FastAero - a precorrected FFT - Fast Multipole Tree Steady and Unsteady Potential Flow Solver

David Willis, Graduate Student, MIT,
Jaime Peraire, Professor, MIT, Department Of Aeronautics and Astronautics,
and Jacob White, Professor, MIT, Electrical Engineering and Computer Science

Abstract— In this paper a precorrected FFT-Fast Multipole Tree (pFFT-FMT) method for solving the potential flow around arbitrary three dimensional bodies is presented. The method takes advantage of the efficiency of the pFFT and FMT algorithms to facilitate more demanding computations such as automatic wake generation and hands-off steady and unsteady aerodynamic simulations. The velocity potential on the body surfaces and in the domain is determined using a pFFT Boundary Element Method (BEM) approach based on the Green's Theorem Boundary Integral Equation. The vorticity trailing all lifting surfaces in the domain is represented using a Fast Multipole Tree, time advected, vortex particle method. Some simple steady state flow solutions are performed to demonstrate the basic capabilities of the solver. Although this paper focuses primarily on steady state solutions, it should be noted that this approach is designed to be a robust and efficient unsteady potential flow simulation tool, useful for rapid computational prototyping.

Index Terms— Aerodynamics, Panel Method, Boundary Element Method, precorrected FFT, Fast Multipole Tree, Vortex Particle Method

I. Introduction

Aircraft design has developed significantly since the invention of the computer. 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][3][4]. Furthermore, with computational simulation taking a more prominent role in design and analysis in other domains, many efficient computational methods and algorithms have been developed. In this paper we present a simulation tool, FastAero, based on advances made in electrostatic simulations (precorrected FFT [5]), large n-body interaction problems (Fast Multipole Tree [6]), as well as other contributions in a diverse set of disciplines.

The program uses a precorrected FFT (pFFT) based fast integral equation solver[5] to compute the potential flow solution on the body. In addition to the pFFT, the simulation tool also uses a Fast Multipole Tree (FMT) algorithm [6][7] to compute the contributions from the shedded wakes. The FMT algorithm is a variation on the Barnes-Hut tree code [7] and the Greengard Fast Multipole Method [6].

This work was supported by the Singapore-MIT Alliance, the NSF and NSERC Canada.

In the first section we introduce the governing fluid dynamic equations. The second section of this paper describes the Boundary Integral Equation (BIE) and the Boundary Element Method (BEM) for computing potential flow. Then, algorithms such as the pFFT and the FMT are briefly presented to introduce these techniques for improving computational efficiency. Results are presented to demonstrate that the method can simulate an aircraft while automatically generating the wake sheet.

II. The Governing Equations

In order to more clearly demonstrate the method of solution in this paper, we present the fluid domain in Fig. 1. The domain of interest is the volume of fluid external to the aircraft. Since we are dealing with higher Reynold's number flows, potential flow assumptions are adequate for an engineering design situation. In addition to the potential flow, the evolution of the lifting surface trailing vorticity is modeled using an unsteady time advancing vortex particle approach. The combined potential flow-vortex particle wake approach is novel in that it allows an automatic generation and evolution of the domain vorticity, a distinct advancement in the use of panel methods. With automatic wake generation and vorticity evolution, the BEM potential flow solver is completely automatic for lifting flows. In effect, we consider a domain whose velocity influence is derived from both a potential flow solution as well as a distribution of vorticity in the domain.

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

Most aerodynamic applications are high Reynold's number flows. As such, the fluid in the domain is assumed to be inviscid and irrotational. In addition, the flow is also assumed to be incompressible. Due to these assumptions the conservation of momentum equations can be decoupled from the conservation of mass equation. In order to maintain a consistent reference frame, all equations are derived assuming a body fixed coordinate system, unless otherwise specified.

