HIGH RESOLUTION WAKE MODELLING USING A SEMI-LAGRANGIAN ADAPTIVE GRID FORMULATION

A. J. Line* and R. E. Brown†

Department of Aeronautics
Imperial College London
Prince Consort Road,
London, United Kingdom.

Abstract

Current interest in improving numerical predictions of helicopter rotor vibration and acoustic signature has driven a requirement to model blade aerodynamic loadings to extremely high resolution. The problem is complicated by the complexity of the flow in the rotor wake and the difficulty in conserving vortical structures in the wake for long enough so that all important vibration- or noise-producing blade vortex interactions are represented properly. The Vorticity Transport Model, developed by Brown, is capable of preserving vortical structures over sufficiently long timescales, but, up to now, has been prohibitively expensive when run at the resolutions required for accurate vibration or acoustic prediction. In this paper we present a new computational grid system which, through the use of adaptive cell management and nested grids, allows significant increases in grid resolution with minimal impact on computational cost. The new grid system is effectively boundary-free, thus eliminating the need for numerical boundary conditions. In addition, the velocity field is optimally evaluated on the new grid using an extremely efficient technique based on the Cartesian Fast Multipole Method. The implementation of both the grid system and the velocity calculation within a new version of the Vorticity Transport Model is described. The performance of the new model is validated against experimental data and some properties of the model, when used to predict rotor loads to a resolution that approaches that required for calculations of rotor vibration or acoustic signature, are illustrated.

Nomenclature

a_k : k^{th} Taylor coefficient of K_δ
A : Rotor area
b_k : k^{th} Taylor coefficient of ϕ_δ
c : cluster
C_L : blade section lift coefficient
C_T : rotor thrust, scaled by $\rho A(\Omega R)^2$
F_k : k^{th} derivative of the velocity field
k : term of multipole series
K : Biot-Savart Kernel
K_δ : regularised Biot-Savart Kernel
m_k : k^{th} moment of vorticity
N : number of grid cells
N_c : number of cells within a cluster
p : order of multipole expansion
R : rotor radius
S : vorticity source
v : flow velocity
v_b : flow velocity relative to blade
β : blade flapping angle
$\beta = \beta_0 + \beta_{1s}\sin\psi + \beta_{1c}\cos\psi$
β_0 : rotor coning angle
β_{1s} : rotor lateral tilt angle
β_{1c} : rotor longitudinal tilt angle
δ : kernel smoothing parameter
Δ : cell edge-length
θ : blade feathering angle
$\theta = \theta_0 + \theta_{1s}\sin\psi + \theta_{1c}\cos\psi$
θ_0 : collective pitch control angle
θ_{1s} : longitudinal cyclic pitch control angle
θ_{1c} : lateral cyclic pitch control angle
μ : rotor forward speed scaled by ΩR
σ : rotor solidity
ϕ_δ : regularised Newtonian potential
ω : vorticity, $\nabla \times v$
ω_b : bound vorticity
Ω : rotor rotational speed $d\psi/dt$

Introduction

This paper outlines the development of a new rotor wake modelling tool based on the existing Vorticity Transport Model (VTM) of Brown (Ref. 1). The VTM is a CFD-based free wake model that has been used in a number of applications, including flight mechanics (Ref. 2), analysis of the vortex ring state (Ref. 3), and the modelling of aircraft wake encounters (Ref. 4). The new version of the VTM described within this paper has been developed to address some of the specific numerical issues posed by the prediction of rotor vibration and acoustic signature.

The results of the 1996 dynamics workshop, as reported by Hansford and Vorwald (Ref. 5), exposed a clear, industry-wide difficulty in predicting accurately the vibra-

*Postgraduate Research Assistant
†Lecturer

Presented at the European Rotorcraft Forum, Friedrichshafen, Germany, Sept. 16-18, 2003.

tion levels generated by helicopter rotors. Hansford and Vorwald concluded that the use of free wake models to predict the aerodynamic environment of the helicopter could "greatly enhance vibration correlation" and, indeed, that introduction of this level of aerodynamic modelling into aeroelastic codes would be essential if the perceived difficulties in vibration prediction were to be overcome.

The potential of free wake models lies in their ability to predict the aerodynamic environment of the rotor in such a way that the evolution of the flow field of the rotor is unconstrained by any preconceptions regarding the geometry of the rotor wake. The improvement in physical fidelity that such approaches offer over simpler modelling techniques should translate into an improved prediction of the aerodynamic loading on the blades, particularly in terms of the locations and strengths of the blade vortex interactions that are key to the accurate prediction of rotor vibration. Nonetheless, the rotor wake is a highly complex vorticity distribution and the potential improvement that any free wake model can offer is limited by the finest spatial resolution that it can provide of the vortical structures present within the rotor wake. Since the smallest features in the blade loading distribution are on the scale of individual blade vortex interactions, it can be argued that a fully convergent numerical representation of the wake will require sufficient resolution to capture the internal structure of vortex cores and sheets. In his discussion of the current challenges in rotorcraft aerodynamics, Caradonna (Ref. 6) points out the significant computational cost and grid-size problems associated with achieving such high resolutions within CFD-type models. The realisation of CFD-based free wake models that can achieve such resolutions is still some distance in the future. It will be essential, however, that the number of computational cells used by these techniques be minimised and that the numerical approach not place excessive demands upon grid resolution solely for the purpose of limiting numerical diffusion of vorticity.

In contrast to the majority of CFD-based rotor analysis codes, the VTM represents the flow directly in terms of the vorticity distribution in the flow field surrounding the rotor. The evolution of the flow field is then modelled via numerical solution of the fluid dynamic equations that govern vorticity transport in an inviscid, incompressible fluid. The approach explicitly enforces Helmholz's law for vorticity conservation and is capable of preserving vortical structures in the flow for very long periods of time. Early versions of the VTM were prohibitively expensive, however, when run at the resolutions required for accurate prediction of rotor vibration or acoustics. This was because these versions of the code used a fixed, uniform, cartesian grid of cells to contain the vorticity surrounding the rotor. This approach is not the most efficient in terms of cell count since many cells never contain vorticity during the course of a calculation. In fact, many cells are present solely to track possible future evolution of the vorticity field or to allow enforcement of far-field boundary conditions.

