Unsteady 3D Wake Modelling for Marine Current Turbines

T. McCombes, C. Johnstone, A. Grant[1]

[1] Energy Systems Research Unit,
Department of Mechanical Engineering,
University of Strathclyde, 75 Montrose Street, Glasgow, Scotland
E-mail: esru@strath.ac.uk

Abstract

We present a numerical model for 3D time resolved wake calculations from marine current turbines. Since the wakes are characterised by the shedding of a vortex sheet from the rotor blades, we have constructed the model based around the vorticity transport equations. A vortex sheet may be considered a jump contact discontinuity in tangential velocity with, in inviscid hydrodynamic terms, certain kinematic and dynamic conditions across the sheet. The kinematic condition is that the sheet is a stream surface with zero normal fluid velocity; the dynamic condition is that the pressure is equal on either side of the sheet. The kinematic condition is satisfied at the trailing edge only, via an approximation of the Kutta condition. The shed vorticity is the span-wise derivative of bound circulation, and the trailed vorticity is the time derivative of bound circulation, and is convected downstream from the rotors using a finite volume solution of vorticity transport equations.

Keywords: Marine Current Turbine, Wake, Unsteady Modelling, Vorticity Transport

Nomenclature

Re	=	Reynolds number
p, C_p	=	Pressure, pressure coefficient
$\mathbf{e_i}$	=	i^{th} Cartesian basis vector
$\mathbf{u}, \mathbf{U}$	=	Eulerian velocity field, velocity vector
ω	=	Vorticity vector
Φ	=	Velocity potential
$\mathbf{K}_{[...]}$	=	Biot-Savart kernel function in $[\ldots]$
$\mathbf{R}, R$	=	Displacement, distance
δ	=	Kernel smoothing parameter
$\mathbf{x}_{[...]}$	=	Position of point $[\ldots]$ in Cartesian space
$\mathbf{a}_{[...]}$	=	$[\ldots]^{th}$ Taylor (multipole) coefficient tensor
$b_{[...]}$	=	$[\ldots]^{th}$ recurrence coefficient
$\mathbf{m}_{[...]}$	=	$[\ldots]^{th}$ cluster moment (vorticity) tensor
β	=	Flux limiter parameter
ψ	=	Flux limiter; yaw angle
Ψ	=	Azimuth angle - zero along z-axis
Γ	=	Circulation
ν	=	Kinematic viscosity

Subscripts

$\mathbf{k}$	=	3D multi-index (k_1, k_2, k_3)
c	=	Value at cluster centre
∞	=	Free-stream
t	=	time
C	=	Cell centred value C
Str	=	Associated with vortex stretching/tilting
$Prod$	=	Associated with vorticity production

Superscripts

$\bar{\ }$	=	mean value
n	=	time step

1 Introduction

The performance of a marine current turbine is contingent on the inflow at the rotor, and the inflow itself is primarily determined by wake induced effects [1]. The wake induced flow velocities are themselves determined by the hydrodynamics of the rotor leading to the statement that a fundamental treatment of the wake physics is essential if the performance of a rotor is to be determined correctly. This is especially important in situations where there is a strong spatio-temporal variation in rotor inflow (with similarly coupled wake effects) such as a turbine operating in yaw or in the wake of another turbine – indeed these off-design conditions often pose a problem when modelling these devices.

The inflow at the rotor is strongly coupled to the wake vorticity induced velocity. A vortex sheet is shed from the blades at the root and tip due to leakage to the suction side, and along the span due to complex boundary layer phenomena. Considering a vortex sheet as a step discontinuity in tangential velocity in a parallel shear flow, at the smallest scale parcels of fluid must be rotating and, due to viscosity, inducing a velocity in their neighbours - figure 1. This is dynamically unstable and rapidly breaks down with the sheet curling up into familiar vortex wake shapes, with the velocity at a point on or near the sheet dependant on the shape and strength of the

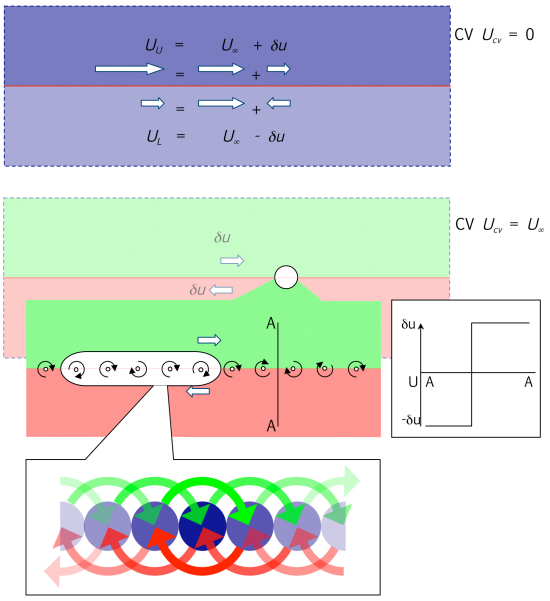

Figure 1: Considered as a delta-function in velocity, a vortex sheet curls up into familiar wake shapes as instabilities propagate

whole sheet. Blade element/momentum theory (BEMT) relies on modelling the rotor as a set of *isolated* two-dimensional blade elements to which we can then apply 2-dimensional aerodynamic theory individually and then perform an integration to find thrust and torque. The ramification is that we must assume uniform inflow over the annular elements in addition to the fact that there must be zero flow normal to stream-tube boundaries. Of course, a number of post-fixes exist in order to append to and correct for the additional physics, however, these are purely in terms of free-parameters of empirical derivation. The dependency of the quality of BEMTs prediction on the empirical input was highlighted during the blind comparison of wind turbine predictive codes against the U.S. National Renewable Energy Labs' Unsteady Aerodynamic Experiment [2] (NREL-UAE) , especially with regard to the treatment of 2D section characteristics [3] which are found to predict poorly 3D rotational effects, especially at the root, and the inclusion of unsteady effects via often physically parsimonious computational post-fixes.

The second treatment, computational fluid dynamics (CFD), is able to fundamentally treat the vast majority of the pertinent flow physics. Practical CFD involves a solution to the Reynolds Averaged Navier-Stokes equations (RANS) discretised onto a computational mesh, with suitable boundary conditions, however, the meshed basis poses a number of issues that currently limit the effectiveness of CFD in treating the rotor fluid dynamics. The first of these issues is the requirement to solve the fundamental transport equations in all elements of the mesh, even if these elements contain nothing of immediate interest. This requires the existence of a large domain, which must extend and be represented sufficiently far from points of interest that far-field boundary conditions may be applied.