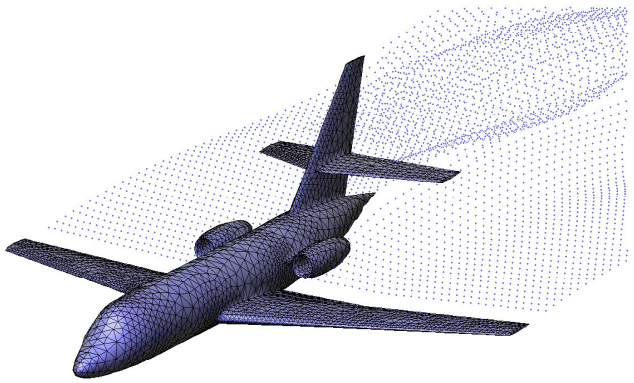

Fig. 1. A figure showing the domain of interest. The domain of interest includes all fluid external to the aircraft surface. The particles trailing the wing section, the vertical tail and horizontal stabilizer demonstrate the regions in which vorticity exists due to the lifting surface trailing shear layer. In terms of domains of interest, the fluid can be assumed to be irrotational, other than in the regions where the vorticity lies. As such potential flow can be assumed.

B. Velocity Definition

In this particular simulation code, the velocity, $\vec{U}(\vec{x},t)$ at a given point in the domain is defined as the superposition of a scalar potential component, $\vec{u}_\Phi(\vec{x},t)$ and a vector potential component, $\vec{u}_\Psi(\vec{x},t)$:

$$\vec{U}(\vec{x},t) = \vec{u}_\Phi(\vec{x},t) + \vec{u}_\Psi(\vec{x},t) = \nabla\Phi + \nabla\times\vec{\Psi}, \qquad (1)$$

The gradient of the scalar potential Φ, is the irrotational component of the velocity, $\vec{u}_\Phi(\vec{x},t)$. The scalar potential is typically the unknown quantity in most panel methods. The curl of the vector potential, $\vec{\Psi}$, gives the velocity, $\vec{u}_\Psi(\vec{x},t)$ due to any vorticity, $\vec{\omega}$, in the domain.

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:

$$\nabla\cdot(\nabla\Phi + \nabla\times\vec{\Psi}) = \nabla\cdot(\nabla\Phi) = \nabla^2\Phi = 0 \qquad (2)$$

Which is the Laplace equation for the scalar potential.

D. The Boundary Conditions

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

$$\hat{n}_{body}\cdot\vec{U}(\vec{x},t) = 0,$$

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

$$\hat{n}_{body}\cdot(\nabla\Phi + \nabla\times\vec{\Psi}) = 0. \qquad (3)$$

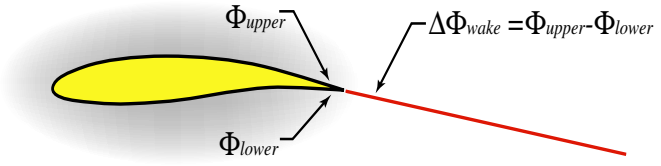

Fig. 2. A figure demonstrating the Kutta Condition governing the stream wise vorticity release into the domain. The Kutta condition requires a potential jump across the wake corresponding to the difference between the upper and lower trailing edge potentials.

Note: It is not possible to enforce a no-slip tangential flow condition at the body surface. This violates the physical no-slip condition at the wall; however, due to the assumption of inviscid and irrotational flow this wall slip condition is mathematically acceptable. The physical existence of the boundary layer can be accounted by a viscous correction. This would involve solving the potential flow on a surface boundary slightly displaced from the physical body surface to account for the boundary layer mass deficit. In this displaced region of the flow, the surface tangent flow is present. At high Reynold's numbers the boundary layer is thin, thus justifying the first approximation that potential flow is valid up until the body surface.

In the body fixed reference frame, the velocity in the farfield is:

$$\lim_{\vec{x}\to\infty}\vec{U}(\vec{x},t) = \vec{U}_\infty(\vec{x},t) = \nabla(\Phi_\infty)$$

This condition on the farfield velocity requires that any perturbations of the velocity field due to a body tend towards zero at infinity.