A fixed grid system wastes memory and places an unnecessarily low limit on the finest practicable resolution of the rotor flow. A new grid system has been developed for the VTM that addresses this deficiency. The new grid system uses adaptive cell generation and grid nesting to achieve typically an order of magnitude reduction in cell count compared to earlier versions of the code. A very efficient use of memory can be achieved by taking advantage of the vorticity-based framework of the VTM, which requires that cells exist only in regions of space where the vorticity is non-zero. The new grid system works by creating and destroying cells on a fixed background stencil so that the grid follows regions of vorticity within the flow in a semi-Lagrangian fashion. It should be noted though that, in the present implementation, individual cells remain fixed within the frame of reference attached to the rotor hub (rather than moving physically with the flow in a truly Lagrangian sense) and that the transfer of vorticity from cell to cell is modelled using an explicitly Eulerian approach.

The relatively unstructured nature of the resulting grid lends itself particularly well to the use of fast-particle type algorithms for calculating the velocity field associated with the vorticity distribution in the flow. In the present implementation of the VTM, the velocity throughout the grid is calculated using a technique based on the Cartesian form of the Fast Multipole Method (FMM). This method is extremely efficient, and, in conjunction with the new grid structure, permits a velocity calculation on a grid containing N computational cells to be performed in just $O(N)$ operations. This should be compared to the $O(N^2)$ operations count of more traditional techniques based on direct evaluation of the Biot-Savart integral.

Validation of the enhanced VTM code against the well known experimental data of Harris (Ref. 7) is presented within this paper, and some of the properties of the code, when used to predict rotor loads to a resolution that approaches that required for calculations of rotor vibration or acoustic signature, are illustrated.

Flow Model

Arguably the most efficient way to model the vorticity-dominated aerodynamic environment of a helicopter rotor is to model the rotor wake directly as a time-dependent vorticity distribution in the region of space surrounding the rotor. If v is the flow velocity, then the associated vorticity distribution $\omega = \nabla \times v$ evolves according to the unsteady vorticity transport equation

$$\frac{\partial}{\partial t}\omega + v \cdot \nabla\omega - \omega \cdot \nabla v = S \qquad (1)$$

This equation can be derived from the incompressible Navier-Stokes equation in the limit of zero viscosity. The differential form of the Biot-Savart equation then relates the velocity and vorticity fields throughout the flow:

$$\nabla^2 v = -\nabla \times \omega \qquad (2)$$

The Vorticity Transport Model (VTM) developed by Brown (Refs. 1, 2) employs a direct computational solution of Eq. 1 to simulate the evolution of the rotor

flow field. After enclosing the rotor and its surroundings within an appropriate grid of computational cells, the vorticity distribution in the flow is advanced through time using a computational discretisation of Eq. 1, using Toro's Weighted Average Flux (WAF) algorithm (Ref. 8) to construct the inter-cell vorticity fluxes.

The use of a vorticity-velocity formulation presents a number of advantages over standard approaches written in terms of primitive variables. The most important of these is that Eq. 1 can be implemented in a form that explicitly enforces Helmholz's law for vorticity conservation. When a suitable flux limiting function is used in conjunction with the WAF algorithm, diffusion of vorticity can then be controlled to the extent that vortical structures in the flow are preserved - even during very long computations (Ref. 9).

Rotor Model

At present, the rotor system is physically described within the VTM as a set of rigid blades attached to a hub that remains fixed in its position and orientation within the computational domain. The computational domain is thus assumed to rotate and translate along with the rigid-body motions of the helicopter's fuselage. The hub itself is modelled as a collection of discrete hinges along with their associated springs and dampers. The deformations that the rotor can undergo are described in terms of set of generalised coordinates. The kinetic energy, potential energy and energy dissipation associated with the deformation of the rotor are then used to form the Lagrangian of the system. Finally, the dynamics of the rotor system are obtained at each computational timestep by numerical differentiation of the Lagrangian to yield the appropriate equations of motion. After converting to a first-order system written in terms of the generalised coordinates and their rates of change, the equations of motion are integrated numerically to obtain the blade motions. Although not yet implemented, such an approach can, in principle, be extended to modelling the dynamics of rotors having blades with elastic degrees of freedom.

The aerodynamic behaviour of the rotor blades is modelled by partitioning each blade into a number of spanwise panels and applying lifting line aerodynamics to each panel. The local flow velocity, v_b, at a collocation point located on each panel is calculated as the sum of the velocity induced by the vorticity in the system, the free-stream velocity and the structural motion of the blade. Setting the component of the velocity normal to each panel 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. Conditioning of the chordwise position of the collocation points allows the full 360-degree aerodynamic performance of the blade's aerofoils to be captured. To improve the accuracy of the calculation of the unsteady aerodynamics of the blade, particularly at very high reduced frequencies, the vorticity generated by each panel is captured on a high-resolution vortex lattice extending typically two chord-lengths behind each blade. The geometry of this lattice is allowed to evolve freely under the influence of the local flow velocity, and the vorticity on the lattice is systematically transferred into the computational grid where it then evolves according to Eq. 1. This transfer is done by constructing the source term, S, in Eq. 1 in terms of the shed and trailed vorticity from the rotor blades as follows:

$$S = -\frac{d}{dt}\omega_b + v_b \nabla \cdot \omega_b \qquad (3)$$

thus coupling the wake evolution into the aerodynamic loading, and hence the dynamics, of the rotor system.

Grid Formulation

If the flow is given sufficient time to evolve, vorticity will eventually be required to cross the boundaries of any computational grid that has finite spatial extent. A special feature of rotor flows is that the vortical structures produced in the rotor wake are so complex and time-dependent, even far downstream from the rotor, that the influence on the rotor of any vorticity lost through the grid boundaries cannot rigorously be accounted for simply by applying algebraic or differential conditions at the grid boundaries. This is because the long-range interactions that result from the vorticity-velocity relationship (Eq. 2) strongly couple the present state of the flow near the rotor to vorticity sourced into the flow at very much earlier times. Stated in equivalent terms, the evolution of the flow, and hence the blade loading, is strongly dominated by its own history. Truncation of the wake without appropriate application of boundary conditions will artificially alter the structure of the wake, thereby contaminating its future evolution.