The second issue is that solid surfaces (e.g. blades) must be represented with sufficient detail to accommodate small-scale fluid phenomena, for example the near-field effects associated with the development of the boundary layer (BL). When developing a mesh at the rotor blades the first requirement is that the surface is accurately rendered within the mesh – this is imperative since, once in the solver, although the number of grid points representing the surface can be increased, their position would be an interpolation between the points already in existence. The second requirement is that the boundary layer is resolved sufficiently that local instabilities are not lost in the mesh – this tends to increase the required number of surface cells yet further. Generally, in order to achieve the first requirement one would use [Sugoi Gomez-Iradi & George Barakos, personal communication], say, X-Foil [4] to determine the number of chord-wise points required to achieve a certain tolerance to foil shape representation with an inviscid solution, then use a viscous solution to determine the additional number of points to capture BL growth plus gain an idea of BL thickness to use when generating and "extruding" a surface mesh. For the Reynolds numbers associated with turbine blades we are looking at around 200 non-uniformly distributed chord-wise points to comfortably capture the geometry and BL, plus normal-to-surface spacing which is (initially) of the same order.

At the other end of the scale, we must allow sufficient evolution of the wake to capture pertinent physics. It has been suggested that in order to fully resolve a tip vortex, at least 15 points are required across the core [5], and since the wake notionally persists many chord-lengths downstream this poses a dichotomy: for a given finite computational overhead we can either cluster points to capture the rotor flow-field (the inner problem) or the wake (the outer problem), and it transpires that for a reasonable computational outlay (total CPU time of under, say, a couple of months) these are mutually exclusive and the tendency in current research is to focus on the rotor near-field giving rise to very aggressive mesh sizing functions in order to capture a sufficient fluid domain.

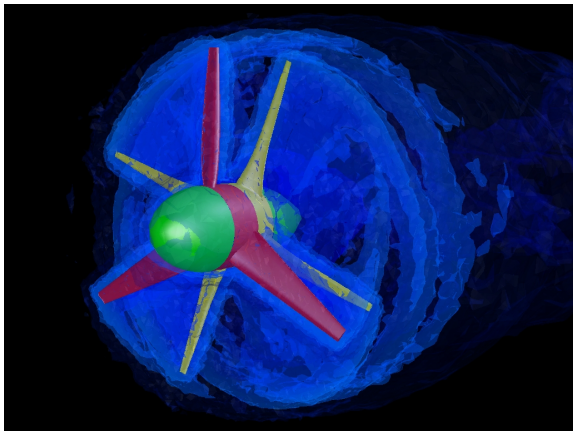

Figure 2: The vorticity field aft of a rotor is quickly smeared across the mesh when using traditional CFD approaches, and within a few turns is destroyed due to artificial dissipation.

The third and most important issue is concerned with artificial diffusion of the wake, and is a manifestation of the grid-based solution itself. The diffusions slip in via the coupled facts that 1) the grid must be a reasonable shape, preferably hexahedral to avoid too much interpolation, and the surfaces of the elements are not necessarily aligned with the local velocities and 2) the discretised governing equations themselves must be of sufficient accuracy to make the computation worthwhile. The problems associated with the first are intuitive and most significant – the transported variables are smeared spatially over the mesh due to non-physical, local truncation error (LTE) related diffusion – and give rise to the second which is a result of the fact that in order to obviate this a higher order discretisation scheme must be used, entailing oscillations in the solution around discontinuities which must be damped via an additional artificial viscosity. Properly implemented, and in smooth regions, the artificial viscosity should be negligible as they are designed to operate below the LTE; however, the wake vorticity from a turbine occupies a relatively compact and coherent region, the edges of which are discontinuous over adjoining computational cells, and if these cells are, due to cell clustering, relatively large so too is the LTE and thus the additional effect of the damping. Even with specifically designed, and (apparently) well posed and implemented problems, wake breakdown cannot be properly prevented, see for example figure 2

2 Introduction to the Current Method

In an effort to move away from the limitations of these standard approaches, wake modelling methods have evolved in which the wake vorticity is treated directly, and not as a by-product of the velocity field as in traditional CFD, or an extension/post-fix as in BEMT. These methods fall broadly into two camps: the first is a panel based potential flow solver, where bound and shed vorticity is represented using panels composed of combinations of filaments of vorticity, sources and sinks as solutions of the Laplace equation – see for example [6]; the second is a specific treatment of a vorticity conservative formulation of the Navier-Stokes (NS) equations. The current method falls (mainly) into the latter and is described fully below. The former method has found use in rotor-craft aeromechanics and wind turbine simulations with varying success – a limitation being that the time-evolution of the wake is subject to accumulating computationally induced errors leading to (Kelvin-Helmhotz) instabilities after a finite time. In order to get round this, a number of approaches use a prescribed wake geometry via either a prescribed screw-surface (e.g. the newly implemented code in AeroDyn [7]) of by correlation with experiment (e.g. [8]), or truncate a free wake after a certain age as in [6] and [9]. Further advances in this approach use vortex particles to satisfy the vorticity transport equation , e.g. [10], and this method appears to be the logical progression of these boundary element methods. The NREL-UAE experiments blind comparison demonstrated the shortcomings of these methods, however, the truncated-wake approach remains useful in current rotor-craft analysis, and the screw-surface method still provides a quick and valuable insight within certain design scenarios.

2.1 Method

An exciting and highly promising alternative to primitive-variable CFD lies in the vorticity-velocity formulation of the NS equations, and forms the core of the current model, which is based on that of Brown and coworkers e.g. [11], [12]. The governing equations are derived from the incompressible, pressure-velocity (primitive variable) NS equations:

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 \\ \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} &= -\frac{1}{\rho} + \upsilon \nabla^2 \mathbf{u} \end{aligned} \tag{1}$$

by taking the curl to yield the vorticity transport equation:

$$\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega = \omega \cdot \nabla \mathbf{u} + \upsilon \triangle \omega + S \tag{2}$$

for the vorticity field defined by

$$\omega = \nabla \times \mathbf{u} \tag{3}$$

The appended compound source term S comes into play in introducing vorticity to the flow, as well as during application of flux-limiters and other terms (discussed later). The computational domain is divided into cubic cells through which the vorticity is free to move according to the governing equations. As the transport variable is vorticity, we require some method of recovering the velocity field from the vorticity distribution during the calculation. This is done via the Biot-Savart law relating the velocity induced by a vortex with the range of interaction and vortex strength. The Biot-Savart law may be written in integral form so as to accommodate all the distributed vorticity throughout the computational domain, thus the influence of the j^{th} vorticity containing cell on a point at some displacement $\mathbf{R}_j$ is:

$$\mathbf{u} = -\frac{1}{4\pi} \sum_{j=0}^{n} \frac{\mathbf{R}_j}{\left|\mathbf{R}_j\right|^3} \times \omega_j \tag{4}$$

Since we require the velocity at all cells containing vorticity, we now have in effect an n-body problem requiring $O(n^2)$ effort calculations to gain the required velocities, and it would seem sensible to use one of the accelerated, so called fast multipole methods (FMM) , algorithms to perform the calculation. In order to solve the finite-volume version of the vorticity-velocity form of the Navier-Stokes equations, it is necessary to calculate the fluxes through the cell faces of the mesh. This requires the velocities at the cell face centres be known, and this can be accomplished by a combination of a direct summation of equation 4 and through use of a fast multipole expansion.

