43rd AIAA Aerospace Sciences Meeting and Exhibit, 10-13 January 2005, Reno. N.V

A Combined pFFT-Multipole Tree Code, Unsteady Panel Method with Vortex Particle Wakes

David J. Willis* Jaime Peraire †

Jacob K. White ‡

Massachusetts Institute Of Technology, 77 Massachusetts Ave., Cambridge, MA, 01239, USA

The two main difficulties in using potential flow solvers in aircraft design 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.

Nomenclature

X, Y, Z	Global reference frame coordinates
t	Time
$\vec{R}$	Position in global reference frame
$\vec{V}$	Body Velocity in global reference frame
$\vec{\Omega}$	Body Angular Velocity
ϕ	Scalar Velocity Potential
$\vec{\Psi}$	Vector Velocity Potential
$\vec{\omega}$	Vorticity
$\vec{U}$	Fluid Velocity
p	Static Pressure
ρ	Fluid Density
$\hat{n}$	Body Normal Vector
Γ	Circulation
S	Surface, Body Surface
V	Volume, Domain Volume
$[A]$	Double Layer Aerodynamic Influence Matrix
$[B]$	Single Layer Aerodynamic Influence Matrix
$[C]$	Vortex Particle Aerodynamic Influence Matrix
$[I]$	pFFT Interpolation Matrix
$[P]$	pFFT Projection Matrix
$[D]$	pFFT Direct Interactions Matrix
c	Airfoil Chord
F	Force
ω	Frequency
ω_r	Reduced Frequency

*Graduate Student, Department Of Aeronautics and Astronautics, MIT.

†Professor, Department Of Aeronautics and Astronautics, MIT.

‡Professor, Electrical Engineering and Computer Science, MIT.

n	Number of Points or Elements in the Domain
Subscript	
p	Point
G	Center of Gravity
b	Body
w	Wake
∞	Infinity, Farfield
L	Local, Local Interaction

I. Introduction

Hess and Smith, while working at Douglas Aircraft company in the early 1960's 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[2–4] . Panel methods 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 interference to determine the wake position and wake-body intersection. In this paper we present a simulation tool, FastAero, which attempts to address many of the drawbacks in using panel methods. The approach is based on advances made in electrostatic simulations (precorrected FFT[5]), and in large n-body interaction problems (Fast Multipole Tree[6]).

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 wing trailing shedded wake 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 precorrected FFT (pFFT) accelerated iterative integral equation solver[5] 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[6, 7] to rapidly compute the velocity contribution from the time varying shedded wakes. The FMT algorithm is a variation on the Barnes-Hut tree code[7] and the Greengard Fast Multipole Method.[6] The combined potential flow-vortex particle wake approach is novel in that it allows a rapid automatic generation and evolution of the domain vorticity, a distinct advancement in the use of panel methods.

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.

II. The Governing Equations

We present the external fluid domain in Fig. 1. The flows we will consider are high Reynold's Number incompressible flows, and potential flow assumptions will be an adequate approximation. A point position in space at a given time, defined in the global reference frame is $\vec{R}(X, Y, Z, t)$. The global reference frame is fixed in space. The position of any point in a global reference frame is denoted:

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

where, $\vec{R}_G$ is the position of the center of gravity of the body, and $\vec{r}_{Gp}$ is the position of point p relative to the center of gravity. For simplicity, a body fixed reference frame can be considered such that the relative position of points on the body to the center of gravity are easily determined. In this paper, we will assume that the analysis is performed in the global reference frame unless otherwise specified. As such, a given point

on the body will will have a velocity with respect to the global reference frame given by:

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

where $\vec{V}_G$ represents the velocity of the center of gravity in global coordinates, $\vec{\Omega}$ is the angular velocity, and $\vec{V}_{Gp}$ represents the relative motion of the surface due to deformation of the body (eg. deflection of a control surface). In this paper, we will consider only rigid bodies, hence $\vec{V}_{Gp} = 0$. For clarity purposes, we will denote body velocities by $\vec{V}$ and fluid velocities by $\vec{U}$.

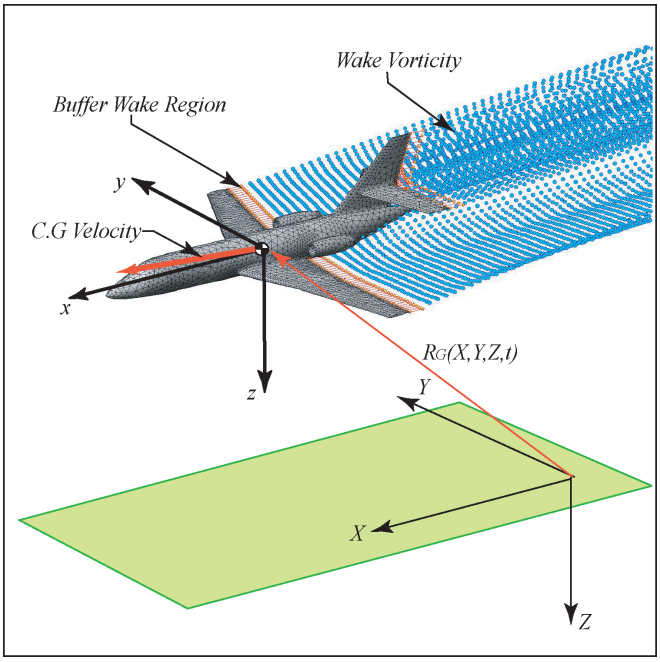


Figure 1. The domain of interest includes all fluid external to the aircraft surface. The Particles trailing the wing section, the vertical tail and horizontal stabilizer represent regions in which vorticity exists due to the lifting surface trailing shear layer.

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

A. Fluid Domain 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.

B. Velocity Definition

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

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

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

C. The Continuity Equation

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

$$\nabla \cdot \vec{U} = 0,$$

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

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

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

D. The Vorticity, Velocity, and Vector Potential Relationships

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

$$\nabla \times \vec{U} = \vec{\omega}$$

The velocity due to the *Vector Potential*, $\vec{\Psi}$, is:

$$\nabla \times \vec{\Psi} = \vec{U}_\Psi$$

Substituting the vector potential relationship into the definition of vorticity, and manipulating the equations, results in:

$$\nabla^2\vec{\Psi} = -\vec{\omega},$$

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

The vorticity evolution equation is derived from the incompressible Euler equations,

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

where, ρ, is the fluid density, and p is the pressure. Taking the curl of eqn. 5, the resulting equation for the vorticity evolution in the domain is,

$$\frac{\partial\vec{\omega}}{\partial t} + \vec{U} \cdot \nabla\vec{\omega} = \vec{\omega} \cdot \nabla\vec{U}$$

where the term $\vec{\omega} \cdot \nabla\vec{U}$ 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 fluid field).