E. The Kutta-Joukowski Theorem

By conservation of vorticity (Kelvin's Theorem), the vorticity generated on the wing surface must be shed into the domain. Since the flow is assumed to be inviscid, a Kutta condition is applied at the trailing edge of the wing. The condition allows for a jump discontinuity in the surface potential across the geometric cusp which represents the trailing edge. In order to prescribe the stream wise vorticity in the domain, a linearized version of the pressure continuity at the trailing edge is used [8],

$$\Phi_{upper} - \Phi_{lower} = \Delta\Phi_{Wake}. \qquad (4)$$

Here, *upper* and *lower* refer to control 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 is also enforced:

$$\left[\frac{d\vec{\gamma}_{span}}{dt}\right]_{wing} = -\left[\frac{d\vec{\gamma}_{span}}{dt}\right]_{wake}, \qquad (5)$$

where γ represents the strength of a line of span wise vorticity on the wing and in the wake.

In addition to the Kutta condition, a method for representing the vorticity in the domain due to the wake shear layer must be present. Close to the wing, 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 sheet of potential jump is used to account for the vorticity recently shed into the domain. At each time step of the solution, a potential jump wake sheet is constructed to account for the wake shed vorticity. After the wake vorticity has convected away from the wing, it is represented using a discrete particle vorticity representation. These vortex particles are more easily manipulated than the wake sheet representation. The particles are treated with automatic wake evolution procedures, without having to worry about wake-body intersections.

F. The Representation of the Vorticity in the Domain

Vorticity, $\vec{\omega}$, is defined as the curl of the velocity:

$$\nabla \times \vec{u}_\Psi = \vec{\omega}$$

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 Poisson equation relating the vector potential to the vorticity.

With the relationships between the velocity, vorticity and vector potential defined, the governing equations for vorticity evolution are derived. The vorticity evolution equation is derived starting from the incompressible Navier-Stokes equations,

$$\frac{\rho \partial \vec{U}}{\partial t} + \rho \vec{U} \cdot \nabla \vec{U} = -\nabla p + \mu \nabla^2 \vec{U}, \tag{6}$$

where, ρ, is the fluid density, μ is the fluid viscosity, and p is the pressure. Taking the curl of eqn. 6, and assuming an inviscid flow, 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).

The vorticity in the domain is represented as the summation over all of the discrete vortex particles in the domain:

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

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

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

$$\frac{d}{dt}\vec{x}_p(t) = \vec{U}_p(\vec{x}(t),t). \tag{7}$$

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

$$\frac{D\vec{\alpha}_p(t)}{Dt} = \vec{\alpha}_p(t) \cdot \nabla \vec{U}_p(\vec{x}(t),t). \tag{8}$$

Each of the 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].

III. The Pressure–Velocity Relationship

The Bernoulli Equation is used in order to determine the forces and pressures on the body:

$$\frac{d\Phi}{dt} + \frac{\rho}{2}\vec{U}^2 + p_{static} = Const,$$

where p_{static} refers to the local static pressure. The steady state Bernoulli Equation is:

$$\frac{\rho}{2}\vec{U}^2 + p_{static} = Const. \tag{9}$$

From this pressure-velocity relationship, the forces and moments can easily be computed for the body.

