INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2007; **53**:1399–1422
Published online 19 June 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1240

A combined pFFT-multipole tree code, unsteady panel method with vortex particle wakes

David J. Willis[‡], Jaime Peraire[*, †] and Jacob K. White[§]

Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 01239, U.S.A.

SUMMARY

Potential flow solvers for 3-D aerodynamic flows are commonly used in industrial applications. Two main difficulties preventing the even more widespread use of these codes are the limitations on the number of discretization elements and the user expertise and effort required to specify the wake location. In the paper we present an automatic wake generation strategy for a potential flow solver, and accelerate the method using a pFFT-Fast Multipole Tree algorithm. The combined method can be used to simulate both steady and unsteady flows. The steady state solution is achieved by running an unsteady flow simulation until it reaches a steady state. Computation results are given to demonstrate that the method is fast enough to automatically simulate entire heaving and flapping wing crafts in under and hour on a desktop computer. Copyright © 2006 John Wiley & Sons, Ltd.

Received 26 October 2005; Revised 27 February 2006; Accepted 19 March 2006

KEY WORDS: Panel method; pFFT; multipole; vortex particle method

1. INTRODUCTION

Hess and Smith, while working at Douglas Aircraft company in the early 1960s, started work on what is now commonly referred to as the Aerodynamic Panel Method [1]. The panel method is a boundary element method (BEM) approach for solving the potential flow around aerodynamic bodies. Since Hess and Smith first developed the panel method, the aerospace industry and multiple research institutions have further investigated and advanced the approach [2–6]. Panel methods

[*]Correspondence to: Jaime Peraire, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 01239, U.S.A.
[†]E-mail: peraire@mit.edu
[‡]E-mail: djwillis@mit.edu
[§]E-mail: white@mit.edu

Contract/grant sponsor: Singapore–MIT Alliance
Contract/grant sponsor: Natural Sciences and Engineering Research Council of Canada
Contract/grant sponsor: National Sciences Foundation

continue to be widely used for initial design studies due to their ease of use and rapid solution times. The ease of use is a result of the surface only discretization. Although easy to use, most modern panel methods have several drawbacks, mostly resulting from the geometry discretization requirements, and the limit on the number of discrete elements allowed for a given simulation. In addition, most panel method implementations require significant user expertise and effort to determine and specify wake positions and wake–body intersections. In this paper we present a new simulation program, FastAero, which addresses the issues with panel methods. The approach is based on advances made in electrostatic simulations (precorrected FFT [7]), and in large n-body interaction problems (Fast Multipole Tree [8]). Similar acceleration approaches have been previously used in panel methods and vortex particle methods [9, 5].

The panel method which is presented in this paper implements a Green's Theorem Boundary Integral Equation formulation to determine the potential on the surface of the body. In addition, the trailing wake shed by the wing is represented using a vortex particle approach. The vortex particle approach allows the vorticity in the wake to advect automatically in the domain for both steady and unsteady flow simulations. A similar coupled potential flow vortex particle approach is presented in the work of Voutsinas *et al.* [10] with application to wind turbine analysis. A precorrected FFT (pFFT) accelerated iterative integral equation solver [7] is incorporated to determine the potential flow solution for the fixed body geometry. In addition to the pFFT, the simulation tool also uses a Fast Multipole Tree (FMT) algorithm [8, 11] to rapidly compute the velocity contribution from the time varying shedded wakes. The FMT algorithm is a variation on the Barnes–Hut tree code [11] and the Greengard Fast Multipole Method [8]. The combined potential flow-vortex particle wake approach is novel in that it allows a rapid automatic generation and evolution of the domain vorticity. The method is found to be particularly effective for unsteady applications involving variable geometries, such as flapping wings. In such cases, the proposed approach delivers fast and accurate solutions where more sophisticated higher fidelity models based on the Euler or Navier–Stokes equations are impractical.

In the first section we introduce the governing fluid dynamic equations used in FastAero for steady and unsteady flow simulations. The second section of this paper describes the Boundary Integral Equation (BIE) and the Boundary Element Method (BEM) for computing potential flow. Following this, algorithms such as the pFFT and the FMT are briefly presented to introduce these techniques for improving computational efficiency. Finally, results are presented to demonstrate the effectiveness of the approach applied to both steady and unsteady simulations.

2. THE GOVERNING EQUATIONS

The problem geometry is illustrated in Figure 1. The flows we will consider are high Reynolds Number incompressible flows, and we will assume potential flow is an adequate approximation. A point position in space at a given time, defined in the global reference frame is $\mathbf{R}(X, Y, Z, t)$. The global reference frame is fixed in space. The position of any point in a global reference frame is denoted by

$$\mathbf{R}_p = \mathbf{R}_G + \mathbf{r}_{Gp} \tag{1}$$

where $\mathbf{R}_G$ is the position of the centre of gravity of the body, and $\mathbf{r}_{Gp}$ is the position of point p relative to the centre of gravity. For simplicity, a body fixed reference frame is considered such that the position of points on the body relative to the centre of gravity are easily determined.

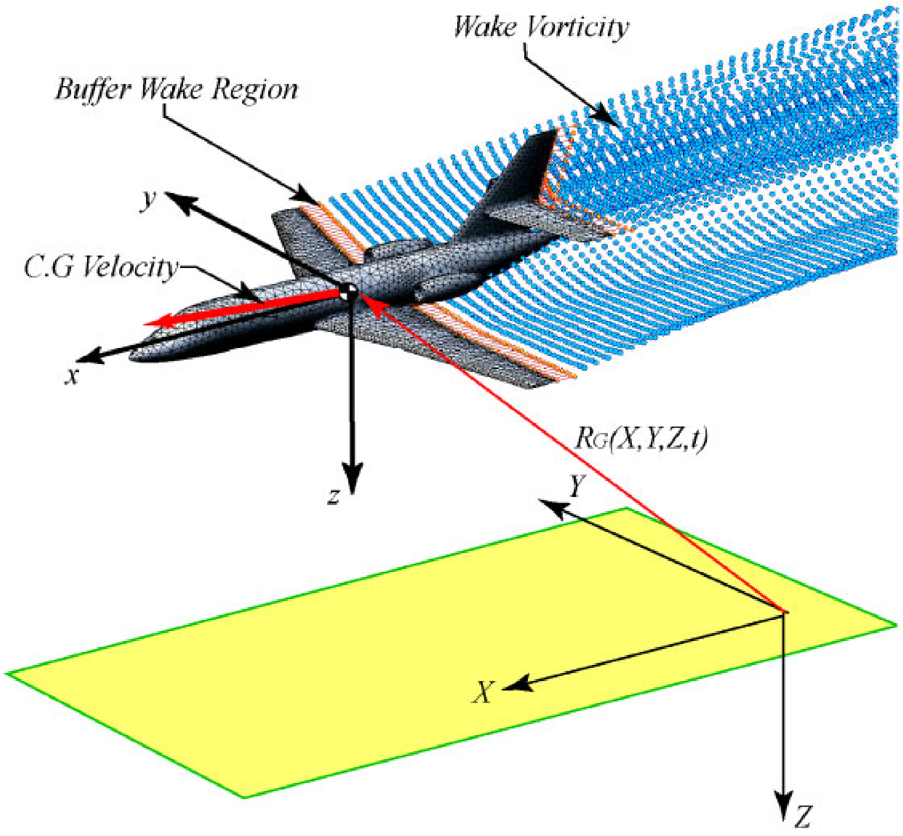

Figure 1. The domain of interest includes all fluid external to the aircraft surface. The particles trailing the wing section, the vertical tail and horizontal stabilizer represent regions in which vorticity exists due to the lifting surface trailing shear layer.

Therefore, a given point on the body will have a velocity with respect to the global reference frame given by

$$\mathbf{V}_p = \mathbf{V}_G + \mathbf{V}_{Gp} + (\boldsymbol{\Omega} \times \mathbf{r}_{Gp}) \tag{2}$$

where $\mathbf{V}_G$ represents the velocity of the centre of gravity in global coordinates, $\boldsymbol{\Omega}$ is the angular velocity, and $\mathbf{V}_{Gp}$ represents the relative motion of the surface due to deformation of the body (e.g. deflection of a control surface). For clarity, we denote body velocities by $\mathbf{V}$ and fluid velocities by $\mathbf{U}$.

In the paragraphs which follow, we present the governing equations and assumptions we make in the development of the flow solver.

2.1. Flow assumptions

The flow is assumed to be inviscid, incompressible and have constant density. Any vorticity in the domain is localized on the thin wake regions trailing the lifting surfaces. Otherwise, the flow is assumed to be irrotational. These assumptions greatly simplify the form of the governing equations.

2.2. Velocity definition

The fluid velocity, $\mathbf{U}(\mathbf{R}, t)$ at a given point in the domain is defined as the superposition of a scalar potential component, $\mathbf{U}_\phi(\mathbf{R}, t)$, and a vector potential component, $\mathbf{U}_\Psi(\mathbf{R}, t)$, using a Helmholtz decomposition [12]

$$\mathbf{U}(\mathbf{R}, t) = \mathbf{U}_\phi(\mathbf{R}, t) + \mathbf{U}_\Psi(\mathbf{R}, t) = \nabla\phi + \nabla \times \boldsymbol{\Psi} \tag{3}$$

The scalar potential component of the velocity is irrotational and any vorticity effects are captured in the vector potential component.