E. The Boundary Conditions

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

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

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

$$\hat{n} \cdot (\nabla\phi + \nabla \times \vec{\Psi}) = \hat{n} \cdot (\vec{V}_G + \vec{\Omega} \times \vec{r}_{Gp}). \tag{6}$$

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:

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

which implies that any perturbations of the velocity field due to the presence of the body tend to zero at infinity.

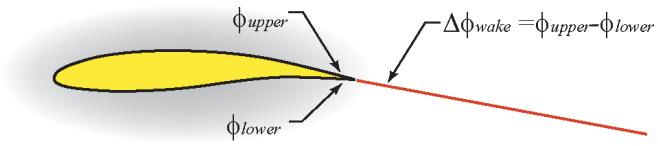


Figure 2. A figure illustrating the Kutta Condition governing the streamwise vorticity release into the domain. The Kutta condition requires a potential jump across the wake corresponding to the difference between the upper and lower trailing edge potentials.

F. 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,[8]

$$\phi_{upper} - \phi_{lower} = \Delta\phi_{wake}. \tag{7}$$

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 presented in Fig. 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 of the condition is that the combined change with respect to time of the wing bound circulation and and the wake circulation add to zero:

$$\left[\frac{d\Gamma_{span}}{dt}\right]_{wing} = -\left[\frac{d\Gamma_{span}}{dt}\right]_{wake}, \tag{8}$$

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

G. The Pressure–Velocity Relationship

The Bernoulli Equation is used to determine the forces and pressures on the body. Since this paper presents a potential-vorticity approach, we will briefly derive the applicable unsteady Bernoulli equation, starting from the incompressible Euler equations:

$$\frac{\partial\vec{U}}{\partial t} + \vec{U}\cdot\nabla\vec{U} = -\frac{\nabla p}{\rho}, \tag{9}$$

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:

$$\frac{\partial\vec{U}}{\partial t} + \frac{1}{2}\nabla|\vec{U}|^2 = -\frac{\nabla p}{\rho}. \tag{10}$$

The definition of the velocity, given by equation 3, can be substituted resulting in:

$$\frac{\partial(\nabla\phi) + \partial(\nabla\times\vec{\Psi})}{\partial t} + \frac{1}{2}\nabla|\nabla\phi + \nabla\times\vec{\Psi}|^2 = -\frac{\nabla p}{\rho}. \tag{11}$$

Collecting like terms, and re-arranging, we obtain:

$$\frac{\partial(\nabla\times\vec{\Psi})}{\partial t} + \nabla\frac{\partial\phi}{\partial t} + \frac{1}{2}\nabla|\nabla\phi + \nabla\times\vec{\Psi}|^2 + \nabla\left(\frac{p}{\rho}\right) = 0. \tag{12}$$

At this point we can integrate the equation between any two points on a given streamline in the domain. For reference frame simplicity, we choose a farfield reference point at ∞. At $\vec{R} = \infty$, $\vec{U}_{t=0 \to t_\infty} = 0$, and $p = p_\infty$.

$$\int_\infty^{p_{x_1}} \frac{\partial(\nabla \times \vec{\Psi})}{\partial t} dC + \left(\frac{\partial \phi}{\partial t}_{x_1} + \frac{1}{2} |\nabla\phi + \nabla \times \vec{\Psi}|^2_{x_1} \right) = \frac{p_\infty - p_{x_1}}{\rho}. \tag{13}$$

Furthermore, we note that the term $\frac{\partial \phi}{\partial t}$ 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:

$$\frac{\partial \phi}{\partial t}|_{eulerian} = \frac{\partial \phi}{\partial t}|_{body} - (\vec{V}_G + (\vec{\Omega} \times \vec{r}_{Gp})) \cdot \nabla\phi \tag{14}$$

The overall Unsteady Bernoulli equation is therefore:

$$\int_\infty^{p_{x1}} \frac{\partial(\nabla \times \vec{\Psi})}{\partial t} dC + \frac{\partial \phi}{\partial t}|_{body} - (\vec{V}_G + (\vec{\Omega} \times \vec{r}_{Gp})) \cdot \nabla\phi + \frac{1}{2} |\nabla\phi + \nabla \times \vec{\Psi}|^2 = \frac{p_\infty - p_{x_1}}{\rho}. \tag{15}$$

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

III. 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.

A. 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, $\vec{r}$ in the domain is:

$$\phi(\vec{r}) = \frac{1}{4\pi} \int\int_{S'_b} \phi \frac{\partial}{\partial n} \frac{1}{\|\vec{r} - \vec{r'}\|} dS'_b + \frac{1}{4\pi} \int\int_{S'_w} \Delta\phi_{wake} \frac{\partial}{\partial n} \frac{1}{\|\vec{r} - \vec{r'}\|} dS'_w - \frac{1}{4\pi} \int\int_{S'_b} \frac{\partial \phi}{\partial n} \frac{1}{\|\vec{r} - \vec{r'}\|} dS'_b.$$

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:

$$\frac{\partial \phi}{\partial n} = \hat{n} \cdot \left(\vec{V}_G + (\vec{\Omega} \times \vec{r}_{Gp}) - \nabla \times \Psi \right) \tag{16}$$

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

$$\phi(\vec{r}) = \frac{1}{4\pi} \int\int_{S'_b} \phi \frac{\partial}{\partial n} \frac{1}{\|\vec{r} - \vec{r'}\|} dS'_b + \frac{1}{4\pi} \int\int_{S'_w} \Delta\phi_{wake} \frac{\partial}{\partial n} \frac{1}{\|\vec{r} - \vec{r'}\|} dS'_w -$$
$$\frac{1}{4\pi} \int\int_{S'_b} \left[\hat{n} \cdot \left(\vec{V}_G + (\vec{\Omega} \times \vec{r}_{Gp}) - \vec{U}_\Psi \right) \right] \frac{1}{\|\vec{r} - \vec{r'}\|} dS'_b \tag{17}$$

The farfield boundary conditions on the potential are satisfied due to the decay of the Green's function, as well as the fact for a closed body, the net sum of the single layer strength is zero. Once the potential is known, the flow field for the entire domain is also known. At each time step in a discretized unsteady flow solution, it is necessary to solve the boundary integral equation given in eqn. 17.