A new grid system that eliminates all physical grid boundaries, and thus avoids the problems introduced by truncation of the wake at grid boundaries, has been implemented within the VTM. A process of cell generation and destruction is used adaptively to encapsulate and track any regions of flow containing vorticity as follows:

First, an underlying cartesian grid-stencil or framework is generated, upon which cells can be created and destroyed. 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. The grid cells on the stencil are cubic, with edge-length Δ_0.

When vorticity is sourced into the grid from the rotor model, a cell is created at the appropriate point on the stencil. To allow the vorticity to advect, the immediate neighbours of the newly created cell must also be created, as shown in Fig. 1. Assuming that the velocity field is known at every cell interface, this grid structure is all that is required to evolve the flow according to Eq. 1. At successive timesteps, additional neighbour 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 neighbours of vorticity-containing cells, are destroyed. This process of cell generation and destruction is repeated at every timestep to allow the flow to evolve freely whilst minimising cell count. Since the grid stencil has no boundaries, the grid is free to expand in space as the wake structure expands. Thus, the resulting grid structure effectively tracks vortical regions of flow through time.

The drawback of the system as described is that the computer's memory limit is soon reached if the number of cells grows too rapidly. At that point, the model is unable to accommodate any further growth in the grid, forcing the simulation to terminate. This problem is overcome by introducing a number of nested grids with varying spatial resolution.

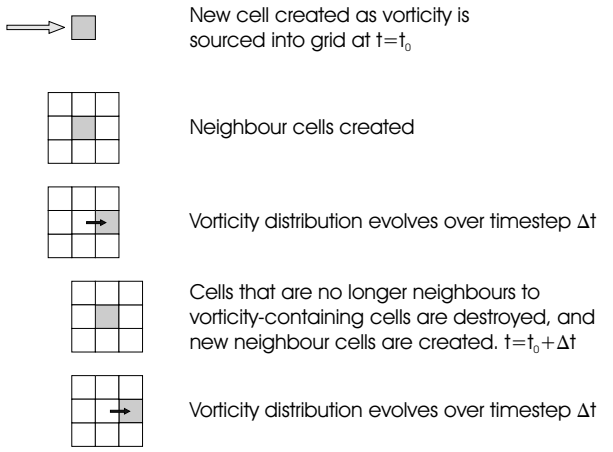

Figure 1: *Adaptive grid generation and destruction tracks regions of vorticity.*

At the finest level, the rotor is enclosed in a grid with cells having edge-length Δ_0, equal to that of the underlying stencil. At some distance away from the rotor, the grid is coarsened to a resolution half that of the previous grid. This process can be repeated as many times as is required (by creating the cells on grid level i to have side-length $\Delta_i = \Delta_0 \times 2^i$) to generate a computational domain that systematically decreases in resolution as the distance from the rotor is increased. A schematic representation of a computational domain with such a nested grid structure is shown in Fig. 2.

Because of the exponential growth in cell size on moving up through the hierarchy of grids, the use of nesting slows the growth rate of the grid in terms of cell count. By sizing and positioning the nested grids appropriately, the time taken to perform a rotor simulation can be tailored to avoid exceeding memory limits. Furthermore, this approach allows the available computational power to be focused efficiently into the region of flow closely surrounding the rotor, whilst still accounting for the influence and evolution of the far wake. The resulting grid structure intrinsically encapsulates the entire wake-history of the rotor while postponing indefinitely any need to prescribe boundary conditions.

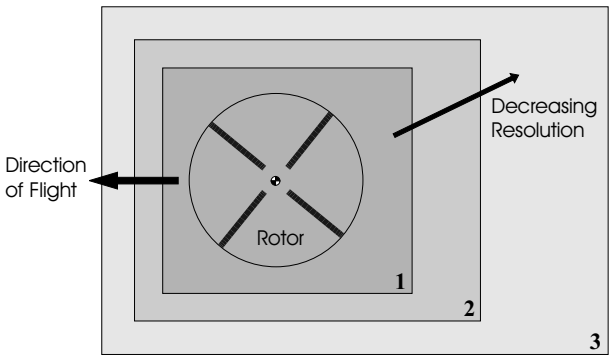

Figure 2: *The rotor is embedded within a nested grid structure that decreases in resolution with distance from the hub.*

A further benefit of using multiple nested grid levels 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. 1 through time, the Courant-Friedrichs-Lewy (CFL) condition sets the maximum allowable timestep based on the local velocity and the local resolution of the grid. Since 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 timestep very close to 2^i times the timestep used for the finest grid level. Computational effort is thus focused where it is most required, that is, on the flow in the highly resolved regions of the computational domain closest to the rotor.

Successful implementation of grid nesting requires an effective procedure for transferring 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, by two cell-widths as measured on the coarser grid level, the cells at the interface between adjacent levels of the hierarchy and reconciling the vorticity contained within the resulting overlap regions at appropriate points during the course of the calculation. In this way it is possible to maintain conservation of vorticity and to preserve (locally) the second-order and monotonic properties of the WAF procedure at all cell interfaces.

Velocity Calculation

In the original versions of the VTM, evaluation of the velocity field throughout the computational domain was achieved by inverting Eq. 2 using Schumann and Sweet's (Ref. 10) Method of Cyclic Reduction after applying appropriate boundary conditions at the edges of the computational domain. The velocity at each cell within the computational domain is a function of the position and strength of every other cell containing vorticity within the domain. This results in a classical N-body interaction problem. For a computational domain containing N cells, cyclic reduction reduces the cost of the calculation of the N-body problem to $O(N \log N)$, compared with the $O(N^2)$ cost of a

direct Biot-Savart approach. It is, however, very difficult to implement cyclic reduction together with appropriate boundary conditions without contaminating the numerical solution - especially during calculations involving many rotor revolutions. Furthermore, the unstructured and dynamic nature of the new grid system renders implementation of cyclic reduction impractical, both in terms of the application of boundary conditions and in terms of the computational cost of re-structuring the method after every timestep.

