

Consider the airfoil section shown in Fig. 4 with the positive directions for twist (θ) defined as is conventional. We consider a general inflow of V_x and V_y . For an ideal condition $V_x = V_\infty$, and $V_y = \Omega r$, but because of wind, geometry, and blade motion we allow for any general velocity vectors.

We assume that the airfoil is generating positive lift, but the solution will dictate the correct sign. For positive lift, the direction of circulation Γ is into the page, and from the Kutta-Joukowski theorem:

$$\vec{F}' = \rho \vec{V} \times \vec{\Gamma} \quad (15)$$

We can compute the resulting forces from the V_x and V_y components of velocity. The resulting induced velocities oppose these forces. For example, V_x creates a

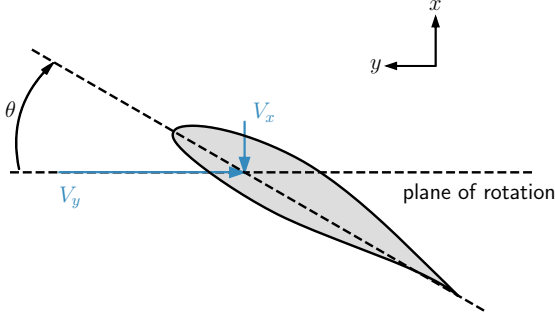


Fig. 4: Definition for positive twist and coordinate system for the blade element theory.

force to the right, and thus an induced velocity to the left opposing that of V_y . Conversely, V_y creates a force upward, and thus an induced velocity downward adding to V_x . As discussed previously, we normalize these induced velocities as follows: $a = u/V_x$ and $a' = v/V_y$, where u and v are the x- and y-components of induced velocity respectively. The resulting total inflow velocity vector W , positive direction for the inflow angle ϕ , and angle of attack α are shown in Fig. 5.

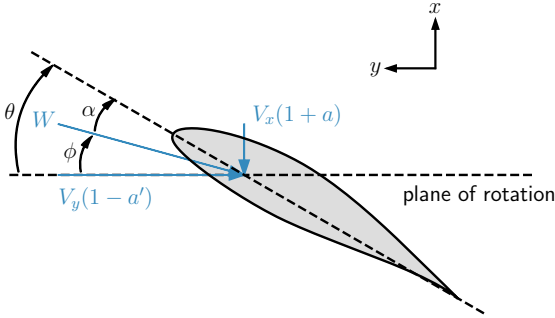


Fig. 5: Additional nomenclature used in blade element theory.

From the definition of the angles we can relate the angle of attack, twist, and inflow angle:

$$\alpha = \theta - \phi \quad (16)$$

From the angle of attack we compute the sectional lift and drag coefficient. The lift and drag coefficients may be functions of the Reynolds number and Mach number. Because we do not know a and a' we usually approximate the Reynolds number using the local velocity without induction included:

$$W_0 = \sqrt{V_x^2 + V_y^2} \quad (17)$$

$$Re = \frac{\rho W_0 c}{\mu}$$

The impact of this approximation is almost always negligible. Inclusion of a and a' usually has only a small effect on the Reynolds number, and typically only order of magnitude changes in Reynolds number are important. Additionally, airfoil data is rarely precise enough that a minor change in Reynolds number would be significant. Mach number does not operate on such a large scale, but is typically only considered with an approximate correction using the Prandtl-Glauert rule. Still, if exactness in Reynolds or Mach number is needed, this can be achieved with one or two extra iterations [52]. We can now compute the lift and drag coefficients using any appropriate method (e.g., table look-up, a panel method, 2D RANS, etc.). We denote these functions as f_L and f_D .

$$c_l = f_L(\alpha, Re, M) \quad (18)$$

$$c_d = f_D(\alpha, Re, M)$$

Note that the Mach number corrections are typically computed at each iteration rather than precomputed in the spline.

For 2D data, whether from simulation or wind tunnel measurements, it is essential to apply rotational corrections and extrapolate the data to high angles of attack. As referenced in the introduction, various methods exist for these corrections/extensions. In this paper we use the basic rotational corrections from Du and Selig for lift [17] and from Eggers for drag [18]. Extrapolation to high angles of attack uses Viterna's method [72]. Dynamic implementations should account for unsteady aerodynamic airfoil behavior, and so the lift and drag coefficients may require additional inputs. It is important that these functions are continuously differentiable for use in gradient-based optimization. In our case we fit the underlying data with Akima splines [1] to ensure smooth output. Neglecting smoothness in the airfoil data is a common source of convergence difficulty when using gradient-based optimization.

Using the Kutta-Joukowski theorem again, the directions for the lift and drag coefficients, c_l and c_d are as shown in Fig. 6. We need to resolve these forces into the normal and tangential directions, which by convention, corresponds to our positive x and negative y axes respectively.

$$c_n = c_l \cos \phi - c_d \sin \phi \quad (19)$$

$$c_t = c_l \sin \phi + c_d \cos \phi$$

The total thrust and torque for this blade section, multiplied by the number of blades B is:

$$T = BN' dr \quad (20)$$

$$T = Bc_n \frac{1}{2} \rho W^2 c dr$$

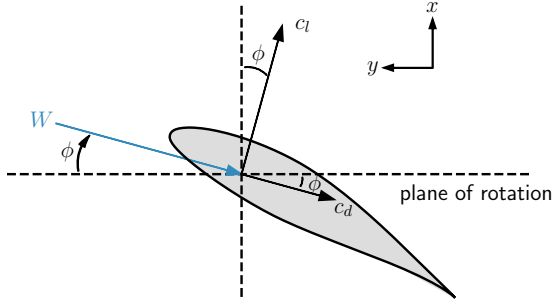


Fig. 6: Directions for the lift and drag forces for positive circulation.

$$Q = BrT'dr$$

$$Q = Br c_t \frac{1}{2} \rho W^2 c d r \quad (21)$$

where W must be the full velocity:

$$W = \sqrt{[V_x(1+a)]^2 + [V_y(1-a')]^2} \quad (22)$$

For normalization of thrust and torque coefficients we use the same dynamic pressure and reference area as used in the momentum balances. Furthermore, we define the local solidity as:

$$\sigma' = \frac{Bc}{2\pi r} \quad (23)$$

Performing the normalization results in:

$$\begin{aligned} C_T &= \frac{T}{q_x A_d} & C_Q &= \frac{Q}{q_x A_d r} \\ &= c_n \sigma' \left(\frac{W}{V_x} \right)^2 & &= c_t \sigma' \left(\frac{W}{V_x} \right)^2 \end{aligned} \quad (24)$$

Ultimately, we want to relate this expression in terms of the induced velocities. Using Fig. 5, we see that

$$W = \frac{V_x(1+a)}{\sin \phi} \quad (25)$$

or

$$W = \frac{V_y(1-a')}{\cos \phi} \quad (26)$$

It will be convenient later to use the first substitution in the thrust definition, and one of each in the torque definition:

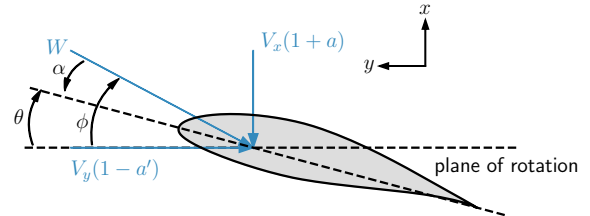
$$C_T = c_n \sigma' \left(\frac{1+a}{\sin \phi} \right)^2 \quad (27)$$

$$C_Q = c_t \sigma' \left(\frac{1-a'}{\cos \phi} \right) \left(\frac{1+a}{\sin \phi} \right) \left(\frac{V_y}{V_x} \right) \quad (28)$$

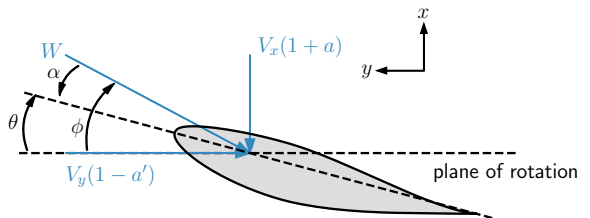
Other Wind Directions: Nothing in this derivation changes for other inflow directions, as long as we keep the definitions consistent (i.e., positive direction for ϕ , α , etc.).

Turbines: Everything in the above derivation will continue to work in turbine operation. As the twist decreases, and/or as the inflow velocity vector tilts away from the plane of rotation, eventually the angle of attack becomes negative as shown in the figure. The rotor switches from propeller to turbine operation once the angle of attack becomes negative enough that the lift coefficient changes direction, or in other words that the direction of circulation flips sign. This scenario is depicted in Fig. 7a, where we have changed both the twist and the inflow angle.

Everything in the above derivation continues to work correctly, the lift changes signs, as do the inductions (at the solution at least). While this section would operate as a turbine, if the airfoil is cambered it will be an inefficient turbine as the airfoil is not designed for these negative angles of attack. Instead, we would want to design it with the airfoil flipped over (i.e., reverse the camber) as shown in Fig. 7b. Then the “negative angle of attack” is positive from the perspective of the airfoil. In the wind energy community we would visualize this with the entire figure flipped over the plane of rotation so that the airfoil looked right-side up.



(a) As we decrease twist the section switches from propeller operation to turbine operation.



(b) For a turbine we would flip the camber so that the airfoil would perform as designed, with a positive angle of attack.

Fig. 7: Turbine operation requires no changes in the methodology, but a good turbine design would flip the camber direction.

Allowing for this camber switch is the only input needed to change a propeller BEM to a turbine BEM (or vice-versa). This change is contained to the airfoil function only, which is generally a preprocessing step:

$$\begin{aligned} f_L &= -c_l(-\alpha, Re, M) \\ f_D &= c_d(-\alpha, Re, M) \end{aligned} \quad (29)$$

Again, for a symmetric airfoil this change would be irrelevant.

For turbines another change might be included just for convenience, and that is flipping the sign on most all of the outputs. The angle of attack, induced velocities, and forces, are all consistent with the defined coordinate system, but would be considered to have the opposite sign in conventional turbine notation. Using typical wind turbine notation the following outputs would change sign:

$$a, a', u, v, c_n, c_t, N', T', \alpha, c_l \quad (30)$$

whereas the following would not change (ϕ is defined the same between the two, and the remaining quantities are always positive):

$$\phi, c_d, W, F \quad (31)$$

2.4 Blade Element Momentum

We can now combine the results from momentum theory and blade element theory. We first equate the linear momentum equations (thrust), and next the angular momentum equations (torque). The next section will discuss the residual equation which determines whether or not we have consistency between the momentum and blade element theories.

2.4.1 Axial Inflow

We equate the thrust from momentum theory and blade element theory. The thrust from blade element theory is always the same, but the momentum thrust changes depending on the sign of V_x and a (Eq. 7). There are multiple cases that must be considered depending on the sign of V_x and the magnitude of a .

