Finite Element Based Lagrangian Vortex Dynamics Model for Wind Turbine Aerodynamics

Michael K. McWilliam, Curran Crawford
Department of Mechanical Engineering, University of Victoria, 3800 Finnerty Road, Victoria, British Columbia, Canada, V8P 5C2
E-mail: mmcwilli@uvic.ca, curranc@uvic.ca

Abstract.
This paper presents a novel aerodynamic model based on Lagrangian Vortex Dynamics (LVD) formulated using a Finite Element (FE) approach. The advantage of LVD is improved fidelity over Blade Element Momentum Theory (BEMT) while being faster than Numerical Navier-Stokes Models (NNSM) in either primitive or velocity-vorticity formulations. The model improves on conventional LVD in three ways. First, the model is based on an error minimization formulation that can be solved with fast root finding algorithms. In addition to improving accuracy, this eliminates the intrinsic numerical instability of conventional relaxed wake simulations. The method has further advantages in optimization and aero-elastic simulations for two reasons. The root finding algorithm can solve the aerodynamic and structural equations simultaneously, avoiding Gauss-Seidel iteration for compatibility constraints. The second is that the formulation allows for an analytical definition for sensitivity calculations. The second improvement comes from a new discretization scheme based on an FE formulation and numerical quadrature that decouples the spatial, influencing and temporal meshes. The shape for each trailing filament uses basis functions (interpolating splines) that allow for both local polynomial order and element size refinement. A completely independent scheme distributes the influencing (vorticity) elements along the basis functions. This allows for concentrated elements in the near wake for accuracy and progressively less in the far-wake for efficiency. Finally the third improvement is the use of a far-wake model based on semi-infinite vortex cylinders where the radius and strength are related to the wake state. The error-based FE formulation allows the transition to the far wake to occur across a fixed plane.

1. Introduction
Fast and accurate aerodynamic models are crucial for wind turbine design and optimization. There are two proven approaches, the first and more common in design is Blade Element Momentum (BEM) theory[1]. BEM relies on control volume analysis and blade element theory. These simplifications create a very efficient model with acceptable accuracy. However the assumptions limit the application to straight blades rotating in a normal plane. This limitation prevents the exploration of new blades that incorporate sweep, winglets and large coning.

The second approach is Computational Fluid Dynamics (CFD) typically based on various control volume discretizations of the Navier Stokes equations. The complexity of wind turbine aerodynamics typically demands large models with many mesh cells where the computational load is far too great to be useful for early design or to simulate unsteady loads for design certification. Thus CFD is only applied in the latter design stages to fine tune the geometry.
This leaves a gap that there is not a suitable aerodynamic model that is fast and efficient at modeling new blade designs. This research explores an alternative approach based on Lagrangian Vortex Dynamics (LVD)\([2]\). LVD is similar to potential flow, in that the goal is to solve the location and strength of all the vorticity in the flow. LVD can be derived from the Navier-Stokes equations by starting with the velocity-vorticity form, then the Helmholtz decomposition is used to represent velocity as a scalar and vector potential where the vector potential is then simplified using vortex elements (see Batchelor \([3]\) for details). In LVD a core model is used to remove the inherent singularity and simulate core-spreading \([2]\). Once the vorticity solution is known one can calculate the flow field and evaluate any aerodynamic forces.

The conventional approach to solve these systems is to apply time marching algorithms with explicit advection schemes. One of the main reasons LVD has not been widely adopted is the challenge in obtaining converged results for relaxed wake simulations. These methods are known to exhibit instabilities; some argue that these are inherent in the physics \([4, 5]\). However comparisons with experimental results show that the conventional numerical methods exacerbate these instabilities \([6, 7]\).

The problems are exacerbated in optimization applications \([8]\). Since optimization algorithms tend to explore designs where chaotic behavior is more pronounced (\(i.e.\) high blade loading and high tip speed ratios) these models need to be made more robust for these difficult conditions. The more efficient gradient-based optimization algorithms require gradients that are typically based on the difference between two similar solutions (\(e.g.\) finite differencing). This operation introduces subtractive cancellation error were the errors associated with the numerical noise are greatly amplified \([8]\). Overall, the lack of robustness and the inaccuracy in gradient calculations renders conventional relaxed wake LVD methods unsuitable for optimization.

This work overcomes these numerical stability problems by developing a new method to solve these equations. In this work the problem is formulated as an error function. In some circumstances a steady state solution is expected. Under these conditions the coupling between space and time can be used to avoid time marching. The full system can then be solved by minimizing the error; in the work presented here Newtons iteration with adaptive relaxation is used to solve these problems.