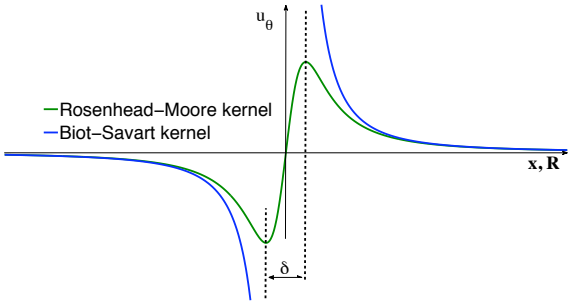

Figure 3: The smoothed and Biot-Savart kernels.

The approach adopted, and roughly as applied in [12], follows the mathematical formulation for the multipole expansion presented in [13] and is structured loosely according to the methodology proposed by [14] for fast particle interactions via the Laplace equations. The principle is to render local vorticity fields into a differentiable form whereby the velocity field at some remote point can be evaluated by means of Taylor type series expansions. In practice this entails representation of the vorticity field within a source cluster as local expansion about the cluster centre, and an additional expansion representing the velocity field at some distant target cluster whose components are a result of the source cluster expansion.

In order to perform the expansions, the Biot-Savart kernel is modified by addition of a smoothing parameter (δ), which destroys the singularity and results in the Rosenhead-Moore kernel $\mathbf{K}_\delta$ (see figure 3). Fortunately it transpires that the Rosenhead-Moore kernel is the gradient of a form of the regularised solution to Laplace's equation in 3D (the Plummer potential):

$$\begin{aligned} \mathbf{u} &= \sum_j \mathbf{K}_\delta \times \omega_j \\ \mathbf{K}_\delta &= -\frac{1}{4\pi} \frac{\mathbf{R}_j}{\left(|\mathbf{R}_j|^2 + \delta^2\right)^{\frac{3}{2}}} \end{aligned} \tag{5}$$

2.2 Methodology

Using multi-index notation it is proven in [13] that for the velocity induced at the i^{th} distant point $\mathbf{x}_i$ by a cluster of n vorticity containing cells at points $\mathbf{y}_j$ centred at $\mathbf{y}_c$:

$$\begin{aligned} \sum_{j=0}^{n} \mathbf{K}_\delta(\mathbf{x}_i, \mathbf{y}_j) \quad \times \quad \omega_j &= \sum_{j=0}^{n} \mathbf{K}_\delta(\mathbf{x}_i, \mathbf{y}_j + (\mathbf{y}_j - \mathbf{y}_c)) \times \omega_j \\ &= \sum_{\mathbf{k}} \frac{1}{\mathbf{k}!} D_y^{\mathbf{k}} \mathbf{K}_\delta(\mathbf{x}_i, \mathbf{y}_c) \times \sum_{j=0}^{n} (\mathbf{y}_j - \mathbf{y}_c)^{\mathbf{k}} \omega_j \\ &= \sum_{\mathbf{k}} \mathbf{a}_{\mathbf{k}}(\mathbf{x}_i, \mathbf{y}_c) \times \mathbf{m}_{\mathbf{k}}(c) \end{aligned} \tag{6}$$

Here the terms $\mathbf{a}_\mathbf{k}$ and $\mathbf{m}_\mathbf{k}$ are respectively the $\mathbf{k}^{th}$ Taylor term of the derivative D about $\mathbf{y}$ of $\mathbf{K}_\delta$ at the source cluster centre, and the clusters moment of vorticity, again evaluated at the cluster centre and, in practice, the expansion is limited to a finite $\mathbf{k}$. The gradients of this potential (and in fact any function which satisfies a linear differential equation with polynomial characteristic equations) can be found efficiently via recursion. In this case the solution is of the form:

$$\begin{aligned} &\|\mathbf{k}\| R^2 b_\mathbf{k} - (2\|\mathbf{k}\| - 1) \sum_{i=1}^{3} (x_i - y_i) b_{\mathbf{k}-\mathbf{e}_i} \\ &+ (\|\mathbf{k}\| - 1) \sum_{i=1}^{3} (x_i - y_i) b_{\mathbf{k}-2\mathbf{e}_i} = 0 \end{aligned} \tag{7}$$

for all positive $\mathbf{k}$, where $\mathbf{e}_i$ is the i^{th} Cartesian basis vector and the $\mathbf{a}$ (Taylor) matrix may be reconstructed via:

$$\mathbf{a}_\mathbf{k} = \sum_{i=1}^{3} (\mathbf{k}_i + 1) b_{\mathbf{k}-\mathbf{e}_i} \mathbf{e}_i \tag{8}$$

The velocity field is computed at the target cluster centre by performing the cross product of the vector elements of the $\mathbf{a}$ and $\mathbf{b}$ tensors, and the velocity at a point nearby may be attained by translation of the centre of the expansion (similar to a phase shift) before collapsing the derivatives to recover a velocity. The process is summarised in figure 6 . The finite volume method (FVM) requires that the domain is divided into volumetric cells, and this can be linked with the structural requirements for an effective fast multipole application. Therefore a sensible data-structure based around octrees seems to be the prime candidate, and the approach in [15] has been used (although certain other methodologies, such as B, R or K-d trees and non-uniform adaptive trees were considered). The idea is that at the finest level the domain is tessellated into cells, but only cells containing vorticity and their neighbours are retained. Nearby cells are linked into groups of up to eight based on their proximity to what will become their cluster-centre. The process is repeated recursively with the newly created clusters until the entire vorticity field is contained in a single super-cluster. The principle advantages of this approach are the geometric similarities (figure 5a) between different levels of the octree which allows for the fact that, per level, the number of calculations actually required for the $\mathbf{a}$ terms in the FMM can be kept to a minimum. Once the tree is constructed, the cell/cluster neighbours at each level are determined, another advantage of octrees being that this may be done relatively efficiently. The cluster moments are calculated at the centre of each cluster, then there is a sweep from the super-cluster down to the cells where at each level the velocity field at the cluster's centre is inherited by translation from its parent. Induced velocity fields are calculated at the centre of each target cluster based on the effects of the vorticity moments of source clusters which are the children of the target cluster's parent's neighbours. When the cell centred velocity and vorticity are known throughout the domain, FVM may be used.