Given the structure of the new grid system, large increases in computational efficiency can be achieved by replacing the Method of Cyclic Reduction with the Fast Multipole Method (FMM). The FMM is one of a number of hierarchical $O(N)$ algorithms which have matured greatly since their first appearance in the mid-1980's. The earliest methods of this type, such as those of Appel (Ref. 11) and Barnes and Hut (Ref. 12), used hierarchical decomposition of the computational domain to reduce the computational cost of the problem to $O(N \log N)$. These ideas were extended by Greengard and Rokhlin (Ref. 13) to further reduce the cost to $O(N)$. Early forms of the FMM were applied primarily to the study of gravitational and electrostatic fields. It is possible, however, to adapt the method to analyse any particle-based system where the long-range interactions can be expressed in terms of an appropriate Green's function.

Fast Multipole Velocity Calculation

The FMM, in its Cartesian form, has been implemented within the latest version of the VTM to evaluate the velocity field throughout the computational domain. The CPU time required by the new code (on a Pentium 4 2GHz processor with 1Gb of memory) to evaluate the velocity at all cell interfaces of a computational domain containing N computational cells is shown in Fig. 3. The figure demonstrates convergence of the code on the theoretical $O(N)$ cost for $N > 10,000$.

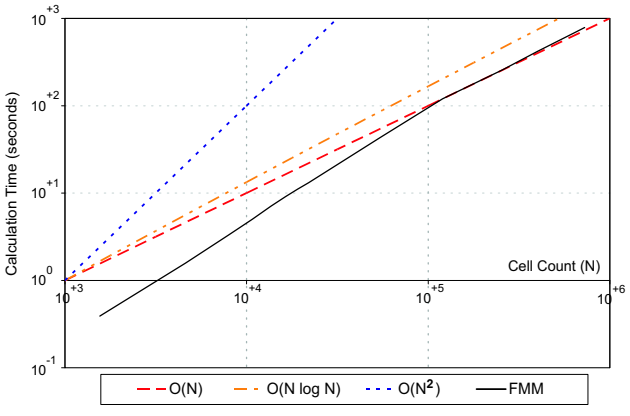

Figure 3: *Computational cost of the FMM for a grid system containing N computational cells.*

The appropriate Green's function to be used by the FMM for inversion of Eq. 2 is the Biot-Savart kernel:

$$K(\mathbf{x},\mathbf{y}) = -\frac{1}{4\pi}\frac{(\mathbf{x}-\mathbf{y})}{|\mathbf{x}-\mathbf{y}|^3} \tag{4}$$

To acknowledge the fact that the computational cells contain a uniform distribution of vorticity rather than a vortex singularity, the Biot-Savart kernel is modified using a smoothing parameter, δ, to produce a regularised kernel, more commonly known as the Rosenhead-Moore kernel:

$$K_\delta(\mathbf{x},\mathbf{y}) = -\frac{1}{4\pi}\frac{(\mathbf{x}-\mathbf{y})}{(|\mathbf{x}-\mathbf{y}|^2+\delta^2)^{3/2}} \tag{5}$$

The value of δ is chosen such that the maximum velocity induced by the vorticity within a cell is located on the face of the cell, as shown in Fig. 4.

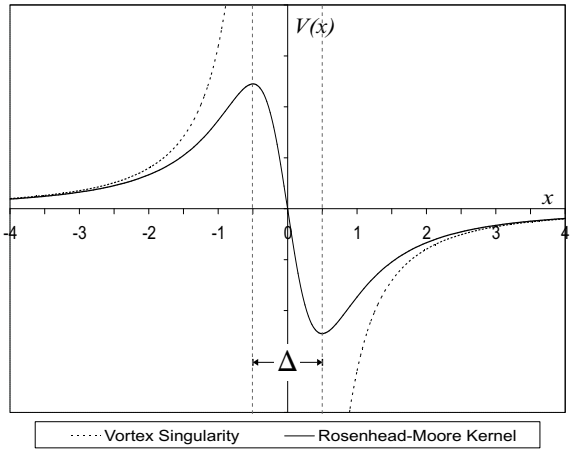

Figure 4: *Comparison between the velocity field generated by a vortex singularity and that of a cell containing a uniform distribution of vorticity.*

Theory

The Fast Multipole algorithm builds on the idea that the velocity field induced by a number of distant, but closely grouped, cells of vorticity can be approximated by a single interaction with a multipole source representing the vorticity contained within those cells. Fig. 5 shows a cluster, c, containing N_c cells, each containing vorticity. The vorticity-weighted centre of the cluster lies at $\mathbf{y}_c$ and the centre of each cell lies at $\mathbf{y}_j$. The point $\mathbf{x}$ is located outside of c and is a distance R from $\mathbf{y}_c$.

The velocity induced at point $\mathbf{x}$ by the vorticity within cluster c can be approximated by

$$\mathbf{v}(\mathbf{x}) = \sum_{j=1}^{N_c} \mathbf{K}_\delta(\mathbf{x},\mathbf{y}_j) \times \omega_j \tag{6}$$

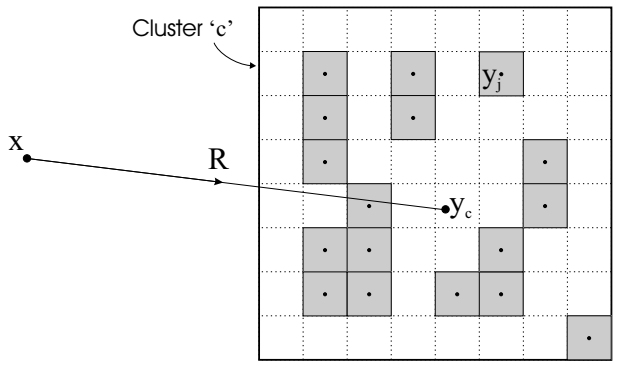

Figure 5: *Evaluation of the velocity field at point* **x**, *due to the vorticity contained with the cluster c.*