- $V_x > 0$ and $a \geq -0.4$ or $V_x < 0$ and $a \leq -1$

These two cases yield the same result for momentum theory Eq. 7, which we equate to that of blade element theory Eq. 27:

$$\begin{aligned} 4a(1+a)F &= c_n \sigma' \left(\frac{1+a}{\sin \phi} \right)^2 \\ 4aF &= c_n \sigma' \frac{(1+a)}{\sin^2 \phi} \end{aligned} \quad (32)$$

We now define a new nondimensional quantity for convenience:

$$\kappa = \frac{c_n \sigma'}{4F \sin^2 \phi} \quad (33)$$

Making this substitution, we can derive a simple expression for a :

$$a = \frac{\kappa}{1 - \kappa} \quad (34)$$

The criteria for this equation was expressed in terms of V_x and a . However, a criteria in terms of a is not convenient, because that is the quantity we are solving for. Instead, we will express the criteria in terms of κ , which can be computed beforehand. Additionally, rather than using V_x to distinguish the cases, we will transform the criteria in terms of ϕ . This will allow for consolidation of the various cases.

First, from inspection of Fig. 5 we see that if $V_x > 0$ and $a \geq -0.4$ we must have $\phi > 0$. Or, if $V_x < 0$ and $a \leq -1$ we will also have $\phi > 0$. For the first case, this equation only applies if $a \geq -0.4$ or in other words:

$$\begin{aligned} \frac{\kappa}{1 - \kappa} &\geq -0.4 \\ \kappa &\geq -0.4(1 - \kappa), \text{ (assuming } 1 - \kappa > 0) \\ &\text{or in other words } \kappa < 1) \end{aligned} \quad (35)$$

$$\begin{aligned} 0.6\kappa &\geq -0.4 \\ \kappa &\geq -\frac{2}{3} \end{aligned}$$

Thus, this first cases applies if $-2/3 \leq \kappa < 1$.

The second cases applies if $a \leq -1$:

$$\begin{aligned} a &\leq -1 \\ \frac{\kappa}{1 - \kappa} &\leq -1 \\ \kappa &\geq -1 + \kappa, \text{ (assuming } 1 - \kappa < 0, \\ &\text{or in other words } \kappa > 1) \\ 0 &\geq -1, \text{ always true, assuming above condition} \end{aligned} \quad (36)$$

Thus, the second cases requires $\kappa > 1$.

Combining the two cases we see that the derived expression for a is valid for $\kappa \geq -2/3$ except for the point $\kappa = 1$. We will see later what the $\kappa = 1$ case corresponds to.

- $V_x < 0$ and $a \geq -0.4$ or $V_x > 0$ and $a \leq -1$

This scenario is identical to the previous, except for a negative sign in front of the momentum term. In short, we use the same equation and ranges from the previous case if we replace κ with $-\kappa$. From Fig. 5 we can see that these cases correspond to $\phi < 0$.

– $V_x > 0$ and $-1 < a < -0.4$

$$\left(4F - \frac{50}{9}\right)a^2 + \left(4F - \frac{40}{9}\right)a - \frac{8}{9} = c_n \sigma' \left(\frac{1+a}{\sin \phi}\right)^2 \quad (37)$$

This yields a quadratic formula that can be solved for a . After simplification it yields (noting that only the positive sign in the quadratic formula is physically possible):

$$a = \frac{\gamma_1 + \sqrt{\gamma_2}}{\gamma_3} \quad (38)$$

where

$$\begin{aligned} \gamma_1 &= F(2\kappa - 1) + \frac{10}{9} \\ \gamma_2 &= F \left(F - 2\kappa - \frac{4}{3} \right) \\ \gamma_3 &= 2F(1 - \kappa) - \frac{25}{9} \end{aligned} \quad (39)$$

If the denominator in Eq. 38 is exactly zero (i.e., $\gamma_3 = 0$), then the numerator is also exactly zero. However, the expression can still be evaluated using L'Hôpital's rule and can be shown to be equal to

$$a \xrightarrow{\gamma_3 \rightarrow 0} \frac{1}{2\sqrt{\gamma_2}} - 1 \quad (40)$$

From Fig. 5 we see that $\phi > 0$ for our conditions on V_x and a . This expression will always yield $a > -1$ so the limit we are concerned with is $a < -0.4$. We can show that this occurs for $\kappa < -2/3$, which should make sense as this region was designed to connect at the border of the momentum region.

– $V_x < 0$ and $-1 < a < -0.4$

The final case is identical to the previous, except for the negative sign in the thrust from momentum theory. Again, we can reuse the previous result, but replace κ with $-\kappa$ and the limit on ϕ changes to $\phi < 0$.

These cases can be consolidated with the logic shown in Alg. 1. Recall the existence of a invalid range at $\kappa = 1$ for $\phi > 0$ and at $\kappa = -1$ for $\phi < 0$. We will see in Sec. 3 that the $|\kappa| = 1$ case is only physically consistent if $V_x = 0$. For nonzero inflow, we know that κ cannot equal 1 so if any intermediate iterations produces $\kappa = 1$ we can simply return a nonzero residual and continue iterating.

Algorithm 1 Solve for the axial induced velocity.

if $\phi < 0$ **then**

$\kappa = -\kappa$

end if

if $\kappa \geq -2/3$ **then**

$a = \kappa/(1 - \kappa)$ \triangleright if $\kappa = 1$ return any nonzero residual.

else

$a = (\gamma_1 + \sqrt{\gamma_2})/\gamma_3$

\triangleright if $\gamma_3 = 0$ use Eq. 40.

end if

2.4.2 Tangential Inflow

We now equate the torque from blade element theory (Eq. 28) with the torque from momentum theory (Eq. 14). There are two cases. First, $V_x > 0$:

$$\begin{aligned} c_t \sigma' \left(\frac{1-a'}{\cos \phi} \right) \left(\frac{1+a}{\sin \phi} \right) \frac{V_y}{V_x} &= 4Fa'(1+a) \frac{V_y}{V_x} \\ c_t \sigma' \left(\frac{1-a'}{\cos \phi \sin \phi} \right) &= 4Fa' \end{aligned} \quad (41)$$

Similar to the axial induction derivation, we define a new nondimensional quantity for convenience:

$$\kappa' = \frac{c_t \sigma'}{4F \sin \phi \cos \phi} \quad (42)$$

With that substitution we can solve for a' as

$$a' = \frac{\kappa'}{1 + \kappa'} \quad (43)$$

Note that this is defined everywhere except when $\kappa' = -1$ (we will see in Sec. 3 that this case is only physically consistent if $V_y = 0$).

For $V_x < 0$ we can use the same solution by negating the value for κ' . These two cases can be combined into the algorithm shown in Alg. 2.

Algorithm 2 Solve for the tangential induced velocity.

if $V_x < 0$ **then**

$\kappa' = -\kappa'$

end if

$a' = \kappa'/(1 + \kappa')$ \triangleright if $\kappa' = -1$ return any nonzero residual.

2.5 Loads

The solution of the residual equations is discussed in the following section, but once the residuals are solved, the distributed loads (force per unit length) can be computed. It is important to use the full induction for this

calculation.

$$\begin{aligned} W &= \sqrt{V_x(1+a)^2 + V_y(1-a')^2} \\ q &= \frac{1}{2}\rho W^2 \\ N' &= c_n qc \\ T' &= c_t qc \end{aligned} \quad (44)$$

If we define the local precone angle as Φ then the thrust can be found through integration as:

$$T_i = B \int_0^R N' \cos \Phi dr \quad (45)$$

the torque is given by

$$Q_i = B \int_0^R T' r \cos \Phi dr \quad (46)$$

These integrals actually give the instantaneous thrust and torque, at a particular azimuthal angle, and for one blade. If significant azimuthal variation exists, we should azimuthally average:

$$\begin{aligned} T &= \frac{1}{2\pi} \int_0^{2\pi} T_i d\theta \\ Q &= \frac{1}{2\pi} \int_0^{2\pi} Q_i d\theta \end{aligned} \quad (47)$$

Usually, this averaging is only relevant for wind turbines because of their large size, and even in that case integration is typically fairly coarse (e.g., 4 or 8 azimuthal positions). Power is then given by

$$P = Q\Omega \quad (48)$$

3 Theory: No Inflow Velocity in One Direction

Sometimes it is desired to explore cases where one of the inflow velocities is zero (i.e., either $V_x = 0$ or $V_y = 0$). Example applications include a hovering propeller for the former, and extreme loads on a parked wind turbine for the latter. Induction factors $a = u/V_x$ or $a' = v/V_y$ are no longer defined, and one cannot just introduce zero into the BEM equations, at least not without some algebraic manipulations.

3.1 Linear Momentum

Consider the case when $V_x = 0$. With no axial inflow there will be no tangential induction, but there will be an axially induced velocity. As noted, we cannot use the induction factor and so use the induced velocity u .

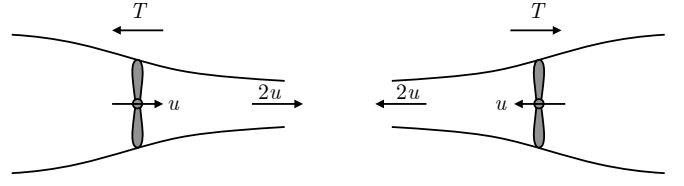


Fig. 8: Two possibilities for the direction of thrust and induced velocity for a rotating blade with no inflow (e.g., hover).

Because the thrust and induced velocity must be in opposite directions, by Newton's third law, two scenarios are possible as shown in Fig. 8.

Using the standard BEM derivation, but in dimensional terms, the thrust is given by:

$$T = -4\pi r \rho u^2 F dr \quad (49)$$

This is the standard momentum contribution used in BEM implementations for helicopters. For more generality, we should consider the sign of the velocity. We define the positive convention for u , as consistent with our positive convention for a . In other words, a positive u is in the $-x$ direction (opposite the positive direction for thrust). Thus, our equation for thrust becomes:

$$T = \text{sign}(u) 4\pi r \rho u^2 F dr \quad (50)$$

If $V_y = 0$, then there is no axially induced velocity ($a = 0$), and V_x passes through unchanged. Momentum theory predicts no thrust. There is still (negative) thrust, from the drag on the blades, but this arises from blade element theory and is treated later.

3.2 Angular Momentum Balance

If $V_x = 0$, then there is no induced velocity in the tangential direction ($v = 0$). In that case, no torque is predicted from momentum theory, but some torque will be generated from blade element theory as will be treated later.

If $V_y = 0$, then there is no induced velocity in the axial direction ($u = 0$). There is induced velocity in the tangential direction, but we cannot use the normalization for $a' = v/V_y$ because $V_y = 0$. Instead, we refer to the total induced velocity in the plane of the rotor as v . The resulting torque from Eqs. 9 and 10 is:

$$\begin{aligned} -rv_w \dot{m} + r(v_w - 2v) \dot{m} &= -Q \\ 2rv \dot{m} &= Q \\ Q &= 2rv \rho V_x A_d \end{aligned} \quad (51)$$

We can normalize, because $V_x \neq 0$, but it will be convenient to keep this expression in the unnormalized form. We do, however, need to add the hub/tip loss factor

The sign of v is consistent with the sign of Q automatically. However, as discussed in Sec. 2.2, we must take the absolute value of V_x to account for reversals in inflow direction.

$$Q = 4\pi r^2 \rho v |V_x| F dr \quad (52)$$

3.3 Blade Element

As mentioned, if $V_x = 0$ then $a' = 0$. Also, a is undefined, because of the normalization by V_x and so we must refer to the total axial induced velocity u . There are two possible directions for u (Fig. 9). From the figure we also see that we can define W as:

$$W = \frac{u}{\sin \phi} = \frac{V_y}{\cos \phi} \quad (53)$$

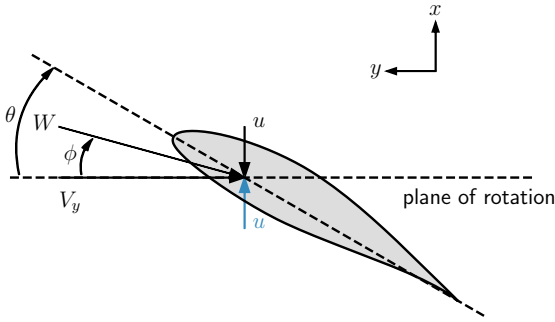


Fig. 9: Two possible directions for the induced axial velocity u . The black u is positive according to our sign convention, and the blue u is negative.