2.3 Finite Volume Method

Applying divergence theorem to the computational cells, over a time-step Δt the vorticity transport equa-

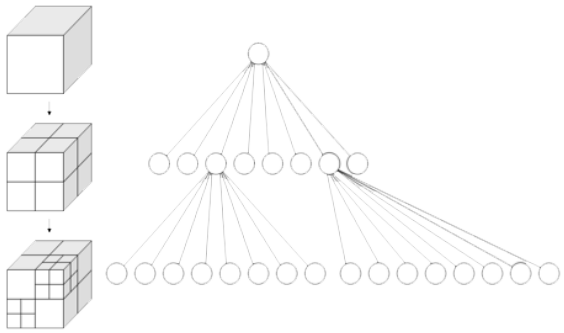

Figure 4: Octrees provide a suitable recursive geometric structure.

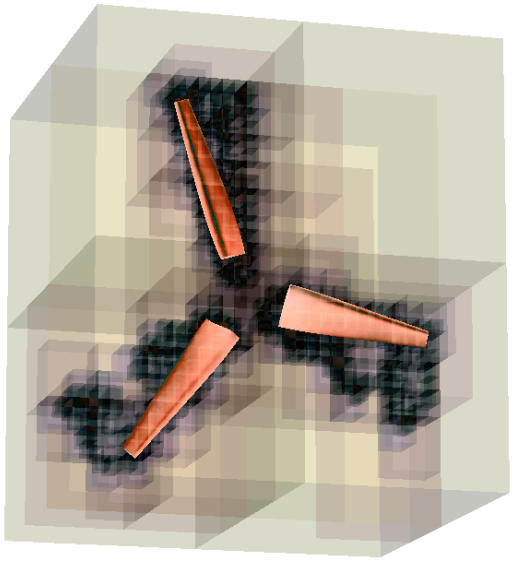

Figure 5: Octree data structures applied generating an adaptive Lagrangian mesh in the vicinity of the blades

tion may be written as follows for the i^{th} component of vorticity:

$$\int_{\Delta t}\int_{V}\frac{\partial \omega_i}{\partial t}dV\,dt+\int_{\Delta t}\int_{A}\omega_i(\mathbf{u}\cdot\mathbf{dA})dt=\int_{\Delta t}\int_{V}SdV\,dt \tag{9}$$

Here, in order, the terms represent the rate of change in i vorticity in time, due to convection and a source term accounting for axis tilting/stretching, diffusion and vorticity creation. The terms A, V are face areas and volumes of the computational cell. The stretching/tilting term merits some minor discussion: it is responsible for the change in axis orientation (tilting) and magnitude (by stretching of the vorticity). Physically the tilting is most intuitive, but both are the result that given a velocity differential, a vortex filament will have its ends either pushed together or apart. If the differential acts as a shear over the filament the axis will rotate orthogonally; otherwise it will be either stretched or compressed, whereby conservation of angular momentum (thus circulation) will change the vorticity appropriately. Regarding the viscous term, since we are dealing with high Reynolds number flow, an assumption that we operate in the limit of zero viscosity for regions away from surfaces has be adopted, with the knowledge that there will be

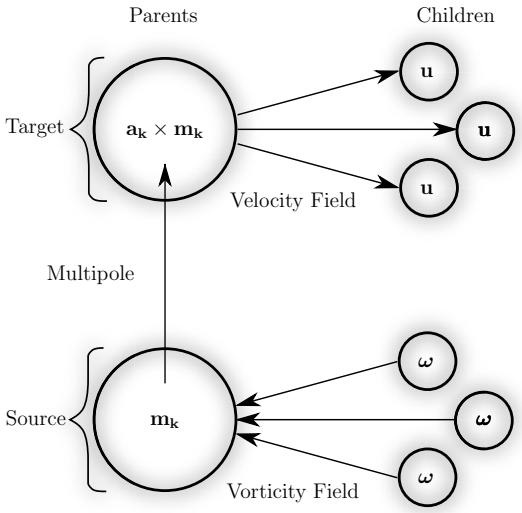

Figure 6: Relation between expansions used in velocity calculation.

some diffusive losses associated with the grid, and only a very basic finite-difference to the Laplacian is calculated (if desired). Future work will seek to address this term more fully, and determine the appropriateness of

One of the tenets of the finite-volume approach is that the quantity being evaluated by the general transport equation is assumed to be integral averaged over the volume of the cell with the cell integrated value concentrated at the centre of the cell. As such an approximation to the cell face value for the vorticity must be found given that the vorticity field is represented by piece-wise constant vorticity strengths over adjacent finite volumes, separated by a discontinuity. The flux between these cells must be carefully represented numerically if the relatively compact vorticity region is to be preserved – especially at the advancing interface of vorticity containing and "empty" domains. Here the local gradients would be very steep but highly localised, and when represented on a mesh would either be missed completely, or be smeared far more globally than was realistic with additional non-physical oscillations. A flux limiter detects such local extrema, and preserves them while, most importantly, retaining the small scale nature of the disturbance. In addition it is desirable that there are no under or overshoots in the solution at these discontinuities - that is, we want the approximation to be as close to matching the discontinuity shape as possible without generating any spurious oscillations in the solution (which, when notionally stable, also serve damp out sharp fronts).

There are a number of options available here for the evaluation of vorticities at the cell faces: we have adopted a second-order centrally weighted essentially non-oscillatory (CWENO) scheme. The gradients used by the CWENO scheme are calculated via a parameterised `MINMOD` flux limiter is used in which monotonicity in the solution and wiggles (Gibbs phenomena - running waves in the numerical solution due to amplification of the odd-order derivatives of the LTE) are respec-

tively enforced and suppressed. The flux limiter essentially limits the effective spatial gradients to physically realistic values when a sharp front or discontinuity is present. So at each cell face f the flux limiter chooses some weighted average between suitable high and a low resolution schemes for the derivatives at the cell face:

$$\omega_{i,f} = \omega_{i,C} + \frac{1}{2}\psi_{\texttt{MINMOD}}(\nabla\omega_i, \beta)\cdot d\mathbf{A} \tag{10}$$

The finite volume formulation of the governing equations may be written

$$\int_{\Delta t}\int_V \frac{\partial}{\partial t}\omega_i dV dt = \int_{\Delta t}(b_P\omega_{i,P} + \sum_{faces} a_j\omega_{i,f} + S_{Str} + S_{Prod})dt \tag{11}$$

where the terms a and b are derived from cell face flux expressions, with the source (S) terms accounting for the tilting/stretching effects due to the velocity differential over the cell as discussed previously, and the flux contributions from the flux limiters.