2. The Mathematical Model
The aerodynamic model is based on LVD and nonlinear Weissigner Lifting Line Theory (WLLT) \([9]\). WLLT represents the wind turbine blade as bound circulation \((\Gamma)\) proportional to the local lift on the blade as shown in equation (1) where vectors are expressed in bold font, \(c\) is the chord, \(c_l\) is the coefficient of lift (typically obtained from experimental results and a nonlinear function of the local angle of attack \(\alpha\)) and \(u\) is the relative wind speed.

\[
\Gamma_b = \frac{cc_l\|u\|}{2} \tag{1}
\]

Circulation is a conserved quantity according to Helmholtz theory so the blade generates trailed circulation in the wake proportional to the span-wise circulation gradient as shown in \((2)\) where \(s\) is distance along the lifting line. In unsteady conditions the blade would also generate shed vorticity which is ignored here since this report focuses on the steady state form of the model.

\[
\Gamma_t = \frac{d\Gamma_b}{ds} \tag{2}
\]

The vorticity generated in the wake are material elements and follow the local flow field. In wind turbines this creates a vortex system similar to the one shown in figure 1 where the trailed
vorticity has been lumped into line elements. The blade is on the left and extends straight up; the spiral curves coming off that line are the trailed vorticity. The vorticity associated with other blades would give a similar system but is not shown here for clarity.

This vorticity affects the flow field through the Biot-Savart law shown in equation (3) where $u$ is the velocity, and $d$ is the vector from a point on the vortex filament to the point of interest and $s$ is the tangent vector of the vortex filament. The Biot-Savart law integrates all the vorticity in the flow field to get the velocity at a single point, while the velocity is needed at all locations where vorticity exists (either to evaluate (1), or to evaluate the motion of the vorticity in the wake). Unlike grid based CFD, every computational element in LVD is coupled to every other element, leading to an inherent $O(N^2)$ computational complexity. The computational efficiency is therefore increased by using a relatively small number of computational elements to represent complicated flow structures.

$$u = \frac{\Gamma}{4\pi} \int \frac{ds \times d}{\|d\|^3}$$

Solving the flow field at the blade (and subsequently the aerodynamic forces) requires the position of the vorticity in the wake which also requires a solution to the flow field. The greatest challenge in LVD is reconciling this circular dependency and solving the position of the trailed vorticity. The focus of this paper is on novel ways of parameterizing the streaklines\(^1\) of vorticity and solving the wake positions. Note that in this formulation the blade need not be straight, but can have arbitrary in-plane and/or out-of-plane curvature.

3. Conventional LVD solution methods

The conventional approach to solving LVD problems is to represent the trailed elements as a chain of line elements. The line element is convenient because the Biot-Savart has an analytical solution for straight filaments (equation (4)) given by Van Garrel [10]. The vectors in equation (4) are defined in figure 2, $d_c$ is an added numerical parameter to avoid singularities and $l$ is the length of the element.

$$u = \frac{\Gamma (\|d_1\| + \|d_2\|) (d_1 \times d_2)}{4\pi \left(\|d_1\|\|d_2\| (\|d_1\|\|d_2\| + d_1 \cdot d_2) + (d_1, l)^2\right)}$$

\(^1\) In steady state flow-fields streamlines and streaklines are the same
In conventional LVD models [10] the end point positions ($x_1$ and $x_2$) and the strength ($\Gamma$) become the primary state variables for the wake. Then the time evolution of the system is discretized and solved with a time-marching algorithm. It is assumed that a steady state solution is achieved when the change in state between successive iterations is below a specified value. This formulation leads to several problems:

1) **High computational complexity from time step and element size coupling:** The physical size of the influencing elements is governed by the time-step. Thus small elements require small time steps and vice-versa. This scaling is further exacerbated since the required number of time steps for a given wake length scales with $1/\Delta t$.

2) **Inherent uncertainty with iterative convergence:** If a simulation exhibits slow convergence it will end prematurely with an erroneous solution.

3) **The numerical evolution in time is fundamentally unstable:** Pure vortex systems are fundamentally chaotic [2], time marching algorithms adopt this chaotic behavior, then numerical error in the discretization amplifies it further.

These problems are demonstrated in figure 3 that shows the $C_p$ vs. iteration for a nonlinear lifting line model with a first order explicit forward Euler time marching scheme [11]. The effect of inherent instability is shown in the numerical noise; as the simulation progresses it is clear the noise is increasing and will eventually dominate the simulation. In the presence of noise iterative convergence is assessed based on historical averages, yet the linear curve fit for the last 300 iterations shows these moving averages will also fail to converge. Much of this noise is caused by numerical error that in-turn could be reduce by smaller time-steps, however the coupling in the spatial and temporal discretization means this refinement is prohibitively expensive. In any case, explicit free-wake codes cannot be numerically converged to an arbitrary level, as is required for accurate gradient calculations.

The results presented in figure 3 are based on a worst-case scenario where there is high blade loading and a high tip-speed ratio. Although more realistic conditions would give a more benign response, this case is still important considering an optimization algorithm may explore this design space and it is required to evaluate off-design performance.

