AIAA JOURNAL
Vol. 43, No. 7, July 2005

Efficient High-Resolution Wake Modeling Using the Vorticity Transport Equation

Richard E. Brown* and Andrew J. Line†
Imperial College, London, England SW7 2AZ, United Kingdom

Computational fluid dynamic models, through their fundamental treatment of the flow physics, present a unique and powerful tool for analyzing the aerodynamics of helicopter rotors. Their effectiveness is often limited though by difficulties in retaining the structure and form of the rotor wake that result from their tendency to dissipate vorticity. The vorticity transport model, developed some years ago, addresses the problem of vorticity diffusion by solving the fundamental fluid dynamic equations in vorticity conservation form. A development of the original model is described that results in a more efficient computational implementation. The new model uses an adaptive grid system to increase grid resolution significantly, with minimal impact on computational cost. The new grid system is effectively boundary free, thus eliminating the need for numerical boundary conditions at the edges of the computational domain. In addition, the new grid system allows very efficient evaluation of the velocity field using a technique based on the Cartesian fast multipole method. The implementation of both the grid system and the velocity calculation is described, and the performance of the new model is validated against some well-known experimental data.

Nomenclature

A	=	rotor disk area, πR^2
a_k	=	kth Taylor coefficient of K_δ
b_k	=	kth Taylor coefficient of ϕ_δ
C_T	=	rotor thrust, scaled by $\rho A(\Omega R)^2$
K	=	Biot–Savart kernel
K_δ	=	regularized Biot–Savart kernel
m_k	=	kth moment of vorticity
N	=	number of grid cells
p	=	pressure in the flow
R	=	rotor radius
S	=	vorticity source
$\boldsymbol{u}$	=	flow velocity
$\boldsymbol{u}_b$	=	flow velocity relative to blade
Δ	=	cell edge length
δ	=	kernel smoothing parameter
μ	=	rotor forward speed scaled by ΩR
ν	=	kinematic viscosity of the flow
ρ	=	flow density
σ	=	rotor solidity
ϕ_δ	=	regularized Newtonian potential
Ω	=	rotor rotational speed
ω	=	vorticity
ω_b	=	bound vorticity

Introduction

COMPUTATIONAL fluid dynamic (CFD) methods have the potential to provide a fundamental insight into the complex aerodynamic processes governing the evolution of helicopter rotor wakes, employing, as they do, various numerical techniques to solve the Navier–Stokes equations that govern viscous fluid flow. Because the dynamics of the airflow around the helicopter are very well described by these equations, the power of CFD methods stem from their treatment of the rotor aerodynamics at a very elementary level of description.

CFD methods do, however, have limitations, which, if not properly addressed, limit the success of these techniques in modeling rotorcraft related flows. Given that the environment in which a helicopter rotor operates is dominated by complex and persistent vortical structures, any method selected to model a rotor wake needs to preserve the integrity of the vortical structure of the wake over significant periods of time. Standard CFD methods that rely on a primitive variable formulation of the Navier–Stokes equations, that is, in terms of velocity and pressure, are susceptible to excessive numerical dissipation of vorticity.[1] If not controlled, this dissipation leads ultimately to the loss of coherence of vortical structures within the flow and, hence, for instance, to significant error in predicting the loading on, and subsequent dynamics of, the rotor. A further limitation of CFD-based techniques lies in their rather intensive use of computational resources. Generally, the computational resources consumed by typical rotor CFD simulations require some concessions to be made regarding the range of flow scales that can be resolved within a given simulation. This is particularly a problem when CFD methods are used to study some of the more challenging rotorcraft problems, such as vibration, acoustics, and interactional aerodynamics, where multiple flow scales are involved in the processes of interest. Usually the only practical approach is to apply boundary conditions at some distance away from the region of interest, for example, the rotor, and, hence, to forego an explicit representation of the development of the flow far from the region of interest, often with rather poorly defined consequences for the accuracy of the computation. Obviously, the more efficient the computation in terms of memory and time usage, the less the solution need be compromised by the availability of these resources.

This paper describes a development of the vorticity transport model (VTM), originally developed by Brown,[2] that specifically addresses the issue of computational efficiency. The VTM simulates the flow using the vorticity–velocity form of the Navier–Stokes equations, allowing rigorous control of the effects of numerical diffusion and dissipation of vorticity. The advancements described in this paper rely on the fact that, to evolve the flow using the Navier–Stokes equations in vorticity–velocity form, only regions of the flow containing vorticity need to be contained within the computational domain. In the new version of the VTM, these regions of vorticity are tracked using a semi-Lagrangian adaptive grid system that follows the vorticity distribution as it evolves over time. This approach minimizes the number of grid cells and, hence, computational resources, required to model the rotor and its wake. In addition, the

Received 27 September 2004; revision received 10 January 2005; accepted for publication 24 January 2005. Copyright © 2005 by Richard E. Brown and Andrew J. Line. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0001-1452/05 $10.00 in correspondence with the CCC.

*Senior Lecturer, Department of Aeronautics.

†Postgraduate Research Assistant, Department of Aeronautics. Student Member AIAA.

grid system is effectively boundary free, allowing the computation at any moment to contain all of the vorticity generated by the rotor during the entire course of the simulation. Errors due to wake truncation at boundaries are, thus, avoided.

Finally, the new grid system allows a highly efficient algorithm, based on the Cartesian fast multipole method, to be used to evaluate the velocity field throughout the computational domain. The similarities in the algorithmic structures of the fast multipole method and the adaptive grid system lead to a particularly efficient numerical implementation. In particular, both methods use an octree-based hierarchical decomposition of the computational domain and, therefore, can share common data structures, yielding considerable savings in terms of memory usage and computation time.

Model Description

The Navier–Stokes equations governing the flow surrounding the rotor can be written in primitive variable (velocity–pressure) form as

$$\frac{\partial}{\partial t}\boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\boldsymbol{u} \tag{1}$$

$$\nabla\cdot\boldsymbol{u} = 0 \tag{2}$$

if it is assumed that the flow in the wake is effectively incompressible. (This assumption can usually be reasonably well justified, of course, except perhaps near to the tips of the blades or in parts of the cores of the wake vortices generated by particularly highly loaded rotors.)

Arguably, though, the most efficient way to model the vorticity-dominated aerodynamic environment of the helicopter rotor is to model the rotor wake directly as a time-dependent vorticity distribution in the space surrounding the rotor. Taking the curl of Eq. (1) yields the unsteady transport equation

$$\frac{\partial}{\partial t}\omega + \boldsymbol{u}\cdot\nabla\omega - \omega\cdot\nabla\boldsymbol{u} = \nu\nabla^2\omega \tag{3}$$

for the vorticity field $\omega = \nabla\times\boldsymbol{u}$. Vorticity now becomes the fundamental conserved flow property, and the transport of vorticity becomes the mechanism via which the flow evolves. In the limit of zero viscosity, the viscous term in Eq. (3) becomes nonzero only on surfaces immersed in the flow and, thus, can be replaced by a local vorticity source. Thus, for the high-Reynolds-number flows relevant to most rotor aerodynamic problems, the vorticity transport equation can be written as

$$\frac{\partial}{\partial t}\omega + \boldsymbol{u}\cdot\nabla\omega - \omega\cdot\nabla\boldsymbol{u} = S \tag{4}$$

where the source term S accounts for the vorticity production associated with the generation of aerodynamic loads on the rotor blades.

The differential form of the Biot–Savart relationship

$$\nabla^2\boldsymbol{u} = -\nabla\times\omega \tag{5}$$

relates the velocity at any point in the flow to the vorticity distribution. The solution to this equation can be written in integral form as

$$\boldsymbol{u}(\boldsymbol{x}) = \int_V K(\boldsymbol{x},\boldsymbol{y})\times\omega(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \tag{6}$$

where

$$K(\boldsymbol{x},\boldsymbol{y}) = -(1/4\pi)[(\boldsymbol{x}-\boldsymbol{y})/|\boldsymbol{x}-\boldsymbol{y}|^3] \tag{7}$$

is the Biot–Savart kernel and the domain V contains all of the vorticity in the flow.