The current evolution of the model uses a second order semi-discrete formulation for the cell face fluxes, namely a MUSCL type reconstruction via a Riemann-solver free technique [16] which uses an approximation to the eigenvalues of the flux Jacobians to arrive at estimates for characteristic (wave propagation) speeds and thus provide the limiters with the information they require to "de-clutch" the higher order method. Once the value of $\frac{\partial\omega}{\partial t}$ has been evaluated we advance the solution using Heun's second order predictor-corrector method.

3 Rotor Representation

The vorticity transport equation contains a source term through which vorticity generation may occur. In the case of the current method, this source term can be broken down into a tilting/stretching component, a diffusion component if required and the vorticity source due to rotor modelling.

The rotor is represented using traditional boundary-element methods: either a lifting line or surface representation of the blades can be used, or a full 3D panel code. The panel code representation involves discretising the surface of the rotor blades, hub and any other bodies into a number of quadrilateral panels and is capable of therefore modelling bodies with thickness. A linear or constant strength source of strength σ and/or doublet (strength μ) is distributed over each panel, and wake is shed from trailing edges in the form of vortex loop panels.

The formalism of this model is well known (see Katz and Plotkin for example) and only the core constituents are described herein. Given a velocity field that may be described by Helmholtz decomposition as having a solenoidal and an irrotational part:

$$\mathbf{u} = \sum_{i=1}^{3}\mathbf{U}_{\infty\mathbf{e}i} + \nabla\Phi \tag{12}$$

where Φ is the perturbation potential associated with fluid boundaries, at a solid boundary we expect that the flow of fluid will be equal to the boundary motion in order to satisfy the condition that fluid does not penetrate the boundary (given that this is an inviscid method we do not expect to apply a no-slip condition). This being so, for all panels making up a body we solve for Φ that satisfies this condition accounting for the interaction between all panels making up the body and also the shed wake induced velocity. This is accomplished via solution of a system of i linear equations

$$\frac{\partial\Phi_i}{\partial\mathbf{n}_i} = -[\mathbf{u}\cdot\mathbf{n}]_i \tag{13}$$

which is made unique via an appropriate Kutta condition.

Vortex ring elements are shed from a proto-wake element attached to user specified points in accordance Helmholtz and Kutta conditions, and it is these which are used as the production source in equation 12. The panels are shed and form a growing patch which is allowed to convect downstream from the blade over a number of sub-steps over the global time-step. Once sufficient panels have been shed, and given that vorticity can only exist in the wake (as we have inviscid surface boundary conditions), we generate a number of computational cells to contain fully the patches which have been created. From here there are a several options for coupling with the vorticity transport equation, since the wake patches are specified in terms of vortex filaments. These include:

1. Using Stokes theorem to convert the circulation around the planes of a cell (see figure 7a) to the vorticity in the cell, so for example in the $x-y$ plane we have:

$$(\overline{\omega_z})_{i,j} = \frac{1}{A}\Gamma_{i,j} = \frac{1}{A}\oint_{l(X,Y)}(U,V)\cdot dl \approx \frac{\Gamma_{i,j}}{4\Delta X\Delta Y} \tag{14}$$

2. Via the definition of vorticity (equation 3) using finite difference approximations to the curl of the velocity field (figure 7b);

3. An engineering approximation whereby the vortex filaments are segmented and represented by a number of "blobs" which are dissolved onto the mesh (see figure 8).

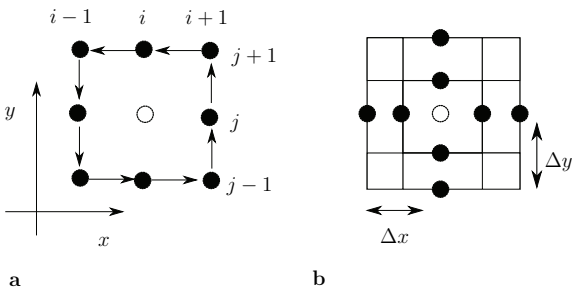

Figure 7: The vorticity can be calculated using 12 additional velocity calculations around the cell and using Stokes theorem

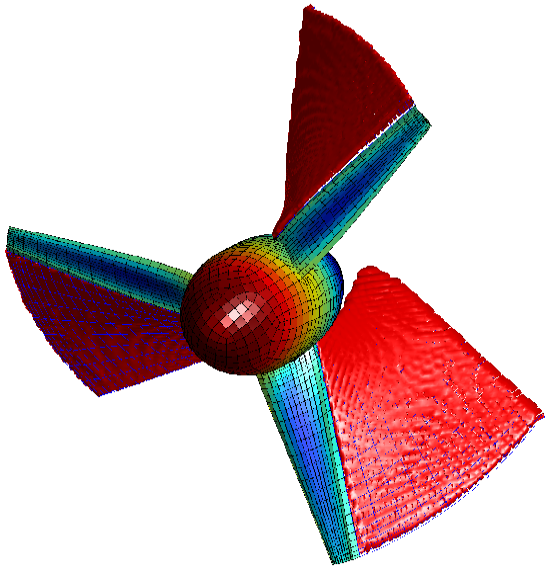

Figure 8: Panel to blob to volumetric mesh interpolation for the vorticity field aft of a turbine rotor. The turbine is coloured by doublet strength μ

The first method mirrors the techniques used in experimental work in attaining vorticity fields from PIV data, however shares the disadvantages of the second method in that not only must multiple additional velocities be calculated but finite differencing errors can yield over/underestimation of vorticity strength. In actuality, there are instabilities introduced during each method, however the second method can be used to ensure a divergence free velocity, and in practice we use this as a discrete approximation to the tilting/stretching contribution.

Once the vorticity has been transferred into the octree the wake panels are destroyed and the process begins again. The number of sub time intervals in which the panel wake representation is advanced is dependant on both the Courant number type condition in the FVM, the desired reporting resolution and device mechanics. Given that the model does not represent truly coupled fluid structure interaction in its current incarnation, we limit the age of the wake patches to that which is sufficient to interpolate the vorticity. The unsteady Bernoulli equation is used to recover the pressures on submerged body surfaces

$$C_p = 1 - \frac{\left|\mathbf{u}_{body}\right|^2}{\left|\mathbf{U}_\infty\right|^2} - \frac{2}{\left|\mathbf{U}_\infty\right|^2}\frac{\partial \Phi}{\partial t} \tag{15}$$

with $\mathbf{u}_{body}$ being evaluated as the vector sum of kinematic, wake panel, fast-multipole and free-stream velocity contributions (and is tangential to the body surface as per the slip condition, equation 13).