2.3. The continuity equation

The governing continuity equation for a constant density fluid is expressed in differential form as

$$\nabla \cdot \mathbf{U} = 0$$

Substituting the velocity potential relationships into the continuity equation, the resulting equation simplifies to

$$\nabla \cdot (\nabla\phi + \nabla \times \boldsymbol{\Psi}) = \nabla \cdot (\nabla\phi) = \nabla^2\phi = 0 \tag{4}$$

which is the Laplace's equation for the scalar potential.

2.4. The vorticity, velocity, and vector potential relationships

The vorticity in the domain, $\boldsymbol{\omega}(\mathbf{R}, t)$, is defined as the curl of the velocity

$$\nabla \times \mathbf{U} = \boldsymbol{\omega}$$

The velocity due to the *vector potential*, $\boldsymbol{\Psi}$, is

$$\nabla \times \boldsymbol{\Psi} = \mathbf{U}_\Psi$$

Substituting the vector potential velocity relationship into the definition of the vorticity results in:

$$\nabla^2\boldsymbol{\Psi} = -\boldsymbol{\omega}$$

which is a vector Poisson equation relating the vector potential to the vorticity.

The vorticity evolution equation is derived from the incompressible Euler equations

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla\mathbf{U} = -\frac{\nabla p}{\rho} \tag{5}$$

where ρ, is the fluid density, and p is the pressure. Taking the curl of Equation (5), the resulting equation for the vorticity evolution in the domain is [12]

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \mathbf{U} \cdot \nabla\boldsymbol{\omega} = \boldsymbol{\omega} \cdot \nabla\mathbf{U} \tag{6}$$

where the term $\boldsymbol{\omega} \cdot \nabla \mathbf{U}$ on the right-hand side represents the vorticity stretching (or how the strength and magnitude of the vorticity changes as it is exposed to velocity gradients in the flow field).

2.5. The boundary conditions

At any point on a solid surface in the domain, a no penetrating flux boundary condition is given by

$$\hat{n} \cdot \mathbf{U}(\mathbf{R}, t) = \hat{n} \cdot (\mathbf{V}_G + \mathbf{V}_{Gp} + \boldsymbol{\Omega} \times \mathbf{r}_{Gp})$$

where $\hat{n}$ is the outward unit normal vector on the body at a given point $\mathbf{R}$ on the body surface. In terms of the vector and scalar potentials, the boundary condition is

$$\hat{n} \cdot (\nabla \phi + \nabla \times \boldsymbol{\Psi}) = \hat{n} \cdot (\mathbf{V}_G + \mathbf{V}_{Gp} + \boldsymbol{\Omega} \times \mathbf{r}_{Gp}) \tag{7}$$

We note that due to the inviscid flow assumption, only the normal velocity boundary condition is applied at the body surface.

In the global reference frame, the fluid velocity at the farfield satisfies

$$\lim_{\mathbf{R} \to \infty} \mathbf{U}(\mathbf{R}, t) = 0$$

which implies that any perturbations of the velocity field due to body motion decay to zero infinitely far from the body.

2.6. The wing trailing edge Kutta condition

Since the flow is assumed to be inviscid, a Kutta condition is applied at all wing trailing edges. The Kutta condition prescribes a jump discontinuity in the surface potential across the geometric cusp which represents the trailing edge, thereby resulting in smooth and finite trailing edge velocities. In order to prescribe the streamwise vorticity release at the trailing edge, a linearized version of the pressure continuity at the trailing edge is used [13]

$$\phi_{\text{upper}} - \phi_{\text{lower}} = \Delta \phi_{\text{wake}} \tag{8}$$

Here, the subscripts *upper* and *lower* refer to points on the upper and lower surfaces of the trailing edge of the wing. This Kutta condition is illustrated in Figure 2.

For unsteady flows, a time dependent component of the Kutta condition is also enforced. This additional condition requires that any increase in bound vorticity on the wing be balanced by an equivalent increase in vorticity in the wake. This increased vorticity is oriented in the direction parallel to the trailing edge. The formal statement for this condition is that the combined change with respect to time of the wing bound circulation and the wake circulation add to zero

$$\left[\frac{\mathrm{d}\Gamma_{\text{span}}}{\mathrm{d}t}\right]_{\text{wing}} = -\left[\frac{\mathrm{d}\Gamma_{\text{span}}}{\mathrm{d}t}\right]_{\text{wake}} \tag{9}$$

where Γ represents the circulation strength of the wing and body. One can easily relate the circulation to the vorticity.

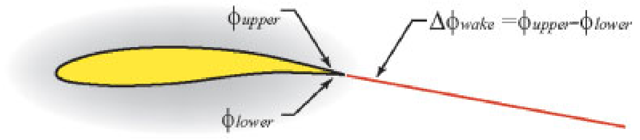

Figure 2. A figure illustrating the Kutta condition governing the streamwise vorticity release into the domain. The Kutta condition requires a potential jump across the wake corresponding to the difference between the upper and lower trailing edge potentials.

2.7. *The pressure–velocity relationship*

The Bernoulli equation is used to determine the forces and pressures on the body. Since a potential-vorticity approach is used, we briefly derive the applicable unsteady Bernoulli equation [14]. The incompressible Euler equations yield

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla \mathbf{U} = -\frac{\nabla p}{\rho} \tag{10}$$

If we consider those regions of the flow which have zero vorticity (all of space excluding the trailing vortex wake region), the resulting equation is

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{1}{2}\nabla|\mathbf{U}|^2 = -\frac{\nabla p}{\rho} \tag{11}$$

The definition of the velocity, given by Equation (3), can be substituted resulting in

$$\frac{\partial(\nabla\phi + \nabla\times\mathbf{\Psi})}{\partial t} + \frac{1}{2}\nabla|\nabla\phi + \nabla\times\mathbf{\Psi}|^2 = -\frac{\nabla p}{\rho} \tag{12}$$

Collecting like terms, and re-arranging, we obtain

$$\frac{\partial(\nabla\times\mathbf{\Psi})}{\partial t} + \nabla\frac{\partial\phi}{\partial t} + \frac{1}{2}\nabla|\nabla\phi + \nabla\times\mathbf{\Psi}|^2 + \nabla\left(\frac{p}{\rho}\right) = 0 \tag{13}$$

Integrating Equation (13) along the streamline from surface point x_1, to a farfield reference point at ∞; where at $\mathbf{R} = \infty$, $\mathbf{U}_{t=0\to t_\infty} = 0$, and $p = p_\infty$; results in

$$\int_\infty^{p_{x_1}} \frac{\partial(\nabla\times\mathbf{\Psi})}{\partial t}\,\mathrm{d}C + \left(\frac{\partial\phi}{\partial t}_{x_1} + \frac{1}{2}|\nabla\phi + \nabla\times\mathbf{\Psi}|^2_{x_1}\right) = \frac{p_\infty - p_{x_1}}{\rho} \tag{14}$$

Furthermore, we note that the term $\partial\phi/\partial t$ is defined in an Eulerian reference frame. We can compute the change in potential with respect to time for a point on the body surface by converting to a body Lagrangian reference frame

$$\left.\frac{\partial\phi}{\partial t}\right|_{\text{Eulerian}} = \left.\frac{\partial\phi}{\partial t}\right|_{\text{body}} - (\mathbf{V}_G + \mathbf{V}_{Gp} + (\mathbf{\Omega}\times\mathbf{r}_{Gp}))\cdot\nabla\phi \tag{15}$$

The overall unsteady Bernoulli equation is therefore

$$\int_{\infty}^{p_{x_1}} \frac{\partial(\nabla\times\boldsymbol{\Psi})}{\partial t}\,\mathrm{d}C + \left.\frac{\partial\phi}{\partial t}\right|_{\text{body}} - (\mathbf{V}_G + \mathbf{V}_{Gp} + (\boldsymbol{\Omega}\times\mathbf{r}_{Gp}))\cdot\nabla\phi + \frac{1}{2}|\nabla\phi + \nabla\times\boldsymbol{\Psi}|^2 = \frac{p_\infty - p_{x_1}}{\rho} \tag{16}$$

From this pressure–velocity relationship, the forces and moments can be computed by integration.

3. THE BOUNDARY INTEGRAL EQUATIONS FOR THE VECTOR AND SCALAR POTENTIALS

In the section which follows, the implementation of the above theory is discussed. First, the boundary integral equation formulations are presented followed by some more specific implementation details.