The VTM is a finite volume CFD tool that solves Eqs. (4) and (5) numerically to simulate the evolution of rotor wakes.[2] The model has been used to investigate a variety of rotorcraft problems, including the generation of linearized models for flight mechanics,[3] the onset of the vortex ring state,[4] and the sensitivity of rotorcraft to encounters with aircraft wakes.[5]

In the VTM, the rotor and all vorticity contained within the domain is first enclosed within a structured mesh of computational cells. Equation (5) is inverted at the beginning of each computational time step to yield $\boldsymbol{u}$ at the faces of each of the computational cells. The vorticity distribution in the flow is then advanced to the end of the time step using a numerical discretization of Eq. (4) constructed using Toro's weighted average flux (WAF) scheme[6] to evaluate the intercell vorticity fluxes. The procedure is then repeated for subsequent time steps.

The principal benefit of this approach over primitive variable schemes is that numerical diffusion and dissipation can in no way lead to the destruction of vorticity. Numerical diffusion will, however, lead to the spatial smearing of vorticity, thus, reducing the compactness of vortical structures. These effects can be controlled by introducing an appropriate flux limiter into the WAF scheme,[2] and this allows the spatial compactness of vortical structures in the flow to be maintained over very long computational times.

Because the vorticity conserving properties of the vorticity transport approach are effectively independent of grid resolution, the VTM has the flexibility to run at the resolution that is relevant to the problem at hand. Situations requiring only the prediction of gross flow features, such as flight mechanics simulations, can be modeled successfully using surprisingly low grid resolutions.[2] Conversely, in cases where small-scale flow features, such as blade–vortex interactions, become significant, the code can be run at higher resolution to expose the detail of these structures.

In the present version of the VTM, the aerodynamic loading on each blade is determined using a discrete implementation of unsteady lifting-line theory at a series of collocation points along the length of the blade. The local flow velocity $\boldsymbol{u}_b$ at each collocation point is calculated as the sum of the wake-induced velocity, the freestream velocity, and the structural motion of the blade. Setting the velocity component normal to each collocation point equal to zero yields a set of algebraic equations that can be solved for the strength of the bound vorticity ω_b on each panel. The structure of the model does not preclude replacement of this model with a more sophisticated treatment of the blade aerodynamics.

The dynamics of the rotor blades are modeled by numerical reconstruction of the nonlinear Lagrangian of the system so that the coupled flap–lag–feather dynamics of a set of rigid blades are fully represented. Fully flexible blades can be handled using the same formalism, but this capability has yet to be integrated into the code.

The dynamics of the rotor system are coupled to the blade aerodynamics by writing the vorticity source S in Eq. (4) in terms of the shed and trailed vorticity from the rotor blades by using the relationship

$$S = -\frac{\mathrm{d}}{\mathrm{d}t}\omega_b + \boldsymbol{u}_b\nabla\cdot\omega_b \tag{8}$$

Grid Formulation

In previous versions of the VTM, the computational domain consisted of a box enclosing the helicopter and a sufficiently large part of its surroundings, which was then populated throughout with equally sized cubic grid cells. When using this approach, the rotor wake occupied only a small portion of the total grid. Typically just 20–30% of the cells contained vorticity, whereas the remaining cells served only to fill the computational domain so that the boundary conditions required by the velocity calculation could be applied at its edges, or to provide space for the possible future evolution of the vorticity distribution.

The latest version of the VTM removes this obvious inefficiency by replacing the velocity calculation with a method that does not require direct application of boundary conditions. This provides the freedom to introduce a much more efficient grid formulation. If the evolution of the flow is modeled by using Eq. (4), only regions of the flow that actually contain vorticity need to be contained within the computational domain. In the new code, the number of computational cells, and, hence, the memory required by the calculation, is minimized by using a grid that contains a variable number of cells

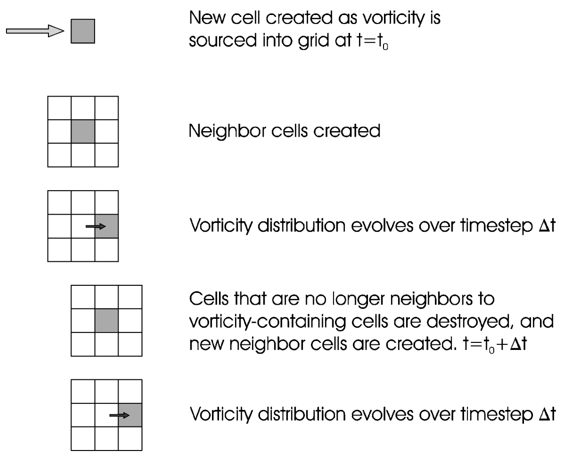
New cell created as vorticity is sourced into grid at t=t₀
Neighbor cells created
Vorticity distribution evolves over timestep Δt
Cells that are no longer neighbors to vorticity-containing cells are destroyed, and new neighbor cells are created. t=t₀+Δt
Vorticity distribution evolves over timestep Δt

Fig. 1 Adaptive grid generation and destruction tracks regions of vorticity within the flow.

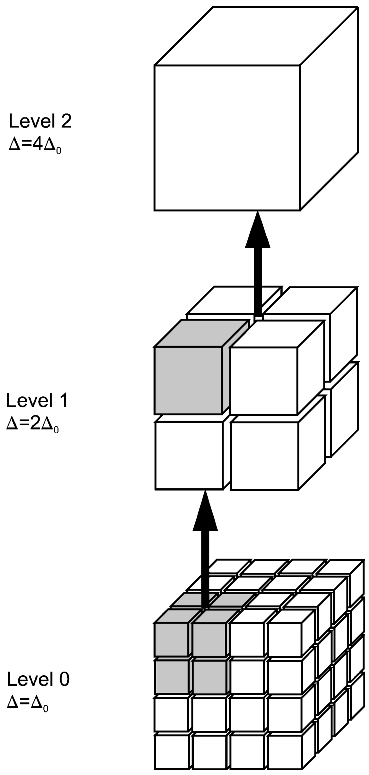

Fig. 2 Cell locations organized into octree data structure.

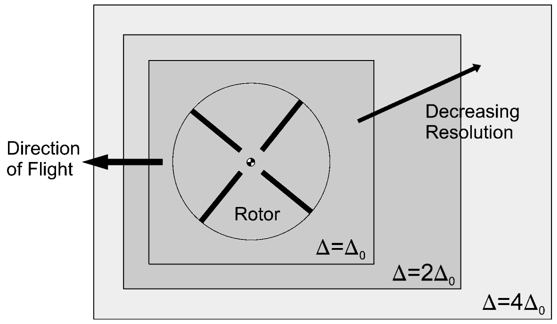

Fig. 3 Rotor is embedded within a nested grid structure that decreases in resolution with distance from the hub.

that are adaptively generated and destroyed to follow the vorticity-containing regions of the flow.

Underlying the grid is a virtual Cartesian stencil or framework, upon which cells can be created and destroyed. Each location on the stencil is a cube, with edge length Δ_0, that can be occupied by a single computational cell. The stencil is fixed in the frame of reference of the rotor hub and extends to infinity in all directions to encapsulate the entire space surrounding the rotor. The stencil, thus, provides an infinite number of discrete locations that can be occupied by cells at any given time.

As vorticity is sourced into the grid from the rotor, cells are created at the appropriate locations on the stencil. To allow the vorticity to advect through the grid, the immediate neighbors of any newly created cells must also be created, as shown in Fig. 1.

Once the velocity is known at every cell interface, this grid structure is sufficient to allow the VTM to evolve the flow according to Eq. (4). At successive time steps, additional neighbor cells are generated around all vorticity-containing cells to allow the new vorticity distribution to evolve. Simultaneously, any cells that no longer contain vorticity, or that are not neighbors of vorticity-containing cells, are destroyed. This process of cell generation and destruction is repeated at every time step to allow the flow to evolve freely while minimizing cell count.

The creation and destruction of cells is managed using an octree data structure. At the lowest level of the octree lie the cells themselves. The next level of the octree is formed by collecting together the cells in eight neighboring cell locations to form a parent cluster, as shown in Fig. 2. This process of clustering is repeated on each level of the octree until a level is reached on which there exists one single cluster that effectively contains the entire computational domain.

Use of an octree system makes the task of managing the constantly evolving grid structure particularly efficient. Not only does the octree system allow efficient, standard algorithms[7] to be used to locate, generate, and destroy elements on the tree, but it also provides the essential infrastructure that is required to evaluate the velocity field using the fast multipole method described later in this paper.