The direction of u and c_n must be opposite. Thus, for the case of $u > 0$ (remember a positive u is in the negative x -direction using our convention), then c_n must be positive, and $\phi > 0$. The opposite is true for the case with $u < 0$: $c_n < 0$ and $\phi < 0$. We can determine these signs a priori, unlike the more general case, because the velocity is strictly determined by u , and the tangential velocity has no induction.

If $V_y = 0$ then $a = 0$. Also, a' is undefined, because of the normalization by V_y and so we must refer to the total tangential induced velocity v . There are two possibilities for v shown in black and blue in Fig. 10. We know that c_t and v must be in opposite directions (which is the same sign in this convention). From Fig. 10 we see that:

$$W = \frac{V_x}{\sin \phi} = \frac{-v}{\cos \phi} \quad (54)$$

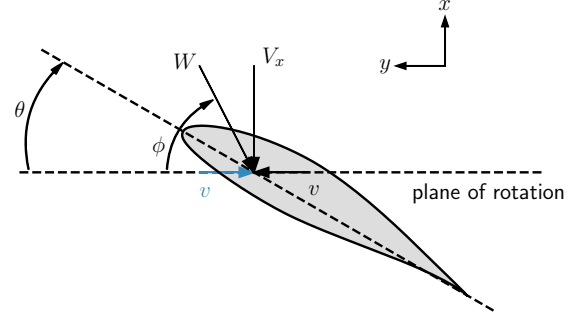


Fig. 10: Two possible directions for the induced tangential velocity v . The black v is positive according to our sign convention, and the blue v is negative.

3.4 Blade Element Momentum

We now put the pieces together for the two different cases separately.

3.4.1 Hover

If $V_x = 0$, equating thrust from the momentum and blade element theories yields for $u > 0$:

$$\begin{aligned} 4\pi r \rho u^2 F dr &= B c_n \frac{1}{2} \rho W^2 c dr \\ 4u^2 F &= \sigma' c_n W^2 \end{aligned} \quad (55)$$

The value for W is shown in Eq. 53 for $V_x = 0$. Let's first use the first option: $W = u / \sin \phi$. Substituting into the above expression and simplifying yields:

$$\kappa = 1 \quad (56)$$

Conversely, if $u < 0$ the methodology yields.

$$\kappa = -1 \quad (57)$$

We note that the first case with $u > 0$ implies that $\phi > 0$ (see velocity triangle in Fig. 9) and results in the expression $\kappa = 1$. This case fills in our singularity noted in Alg. 1 and the discussion proceeding that algorithm. In other words, the case $\phi > 0, \kappa = 1$, only exists for $V_x = 0$. Similarly, the derivation for $u < 0$ corresponds to $\phi < 0$ with the result that $\kappa = -1$, which fills in the other singularity. Thus, the previous formulation ($V_x \neq 0, V_y \neq 0$) should ignore the case $\kappa = 1$ as it won't satisfy the residual equation. Any nonzero residual could be returned. We deal with this case explicitly in this section with a different formulation.

While, the substitution $W = u / \sin \phi$ led to the insight of filling in our singularity, it doesn't help us solve

the equation. Instead we will use one of each substitution from Eq. 53. Note that we can also use the second substitution twice. That formulation requires a little more logic to solve correctly as it involves a square root for which the sign must be protected. Either way one should get the same solution once the residual is converged. To simplify the logic we use one substitution from each. For $u > 0$:

$$4u^2 F = \sigma' c_n \frac{V_y}{\cos \phi} \left(\frac{u}{\sin \phi} \right) \quad (58)$$

$$u = \frac{\sigma' c_n}{4F \sin \phi \cos \phi} V_y$$

The physics of this case is only consistent with $c_n > 0$, $u > 0$, $\phi > 0$ and additionally V_y and $\cos \phi$ must have the same sign. All of this is handled automatically by the residual.

Conversely if $u < 0$ then the sign on the momentum portion of thrust switches and we have:

$$-4u^2 F = \sigma' c_n \frac{V_y}{\cos \phi} \left(\frac{u}{\sin \phi} \right) \quad (59)$$

$$u = -\frac{\sigma' c_n}{4F \sin \phi \cos \phi} V_y$$

This solution is consistent as long as $c_n < 0$ and $\phi < 0$. We can combine these two cases based on the sign of ϕ :

$$u = \text{sign}(\phi) \kappa V_y \tan \phi \quad (60)$$

$$(61)$$

The residual equation must be algebraically simplified using $V_x = 0$ and $v = 0$. Alternatively, we could use the new velocity triangle in Fig. 9. Either way leads to the equation:

$$\frac{\sin \phi}{u(\phi)} - \frac{\cos \phi}{V_y} = 0 \quad (62)$$

While we could leave it in this form, it is numerically beneficial to simplify further as some terms will cancel. After some simplification using the formula for u above we find:

$$\text{sign}(\phi) - \kappa = 0 \quad (63)$$

We see that there is only a solution when ϕ and c_n have the same sign, as is consistent with the physics.

The relevant quadrants can be determined from the sign of V_y , although typically only $V_y > 0$ is of interest for hover. Because $a' = 0$, just knowing the sign of V_y immediately eliminates two of the four quadrants. The sign of u is not known, but knowing the sign of the twist gives us a good idea of what the sign of u will be. For a positive twist, the most likely scenario is $c_n > 0$ thus

Table 1: $V_x = 0$, θ includes pitch

V_y	θ	quadrant order
+	+	I, II
+	-	II, I
-	+	III, IV
-	-	IV, III

$u > 0$, and vice-versa for a negative twist. The order of quadrants is defined in Tab. 1.

Once we solve this 1D equation for ϕ , we can compute the torque distribution purely from blade element theory as momentum theory does not predict any torque for the case $V_x = 0$. We can use either expression for W in Eq. 53.

3.4.2 Parked Rotor

If $V_y = 0$ we equate momentum theory with blade element theory giving:

$$4\pi F r^2 v \rho |V_x| dr = B r c_t \frac{1}{2} \rho W^2 c dr \quad (64)$$

$$4F v |V_x| = \sigma' c_t W^2$$

If we use one of each substitution for $W = V_x / \sin \phi$ and $W = -v / \cos \phi$ from Eq. 54 in the above expression we have:

$$4F v |V_x| = -\sigma' c_t \frac{V_x v}{\sin \phi \cos \phi} \quad (65)$$

This expression simplifies to $\kappa' = -1$ for $V_x > 0$ and to $\kappa' = 1$ for $V_x < 0$. Like the axial inflow case, these scenarios fill in the singularity shown in Alg. 2. As in the axial inflow case, we simply return a nonzero residual if that case occurs for $V_x, V_y \neq 0$, and we handle the case $V_y = 0$ explicitly below.

Like before, the simplifications yielding $\kappa' = -1$ are insightful, but don't lead to a solution process. We can instead use either substitution for W . After simplification both lead to the same residual function. In the following we will use $W = V_x / \sin \phi$. This yields the following:

$$v = \frac{\kappa' |V_x|}{\tan \phi} \quad (66)$$

We see that v follows the sign of c_t , which is consistent with the physics.

After simplification the residual becomes:

$$\text{sign}(V_x) + \kappa' = 0 \quad (67)$$

The quadrant search order is defined in Tab. 2, and the methods of this section are summarized in Alg. 3.

Table 2: $V_y = 0$, θ includes pitch

V_x	$ \theta $	quadrant order
+	$< \pi/2$	I, III
-	$< \pi/2$	II, IV
+	$> \pi/2$	III, I
-	$> \pi/2$	IV, II

Algorithm 3 Solve the residual equation for the no wind cases.

```

if  $V_x = 0$  then
   $u = \text{sign}(\phi)\kappa V_y \tan \phi$ 
   $v = 0$ 
   $R = \text{sign}(\phi) - \kappa$ 

else if  $V_y = 0$  then
   $u = 0$ 
   $v = \kappa'|V_x|/\tan \phi$ 
   $R = \text{sign}(V_x) + \kappa'$ 

else
  Use the normal procedure for computing induction.

end if
 $W = \sqrt{(V_x + u)^2 + (V_y - v)^2}$   $\triangleright$  Could use dimensional
form to consolidate velocity calculation for all cases.
```

4 Solution Approach

While the core BEM theory is fairly standard, a wide variety of implementations exist because of the various corrections and extensions discussed in the introduction. In common to all of these, a set of residual equations must be converged, ensuring compatibility between momentum theory and blade element theory. The induction factors at the current iteration are in general not consistent as they depend on the angle of attack, which in turn depends on the induction factors themselves. Thus, an iterative, or root-finding method is necessary.

Traditionally, the residuals are solved by considering the induction factors a and a' as the unknown variables, and from those inputs, we compute updated predictions for a and a' :

$$a_{k+1}, a'_{k+1} = f(a_k, a'_k) \quad (68)$$

where k is the iteration index. At the end of each iteration we can then form two residuals:

$$\begin{aligned} \mathcal{R}_1(a, a') &= a_{k+1} - a_k = 0 \\ \mathcal{R}_2(a, a') &= a'_{k+1} - a'_k = 0 \end{aligned} \quad (69)$$

This form lends itself well to fixed point iteration, or to a multidimensional root finding algorithm.

If the selected BEM models fit this form then an alternative exists that is much more robust and just as

easy to implement. The inflow for an airfoil section, using propeller conventions, is shown in Fig. 5. As shown in our previous work, the residuals can be solved much more effectively by realizing that we can define the velocity vectors equivalently by considering the two unknowns to be W and ϕ rather than a and a' [52]. This choice results in a big simplification because W only appears in the Reynolds and Mach numbers, and as discussed in Sec. 2 we can almost always safely neglect the induction factors in these calculation (or if we really want to include them this can be done easily in an extra iteration). In other words, we can reduce our unknown variables to one: ϕ . This reduction has the massive advantage that one-dimensional root finding problems are much easier to solve, and unlike multi-dimensional root finding algorithms, we can guarantee convergence as long as we can find a suitable bracket.