Expanding the right hand side in a Taylor series about the centre of the cluster and truncating the series after the p^{th} term yields

$$\begin{aligned}\mathbf{v}(\mathbf{x}) &= \sum_{j=1}^{N_c} K_\delta(\mathbf{x}, \mathbf{y}_c + (\mathbf{y}_i - \mathbf{y}_c)) \times \omega_j \\ &= \sum_{j=1}^{N_c} \sum_{\mathbf{k}} \frac{1}{\mathbf{k}!} D_{\mathbf{y}}^{\mathbf{k}} \mathbf{K}_\delta(\mathbf{x}, \mathbf{y}_c)(\mathbf{y}_j - \mathbf{y}_c)^{\mathbf{k}} \times \omega_j \\ &\approx \sum_{\mathbf{k}=\mathbf{0}}^{\mathbf{p}} \mathbf{a}_{\mathbf{k}}(\mathbf{x}, \mathbf{y}_c) \times \mathbf{m}_{\mathbf{k}}(c) \end{aligned} \quad (7)$$

where $D_{\mathbf{y}}^{\mathbf{k}} = \partial/\partial y_1^{k1} \partial y_2^{k2} \partial y_3^{k3}$, $\mathbf{k}! = k_1!k_2!k_3!$, $\mathbf{x}^k = x_1^{k_1} x_2^{k_2} x_3^{k_3}$ for $k_i \geq 0$, and the subscripts 1, 2 and 3 refer to the Cartesian directions. The multipole tensors $\mathbf{a_k}$ are functions of the range of the interaction alone, whereas the moments of vorticity, $\mathbf{m_k}$, describe the local distribution of vorticity within the cluster c:

$$\mathbf{a}_{\mathbf{k}}(\mathbf{x}, \mathbf{y}_c) = \frac{1}{\mathbf{k}!} D_{\mathbf{y}}^{\mathbf{k}} \mathbf{K}_\delta(\mathbf{x}, \mathbf{y}_c) \quad (8)$$

$$\mathbf{m}_{\mathbf{k}}(c) = \sum_{j=1}^{N_c} (\mathbf{y}_j - \mathbf{y}_c)^{\mathbf{k}} \omega_j \quad (9)$$

Differentiation of Eq. 7 allows the velocity field at point **x** to be described in terms of **p** local derivatives:

$$\mathbf{F}_{\mathbf{k}}(\mathbf{x}) = \sum_{\mathbf{n}=\mathbf{k}}^{\mathbf{p}} (-1)^{\mathbf{n}+\mathbf{1}} \frac{\mathbf{n}!}{(\mathbf{n}-\mathbf{k})!} \mathbf{a}_{\mathbf{n}}(\mathbf{x}, \mathbf{y}_c) \times \mathbf{m}_{\mathbf{n}-\mathbf{k}}(c) \quad (10)$$

where $\mathbf{F}_{\mathbf{k}}(\mathbf{x}) = D_{\mathbf{x}}^{\mathbf{k}} \mathbf{v}(\mathbf{x})$.

A final result is required to translate a description of the local velocity field from one point to some other nearby point. Eq. 11 translates the centre of the Taylor Series expansion of the velocity from its original position, $\mathbf{x}_A$ to a new location $\mathbf{x}_B$:

$$\mathbf{F}_{\mathbf{k}}(\mathbf{x}_B) = \sum_{\mathbf{n}=\mathbf{k}}^{\mathbf{p}} \frac{(\mathbf{x}_B - \mathbf{x}_A)^{\mathbf{n}-\mathbf{k}}}{(\mathbf{n}-\mathbf{k})!} \mathbf{F}_{\mathbf{n}}(\mathbf{x}_A) \quad (11)$$

Tensor Calculation

The tensors, $\mathbf{a_k}$, defined in Eq. 8 can be evaluated efficiently using recursion. The following result relies on the fact that the gradient of the Rosenhead-Moore kernel is the regularised Newtonian potential

$$\phi_\delta(\mathbf{x}, \mathbf{y}) = \frac{1}{4\pi} \frac{1}{(|\mathbf{x}-\mathbf{y}|^2 + \delta^2)^{1/2}} \quad (12)$$

If the scaled derivatives of the Newtonian potential,

$$b_{\mathbf{k}}(\mathbf{x}, \mathbf{y}_c) = \frac{1}{\mathbf{k}!} D_{\mathbf{y}}^{\mathbf{k}} \phi_\delta(\mathbf{x}, \mathbf{y}_c) \quad (13)$$

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

$$\begin{aligned}\|\mathbf{k}\| R^2 b_{\mathbf{k}} - (2\|\mathbf{k}\| - 1) \sum_{i=1}^{3} (x_i - y_i) b_{\mathbf{k}-\mathbf{e}_i} \\ + (\|\mathbf{k}\| - 1) \sum_{i=1}^{3} b_{\mathbf{k}-2\mathbf{e}_i} = 0 \end{aligned} \quad (14)$$

where $R^2 = |\mathbf{x}-\mathbf{y}|^2 + \delta^2$, $\|\mathbf{k}\| = k_1 + k_2 + k_3$ and $\mathbf{e}_i$ is the i^{th} Cartesian basis vector. This result is derived in full by Lindsay and Krasny (Ref. 14). Once all $b_{\mathbf{k}}$ are known, the tensors $\mathbf{a_k}$ can be reconstructed as

$$\mathbf{a}_{\mathbf{k}}(\mathbf{x}, \mathbf{y}_c) = -\sum_{i=1}^{3} (k_i + 1) b_{\mathbf{k}+\mathbf{e}_i} \mathbf{e}_i \quad (15)$$

Implementation

Starting from the underlying grid structure, a level of parent clusters is created by grouping cubes of eight cells together, as shown in Fig. 6. These clusters are then grouped in the same way to form the next level of larger parent clusters. This process is repeated until the entire grid is contained within a single root cluster. In three dimensions this process then yields an octree data structure containing the cells and their parents at each level.