Several new problems are introduced when the conventional LVD algorithm is used in aero-elastic analysis and optimization:

4) **Multiple solutions are required in aero-elastic simulations:** LVD requires lifting line positions from the structural model, while the structural model requires the aerodynamic forces from LVD. Ideally these two models could be converged simultaneously, yet in practice this introduces more noise [8]. Instead the user is forced to solve multiple alternating aerodynamic and structural solutions in a Gauss-Siedel iteration.

5) **Path dependency prevents fast convergence in iterative schemes:** In most optimization and Gauss-Siedel routines previous numerical solutions are used to accelerate
Figure 3: Coefficient of power at every 5th iteration for an explicit time marching code [11]

the convergence of simulations. Yet the solutions of time marching algorithms exhibit path dependency that in practice will foul the convergence of these methods.

6) **Numerical noise produce erroneous gradients:** In optimization and design applications it is useful to gather sensitivity information. Conventional numerical schemes such as finite differencing tend to amplify the error in the noise [12], while the use of a time marching evolution with iterative convergence criteria makes it difficult to derive a compatible analytical scheme.

The primary goal of the new formulation is to overcome these limitations to develop an aerodynamic tool suitable for the optimal aeroelastic design of wind turbine blades.

4. The Improved Finite Element Formulation

The new formulation is based on a nonlinear lifting line vortex model with a free wake. The lift coefficient for the lifting line is based on experimental measurements of two dimensional lift coefficients for a range of attack angles and Reynolds number. The induction is calculated with the Van-Garrel vortex element shown in equation (4). This formula incorporates a core model with a radius $d_c$ which is initially taken as the airfoil thickness and then grows at a constant rate (typically 1cm/s).

The formulation presented here is for steady state solutions where only the trailing vorticity is considered. However this formulations can easily be extended for unsteady simulations and will be presented in future work.

This paper introduces two new innovations that together creates a totally new formulation. First is a Finite Element (FE) description of state discussed in section 4.1 that decouples the spatial and temporal descretization formulations and allows for local refinement. The next contribution is the use of an error function to solve the equations discussed in section 4.2. Finally, section 4.3 describes how all the relevant equations are combined in a solution algorithm to solve the model. Together these innovations address many of the issues discussed in section 3 and more.

4.1. **The Finite Element Description of State**

The first deficiency is addressed by adopting a different kinematic description for the wake structure (see figure 4). Instead of focusing on individual influencing element end-points the set of three-dimensional streaklines are parametrized using a FE description based on a set of shape functions $\eta_j(\tau)$ and control points $X_{xj}$ for any vortex age $\tau$. The position along the streakline is related to the vortex age through equation (5a). These streaklines are broken up into sections
Points
Influencing element (Biot Savart law)
Node for an influencing element
where \( x = \sum_j \eta_j(\tau)X_{xj} \) (Biot Savart law)
defined by the function \( \eta_j(\tau) \)
defines \( X_{xj} \)

(a) Chain of quadratic bases with 3 influence elements
(b) A refined influence configuration with more elements
(c) Reduced number of state variables with a chain of linear bases

Figure 4: Influencing elements (black-solid) connected to basis sections (grey-dashed)

and share end points with adjacent sections. These sections will be referred to as ‘basis sections’ while the control points \( X_{xj} \) will be referred to as ‘points’. This FE description gives the user two ways of refining the streakline: the user can chose to use many small basis sections with simple shape functions or can use a large basis section with high-order shape functions. In steady state flows this description also provides a definition for the vortex velocity shown in equation (5b).

\[
x = \sum_j \eta_j(\tau)X_{xj} \quad (5a)
\]

\[
\dot{x} = \sum_j \dot{\eta}_j(\tau)X_{xj} \quad (5b)
\]

In this formulation it is the space curve in equation (5a) (not the influencing elements) that describe the locations of vorticity. This leaves open the problem of calculating the influence associated with this curve. This can be calculated by applying numerical quadrature of equation (3), however this integral is improved by using a semi-analytical form that replaces the integration points with linear influencing elements described by equation (4)\(^2\). It is only through the application of this semi-analytical integration that this model incorporates the influencing elements shown in figure 4.

The key advantage is how the FE description addresses the first limitation of conventional LVD:

1) **Independent number of control points and influence elements**: Since a position on a streakline can be evaluated at any age, this parametrization places no restrictions on

\(^2\) Numerical quadrature of equation (3) assumes that the orientation and the relative distance (\( d \)) is constant for a distance \( \Delta s \) at each integration point, while equation (4) gives an analytical solution of equation (3) that accounts for the change in the relative distance. Furthermore equation (4) incorporates a core model that simulates diffusion and eliminates the singularity when \( d \to 0 \).
the number of influencing elements used to represent a vortex filament (e.g. figure 4b). Similarly since the nodes of the influence element are not constrained to the control points defining the streakline, control points can either be added for improved fidelity or removed to accelerate the computations (e.g. figure 4c).