Because the grid stencil has no boundaries, the resulting cell structure is free to expand in space as the wake structure expands. Consequently, the growth rate of the grid, in terms of cell count, must be controlled so that the available computer memory is not quickly exhausted. This problem is solved in the VTM by introducing a number of nested grid levels, of varying resolution, to decrease systematically the resolution of the grid on moving away from the rotor into the far wake. The octree data structure presents a particularly convenient means with which to achieve the required changes in grid resolution. First, the rotor is positioned within the lowest grid level, in which the vorticity-containing cells have side length Δ_0 equal to that of the underlying stencil. At some distance from the rotor, the resolution of the grid can be reduced by simply reducing the depth of the octree so that the vorticity-containing cells now lie on level 1 and have side length $\Delta_1 = 2\Delta_0$. This process can then be repeated to generate any number of grid levels. A schematic representation of a computational domain with such a nested grid structure is shown in Fig. 3. The effectively exponential rate of growth of the computational domain that is made possible using this scheme allows the total number of cells to remain within manageable limits while still precluding the vorticity from ever encountering the boundary of the computational grid during the course of a calculation.

A powerful feature of the nested grid structure relates to the maximum stable rate at which the computation can be advanced on each level of the hierarchy of grids. For explicit methods, such as the WAF algorithm used to advance Eq. (4) through time, the maximum stable time step is set by the Courant–Friedrichs–Lewy (CFL) condition, which is based on the local velocity and the local resolution of the grid. Because the side length of the cells doubles on moving from one nested grid level to the next, the flow on grid level i can usually be evolved using a time step very close to 2^i times the time step used for the finest grid level. Computational effort is, thus, focused, both spatially and temporally, where it is most required, that is, on the flow in the most highly resolved regions of the computational domain closest to the rotor.

Successful implementation of grid nesting requires an effective procedure for advecting vorticity from the computational cells on one level of the hierarchy to the cells on the next highest (or next lowest) level. This is done by overlapping the cells at the interface

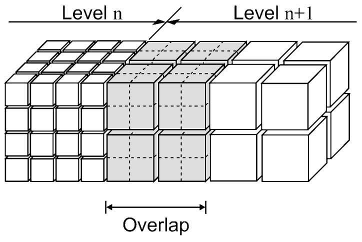

Fig. 4 Overlapping boundaries of grid levels provides both second-order accuracy and monotonicity.

between adjacent levels of the hierarchy, as shown in Fig. 4. The vorticity flux across the interfaces of the cells in the overlap region, calculated on the finer grid, is then extrapolated onto the coarser grid at appropriate points during the course of the calculation. In this way it is possible to maintain conservation of vorticity and to preserve the second-order and monotonic properties of the WAF scheme at all cell interfaces.

Velocity Calculation

Before the WAF algorithm can be used to calculate the vorticity flux between each vorticity-containing cell and its neighbors, the flow velocity on the faces of the cells must be calculated. In the new version of the VTM, the velocity at any point within the domain is calculated by approximating the Biot–Savart integral, Eq. (6), as

$$\boldsymbol{u}(\boldsymbol{x}) \approx \int_V K_\delta(\boldsymbol{x},\boldsymbol{y}) \times \omega(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \tag{9}$$

where the regularized Rosenhead-Moore kernel

$$K_\delta(\boldsymbol{x},\boldsymbol{y}) = -\frac{1}{4\pi}\frac{(\boldsymbol{x}-\boldsymbol{y})}{(|\boldsymbol{x}-\boldsymbol{y}|^2+\delta^2)^{\frac{3}{2}}} \tag{10}$$

is used in place of the Biot–Savart kernel to account for the fact that the cells contain a uniform distribution of vorticity, rather than a vortex singularity. The value of δ is chosen such that the maximum velocity induced by the vorticity within a cell is located on the faces of the cell, as shown in Fig. 5.

If the computational domain consists of N vorticity-containing cells, the task of evaluating Eq. (9) on each cell yields a classical N-body problem. The simplest method, yet also the most naïve, involves the brute-force summation of every discrete Biot–Savart interaction present in the domain and results in an unnecessarily large $\mathcal{O}(N^2)$ calculation. For any realistic simulation, such an approach would quickly become prohibitive as the number of computational cells was increased.

An alternative to direct summation is to solve Eq. (5) directly in differential form. In previous versions of the VTM, the method of cyclic reduction[8] has been used to solve Eq. (5), thus reducing the computational cost to $\mathcal{O}(N \log N)$. Appropriate conditions must first be applied on the boundary of the computational domain if Eq. (5) is to be solved using this approach. Obviously, if the numerical boundary conditions do not represent accurately the flow physics on the outside of the domain, contamination of the solution is possible. Given the complexity of the vortical structures in the rotor wake, the task of deriving these boundary conditions can become extremely difficult, particularly at outflow boundaries where vorticity will eventually be transferred into the unresolved space surrounding the computational domain.

To address this problem in the new version of the VTM, the method of cyclic reduction has been replaced with a method for evaluating Eq. (9) using a fast summation method called the Cartesian fast multipole method. Fast multipole algorithms were first devised by Greengard and Rokhlin[9] to evaluate gravitational and electrostatic potential fields. The method can be adapted, though, to model any system where the pairwise interactions between the elements of the system can be represented by a suitable kernel function. The Cartesian fast multipole method (FMM) is a specific implementation of the fast multipole algorithm that exploits the underlying rectangular topology of the computational grid to expand the integral in Eq. (9) using algebraic series rather than the trigonometric expansions of the more general formulation of the algorithm. The

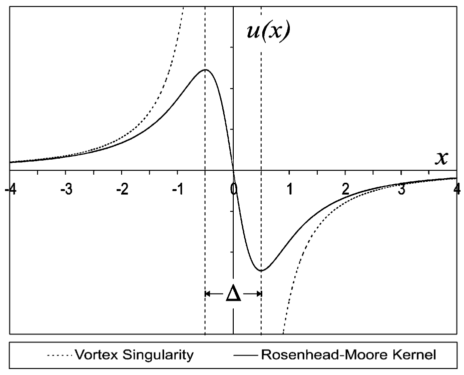

Fig. 5 Velocity field generated by - - - -, Biot–Savart kernel (vortex singularity) and ——, Rosenhead–Moore kernel (approximate representation of cell containing uniform distribution of vorticity).

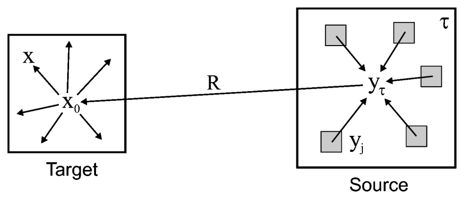

Fig. 6 Vorticity contained in source-cluster τ generates a velocity field throughout target cluster, which is then expanded as a Taylor series centered about x_0.

cost of the velocity calculation is reduced by exploiting the rapid decrease in magnitude of the Biot–Savart interaction with distance to group multiple, long-range interactions together so that they can be expressed using a single, converging algebraic series. Truncating this series yields an approximate method, the accuracy of which can be tailored by controlling the degree of truncation. When this approach is used, the cost of the velocity calculation can be reduced to just $\mathcal{O}(N)$, compared to the $\mathcal{O}(N^2)$ or $\mathcal{O}(N \log N)$ cost of other commonly used methods.

Cartesian FMM

In this section, some of the basic results required to implement the Cartesian FMM are summarized. Figure 6 shows a cluster τ enclosing N_τ of the total of N vorticity-containing cells that exist within the computational domain. The vorticity-weighted center (centroid) of the cluster lies at $\boldsymbol{y}_\tau$, and the center of each vorticity-containing cell lies at $\boldsymbol{y}_j$. The velocity is to be evaluated on a second cluster τ', well separated from τ, which has its centroid at $\boldsymbol{x}_0$.

