49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition
4 - 7 January 2011, Orlando, Florida

AIAA 2011-150

# An Investigation of WT_Perf Convergence Issues

David C. Maniaci[1]
*The Pennsylvania State University,University Park, PA,16802*

**WT_Perf is a wind turbine performance prediction code released by the National Renewable Energy Laboratory (NREL). It is based on blade-element momentum (BEM) theory with capabilities to account for yawed and tilted operation as well as operation in the turbulent windmill and propeller brake states. Historically, for some cases which have a real solution, WT_Perf has had issues iterating to a converged solution. This paper will mainly focus on one of these convergence issues: a singularity at an axial induction factor of one. It will also briefly discuss a second convergence issue, the selection of the most logical solution when multiple solutions are possible. Although these issues are mentioned in previous literature, methods to solve them have not been previously published. These issues typically occur outside the normal operating conditions of a rotor, but were found to still cause convergence issues for cases which have real solutions in the normal operating range. Additionally, during startup and shutdown, a rotor will run through the operating range where these convergence issues occur. The convergence issues can also be challenging for optimization routines, which generate a large number of cases, some of which will run through the problematic operating ranges. Solutions to these problems have been implemented in the newest release of WT_Perf. An overview of the fluid-dynamic theory behind WT_Perf will be presented, in addition to an explanation of some of the newest features.**

## Nomenclature

| | | |
|---|---|---|
| $A_D$ | = | rotor disc area |
| a | = | axial induction factor |
| a' | = | tangential induction factor |
| B | = | number of blades |
| $C_d$ | = | sectional drag coefficient |
| $C_l$ | = | sectional lift coefficient |
| $C_p$ | = | sectional coefficient of power |
| $C_n$ | = | sectional normal force coefficient ($C_x$) |
| $C_t$ | = | sectional tangential force coefficient ($C_y$) |
| $C_{T_L}$ | = | local thrust coefficient (also referred to as DCT) |
| c | = | chord |
| F | = | Prandtl Tip-loss factor (also referred to as "Loss") |
| LSR | = | local speed ratio |
| R | = | total blade radius |
| r | = | radius of a spanwise station |
| TSR | = | $\frac{\Omega R}{U_\infty}$, tip speed ratio |
| $U_\infty$ | = | wind speed |
| $\Phi$ | = | local flow angle |
| $\Omega$ | = | angular velocity |
| $\alpha$ | = | angle of attack |
| $\beta$ | = | pitch angle between airfoil zero lift line and the rotor disc plane |
| $\rho$ | = | density |
| $\sigma$ | = | blade solidity |
| $\sigma_r$ | = | $\frac{Bc}{2r\pi}$ chord solidity, total blade chord (B * c) divided by circumferential length for a radial station |

[1] Ph.D. Candidate, Dept. of Aerospace Engineering, 229 Hammond Bldg., AIAA Student Member.



Copyright © 2011 by D. Maniaci. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

# I. Introduction

THIS paper summarizes changes made to the newest version of WT_Perf to address some convergence issues users have experienced with the code. WT_Perf is a steady-state horizontal axis wind turbine (HAWT) performance analysis tool maintained by the National Renewable Energy Laboratory;s National Wind Technology Center (NWTC). The code was created by M. Buhl of the NWTC and a brief description can be found in a paper by Buhl.[1] Versions of the code and the user's guide can be found on the NWTC Design Codes website.[2] The code is a descendant of the PROP code; the original theory is covered in several papers.[3,4,5]

WT_Perf is based on blade-element momentum theory (BEMT) with capabilities to account for yawed and tilted operation as well as operation in the turbulent windmill and propeller brake states. It has the option to account for swirl in the flow, including lift and optionally drag forces of the blade sectional elements. To find a solution to the BEMT equations, WT_Perf iterates between the forces produced on the rotor via blade element analysis and the forces produced via a modified momentum theory analysis.[6] In previous versions of WT_Perf, there have been cases where this iteration routine has not converged. This paper will give a general overview of how WT_Perf iterates in a solution to the BEMT equations. The paper will then explore some of the iteration convergence issues found in previous versions of WT_Perf and explain how these issues have been addressed in the newest release of WT_Perf.

## A. Summary of Blade Element Momentum theory

Two important terms in the solution of the BEMT equations are axial induction factor and tangential induction factor. Axial induction factor determines the amount of flow that is going through the rotor disc. Tangential induction factor is a measure of how much rotation is in the flow behind the rotor.

Blade Element Momentum Theory assumes that the force produced by a blade element is solely responsible for the rate of change of momentum of the fluid passing through the annulus created by the rotation of that blade element. The BEMT equations are set up by setting the thrust produced by a rotor equal to the rate of change of axial momentum; the resulting equation is mainly dependent upon the axial induction factor. Torque is set equal to the rate of change of angular momentum; the resulting equation is primarily dependant on the tangential induction factor. Blade element theory is used to find the forces and moments (thrust and torque) which the blade can produce based on the local flow angle of attack and the airfoil data. Momentum theory, along with empirical data, is used to balance the equations.

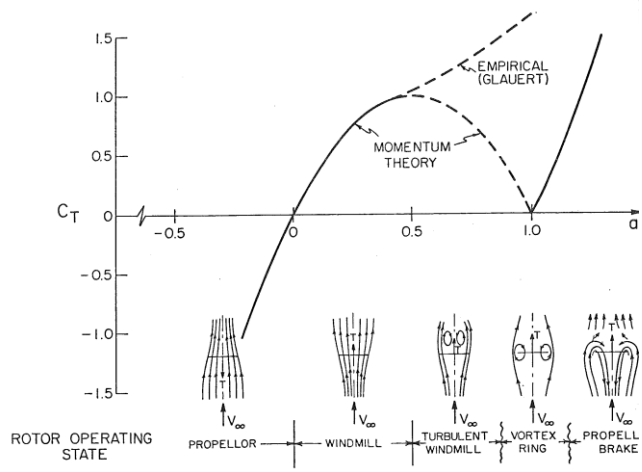


**Figure 1. Rotor operating states.** *Figure from Wilson.*[3]

## B. Momentum Theory and Empirical Data

Momentum theory under predicts the thrust of a rotor for values of axial induction factor greater than 0.5, where the rotor enters the turbulent wake state, shown in figure 1.[3] The turbulent wake state typically occurs at low wind speeds when the wake has large amounts of mixing with the outer flow. This viscous mixing causes streamlines to be not well defined and also causes inviscid assumptions to be no longer valid, driving the solution away from that predicted by classical momentum theory.[3] Empirical data, typically the Glauert empirical relationship, is used to account for the difference between measured thrust and thrust predicted by classical momentum theory. The resulting use of the combination of momentum theory for the windmill operating state and empirical data for the turbulent wake state is called modified momentum theory in this paper. A detailed description of the implementation of the Glauert empirical relationship in WT_Perf is described in the paper by Buhl.[6]

Near an axial induction factor of 1.0, the rotor enters the vortex ring state, where momentum is transferred via unsteady vortices. Blade element momentum theory is not valid in the vortex ring state because the axial induction factor changes drastically across the rotor disc, possibly changing sign due to recirculation from one part of the rotor to another.