A. The Laplace Equation in Integral Equation Form

In the method presented in this paper we use integral equations to determine the potential flow solution as well as the vorticity induced velocity. In this section, the potential flow solution is highlighted. First we define the total potential,

$$\Phi = \phi + \Phi_\infty$$

where the free stream potential, Φ_∞ induces the free stream velocity. For a constant velocity flow field, the free stream potential will vary linearly in the direction of the flow, with a proportionality constant equivalent to the flow velocity. The perturbation potential, ϕ is the potential which causes flow perturbations due to the body and the small wake strip with a Kutta condition prescribed potential jump. Similarly the total velocity at any given point is the superposition of velocities:

$$\vec{U} = \vec{u}_\phi + \vec{U}_\infty + \vec{u}_\Psi$$

In the total velocity equation, the $\vec{u}_\Psi$ component of the velocity due to the vorticity in the domain is added. The integral equation for the total potential at any point in the domain is therefore:

$$\Phi(x) = \frac{1}{4\pi}\int\int_{S'_{b+w}} \phi \frac{\partial}{\partial n}\frac{1}{\|\vec{x} - \vec{x}'\|} dS'_{b+w} -$$
$$\frac{1}{4\pi}\int\int_{S'_b} \frac{\partial \phi}{\partial n}\frac{1}{\|\vec{x} - \vec{x}'\|} dS'_b + \Phi_\infty.$$

The no-flux boundary condition applied to the potential flow integral equation is:

$$\hat{n}_{body}\left[\nabla \cdot \Phi + \nabla \times \Psi\right] = 0 = \hat{n}_{body}\nabla \cdot (\phi + \Phi_\infty) + \hat{n} \cdot \left[\nabla \times \Psi\right],$$

Which reduces to:

$$\frac{\partial \phi}{\partial n} = -\left[\vec{U}_\infty + \vec{u}_\Psi\right] \cdot \hat{n} \tag{10}$$

From this a perturbation potential integral equation for the unknown perturbation potential can be expressed as:

$$\phi(x) = \frac{1}{4\pi}\int\int_{S'_{b+w}} \phi \frac{\partial}{\partial n}\frac{1}{\|\vec{x}-\vec{x}'\|}dS'_{b+w} - \frac{1}{4\pi}\int\int_{S'_b}\left[-\left[\vec{U}_\infty + \vec{u}_\Psi\right]\cdot\hat{n}\right]\frac{1}{\|\vec{x}-\vec{x}'\|}dS'_b \tag{11}$$

The farfield boundary conditions on the perturbation potential are satisfied due to the decay of the Green's function. Once the perturbation 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 BIE in eqn. 11.

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

$$\vec{U}(x) = \nabla\left(\Phi(x)\right) = \frac{1}{4\pi}\nabla\int\int_{S'_b}\frac{\partial\phi}{\partial n}\left(\frac{1}{\|\vec{x}-\vec{x}'\|}\right)dS'_b + \frac{1}{4\pi}\nabla\int\int_{S'_{b+w}}\phi\left(\frac{\partial}{\partial n}\frac{1}{\|\vec{x}-\vec{x}'\|}\right)dS'_{b+w} + \vec{U}_\infty + \vec{u}_\Psi, \tag{12}$$

where, $\frac{\partial}{\partial n} = -\left[\vec{U}_\infty + \vec{u}_\Psi\right]\cdot\hat{n}$. If the surface of the body is discretized into body panels or elements, the surface singularity strengths (single and double layers) can be represented using basis functions. In the current method, a triangulation of the surface is used to represent the overall geometry. The Boundary Integral Equations become summations of the integrals over the triangular elements. By representing the solution ϕ, and the boundary condition $\frac{\partial\phi}{\partial n}$, on the surface of the body using constant basis functions, and enforcing the integral equation be satisfied at specified centroidal collocation points, a boundary element method linear system of equations is formed for the constant collocation BEM,

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

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 Greens functions over triangular elements. Analytical expressions for these integrals are presented in Hess and Smith [10] as well as in Newman[11].

Although the current method uses constant collocation, there is a significant amount of effort currently in place to increase the solution fidelity via higher order methods and curved panel geometry representation. Although initial panel integral calculations will be more costly, the overall solution time should decrease due to the faster solution of a reduced size matrix.