The required changes to the above form are minimal. As discussed, we conventionally consider a and a' as inputs. The next step is to compute the inflow angle from the induction factors using the relationship seen in Fig. 5:

$$\tan \phi = \frac{V_x (1 + a)}{V_y (1 - a')} \quad (70)$$

However, for the one equation method we take ϕ in directly as an input and thus bypass this step. The rest of the methodology proceeds in exactly the same way as the traditional one. Just like the traditional method we compute updated values for a and a' except that they are now functions of ϕ rather than prior guess for a and a' : $(a(\phi), a'(\phi))$. The only other change needed is to define a new residual function. This is done by using the expression, Eq. 70, that has not yet been used in this reordered formulation. This equation can be arranged many different ways to form a residual function, but not all will lead to a reliably convergent method. Singularities in the residual function are unavoidable, but it is convenient to have the quantities $(1 + a)$ and $(1 - a')$ in the denominator so that singularities occur at the predefined locations: $\phi = 0, \pm\pi$ [52]. These locations are particularly convenient because they also separate regions where the physics change. We rearrange the equation into the residual form below:

$$\mathcal{R}(\phi) = \frac{\sin \phi}{1 + a(\phi)} - \frac{V_x \cos \phi}{V_y (1 - a'(\phi))} = 0 \quad (71)$$

This small change provides a significant simplification, but the benefits are not realized if an appropriate solver is not used. Bracketing methods are slow, and quadratic interpolation methods can have similar convergence problems to the two dimensional approach. Hybrid methods combine bracketing with quadratic or

cubic interpolation to provide guaranteed convergence with typically superlinear convergence. Brent’s method is perhaps the most well-known of these algorithms and is what is used in this paper. The concept was proposed by Dekker [14], which was soon after made significantly more efficient by Brent [7]. In turn, more recent methods have proposed small improvements over Brent’s method like using cubic interpolation instead of quadratic [3].

The remaining consideration is to determine a bracket. In prior work, we showed that for a typical implementation with standard inflow ($V_x > 0$ and $V_y > 0$) the brackets can be determined *a priori* [52]. However, for more general inflow conditions and/or use of other correction functions, the brackets may need to be established numerically. All that is needed to establish a bracket, $[\phi_L, \phi_U]$, is to find two points between which the residual function changes sign (i.e., $\mathcal{R}(\phi_L)\mathcal{R}(\phi_U) < 0$). Fortunately, from the signs of V_x and V_y , we can determine where a bracket is most likely to be found. Figure 11 divides the ϕ range into four quadrants. We search quadrants in the order shown in Tab. 3. It would be very rare that the solution would not occur in the first quadrant listed, but for completeness we list all possibilities in the order of likelihood. Within each quadrant, we still need to establish a bracket around a root. Generally there is not more than one solution in a given quadrant. However, the numerical possibility of multiple solutions exists and so we search for the solution closest to $\phi = 0$, as that is the most physically likely. The method we use subdivides the quadrant into n_{int} intervals, where n_{int} is a user-defined parameter. Starting at the lower bound, we march forward towards the upper bound looking for a change in sign. Finding a bracket is straightforward, and once a bracket is determined the convergence is guaranteed. In practice, we have found this approach to always converge, and to machine precision, across a wide range of possible inputs.

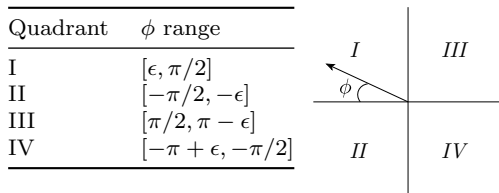


Fig. 11: Quadrants where a potential solution may exist. The use of ϵ (a small number like 10^{-6}) is to avoid the singularities at $\phi = 0$ and $\phi = \pi$.

In this paper we also explore using the traditional two-dimensional residuals. This form could be solved

Table 3: Quadrant search order based on the location where a solution is most likely to be found.

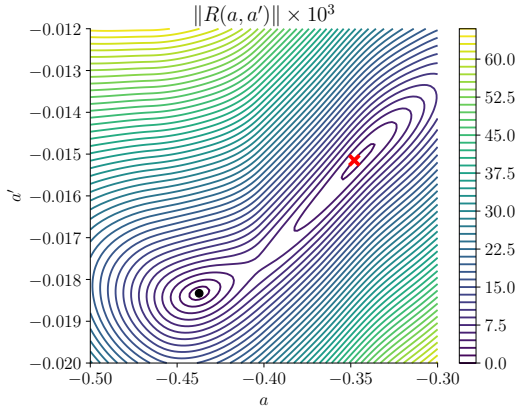
V_x	V_y	quadrant order
+	+	I, II, III, IV
-	+	II, I, IV, III
+	-	III, IV, I, II
-	-	IV, III, II, I

with fixed point iteration, but convergence is not always reliable. Newton’s method is another common choice but it also suffers from relatively high failure rates during an optimization. In this study we use the trust-region method in NLSolve.jl². For the problems tested in this paper, this method was found to be the fastest of the methods in the NLSolve package, and was found to be more consistently reliable than the solvers in Python’s `scipy.optimize.root`. To further speed up the 2D approach, we compute the Jacobian of the residuals using algorithmic differentiation. We scaled the second residual by an order of magnitude, the one associated with a' , as that improved overall convergence (this trust region method actually performs auto scaling, but scaling helped the other methods that we compared against and makes the contour plots easier to visualize).

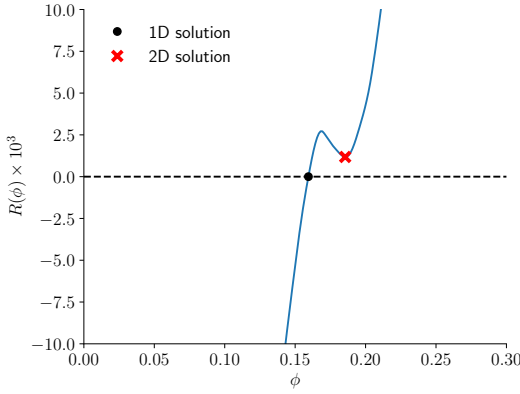
Even still, the methodology is prone to occasional failures. There are many reasons why the equations could fail, a rather benign, but frequently occurring case is discussed below. During a wind turbine optimization a failure point was noted and saved for further inspection. The induction factors a and a' were varied near the failure point to visualize the solution space. A contour plot was created using the norm of the two residuals in Fig. 12a. The point identified by the 2D solver is denoted by the red x. While at first glance it may appear to be a solution, this is not a minimization problem, but rather a root finding problem and the local valley does not actually contain a root. Because the function increases in every direction, the root finding method terminates and cannot improve.

In contrast, the 1D solution method can correctly identify the root (shown by the black dot) in the same figure. Across a wide range of optimizations the 1D solution method never failed. We can view these two solutions on the 1D solution space as well as shown in Fig. 12b. The reason this behavior occurs is that this is a section near the hub of the blade and it is operating near stall. It is common for large wind turbines to have sections near the hub operating near stall. Because of the nature of the lift coefficient near stall there is a reversal in the lift coefficient and subsequently in the

² <https://github.com/JuliaNLSolvers/NLSolve.jl>



(a) Solution space for the two-dimensional residual form. The red x found by the 2D algorithm is not actually a root (just a minimum).



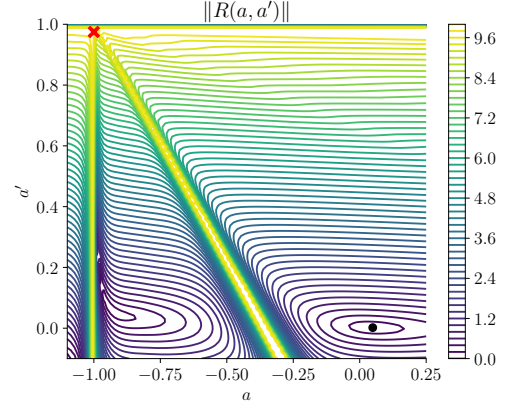
(b) Solution space for the one-dimensional residual form.

Fig. 12: Contrast between the correct solution found by the 1D algorithm, and a non-solution where the 2D algorithm is forced to terminate.

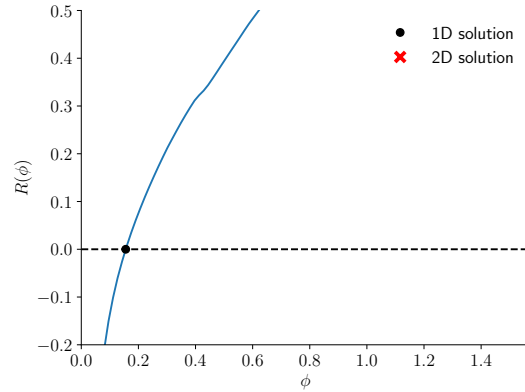
residual. A bracketing method (such as Brent's method used in the 1D solution algorithm) has no problem navigating a nonmonotonic function and correctly identifies the solution, but the 2D approach has no such guarantees and can easily get stuck in a point that does not satisfy the governing equations.

In this case the error is relatively insignificant, but much more significant errors can easily occur. In another scenario if we take the starting reference wind turbine (discussed in Sect. 6), and add just five degrees of pitch, multiple convergence failures occur across the blade. One such scenario is depicted in Fig. 13. The diagonal ridge in Fig. 13a represents the separation between solutions with positive and negative circulation from the blade element formulation and the 2D algorithm gets stuck at very erroneous induction factors. In contrast, the 1D method easily identifies the correct so-

lution where this section has a negative induction and produces some thrust (a section that locally behaves like a propeller even though the rotor as a whole behaves like a turbine). In Fig. 13b the 2D solution does not appear as it is way off the plot.



(a) Solution space for the two-dimensional residual form. The red x from the 2D algorithm is clearly not a solution.



(b) Solution space for the one-dimensional residual form.

Fig. 13: A more catastrophic failure when searching the 2D space.

Of course better starting points can lead to better convergence in the 2D space, but for the 2D case it is not always easy to know what the starting values should be beforehand. In the last case, for example, it would have been expected to have negative induction factors as the rotor was a wind turbine and most of the rotor was acting as a turbine. Alternatively, stochastic approach can be helpful in avoiding this type of premature convergence, but these methods are much less efficient computationally, especially in the cases discussed in this paper where we are interested in tight tolerances to produce accurate derivatives. In contrast, the 1D so-

lution only requires brackets that are easily identifiable and yield guaranteed convergence.

It should be mentioned that there are some model choices that may not lend themselves to this form and thus cannot be reformulated into the one-dimensional form. For example, one skewed inflow models does not allow for decoupled calculation of a and a' [53]. It is not problematic if the expressions for a and a' are implicit, but they do need to be separable. In these less common scenarios, we must stick with the two-dimensional version and form residuals on the forces that are always separable as the methods are distinct:

$$\begin{aligned}\mathcal{R}_1(a, a') &= C_{T \text{ momentum}} - C_{T \text{ blade element}} = 0 \\ \mathcal{R}_2(a, a') &= C_{Q \text{ momentum}} - C_{Q \text{ blade element}} = 0\end{aligned}\quad (72)$$

5 Derivative Computation

Accurate derivative computational is important for utilizing BEM theory in optimization applications. Finite differences are common, but are inaccurate and scale inefficiently. Gradient-free optimization approaches are straightforward to apply but scale even more inefficiently. As blade design is just one aspect of a wider multidisciplinary problem, and there is increased interest in optimizing propellers across trajectories with many variables, it is desirable to allow for efficient scaling with increasing design variables.

The main complication to computing derivatives in BEM theory is the need to solve a residual equation internally (Eq. 71). This residual is applied at each radial section on the blade independently and also in time independently. The phrase “in time” could apply to a quasi-static simulation (e.g., a flight maneuver), or could denote multiple operating conditions that aren’t necessarily correlated in time per se but rather through some other operating variable like wind speed (e.g., a wind turbine power curve).