3.1. The Laplace's equation in integral equation form

In the method presented in this paper, integral equations are used to determine the potential flow solution as well as the vorticity induced velocity. In this section, the potential flow boundary integral equation solution is outlined. The integral equation representation of Laplace's equation for the potential at any point, $\mathbf{r}$, in the domain is

$$\phi(\mathbf{r}) = \frac{1}{4\pi}\iint_{S'_\mathrm{b}} \phi(\mathbf{r}')\frac{\partial}{\partial n_{\mathbf{r}'}}\frac{1}{\|\mathbf{r}-\mathbf{r}'\|}\,\mathrm{d}S'_\mathrm{b} + \frac{1}{4\pi}\iint_{S'_\mathrm{w}} \Delta\phi_{\text{wake}}(\mathbf{r}')\frac{\partial}{\partial n_{\mathbf{r}'}}\frac{1}{\|\mathbf{r}-\mathbf{r}'\|}\,\mathrm{d}S'_\mathrm{w} - \frac{1}{4\pi}\iint_{S'_\mathrm{b}} \frac{\partial\phi}{\partial n_{\mathbf{r}'}}(\mathbf{r}')\frac{1}{\|\mathbf{r}-\mathbf{r}'\|}\,\mathrm{d}S'_\mathrm{b}$$

In this equation, S_b represents the surface of the body, while S_w represents the portion of the wake sheet defined using dipole sheets (the buffer wake region to be described below). The no-flux boundary condition applied to the potential flow integral equation is

$$\frac{\partial\phi}{\partial n_{\mathbf{r}'}}(\mathbf{r}') = \hat{n}_{\mathbf{r}'}\cdot(\mathbf{V}_G + \mathbf{V}_{Gp} + (\boldsymbol{\Omega}\times\mathbf{r}_{Gp}) - \nabla\times\Psi) \tag{17}$$

From this, an integral equation for the unknown potential $\phi(\mathbf{r})$ can be expressed as

$$\phi(\mathbf{r}) = \frac{1}{4\pi}\iint_{S'_\mathrm{b}} \phi\frac{\partial}{\partial n}\frac{1}{\|\mathbf{r}-\mathbf{r}'\|}\,\mathrm{d}S'_\mathrm{b} + \frac{1}{4\pi}\iint_{S'_\mathrm{w}} \Delta\phi_{\text{wake}}\frac{\partial}{\partial n}\frac{1}{\|\mathbf{r}-\mathbf{r}'\|}\,\mathrm{d}S'_\mathrm{w} - \frac{1}{4\pi}\iint_{S'_\mathrm{b}} [\hat{n}\cdot(\mathbf{V}_G + \mathbf{V}_{Gp} + (\boldsymbol{\Omega}\times\mathbf{r}_{Gp}) - \mathbf{U}_\Psi)]\frac{1}{\|\mathbf{r}-\mathbf{r}'\|}\,\mathrm{d}S'_\mathrm{b} \tag{18}$$

The farfield boundary conditions on the potential will be satisfied [15]. Once the potential is known, the flow field for the entire domain is also known. At each time step, it is necessary to solve the boundary integral equation given in Equation (18).

The integral equation for computing the velocity in the domain, away from the body surface, due to the body and wakes is determined by taking the gradient of the integral equation for the potential

$$\mathbf{U}_\phi(\mathbf{r}) = \frac{1}{4\pi}\nabla \int\!\!\int_{S'_\mathrm{b}} \phi \left(\frac{\partial}{\partial n}\frac{1}{\|\mathbf{r}-\mathbf{r}'\|}\right) \mathrm{d}S'_\mathrm{b} + \frac{1}{4\pi}\nabla \int\!\!\int_{S'_\mathrm{w}} \Delta\phi_{\mathrm{wake}} \left(\frac{\partial}{\partial n}\frac{1}{\|\mathbf{r}-\mathbf{r}'\|}\right) \mathrm{d}S'_\mathrm{w} - \frac{1}{4\pi}\nabla \int\!\!\int_{S'_\mathrm{b}} \frac{\partial\phi}{\partial n} \left(\frac{1}{\|\mathbf{r}-\mathbf{r}'\|}\right) \mathrm{d}S'_\mathrm{b} + \mathbf{U}_\Psi \tag{19}$$

where, again, $\partial\phi/\partial n$ is known from the boundary conditions. In order to determine the tangential surface flow velocity due to the potential, the gradient of the body surface potential is evaluated.

As in most boundary element methods, the surface of the body is discretized into body panels or elements, and the surface singularity strengths (single and double layers) are represented using basis functions. In our implementation, a surface grid of triangles and quadrilaterals is used to represent the overall geometry, with ϕ and the boundary condition $\partial\phi/\partial n$ on the surface represented using constant basis functions. The integral equation is satisfied at specified *centroidal collocation points* [16], generating a linear system of equations

$$[A][\boldsymbol{\phi}] = [B]\left[\frac{\partial\boldsymbol{\phi}}{\partial n}\right] \tag{20}$$

The matrices $[A]$ and $[B]$ represent the discretized double layer and single layer integrals, respectively. Both of these matrices are dense because the normal velocity at a point contributes to the potential globally. The individual entries in the $[A]$ and $[B]$ matrices are integrals of the single and double layer Green's functions over triangular or quadrilateral elements. Analytical expressions for these integrals are presented in Hess and Smith [17] as well as in Newman [18].

Although, we give here a description of the method for constant basis functions, the use of higher order approximations is also possible [19].

3.2. The integral equation relationships for the vorticity in the domain

The vector Poisson equation governs the vector velocity potential. In integral form, the vector potential due to the vorticity in the domain is

$$\boldsymbol{\Psi}(\mathbf{R},t) = \frac{1}{4\pi}\int\!\!\int\!\!\int_{V'} \frac{\boldsymbol{\omega}}{\|\mathbf{R}-\mathbf{R}'\|}\,\mathrm{d}V'$$

The vorticity induced velocity is determined by taking the curl of the above equation

$$\nabla\times\boldsymbol{\Psi}(\mathbf{R},t) = \nabla\times\frac{1}{4\pi}\int\!\!\int\!\!\int_{V'} \frac{\boldsymbol{\omega}}{\|\mathbf{R}-\mathbf{R}'\|}\,\mathrm{d}V' \tag{21}$$

Similarly, the gradient of the velocity term used for the vorticity stretching in the vorticity evolution equation is determined by taking the gradient of the above relationship.

3.3. The representation of the vorticity in the domain

In order to model the vorticity in the domain a vortex particle method is used. The domain vorticity is represented as the summation over all of the discrete vortex particles in the domain, which is written as [20]

$$\boldsymbol{\omega}(\mathbf{R},t)=\sum_p \boldsymbol{\omega}_p(t)\mathrm{vol}_p\delta(\mathbf{R}-\mathbf{R}_p(t))=\sum_p \boldsymbol{\alpha}_p(t)\delta(\mathbf{R}-\mathbf{R}_p(t))$$

where $\boldsymbol{\omega}_p(t)\mathrm{vol}_p$ is written as $\boldsymbol{\alpha}_p(t)$. Here, vol_p refers to the volume of the fluid domain which is represented by the vortex particle.

For a vortex particle representation of the vorticity, the discrete vector potential can be determined as

$$\boldsymbol{\Psi}_p(\mathbf{R},t)=\frac{1}{4\pi}\sum_p \boldsymbol{\alpha}(\mathbf{R},t)\frac{1}{\|\mathbf{R}-\mathbf{R}_p(t)\|}$$

The vorticity induced velocity can be computed from

$$\nabla\times\boldsymbol{\Psi}_p(\mathbf{R},t)=\frac{1}{4\pi}\sum_p \nabla\frac{1}{\|\mathbf{R}-\mathbf{R}_p(t)\|}\times\boldsymbol{\alpha}(\mathbf{R},t) \tag{22}$$

Similarly, the gradient of the velocity term used for the vorticity stretching in the vorticity evolution equation is

$$\nabla(\nabla\times\boldsymbol{\Psi}(\mathbf{R}))=\frac{1}{4\pi}\sum_p \nabla\left(\nabla\frac{1}{\|\mathbf{R}-\mathbf{R}_p(t)\|}\times\boldsymbol{\alpha}(\mathbf{R},t)\right) \tag{23}$$

We use a Lagrangian reference frame for the evolution of the vorticity, such that the position $\mathbf{R}_p(t)$ of a discrete vortex particle at any given time is governed by

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{R}_p(t)=\mathbf{U}_p(\mathbf{R}(t),t) \tag{24}$$

The evolution of the vortex particle strength as it travels through the domain can be represented as

$$\frac{\mathrm{D}\boldsymbol{\alpha}_p(t)}{\mathrm{D}t}=\boldsymbol{\alpha}_p(t)\cdot\nabla\mathbf{U}_p(\mathbf{R}(t),t) \tag{25}$$

Each of the vortex particles, or vortons, has an associated core in order to mimic the physical vortex core as well as to reduce the numerical instability of the vortex interactions. More information about vortex particle core functions and vortex methods in general can be found in Reference [20]. The evaluation of the vorticity induced vector potential can be represented as a matrix vector product

$$[C][\boldsymbol{\alpha}]=\boldsymbol{\Psi}_p$$

Here the vorticity is known and a single matrix vector product results in the vector potential.

3.3.1. Discrete form of the vorticity evolution equation. The evolution of vorticity is computed by discretizing the ODEs governing the vorticity evolution and computing the vortex particle position at time, $t+1$. A simple approach to this would be to use a forward Euler equation

$$\mathbf{R}(t+1)=\mathbf{R}(t)+\mathbf{U}_p(\mathbf{R}(t),t)\Delta t \tag{26}$$

and then update the strength of the vortex as

$$\boldsymbol{\alpha}_p(t+1)=\boldsymbol{\alpha}_p(t)+\boldsymbol{\alpha}_p(t)\cdot\nabla\mathbf{U}_p(\mathbf{R}(t),t)\Delta t \tag{27}$$

It should be noted that the use of a higher order symplectic method [21], such as a Runge–Kutta scheme will be beneficial in increasing solution fidelity.