The velocity induced at point $\boldsymbol{x}_0$ by the vorticity within cluster τ can be evaluated by approximating Eq. (9), using Euler integration, as

$$\boldsymbol{u}(\boldsymbol{x}_0) = \sum_{j\in\tau} \boldsymbol{K}_\delta(\boldsymbol{x}_0,\boldsymbol{y}_j) \times \omega_j \tag{11}$$

Summing over all clusters and using this expression directly to evaluate the velocity on all cells within the computational domain (for instance, by constructing τ' to consist of a single cell) results in a computation that is $\mathcal{O}(N^2)$. The cost of the computation can be reduced to $\mathcal{O}(N \log N)$ by expanding Eq. (11) as a Taylor series about the center of the source cluster τ then approximating the result of the summation over the cluster by truncating the series after the $\boldsymbol{p}$th term:

$$\begin{aligned}\boldsymbol{u}(\boldsymbol{x}_0) &= \sum_{j\in\tau} K_\delta(\boldsymbol{x}_0, \boldsymbol{y}_\tau + (\boldsymbol{y}_i - \boldsymbol{y}_\tau)) \times \omega_j \\ &= \sum_{j\in\tau}\sum_{k}^{\infty} \frac{1}{\boldsymbol{k}!} D_y^k \boldsymbol{K}_\delta(\boldsymbol{x}_0,\boldsymbol{y}_\tau)(\boldsymbol{y}_j-\boldsymbol{y}_\tau)^k \times \omega_j \\ &\approx \sum_{k=0}^{p} \boldsymbol{a}_k(\boldsymbol{x}_0,\boldsymbol{y}_\tau) \times \boldsymbol{m}_k(\tau)\end{aligned} \tag{12}$$

where $\boldsymbol{k} = (k_1, k_2, k_3)$, $D_y^k = \partial/\partial y_1^{k_1} \partial y_2^{k_2} \partial y_3^{k_3}$, $\boldsymbol{k}! = k_1!k_2!k_3!$, $\boldsymbol{x}^k = x_1^{k_1} x_2^{k_2} x_3^{k_3}$ for $k_i \geq 0$, and subscripts 1–3 refer to the Cartesian directions. The multipole tensors

$$\boldsymbol{a}_k(\boldsymbol{x}_0, \boldsymbol{y}_\tau) = (1/\boldsymbol{k}!) D_y^k \boldsymbol{K}_\delta(\boldsymbol{x}_0, \boldsymbol{y}_\tau) \tag{13}$$

are functions of the range of the interaction alone, whereas the moments

$$\boldsymbol{m}_k(\tau) = \sum_{j \in \tau} (\boldsymbol{y}_j - \boldsymbol{y}_\tau)^k \omega_j \tag{14}$$

describe the local distribution of vorticity within the cluster τ.

The $\mathcal{O}(N)$ cost of the FMM calculation is achieved by calculating the velocity in all of the cells within the target cluster τ' using a truncated Taylor expansion of the velocity about the centroid of the cluster. This expansion requires the spatial derivatives of $\boldsymbol{u}$ at $\boldsymbol{x}_0$. Differentiating Eq. (12) with respect to $\boldsymbol{x}_0$ gives

$$\boldsymbol{u}^n(\boldsymbol{x}_0) = (-1)^n \sum_{k=n}^{p} \frac{\boldsymbol{k}!}{(\boldsymbol{k}-\boldsymbol{n})!} \boldsymbol{a}_k(\boldsymbol{x}_0, \boldsymbol{y}_\tau) \times \boldsymbol{m}_{k-n}(\tau) \tag{15}$$

where $\boldsymbol{u}^n(\boldsymbol{x}) = D_x^n \boldsymbol{u}(\boldsymbol{x})$. Note, too, that the velocity derivatives calculated at $\boldsymbol{x}_0$ can be shifted to some other nearby point $\boldsymbol{x}$ by expanding Eq. (15) as a Taylor series about $\boldsymbol{x}_0$ so that

$$\boldsymbol{u}^n(\boldsymbol{x}) = \sum_{k=n}^{p} \frac{(\boldsymbol{x}-\boldsymbol{x}_0)^{k-n}}{(\boldsymbol{k}-\boldsymbol{n})!} \boldsymbol{u}^n(\boldsymbol{x}_0) \tag{16}$$

It is clear from the preceding paragraphs that the speed advantage of the FMM over direct summation is obtained at the disadvantage of an approximate, rather than an exact, evaluation of the velocity field. An important issue in the use of the Cartesian FMM is, thus, the control of the error introduced into the calculation by truncation of the series expansion of the velocity.

Formally, a measure of the local error, ε_p, introduced into the calculation by truncation of the expansion on cluster τ to order $\boldsymbol{p}$ is given by the sum of the omitted terms in Eq. (12):

$$\varepsilon_p = \left| \sum_{k=p+1}^{\infty} \boldsymbol{a}_k(\boldsymbol{x}, \boldsymbol{y}_\tau) \times \boldsymbol{m}_k(\tau) \right| \tag{17}$$

where $|\boldsymbol{x}| = \sqrt{(x_1^2 + x_2^2 + x_3^2)}$. This result can be simplified by using the mean-value theorem to yield

$$\varepsilon_p = \left| \boldsymbol{a}_{k+1}(\boldsymbol{x}, \boldsymbol{y}^*) \times \boldsymbol{m}_{k+1}(\tau) \right| \tag{18}$$

where $\boldsymbol{y}^*$ is some point located in a circle, center $\boldsymbol{y}_\tau$ and with radius $d_\tau = \max |\boldsymbol{y}_j - \boldsymbol{y}_\tau|$, $j \in [1, \ldots, N_\tau]$. This measure of the error is clearly not convenient to use in practice because its evaluation requires the calculation of additional terms of the Taylor series. Instead, a heuristic estimate of the error, based on the analysis of Lindsay and Krasny,[10]

$$\varepsilon_p \approx \max \left\{ \frac{(p_i+1)^2 d_\tau^{\|p\|} |\boldsymbol{m}_0(\tau)|}{4\pi R^{(p_i+2)}}, i = 1, 2, 3 \right\} \tag{19}$$

is used, where $R^2 = |\boldsymbol{x} - \boldsymbol{y}_\tau|^2 + \delta^2$ and $\|\boldsymbol{p}\| = p_1 + p_2 + p_3$. After interpreting ε_p as a prespecified maximum allowable local error in the velocity field, this expression can be solved to determine the appropriate value of $\boldsymbol{p}$ at which to truncate the expansion when evaluating each cluster interaction.

Tensor Calculation

The tensors $\boldsymbol{a}_k$ defined in Eq. (13) can be evaluated efficiently using recursion. The following result relies on the fact that the gradient of the Rosenhead–Moore kernel is the regularized Newtonian potential:

$$\phi_\delta(\boldsymbol{x}, \boldsymbol{y}) = 1/[4\pi(|\boldsymbol{x} - \boldsymbol{y}|^2 + \delta^2)^{\frac{1}{2}}] \tag{20}$$

If the scaled derivatives of the Newtonian potential

$$b_k(\boldsymbol{x}, \boldsymbol{y}_\tau) = (1/\boldsymbol{k}!) D_y^k \phi_\delta(\boldsymbol{x}, \boldsymbol{y}_\tau) \tag{21}$$

are defined such that $b_0 = \phi_\delta(\boldsymbol{x}, \boldsymbol{y})$ and $b_k = 0$ for $k_i < 0$, then successive values of b_k are related by

$$\|\boldsymbol{k}\| R^2 b_k - (2\|\boldsymbol{k}\| - 1) \sum_{i=1}^{3} (x_i - y_i) b_{k-e_i} + (\|\boldsymbol{k}\| - 1) \sum_{i=1}^{3} b_{k-2e_i} = 0 \tag{22}$$

where $R^2 = |\boldsymbol{x} - \boldsymbol{y}|^2 + \delta^2$, $\|\boldsymbol{k}\| = k_1 + k_2 + k_3$, and $\boldsymbol{e}_i$ is the ith Cartesian basis vector. Once all b_k are known, the tensors $\boldsymbol{a}_k$ can be reconstructed as

$$\boldsymbol{a}_k(\boldsymbol{x}, \boldsymbol{y}_c) = -\sum_{i=1}^{3} (k_i + 1) b_{k+e_i} \boldsymbol{e}_i \tag{23}$$

This result is derived in full in Ref. 10.

Implementation

Key to the implementation of the FMM is the hierarchal decomposition of the computational domain into clusters of cells. A convenient hierarchical structure already exists in the VTM in the form of the octree used to generate and manage the adaptive grid system, and this same octree data structure is used in the implementation of the FMM.

The first stage in applying the FMM to evaluate Eq. (9) involves an upward sweep through the octree, calculating the moments of each cluster according to Eq. (14). This stage can be accelerated by expressing the moments of a cluster as a binomial sum of the moments of its children:

$$\boldsymbol{m}_k(\tau) = \sum_{\bar{\tau} \in \tau} \sum_{n=0}^{k} \binom{k}{n} (\boldsymbol{y}_{\bar{\tau}} - \boldsymbol{y}_\tau)^k \boldsymbol{m}_{k-n}(\bar{\tau}) \tag{24}$$

where τ and $\bar{\tau}$ refer to the parent and child clusters, respectively.