4 Preliminary Results

Since the FMM is an approximation to the cell face velocities, it is important to be able to quantify the error due to truncation of the infinite series. This can be done a number of ways, but essentially involves summing the largest order terms which have been discarded during the construction of the various multipole expansion components.

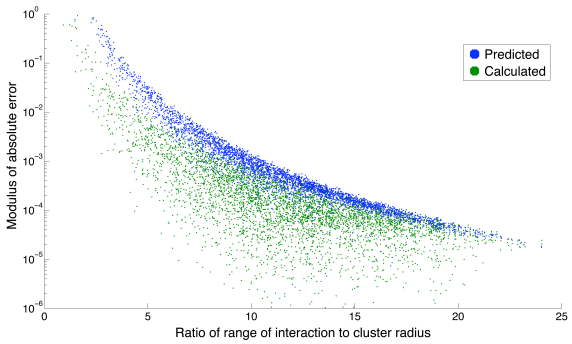

Figure 9: Accuracy of FMM calculation for the ratio of range of interaction to cluster radius. There are 35 retained Taylor terms.

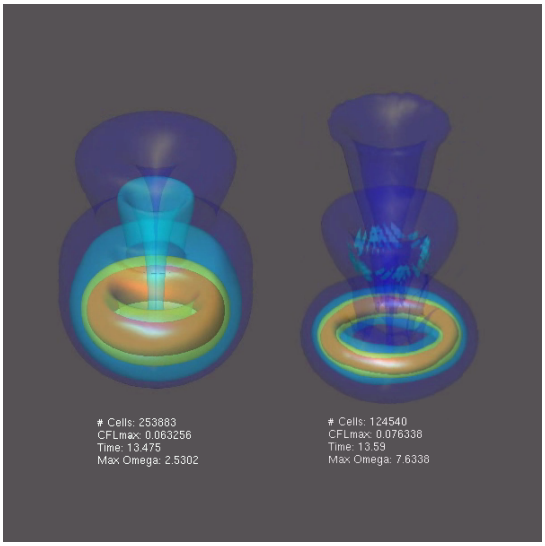

Figure 10: 3D evolution of a circular vortex sheet using `MINMOD` (left) and `SUPERBEE` (right) limiters ($\beta = 1$ and 2 respectively)

4.1 3D Results

The code is then used to simulate the cases as used by [13], that is the evolution of a vortex sheet parameterised as follows:

$$\begin{aligned} x &= \left(1-\Gamma^2\right)^{\frac{1}{2}}\cos\theta \\ y &= \left(1-\Gamma^2\right)^{\frac{1}{2}}\sin\theta \\ z &= 0 \end{aligned} \tag{16}$$

$$(0 \le \Gamma \le 1)\,(0 \le \theta \le 2\pi) \tag{17}$$

The dropoff in circulation Γ at the edge of the sheet will induce the rolling up of the sheet, into a 3D ring - this is a familiar occurrence as it is the process that generates smoke rings, and the results should be intuitive. The

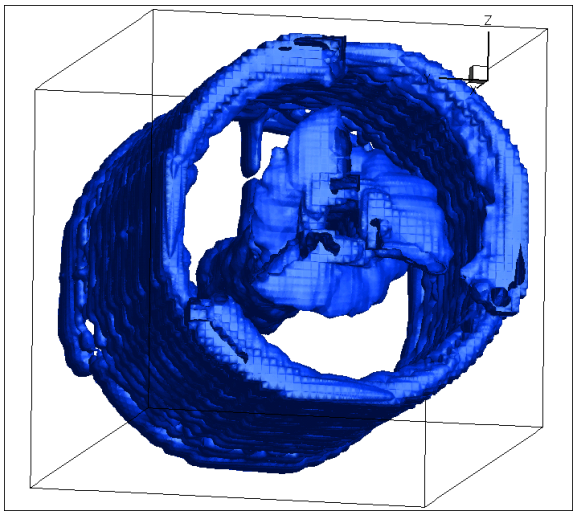

Figure 11: Isosurface of vorticity showing tip vortices and starting vortex remnant as seen looking down x-axis.

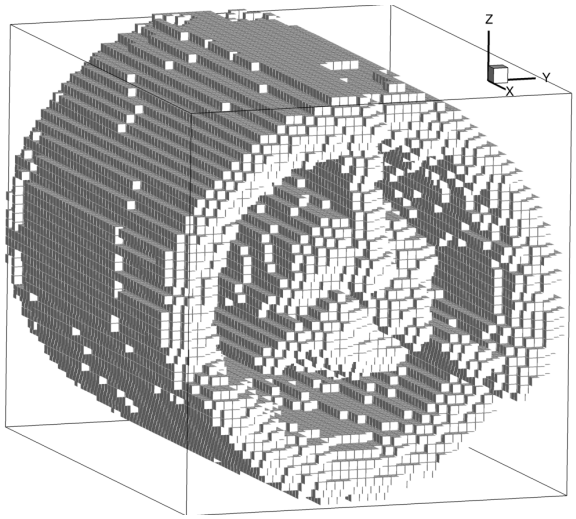

Figure 12: All active computational cells within the time-step.

results shown give the evolution after approximately 15 seconds of two cases, the left using the `MINMOD` flux limiter with parameter β set to 1 which is the most diffusive, and the right using a $\beta = 2$, equivalent to a `SUPERBEE` limiter, the most compressive. The difference is severe - the position and shape of the resulting flow features are significantly different, with the diffusive limiter spreading the vorticity over about double (254,000 compared to 125,000) the number of cells of the compressive case indicating that not only is the process more "lossy" but that it will also take longer to compute. However, it is likely that a real world case would be somewhere in between, therefore final choice of the flux limiter parameter should be based on the results of validation.

4.2 Preliminary Turbine Results

Further to the model proving results above, and subsequent to the previous work [17] we have moved from lifting line solutions to a 3D panel code. Figures 11 and 12 show results obtained using a model of a three bladed tidal turbine, operating at a tip speed ratio of 6 with a positive uniform initial inflow along the x-direction.

As can be seen from figure 11, the wake is preserved for some considerable distance downstream, until it approaches the (artificially - for test purposes) imposed limits of the domain. A principle advantage of the Lagrangian mesh formulation is that the domain is essentially limited only by memory constraints of the computer the code is run on, and can be arbitrarily large as long as vorticity is sparsely distributed. Figure 12 shows the distribution of vorticity containing cells within the domain, this time looking along the x-axis towards the origin past the still preserved starting vortex. Here the total number of cells is around 75,000 for this case and the model clearly preserves the starting and tip vortices.