## C. Blade Element Momentum Theory Iteration

An iteration routine is used to solve the axial and tangential velocities acting on a blade element, so that the forces on that element can be calculated based on airfoil lift and drag data. Iteration can be performed based on sectional thrust coefficient, lift coefficient or axial induction factor. WT_Perf iterates on axial induction factor and the tangential induction factor, both of which must be iterated on until convergence is reached. In the previous release of WT_Perf, the iteration loop for tangential induction factor was nestled within the iteration loop for axial induction factor. The newest release uses a two-dimensional Newton-Raphson iteration routine, iterating on both a and a' rather than separating them into two separate iteration loops. The physics are the same for both methods, and it was decided that explaining the method of the previous iteration method would make it easier to grasp the fundamental concepts.

The blade element iteration in WT_Perf begins with initial values for axial induction factor and tangential induction factor, as shown in figure 2 These values started as zero in the old version of WT_Perf; the newest version uses the converged value from the previous neighboring blade element to initialize the iteration. If no converged value is available, it begins the iteration with values of zero, as in the previous version of the code. In figure 2, the starting values are listed as 'Initial Guess a, a'". The blade element routine then calculates the local flow velocities and the LSR (local speed ratio).

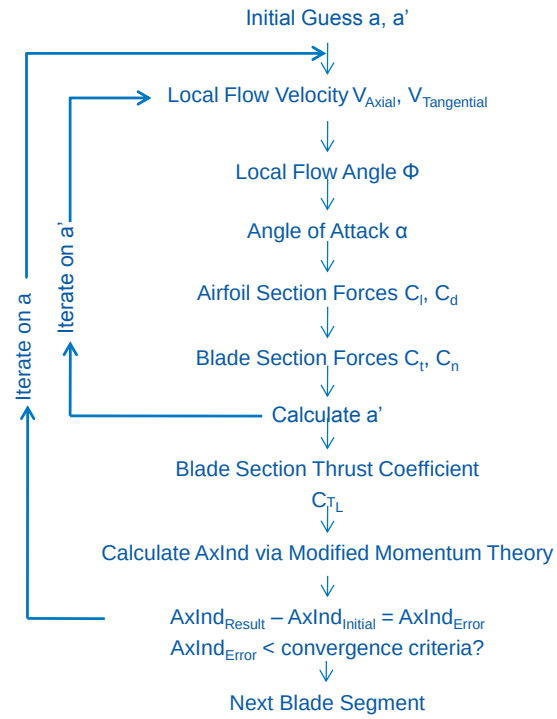


**Figure 2. Blade element momentum theory iteration routine.**

The LSR (local speed ratio) is the ratio of total tangential velocity on a blade element section divided by the total normal velocity on that section, where tangential means tangent to the circle inscribed by that blade element section and normal means normal to the rotor disc plane. If cone angle, tilt angle, yaw angle, and wind shear are ignored, LSR reduces to equation 1, which is like TSR (Tip Speed Ratio), except with the local radius of the blade element section of the current case.

$$LSR = \frac{\Omega r}{U_\infty} \tag{1}$$

The local flow angle can now be calculated based on a, a′, and the LSR, as shown in equation 2. During the tangential induction factor iteration loop, labeled "Iterate on a'" in figure 1, axial induction factor is held constant.

$$\Phi = \tan^{-1}\left(\frac{1-a}{(1+a')*LSR}\right) \tag{2}$$

Next, the local section angle of attack is found from the local flow angle and the local blade geometric pitch angle, as shown in equation 3.

$$\alpha = \Phi - \beta \tag{3}$$

The lift and drag coefficients are found from table lookup of the airfoil profile data as a function of the local angle of attack.

$$(C_l, C_d) = \text{function}(\alpha) \tag{4}$$

The local tangential and normal force coefficients are found, given $C_l$, $C_d$, and the local flow angle.

$$C_t = C_l \sin(\Phi) + C_d \cos(\Phi) \tag{5}$$

$$C_n = C_l \cos(\Phi) + C_d \sin(\Phi)$$

Given the local flow angle, the local tangential force coefficient and the value of the Prandtl Tip-loss function, a new value for a' can be calculated based on equation 6, which can be derived via blade element momentum theory.[7,8]

$$a' = 1.0 \Big/ \left( \frac{4.0 * Loss * \sin(\Phi) * \cos(\Phi)}{\sigma_r * C_t} - 1.0 \right) \tag{6}$$

The new value for a' is compared to the value from the previous timestep; if the difference is greater than a convergence criteria, the routine returns to start a new iteration. The new iteration begins by calculating new local flow velocities based on the new value for a'. Typically, only a few iterations are required until convergence for a' is reached. Once a' is considered converged, the code moves on to calculate the local thrust coefficient of the annular blade segment based on the normal force found from blade element theory and airfoil profile data, as shown in equation 7. Equation 7 is based on blade element theory ; a derivation of equation 7 can be found in references 4 and 8.

$$C_{T_L} = \frac{\sigma_r C_n (1-a)^2}{\sin^2 \Phi} \tag{7}$$

In summary, the value of axial induction factor at the beginning of an iteration is used to calculate the local flow angle, which is used to calculate the thrust on the blade element. Next, the thrust force from the previous step is used to calculate the axial induction factor required to produce that thrust force based on modified momentum theory. Modified momentum theory is defined as the standard momentum theory relation between thrust coefficient and axial induction factor modified with Glauert's empirical data. If the value for local thrust coefficient is in the bounds where momentum theory is valid, the code uses the momentum theory solution in equation 8 to calculate the axial induction factor required to produce the thrust on the blade element segment. Equation 8 is equation 9 solved for axial induction factor. A derivation of equation 9 from basic blade element momentum theory assumptions can be found in references 3, 4, 7, and 8.

$$a = 0.5 \left( 1 - \sqrt{1 - \frac{C_{T_L}}{F}} \right) \tag{8}$$

$$C_{T_L} = 4aF(1-a) \tag{9}$$

The error in axial induction factor for the current iteration is found by subtracting the value of a in the beginning of the iteration from the value of a computed from equation 8. If the error in axial induction factor is greater than the convergence criterion, the next iteration is begun using a new guess for axial induction factor in order to calculate the new local flow velocities. If the error in axial induction factor is less than the convergence criterion, the final forces are computed and the code moves radially outward to iterate between the velocities and forces of the next blade element. As mentioned earlier in this section, the code uses the final value of axial induction factor from the current blade element as the initial value for the next element.