B. The Poisson Equation In Integral Equation Form

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

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

In the current flow solver, vortons or vortex particles are used to represent the vorticity. For a vortex particle representation of the vorticity, the discrete vector potential equation is:

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

The vorticity induced velocity can be computed from:

$$\nabla\times\left(\vec{\Psi}_p(x,t)\right) = \frac{1}{4\pi}\sum_p \nabla\frac{1}{\|\vec{x}-\vec{x}_p(t)\|}\times\vec{\alpha}(x,t). \tag{13}$$

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

$$\nabla\left(\nabla\times\left(\vec{\Psi}(x)\right)\right) = \frac{1}{4\pi}\sum_p \nabla\left(\nabla\frac{1}{\|\vec{x}-\vec{x}_p(t)\|}\times\vec{\alpha}\right). \tag{14}$$

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

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

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 (one of the options in the current solver structure),

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

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

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

It should be noted that the use of higher order time stepping method will be beneficial in both maintaining stability as well as solution fidelity.

IV. Putting the Equations Together

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 we accomplish this by a time stepping procedure. The following steps occur during each time step during the solution process:
Step 1) Solve the potential flow equation (eqn. 11), to determine the unknown perturbation potential values on the surface of the body. The $\frac{\partial\phi}{\partial n}$ value is computed for the current $\vec{U}_\infty$ and $\vec{u}_\Psi$ by eqn. 10. Included in this flow

solution is a "buffer wake". This buffer wake is a thin wake composed of double layer panels which aid in the enforcement of the potential Kutta condition (eqn. 4). The buffer wake panels are of a length $\vec{U} \cdot \Delta t$ and are aligned with the local free stream velocity behind each trailing surface. This buffer wake would traditionally be the entire wake representation in traditional panel methods. This first step involves a linear system solve, and is a relatively expensive part of the 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 determined from the small double layer buffer wake sheet which is used in the potential solve. The stream wise vorticity is determined by taking the gradient of the jump in the potential along the span wise direction. This is easily done with constant strength panels, since the stream wise vorticity is merely the jump in the representation of wake strength in the span wise direction. In order to compute the time dependent span wise vorticity, the change in strength of the buffer sheet from time step t to time step $t+1$ is easily converted from the jump in stream wise potential to a vortex line (eqn.5).
Step 3) Determine the velocity and gradient of the velocity influence from the body onto the vortex particles in the wake using appropriate the boundary integral equations (eqn. 12). 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. 13 and eqn. 14). 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. 9), to compute 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. 7 and eqn. 8). 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. 13). From this influence, form a new value of $\frac{\partial \phi}{\partial n}$ for time step $t+1$ (eqn. 10).
Step 7) Start over at (1) unless the iteration stopping condition has been reached.

A. The Farfield Approximation Model

Once the vortex particles have convected sufficiently far away from the body, they are lumped into multipole expansion representations, and treated as simple 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.

V. 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. 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 $O(n)$. 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 potential flow we implemented a GMRES [12] iterative solver. The operation count bottleneck in the GMRES is computing the matrix vector product (MVP) which costs $O(n^2)$ operations because the matrix is dense. 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 implements a Fourier domain multiplication for the convolution product in the integral equation. This pFFT approach was developed by Philips and White [5], and the current implementation makes use of the pFFT++ code developed by Zhu, Song and White [13].

The pFFT algorithm is outlined in Fig. 3 linked with the numerically listed operations. In the initial pFFT setup an FFT grid is constructed in the domain surrounding the body. The steps in the algorithm are:
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. 3 and Fig. 4.
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 after an inverse FFT giving the grid potentials.
Step 3) The grid potentials are interpolated back to the panels or evaluation points. This operation is similar to the projection operation, and in the case of a Galerkin BEM approach, the projection and interpolation are transpose relations. 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 interaction. 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