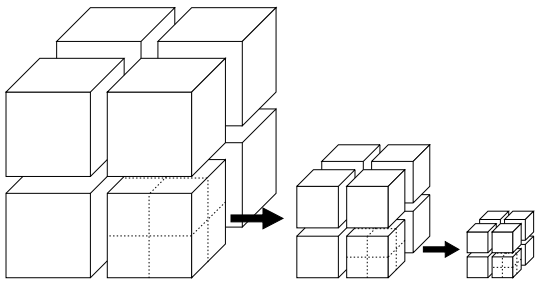

Figure 6: *Octree data structure.*

The first stage of the FMM involves an upward sweep through the tree, calculating the moments of each cluster

according to Eq. 9. This stage can be accelerated by expressing the moments of a cluster as a binomial sum of the moments of its children:

$$\mathbf{m_k}(\tau) = \sum_{\bar{\tau}} \sum_{\mathbf{n}=0}^{\mathbf{k}} \binom{k}{n} (\mathbf{y}_{\bar{\tau}} - \mathbf{y}_{\tau})^{\mathbf{k}} \, \mathbf{m_{k-n}}(\bar{\tau}) \qquad (16)$$

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

The second stage of 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. The velocity field within any arbitrary cell *i* within the tree can be considered as the sum of a far-field component and a near-field component. The near-field contribution to the velocity in cell *i* is obtained as the sum of the interactions with its neighbouring cells, after partitioning these cells into two separate sets as follows. The interaction set *A* for cell *i* is defined as all of cell *i*'s nearest neighbours, including cell *i* itself. Interaction set B is defined as the children of the nearest neighbours of cell *i*'s parent, excluding the cells contained within interaction set A. The structure of the interaction sets is illustrated in Fig. 7. For clarity the 2-dimensional interaction sets for an arbitrary cell *i* have been shown, but the diagram extends easily to three dimensions.

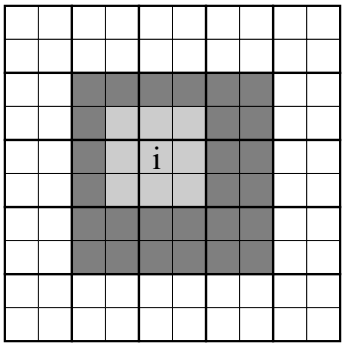

Figure 7: *The 2D Interaction Sets for arbitrary cell i. Light grey cells are members of Set A. Dark grey cells are members of Set B.*

The velocity induced at the centre of cell *i* by each member of its interaction set *B* is evaluated directly using Eq. 10. The economy of the FMM comes about because, instead of directly evaluating the far-field contribution to the velocity field at the centre of cell *i* resulting from all cells outside of interaction set *B*, the velocity contribution of these cells is inherited from the parent of cell *i* by translating the velocity of the parent cluster to the child according to Eq. 11. This process of evaluation and inheritance is performed on all clusters on a given level before descending to the next level of the tree. Once the downward sweep through the tree is complete, the velocity at any point within cell *i* can be evaluated as follows:

$$\mathbf{v}(\mathbf{x}) = \sum_{\mathbf{k}=0}^{\mathbf{p}} \frac{(\mathbf{x} - \mathbf{x}_p)^{\mathbf{k}}}{\mathbf{k}!} \mathbf{F_k}(\mathbf{x}_p) \qquad (17)$$

where $\mathbf{x}_p$ is the centre of cell *i*'s parent. The only remaining task is then to add on the contributions from the cell's nearest neighbours, or, in other words, those contained in its interaction set *B*, by direct application of the Biot-Savart relationship

$$\mathbf{v}(\mathbf{x}) = \sum_{j \in A} \mathbf{K}_{\delta}(\mathbf{x}, \mathbf{y}_j) \times \omega_j \qquad (18)$$

Code Validation

A validation of the new code against the wind tunnel data produced by Harris (Ref. 7) is presented in Fig. 8. Harris measured the flapping behaviour of an isolated 4-bladed 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. A comparison of the experimental data and calculated disc tilts produced using both the new and old versions of the VTM code is shown in the figure. The error bars represent Harris' own estimate of the accuracy of his measurements. All numerical results were produced using 20 blade aerodynamic collocation points in a cosine distribution along each blade, and by resolving the blade radius across 25 computational cells at the finest grid level. This level of resolution is relatively coarse, but is completely adequate for predicting the performance of an isolated rotor with simple geometry when using the VTM. Four levels of grid expansion were used in the new version of the VTM, with transitions between grid

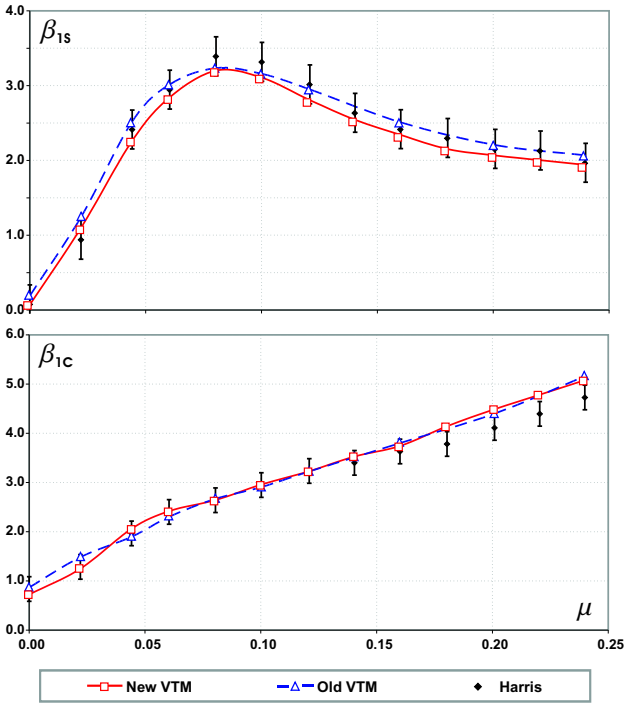

Figure 8: *Validation of VTM codes against Harris' experimental data: Lateral Disc Tilt (top) and Longitudinal Disc Tilt (bottom).*