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:

$$\vec{U}(\vec{r}) = \frac{1}{4\pi} \nabla \int\int_{S'_b} \phi \left(\frac{\partial}{\partial n} \frac{1}{\|\vec{r} - \vec{r'}\|} \right) dS'_b + \frac{1}{4\pi} \nabla \int\int_{S'_w} \Delta\phi_{wake} \left(\frac{\partial}{\partial n} \frac{1}{\|\vec{r} - \vec{r'}\|} \right) dS'_w -$$

$$\frac{1}{4\pi}\nabla \int\int_{S_b'} \frac{\partial \phi}{\partial n}\left(\frac{1}{\|\vec{r}-\vec{r'}\|}\right) dS_b' + \vec{U}_\Psi, \tag{18}$$

where, again, $\frac{\partial \phi}{\partial n}$ 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 $\frac{\partial \phi}{\partial n}$ on the surface represented using constant basis functions. The integral equation is satisfied at specified *centroidal collocation points*, generating a linear system of equations,

$$[A]\left[\vec{\phi}\right] = [B]\left[\frac{\partial \vec{\phi}}{\partial n}\right]. \tag{19}$$

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[10] as well as in Newman.[11]

Although our current implementation uses piecewise constant basis functions, we are extending the implementation to allow for higher order methods.

B. 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:

$$\vec{\Psi}(\vec{R},t) = \frac{1}{4\pi}\int\int\int_{V'} \frac{\vec{\omega}}{\|\vec{R}-\vec{R'}\|} dV'.$$

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

$$\nabla \times \vec{\Psi}(\vec{R},t) = \nabla \times \frac{1}{4\pi}\int\int\int_{V'} \frac{\vec{\omega}}{\|\vec{R}-\vec{R'}\|} dV'. \tag{20}$$

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.

C. 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:[9]

$$\vec{\omega}(\vec{R},t) = \sum_p \vec{\omega}_p(t) vol_p \delta(\vec{R}-\vec{R}_p(t)) = \sum_p \vec{\alpha_p}(t)\delta(\vec{R}-\vec{R}_p(t)),$$

where, $\vec{\omega}_p(t)vol_p$ is represented as $\vec{\alpha}_p(t)$. 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:

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

The vorticity induced velocity can be computed from:

$$\nabla \times \vec{\Psi}_p(\vec{R},t) = \frac{1}{4\pi}\sum_p \nabla \frac{1}{\|\vec{R}-\vec{R}_p(t)\|} \times \vec{\alpha}(\vec{R},t). \tag{21}$$

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

$$\nabla\left(\nabla\times\vec{\Psi}(\vec{R})\right)=\frac{1}{4\pi}\sum_{p}\nabla\left(\nabla\frac{1}{\|\vec{R}-\vec{R}_p(t)\|}\times\vec{\alpha}(\vec{R},t)\right). \tag{22}$$

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

$$\frac{d}{dt}\vec{R}_p(t)=\vec{U}_p(\vec{R}(t),t). \tag{23}$$

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

$$\frac{D\vec{\alpha}_p(t)}{Dt}=\vec{\alpha}_p(t)\cdot\nabla\vec{U}_p(\vec{R}(t),t). \tag{24}$$

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.[9] The evaluation of the vorticity induced vector potential can be represented as a matrix vector product:

$$[C]\,[\vec{\alpha}]=\vec{\Psi}_p$$

Where the vorticity is known and a single matrix vector product results in the vector potential. From this the velocity and gradient of velocity computations can easily be determined merely by taking the curl and gradient of the curl of the above relationship.

1. *Discrete Form of the Vorticity Evolution Equation*

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

$$\vec{R}(t+1)=\vec{R}(t)+\vec{U}_p(\vec{R}(t),t)\Delta t$$

and then the strength of the vortex can be updated as:

$$\vec{\alpha}_p(t+1)=\vec{\alpha}_p(t)+\vec{\alpha}_p(t)\cdot\nabla\vec{U}_p(\vec{R}(t),t)\Delta t.$$

It should be noted that the use of higher order time stepping method such as a Runge-Kutta scheme will be beneficial in increasing solution fidelity.

D. Conversion of the Dipole Wake Sheet to Vortex Particles

In addition to the Kutta condition, a method for representing the potential jump or vorticity in the domain due to the wake shear layer must be implemented. Close to the wing, in the current implementation, the domain vorticity is represented using a wake sheet slicing the domain which imposes the prescribed potential jump in the normal direction across the wake sheet. This *buffer wake sheet*, of prescribed potential jump is used to account for the vorticity recently shed into the domain from the wing trailing edge (See figure 3 for a pictorial description of the current section). After the wake vorticity has convected away from the wing, it is represented using a discrete particle vorticity method. These vortex particles are more easily manipulated than the traditional wake sheet representation.

The buffer wake sheet has two purposes. The first is to provide a well defined potential jump at the trailing edge. The second is to satisfy the Kutta condition. The buffer wake sheet is determined automatically once the trailing edge vertices are given. The buffer sheet is composed of two rows of panels trailing from the wing trailing edge. The first row of buffer wake panels, closest to the trailing edge initially have an unknown strength (their strength is determined by the Kutta Condition during the potential flow boundary integral equation solve), and as such are denoted as "unknown buffer wake panels". The second row of panels in the buffer wake correspond to the convected previous timestep unknown buffer wake panels, however, since the

Kutta condition was resolved at $t = t - \Delta t$, their strength is known, hence they are denoted "Known Buffer Wake Panels".

A standard result is that the change in dipole strength along a surface is equivalent to vorticity oriented in the surface tangential direction normal to the dipole gradient.[15] 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 panels is merely the difference in dipole strengths.