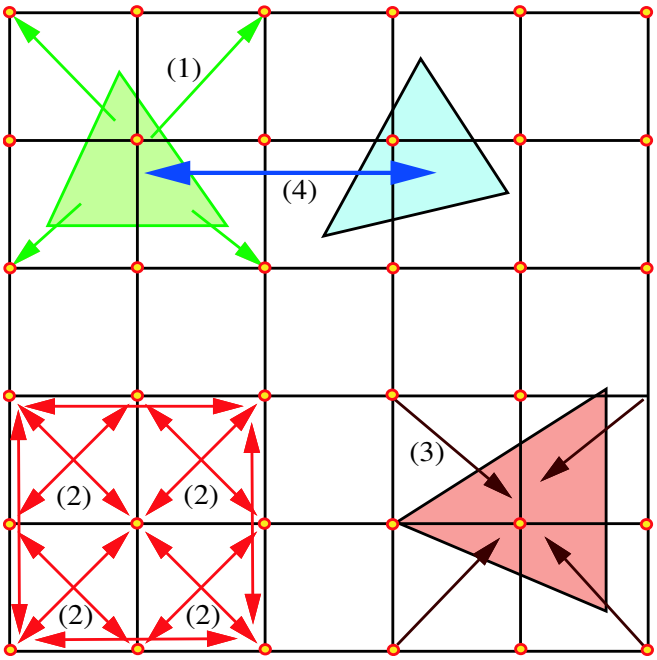

Fig. 3. A schematic outlining the pFFT algorithm. The schematic demonstrates the (1) projection of the panel singularity strength onto the FFT grid, (2) the convolution operation becomes a multiplication after being transformed to the Fourier domain, to give the grid potential (3) after an inverse FFT, 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.

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 $O(n^2)$ MVP to one of $O(nlog(n))$. 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 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 [6].

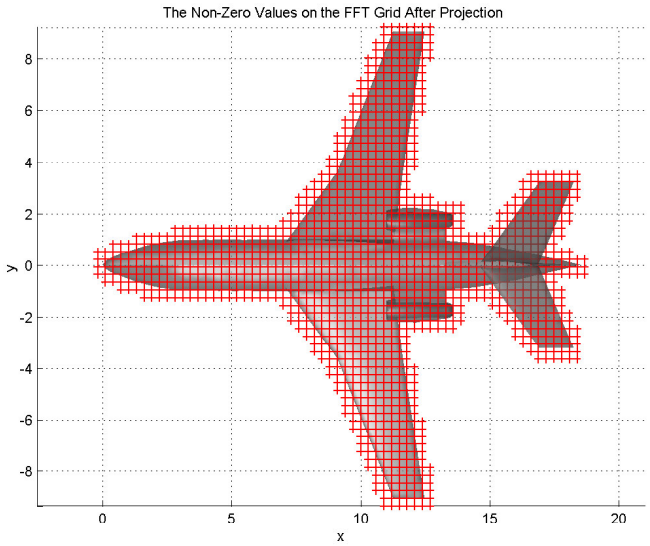

Fig. 4. The pFFT grid overlay on the business jet. Points highlighted with '+' markers are those non-zero grid values after projection. The total FFT grid is a rectangular domain encapsulating the aircraft.

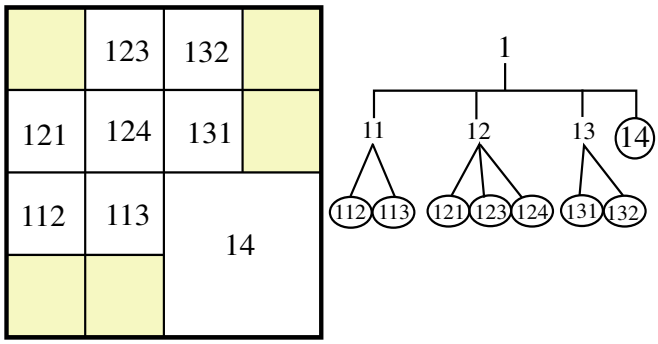

Fig. 5. The general tree structure (note this is a 2D quadtree, rather than a 3D octree). The leaf cells are shown with circles in the tree structure. Empty cells without data are shaded. In the current simulation code, this tree structure is constructed using a top down approach starting from the root cell.