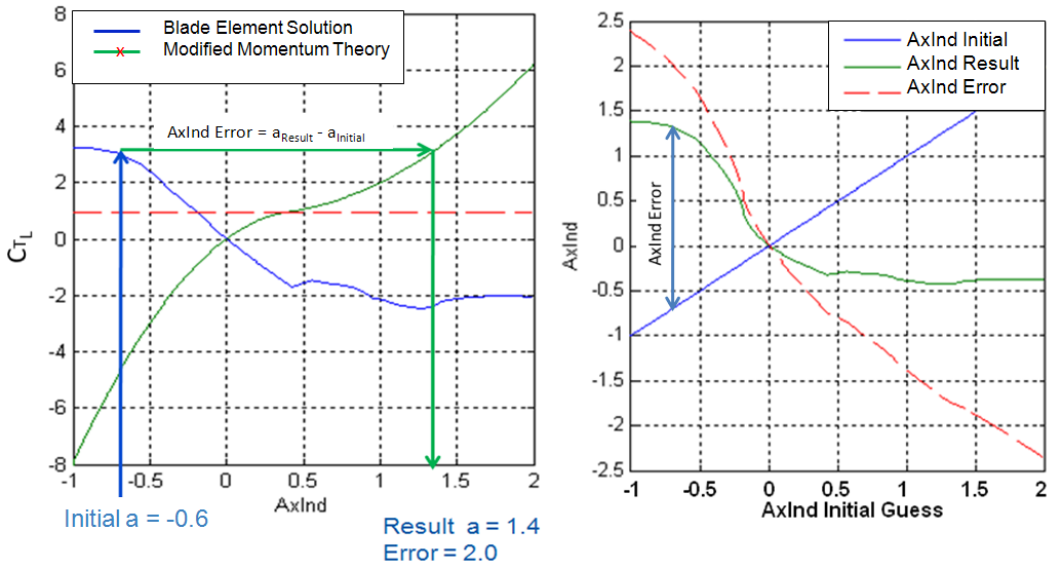


**Figure 3. Iteration routine for local thrust coefficient and sectional axial induction factor.**

An example iteration on axial induction factor is shown graphically in figure 3. The line labeled "Modified Momentum Theory" is the same as the curve labeled "Empirical (Glauert)" in figure 1, transitioning to the momentum theory solution for axial induction factor values below approximately 0.5. Figure 3 also shows a plot of the axial induction factor computed using equation 8, labeled "AxInd Result" versus the initial value of axial induction factor. The error between the two is labeled "AxInd Error"; when this error is zero, the solution is considered converged. In the thrust coefficient versus axial induction factor plot, the example iteration starts with an initial value of a = -0.6. Then, equations 1-7 are used to calculate the local thrust coefficient based on this axial induction factor, equaling approximately 3.0 in this example. Next, the axial induction factor needed to produce this value of thrust is found from modified momentum theory, giving a value of a = 1.4. Since the error of 2.0 is much larger than the convergence criterion, the code will try a new initial value for axial induction factor. The code would then march to a new initial value for a, say a=-0.5, and repeat the process. The previous release of WT_Perf would march to the solution, changing the marching direction and step-size when the sign of the error would change. The newest release uses a combination of Newton-Raphson and binary search algorithms to find solutions for the axial and tangential induction factors.

## II. Convergence Issue

Users working with the previous version of WT_Perf have experienced convergence issues for some cases. Some of these cases are for non-typical blade geometries or flow conditions, such as wind turbine startup or shutdown. It is not surprising that cases operating near the vortex ring state at an axial induction factor of one would fail to converge, since the blade element momentum theory is not valid in this region. Blade element moment theory is not valid in this region because it is not able to model the method of momentum transfer in the vortex ring state due to the method's assumption of radial independence. Unexpectedly, some of the non-converged cases were for cases with real solutions in the normal wind turbine operating range of axial induction factor, labeled the "Windmill" operating state in figure 1. The reason for these convergence issues for cases in the normal operating range is that the iteration routine runs

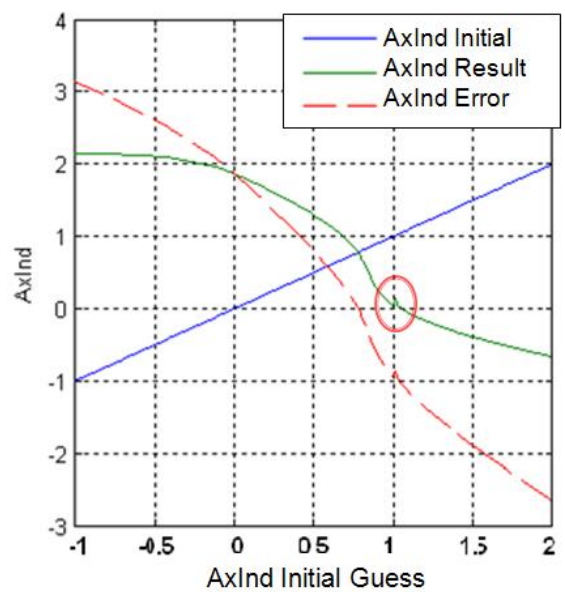


**Figure 4. Spike in axial induction factor solution.** *Spike circled in red. TSR 17, RotSpd 72 RPM, Pitch 0.05, Segment 8, Sector 2 (of 4).*

through the vortex ring state and gets trapped in that region, despite the fact that the actual solution exists well away from this operating state.

A case which would be expected to converge is shown in figure 4, which shows the iteration space for one radial blade segment of a normally operating wind turbine blade. The example case in figure 4 has a real solution where the error in axial induction factor decreases to zero, which is at an axial induction factor of 0.75 for this example case. The error in axial induction factor is zero where the initial "guess" value for axial induction factor equals the final resultant value from equation 8. The other blade segments in this blade analysis case had no convergence issues. A small spike can be seen near an axial induction factor of 1.0, which does not appear in the neighboring, convergent blade segments.