In this paper we use algorithmic differentiation (AD) to compute derivatives. Because there is an internal solver, we can either propagate dual numbers through solvers, or use an analytic approach around the solver (i.e., direct or adjoint). Using AD through a solver is simpler but less efficient. Also, when using AD through a solver one generally must use a tighter tolerance as the dual values converge slower than the primal values, fortunately the 1D algorithm allows for convergence to machine precision. We explore both procedures in this paper. The formulation for using AD with an analytic approach around the implicit equations requires a bit more effort, but does provide a speed advantage for some of the cases. We tested two different AD approaches, a forward mode and a reverse mode AD

method, using the ForwardDiff.jl [63] and ReverseDiff.jl³ packages respectively.

One key insight into computing gradients efficiently is to take advantage of sparsity that often exists in the Jacobians of these problems. As an example, consider a case using all the geometric inputs to the BEM solver and the two main outputs: thrust and torque. The inputs used include:

$$x = [r_1, r_2, \dots, r_m, c_1, c_2, \dots, c_m, \theta_1, \theta_2, \dots, \theta_m, R_{hub}, R_{tip}, \Phi, \rho, V_\infty, \Omega, \theta_p] \quad (73)$$

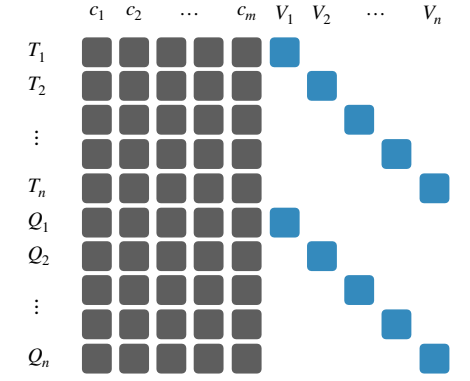
where m represents the number of radial stations along the blade, and the variables are radial stations, chord lengths, twist angles, hub and tip radii, precone angle, density, airspeed, rotation speed, and the pitch angle. For this sample problem there were 11 radial stations and so the Jacobian was of size 2×40 and was fully dense. Even though the loads at radial station i only depend on the geometry at station i (r_i, c_i, θ_i), the thrust and torque are integral quantities and thus there is no sparsity to take advantage of. One would expect reverse mode AD to be advantageous in this situation as the number of outputs is much smaller than the inputs, however, the cost of constructing the computational graph took more time than just using the forward mode. In either case, the Jacobian is small, so the differences are not that significant.

The more interesting case occurs as we consider multiple inflow conditions. A common evaluation scenario requires using the BEM for the same geometry but with different operating points. For example, flight conditions across an aircraft’s trajectory, or operating points in a wind turbine’s power curve. For these scenarios the inputs are as follows:

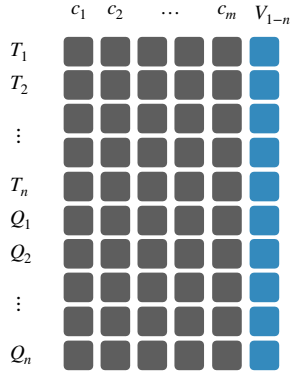
$$x = [r_1, r_2, \dots, r_m, c_1, c_2, \dots, c_m, \theta_1, \theta_2, \dots, \theta_m, R_{hub}, R_{tip}, \Phi, \rho, V_{\infty 1}, V_{\infty 2}, \dots, V_{\infty n}, \Omega_1, \Omega_2, \dots, \Omega_n, \theta_{p1}, \theta_{p2}, \dots, \theta_{pn}] \quad (74)$$

where n represents the number of inflow conditions. In this case there are $2n$ outputs corresponding to the thrust and torque at all n inflow conditions. For 100 inflow conditions the Jacobian is of size 200×337 . However, this time there is significant sparsity. The thrust at inflow condition j only depends only inflow conditions j ($V_{\infty j}, \Omega_j, \theta_{pj}$). Thus, the structure of the Jacobian looks like that shown in Fig. 14a. For simplicity only one geometric variable (c) and one inflow variable (V) is illustrated. The structure is simply repeated with the additional variables. This is an advantageous structure as graph coloring theory [19] allows us to collapse the Jacobian structure into the sparse representation shown in Fig. 14b

³ <https://github.com/JuliaDiff/ReverseDiff.jl>



(a) Jacobian sparsity pattern considering the geometric variable c and the inflow variable V .



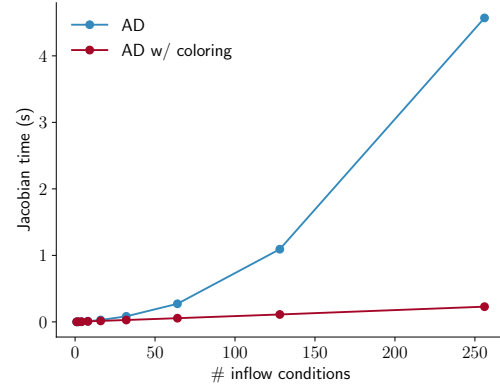
(b) Compressed form of the Jacobian.

Fig. 14: An illustration of the sparsity pattern for many BEM problems and how we can leverage graph coloring.

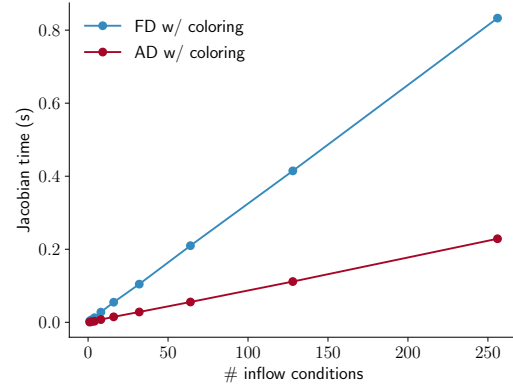
Thus, for our example Jacobian of size 200×337 we would only need to evaluate 40 forward passes, the same as our original problem that had only one inflow condition. We use the package `SparseDiffTools.jl`⁴ to compute the coloring vector and compute the Jacobian using forward mode AD, taking advantage of the graph coloring. To illustrate the benefit of this approach we compare the time to construct the Jacobian using forward mode AD, versus the time to construct the Jacobian using forward mode AD and graph coloring, for problems with increasing numbers of inflow conditions for $m \in [1, 2, 4, 8, 16, 32, 64, 128, 256]$. The timing is computing using `BenchmarkTools.jl`⁵, which runs multiple samples in order to produce consistent performance predictions. All Jacobians and intermediate vectors are cached so that computations are done in place to reduce memory usage. The results of the comparison are shown in Fig. 15a.

⁴ <https://github.com/JuliaDiff/SparseDiffTools.jl>

⁵ <https://github.com/JuliaCI/BenchmarkTools.jl>



(a) Forward mode AD compared to forward mode AD with graph coloring.



(b) Finite differencing with graph coloring as compared to AD with graph coloring.

Fig. 15: The time required to compute the Jacobian as a function of the number of inflow conditions.

For a small number of inflow conditions there is no advantage to using graph coloring, the timings are essentially identical. After 16 inflow conditions and beyond, for this problem, it is advantageous to use the graph coloring approach. For large number of inflow conditions the time savings exceeds an order of magnitude. This is a tremendous speed advantage that requires only a very small change in the way the Jacobian is computed.

While we emphasize AD, the same trends are realized with finite differencing. Finite differencing can leverage the graph coloring techniques in exactly the same way. Both require the same number of forward operations, although algorithmic differentiation has the additional flexibility that it could be used in reverse mode. Figure 15b shows a comparison using the sparse formulation for both AD and forward finite differencing

with the package `FiniteDiff.jl`⁶. AD provides about a 4 times speed advantage (about the same factor if both use dense methods). The real advantage of AD isn't the speed but the accuracy. In this case finite differencing only achieves about 7 digits of accuracy out of 15, whereas the derivatives of AD are exact within machine precision.

6 Optimization

As a demonstration of these considerations we perform a wind turbine blade optimization comparing different strategies. The goal of the optimization is not to design a blade, indeed blade optimization is not particularly meaningful unless other multidisciplinary considerations are also included (e.g., structures, acoustics). The goal is to illustrate the impact of the considerations discussed in this paper and the types of challenges encountered in blade optimization with blade element momentum theory.

The baseline design is the NREL 5-MW reference turbine [36], which has 17 radial stations. The optimization objective is to maximize the annual energy production by changing the blade chord distribution, twist distribution, tip-speed ratio in Region 2 of the power curve, and the pitch at 80 wind speeds from the cut-in wind speed to the cut-out wind speed. The annual energy production is computed assuming a Rayleigh distribution with a mean wind speed of 6 m/s. To reduce the dimensionality, chord and twist are specified only at a small number of radial locations and Akima splines are used to generate continuous distributions along the blades. Chord is specified at five radial stations: (0%, 25%, 50%, 75%, and 100%) and twist is specified at four: (11.1%, 40%, 70%, and 100% along the blade). The twist specification starts at 11.1% along the blade because inboard of that radius the blade uses cylindrical sections where the twist is meaningless. Eighty points is generally more than sufficient to resolve the power curve, but is a small number for a trajectory optimization, and so this is representative of a medium sized problem (90 design variables).

The first set of constraints are that the power at each wind speed must stay below rated power, which is essentially a constraint on maximum torque and is primarily met by feathering the blades (increasing pitch). The second set of constraints is that the thrust at each wind speed must stay below a specified level. The level (600 kN) is chosen to be below the 5-MW reference turbine so that the constraint will be active. The chord at the root is constrained to not decrease below its initial

value (3.542 m) in order to facilitate connection to the hub. A maximum rotation speed of 12 RPM is specified, but this can be imposed in the analysis without requiring additional constraints. The turbine rotates at the tip-speed ratio specified by the optimization, until it reaches the maximum allowed speed, after which it remains at that max speed through the remainder of the power curve. As a surrogate for a structural stress constraint, the flapwise loads are constrained near rated speed where the loads are highest. Again, the constraint is imposed at a level (6,500 N/m) below that of the baseline design so that the constraint will likely be active. Finally, a constraint is added to force the pitch to increase throughout the power curve, which should happen anyway. This is not a strictly necessary constraint but helps with the numerical behavior. Sometimes there is a low and a high pitch solution that can satisfy the power constraint, and because each pitch variable is independent the solution could contain large jumps in pitch in nearby speeds. Such large changes wouldn't be desirable for a controller. Alternatively, pitch could be nested with a root solver in the analysis, but was easier to impose in the optimization for this case. The problem is specified as follows:

$$\begin{aligned}
 & \text{maximize} && \text{annual energy production} \\
 & \text{by varying} && \text{chord}_i \text{ for } i = 1 \dots 5 \\
 & && \text{twist}_i \text{ for } i = 1 \dots 4 \\
 & && \text{tip-speed ratio} \\
 & && \text{pitch}_i \text{ for } i = 1 \dots 80 \\
 & \text{subject to} && \text{power}_i \leq 5 \text{ MW for } i = 1 \dots 80 \\
 & && \text{thrust}_i \leq 600 \text{ kN for } i = 1 \dots 80 \\
 & && \text{flapwise loads}_j \leq 6500 \text{ N/m for } j = 1 \dots 17 \\
 & && \text{pitch}_{i+1} > \text{pitch}_i \text{ for } i = 1 \dots 79
 \end{aligned} \tag{75}$$

where the 80 (and 79) corresponds to the wind speeds, and 17 corresponds to the number of radial locations. The problem has 90 design variables and 256 constraints. The objective and all constraints were normalized so that their order of magnitude was approximately unity.