The second stage in applying the FMM involves a sweep back down the tree. This sweep is responsible for generating and refining the velocity field on each level of the tree. Consider a cell i located on level n of the octree. The velocity field within cell i can be considered as the sum of a far-field component, a middle-field component, and a near-field component. The FMM defines these regions spatially as shown in Fig. 7. For clarity, the two-dimensional interaction sets have been shown, but the diagram extends easily to three dimensions. Note that these interaction sets are geometrically similar on each level of the grid. This feature can be exploited to refine the velocity in recursive fashion on descending the levels of the octree as follows.

On each level n, the far-field component of the velocity at cell i is inherited from cell i's parent cluster, located on level $n+1$ of the octree. The middle-field component is evaluated by applying Eq. (15) to each of the clusters, located on level n, within cell i's middle field. The near-field component is not evaluated at this point because it will be accounted for when the process is repeated at the lower levels of the octree. The far-field and middle-field components are summed together and then translated, using Eq. (16), to cell i's children on level $n-1$ of the octree, except if cell i is on the lowest level of the octree.

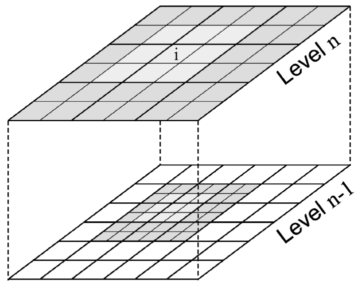

Fig. 7 Evaluation of velocity field on cell *i* split into near-field (light gray), middle-field (dark gray), and far-field contributions (white).

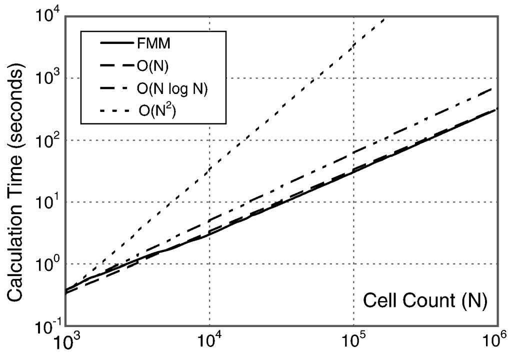

Fig. 8 Computational cost of using the FMM to calculate the velocity on N vorticity-containing cells (by using a Pentium4, 2-GHz processor with 1 Gb of memory).

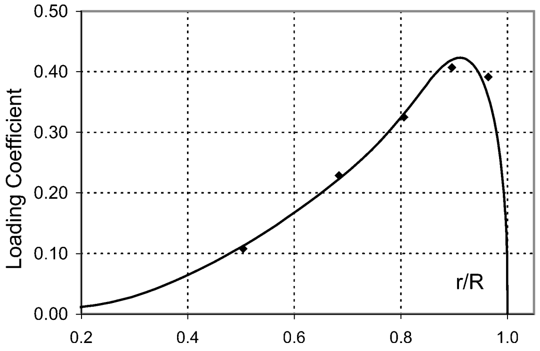

Fig. 9 Comparison between experimental and VTM-calculated blade loading for Caradonna and Tung's rotor, $C_T = 0.0079$ and $\Omega R = 150\ \mathrm{ms}^{-1}$: ◆, experimental data and ——, VTM calculation.

This process of evaluation and inheritance is performed on all clusters on a given level of the tree before descending to the next level and repeating the entire process. Once the downward sweep through the tree is complete, the velocity on any given cell is given by

$$\boldsymbol{u}(\boldsymbol{x}) = \sum_{k=0}^{p} \frac{(\boldsymbol{x} - \boldsymbol{x}_p)^k}{\boldsymbol{k}!} \boldsymbol{u}^k(\boldsymbol{x}_p) \tag{25}$$

where $\boldsymbol{x}_p$ is the center of the cell's parent cluster. On the lowest level of the octree, the calculation is completed by adding the velocity contribution

$$\boldsymbol{u}(\boldsymbol{x}) = \sum_j \boldsymbol{K}_\delta(\boldsymbol{x}, \boldsymbol{y}_j) \times \omega_j \tag{26}$$

from the cell's near field. This contribution is calculated directly as the sum of Biot–Savart interactions between the cell and all vorticity-containing cells within the cell's near field.

Code Performance

The CPU time required to evaluate the velocity on all of the cell interfaces within a computational domain consisting of N vorticity-containing cells is shown in Fig. 8. Figure 8 clearly demonstrates that the FMM quickly converges on its $\mathcal{O}(N)$ theoretical cost as the number of cells contained in the computational domain is increased.

Code Validation

Wake Geometry in Hover

The results of a validation of the VTM against the well-known experimental data for a hovering, two-bladed model rotor produced by Caradonna and Tung[11] are presented in Figs. 9 and 10. Numerical results were generated using 32 blade collocation points in a cosine distribution along each blade and by resolving the rotor radius across 40 computational cells at the finest grid level.

Figure 9 shows a comparison of the calculated loading distribution along the rotor blade with the experimental data. Although the experimental data are relatively sparse, and the associated error does not appear to have been quantified in the original reference, the loading predicted by the VTM is in close agreement with experiment, both in terms of its magnitude and in terms of the general shape of the spanwise loading distribution.

The calculated wake structure, in terms of the location of the tip vortices in the flow downstream of the rotor, is compared with Caradonna and Tung's experimental data in Fig. 10. The bars attached to the numerical data represent the estimated error associated with extracting the exact vortex core locations from the discrete, cellwise-constant representation of the vorticity that is used in the VTM. The calculated vortex positions compare well with the experimental data, particularly over the first 270 deg of vortex age. Beyond this point, the unsteadiness in the far wake of the hovering rotor, caused by the development of a vortex pairing instability between adjacent vortex filaments, manifests itself as an increase in

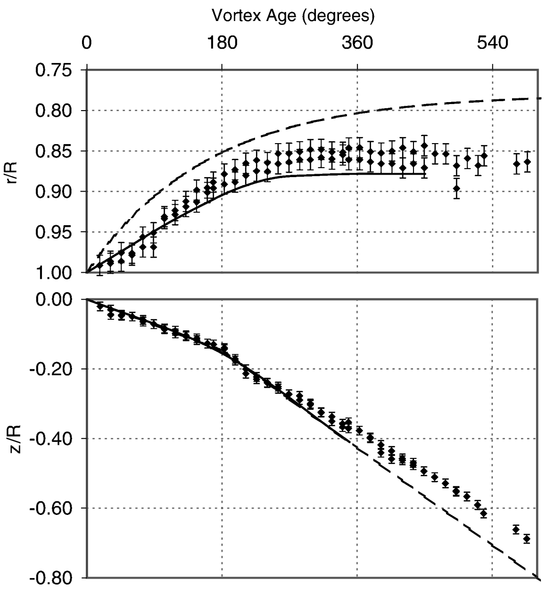

Fig. 10 Comparison of VTM-calculated wake geometry vs Caradonna and Tung's experimental data and Kocurek and Tangler's empirical model, $C_T = 0.0079$ and $\Omega R = 150\ \mathrm{ms}^{-1}$: ◆, VTM; ——, Caradonna and Tung; and – – –, Kocurek and Tangler.

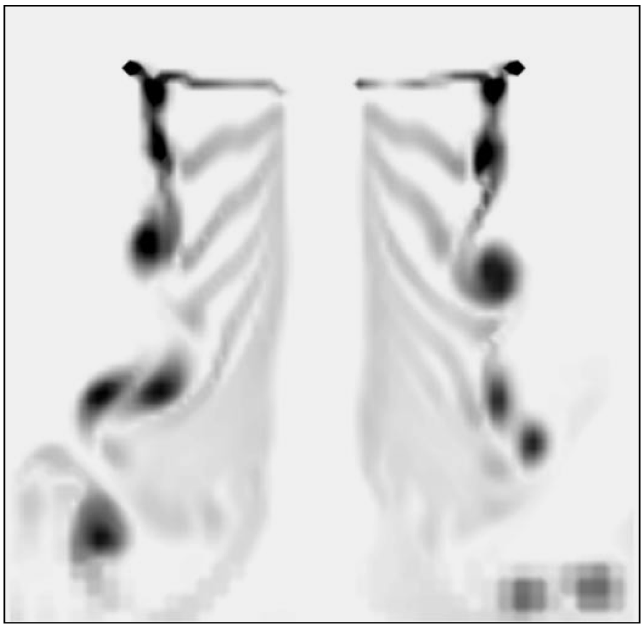

Fig. 11 Calculated wake geometry for Caradonna and Tung's rotor showing development of vortex pairing instability in far wake.