The arrangement of the buffer sheet also requires some care and consideration (see Fig. 3). The "unknown buffer wake panels", have a length dimension in the streamwise direction corresponding to $c_W \cdot |\vec{V}_G| \cdot \Delta t$. If $c_W = 0.5$ the current spanwise vortex is at the midpoint of the distance covered during the timestep; however, as described by Katz and Plotkin,[15] the value of c_w is typically chosen to be $0.2 \rightarrow 0.3$. This reduced value of c_w is a result of the use of ring vortices to model the wake vorticity rather than higher order vorticity distributions. In the current work, a value of 0.3 was found to be adequate. The second row of buffer wake panels have a strength corresponding to the previous timestep trailing edge potential jump (which is a known quantity), and a length in the streamwise direction of $|\vec{V}_G| \cdot \Delta t$.

The buffer wake is converted into vortex particles at each time step. This occurs by collapsing the line vortices to a given vortex particle. This is performed in a manner such that the zeroth and first order multipole moments are conserved.

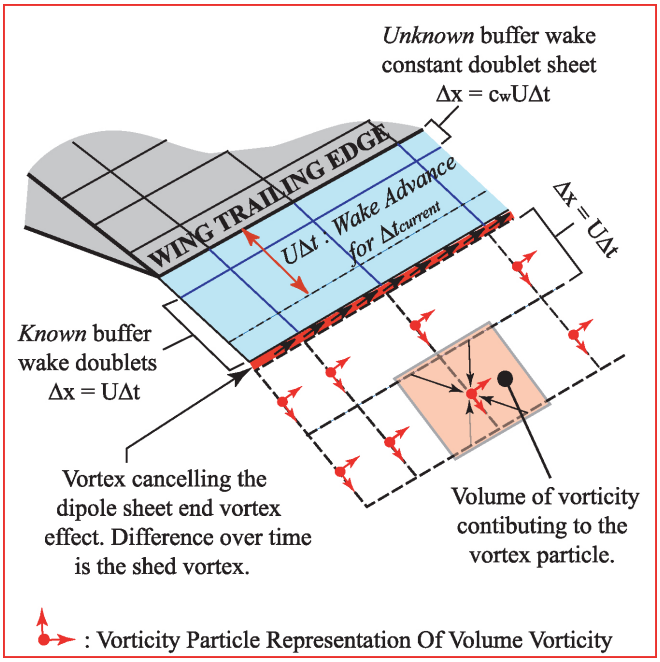


Figure 3. A description of the unknown and known buffer wake regions at the trailing edge of a wing. Notice also the conversion of the line vortices resulting from the constant strength dipoles, into point vortices.

E. 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 our 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 (eqn. 17), 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 $\frac{\partial \phi}{\partial n}$ is computed for the current body translational and rotational velocities as well as the current domain vorticity influence, $\vec{U}_\Psi$ is given by eqn. 16. Included in this flow solution is the determination of the potential jump across unknown buffer wake (Fig. 3), which enforces the potential Kutta condition (eqn. 7,and Fig. 2). This first step involves a linear system solve, and is a relatively expensive part of the overall solution algorithm.

Step 2) Determine the strength of the new vorticity released into to domain (due to the Δt time step). This new vorticity release is the converted vorticity from the previous timestep known buffer wake layer (see Fig.

3). In order to compute the time dependent spanwise vorticity, the change in strength of the buffer sheet from time step t to time step $t+1$ is easily converted from a jump in streamwise potential to a vortex line (eqn.8).
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 (eqn. 18). 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 (eqn. 21 and eqn. 22). This is merely 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 (eqn. 15), to compute 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 (eqn. 23 and eqn. 24). 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 (eqn. 21). From this influence, form a new value of $\frac{\partial \phi}{\partial n}$ for time step $t+1$ (eqn. 16).
Step 7) Start over at (1) unless the iteration stopping condition has been reached.

F. 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, their position and strength remains unchanged in the global coordinate system.

IV. 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 5,000-10,000 panels and 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 $O(nlog(n))$. We briefly present the methods used for accelerating the potential flow solution and the velocity-vorticity evaluation.

A. The pFFT Potential Flow Solver

In order to solve the discretized potential flow equation (equation 19 representing the BIE of equation 17) we implement a GMRES[12] iterative solver. As the matrix, A, is dense, the main computational cost in the GMRES 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 $O(nlog(n))$ 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,[5] based on approaches developed by Hockney and Eastwood.[14] Our implementation makes use of the pFFT++ code developed by Zhu, Song and White.[13]

During the initial pFFT setup an FFT grid is constructed in the domain surrounding the body. The steps in the algorithm are (see Fig. 4):
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 seen in Fig. 4 and Fig. 5.

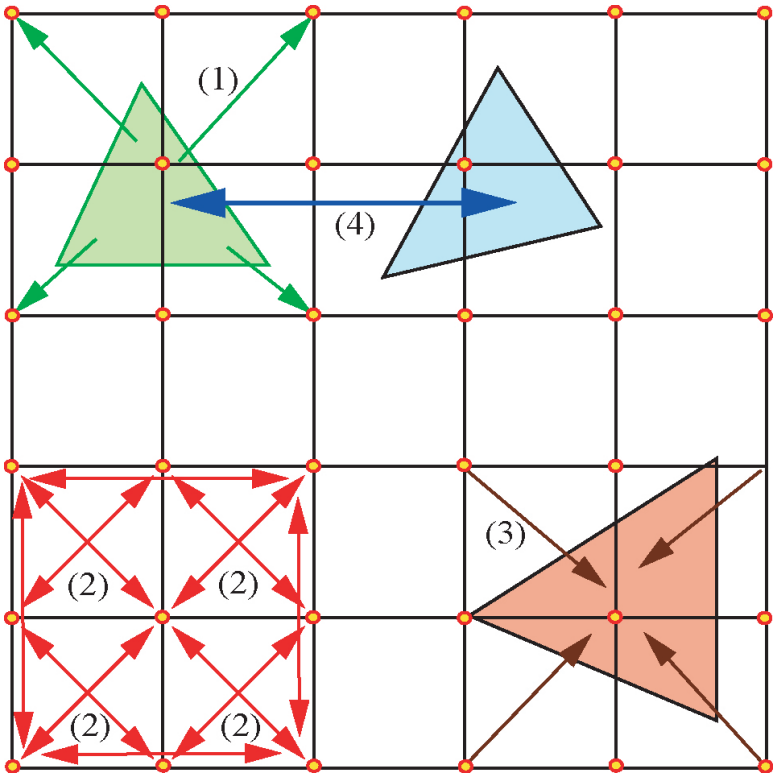


Figure 4. A schematic outlining the pFFT algorithm. The schematic demonstrates the (1) projection of the panel singularity strength onto the FFT grid, (2) the convolution operation becomes a multiplication after being transformed to the Fourier domain, to give the grid potential (3) after an inverse FFT, the grid potential is interpolated back onto the panels/evaluation points, (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 $O(nlog(n))$ time.