### A. Description of Convergence Issue

This spike is caused by a singularity in the strip-method equations that are currently being used calculate the thrust coefficient. The singularity arises in the initial calculation of thrust coefficient via blade element theory. At an axial induction factor of 1.0, for large positive or negative pitch angles, the calculation of thrust coefficient via

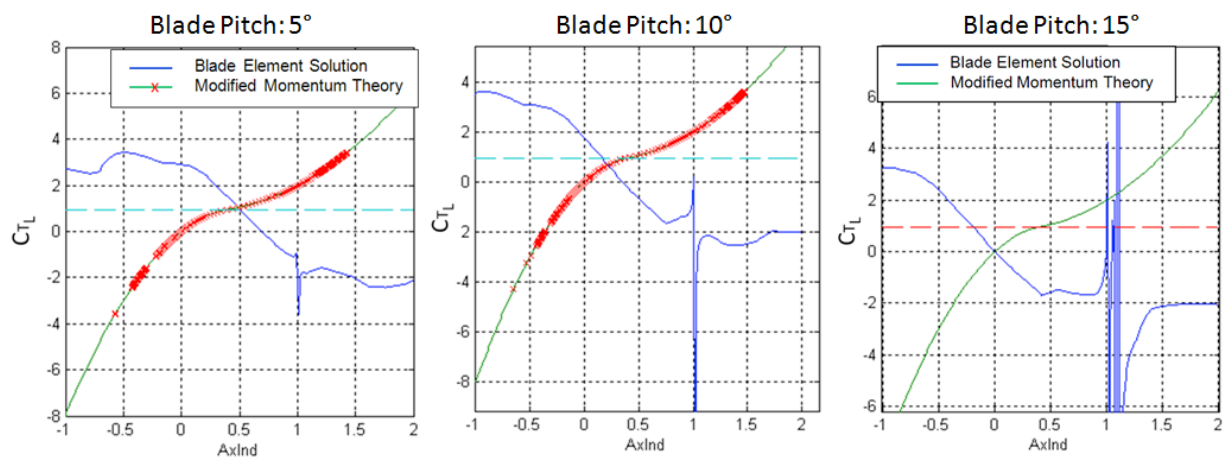


**Figure 5. Effect of blade pitch angle on numeric spike.**

blade element (strip) theory becomes singular (goes to +/- infinity). Figure 5 shows the blade element theory (or strip theory) solution for the thrust coefficient versus axial induction factor. A spike can be seen at a value of axial induction factor of one; note that the pitch is at a value of 5 degrees.

As the pitch is increased from 5 to 10 degrees, the spike appears to be asymptotically increasing and decreasing on either side of an axial induction factor of one. At 15 degrees of pitch, the blade element solution shows a range of axial induction factor where the blade element solution oscillates rapidly. This region exists between a range of 1.0 to 1.2 of axial induction factor. If the iteration routine would try to analyze a value of axial induction factor within this region of rapid oscillation, the routine will become trapped and just oscillate between axial induction factors close to one. A real solution for this case exists at an axial induction factor of one, where the blade element and modified momentum theory solutions converge. However, the iteration routine would never be able to reach this real converged solution if it becomes trapped in the region of rapid oscillation in the blade element solution.

The data in figure 5 indicate three main theories: (1) there is a singularity in the BEMT equations near an axial induction factor of one, indicated by the spike in the blade element solution, (2) the influence of this singularity or spike depends on the pitch angle of the blade, and (3) a range of values of axial induction factor can be found where the iteration routine could become trapped.

### B. Explanation

A potential singularity is apparent in equation 7, which is used to calculate the local thrust coefficient via blade element theory. As the local flow angle ($\Phi$) approaches zero, the $1/\sin^2\Phi$ term tends towards infinity. It was initially assumed that the singularity was caused by the $1/\sin^2\Phi$ term in equation 7, so the equation was rederived to remove this term, resulting in equation 10.

$$C_{T_L} = \frac{\sigma_r C_n (1-a)^2}{\cos\psi}\left(\frac{\Omega^2 r^2 (1+a')^2}{U_\infty^2} + (1-a)^2\right) \tag{10}$$

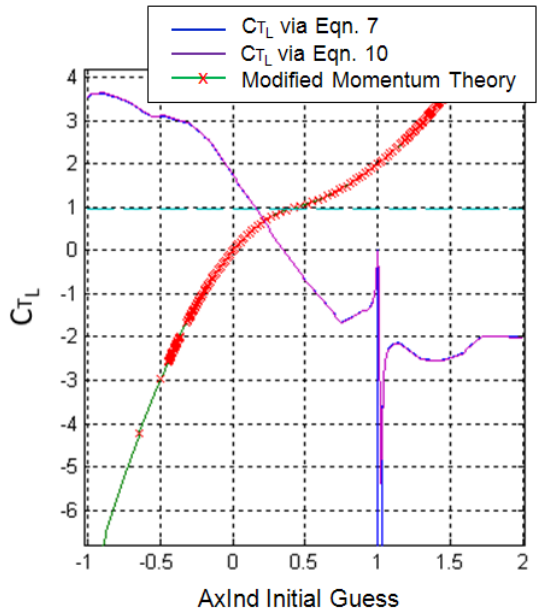


**Figure 6. $C_{T_L}$ vs axial induction factor for the two derivations of the blade element solution for $C_{T_L}$ .**

Figure 6 compares the results from the two equations, with the original version of equation 7 shown in dark blue, and equation 10 shown in violet. As expected, the results are similar away from the singularity, but equation 10 does not greatly affect the results near the singularity. Looking at the entire equation, rather than just the $1/\sin^2\Phi$ term, it becomes apparent that equation 7 does not actually have a singularity near an axial induction factor of one. The $1/\sin^2\Phi$ term goes to infinity at the same rate as the $(1\text{-}a)^2$ term approaches zero, meaning that the two terms essentially cancel each other. This can also be proven through L'Hôpital's rule. Although equation 10 did not remove the singularity, it does illustrate that the singularity is caused by the tangential induction factor (a'), the only remaining term which can cause a convergence issue.

### C. Tangential Induction Factor

The calculation of the tangential induction factor shows a singularity about an axial induction factor of one, as explained in the previous section. This singularity in the calculation of tangential induction factor (equation 6) is caused by the $\frac{1}{\sin\Phi}$ term. The relative impact of this singularity depends on the tangential component of force of a blade segment ($C_t$). The local thrust coefficient, $C_{T_L}$, increases as the pitch angle gets further from the zero-lift angle of attack (near zero pitch), causing the effect of the singularity to increase with increasing pitch angle, as illustrated in figure 5. Given the blade element forces on a blade segment, the tangential induction factor can also be calculated with equations 11 and 12, which together are an alternate form of equation 6.

$$a' = \frac{a'_{Coeff}}{\left(1 - a'_{Coeff}\right)} \tag{11}$$

$$a'_{Coeff} = \frac{\sigma_r * C_t}{4.0 * F * \sin(\Phi) * \cos(\Phi)} \tag{12}$$