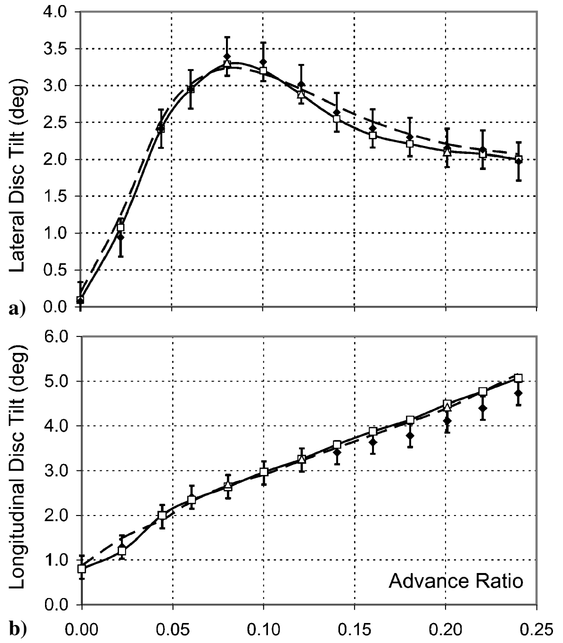

Fig. 12 New and old versions of VTM vs Harris's experimental data: a) lateral disk tilt and b) longitudinal disk tilt: ◆, experimental data; – – –, old VTM; □, new VTM $\Delta x = 0.04R$; and Δ, new VTM $\Delta x = 0.016R$.

scatter of the data. Interestingly, this increase in scatter can also be seen in the experimental data presented in the original paper.[11]

To obtain a better quantitative measure of the accuracy of the VTM's predictions of the wake geometry, Fig. 10 also includes a comparison against Kocurek and Tangler's widely-used empirical wake trajectory.[12] Interestingly, although the predictions of the VTM lie between the two published sets of results, the axial vortex locations produced by both experiment and the VTM are consistent with an overprediction of the contraction rate of the wake by Kocurek and Tangler's model.

Figure 11 shows an example of the calculated wake structure produced during the VTM's simulation of Caradonna and Tung's rotor, visualized here by plotting contours of vorticity magnitude on a vertical plane containing the axis of the rotor shaft. The development of the vortex pairing instability in the far wake of the rotor, mentioned earlier in connection with the scatter in the observed tip vortex locations, can clearly be seen in Fig. 11.

Rotor Performance in Forward Flight

To illustrate the ability of the VTM to predict rotor performance in steady forward flight, a validation against the wind-tunnel data produced by Harris[13] is presented in Fig. 12. Harris measured the flapping behavior of an isolated, four-bladed model rotor over a range of flight speeds. In all of the tests, the rotor was trimmed to a preset thrust coefficient, the cyclic control angles were held fixed, and the free response of the rotor in flap was measured. Figure 12 shows a comparison of the calculated disk tilts produced using both the new and old versions of the VTM code with Harris's experimental data. The error bars represent Harris's own estimate of the accuracy of his measurements. All numerical results were produced using 32 blade collocation points in a cosine distribution along each blade. Two sets of results for the new VTM are presented, one obtained using a fairly coarse grid that resolved the rotor radius across 25 computational cells at the finest grid level and another obtained after refining the calculation so that the rotor radius was resolved instead across 62 computational cells. When the new version of the VTM was used, four levels of grid expansion were used, with transitions between grid levels positioned at $2R$, $6R$, and $10R$ from the rotor hub.

This case has proven to be a significant challenge to computational methods in the past, but both versions of the VTM show good agreement with Harris's experimental data for both the lateral

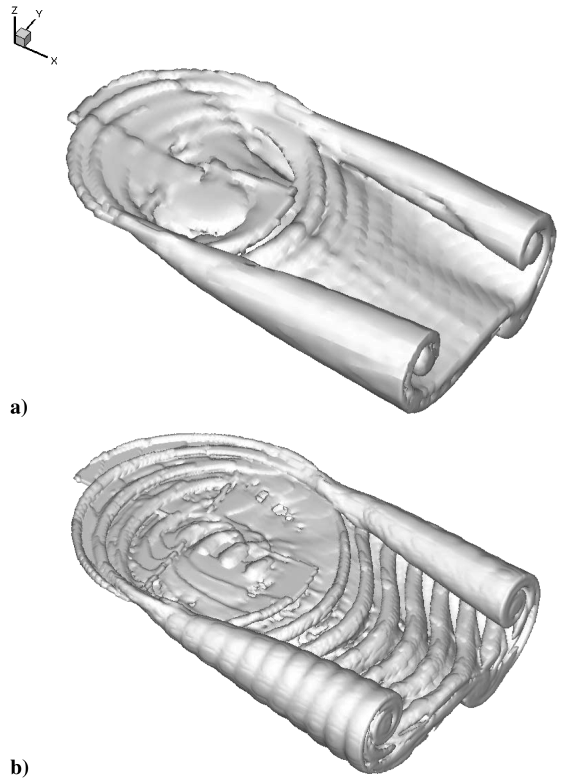

Fig. 13 Predicted wake structure for Harris's rotor at advance ratio $\mu = 0.12$: a) calculation with rotor radius resolved across 25 cells, $\Delta x = 0.04R$, and b) refined calculation with rotor radius resolved across 62 cells, $\Delta x = 0.016R$.

and the longitudinal tilt of the rotor. The new version of the VTM produces somewhat better correlation than the old at low forward speeds, and this is consistent with previous experience that the old version of the VTM was more likely to be contaminated by inappropriately chosen boundary conditions at low to transitional advance ratios than at high. The predicted values of the lateral disk tilt, although marginally lower than the experimental values, lie within the experimental error bounds over the full range of flight speeds. The consistent behavior between the two versions of the code in slightly overpredicting the longitudinal tilt at high forward speed, $\mu > 0.20$, is indicative of a deficiency in the present blade aerodynamic model at high advance ratio, most likely in its treatment of the poststall lift variation in the reverse-flow region on the retreating side of the rotor.

Figure 13 shows the effect of grid resolution on the predicted geometry of the rotor wake. Although the results presented are for $\mu = 0.12$, Fig. 13 is representative of the performance of the code over the full range of flight speeds shown in Fig. 12. The wake generated on the coarse grid (Fig. 13a) shows some distinctly under-resolved flow structures compared to the more refined calculation (Fig. 13b). Figure 14 shows the resultant effect on the predicted rotor loading. In Fig. 14, the loading on a single blade of Harris's rotor is shown as a function of blade azimuth ($\Psi = 0$ deg being at the trailing edge of the rotor disk). The calculation on the coarser grid does not resolve the sharp peaks in loading, induced by localized blade–vortex interactions, that the refined calculation shows the blade to experience as it traverses across the forward half of the rotor. Thus, although the coarser grid is demonstrably adequate to resolve the low-frequency, first harmonic flapping dynamics of the rotor, it would arguably not be fine enough to predict the higher frequency dynamics of the system (associated, for instance, with vibration or acoustic signature) that require accurate resolution of the fine-scale features of the rotor loading.