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*.

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 matrix vector product is one of the most efficient currently available for the solution of the potential problem.

B. 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 $O(nlog(n))$. The FMT algorithm is a variant on the Barnes-Hut[7] tree algorithm and Greengard's Fast Multipole Method.[6] The FMT constructs an octree structure complete with a multipole

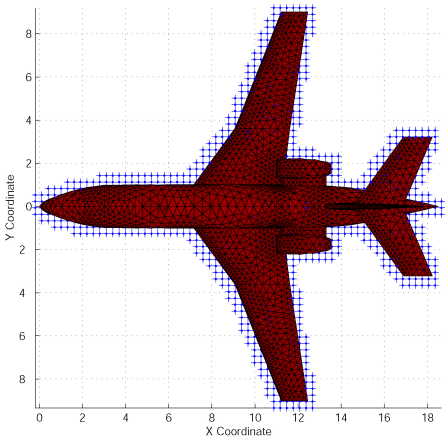


Figure 5. The pFFT grid overlay on the business jet. Points highlighted with '+' markers are those non-zero grid values after projection. The total FFT grid is a rectangular domain encapsulating the aircraft.

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.[6]

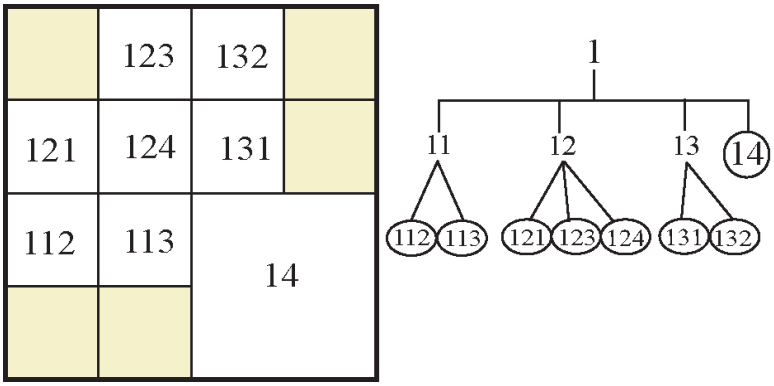


Figure 6. The general tree structure (note this is a 2D quadtree, rather than a 3D octree). The leaf cells are shown with circles in the tree structure. Empty cells without data are shaded. In the current simulation code, this tree structure is constructed using a top down approach starting from the root cell.

1. Setting up the FMT Algorithm

The following describes how the multipole tree is setup (see also Fig. 6 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.
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 10-15.
a) If no, 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 centroidally located in 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 yes, 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 to have multipole representations 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. The exact approach to selecting octree cells is presented in the next section.

2. Evaluation of the Matrix Vector Product

The evaluation of the FMT matrix vector product is performed as presented in this section. For each evaluation point, start the following list of operations with 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).
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 the 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 Fig. 7. In order to attain more efficient computations for the MVP, there

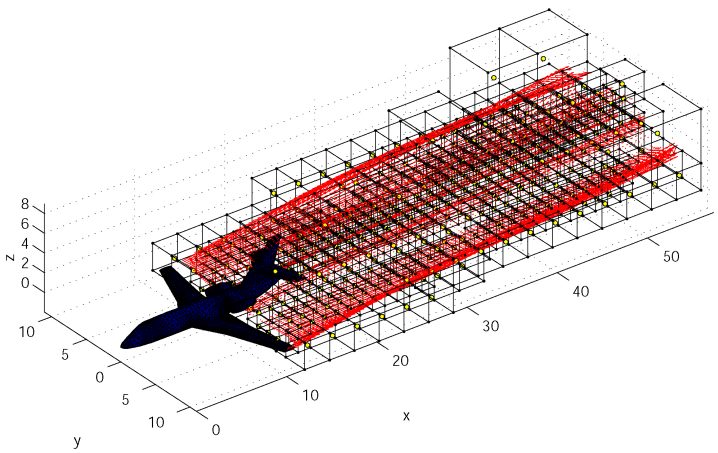


Figure 7. A plot showing the perspective view of the wake development behind a business jet, complete with the Octree representation. Notice that in regions of higher wake vortex particle density, the octree is refined.

are several additional modifications which can be used when applying the FMT; however, the basic method stands as described.

C. Using the pFFT and FMT in the Solution Process

The MVP acceleration methods have been presented. The choice of appropriate acceleration method for each step of the algorithm is briefly highlighted.

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, per given geometry, many MVP's 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. Hence, for the potential flow solution we use the pFFT.

2. The Velocity Influence Computations

In order to evaluate the velocity and velocity gradients in the domain a Fast Multipole Tree algorithm is used to perform the matrix vector product. 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) The geometry of the velocity evaluation is continuously changing. Since the wake particles are dynamic, a complete system setup is required for each time step. This is a costly procedure in both the pFFT and FMT. Since the pFFT precorrection step is significantly more expensive than the FMT when a particle approach is being used, the FMT is a better choice for this part of the simulation.
2) It is simple to use higher order multipole expansions to get the velocity gradients.
3) 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. This causes significant memory cost per simulation, since the FFT grid is heavily influenced by the aircraft discretization. In short, the FMT is more versatile, time efficient and memory efficient when it is used for evaluating quantities in the wake sufficiently far away from the aircraft body.

V. Sample Flow Simulation Results

To demonstrate the applicability of the solution approach presented in this paper we examine several examples. The first example which is explored is the wing startup case. We then consider the heaving oscillation of a wing. Finally the solution of the potential flow around a geometrically complex example, a full business jet configuration, is presented to demonstrate the versatility of the method.