In order to get an idea of code predictive performance, the model has been used to simulate the NREL-UAE. The Phase 5 UAE were a series of experiments conducted using the NASA Ames large wind tunnel and a full size (5m radius) NREL wind turbine, and these have provided a very high quality data-set for use verifying code performance. The turbine in this instance was equipped with two blades, with a NREL S809 section, linear thickness and cubic twist distributions. In these simulations no hub is present, however this was simply to ensure adequate resolution of inboard tip vortices. As part of the quality assurance on their data, NREL prove repeatable pressure distributions at azimuths of 0 and 180° with an inflow velocity of $13ms^{-1}$ at $\psi = 30°$ yaw; these are presented in figures 14 and 15 below along with predictions from the current model. While it is noted that a wind turbine is not an ideal test-case for a code designed for marine current turbines, the volume of data available from the NREL tests make them a valuable resource if one is prepared to overlook the slight changes in physics.

There is clearly some discrepancy between the C_p results, and this will manifest itself as errors in load calculations. At both azimuthal stations the pressure side of the blade can be seen to match fairly well, especially for the low incidence, advancing ($\Psi = 180°$) case. On the retreating blade, the disparity between the measured results is due to shedding a dynamic stall vortex - a purely viscous event which cannot be captured by the model in it's current form. Some work has been done by, for example [18], where a vortex sheet is shed from the blade leading edge when certain dynamic stall onset criteria are met enabling essentially inviscid codes to model this. In this situation the case was chosen as a test case, and it is noted that this represents a significantly off design condition. Furthermore, the blade sees an inflow at an incidence of around 35° - well above static stall (and therefore inviscid) conditions.

The advancing blade local angle of attack reaches a maximum at around 15° and therefore operates just inside the envelope where out-with the limiting assumptions of the model begin to break down. Current work seeks to identify the root of the massive over-prediction on the suction side and although immediate suspects are the inability to predict separation, it is not obvious if

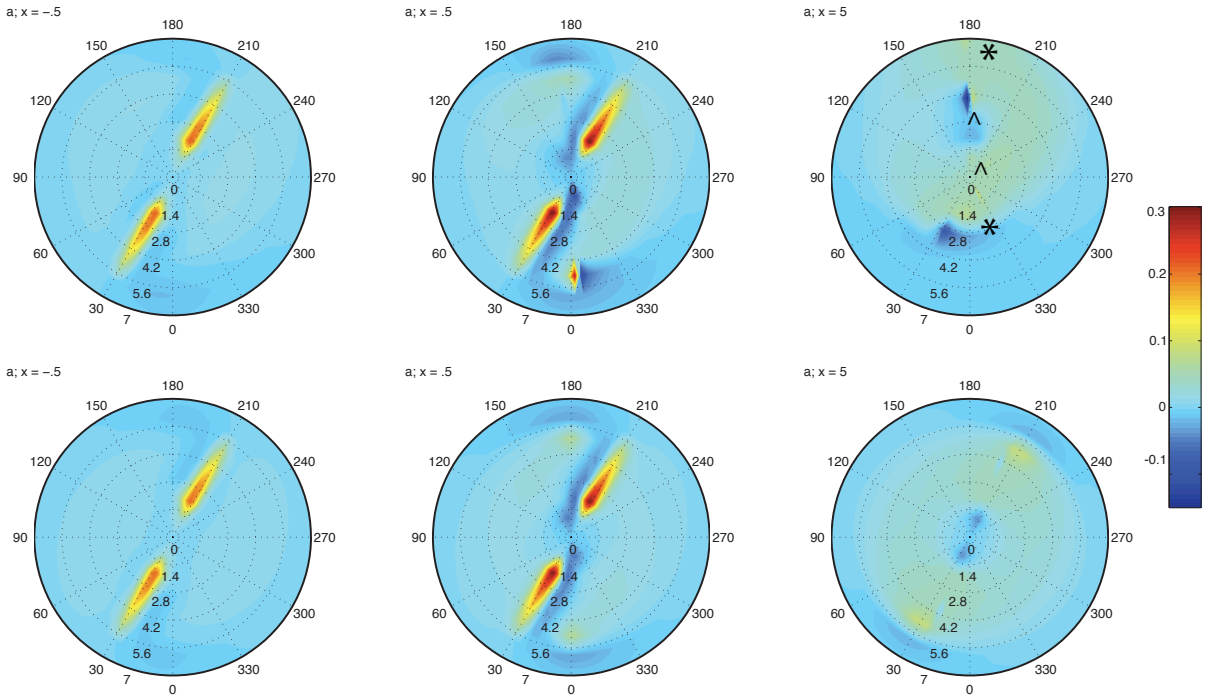

Figure 13: Axial induction factors ($a \equiv \frac{U_\infty - U}{U_\infty}$) computed for NREL UAE turbine. $U_\infty = 13ms^{-1}$, Upper: $\psi = 30°$; Lower: $\psi = 0$.

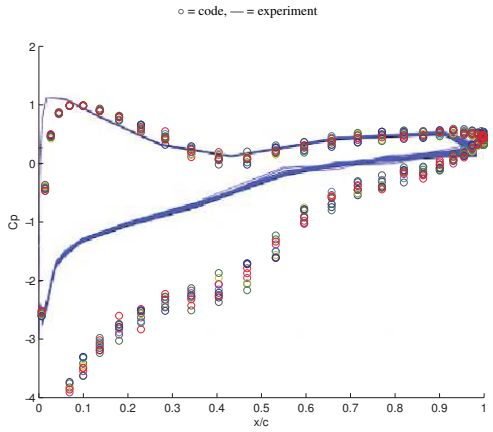

Figure 14: Pressure coefficient at $r/R = 0.47$ and $\Psi = 180°$.

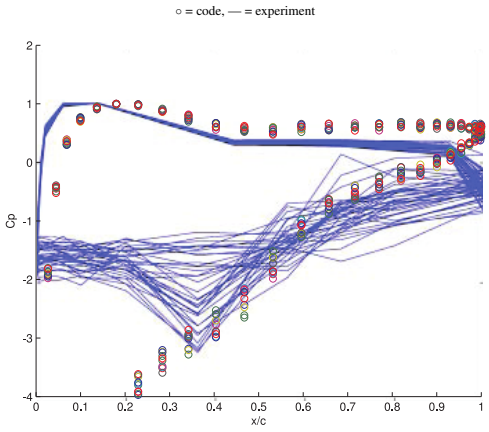

Figure 15: Pressure coefficient at $r/R = 0.47$ and $\Psi = 0°$.

there is not some other process (or error) at work as investigations using simple wings *occasionally* indicate Kutta condition related errors.