Decoupling the influence element discretization from the control point discretization gives another opportunity for accelerating the computational speed. It is well known the structural models with a FE parametrization can be accelerated by using an optimal set of basis functions, this has lead to very fast reduced order models based on modal analysis. By introducing the FE parametrization here there is now a potential that LVD models can be accelerated in a similar way once an optimal set of basis functions is identified. Section 5.4 discusses the potential of applying a basis function approach to accelerating the convergence.

4.2. The Error Function

The second innovation in this new model is the use of an error function to define and solve the problem. The error function associated with the bound circulation strength on the blade (shown in equation (6a)), is based on WLLT. Equation (6a) is defined for the whole lifting line, but is currently evaluated at the mid-point of each lifting line element (see equation (6b)). This could be extended by defining an integral along the lifting line and evaluating it with numerical quadrature.

\[ r_T \equiv \Gamma - \frac{cc_l w}{2} \quad (6a) \quad R_{\Gamma_1} = r_T|_{s=middle} \quad (6b) \]

The error function for the wake is based on the fact that vortex elements follow the local flow field (i.e. are force-free). This means the velocity of any fluid element must be equal to the local flow field; any difference is an error in the solution. This error is shown mathematically in equation (7a) in a rotating reference frame where \( \Omega \) describes the angular velocity of the rotor, \( x_0 \) is a point on the rotation axis, \( u_\infty \) is the wind speed and \( u_\gamma \) is the influence from all the vorticity in the flow field.

\[ r_x \equiv \dot{x} + \Omega \times (x - x_0) - u_\infty - u_\gamma \quad (7a) \]

The error function in equation (7a) is defined continuously for every point along the trailing filaments (i.e. basis sections). This error is mapped to scalars by applying the Galerkin projection (see equation (7b)) of the error (i.e. \( r_x(\tau) \)) onto the weighting function \( \zeta_j(\tau) \): typically the shape function for state (i.e. \( \eta_j(\tau) \)) is used for the weighting function. Equation 7b gives a corresponding error for each control point, where the subscript \( j \) refers to the control points. The integral is approximated with numerical integration, the integration points and the associated weights are denoted by \( \tau_k \) and \( w_k \) respectively. In this work the integrals are approximated with Gauss quadrature\(^3\) (see Logan [13] for more details), one point quadrature was used with linear basis sections\(^4\), while two point quadrature was required for quadratic basis sections to avoid singularities.

\[ R_{xj} = \int_{\tau_0}^{\tau_f} \zeta_j(\tau)r_x(\tau)d\tau \approx \sum_k \zeta_j(\tau_k)r_x(\tau_k)w_k \quad (7b) \]

The error function addresses the second limitation of conventional LVD:

\(^3\) Gauss quadrature will give an exact solution to the integral of a polynomial of specified order with a minimum number of sampling points [13]

\(^4\) This is equivalent to mid-point rule in simple integration and implicit Crank-Nicolson for temporal integration in time marching codes
Lifting Line Elements ($\Gamma_i \rightarrow R\Gamma_i$)
Control Points ($X_{xj} \rightarrow R_{xj}$)

Figure 5: Schematic of an FEM based LVD simulation of a single wind turbine blade

2) The level of convergence is well defined: The model is considered solved when the algorithm finds a set of lifting line strengths (i.e. $\Gamma_i$) and control point positions (i.e. $X_{xj}$) where the error in equation (6b) and equation (7b) approaches 0 for all lifting line elements and control points respectively. There is no uncertainty in a convergence criteria defined by equations (6b) and (7b) because it is only dependent on the current state.

4.3. The Overall Solution Algorithm

Figure 5 shows an example of how computational elements are arranged to simulate the flow of a single wind turbine blade. In this figure the blade is modeled by four lifting line elements of strength $\Gamma_i$. The error $R_{\Gamma_i}$ of these elements is evaluated at the mid-points shown by the squares. A basis section (shown as a grey dashed line) is trailing from the end point of each lifting line element, the position of these basis sections is governed by the control points $X_{xj}$, which is associated with error $R_{xj}$. A set of influencing elements (shown by the black solid lines) are attached to each basis section (see figure 4 for the different ways basis section are combined with influencing elements). The primary state variables are the lifting line strength $\Gamma_i$ and the control points $X_{xj}$, each with a corresponding error $R_{\Gamma_i}$ and $R_{xj}$ respectively.