A. Example 1 : The Wing Startup Case

In this first example, the unsteady lift forces for a NACA 0012 section rectangular wing for a startup case are computed. The startup flow can be thought of as a step function in body velocity at $t = 0$. The current simulation was performed with several different aspect ratio rectangular wings. The discretization uses 5280 quadrilateral panels (44 elements around the chord, 120 elements in the span). The angle of of attack for the simulation is 5 degrees. The discrete timestep is $\Delta t = 0.1$, the velocity of fluid relative to the body is $|\vec{U}| = 1$, and the chord has a value of unity.

The results are compared with those presented for a vortex lattice approach in Katz and Plotkin.[15] The results of the startup flow simulation lift force are presented in figure 8.

In figure 9, the wake vortex particle propagation is presented. This wake propagation has a distinct startup vortex region which resides near the initial position of the wing trailing edge in global coordinates. In addition to the startup vortex, the wingtip vortex roll-up can be clearly seen from this figure.

From figure 8 we can see that there is good agreement between the current approach and that presented in Katz and Plotkin[15] using a vortex lattice method. It should be noted that the current approach will have advantages over the vortex lattice approach when predicting the induced drag as well as other properties which are affected by the geometry of the surface and the thickness of the body.

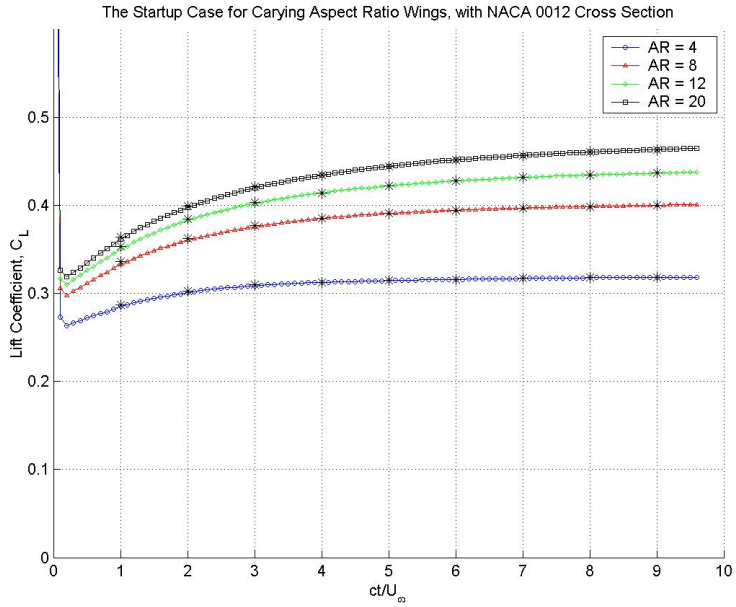


Figure 8. A plot comparing the Startup behavior of several different aspect ratio NACA 0012 section wings. The results of the simulation for each aspect ratio are presented with selected points taken from the plot in Katz and Plotkin[15] shown on the plot with (*).

B. Example 2 : The Heaving Wing Case

In this second example a heaving wing is examined, and compared with the Theodorsen 2-D, low amplitude thin airfoil approximation.[16] The Theodorsen approximation of the Z-Force for a 2-D heaving airfoil plate is:[16]

$$F_z = -\pi\rho U c C(k)\frac{dh}{dt} - \pi\rho\frac{c^2}{4}\frac{d^2h}{dt^2}, \tag{25}$$

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.[16] The wing used for this simulation is the same NACA 0012 rectangular, aspect ratio 12, wing used in the first example. The number of wake particles in each simulation approaches 20,000 by the time several periods of 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 3D vertical force as well as a centerline, quasi-2-D vertical force. The centerline 2D vertical force is computed by examining the pressures along the centerline of the wing and assuming that the behavior is a good approximation to the 2D theory. The heaving velocity in each of the following cases is defined as:

$$\frac{dh}{dt}(t) = -0.05sin(\omega t)$$

The reduced frequency of the oscillations is defined as:

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

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)-10(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, in higher frequency applications 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). In figures 11(a) and

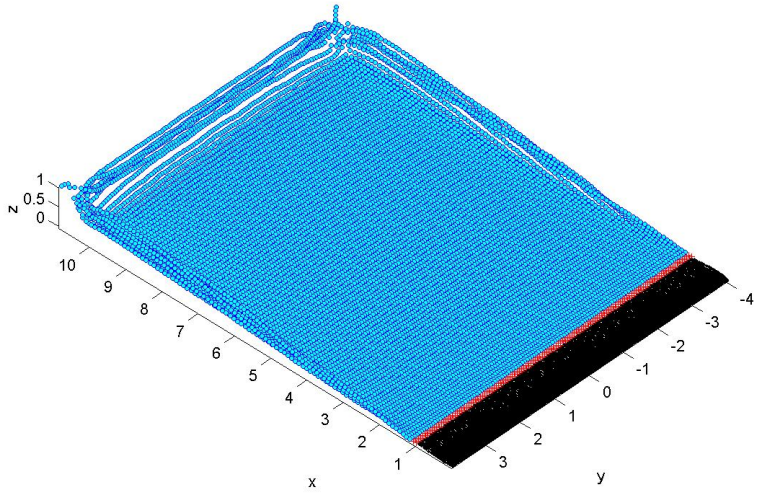


Figure 9. The wake roll-up and shape for a NACA 0012 aspect ratio 8 wing after an initial startup at $t = 0$. The current simulation shows the wake 10 seconds after startup. The plot shows the wake startup vortex as well as the wingtip vortices.

11(b), the wake position and shape is shown. Notice that there is significant self distortion of the wake for the oscillating wing despite the small heaving amplitude.

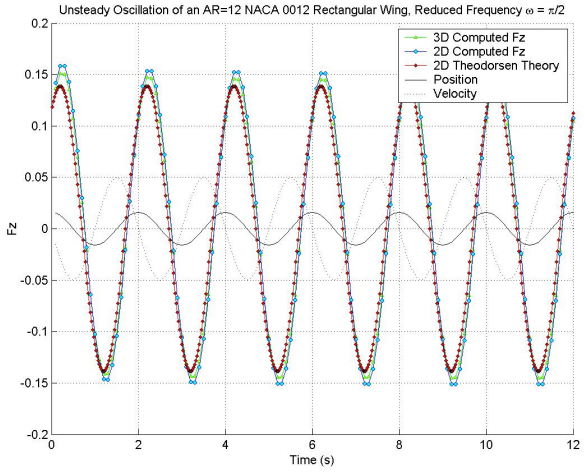
From figures 10(a)-10(d), it can be seen that there is good agreement between the predicted forces from Theodorsen's thin airfoil small amplitude theory,[16] and the computed forces. There is a slight over prediction of the Z-force, which becomes more pronounced as the frequency rises. This is suspected to be due to finite aspect ratio effects in the computed solution for both the 2-D centerline 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 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.