4. FASTAERO WAKE REPRESENTATION

A significant reduction in user setup effort in FastAero compared with traditional potential flow methods is a direct result of the wake model adopted. Below we discuss the finer details of the FastAero wake model and its interaction with the potential flow solution. More specifically, we describe, the conversion from a dipole wake to a vortex wake.

4.1. The general wake description

The wake model used in FastAero is composed of two parts, a dipole buffer wake sheet, and a vortex particle wake. Both the buffer wake sheet and the vortex particle wake are used to represent the wing trailing vorticity; however, the means by which the representation is accomplished is different. The buffer wake sheet and the vortex particles are shown in Figure 3.

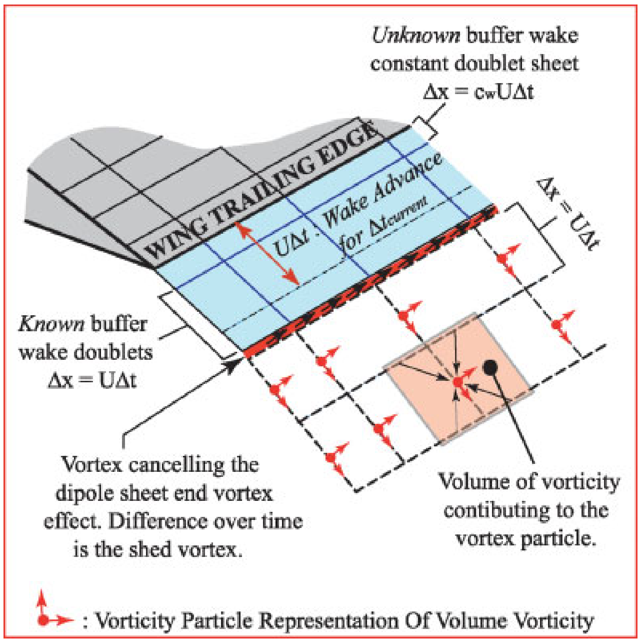

Figure 3. A description of the unknown and known buffer wake regions at the trailing edge of a wing. Note also the conversion of the line vortices resulting from the constant strength dipoles, into point vortices.

The buffer wake sheet is a short dipole/doublet sheet trailing all lifting surfaces. In the spanwise direction, the buffer wake is comprised of the same amount of panels as the wing trailing edge. In the streamwise direction the buffer wake sheet is comprised of two panels. The closest panel to the lifting surface trailing edge has a length of $c\mathbf{V}_G\Delta t$ and a strength which is to be determined by the Kutta condition at the current time step, while the second streamwise panel has a length of $\mathbf{V}_G\Delta t$ and a strength corresponding to the previous timestep trailing edge potential jump. The constant c is traditionally in the range of 0.3–0.5 (where $c = 0.5$ results in the vorticity at the new time t being added to the domain approximately at the midpoint of the distance travelled by a fluid particle during that timestep). The buffer wake sheet is essential to satisfy the potential jump in the wake at the wing trailing edge imposed by the potential Kutta condition. If a different form of the Kutta condition was used, the requirements for the buffer sheet may change or even be eliminated.

The vortex particle wakes are used to represent the wake vorticity outside the buffer region. At each time step, the known buffer wake sheet is converted into the equivalent vortex particle representation which is advected using Equations (26) and (27) above.

4.2. Conversion of the dipole wake sheet to vortex particles

A standard result in potential flow methods is that the change in dipole strength along a surface in a given direction is equivalent to vorticity oriented in the surface tangential direction normal to the dipole gradient [14]. In the case of constant dipole panels, the vortex analogue is a vortex ring around the perimeter of the given panel. Hence, the strength of the vortex line segment between two adjacent constant strength dipole panels is merely the difference in their strengths. This result is helpful, since the dipole buffer wake needs to be converted into an equivalent vortex representation.

The buffer wake is converted into vortex particles at each time step. This occurs in a two step process. First, vorticity in the spanwise direction is computed by determining the difference between the previous wake dipole strength and the current dipole strength. The vorticity in the streamwise direction is also computed in a similar manner, where differences in spanwise dipole strength are taken. In the second step, the downstream equivalent vortex on the dipole sheet must be cancelled out by an equal yet opposite vortex. This process is illustrated in Figure 3. Finally, the vortex particles are created by collapsing the line vortices surrounding the constant dipole, into a vortex particle.

4.3. Handling wing–body intersections in FastAero

Most aerodynamic simulations involve the intersection of a lifting surface with a body or fuselage. Due to the nature of the FastAero wake model special treatment is required in these cases. If a direct conversion of dipole sheets into vortices is performed, there would be a strong vortex shed at the wake–body intersection. This strong vortex streaming down the fuselage is non-physical and exists only to account for the required total potential jump in the wake. In the FastAero, we use an alternative method to account for the potential jump. Instead of streaming the strong vortex along the side of the body, the vortex is passed into the body or fuselage and down the centreline of the body (in the case of a symmetric wing–body problem the through body vorticity appears as a vortex in the spanwise direction only). By passing the vortex through the body, the vortex wake representation of the wake forms a complete closed loop, which is equivalent to a dipole sheet. In effect, the vortex passed through the body represents the body bound vorticity, while the vorticity in the wake represents the shed vorticity. By examining the wake as an equivalent dipole, the

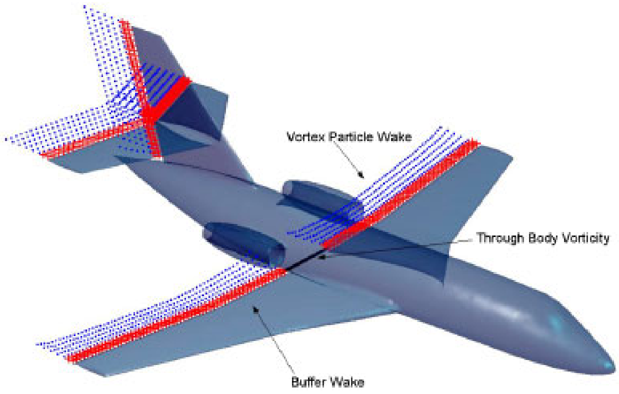

Figure 4. A description of the FastAero wakes. The figure shows the vortex particle wakes as well as the dipole buffer wake and the through body vortex for a symmetric configuration.

required jump in total potential exists due to the closure of the wake vorticity rings. Furthermore, the potential for which the integral equation is solved in FastAero, is continuous along the body where the equivalent vortex wake exists, and as such, there is no need to introduce a cut in the fuselage to ensure that the wake aligns with the body or that the wake and body geometries match. Details are presented in Reference [22].

In order to clarify the wake body problem, Figure 4 is presented. In this figure, the FastAero wake model for wing–body problems is depicted.

4.4. *Summary of solution steps*

In order to simulate a body shedding wakes, it is necessary to compute velocities and use those velocities to advect the wake. In the FastAero program this is accomplished by using a time stepping procedure. The following steps occur during each time step in the solution process:

Step 1: Solve the potential flow equation (18), to determine the unknown potential values on the surface of the body and the potential jump in the wake for the current timestep. The value of $\partial\phi/\partial n$ is computed for the current body translational and rotational velocities. The current domain vorticity influence, $\mathbf{U}_\Psi$, is also calculated according to Equation (17). Included in this flow solution is the determination of the potential jump across unknown buffer wake (Figure 3), which enforces the potential Kutta condition (Equation (8), and Figure 2).

Step 2: Determine the strength of the vorticity that needs to be released into domain. This new vorticity is the converted vorticity from the previous timestep known buffer wake layer (see Figure 3). The time dependent spanwise vorticity is determined after converting the change in strength of the buffer sheet from time t to a time $t+1$ to a line vortex according to Equation (9).

Step 3: Determine the velocity and gradient of the velocity influence from the body onto the vortex particles in the wake using the boundary integral equations (19). This involves evaluating the gradient of the potential flow integral equation at each of the vortex particle positions. This is a single matrix vector product operation.

Step 4: Determine the velocity and gradient of the velocity influence of each of the wake particles on each of the other wake particles using the integral equation for the vorticity–velocity relationship (Equations (22) and (23)). This is results in a matrix vector product evaluation.

Step 4(a): If necessary, compute the pressures and forces acting on the body, prior to updating the position and strength of the wake. This is an application of the Bernoulli equation (Equation (16)), to determine the pressure from the velocity.

Step 5: For each vortex particle in the wake, update the particle position and vortex particle strength using the Lagrangian vortex evolution equations (24) and (25). This involves determining the new position and strength by solving the ODEs governing the evolution of the vorticity.

Step 6: Compute the wake to body influence based on the new wake vorton positions and strengths. This is done by evaluating the matrix vector product represented by the Poisson equation governing the vorticity–velocity relationship (Equation (22)).

Step 7: Start over at (1) unless the iteration stopping condition has been reached.

4.5. The farfield approximation model

Once vortex particles are sufficiently far away from the body, they are lumped into multipole expansion representations, and treated as simple lumped vortex multipole particles. These multipole representations have no inter-particle influence in the farfield; however, they do have influence on the near field vortices and the body. Since these multipole representations have no inter-particle influence, they simply convect with the freestream flow.