Figures 13 show the computed axial induction factors just upstream of the rotor plane for the turbine operating in yawed and un-yawed conditions. It is interesting (from the perspective of, say, coaxial devices) to notice here how far the momentum deficit ($a > 0$) of a turbine extends in an upstream direction, and in both cases the root and tip vorticity is carried far downstream. In the yawed case at a position 0.5m downstream of the rotor, the strong influence of the outboard tip vortex can be seen - a feature which would not be captured using standard blade element models and a strong contender for the over-prediction seen in figure 15. The tip vortices are obvious on all but the top right section - this is due to them being carried "upwards" by the yawed flow, and the inboard vortices are marked with $\wedge$, the outboard with $*$. They can be seen to circulate in the region half-a-radius above the axis.

5 Future Work and Perspectives

The code developed here has been designed to overcome perceived shortcomings in standard "off the shelf" turbine analysis methodologies. Array modelling is likely to become more and more important as developers move from the prototype to deployment mode, so we require a fairly quick and appropriately physical method to analyse the device interactions. Extension of this model to multi-body problems has been included from inception via object-orientated programming, but at the moment the cost of computing influence coefficients at each time-step is a bottleneck. Future development entails high resolution validation campaigns agains known data to ascertain code behaviour, followed by reduced resolution modelling of increasingly large arrays of devices

under more realistic inflow boundary conditions.

6 Conclusions

We present a numerical model for unsteady wake modelling for marine current turbines based on a finite volume solution of the Navier-Stokes equations in vorticity conserving form. We obtain results indicating the importance for turbines of wake induced velocities and demonstrate that an appropriately limited high resolution finite volume method aids in retaining wake longevity. While there is not currently a sub-grid-scale turbulence model, artificial diffusive mixing still acts as a kind of viscocity even though great pains were taken to keep it to a minimum.

Current predictions over-estimate the suction on the upper surface of the blades, however the wake modelling aspect of the code works as intended. The model's inability to accurately model high yaw related effects is primarily due to the inviscid nature of the boundary representation. It is envisaged that future work will seek to adopt an appropriate three-dimensional Cartesian cut-cell approach to the vorticity boundary condition, however it is noted that the predictability of marine currents suggests that severe yaw events will be unlikely, and this plus the relative speed of the boundary element method aid in justifying the approach adopted.

Early results indicate that the modelling methodology is sound, and providing a rigourous validation campaign is adopted, a valuable open source tool will be the by-product of this work.

Obtaining the Code

The code is available under the GNU General Public License and can be obtained either by contacting the authors, or via svn:
https://peta-esru.mecheng.strath.ac.uk/repos/vort-transp

References

[1] M. O. L. Hansen, J. N. Sørensen, S. Voutsinas, N. Sørensen, and H. A. Madsen. State of the Art in Wind Turbine Aerodynamics and Aeroelasticity. *Progress in Aerospace Sciences*, 42(10):285–330, 2006.

[2] D. Simms, S. Schreck, M. Hand, and L. J. Fingersh. NREL Unsteady Aerodynamics Experiment in the NASA-Ames Wind Tunnel: A Comparison of Predictions to Measurements. Technical Report NREL/TP-500-29494, NREL, June 2001.

[3] Frank N. Coton, Tongguang Wang, and Roderick A. McD. Galbraith. An Examination of Key Aerodynamic Modelling Issues Raised by the NREL Blind Comparison. *Wind Energy*, 5(2-3):199–212, 2002.

[4] M. Drela and M. Giles. Viscous-Inviscid Analysis of Transonic and Low Reynolds Number Airfoils. *AIAA Journal*, 25(10):1347–1355, 1987.

[5] Jennifer Dacles-Mariani, Gregory G. Zilliac, Jim S. Chow, and Peter Bradshaw. Numerical/experimental study of a wingtip vortex in the near field. *AIAA Journal*, 33(9):1561–1568, 1995.

[6] Jean-Jacques Chattot. Helicoidal vortex model for wind turbine aeroelastic simulation. *Computers & Structures*, 85(11-14):1072 – 1079, 2007. Fourth MIT Conference on Computational Fluid and Solid Mechanics.

[7] Hugh D. Currin, Frank N. Coton, and Byard Wood. Dynamic Prescribed Vortex Wake Model for AERODYN/FAST. *Journal of Solar Energy Engineering*, 130(3):031007, 2008.

[8] Tonio Sant, Gijs Van Kuik, Wouter Haans, and Gerard J. W. Van Bussel. An approach for the verification and validation of rotor aerodynamics codes based on free-wake vortex methods. volume 2005 of *31st European Rotorcraft Forum*, pages 56–1, Florence, Italy, 2005. Confederation of European Aerospace Societies, CEAS, France.

[9] J. Gordon Leishman, Mahendra J. Bhagwat, and Ashish Bagai. Free-vortex filament methods for the analysis of helicopter rotor wakes. *Journal of Aircraft*, 39(5):759–775, 2002.

[10] Fabrice Maganga, Gregory Pinon, Gregory Germain, and Elie Rivoalen. Numerical Simulation of the Wake of Marine Current Turbines with a Particle Method. In *Procs of the 10th World Renewable Energy Congress*, Glasgow, Scotland, June 2008.

[11] R. Brown. Rotor Wake Modeling for Flight Dynamic Simulation of Helicopters. *AIAA Journal*, 38(1):57–63, 2000.

[12] R. Brown and A. J. Line. Efficient High-Resolution Wake Modeling Using the Vorticity Transport Equation. *AIAA Journal*, 43(7):1434–1443, 2005.

[13] Keith Lindsay and Robert Krasny. A Particle Method and Adaptive Treecode for Vortex Sheet Motion in Three-Dimensional Flow. *J. Comput. Phys.*, 172(2):879–907, 2001.

[14] H. Cheng, L. Greengard, and V. Rokhlin. A Fast Adaptive Multipole Algorithm in Three Dimensions. *J. Comput. Phys.*, 155(2):468–498, 1999.

[15] Charles Hirsch. *Numerical Computation of Internal and External Flows: Vols 1 and 2*. John Wiley and Sons, Inc., New York, NY, USA, 1988.

[16] A. Kurganov and E. Tadmor. New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations. *J. Comput. Phys*, 160:241–282, 2000.

[17] Tom McCombes, Cameron Johnstone, and Andrew Grant. Wake Modelling for Marine Current Turbines. In *To appear in Procs of the 2nd International Conference on Ocean Energy*, Brest, France, October 2008.

[18] J. Baltazar and J.A.C Falcão de Campos. Hydrodynamic Analysis of a Horizontal Axis Marine Current Turbine with a Boundary Element Method. In *Procs of the ASME 27th International Conference on Offshore Mechanics and Arctic Engineering*, Estoril, Portugal, June 2008.