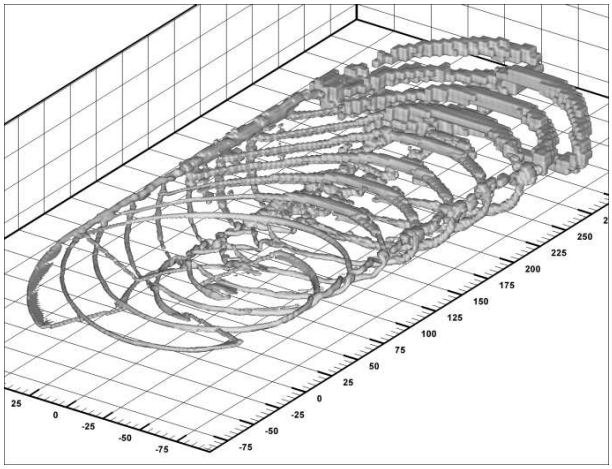

Figure 9: *Visualisation of the rotor wake of a 4-bladed rotor at* $\mu = 0.250$. *(Vorticity contour set to visualise the tip vortex structure.)*

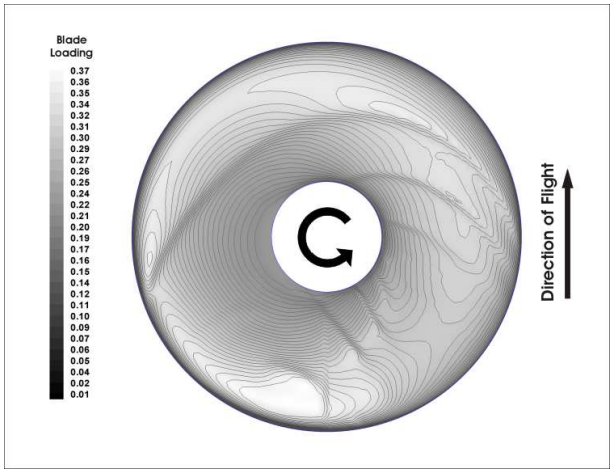

Figure 10: *Azimuthal variation of Blade Loading for 4-bladed rotor,* $\mu = 0.250$.

levels positioned at $2R$, $6R$ and $10R$ from the rotor hub.

The VTM shows good agreement with Harris' experimental data for both the lateral and the longitudinal tilt of the rotor. The predicted values of the lateral disc tilt, although marginally lower than the experimental values, lie within the experimental error bounds over the full range of flight speeds. The new version of the VTM produces somewhat better correlation than the old at low forward speeds, and this is consistent with experience that calculations with the old code were more likely to be susceptible to contamination by boundary conditions at low to transitional advance ratios than at high. The VTM again shows very good correlation with Harris' data for the longitudinal disc tilt. For much the same reasons as for the lateral disc tilt, the new version of the VTM gives better results at low speed compared to the older version of the code. The consistent behaviour between the two versions of the code in slightly over-predicting the longitudinal tilt at high forward speed ($\mu > 0.20$) is indicative of a deficiency in the present blade aerodynamic model, most likely in its treatment of the reverse-flow region on the retreating side of the rotor.

Code Behaviour

In this section, some examples are presented to illustrate the behaviour of the enhanced VTM when used to predict rotor flows to a resolution that approaches that required for accurate calculation of rotor vibration or acoustic signature. Figure 9 shows a visualisation of the wake generated behind a representative four-bladed rotor operating at $C_T/\sigma = 0.072$ and travelling at an advance ratio $\mu = 0.250$. A surface on which the vorticity in the flow has constant magnitude has been plotted - the vorticity contour has been selected to suppress the detail of the inboard wake and to expose the geometry of the root and tip vortices trailed from the blades. In this calculation, the finest grid level gave a resolution of 100 cells along the rotor radius, and 40 aerodynamic collocation points were used in a cosine distribution along each blade. In contrast, earlier versions of the VTM were limited to a maximum resolution of around 50 cells per rotor radius on a desktop PC with 1 gigabyte of memory. Three levels of grid expansion were used before the calculation was terminated, with transitions between grid levels positioned at $1.5R$ and $3R$ from the rotor hub. These transitions are clearly visible in Fig. 9 as two rather obvious step-changes in the resolution of the tip vortices downstream of the rotor. To illustrate how the effects of the early history of the wake can be retained within the computation using the approach described in this paper, the calculation was terminated while the vortices shed by the rotor on start-up were still contained within the third grid level. Further continuation of the calculation would have seen further reduction of the resolution of the starting vortices as they were convected into the increasingly coarser levels of the computational grid further downstream of the rotor. Nevertheless, their effect on the velocity distribution at the rotor would still be captured correctly by the Fast Multipole Method.

Figure 10 shows the azimuthal variation of blade loading associated with the wake structure shown in Fig. 9. Fine resolution of the vorticity in the wake translates into a well-defined sequence of ridges in the blade loading that result from interactions of the blade with the concentrated vortices shed from both the roots and the tips of the rotor. At this advance ratio the layout of the blade-vortex interactions on the disc plane is relatively simple since the rate of self-induced deformation of the wake structure is relatively insignificant when compared to the rate of convection of the vorticity into the flow downstream of the rotor. For comparison, Fig. 11 shows the wake structure generated by the same rotor when travelling at a somewhat reduced advance ratio $\mu = 0.150$. In this case the enhanced distortion of the tip vortices close to the rotor manifests itself as the somewhat more subtle distribution of BVI-induced ridges in the blade loading shown in Fig. 12.