Five different strategies were used to optimize the problem. All five methods used in-place operations for populating the Jacobian so that new arrays were not allocated at each iteration.

1. The traditional two-residual formulation was used. All of the physics calculations were exactly the same in all cases, only the solver was changed as discussed in Sect. 4. Total derivatives were obtained through finite differencing.

⁶ <https://github.com/JuliaDiff/FiniteDiff.jl>

2. The remaining methods used the one-dimensional residual, with Brent’s method, but differ in the way derivatives are obtained. This second case used the same finite differencing procedure as the previous case.
3. Total derivatives were obtained using forward mode algorithmic differentiation with the dual numbers propagated through the solver.
4. Total derivatives were obtained using an implicit analytic method (there is no distinction between direct/adjoint as the residual is 1-dimensional) where forward mode algorithmic differentiation provides the partial derivatives of the residual.
5. Total derivatives were obtained using forward mode algorithmic differentiation with graph coloring. The sparsity pattern for this problem is very similar to Fig. 14a, but with one important difference. The annual energy production is affected by every variable and so there is one row that is completely dense. Thus, if all derivatives were computed together, the sparse formulation would not be reducible in either a forward or reverse mode. There would still be some benefit as the sparse formulation requires less storage and less multiplications inside the optimizer, but the benefit was negligible for the sizes of problems tested. To address this issue, we compute the derivatives of all constraints using the sparse formulation, but the objective gradient (annual energy production) is computed separately. Generally, this could be computed efficiently using reverse mode AD as there is only one output and many inputs. However, in this case the AEP is just a weighted sum of powers, for which we already have derivatives, and so the objective derivative was most efficiently provided analytically. More generally bi-directional coloring schemes can be used for these problems that have dense rows *and* dense columns [12]. Gray et al. have demonstrated the effectiveness of this approach in computing total derivatives across large multidisciplinary problems [23].

We used SNOPT [20], a well-known sequential quadratic programming algorithm, as the optimizer with default options. The timing as reported by SNOPT is summarized in Tab. 4. The solution for each case was compared and was identical within the tolerances.

Switching from the traditional residual formulation to the one-equation formulation reduced computation by about 40%. That isn’t too surprising as solving the one dimensional residuals typically requires about half the number of iterations as solving the two dimensional residuals. A major difference was the number of convergence failures. The one-residual method never failed, but the traditional two-residual method failed frequently

Table 4: Comparison between time to solve optimization problem between the five different approaches.

Method	Time (s)
Traditional with FD	112
One Residual with FD	67
One Residual with AD through solver	24
One Residual with AD/analytic	12
One Residual with Sparse AD	5

during the optimization process. Fortunately, the optimization process was robust and so it still reliably converged to the optimum. Perhaps the main benefit to switching to the one equation model is increased convergence robustness, and tighter tolerances. The two-equation residual cannot in general be converged to tight tolerances at every iteration, which makes it harder to proceed to the subsequent cases using algorithmic differentiation throughout the full analysis.

Switching from finite differencing to numerically exact derivatives improved the convergence speed by an additional factor of three. Note that the AD used is a forward mode so the advantage in time savings is not because of scaling differences between the number of inputs and outputs (both scale with the number of inputs), but because of efficiencies in computing derivatives with AD, and efficiencies in the optimization made possible by more accurate derivatives. Furthermore, using AD only for the partials, and then solving for total derivatives with an implicit analytic approach increased convergence by another factor of two.

Finally, switching to sparse derivatives allowed for an additional 2.5 times speed up, allowing the optimization to be solved in a surprisingly quick 5 seconds. Note that for the sparse case there was no real difference between using AD through the solver, or using AD with an adjoint. The derivative computation was no longer a bottleneck either way. All together, the final method was over 22 times faster than the first method, and the only changes were in the solver used for converting the residuals and the methods used to compute the derivatives. Across the last four cases, the speed up was about 13 times and the only change was in how the Jacobian was computed.

These timings will of course vary from problem to problem, and even for the same problem the optimization path may vary leading to timing differences that should be averaged across many optimizations. The purpose of this analysis is not to dissect specific numbers, but rather to illustrate that small changes in how the residuals are converged and in how derivatives are obtained can have a large impact on optimization performance. This type of performance benefit is enabled by

enabling differentiability upfront during the derivation and implementation of the methodology.

7 No Inflow in One Direction

As discussed in Sec. 3, no inflow in one direction occurs in applications of interest like a hovering rotor, or a parked rotor. While using exactly zero does not work with the blade element momentum equations, one can use a small number for the velocities (e.g., $V_x = 10^{-4}$). However, it remains to be seen whether that leads to numerically accurate values as $V_x \rightarrow 0$ (not to mention the consideration of whether or not the assumptions of blade element momentum theory are even applicable at these extremes). Sec. 3 contains rederivation of the methodology so that zero inflow can be handled robustly.

7.1 Validation

Before comparing these rederived equations against the general formula with small (but nonzero) inflow velocities, we explore a few validation cases. We compare this formulation against experimental data for a hovering rotor with an untwisted blade [62]. The study used a Reynolds number correction based on results from a previous study by Lim et al. [42]. We apply the same correction for the lower Reynolds number of these tests, namely an increase in drag coefficient of 0.014 for the NACA 0012.

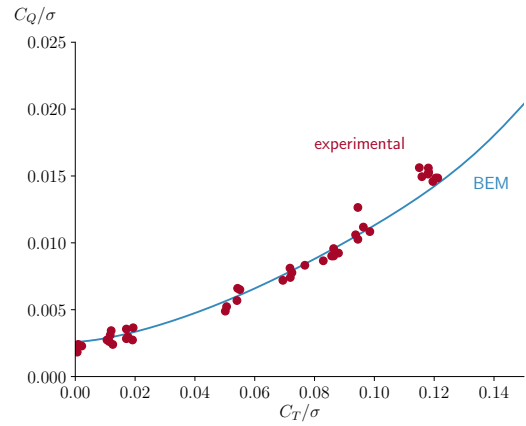
Rather than use the standard Prandtl tip correction we use the method discussed by Johnson [35] for rotorcraft applications. The methodology simply assumes that sections near the tip produce no lift, and only produce drag. Because there is no lift for those few sections near the tip, there is no induction, the residual equation is bypassed, and the torque is computed directly from blade element theory. This tip loss is only applied at sections for which [67]:

$$\frac{r}{R_{tip}} \geq 1 - \frac{2}{3} \frac{C_{(ssr/R_{tip}=0.7)}}{R_{tip}} \quad (76)$$

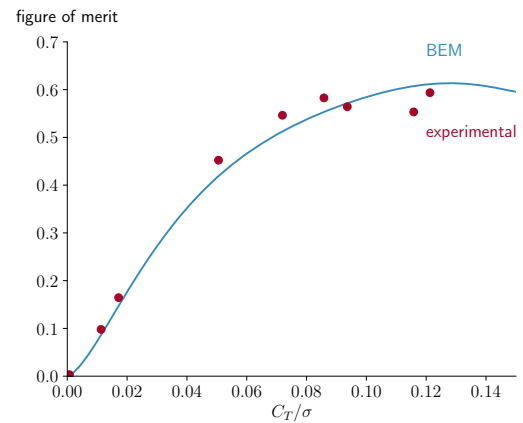
For the rotor studied in this section this corresponds to $r/R_{tip} \geq 0.94$. Using the standard tip correction produces essentially the same results except at the high power end of the curve where the thrust loads are somewhat overpredicted.

Figure 16 compares the thrust coefficient and torque coefficients normalized by solidity, and the figure of merit, which is a measure of hover efficiency. These plots are the same as those in the original study [62].

Of note the BEM compares quite favorably to experimental data, at least for these integral quantities, and avoids some of the spurious jumps shown in the cited study (although the BEM was not the focus of that paper). The study also provides wake velocities, which are compared in Fig. 17, also in the same style as those shown in the original study. Here we see that agreement is reasonable except at the tip where the simple tip loss function forces these velocities to zero. The wake from a rotor in hover plays a significant role in affecting the blade loading, particularly near the tip. The BEM assumption of streamtubes are less justifiable in these recirculating regions. Better tip corrections, or tip corrections tuned for hover scenarios could be used to offer improved loading and wake distributions, but is not the focus of this work.



(a) Thrust coefficient versus torque coefficient.



(b) Thrust coefficient versus figure of merit (efficiency).

Fig. 16: Comparison of integrated loads between BEM and experimental data for a rotor in hover.

Our main interest is understanding the differences between the original BEM and the reformulation dis-

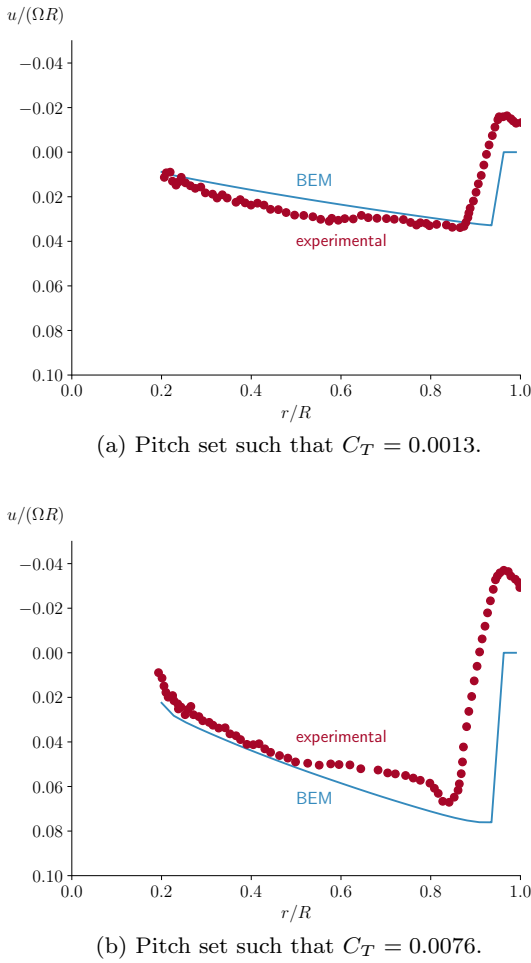


Fig. 17: Comparison of normalized axial induced velocities along radius between BEM and experimental data for a rotor in hover.

cussed in this section that allow for V_x to be exactly zero. For the original case we specified the inflow velocity as a small nonzero value: $V_x = 10^{-4}$. Figure 18 shows the torque coefficient as a function of pitch angle, for the two cases. This is the same data as in the previous figures, only shown as a function of pitch rather than thrust coefficient to more easily see the differences. We see that the original formulation follows the same trend, but occasionally jumps to another solution because of the poor numerical formulation (this is true regardless which tip correction method is used). Because V_x is very small, but nonzero, the axial induction factor is very large. Additionally, as the pitch goes to zero, the torque erroneously jumps to zero for this untwisted blade. In contrast the reformulated version maintains smooth and consistent convergence across the range of pitch angles. For pure analysis, using the original version with a small V_x may be sufficient, but for use in op-

timization, particularly with gradient-based methods, the smooth and reliable convergence of this reformulation may be needed.