Nevertheless, the evident insensitivity to grid resolution of the predicted low-frequency rotor dynamics, revealed by Fig. 12, appears to be a characteristic of the VTM that is endowed by the

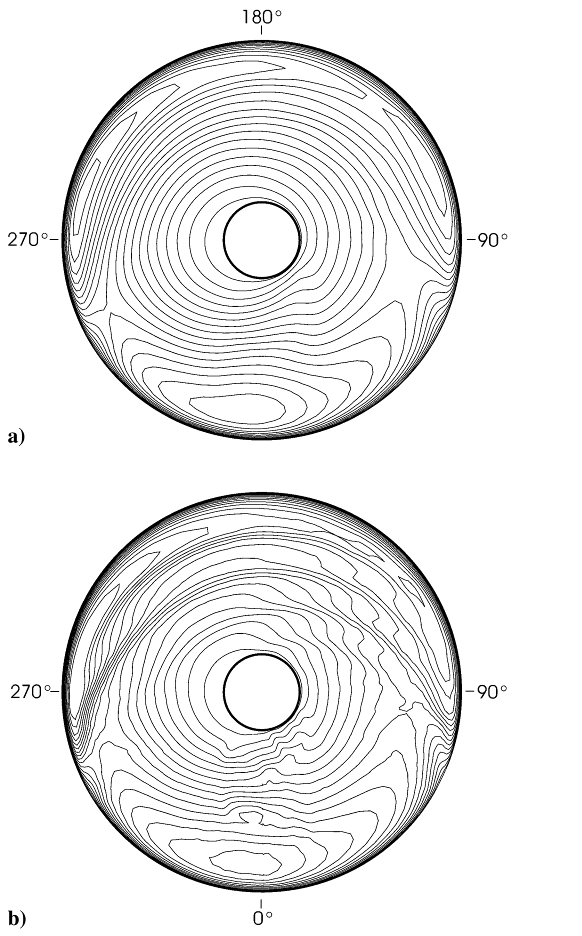

Fig. 14 Blade loading as function of azimuth for Harris's rotor at advance ratio $\mu = 0.12$; same contour interval: a) calculation with the rotor radius resolved across 25 cells, $\Delta x = 0.04R$, and b) refined calculation with the rotor radius resolved across 62 cells, $\Delta x = 0.016R$.

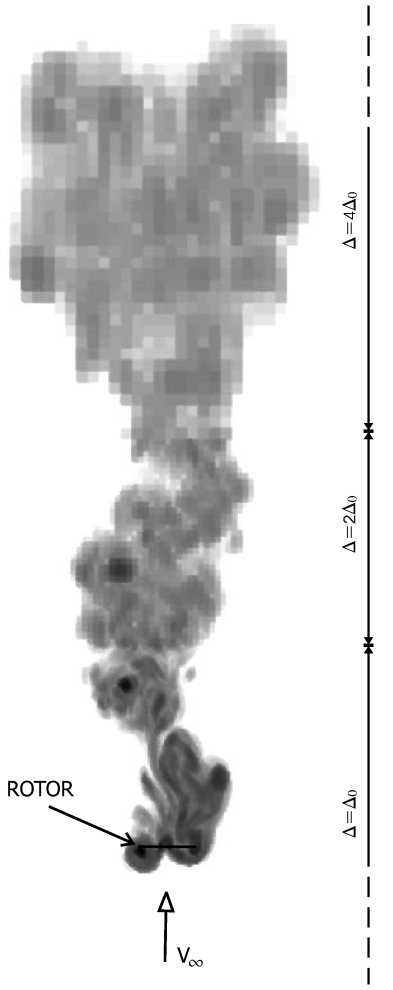

Fig. 15 Visualization of wake of two-bladed teetering rotor in vortex ring conditions.

vorticity-conserving formulation of the method. If approached with care, this feature of the VTM can be used in certain applications to match the resolution of the calculations to the requirements of the simulation, often resulting in a considerable saving in computational effort.[3,5]

Example Application

Figure 15 shows the calculated wake geometry generated by a two-bladed teetering rotor when at a descent rate sufficient to precipitate the onset of the vortex ring state in the flow surrounding the rotor. The flow is visualized by plotting contours of vorticity magnitude on a vertical plane containing the shaft axis. The rotor can be seen at the bottom of the image, with a highly disordered and unsteady wake streaming out above it. In this calculation, the helicopter rotor was resolved using 10 grid cells across the rotor radius. This level of resolution is quite adequate to capture the gross effects of the vortex ring state within the system. In the vortex ring state, the relevant timescales of the flow are on the order of tens of rotor revolutions, and during a typical simulation, a highly extended and substantial wake evolves downstream of the rotor. The use of multiple grid levels, as shown in Fig. 15, has allowed the entire length of the wake to be captured within the simulation. In the example shown, the wake contained within the computational domain has a length of over 25 rotor radii but occupies a total of only 156,000 cells. Note that 64% of the cells used by the simulation were contained within the finest grid level surrounding the rotor, showing how the method is able to concentrate computer resources into the area of most interest near the rotor itself.

This example demonstrates how, as a result of the boundary-free grid formulation, the entire wake history of the rotor can be captured within the computational domain. As the wake convects downstream, the grid resolution used to capture the far wake is gradually reduced, thus requiring progressively fewer cells to capture the vorticity. Nevertheless, the influence of this vorticity on the rotor is still captured to sufficient accuracy by the FMM, allowing the behavior of the rotor in this difficult flight regime to be captured without running the risk of contamination of the simulation by spurious boundary effects.

Modeling Full Helicopter Configurations

In the preceding sections of this paper, attention has been focused largely on the modeling of the aerodynamics of isolated rotors. There are many interesting and practical applications, however, where a more complete model of the rotorcraft is required to yield valid insights into the behavior of the system. In the following sections, two extensions to the VTM that are particularly well adapted to the structure of the method are described in some detail.

Fuselage Interactions

In practice, it is rarely possible to separate the aerodynamic analysis of the helicopter rotor from the influence of the fuselage. First, the fuselage modifies the velocity field surrounding the rotor, leading to subtle, yet important, changes in its aerodynamic loading. Fuselage-induced flow distortion is known, for instance, to have a marked impact on the vibration generated by the rotor.[14] Second, the time-varying nature of the aerodynamic loads induced by the rotor and its wake on the fuselage leads to a variety of challenging problems for the helicopter designer, such as, the proper placement of tail surfaces, auxiliary rotors, engine intakes, and airspeed measuring devices. To analyze these problems successfully, it is usually the case that the aerodynamic interaction between the rotors and fuselage of the helicopter must be captured in fully coupled, time-accurate fashion.

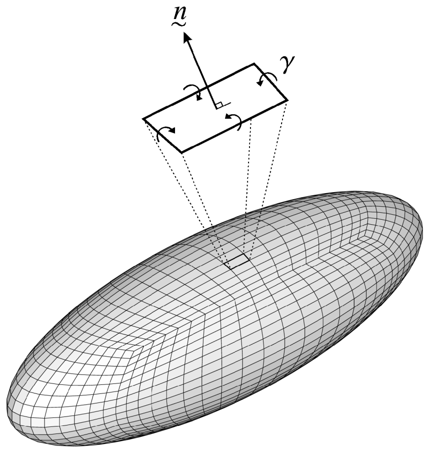

Fig. 16 Vortex panel representation of helicopter fuselage.

The formulation of the VTM in terms of a vorticity distribution in space allows very easy implementation of a time-accurate model for the fuselage aerodynamics using a relatively straightforward extension of classical singularity-panel methods. The surface of the fuselage is first discretized into a set of n rectangular or triangular panels. A closed vortex loop is then placed around the circumference of each panel. If q_i is the normal component of the velocity induced by the rotors, the rotor wake and the freestream, measured at the centroid of panel i, and A_{ij} is the normal component of velocity induced by a vortex loop of unit strength on panel j, also measured at the centroid of panel i, then, by imposing the zero through-flow boundary condition at the centroid of each panel,

$$A_{ij}\gamma_j + q_i = 0 \tag{27}$$

from which the strengths of the vortex loops on the fuselage panels can be calculated at the beginning of each computational timestep. Figure 16 shows a summary of the approach using an ellipsoid as a simple model for the fuselage geometry. The advantage of using a vortex distribution on the surface of the fuselage in conjunction with the VTM is that the velocity field generated by the presence of the fuselage can be calculated in fully unsteady fashion along with the velocity field generated by the remainder of the vorticity in the flow. This is done simply by interpolating the vorticity associated with the fuselage panels into the computational cells at the beginning of each time step.

Note, however, that the model as described accounts only for the nonviscous flow effects generated by the presence of the fuselage within the flow. Extension of the model to include viscous effects and flow separation is possible and will be described in a future publication.

Ground Effect

A fast and efficient method of evaluating the effects of the presence of the ground on helicopter performance is another very useful adjunct to any rotorcraft analysis code, given modern operational requirements to fly at very low altitude and the important effect that the presence of the ground has on defining the limits of rotorcraft performance.[15]

The FMM admits a particularly efficient procedure for modeling the flowfield surrounding a helicopter flying in ground effect. The approach is based on the method of images, in which a plane surface is introduced into the flow so that all of the vorticity in the flow is contained to one side of the surface (conveniently termed the real side of the surface). A modified vorticity distribution is then created in which the real vorticity distribution is augmented by an image vorticity system on the opposing side of the surface such that the modified vorticity system is mirror-symmetric about the surface. The modified vorticity distribution induces a velocity field that has zero perpendicular component on the plane of reflection. Hence, a suitably located reflection plane can be used to represent (in nonviscous fashion, of course) the presence of the ground near the rotor.