By defining error functions (i.e. equation (6b) and (7b)) one can formulate the system as a root finding problem as shown in equation (8) where the goal is to solve the state vector $X$ such that the error $R$ is $0$, where $X = (\Gamma_1 \; \Gamma_2 \; \cdots \; X_{x1}^T \; X_{x2}^T \; \cdots )^T$ and $R = (R_{\Gamma_1} \; R_{\Gamma_2} \; \cdots \; R_{x1}^T \; R_{x2}^T \; \cdots )^T$.

$$R(X) = 0$$

The most compelling feature of this new formulation is the fact that the problem can be formulated in the form of equation (8). Many problems across multiple disciplines are formulated as a nonlinear root solving problem like this. As such there is already a large body of research for different types of algorithms for problems of this form, the most important being the solution algorithms. Many algorithms were explored, but the most effective has been the Newton iteration with adaptive relaxation shown in algorithm 1.
Algorithm 1 Newton iteration with adaptive relaxation for FEM based LVD

Select the initial guess \( X(0) \)
\[
R(0) \leftarrow \text{CALCULATE RESIDUAL}(X(0))
\]
while \( \|R(i)\| > \epsilon \) do
\[
\begin{align*}
[J] & \leftarrow \text{CALCULATE JACOBIAN}(X(i)) \quad \triangleright \text{The Jacobian}^5 \text{ is defined as } [J] \equiv [\partial R/\partial X] \\
& \text{Solve the linear system } [J] \Delta X = -R(i) \text{ for } \Delta X \\
& \text{Update the state vector with optimal relaxation}^6 \quad X(i+1) = X(i) + \beta \Delta X \\
R(i+1) & \leftarrow \text{CALCULATE RESIDUAL}(X(i+1))
\end{align*}
\]
end while

function \text{CALCULATE RESIDUAL}(X)

for all lifting line elements do
\[
u \leftarrow \text{CALCULATE INFLUENCE}(x) \quad \triangleright x \text{ is the midpoint of the lifting line element}
\]
Calculate the angle of attack (\( \alpha \)) and Reynolds number (Re) Based on \( \alpha \) and Re interpolate experimental data to get \( c_l \) Calculate the error with equations (6a) and (6b) Assign the error to the corresponding entry in the vector \( R \)
end for

for all control points in the wake do
Initialize corresponding entries in the vector \( R \) to 0
for all integration points (\emph{i.e.} \( \tau_j \)) in equation (7b) do
\[
u \leftarrow \text{CALCULATE INFLUENCE}(x) \quad \triangleright \text{Position } x \text{ is based on equation (5a) and } \tau_j
\]
Calculate the velocity of the flow element \( \hat{x} \quad \triangleright \hat{x} \text{ is based on equation (5b) }
Calculate the local error with equation (7a)
Calculate the weighted error according to equation (7b)
Add the weighted error to the corresponding entries in the vector \( R \)
end for
end for
return the fully populated error vector \( R \)
end function

function \text{CALCULATE INFLUENCE}(x)
\[
u \leftarrow u_\infty \quad \triangleright \text{Initialize with the free stream wind velocity}
\]
for all influencing elements do
Calculate \( u_{yi} \) at \( x \) with equation (4)
\[
u \leftarrow u + u_{yi}
\]
end for
return the influence \( u \)
end function

Several more deficiencies are addressed by adopting Newton based solution algorithm:

3) The instabilities are eliminated by avoiding time evolution: The instabilities of conventional LVD are rooted in the chaotic time evolution of vortices. Since the solution is

\[5\] The Jacobian \([J]\) is calculated analytically based on the linearized versions of equations (6b) and (7b). For the sake of brevity these equations are omitted.

\[6\] The optimal relaxation parameter \( \beta \) is solved with a one parameter optimization algorithm. For the sake of brevity the details of this algorithm are omitted here, see Vanderplaats [14] for more on line search algorithms.
evolved based on the Jacobian (instead of the time evolution) these instabilities are eliminated.

4) **The aerodynamic and structural models can be solved simultaneously:** Both linear and nonlinear structural models take the same form as equation (8). Typically a relaxed Newton iteration similar to the one shown in algorithm 1 is used to solve nonlinear structural models. Thus this new aerodynamic formulation can be tightly coupled to the structural model and solved in a single Newton iteration. This improves upon conventional LVD by eliminating the additional Gauss-Seidel iteration required to resolve consistency constraints.

5) **Without path dependency there is faster convergence in iterative schemes:** Since the solution in the new formulation is defined by equation (8) the solver should arrive to the same solution regardless of initial guess. This is beneficial in optimization routines because previous numerical solutions can be used as the initial guess in algorithm 1 and reducing the number of iterations required to get the next solution.

The items 3 and 5 above should improve accuracy for gradient calculations based on finite differencing. Yet finite differencing requires a solution to equation (8) for every design variable. This fact highlights another benefit of casting LVD into the form in equation (8):

6) **Fast and analytic gradients can be calculated with adjoint methods:** When the system is expressed in the same form as equation (8) adjoint based gradient algorithms can solve all the design sensitivities by solving a linear system similar to the Jacobian. For more details on adjoint methods see Giles and Pierce [15]. This is a dramatic improvement over finite differencing where algorithm 1 is executed for each design variable which in-turn requires multiple solutions to the Jacobian.