1) Setting up the FMT Algorithm: The following describes how the multipole tree is setup (see also Fig. 5 for the octree structure):

Step 1) Construct the 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 typically user specified.

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.

Some practical applications of the Fast Multipole Tree code for computing the vortex influences behind a business jet are shown in Fig. 6 and Fig. 7. These applications explicitly show the octree structure in the domain.

In order to attain more efficient computations for the MVP, there are several tricks 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.

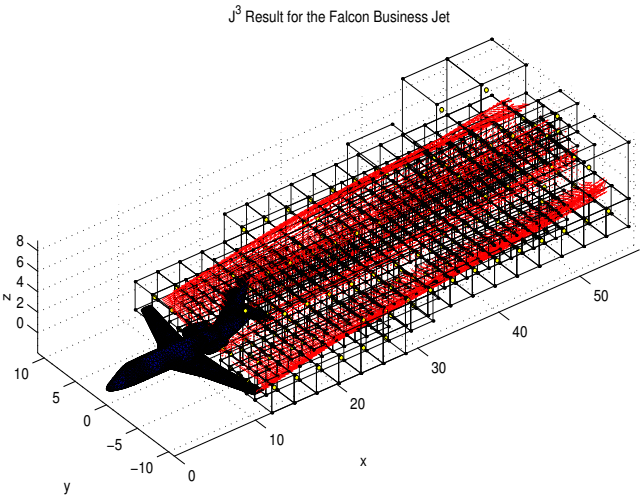

Fig. 6. A plot showing the perspective view of the wake development behind a business jet, complete with the Octree representation. Notice that in regions of higher wake vortex particle density, the octree is refined.

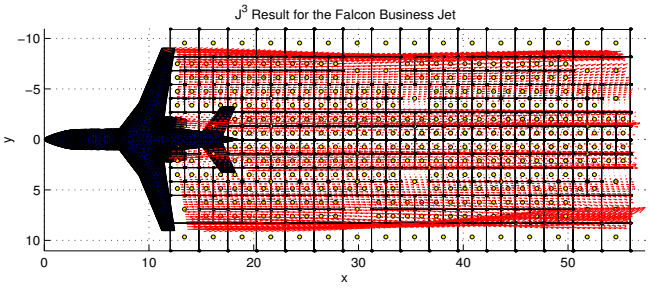

Fig. 7. A view from above showing the Octree structure over the wake system.

D. 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.

E. 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

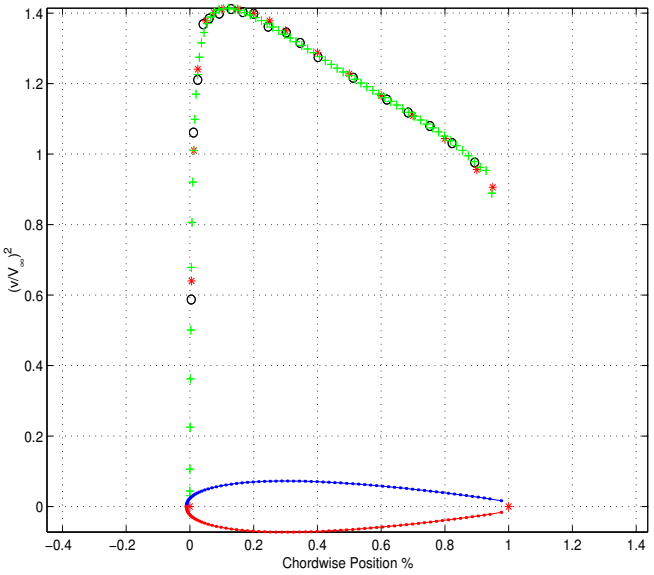

Fig. 8. The result for the NACA 0012 airfoil section at zero angle of attack. The 2D-Abbott and Von Doenhoff [14] data is represented using ('*'), the solution using the current simulation tool is represented using ('o'), and the 2D panel method result is represented using ('+'). This is a non-lifting test case, and it can clearly bee seen that there will be a symmetric pressure distribution. Notice, there is good agreement between the 2D Data and the 3D simulation.