5. ACCELERATING THE FLOW SOLVER

The solution time and memory for the Boundary Element Method—Vortex Particle approach is significant, especially when many surface elements and domain point vortices are used. In a typical medium resolution simulation, the body may have 5000–10 000 panels and there may be 10 000–15 000 wing trailing vortex particles. In addition to the large number of elements, the solution must be computed for each timestep. In a typical simulation, we may require hundreds if not thousands of time steps. For standard or direct iterative methods, as the number of unknowns increase, the complexity of the solution increases with $O(n^3)$ or $O(n^2)$ depending on solution method (here n represents the number of unknowns, elements and/or particles). It is possible to use fast methods to reduce the solution complexity to $O(n)$ or at least $O(n\log(n))$. We briefly present the methods used for accelerating the potential flow solution and the velocity–vorticity evaluation.

5.1. The pFFT potential flow solver

In order to solve the discretized potential flow equation (Equation (20) representing the boundary integral equation in Equation (18)) we implement a GMRES [23] iterative solver. As the matrix A is dense, the main computational cost in the GMRES algorithm is computing the matrix vector products (MVP), each of which costs $O(n^2)$ operations. The pFFT algorithm can be used to compute dense matrix vector products associated with the single and double layer singularities in $O(n\log(n))$ time. The pFFT uses a Fourier domain multiplication to efficiently compute the convolution products in the integral equation, and was originally developed by Philips and White [7], based on approaches developed by Hockney and Eastwood [24]. Our implementation makes use of the pFFT $++$ code developed by Zhu *et al*. [25].

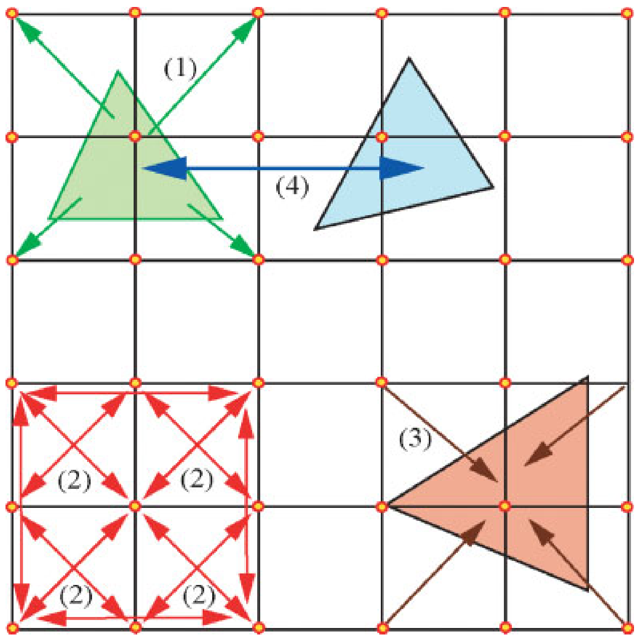

Figure 5. A schematic outlining the pFFT algorithm. The schematic demonstrates: (1) the projection of the panel singularity strength onto the FFT grid; (2) the convolution operation becomes a multiplication after being transformed to the Fourier domain, to give the grid potential; (3) after an inverse FFT, the grid potential is interpolated back onto the panels/evaluation points; and (4) for nearby interactions the solution is precorrected by subtracting the local grid interaction and adding the direct interactions. Overall the pFFT algorithm computes an MVP in $O(n \log(n))$ time.

During the initial pFFT setup an FFT grid is constructed in the domain surrounding the body. The steps in the algorithm are (see Figure 5):

Step 1: The singularity strength (panel charge) is projected onto the FFT grid. The projection operator acts locally, and hence, results in a sparse matrix $[P]$. The locality of the projection operator is illustrated in Figures 5 and 6.

Step 2: The grid strengths are convolved by multiplication via a transformation to the Fourier domain (using an FFT). Once the convolution is complete the result is transformed back to the physical domain using an inverse FFT, giving the grid potentials.

Step 3: The grid potentials are interpolated back to the panels or evaluation points. The computations involved are similar to the projection operation, and in the case of a Galerkin BEM approach, the interpolation is precisely the transpose of the projection. The interpolation results in a sparse matrix $[I]$.

Step 4: The nearby interactions are computed directly by subtracting the grid strengths and adding the direct element to element interactions. The setup of this operation is the most costly in the pFFT setup process due to both the grid correction and the local direct panel influence calculations. In this case, the direct interactions are represented by the sparse matrix $[D]_L$, and the correction is represented by a local subtraction equivalent to $[I_L H_L P_L]$, where the subscript L, refers to *local interactions*.

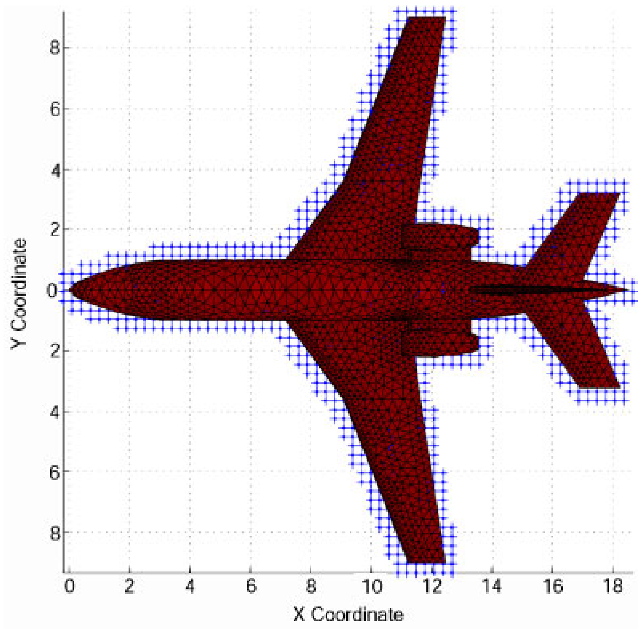

Figure 6. The pFFT grid overlay on the business jet. Points highlighted with '+' markers are those non-zero grid values after projection. The total FFT grid is a rectangular domain encapsulating the aircraft.

The resulting pFFT algorithm matrix vector product can be expressed in matrix form as

$$A\phi = [IHP + [D_L - I_L H_L P_L]]\phi$$

Once the initial overhead of setting up the matrices is complete, the matrix vector product is a multiplication and addition of sparse matrices. The resulting algorithm is one of the most efficient methods currently available for performing the matrix vector products.

5.2. *The FMT vorticity–velocity evaluation*

The Fast Multipole Tree algorithm is an octree based approach for reducing the complexity of the MVP evaluation from $O(n^2)$ to $O(n\log(n))$. The FMT algorithm is a variant on the Barnes-Hut [11] tree algorithm and Greengard's Fast Multipole Method [8]. The FMT constructs an octree structure complete with a multipole expansion of the source terms in each cell of the tree. The MVP is determined by evaluating the appropriate multipole influences at the evaluation point. In this section, we present the Fast Multipole Tree algorithm which is used. For further information about the FMT algorithm, as well as the formulae for the multipole moments, refer to Greengard's treatise of the subject [8].

5.2.1. Setting up the FMT algorithm. The following describes how the multipole tree is setup (see also Figure 7 for the octree structure):

Step 1: Construct a root cell enclosing all of the source points/panels. This cell is a level 1 cell, or root cell.

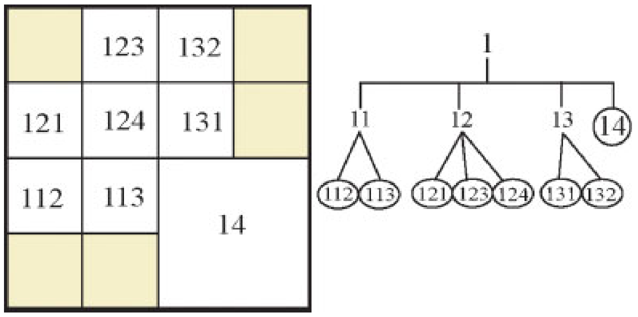

Figure 7. The general tree structure (note this is a 2-D quadtree, rather than a 3-D octree). The leaf cells are shown with circles in the tree structure. Empty cells without data are shaded. In the current simulation code, this tree structure is constructed using a top down approach starting from the root cell.

Step 2: Determine if the number of elements in the current cell is greater than the maximum number of elements permitted per leaf cell. This maximum number of elements per leaf cell is user specified and is typically between 10 and 15.

(a) If the number of elements in this cell is below the maximum allowed, the current cell is a leaf cell. Proceed by computing the leaf cell multipole moment representing the singularity strength distribution in cell. The leaf cell multipole moment is located at the centroid of the leaf cell and compactly represents the distribution of particles in the leaf cell. This leaf cell multipole expansion representation is determined by translating and adding all of the particle representations or monopoles in the cell. Once the leaf cell centroidal multipole representation is constructed, continue down the next branch of the tree.
(b) If the number of elements in this cell is larger than the maximum allowed, split the current cell (now called a parent cell) into 8 sub-cells called children cells. Determine which elements lie in each of the children, and construct new cells (only construct new cells for those children which have elements inside of them). Cycle through each of the children cells starting at step (2).