C. Example 3 : Complex Geometry Full Configuration Simulation

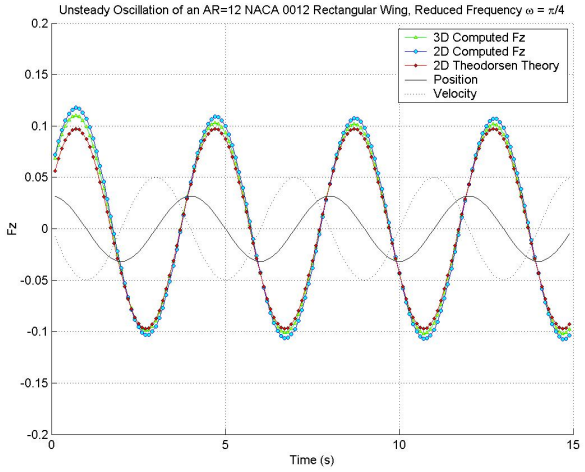
In this example, we demonstrate the ability of the current approach to handle complex geometries, such as full aircraft configurations. In this simulation a 8,650 panel business jet is presented pictorially to demonstrate the wake roll-up. The number of vortex wake particles is up to 12,000 particles. 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 12(a) - 12(c). While examining the results it should be noted that other than the shedding vertex prescription, there is no user assistance in the wake generation procedure. These simulations demonstrate completely hands off simulations from $t = 0$ to $t = t_{final}$.

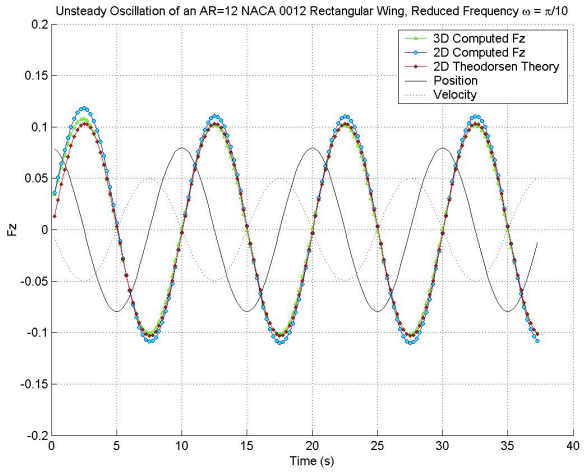
The overall simulation to steady state takes approximately 30-45 minutes to simulate using a moderate timestep resolution, on a 3ghz desktop computer. It is expected that this time consumption can be reduced through more efficient implementation of the multipole tree algorithm, data structure and overall 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.



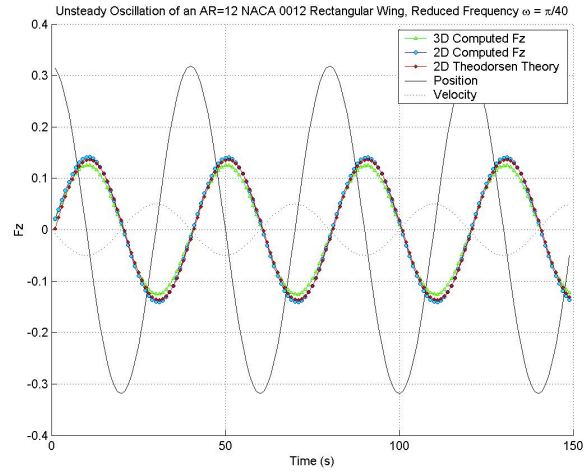
(a) Reduced Frequency $\omega_r = \frac{\pi}{2}$



(b) Reduced Frequency $\omega_r = \frac{\pi}{4}$.



(c) Reduced Frequency $\omega_r = \frac{\pi}{10}$.

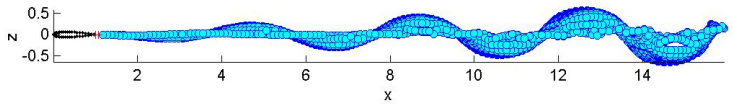


(d) Reduced Frequency $\omega_r = \frac{\pi}{40}$.

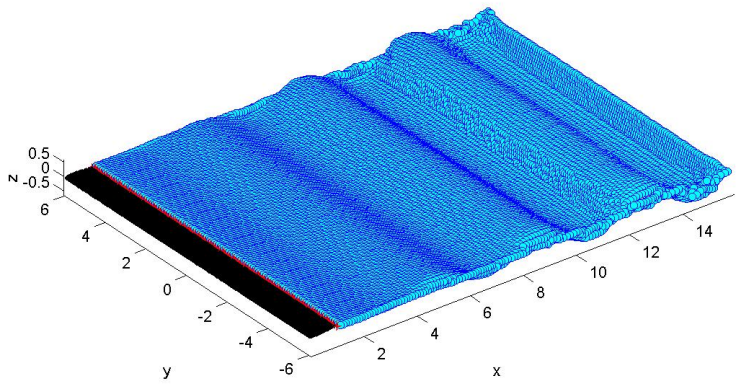
Figure 10. The results for the Z-Force component of a NACA 0012 aspect ratio 12 wing, undergoing heave oscillations at several different reduced frequencies. Notice, as the frequency increases, the Z-force tends to trend with the phase of the position and acceleration, as expected, while the lower frequency oscillations demonstrate a Z-Force phase close to the velocity, as expected.