Finally, Fig. 13 shows the same wake structure as in Fig. 11 but with the vorticity contour selected to expose the overall morphology of the wake on the most finely resolved level of the grid. The evolution of the sheets of

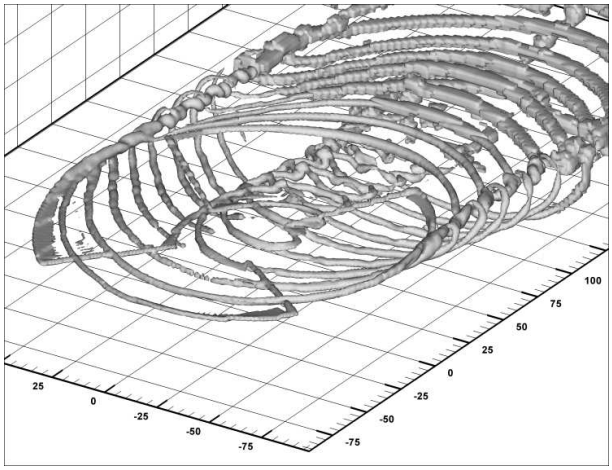

Figure 11: *Visualisation of the rotor wake of a 4-bladed rotor at $\mu = 0.150$. (Vorticity contour set to visualise the tip vortex structure.)*

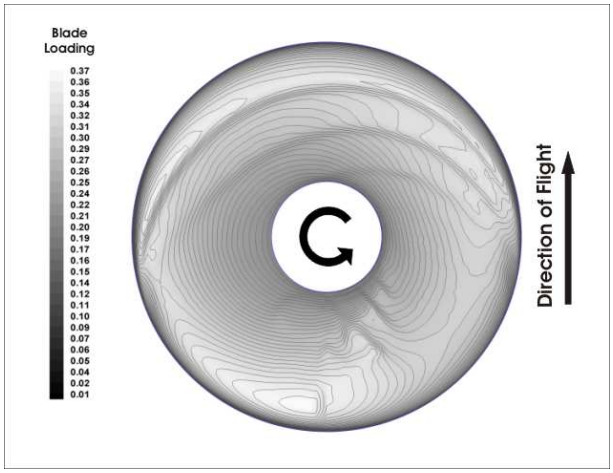

Figure 12: *Azimuthal variation of Blade Loading for 4-bladed rotor, $\mu = 0.150$.*

vorticity shed from the inboard parts of the blades, and their interactions with the blades and the stronger vorticity trailed from the roots and tips of the blades is captured fairly convincingly by the model, but some artifacts of the mesh are visible as a series of parallel striations running across some of these structures. The effects of these spurious features of the calculation on the long-term evolution of the wake have yet to be determined.

Conclusions

A new multi-resolution grid system and velocity calculation have been incorporated into an existing free wake model based on a CFD-type solution of the Vorticity Transport Equation. The new grid system brings about reductions in cell count of typically an order of magnitude compared to the use of a structured, uniform Cartesian mesh by using adaptive creation and destruction of cells to follow the evolution of the vorticity in the flow. When used in conjunction with a Fast Multipole velocity calculation, the new grid system allows the velocity field throughout a computational domain consisting of N cells to be calculated with $O(N)$ computational cost. These features result in an extremely efficient use of computer memory, and allow computational effort to be focused onto the regions of the rotor flow that need to be most highly resolved. The resulting code validates well against existing experimental data, and yields plausible high-resolution predictions of unsteady blade loading. The hope is that, with some further development, the techniques described in this paper might provide a feasible route to the accurate prediction of rotor vibration and acoustic signature.

Acknowledgements

The work presented in this paper is sponsored by a U.K. Engineering and Physical Sciences Research Council CASE Award in partnership with Westland Helicopters Limited.

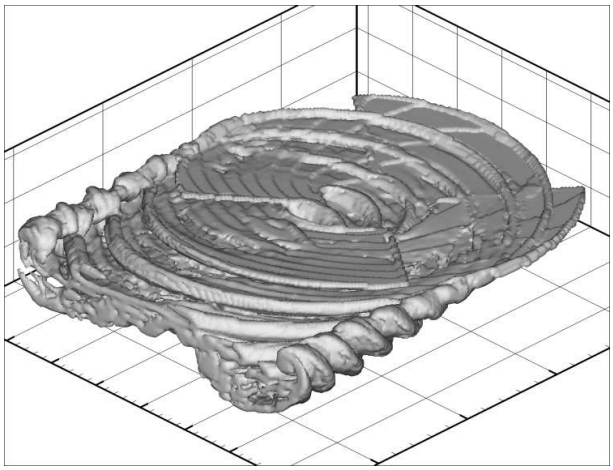

Figure 13: *Visualisation of the rotor wake of a 4-bladed rotor at $\mu = 0.150$. (Vorticity contour set to visualise the global wake morphology.)*

References

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

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

[3]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, September 2002.

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

[5]Hansford, R. E., and Vorwald, J., "Dynamics Workshop on Rotor Vibratory Loads," *52nd Annual Forum of the American Helicopter Socity,* June 1996.

[6]Caradonna, F. X., "Developments and Challenges in Rotorcraft Aerodynamics," *38th AIAA Aerospace Sciences Meeting,* AIAA 2000-0109, Jan 2000.

[7]Harris, F. D., "Articulated Rotor Blade Flapping Motion at Low Advance Ratio," *Journal of the American Helicopter Society,* Jan. 1972, pp. 41-48.

[8]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.

[9]Whitehouse, G. R., and Brown, R. E., "Helicopter Rotor Response to Wake Encounters in Ground Effect," *59th Annual Forum of the American Helicopter Socity,* Phoenix, Arizona, May 2003.

[10]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.

[11]Appel, A.,"An Efficient program for many-body simulation," *SIAM Journal on Scientific Computing,* Vol. 6, 1985, p. 85.

[12]Barnes, J., and Hut, P., "A Hierarchical $O(N \log N)$ Force-Calculation Algorithm," *Nature,* Vol. 324, 1986, p. 449.

[13]Greengard, L., and Rokhlin V., "A Fast Algorithm for Particle Simulations," *Journal of Computational Physics,* Vol. 116, 1995, pp. 69-78.

[14]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, 2001, pp. 879-907.