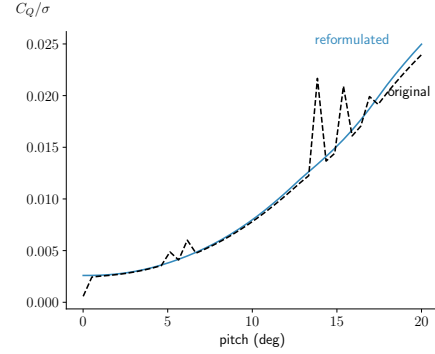


Fig. 18: Comparison between the original BEM formulation with small inflow velocity $V_x = 10^{-4}$ versus the reformulated version that allows for $V_x = 0$ and leads to improved numerical convergence.

For optimizations in hover, or rotors with brakes preventing rotation, the same considerations for converging the residual and computing derivatives from the previous sections would apply. The revised formulation derived in this section is recommended, rather than using the standard formulation with negligibly small inflow velocities, to avoid the convergence challenges highlighted in this section. However, as also highlighted in this section, the assumptions of BEM theory may be less defensible in hover than they are for forward flight, and the accuracy of the results, especially near the tip, may be less than satisfactory without additional correction models specific to these applications.

8 Conclusion

Blade element methods (BEM) are widely used for initial aerodynamic analysis of wind turbine and propeller blades. As optimization problems become increasingly complex, especially with large numbers of design variables, weaknesses in the way the BEM equations are typically solved and the way derivatives are obtained become more limiting. In particular, the standard formulation requires the coupled solution of a pair of residual equations, often with fixed point iteration, Newton's method, or other more sophisticated multidimensional root solvers. Such methods work very well for analysis, but are prone to occasional convergence difficulties, particularly during an optimization. The exploratory nature of optimization sometimes creates input combinations with solutions far from typical starting points

in the root solver. Thus, occasional convergence failures are produced with solutions diverging, or solutions terminating in a local valley that does not contain a root. The latter case can often occur for blade sections operating near stall, because of the reversal in the lift coefficient, as shown in this paper.

There exists a wide range of proposed correction models to improve the accuracy of the physics models, but common to them all a pair of residual equations must be solved ensuring compatibility between the blade element and the momentum formulations. As shown in this paper, by rearranging the sequence of computations the pair of residuals can be recast as a single residual that is only a function of the inflow angle. This reformulation provides a significant benefit because there are one-dimensional root finders with guaranteed convergence, and they are not susceptible to premature termination on nonmonotonic functions like the stall case mentioned earlier. Furthermore, one-dimensional root solvers can be converged reliably to machine precision, facilitating computing of more accurate gradients, especially when using algorithmic differentiation (AD).

Algorithmic differentiation was applied to the BEM formulation. Using a dual number approach for AD required only minimal changes to the code while allowing for exact derivatives (approximately doubling the number of significant digits as compared to finite differencing). Additionally, the time to solve optimization problems is greatly reduced, both because of faster gradient calculations and the improved accuracy of the derivatives allow for more efficient optimization convergence. Furthermore, many BEM optimization problems exhibit significant sparsity in the Jacobians. For example, trajectory optimization problems with propellers, or power curve evaluations for turbines, involve computing performance at multiple independent conditions. Using graph coloring, the Jacobians can be computed much faster than for standard dense formulations. For even moderate sized problems, the speed up in derivative computational time can exceed an order of magnitude. An example optimization was shown highlighting more than an order of magnitude increase in optimization speed with no changes in the physics, only changes in the way the residuals are converged and how the derivatives are computed.

Finally, for cases with zero inflow in one direction, like a hovering rotor or a parked rotor, new residuals were derived. The standard formulation is undefined exactly at zero. While small nonzero inputs can be used, especially for analysis, the formulation is more prone to erroneous jumps in the outputs that are problematic for optimization. New residuals were derived that con-

verge reliably, and show a nice theoretical fit with the standard equations filling in existing singularities. Although this formulation allows for clear improvements in optimization usage, the physics of hovering rotors using blade element momentum theory may be less accurate without additional corrections. The recirculation of tip vortices stretches the streamtube assumptions and leads to higher errors in the tip regions of the blades. The scope of this paper focused on optimization aspects (solvers and derivatives), rather than the physics, but future work in improving BEM implementations for hover would be useful. A particular need for these conditions exist with emerging propeller-driven eVTOL aircraft that move through a range of conditions transitioning from hover to forward flight.

Acknowledgements This research was partially supported by the Department of Energy (DOE) Advanced Research Projects Agency-Energy (ARPA-E) Program award DE-AR0001186 entitled “Computationally Efficient Control Co-Design Optimization Framework with Mixed-Fidelity Fluid and Structure Analysis.”

Replication of Results

A full implementation of the methodology in the Julia programming language, with documentation, is available in an open source repository called CCBlade (<https://github.com/byuflowlab/CCBlade.jl>). The scripts used to generate the results and figures from the paper are available in separate repository (<https://github.com/byuflowlab/ning2020-bem>).

Conflict of Interest

The author states that there is no conflict of interest.

References

1. Akima, H.: A new method of interpolation and smooth curve fitting based on local procedures. *Journal of the ACM (JACM)* **17**(4), 589–602 (1970). DOI 10.1145/321607.321609. URL <http://dx.doi.org/10.1145/321607.321609>
2. Albanesi, A.E., Peralta, I., Bre, F., Storti, B.A., Fachinotti, V.D.: An optimization method based on the evolutionary and topology approaches to reduce the mass of composite wind turbine blades. *Structural and Multidisciplinary Optimization* **62**(2), 619–643 (2020). DOI 10.1007/s00158-020-02518-2. URL <http://dx.doi.org/10.1007/s00158-020-02518-2>
3. Alefeld, G.E., Potra, F.A., Shi, Y.: Algorithm 748: Enclosing zeros of continuous functions. *ACM Transactions on Mathematical Software (TOMS)* **21**(3), 327–344 (1995). DOI 10.1145/210089.210111. URL <http://dx.doi.org/10.1145/210089.210111>

4. Bak, C., Johansen, J., Andersen, P.: Three-dimensional corrections of airfoil characteristics based on pressure distributions. In: European Wind Energy Conference & Exhibition. Athens, Greece (2006)
5. Bohorquez, F., Pines, D., Samuel, P.D.: Small rotor design optimization using blade element momentum theory and hover tests. *Journal of Aircraft* **47**(1), 268–283 (2010). DOI 10.2514/1.45301. URL <http://dx.doi.org/10.2514/1.45301>
6. Branlard, E., Gaunaa, M.: Development of new tip-loss corrections based on vortex theory and vortex methods. *Journal of Physics: Conference Series* **555**, 012012 (2014). DOI 10.1088/1742-6596/555/1/012012. URL <http://dx.doi.org/10.1088/1742-6596/555/1/012012>
7. Brent, R.P.: An algorithm with guaranteed convergence for finding a zero of a function. *The Computer Journal* **14**(4), 422–425 (1971). DOI 10.1093/comjnl/14.4.422
8. Buhl Jr., M.L.: A new empirical relationship between thrust coefficient and induction factor for the turbulent windmill state. Tech. Rep. NREL/TP-500-36834, National Renewable Energy Laboratory (2005)
9. Burton, T., Jenkins, N., Sharpe, D., Bossanyi, E.: *Wind Energy Handbook*, 2nd edn. Wiley, United Kingdom (2011)
10. Chen, J., Wang, Q., Shen, W.Z., Pang, X., Li, S., Guo, X.: Structural optimization study of composite wind turbine blade. *Materials & Design* **46**, 247–255 (2013). DOI 10.1016/j.matdes.2012.10.036. URL <http://dx.doi.org/10.1016/j.matdes.2012.10.036>
11. Coleman, R.P., Feingold, A.M., Stempin, C.W.: Evaluation of the induced-velocity field of an idealized helicopter rotor. Tech. Rep. ARR L5E10, NACA (1945)
12. Coleman, T.F., Verma, A.: The efficient computation of sparse jacobian matrices using automatic differentiation. *SIAM Journal on Scientific Computing* **19**(4), 1210–1233 (1998). DOI 10.1137/s1064827595295349. URL <http://dx.doi.org/10.1137/s1064827595295349>
13. Corrigan, J.J., Schillings, J.J.: Empirical model for stall delay due to rotation. In: American Helicopter Society Aeromechanics Specialists Conference. San Francisco, CA (1994)
14. Dekker, T.J.: Finding a zero by means of successive linear interpolation. *Constructive aspects of the fundamental theorem of algebra* pp. 37–51 (1969)
15. Døssing, M., Madsen, H.A., Bak, C.: Aerodynamic optimization of wind turbine rotors using a blade element momentum method with corrections for wake rotation and expansion. *Wind Energy* **15**(4), 563–574 (2011). DOI 10.1002/we.487. URL <http://dx.doi.org/10.1002/we.487>
16. Drees, J.M.: A theory of airflow through rotors and its application to some helicopter problems. *Journal of the helicopter association of great britain* **3**(2), 79–104 (1949)
17. Du, Z., Selig, M.: A 3-D stall-delay model for horizontal axis wind turbine performance prediction. In: 1998 ASME Wind Energy Symposium, AIAA-1998-21 (1998)
18. Eggers Jr, A.J., Chaney, K., Digumarthi, R.: An assessment of approximate modeling of aerodynamic loads on the uae rotor. In: 41st Aerospace Sciences Meeting and Exhibit, AIAA-2003-0868 (2003). DOI 10.2514/6.2003-868
19. Gebremedhin, A.H., Manne, F., Pothén, A.: What color is your jacobian? graph coloring for computing derivatives. *SIAM Review* **47**(4), 629–705 (2005). DOI 10.1137/s0036144504444711. URL <http://dx.doi.org/10.1137/s0036144504444711>
20. Gill, P.E., Murray, W., Saunders, M.a.: SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM review* **47**(1), 99–131 (2005)
21. Glauert, H.: Airplane propellers. In: *Aerodynamic Theory*, pp. 169–360. Springer (1935)
22. Glauert, H., Committee, A.R.: *A General Theory of the Autogyro*. HM Stationery Office (1926)
23. Gray, J.S., Hearn, T.A., Naylor, B.A.: Using graph coloring to compute total derivatives more efficiently in openmdao. *AIAA Aviation 2019 Forum* (2019). DOI 10.2514/6.2019-3108. URL <http://dx.doi.org/10.2514/6.2019-3108>
24. Gray, J.S., Hwang, J.T., Martins, J.R.R.A., Moore, K.T., Naylor, B.A.: Openmdao: an open-source framework for multidisciplinary design, analysis, and optimization. *Structural and Multidisciplinary Optimization* **59**(4), 1075–1104 (2019). DOI 10.1007/s00158-019-02211-z. URL <http://dx.doi.org/10.1007/s00158-019-02211-z>
25. Gur, O., Rosen, A.: Optimization of propeller based propulsion system. *Journal of Aircraft* **46**(1), 95–106 (2009). DOI 10.2514/1.36055. URL <http://dx.doi.org/10.2514/1.36055>
26. Hansen, M., Gaunaa, M., Madsen, H.: A beddoes-leishman type dynamic stall model in state-space and indicial formulations. Risø-R-1354, Risø National Laboratory (2004)
27. Hansen, M.O.L.: *Aerodynamics of Wind Turbines*, 2nd edn. Earthscan, United Kingdom (2008)
28. He, C.J., Peters, D.A.: Optimization of rotor blades for combined structural, dynamic, and aerodynamic properties. *Structural Optimization* **5**(1-2), 37–44 (1992). DOI 10.1007/bf01744694. URL <http://dx.doi.org/10.1007/bf01744694>
29. Hendricks, E.S., Falck, R.D., Gray, J.S., Aretskin-Hariton, E., Ingraham, D., Chapman, J.W., Schnulo, S.L., Chin, J., Jasa, J.P., Bergeson, J.D.: Multidisciplinary optimization of a turboelectric tiltwing urban air mobility aircraft. *AIAA Aviation 2019 Forum* (2019). DOI 10.2514/6.2019-3551. URL <http://dx.doi.org/10.2514/6.2019-3551>
30. Hjort, S.: Non-empirical bem corrections relating to angular and axial momentum conservation. *Energies* **12**(2), 320 (2019). DOI 10.3390/en12020320. URL <http://dx.doi.org/10.3390/en12020320>
31. Hwang, B., Lee, S., Lee, S.: Optimization of a counter-rotating wind turbine using the blade element and momentum theory. *Journal of Renewable and Sustainable Energy* **5**(5), 052013 (2013). DOI 10.1063/1.4826940. URL <http://dx.doi.org/10.1063/1.4826940>
32. Hwang, J.T., Ning, A.: Large-scale multidisciplinary optimization of an electric aircraft for on-demand mobility. In: *AIAA Structures, Structural Dynamics, and Materials Conference*. Kissimmee, FL (2018). DOI 10.2514/6.2018-1384
33. Ingraham, D., Gray, J.S., Lopes, L.V.: Gradient-based propeller optimization with acoustic constraints. *AIAA Scitech 2019 Forum* (2019). DOI 10.2514/6.2019-1219. URL <http://dx.doi.org/10.2514/6.2019-1219>
34. Jasa, J.P., Hwang, J.T., Martins, J.R.R.A.: Open-source coupled aerostructural optimization using python. *Structural and Multidisciplinary Optimization* **57**(4), 1815–1827 (2018). DOI 10.1007/s00158-018-1912-8. URL <http://dx.doi.org/10.1007/s00158-018-1912-8>
35. Johnson, W.: *Helicopter Theory*. Courier Corporation (2012)
36. Jonkman, J., Butterfield, S., Musial, W., Scott, G.: Definition of a 5-MW reference wind turbine for offshore system development. NREL/TP-500-38060, National Renewable Energy Laboratory, Golden, CO (2009)