Step 3: Cycle through all cells, and propagate the leaf cell multipole up the tree using the multipole translation operators, fully populating the cell-centroid multipole strengths at each level of the tree. The children cell multipoles are translated and added to the parent cell. The end result is a multipole representation of the particles in each of the tree cells at each of the octree levels in the domain.

Having completed the above three steps, we have a sparse, yet sufficiently accurate approximation representing the particles in each of the cells. By appropriate selection of octree cells and multipole moments, the solution at an evaluation point can be determined.

5.2.2. Evaluation of the matrix vector product. In order to perform the FMT matrix vector product, we execute the following recursive algorithm starting at the root cell:

Step 1: Check the current cell to see if it is far enough away from the evaluation point to be considered a farfield interaction. This operation can be handled in many different ways (currently a measure of the cell centroid to evaluation point distance combined with a cell side length measure is used).

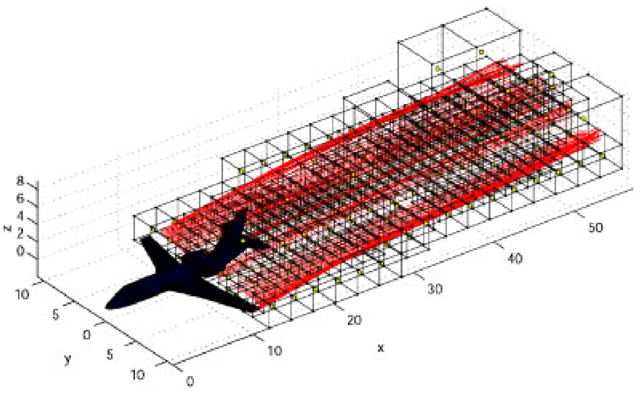

Figure 8. A plot showing the perspective view of the wake development behind a business jet, complete with the Octree representation. Note that in regions of higher wake vortex particle density, the octree is refined.

(a) If the cell is sufficiently far away, evaluate the potential, velocity and/or gradient of velocity at the evaluation point using the multipole expansion at the current cell centroid and add it to the current value at the evaluation point.
(b) If the cell is not a farfield cell, and is not a leaf cell, cycle through all of the populated children cells starting at step 1.
(c) If the cell is not a farfield cell, but is a leaf cell, evaluate the potential, velocity and/or gradient of the velocity using the direct calculations and add the result to the current evaluation point value. A direct computation is used for this portion of the flow, since the multipole expansion is not deemed accurate enough, and there is no further octree splitting to sufficiently separate the cell from the evaluation point.

An example of a practical application of the Fast Multipole Tree code for computing the vortex influences behind a business jet is shown in Figure 8. In order to attain more efficient computations for the MVP, there are several additional modifications which can be used when applying the FMT; however, the basic method stands as described.

5.3. Using the pFFT and FMT in the solution process

The choice of appropriate acceleration method for each step of the algorithm is described.

5.3.1. The potential flow solution. For the potential flow linear system solution, we use a pFFT approach embedded in a GMRES iterative solver. We use the pFFT due to the efficiency it affords in evaluating a matrix vector product once the pFFT matrices are set up. Since, for a given geometry, many MVPs are computed in the GMRES iterative solution process, it is well worth absorbing the setup overhead of the pFFT if the resulting MVP is sufficiently faster than the FMT counterpart.

5.3.2. The velocity influence computations. In order to perform the matrix vector products required for the velocity and velocity gradients, a Fast Multipole Tree algorithm is used. The FMT algorithm is used for (1) wake to wake interactions, (2) wake to body interactions, and (3) body to wake interactions. The FMT is used for the velocity and gradient interactions for the following reasons:

(1) Since the wake particles move, a complete system setup is required for each time step. Since the pFFT precorrection step is significantly more expensive than that for the FMT, the FMT is a better choice for this part of the simulation.
(2) The memory requirements are minimized when using the FMT to evaluate quantities in the wake region. Since the FFT grid encloses the entire domain, when a pFFT approach is used to evaluate quantities in the wake region, a sufficiently regular grid must be used. The FMT is more versatile, time efficient and memory efficient for evaluating quantities in the wake sufficiently far away from the aircraft body.

6. SAMPLE FLOW SIMULATION RESULTS

To demonstrate the applicability of the FastAero solution approach presented in this paper several examples are examined.

6.1. Example 1: the wing startup case

In the first example, the unsteady lift forces resulting from a step function startup on a rectangular wing are computed. The wing section is a NACA 0012 profile. The simulations were performed with several different aspect ratio rectangular wings. The wing discretization consisted of 5280 quadrilateral panels (44 elements around the chord, 120 elements in the span). The angle of attack for the simulation was 5°. The discrete timestep was set to $\Delta t = 0.1$, the velocity of fluid relative to the wing was $|\mathbf{U}| = 1$, with a wing chord of one unit.

The results of the simulations are compared with those presented for a vortex lattice approach in Katz and Plotkin [14] in Figure 9(a), while the wake evolution in the domain is presented in Figure 9(b).

From Figure 9(a) it can be seen that there is good agreement between the current approach and that presented in Reference [14] using a vortex lattice method. It should be noted that the current approach will have advantages over the vortex lattice approach when predicting the unsteady induced drag as well as other properties which are affected by the geometry of the surface and the thickness of the body.

6.2. Example 2: the heaving wing case

In this example, a heaving wing is examined, and compared with the Theodorsen 2-D, low amplitude thin airfoil approximation [26]. The Theodorsen approximation of the Z-Force for a 2-D heaving airfoil plate is [26]

$$F_z = -\pi\rho U c C(k)\frac{\mathrm{d}h}{\mathrm{d}t} - \pi\rho\frac{c^2}{4}\frac{\mathrm{d}^2 h}{\mathrm{d}t^2} \tag{28}$$

where ρ represents the fluid density, c the wing chord, U the fluid velocity relative to the body. The force deficiency factor $C(k)$, is given by Theordorsen [26]. The wing used for this simulation is the same NACA 0012 rectangular wing, with an aspect ratio of 12 used in the first example. The number of wake particles in each simulation approaches 20 000 by the time several periods of

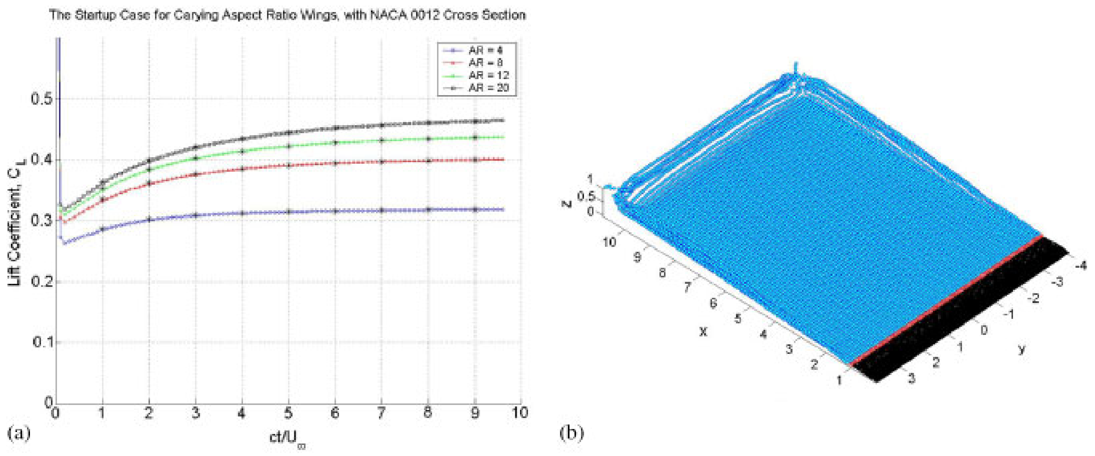

Figure 9. Flow about a wing undergoing a sudden startup: (a) C_L evolution with time. The results due to Katz and Plotkin [14] are shown using (*) markers for comparison; and (b) the actual wake evolution for an AR = 8 wing. Note the wing tip vortex roll-up and initial startup vortex rollup.

motion are resolved. The timestep for each case was chosen in order to ensure a sufficient number of simulation points were used to accurately resolve the solution.

The current simulation results are presented for both the overall 3-D vertical force as well as for the centreline, quasi-2-D vertical force. The centreline 2-D vertical force is computed by examining the pressures along the centreline of the wing and assuming that the behaviour is a good approximation to the 2-D theory. The heaving velocity in each of the following cases is given by:

$$\frac{\mathrm{d}h}{\mathrm{d}t}(t) = -0.05\,\sin(\omega t)$$

The reduced frequency of the oscillations is defined as

$$\omega_r = \frac{\omega c}{2U_{\text{ref}}}$$

where c is the wing reference chord, U_{ref} is the reference velocity (in this case the relative velocity between the wing and the fluid).

In Figures 10(a)–(d), results for the Z-force computation with time are presented. Note the phase differences between the position, velocity and Z-force for the various reduced frequencies. It can be seen that the Z-force is dominated by circulatory lift in low frequency applications; however, for higher frequencies the effect of fluid acceleration in the unsteady Bernoulli equation becomes significant, hence, the Z-Force tends to approach a phase corresponding to the position (or acceleration).