This singularity can be eliminated by removing the drag coefficient term from the calculation of tangential induction factor. When the drag coefficient term is removed, the $\frac{1}{\sin\Phi}$ term cancels out with the $\sin(\Phi\sin(\text{Phi}))$ term associated with the lift coefficient ($C_l$) term in the calculation of $C_t$ via equation 5. Therefore, by removing the profile drag coefficient from the calculation of tangential induction factor, the spike, or singularity, in the tangential induction factor is removed. Figure 7 shows the results of calculating tangential induction factor without the profile drag term as well as the original results with the profile drag included. This figure clearly illustrates that removing the profile drag term eliminates the singularity in the computation of tangential induction factor, which should solve the numerical convergence issue. The drag coefficient term should not be

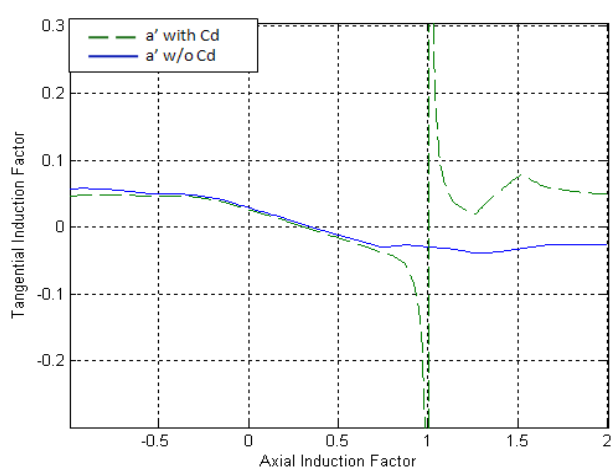


**Figure 7. Tangential Induction Factor vs Axial Induction Factor.** *Computed with and without the profile drag dependent term.*

eliminated completely because it can have a significant effect on the production of blade forces, particularly for a blade element near or post stall where pressure drag forces can be dominant.[7] Therefore, it is desirable to maintain the drag coefficient term in the calculation of tangential induction factor in most regions of operation and only eliminate the drag coefficient term in regions of operation where it causes a strong enough numerical issue to prevent convergence of the axial induction factor iteration routine. A method to accomplish this desire will be explained in the next main section.

It is interesting to note that outside of the influence of the singularity, the two formulations of tangential induction factor in figure 7 only deviate significantly at high or low values of axial induction factor, corresponding to regions of angle of attack near positive and negative stall for this case. Low values of axial induction factor typically correspond to positive stall of the airfoil for a given blade section, while higher values of axial induction factor typically correspond to negative stall. Near stall, the drag is relatively significant, causing the deviations between the two curves in Figure 7 at lower and higher values of axial induction factor.

### D. Physical Explanation

In order to gain a basic understanding of what the singularity in the equation for tangential factor represents, one must look at the derivation of the BEMT equations. The first step in deriving the equations is the basic relation:

Force = Rate of Change of Momentum

The Force term comes from the blade element solution for a radial blade section, while the Rate of Change of Momentum term comes from the rate of change of momentum through a stream tube. The axial induction factor can be solved using this equation for force, while the tangential induction is solved by balancing torque and the rate of change of angular momentum, as shown in equation 13. The tangential induction factor can be found by solving equation 13 for tangential induction factor, as shown in equation 14.

$$\begin{aligned} \text{Torque} &= \text{Rate of Change of Angular Momentum} \\ &= \text{Mass Flow Rate} * \text{Change of Tangential Velocity} * \text{Radius} \\ &= \rho A_D U_\infty (1-a) * 2\Omega a' r * r \end{aligned} \tag{13}$$

Solving for $a'$:

$$a' = \frac{Torque}{\rho A_D U_\infty (1-a) * 2\Omega r * r} = \frac{Torque}{Mass\ Flow\ Rate * 2\Omega r * r} \tag{14}$$

As the axial induction factor (a) goes to one, the mass flow rate through the disc goes to zero (the denominator in equation 14). If the torque would decrease to zero at the same rate as the mass flow rate goes toward zero, the tangential induction factor would maintain a finite and numerically reasonable value. However, if the torque does not decrease to zero and remains finite, then the tangential induction factor will tend toward an infinite value to attempt to balance the mass flow rate going toward zero. The tangential induction factor tending toward positive or negative infinity in order to balance the decreased mass flow rate is the cause of the singularity which has been causing the numerical convergence issues. In reality, the flow through a rotor does not behave in the way that blade element theory predicts for an axial induction factor of one; rather the flow recirculates through the rotor and forms into vortices which transfer energy and momentum through the disc as they are shed. These vortices are not stable, causing large fluctuations in thrust.

## III. Solution Implementation and Results

The solution selected to eliminate or reduce the numerical convergence issue was to remove the singularity in the equation for tangential induction factor. As explained in the previous section, the singularity near an axial induction factor of one is due to the drag-dependant term in the equation for tangential induction factor; therefore, the solution to the numerical convergence issues is to eliminate the drag dependant term for cases of axial induction factor showing effects of the numerical convergence issue. If the mathematical singularity is imagined as a graphical representation, as in the "Blade Pitch: 15°" case of figure 5, then one can visualize the region of rapid oscillation around the singularity being "cut out" by setting the drag term to zero in this region. This "cut out" range of axial induction factor, where the drag term in the formulation of tangential induction factor is set to zero, would be similar to a cut-off distance or Rankine vortex core radius used in singularity solutions of vortex methods. The "cut-out" method was tested, but the jump in tangential induction factor caused by suddenly setting the drag term to zero caused new convergence issues. A method was created to smoothly transition the drag term to zero in the region of the singularity where the iteration routine could become trapped. This method of smoothly transitioning

the drag dependent term to zero is similar to a vortex core model with smooth transition between the induced velocity inside the vortex core and the induced velocity of the outer potential flow solution. A summary of vortex core models can be found in reference 9.

The first main step in smoothly removing the profile drag dependent term in the equation for tangential induction factor was to split the equation into two parts, a lift coefficient dependent term and a drag coefficient dependent term. If the tangential induction factor is defined as in equation 11, then the tangential induction factor coefficient ($a'_{Coeff}$) can be split into a lift and drag dependent terms, as shown in equation 15. Since the singularity depends on the drag coefficient term, this term can be removed in the region of the singularity. This will meet the main goal of granting a real solution to the blade element equations throughout the entire range of axial induction factor while minimizing the influence on solutions not in this region.

$$a'_{Coeff} = a'_{Cl} - a'_{Cd} \tag{15}$$

$$a'_{Cd} = \frac{\sigma_r * C_d}{4.0 * F * \sin(\Phi)} \tag{16}$$

$$a'_{Cl} = \frac{\sigma_r * C_l}{4.0 * F * \cos(\Phi)} \tag{17}$$

In order to be able to transition the drag dependent term to zero, a new variable was created and multiplied by the drag coefficient term in equation 15, resulting in equation 18. This new variable has been termed "SingTransition", meaning singularity transtion multiplier. By adjusting this variable, the influence of $a'_{Cd}$ can be adjusted.

$$a'_{Coeff} = a'_{Cl} - (SingTransition * a'_{Cd}) \tag{18}$$

The singularity transition multiplier is a function of axial induction factor. A proper shape function must be found for this term, as well as the boundaries that will determine where it will influence $a'_{Cd}$. The requirements for this shape function are that it must transition from a value of one to a value of zero, and back to one, and have a smooth transition and continuous slope between these values. A sine curve meets these requirments and was selected as the shape function for the singularity transtion multiplier.