In summary the overall advantages to this new formulation is it provides the required reliability for optimization with several options to make the overall process faster. The reliability comes from the ability to locally refine the model for accuracy; a well defined error measure; and a solution algorithm that is not based on chaotic time evolution. The solution algorithm adds extra computational loads since it requires solutions to the Jacobian. Yet this formulation can be accelerated by reducing the number of control points in less critical regions. When considering the overall aeroelastic optimization process further efficiencies can be found by tightly coupling the aerodynamic and structural model; using previous numerical solutions to accelerate convergence; and relying on adjoint methods to calculate design sensitivities.

5. Results
5.1. Basic Configuration
This new formulation was used to predict the performance of the MEXICO rotor [16]. The first set of results used regularly spaced linear basis sections, with only one element per basis. The predicted Coefficient of Power ($C_P$) and Coefficient of Thrust ($C_T$) is compared to the measurements in figures 6a and 6b with increasing number of elements in the wake. The trend shows that the $C_P$ and $C_T$ both converge towards values above the measured values. With a longer wake there is more induction at the blade causing the circulation strength to drop down to give good agreement with measurements (see figure 6c). Figure 6d shows the radius of the tip filament with increasingly more number of elements in the wake; with more elements the upstream portion of the wake shows improved agreement with Mexico. At the downstream half

---

7 When the aerodynamic and structural models are tightly coupled the state vector $X$ and residual vector $R$ contain terms from both the aerodynamic and the structural model. This leads to a formulation where the fully coupled system takes the form in equation (8) and can be solved as a single system.

8 In gradient based optimization the changes in the system become small where previous solutions are very close to the next solution.
the wake starts to contract again because the wake is truncated and there is no downstream vorticity to maintain the expansion.

When the number of span-wise elements is increased the decreasing strength at the tips is resolved (see figure 7a). This results in multiple strong vortex filaments near the tip that mutually influence each other to initiate roll-up as shown in figure 7b.

5.2. Far-Wake Model
There are two problems with the basic configuration. First, the wake truncation causes the wake to contract downstream and second, accurate results require a long wake associated with slow solver performance. To address this, we investigated the effectiveness of a far-wake model for improving both the accuracy and speed of the solver. The far wake model is based on a vortex cylinder (see Crawford [17] for details on this model).

The cylinder is placed at the end of each filament chain. The cylinder radius is based on the radius of the last node. The axial and azimuthal components of strength for the cylinder are based on the filaments strength and a linear least \( R^2 \) regression of the filament pitch for a fixed number of node positions at the end of the wake. Figure 8a shows that a far-wake model gives better agreement with the measured \( C_P \), while increasing the number of pitch regression nodes improves this agreement further. Figure 8b shows that the far-wake model also gives improved agreement with the measured tip vortex position. Adding a far-wake model increased
Figure 7: Basic configuration, varying span-wise elements, 60 element length, 18° size

circle the solution time by between 18% to 38%, which is small compared to the increase associated with longer wakes alone.

Figure 8b shows distortions in the vortex position, these distortions are amplified with a larger number of regression nodes, while figure 8c shows these distortions are still present in long wakes with a small number of regression nodes. In response, a vortex spiral was added in between the near wake and the far wake model to give a smoother transition in the induction solution. The geometry of this spiral has the same radius and vortex pitch as the cylinder. Comparison between figure 9a and 8a shows that adding the spiral transition slightly degrades the agreement with $C_P$. Figure 9b shows the vortex position still has some wiggles but seems to be less distorted with a small number of pitch regression nodes, yet there is still a large kink with 20 regression nodes.

Overall the far-wake model did improve the agreement in the rotor coefficients and the vortex position near the rotor. The far-wake mostly suppressed the tendency for the wake to contract in the downstream portion. However the far-wake also seemed to introduce a distortion in the vortex position solution in the latter portion of the wake. The results in figure 8d led to the hypothesis that since inboard filaments are terminating early the mix of fully resolved vortex filaments and vortex cylinders is causing these distortions, however the results in section 5.3 show this is not the root cause. Close to the cylinder there is a large difference between the influence solution of a vortex spiral and a vortex cylinder. The other hypothesis was the abrupt transition between the filament and cylinder could cause these distortions however comparisons between figure 8 and figure 9 shows this has a small impact.

The true source of these distortions remains a mystery, yet one untested hypothesis involves the solver behavior. Since a large proportion of the influence is from the far-wake model a large fraction of the residual in the full system is a result of this model. Yet the state data that controls the far-wake configuration is only in the last few elements in the wake. This discrepancy could cause the solver to push that state data into unnatural configurations so the large influence of the far-wake model is in better agreement with the rest of the larger system. It is clear that increasing the number of regression nodes seems to amplify these distortions. The solver still has to satisfy the local residual on these last elements, so when only a small number elements control the far-wake the solver is overly constrained to create large distortions. A possible solution to this problem involves parameterizing the far-wake configuration based on the rotor forces and
using the momentum equations to predict the far-wake flow field, thus limiting the dependency on the elements on the end.