The method of images is implemented in the FMM by augmenting Eq. (15) with an extra term to give the contribution to the velocity field from the image vorticity distribution. Thus,

$$\boldsymbol{u}^n(\boldsymbol{x}_0) = (-1)^n \sum_{k=n}^{p} \frac{\boldsymbol{k}!}{(\boldsymbol{k}-\boldsymbol{n})!}[\boldsymbol{a}_k(\boldsymbol{x}_0, \boldsymbol{y}_\tau) \times \boldsymbol{m}_{k-n}(\tau) + \boldsymbol{a}_k(\boldsymbol{x}_0, \boldsymbol{y}'_\tau) \times \boldsymbol{m}'_{k-n}(\tau)] \tag{28}$$

where $\boldsymbol{y}'_\tau$ is the location of the centroid of the reflection of cluster τ in the ground plane. Note that the moments of the image clusters need not be calculated from first principles but can be inferred directly from the moments of the real clusters by using

$$\boldsymbol{m}'_k(\tau) = (-1)^k \boldsymbol{R}\boldsymbol{m}_k(\tau) \tag{29}$$

The symmetry operator $\boldsymbol{R}$ can be constructed by noting that, on reflection in the ground plane, the component of vorticity parallel to the ground plane changes sign, whereas the component normal to the ground plane remains unchanged. Hence, for example, $\boldsymbol{R} = \mathrm{diag}(-1, -1, 1)$ for a coordinate frame in which the ground plane is normal to the x_3 axis. The only significant overhead in this approach, compared to a calculation out of ground effect, results from the need to calculate from first principles the tensors $\boldsymbol{a}_k(\boldsymbol{x}_0, \boldsymbol{y}'_\tau)$ associated with the image clusters.

Example Application

Figure 17 shows an example calculation that integrates all of the features described earlier. A surface on which the vorticity in the flow has constant magnitude has been plotted for a helicopter consisting of a deep, narrow fuselage typical of an attack helicopter configuration, a conventional four-bladed main rotor, and a tractor type tail rotor. The helicopter is maintaining station close to the ground in a tail wind that is blowing across the vehicle from the starboard aft-quarter. This flight case is quite often encountered during the tactical deployment of this type of helicopter. Clearly visible in Fig. 17 is the strong interaction between the wakes of the two rotors, with the wake of the tail rotor being entrained through the main rotor. The blade–vortex interactions generated by this interaction appear in the loading on the main rotor as a series of concentrated loading spikes near the port aft-quarter of the rotor disk and, although not confirmed as yet, could quite feasibly contribute unfavorably to the acoustic signature of the aircraft under this flight condition. Just upwind of the fuselage, a characteristic horseshoe-shaped bow vortex is formed just above the ground at the junction between the outflow from the main rotor and the oncoming wind. Especially near the nose of the helicopter, this vortex interacts in a complex and highly unsteady fashion with the fuselage to form a region of recirculating flow that constantly shifts in position. A prime practical concern in this case was that the unsteadiness in the flow might induce strong enough variation in the loading on the fuselage for the dynamic characteristics of the vehicle to be degraded, thus, compromising the pilot's ability to perform the mission. The aerodynamic effects driving these practical issues are readily captured by the VTM.

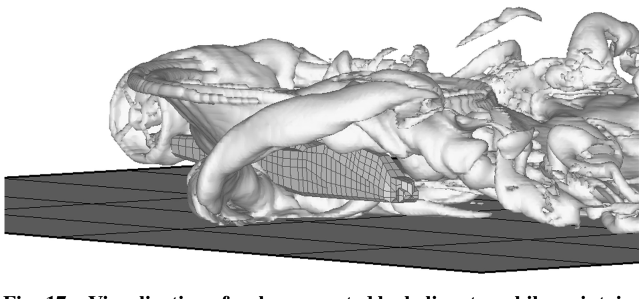

Fig. 17 Visualization of wake generated by helicopter while maintaining station close to ground in wind blowing from starboard aft-quarter.

Conclusions

Standard CFD-based methods, particularly when written in primitive, velocity–pressure form, generally suffer from significant drawbacks when used to analyze rotor flows. This is because they tend to diffuse vorticity and to require extensive computer resources. The original VTM circumvents the vorticity diffusion problem by solving the incompressible Navier–Stokes equations in vorticity-conservation form on a structured, uniform Cartesian computational mesh. A new grid system and fast multipole velocity calculation have now been incorporated into the model. As a result of the adaptive, variable-resolution features of the new grid system, reductions in cell count of typically an order of magnitude, compared to the use of the structured mesh, have been achieved. Use of the Cartesian FMM allows the velocity field throughout a computational domain consisting of N cells to be calculated with $\mathcal{O}(N)$ computational cost compared to the $\mathcal{O}(N \log N)$ cost of the original method used in the VTM. These new components of the VTM result in an extremely efficient use of computer resources and allow computational effort to be focused onto the regions of the rotor flow that need to be most highly resolved.

References

[1]Caradonna, F. X., "Developments and Challenges in Rotorcraft Aerodynamics," AIAA Paper 2000-0109, Jan. 2000.

[2]Brown, R. E., "Rotor Wake Modeling for Flight Dynamic Simulation of Helicopters," *AIAA Journal*, Vol. 38, No. 1, 2000, pp. 57–63.

[3]Brown, R. E., and Houston, S. S., "Comparison of Induced Velocity Models for Helicopter Flight Mechanics," *Journal of Aircraft*, Vol. 37, No. 4, 2000, pp. 623–629.

[4]Brown, R. E., Leishman, J. G., Newman, S. J., and Perry, F. J., "Blade Twist Effects on Rotor Behaviour in the Vortex Ring State," 28th European Rotorcraft Forum, Bristol, England, U.K., Sept. 2002.

[5]Whitehouse, G. R., and Brown, R. E., "Modeling the Mutual Distortions of Interacting Helicopter and Aircraft Wakes," *Journal of Aircraft*, Vol. 40, No. 3, 2003, pp. 440–449.

[6]Toro, E. F., "A Weighted Average Flux Method for Hyperbolic Conservation Laws," *Proceedings of the Royal Society of London, Series A: Mathematical and Physical Sciences*, Vol. 423, No. 1864, 1989, pp. 401–418.

[7]Knuth, D. E., *The Art of Computer Programming—Volume 3: Sorting and Searching*, Addison–Wesley, Reading, MA, 1973, 722 pp.

[8]Schumann, U., and Sweet, R. A., "A Direct Method for the Solution of Poisson's Equation with Neumann Boundary Conditions on a Staggered Grid of Arbitrary Size," *Journal of Computational Physics*, Vol. 20, No. 2, 1976, pp. 171–182.

[9]Greengard, L., and Rokhlin, V., "A Fast Algorithm for Particle Simulations," *Journal of Computational Physics*, Vol. 73, No. 1, 1987, pp. 325–348.

[10]Lindsay, K., and Krasny, R., "A Particle Method and Adaptive Treecode for Vortex Sheet Motion in Three-Dimensional Flow," *Journal of Computational Physics*, Vol. 172, No. 2, 2001, pp. 879–907.

[11]Caradonna, F. X., and Tung, C., "Experimental and Analytical Studies of a Model Helicopter Rotor in Hover," *Vertica*, Vol. 5, 1981, pp. 149–161.

[12]Kocurek, J. D., and Tangler, J. L., "A Prescribed Wake Lifting Surface Hover Performance Analysis," 32nd Annual National Forum of the American Helicopter Society, Washington, DC, May 1976.

[13]Harris, F. D., "Articulated Rotor Blade Flapping Motion at Low Advance Ratio," *Journal of the American Helicopter Society*, Vol. 17, No. 1, 1972, pp. 41–48.

[14]Hansford, R. E., and Vorwald, J., "Dynamics Workshop on Rotor Vibratory Loads," 52nd Annual National Forum of the American Helicopter Society, Washington, DC, June 1996.

[15]Brown, R. E., and Whitehouse, G. R., "Modelling Rotor Wakes in Ground Effect," *Journal of the American Helicopter Society*, Vol. 49, No. 3, 2004, pp. 238–249.

R. So
Associate Editor