The goal of this sine curve multiplier is to remove the spike caused by the singularity at an axial induction factor of one, as well as any regions of rapidly oscillating blade element solutions, as explained in the section about figure 5. The sine curve singularity transition model eliminates the singularity influence at an axial induction factor of one as long as it is centered at this point, meaning that the base of the trough of the sine curve will sit at an axial induction factor of one. The next step is to determine the bounds, or edges, of the sine curve singularity transition term. To find these bounds, plots of the blade element solution for $C_{T_L}$ vs AxInd were created, such as the "Blade Pitch: 15°" case of figure 5, which clearly show the regions of oscillation responsible for the iteration routine convergence issue. The bounds were set equal to the width of the region of the rapid oscillation, with a small safety factor. Many cases were tested to ensure that this method would create wide enough boundaries for the singularity transition term to eliminate the convergence issue for all of the cases considered.

To find these bounds, it was first logically determined what variables the bounds are dependent upon, meaning what variables the width of the region of rapid oscillation of the blade element solution are dependent upon. Equation 12 shows the main variables which influence these bounds: Local-Solidity ($\sigma_r$), $C_t$, Loss (F), and local flow angle ($\Phi$). The Prandtl tip-loss factor (F) will be ignored, as it influences the whole range of axial induction factor, not just the area around one. The local tangential force coefficient, $C_t$, will be used rather than $C_d$, since $C_t$ showed a more linear trend for the singularity transition term bounds.

The local flow angle is a function of the Local Speed Ratio (LSR), which is defined in the section titled "Blade Element Momentum Theory Iteration". Therefore, the trend of the singularity transition term bounds vs LSR will represent the dependence of the width of the convergence issue region as it depends on the local flow angle. The terms $C_t$ and Local-Solidity can be lumped together, as they are multiplied by each other in the above equations. Therefore, there are two terms for which trends will be found: ($\sigma_r * C_t$) and LSR.

The next step was to create the trends of the singularity transition term bounds vs the two terms identified in the previous step. Plots of the blade element solution for $C_t$ vs AxInd were created, as shown in figure 8. This plot is for a blade root pitch angle of 20 degrees. Similar charts were generated for blade root pitch angle values of 10, 20, and 35 degrees to give a range of tangential force coefficient ($C_t$). The TSR was varied between 2 and 45 to give a range of LSR values. The TSR values are only shown between 5 and 45 because a TSR of 2 showed no convergence issues for this case. The vertical spikes represent regions affected by the singularity and would present convergence issues. This region is expanded in Figure 9.

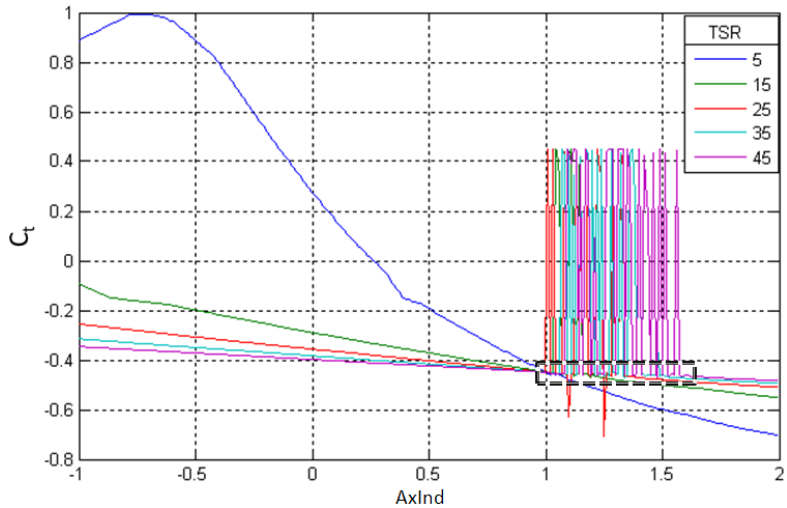


**Figure 8. Tangential Force Coefficient vs Axial Induction Factor for a range of TSR (Tip Speed Ratio).** *Region of convergence issue indicated by dashed box.*

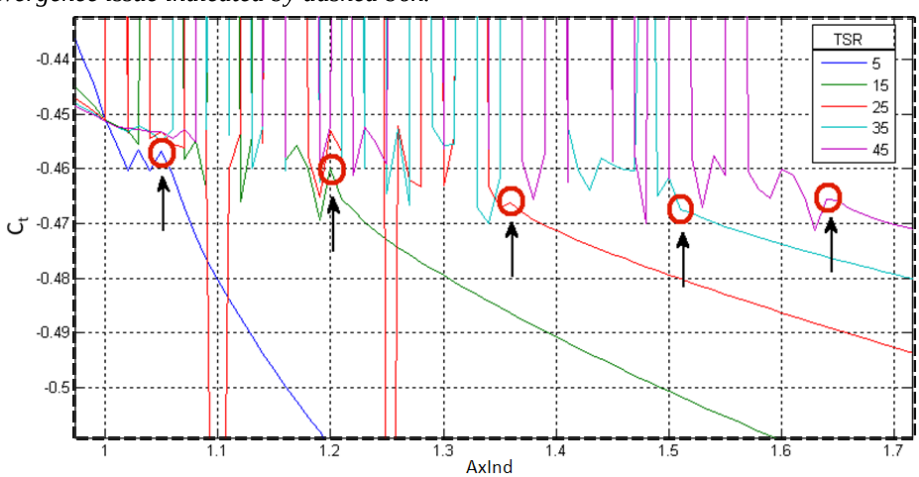


**Figure 9. Tangential Force Coefficient vs Axial Induction Factor for a range of TSR.** *Red circles indicate upper bound of region of convergence issue for each TSR.*

The regions of rapid oscillation in the ratio of $C_t$ vs AxInd begin at or slightly less than an axial induction factor of one, thus the lower bound is typically at AxInd = 1.0. The sine curve singularity transition model has its trough set at AxInd = 1.0 and the sine curve is symmetric, so the lower bound is ignored, since the upper bound is always larger than the lower bound. The points where the solution stops oscillating are circled in red with an arrow below them. These points represent the upper bound of the sine curve singularity transition model, above which a stable solution exists for the given set of equations. Since multiple TSR values were analyzed, figure 9 can be used to find the value of the upper bound of axial induction factor as a function of the LSR. The LSR is a function of TSR, as discussed in the beginning of the paper.