VI. Conclusions

In this paper, we have presented a robust and efficient method for simulating the potential flow around arbitrary bodies with limited user interference and setup time. 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. The vortex particle representation of the wakes is particularly useful when hands-off unsteady and steady potential flow simulations are desired. This hands-off operation for potential flow simulation saves

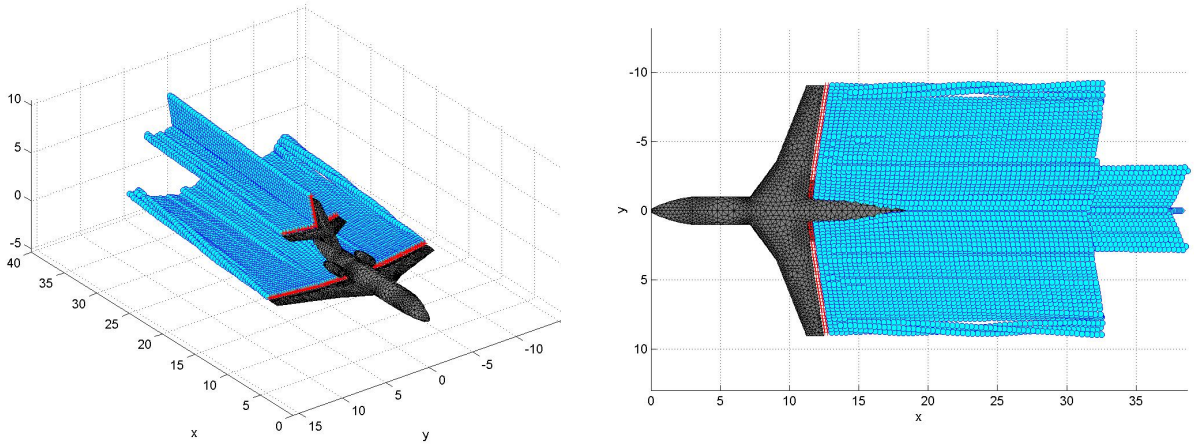


(a) The wake development over time for the $\omega_r = \frac{\pi}{4}$ case as seen from the side of the body. Notice that although the initial heaving motion amplitude is small, the wake self influence causes significant distortion of the wake as time progresses.

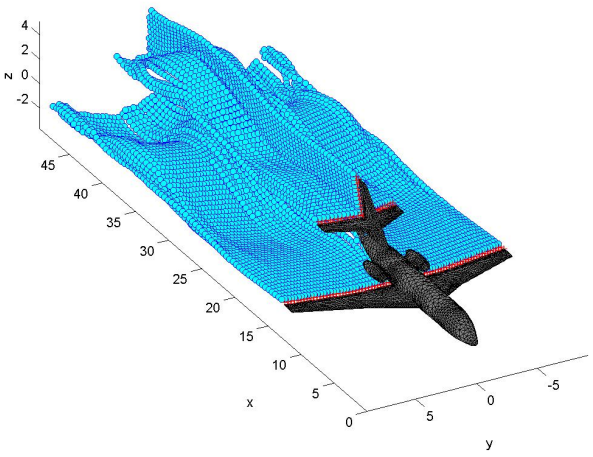


(b) The wake development over time for the $\omega_r = \frac{\pi}{4}$ case as seen from a 3D view. Notice that the wake is finely discretized using vortex particles (18,000).

Figure 11. Some figures demonstrating the wake deformation for the heave oscillation case, where the reduced frequency is $\omega_r = \frac{\pi}{4}$



(a) Illustration of the wake propagation at 5^0 Angle of Attack.(b) A figure demonstrating the fuselage-wake particle behavior.



(c) A figure demonstrating a heaving motion for the aircraft. Notice here that the wake is complicated in shape and as such, an automatic wake generation is necessary.

Figure 12. Several figures demonstrating the application of the FastAero code to complex configurations. Note: The wake was attenuated in figures (a) and (b) in order to focus on the wake in the region nearer the body.

users significant setup costs and permits more automatic optimization to take place.

Acknowledgments

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

References

[1]A. M. O. Smith, *The Panel Method: Its Original Development*, Applied Computational aerodynamics, Vol 125 Progress in Aeronautic Sciences, AIAA, 1990.

[2]D. L. Ashby, M. R. Dudley, and S. K. Iguchi, *Development and Validation of an Advanced Low Order Panel Method*, NASA TM 101024, Oct 1988.

[3]B. Maskew,*PROGRAM VSAERO: A Computer Program for Calculating the Non-Linear Aerodynamic Characteristics of Arbitrary Configurations"*, NASA CR-166476, Nov 1982.

[4]B. Rosen, J. P. Laiosa, *SPLASH Nonlinear and Unsteady Free-Surface Analysis Code for Grand Prix Yacht Design*, 13th Chesapeake Sailing Yacht Symposium, Annapolis, Jan 1997.

[5]J. R. Philips and J. K. White, *A Precorrected-FFT Method for Electrostatic Analysis of Complicated 3-D Structures*, IEEE Transactions On Computer-Aided Design of Integrated Circuits and Systems, IEEE, Vol. 16, 1997.

[6]L. Greengard and V. Rohklin, *A Fast Algorithm for Particle Simulations*, J. Comp. Phys., 73:pp325-384, 1987.

[7]J. Barnes and P. Hut, *A Hierarchical O(N log N) Force Caluclation Algorithm*, Nature 324, pp 446-449, 1986.

[8]L. Morino,*Steady, Oscillatory and Unsteady Subsonic and Supersonic Aerodynamics - Production Version (SOUSSA)* NASA CR-157130, 1980.

[9]G. S. Winckelmansand A. Leonard, *Contributions to Vortex Particle Methods for the Computation of Three-Dimensional Incompressible Unsteady Flows*, J. Comp. Phys, 109, pp 247-273, 1993.

[10]A. M. O. Smith and J. .L Hess, *Calculation of potential flow about arbitrary bodies.*, Progress in Aeronautic Sciences, 8, 1960.

[11]J. N. Newman, *Distribution of Sources and Normal Dipoles Over a Quadrilateral Panel*, J. Eng. Math., 20, 1985.

[12]Y. Saad and M. Schultz, *GMRES: A generalized Minimal Residual Algorithm for Solving Non-Symmetrixc Linear Systems*, SIAM, J. Sci. Stat. Comp., Vol 7, 1986.

[13]Z. Zhu, B. Song and J. K. White, *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.

[14]R.W.Hockney, J.W.Eastwood, *Computer Simulation Using Particles*, Taylor and Francis Inc., Bristol, 1988.

[15]J. Katz, A. Plotkin, *Low-Speed Aerodynamics*,Second Edition, Cambridge Aerospace Series, Cambridge, 2001.

[16]T. Theodorsen, *General Theory of Aerodynamic Instability and the Mechanism of Flutter*,NACA Report 496, 1935.