5.3. Planar Wake Termination
In the standard configuration each point along the filament is associated with a fixed vortex age. Since not all filaments experience the same axial fluid velocity, they will be truncated at different axial positions. An alternative configuration allows the axial position of termination to be specified by allowing the vortex age to float as a state variable. In this configuration the axial position of the last point is constrained to a fixed value, then the total age of the filament becomes a free variable. The age of the intermediate points is based on a fixed proportion of the filaments total age. The affect of this formulation is to truncate the wake for all filaments at the same axial position.

This alternative configuration was used to generate far-wake solutions where all filaments terminated at the same axial position. Figure 10a confirms that this configuration will terminate all filaments at the same downstream distance. While figure 10b shows this did not solve the strange end configuration associated with the far-wake model.

Figure 8: Far wake model configuration, 8 span-wise elements, 18° size
5.4. Solver Speed and Limitations

This section discusses the computational time used to solve a range of problems with this model. The code for each and every simulation was executed in parallel on eight 2.67-GHz Intel Xeon x5550 cores. The computational time is reported as “wall time” which is the real time to complete a simulation, since the code exhibited excellent parallel efficiency the equivalent “CPU time” is approximately 8 times greater. An important factor in the utility of numerical simulations is the rate at which wall time will increase while increasing the number of elements (i.e. increasing the simulation fidelity). This is usually referred to as the complexity which is the largest order term in a least $R^2$ curve fit of the timing data. For this formulation there are four ways the model fidelity can be varied; varying the number of elements in the wake while holding element size constant; varying the number of elements along the span; varying the size of the influencing elements in the wake while holding the wake length constant (see figure 4b for an example); and changing the number of control points in the wake (see figure 4c for an example).

Initial timing studies kept the element size constant and maintained a one to one ratio between control points and elements. These timing studies focused on the length of the resolved wake
and the number of elements along the span. For small problems (short wakes or small number of span-wise elements) the solver exhibits $O(N^{3.1})$ to $O(N^{3.7})$ complexity (see figure 11a for example). When a large number of elements on the blade are combined with a long wake, the complexity increases to $O(N^{7.4})$ to $O(N^{10.0})$ (see figure 11b for example). This indicates there are two domains with different complexity; one where either the wake is short or the number of span-wise elements is small; then a second where there are many span-wise elements with a long wake. In the first domain the simulations are relatively fast, in the second domain the condition number of the Jacobian grows to orders of $10^5$ to $10^7$ and the solver can only make very small changes every iteration. This highlights another benefit to using far-wake models, by shortening the resolved length of the wake simulations with high span-wise fidelity can complete much faster by remaining in the $O(N^{3.1})$ to $O(N^{3.7})$ complexity domain.

A second timing study kept the wake length, span-wise element count and number of control points constant, but varied the element size. Figure 12 shows the affect of increasing the influence fidelity by using smaller influencing elements (increasing the number) while holding the number of state variables constant. Figure 12a demonstrates improved agreement in $C_P$ with more influencing elements. While figure 12b confirms the $O(N^2)$ complexity associated with the number of influence elements.
A third timing study kept the wake length, span-wise element count, and element size constant while varying the number of control points. This study focused on the effect of very large quadratic basis sections (which in effect is a reduced order model) shown in figure 13. Figure 13a demonstrates very fast convergence with increasing number of basis sections. Figure 13b shows the relationship between basis section size and wall time is not monotonic. There is minimum wall time with 12 basis sections, this corresponds to a control point every 60°. This suggests that LVD could be solved faster as a reduced order model with basis functions.

Another observation is the solver is slowest when it was negotiating roll-up; this lead to an algorithm that sought improved solutions along a curved path in state space. For some large problems this gave a 7-13 times speed-up, yet for other large problems it offered no speed-up and for smaller problems the solver was slower. Further work could give a curved search algorithm with more consistent performance.

The Steepest-Descent is an optimization algorithm that is not affected by high condition numbers, similarly the Levenburg-Marquardt algorithm is a blend between the Steepest-Descent and Newton Iteration. Both algorithms were able to significantly reduce the residual in the first few iterations but quickly the performance dropped off. Transitioning to a Newton iteration did not help, as it seems that these optimization algorithms move towards regions with higher condition numbers while the pure Newton iteration navigates around these regions.

Another limitation is this solver can have difficulty at low tip speed ratios when portions of the blade are in stall. This problem is associated with the noise in the airfoil curves where dramatic fluctuations in the lift coefficient with respect to the angle of attack produce local minimum in the residual. This problem can be fixed by smoothing the airfoil data.

6. Conclusions and Future Work
This paper presents a new LVD model of wind turbine aerodynamics for aeroelastic optimization applications. This model addresses many deficiencies of conventional LVD models by using
a FEM formulation for state and a Newton-based solution algorithm in conjunction with an error equation. When this model is combined with a far-wake model it demonstrates excellent agreement with experiments with very fast execution (3 minutes). This model shows poor execution speed when the wake is long and many elements are used along the span. Improving the performance for these high fidelity configurations is an ongoing topic of research.