The trend of the upper bound versus the LSR was found to be very linear, as is shown in Figure 10. Also, the intercept was always very close to AxInd = 1.0. Thus, the slope of axial induction factor upper bound ($AxInd_{Upper}$) versus LSR was plotted versus the tangential force coefficient ($C_t$), which also showed a very linear regression, as is shown in Figure 11. Cases were analyzed for several values of rotor rotational speed, labeled

"RPM", and for two very different wind turbines, the Unsteady Aero Experiment twisted blade and the Wind Pact 1.5MW machine. The results proved insensitive to rotational speed or the wind turbine size, although only these two wind turbines were analyzed.

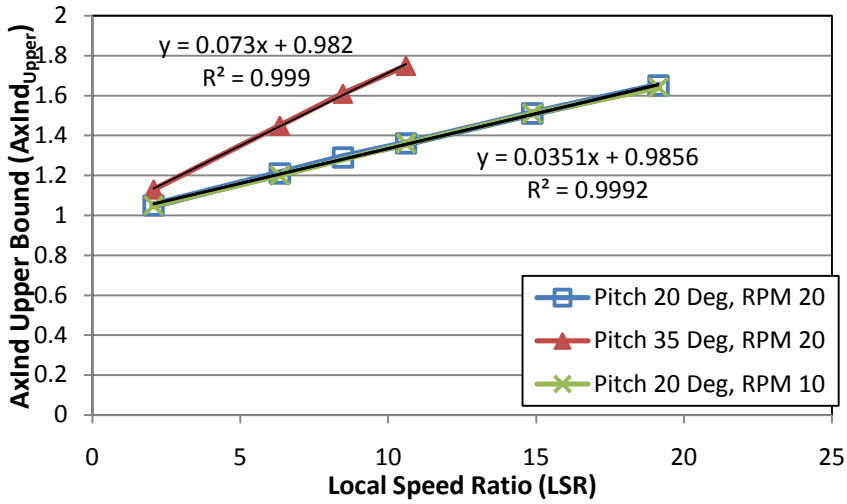


**Figure 10. AxInd Upper Bound Vs Local Speed Ratio.**

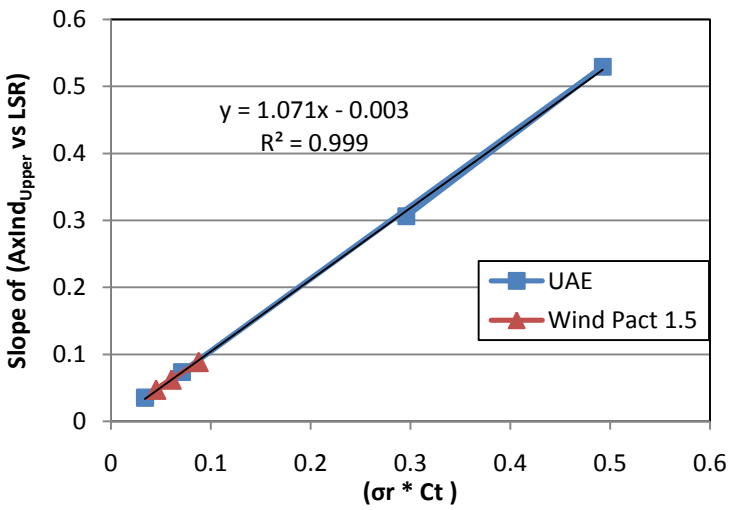


**Figure 11. Slope of (AxInd$_{Upper}$ vs LSR) vs (Local Solidity * Tangential Force Coefficient).** *Data for UAE Phase VI and WindPACT 1.5MW wind turbine blades.*

A linear trend-line was fit through the data in figure 11. An offset was added to the resulting equation for the upper bound of the singularity transition model (AxInd$_{Upper}$). The upper bound of axial induction factor for the singularity transition model results in equation 19. The values of AxInd$_{Upper}$ were shown not to depend on the sign of $C_t$ or LSR, so the absolute value of these variables is taken.

$$AxInd_{Upper} = (1.0718 * |\sigma_r * C_t| - 0.003) * |LSR| + 1.0 + Offset \tag{19}$$

The sine curve singularity transition model is symmetric for fixed values of $(\sigma_r * C_t)$ and LSR. However, since $C_t$ can vary with axial induction factor, the singularity transition term is often not symmetric because the lower bound is based on the local solution of equation 19, which will most likely be different from local values near the upper bound. Despite the dependence on local values, the curve is always continuous in both value and slope. This effect is illustrated in the example singularity transition model shown in figure 12, where the singularity transition model is the black line labeled "SingTransition" and the curve representing the tangential induction factor using this singularity transition model is labeled "TanIndCoef". The curve labeled "TanIndCoef_Old" represents the calculation of tangential induction factor without the singularity transition model.

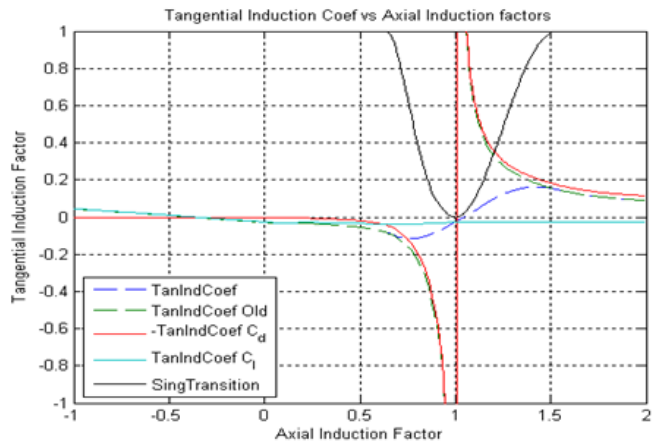


**Figure 12. Tangential Induction Factor vs Axial Induction Factor.** *Computed with and without the profile drag dependent term.*

### E. Results with New Tangential Induction Factor Algorithm

As shown in figures 13 and 14, the tangential induction factor singularity smoothing term works well for the example cases. Figure 13 represents an extreme case, where a real solution exists at an axial induction factor of one and a second real solution exists at an axial induction factor of 1.4. The new iteration routine will allow a solution for both cases, although the first solution is not valid since blade element momentum theory is not capable of correctly modeling the actual flow physics of the vortex ring state. An analysis method with less physical assumptions would be necessary to model the vortex ring state, such as a free-wake potential flow vortex method.