From Figures 10(a)–(d), it can be seen that there is good agreement between the predicted forces from Theodorsen's thin airfoil small amplitude theory [26], and the computed forces. There is a slight over prediction of the Z-force, which becomes more pronounced as the frequency rises. This is suspected to be due to finite aspect ratio effects in the computed solution for both the 2-D centreline approximation and the 3-D full wing simulation. Since the wing has a finite aspect ratio, the trailing edge dropped vorticity has less of an effect than an equivalent infinite line vortex. Hence

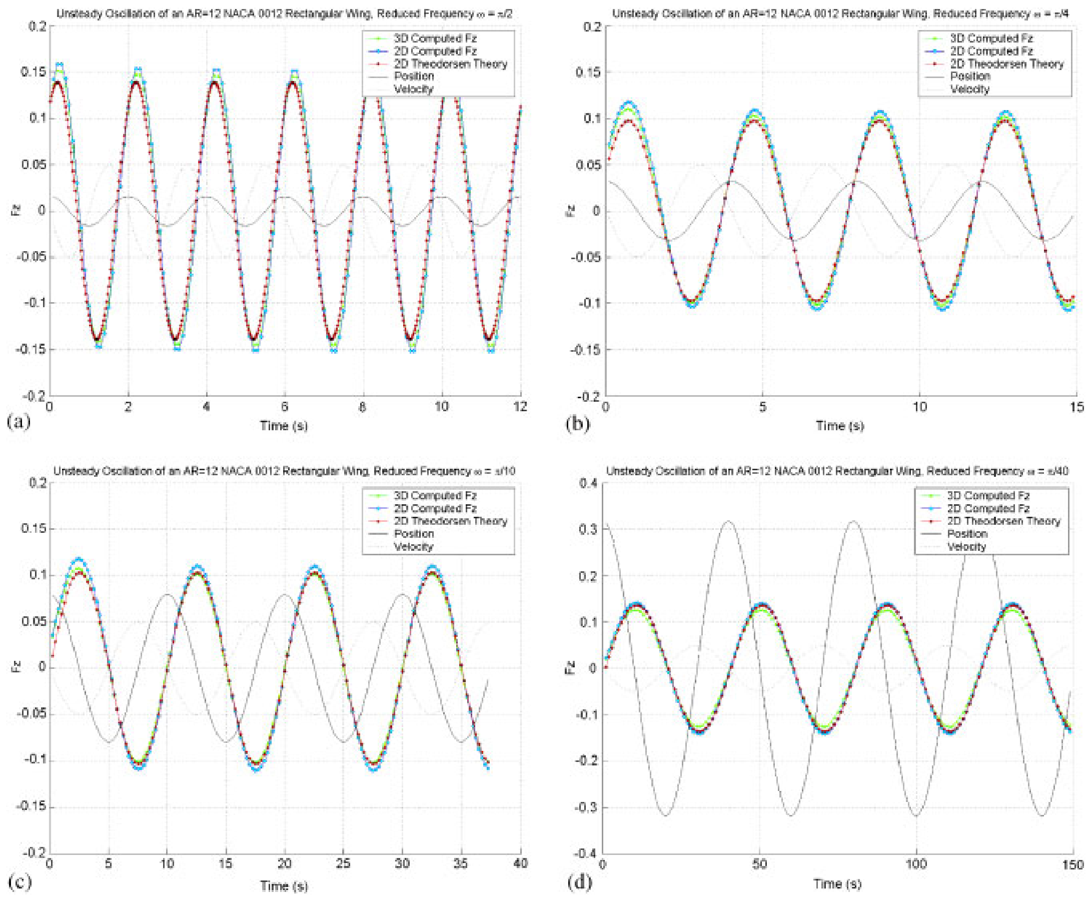

Figure 10. The results for the Z-Force component of a NACA 0012 aspect ratio 12 wing, undergoing heave oscillations at several different reduced frequencies. Note, as the frequency increases, the Z-force tends to trend with the phase of the position and acceleration, as expected, while the lower frequency oscillations demonstrate a Z-force phase close to the velocity, as expected: (a) reduced frequency $\omega_r = \pi/2$; (b) reduced frequency $\omega_r = \pi/4$; (c) reduced frequency $\omega_r = \pi/10$; and (d) reduced frequency $\omega_r = \pi/40$.

the Z-force reduction due to the newly released trailing edge vorticity is less than Theodorsen predicts, resulting in a slightly higher Z-force. It can be seen for low and moderate reduced frequencies as demonstrated in this example, that the computed solution matches well in both phase and amplitude with Theodorsen's theory. It should also be noted that the Kutta condition implemented in the FastAero code will lose applicability as the reduced frequency becomes larger.

6.3. *Example 3: complex geometry full configuration simulation*

In this example, we demonstrate the ability of the current approach to handle complex geometries, such as full aircraft configurations. In this simulation, a 8650 panel business jet is presented pictorially to demonstrate the wake roll-up. The number of vortex wake particles is up to 12 000

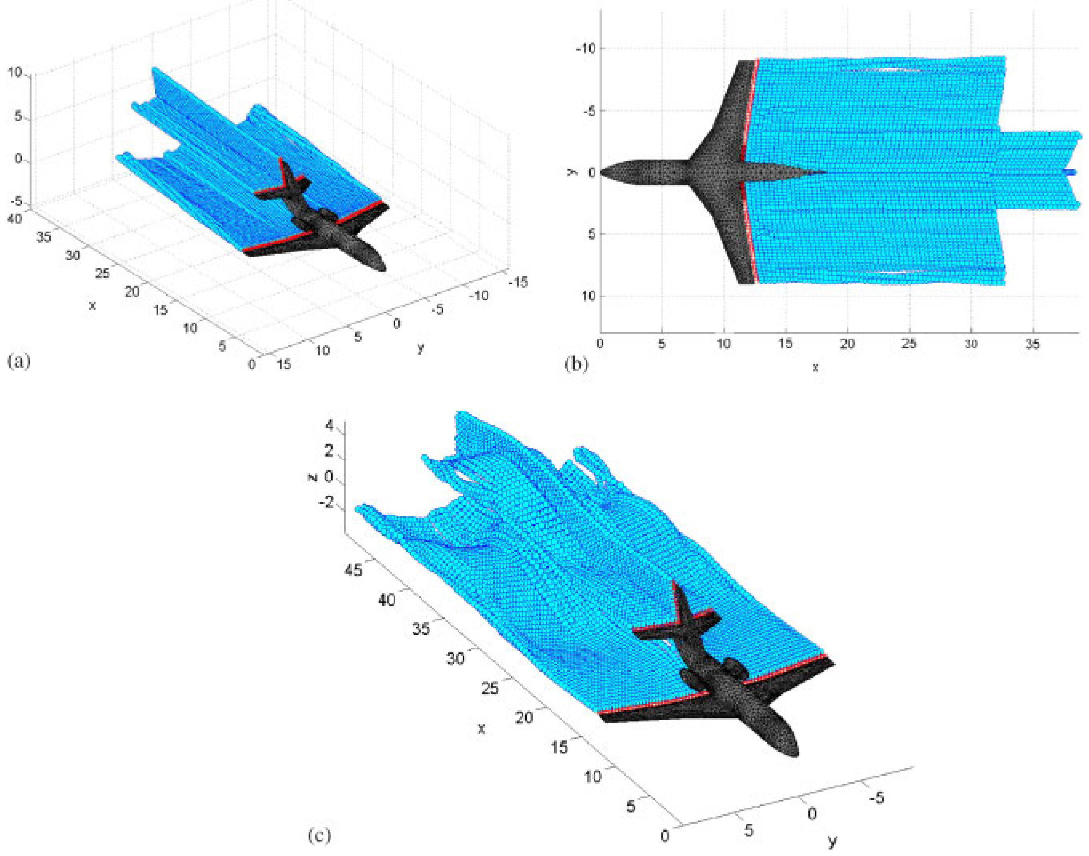

Figure 11. Several figures demonstrating the application of the FastAero code to complex configurations. Note: The full length of the wake used in the simulation is not plotted in figures (a) and (b) in order to focus on the wake in the region nearer the body: (a) illustration of the wake propagation at 5° angle of attack; (b) a figure demonstrating the fuselage-wake particle behaviour; and (c) a figure demonstrating a heaving motion for the aircraft. Note here that the wake is complicated in shape and as such, an automatic wake generation is necessary.

particles. In this simulation, the versatility and true flexibility of the FastAero solver is highlighted. Here again, the wake was generated automatically without user assistance.