Another benefit of this formulation is the ability to generate fast reduced order forms of the model. An example is using quadratic basis functions that spans a large length of the filament as shown in figure 4. Preliminary results shown in figure 13 show that reducing the number of state variables improves the solver speed while having negligible affect on the solution. It also illustrates how the model may be tuned by separate adjustment of streakline and element resolution to optimize performance vs. solution fidelity. Future work will investigate other approaches of controlling fidelity; like higher order basis functions; basis functions that control multiple filaments; and basis functions in a cylindrical coordinate system.

An unsteady version of this model is a future topic of research. The unsteady version will still rely on a FEM description of state and a residual formulation. This unsteady formulation has two configurations, a conventional time marching simulation or one that solves the periodic response from cyclical forcing. The former is useful for unsteady conditions while the later useful for simulating the turbine response in a sheared flow. The time marching version can be used to give better initial guesses to the Newton iteration. This could avoid the slow progress in the early iterations by bypassing many of the complicated nonlinearities.

Finally this model will be incorporated into a larger Multi-disciplinary Design Optimization (MDO) framework. This involves coupling this LVD model with a structural model. For sensitivity analysis in optimization a fast gradient algorithm based on the adjoint method will also be developed. This framework in conjunction with optimization will be used to explore advanced turbine designs like bend-twist coupling, sweep, coning and other rotor concepts.

References

[1] Tony Burton, David Sharpe, Nick Jenkins, and Ervin Bossanyi. *Wind Energy Handbook*. John Wiley and Sons, Chichester, England, 2001.
[2] Georges-Henri Cottet and Petros D. Koumoutsakos. *Vortex Methods Theory and Practice*. Cambridge University Press, Cambridge, England, 2000.
[3] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, Cambridge, England, 2000.
[4] V. L. Okulov. On the stability of multiple helical vortices. *Journal of Fluid Mechanics*, 521:319–342, 12 2004.
[5] J. H. Walther, M Gunot, E Machefaux, J T Rasmussen, P Chatelain, V L Okulov, J N Srensen, M Bergdorf, and P Koumoutsakos. A numerical study of the stability of helical vortices using vortex methods. *Journal of Physics: Conference Series*, 75(1):012034, 2007.
[6] Ashish Bagai and J. Gordon Leishman. Rotor Free-Wake Modeling using a Pseudo-Implicit Technique - Including Comparisons with Experimental Data. *Journal of The American Helicopter Society*, 40(3):29–41, 1995.
[7] Mahendra J. Bhagwat and J. Gordon Leishman. Stability, Consistency and Convergence of Time-Marching Free-Vortex Rotor Wake Algorithms. *Journal of The American Helicopter Society*, 46(1):59–71, 2001.
[8] Michael McWilliam, Stephen Lawton, and Curran Crawford. Towards a Framework for Aero-elastic Multidisciplinary Design Optimization of Horizontal Axis Wind Turbines. The 51st AIAA Aerospace Sciences Meeting, 2013.
[9] J. Weissinger. The lift distribution of swept-back wings. *National Advisory Committee for Aeronautics Technical report*, 1947.
[10] A. van Garrel. Development of a Wind Turbine Aerodynamics Simulation Module. Technical report, The Energy Research Center of the Netherlands (ECN), 1755 ZG Petten, Netherlands, 2003.
[11] S. Lawton and C Crawford. Development and validation of LibAero, a potential flow aerodynamics library for horizontal-axis wind turbines. 50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, January 2012.
[12] Stephen Lawton and Curran Crawford. The Development and Validation of LibAero, a Potential Flow Aerodynamics Library for Horizontal-Axis Wind Turbines. Accepted to the 50th AIAA Aerospace Sciences Meeting, Control ID 1134189, 2011.

[13] Daryl L. Logan. A First Course in the Finite Element Method Fourth Edition. Thomson, 1120 Birchmount Road, Toronto, Ontario, M1K 5G4 Canada, 2007.

[14] Garret N. Vanderplaats. Numerical Optimization Techniques for Engineering Design. 3rd ed. Vanderplaats Research & Development Inc., Colorado Springs, 2001.

[15] Michael B. Giles and Niles A. Pierce. An Introduction to the Adjoint Approach to Design. Flow, Turbulence and Combustion, 65:393–415, 2001.

[16] H. Snel, J. G. Schepers, and B. Montgomeric. The MEXICO project (Model Experiments in Controlled Conditions): The database and first results of data processing and interpretation. Journal of Physics: Conference Series, 75:1–11, 2007.

[17] Curran Crawford. Advanced Engineering Models for Wind Turbines with Application to the Design of a Coning Rotor Concept. PhD thesis, Trinity College, Department of Engineering, University of Cambridge, October 2006.