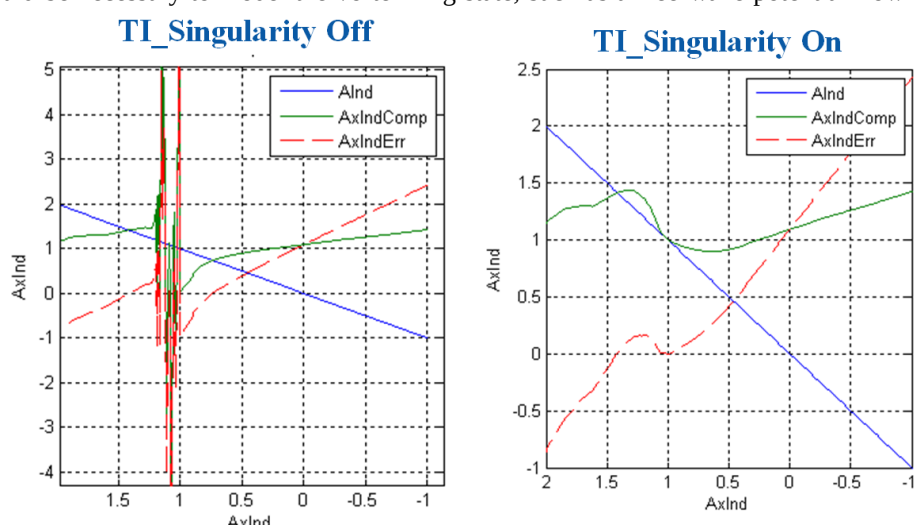


**Figure 13. Comparison of Results for Axial Induction Factor Iteration.** *TSR 17.0, RotSpd 72.0, Pitch 15deg. Note: the x-axis is reversed compared with other figures in this paper.*

The singularity smoothing method was not developed to attempt to solve solutions in the vortex ring state; instead it was designed to eliminate convergence issues developed when the iteration routine runs through this state. Figure 14 shows a case where a real solution exists near an axial induction factor of zero. This example case did not converge in the old version of WT_Perf, despite a real solution existing. Instead, the old iteration routine would oscillate about an axial induction factor of one until the maximum number of iterations was reached. The new version of WT_Perf has no trouble finding the real solution, illustrating its success at having improved rates of convergence. The new iteration scheme and smoothing method typically take longer to find a solution than that previous version of WT_Perf; however, the new version has a one-hundred percent convergence rate for all of the cases examined during the development of the singularity transition smoothing method.

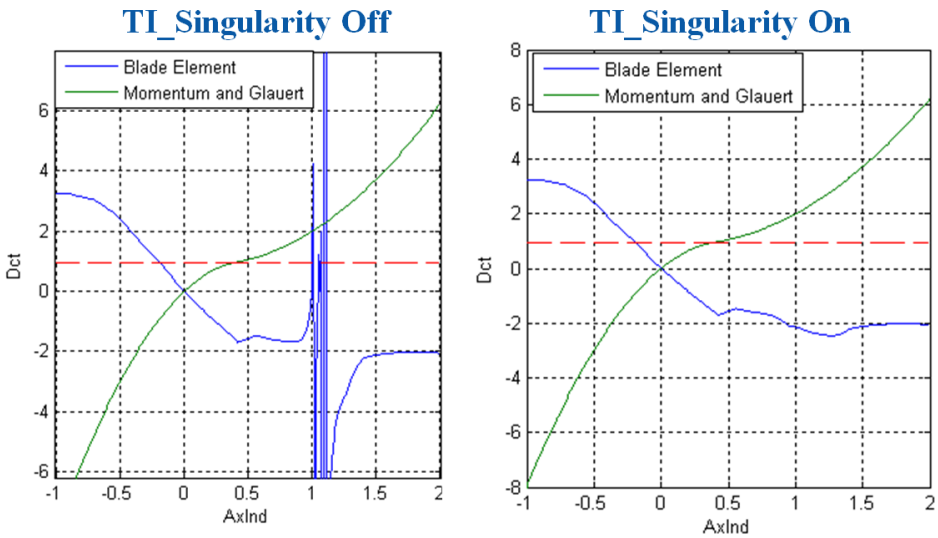


**Figure 14. Comparison of Results for Local Thrust Coefficient (Dct) vs. Axial Induction Factor.** *TSR 17.0, RotSpd 72.0, Pitch 15deg. Note: the x-axis is reversed compared with other figures in this paper.*

## IV. Conclusion

The newest release of WT_Perf addresses a convergence issue in the solution of the blade element momentum theory equations. The method used to fix these convergence issues is robust, but slows down the run time of the code. A new iteration routine has been implemented to improve the speed of the code, offsetting some of the delay caused by the methods used to improve convergence.

## Acknowledgments

The majority of this work was funded by the National Renewable Energy Laboratory as part of the Research Participant Program. The author thanks the mentorship of Marshall Buhl during the course of the work.

## References

[1]Buhl, M.L., Jr.; Wright, A.D.; Tangler, J.L. (December 1997). *Wind Turbine Design Codes: A Preliminary Comparison of the Aerodynamics*. NREL/CP-500-23975. Golden, CO: National Renewable Energy Laboratory.

[2]Buhl,M., "NWTC Design Codes,"http://wind.nrel.gov/designcodes/simulators/wtperf/. Last modified Sept. 23,2009 [cited May 28, 2010].

[3]Wilson, R.E.; Lissaman, P.B.S. 1974. *Applied Aerodynamics of Wind-power Machines.* NTIS: PB-238-595. Corvallis: Oregon State University.

[4]Wilson, Robert E.; Lissaman, Peter B.S.; Walker, Stel N.; *Aerodynamic Performance of Wind Turbines*. Corvallis, Oregon: Oregon State University, June 1976.

[5]Hibbs, B., Radkey, R. L. 1 98 1 . *Small wind energy conversion systems rotor performance model comparison study.* Rockwell Int., Rocky Flats Plant, RFP-4074/1 3470/3633 1 /8 1-0

[6]Buhl, M.L., Jr. 2004. *A New Empirical Relationship between Thrust Coefficient and Induction Factor for the Turbulent Windmill State.* NREL/TP-500-36834. Golden, CO: National Renewable Energy Laboratory, September.

[7]Burton, T.; Sharpe, D.; Jenkins, N.; Bossanyi, E. 2001. *Wind Energy Handbook*, New York: Wiley & Sons.

[8]Manwell, J.F.; McGowan, J.G.; Rogers, A.L. 2002. *Wind Energy Explained: Theory, Design and Application*. New York: John Wiley & Sons.

[9]Leishman, J.G. 2000. *Principles of Helicopter Aerodynamics*. Cambridge, UK: Cambridge University Press. pg. 436.