The simulation of the business jet was performed for both a startup to steady state case, as well as a heaving case. The results are presented graphically in Figures 11(a)–(c). While examining the results it should be noted that other than prescribing the trailing edges that will shed vorticity, there is no user assistance in the wake generation procedure. These simulations demonstrate completely hands-off simulations from $t = 0$ to t_{final}. In order to validate the solution, two separate studies were performed. A convergence study was performed for the surface potential, yielding a rate of convergence proportional to the number of panels as expected. In addition, the lift force at the

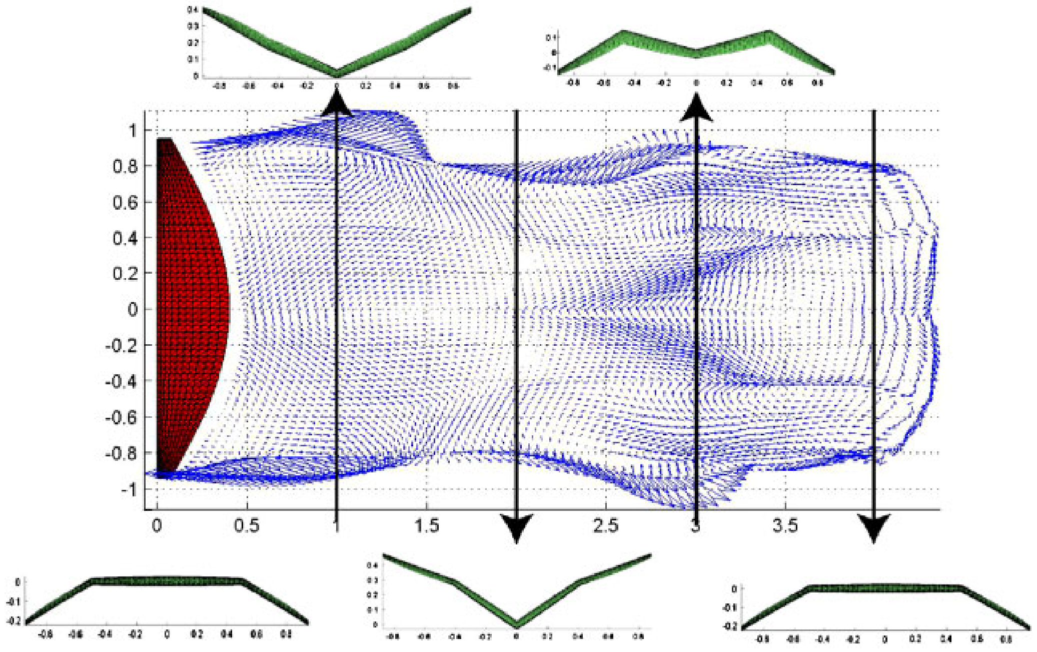

Figure 12. The results of simulating an articulated flapping wing using FastAero. The sections of wing between the joints are assumed to be rigid, however, may have varying washout depending on the stroke position.

highest grid resolution was compared with a Felissa Euler [27] solution at the same grid refinement. The results showed agreement to within 5 per cent. Considering the surface pressure computation in the FastAero code is found to be rather inaccurate for triangular constant strength panels, this result shows significant promise for future higher order versions of FastAero where the pressure computations will be significantly more accurate.

The overall simulation to steady state takes approximately 30–60 min using a moderate timestep resolution, on a 3 GHz desktop computer. It is expected that this time can be further reduced through more efficient implementation of the multipole tree algorithm, data structure and overall computer program. In addition, it should be mentioned that the simulation time depends significantly on the choice of timestep, multipole and pFFT accuracy, as well as the farfield multipole wake model implementation position.

6.4. *Example 4: qualitative articulated flapping wing analysis*

An example which demonstrates the strengths of the FastAero model is the case of flapping wing flight. In a preliminary application of FastAero to the flapping flight problem, the wake and vorticity evolution for a flapping articulated wing was simulated. The wing was prescribed to travel a length of two spans per flapping cycle. Figure 12 shows the result of the simulation.

FastAero is particularly well suited to this flapping flight analysis. Firstly, the unsteady wake model used in FastAero has the advantage of being automatic in both generation and evolution.

This is essential to the flapping flight problem. In addition to the automatic wake generation, the use of acceleration tools such as the pFFT and FMT methods permits the exploration of a system with many degrees of freedom. Although we present only one solution in this paper, it is conceivable that an optimal set of wing flapping parameters might be desired. In order to determine the optimum flapping frequency, with the various joint degrees of freedom many hundreds of simulations might be required. The FastAero tool is well suited due to the rapid solution times.

7. CONCLUSIONS

In this paper we presented a robust and efficient method for simulating the potential flow around arbitrary bodies that reduces user effort by automatically generating wake sheets. The method uses the precorrected FFT approach to accelerate the potential flow solution, while also implementing a fast multipole tree algorithm to compute the velocity and vorticity stretching influences in the domain. The approach has been demonstrated to be efficient and accurate for steady and unsteady design and analysis problems. The method presented is particularly well suited to unsteady lifting body flow simulations due to the use of the unique vortex particle approach. Computational results presented demonstrate that FastAero is fast enough to automatically simulate entire heaving and flapping wing crafts in under an hour on a desktop computer.

ACKNOWLEDGEMENTS

The authors would like to thank the Singapore–MIT Alliance (SMA), the National Sciences Foundation (NSF) and the Natural Science and Engineering Research Council of Canada (NSERC) for their financial support of this research. In addition, we would like to thank Mark Drela for his guidance and discussions, Zhenhai Zhu, and Ben Song for the use of their pFFT + + implementation, and Jaydeep Bardhan for his fruitful discussions of this work.

REFERENCES

1. Smith AMO. The panel method: its original development. *Applied Computational Aerodynamics*, vol. 125. Progress in Aeronautic Sciences. AIAA: New York, 1990.
2. Ashby DL, Dudley MR, Iguchi SK. Development and validation of an advanced low order panel method. *NASA TM 101024*, October 1988.
3. Maskew B. PROGRAM VSAERO: a computer program for calculating the non-linear aerodynamic characteristics of arbitrary configurations. *NASA CR-166476*, November 1982.
4. Rosen B, Laiosa JP. SPLASH nonlinear and unsteady free-surface analysis code for grand prix yacht design. *13th Chesapeake Sailing Yacht Symposium*, Annapolis, January 1997.
5. WAMIT 6.2. *WAMIT User Manual*, *Version 6.2*, *6.2PC and 6.1S*, *6.1S-PC*. Department of Ocean Engineering, M.I.T.
6. Quackenbush TR, Washspress DA, Boschitsch AH. Rotor aerodynamic loads computation using a constant vorticity contour free wake model. *Journal of Aircraft* 1995; **32**(5):911–920.
7. Philips JR, White JK. A precorrected—FFT method for electrostatic analysis of complicated 3-D structures. *IEEE Transactions On Computer-Aided Design of Integrated Circuits and Systems* 1997; **16**.
8. Greengard L, Rohklin V. A fast algorithm for particle simulations. *Journal of Computational Physics* 1987; **73**:325–348.
9. Koumoutsakos P. Multiscale flow simulations using particles. *Annual Review of Fluid Mechanics* 2005; **37**:457–487.
10. Voutsinas SG, Rados KG, Zervos A. On the effect of the rotor geometry on the formation and the development of its wake. *Journal of Wind Engineering and Industrial Aerodynamics* 1992; **39**:283–291.

11. Barnes J, Hut P. A hierarchical $O(N \log N)$ force calculation algorithm. *Nature* 1986; **324**:446–449.
12. Batchelor GK. *An Introduction to Fluid Dynamics*. Cambridge University Press: Cambridge, 2000.
13. Morino L. Steady, oscillatory and unsteady subsonic and supersonic aerodynamics—production version (SOUSSA). *NASA CR-157130*, 1980.
14. Katz J, Plotkin A. *Low-Speed Aerodynamics* (2nd edn). Cambridge Aerospace Series: Cambridge, 2001.
15. Kress R. *Linear Integral Equations*. Springer: New York, 1989.
16. Atkinson KE, Han W. *Theoretical Numerical Analysis*: *A Functional Analysis Framework*. Springer: New York, 2001.
17. Smith AMO, Hess JL. Calculation of potential flow about arbitrary bodies. *Progress in Aeronautic Sciences* 1966; **8**:1–138.
18. Newman JN. Distribution of sources and normal dipoles over a quadrilateral panel. *Journal of Engineering Mathematics* 1986; **20**:113–126.
19. Willis DJ, Peraire J, White JK. A quadratic basis function, quadratic geometry, high order panel method. *Paper AIAA-2006-1253*. *44th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA 2006-1253, 2006.
20. Winckelmansand GS, Leonard A. Contributions to vortex particle methods for the computation of three-dimensional incompressible unsteady flows. *Journal of Computational Physics* 1993; **109**:247–273.
21. Hairer E, Lubich C, Wanner G. *Geometric Numerical Integration*, Springer Series in Computational Mathematics. Springer: Berlin, 2004.
22. Willis DJ, Peraire J, White JK. Body piercing wakes for automatic wake generation in the potential panel method. Work in Progress.
23. Saad Y, Schultz M. GMRES: a generalized minimal residual algorithm for solving non-symmetric linear systems. *SIAM Journal on Scientific and Statistical Computing* 1986; **7**:856–869.
24. Hockney RW, Eastwood JW. *Computer Simulation Using Particles*. Taylor & Francis Inc.: Bristol, 1988.
25. Zhu Z, Song B, White JK. Algorithms in FastIMP: a fast wideband impedance extraction program for complicated 3D geometries. *Proceedings of IEEE/ACM Design Automation Conference*, Anaheim, California, June 2–6 2003.
26. Theodorsen T. General theory of aerodynamic instability and the mechanism of flutter. *NACA Report 496*, 1935.
27. Peraire J, Peiro J, Formaggia L, Morgan K, Zienkiewicz OC. Finite element Euler computations in three dimensions. *International Journal for Numerical Methods in Engineering* 1988; **26**:2135–2159.