37. Kenway, G., Martins, J.: Aerostructural shape optimization of wind turbine blades considering site-specific winds. In: 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (2008). DOI 10.2514/6.2008-6025
38. Kwon, H.I., Yi, S., Choi, S., Kim, K.: Design of efficient propellers using variable-fidelity aerodynamic analysis and multilevel optimization. *Journal of Propulsion and Power* **31**(4), 1057–1072 (2015). DOI 10.2514/1.b35097. URL <http://dx.doi.org/10.2514/1.b35097>
39. Larsen, J., Nielsen, S., Krenk, S.: Dynamic stall model for wind turbine airfoils. *Journal of Fluids and Structures* **23**(7), 959–982 (2007). DOI 10.1016/j.jfluidstructs.2007.02.005. URL <http://dx.doi.org/10.1016/j.jfluidstructs.2007.02.005>
40. Leishman, J.G.: Validation of approximate indicial aerodynamic functions for two-dimensional subsonic flow. *Journal of Aircraft* **25**(10), 914–922 (1988). DOI 10.2514/3.45680. URL <http://dx.doi.org/10.2514/3.45680>
41. Leishman, J.G., Beddoes, T.S.: A semi-empirical model for dynamic stall. *Journal of the American Helicopter Society* **34**(3), 3–17 (1989). DOI 10.4050/jahs.34.3. URL <http://dx.doi.org/10.4050/jahs.34.3>
42. Lim, J.W., McAlister, K.W., Johnson, W.: Hover performance correlation for full-scale and model-scale coaxial rotors. *Journal of the American Helicopter Society* **54**(3), 32005–3200514 (2009). DOI 10.4050/jahs.54.032005. URL <http://dx.doi.org/10.4050/jahs.54.032005>
43. Lindenbarg, C.: Modeling of rotational augmentation based on engineering considerations and measurements. In: European Wind Energy Conference. London (2004)
44. Madsen, H.A., Bak, C., Døssing, M., Mikkelsen, R., Øye, S.: Validation and modification of the blade element momentum theory based on comparisons with actuator disc simulations. *Wind Energy* **13**(4), 373–389 (2010). DOI 10.1002/we.359. URL <http://dx.doi.org/10.1002/we.359>
45. Mangler, K.W., Squire, H.B.: The induced velocity field of a rotor. *ARC R&M* **2642** (1950)
46. Maniaci, D.C.: An investigation of WT_Pref convergence issues. In: AIAA Aerospace Sciences Meeting, AIAA 2011-150 (2011)
47. Manwell, J.F., McGowan, J.G., Rogers, A.L.: *Wind Energy Explained*, 2nd edn. Wiley, United Kingdom (2009)
48. McCrink, M.H., Gregory, J.W.: Blade element momentum modeling of low-reynolds electric propulsion systems. *Journal of Aircraft* **54**(1), 163–176 (2017). DOI 10.2514/1.c033622. URL <http://dx.doi.org/10.2514/1.c033622>
49. McWilliam, M., Crawford, C.: The behavior of fixed point iteration and Newton-Raphson methods in solving the blade element momentum equations. *Wind Engineering* **35**(1), 17–32 (2011). DOI 10.1260/0309-524X.35.1.17
50. Montgomerie, B.: Methods for root effects, tip effects and extending the angle of attack range to $\pm 180^\circ$, with application to aerodynamics for blades on wind turbines and propellers. FOI-R-1305-SE, Swedish Defence Research Agency, Stockholm (2004)
51. Moore, K., Ning, A.: Takeoff and performance tradeoffs of retrofit distributed electric propulsion for urban transport. *Journal of Aircraft* **56**(5), 1880–1892 (2019). DOI 10.2514/1.C035321
52. Ning, A.: A simple solution method for the blade element momentum equations with guaranteed convergence. *Wind Energy* **17**(9), 1327–1345 (2014). DOI 10.1002/we.1636. URL <http://dx.doi.org/10.1002/we.1636>
53. Ning, A., Hayman, G., Damiani, R., Jonkman, J.: Development and validation of a new blade element momentum skewed-wake model within AeroDyn. In: 33rd ASME Wind Energy Symposium. Kissimmee, FL (2015). DOI 10.2514/6.2015-0215
54. Øye, S.: Dynamic stall simulated as time lag of separation. In: IEA symposium on the aerodynamics of wind turbines (1990)
55. Øye, S.: Induced velocities for rotors in yaw. In: Proceedings of the Sixth IEA Symposium. ECN, Petten, Holland (1992)
56. Peters, D.A., Boyd, D.D., He, C.J.: Finite-state induced-flow model for rotors in hover and forward flight. *Journal of the American Helicopter Society* **34**(4), 5 (1989). DOI 10.4050/jahs.34.5. URL <http://dx.doi.org/10.4050/jahs.34.5>
57. Pitt, D.M., Peters, D.A.: Theoretical prediction of dynamic-inflow derivatives. *Vertica* **5**(1), 21–34 (1981)
58. Polat, O., Tuncer, I.H.: Aerodynamic shape optimization of wind turbine blades using a parallel genetic algorithm. *Procedia Engineering* **61**, 28–31 (2013). DOI 10.1016/j.proeng.2013.07.088. URL <http://dx.doi.org/10.1016/j.proeng.2013.07.088>
59. Ponta, F.L., Otero, A.D., Lago, L.I., Rajan, A.: Effects of rotor deformation in wind-turbine performance: The dynamic rotor deformation blade element momentum model (drd-bem). *Renewable Energy* **92**, 157–170 (2016). DOI 10.1016/j.renene.2016.01.098. URL <http://dx.doi.org/10.1016/j.renene.2016.01.098>
60. Prandtl, L., Betz, A.: Vier abhandlungen zur hydrodynamik und aerodynamik. In: Göttinger Nachr (1927)
61. Rajan, A., Ponta, F.: A novel correlation model for horizontal axis wind turbines operating at high-interference flow regimes. *Energies* **12**(6), 1148 (2019). DOI 10.3390/en12061148. URL <http://dx.doi.org/10.3390/en12061148>
62. Ramasamy, M., Gold, N.P., Bhagwat, M.J.: Rotor hover performance and flowfield measurements with untwisted and highly-twisted blades. In: European Rotorcraft Forum. Paris, France (2010)
63. Revels, J., Lubin, M., Papamarkou, T.: Forward-Mode Automatic Differentiation in Julia. ArXiv e-prints (2016)
64. Rodrigues, S.a.S., Marta, A.C.: On addressing noise constraints in the design of wind turbine blades. *Structural and Multidisciplinary Optimization* **50**(3), 489–503 (2014). DOI 10.1007/s00158-014-1072-4. URL <http://dx.doi.org/10.1007/s00158-014-1072-4>
65. Schmitz, S.: *Aerodynamics of Wind Turbines: A Physical Basis for Analysis and Design*. John Wiley & Sons (2019)
66. Shen, W.Z., Mikkelsen, R., Sørensen, J.N., Bak, C.: Tip loss corrections for wind turbine computations. *Wind Energy* **8**(4), 457–475 (2005). DOI 10.1002/we.153. URL <http://dx.doi.org/10.1002/we.153>
67. Sissingh, G.: Contribution to the aerodynamics of rotating-wing aircraft. NACA-TM-921, NACA (1939)
68. Snel, H., Houwink, R., Bosschers, J.: Sectional prediction of lift coefficients on rotating wind turbine blades in stall. ECN-C-93-052, Netherlands Energy Research Foundation (ECN), Petten, Netherlands (1994)
69. Snel, H., Schepers, J., Nederland, S.E.C.: Joint investigation of dynamic inflow effects and implementation of an engineering method. Tech. Rep. ECN-C-94-107, Netherlands Energy Research Foundation (1995)
70. Sun, Z., Shen, W.Z., Chen, J., Zhu, W.J.: Improved fixed point iterative method for blade element momentum computations. *Wind Energy* **20**(9), 1585–1600

- (2017). DOI 10.1002/we.2110. URL <http://dx.doi.org/10.1002/we.2110>
71. Tran, C.T., Petot, D.: Semi-empirical model for the dynamic stall of airfoils in view of the application to the calculation of responses of a helicopter blade in forward flight. *Vertica* **5**(1), 35–53 (1981)
 72. Viterna, L.A., Janetzke, D.C.: Theoretical and experimental power from large horizontal-axis wind turbines. Tech. Rep. DOE/NASA/20320-4, National Aeronautics and Space Administration, Lewis Research Center (1982)
 73. de Vries, O.: Fluid dynamic aspects of wind energy conversion. Agard report ag-243, Advisory Group for Aerospace Research and Development (1979)
 74. Wilson, R.: Wind turbine technology: Fundamental concepts of wind turbine engineering. chap. Aerodynamic behavior of wind turbines. ASME (1994)
 75. Wilson, R., Lissaman, P.: Applied aerodynamics of wind power machines. Tech. rep., Oregon State University (1974)