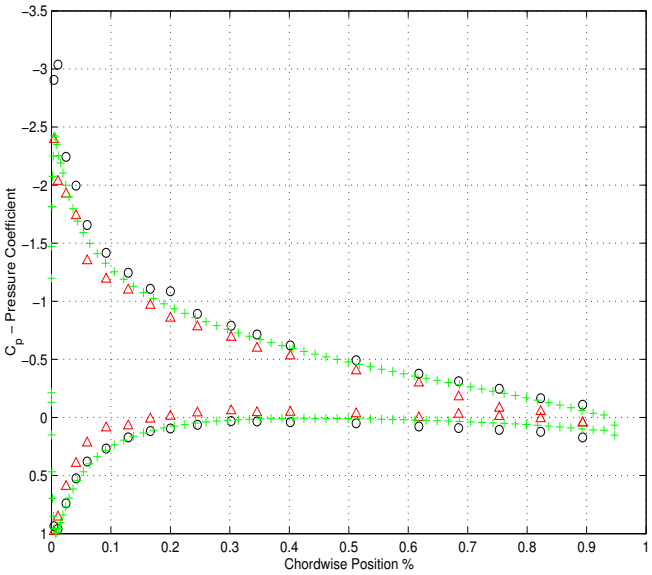

Fig. 9. The result for the high aspect ratio NACA 0012 airfoil wing section at 6 degrees angle of attack. The 3D simulation result is represented using ('o') for the aspect ratio = 16 case, with (triangles) for the aspect ratio = 8, while the 2D panel method result (AR= ∞), is represented using ('+'). This is a lifting test case. Notice, there is good agreement between the 2D Simulation results and the 3D simulation. In addition it is prudent to notice that reducing the aspect ratio reduces the lift as expected. Since the 3D simulation cases are both sufficiently high aspect ratio wings we do not expect to see significant lift reductions due to downwash.

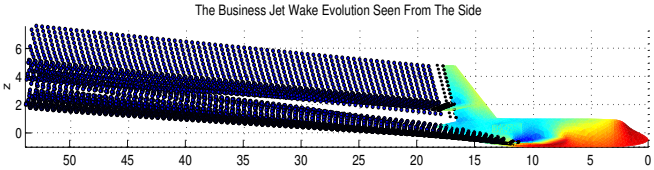

Fig. 10. A side view of the wake evolution for the business jet at 6 degrees angle of attack. Notice the downwash effect of the main wing vortex wake sheet on the horizontal and vertical tail wake sheets.

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 and more accurate to compute the gradient of velocity using the multipole vs. the pFFT.
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.

VI. Sample Flow Simulation Results

In order to validate the current approach the pressure distribution on a high aspect ratio, finite rectangular wing is used as a test case. The wing used as a test case has a NACA 0012 airfoil cross section. The aspect ratio of the wing (ratio of span to chord) is 16, hence we would expect that the flow along the centerline position to be approximately 2-D flow. In this case we sample the 3-D wing at a given chordwise station near centerline. In Fig. 8 the results for a zero angle of attack simulation are compared with both known data from Abbott and Von Doenhoff [14], as well as a 2-D panel method. Since the NACA 0012 airfoil is symmetrical, as expected there is a zero net lift and drag force.

The high aspect ratio NACA 0012 cross section wing was also used for a lifting body simulation. In this case the 3D pressure coefficient distribution along the wing centerline is expected to be virtually identical to the 2D distribution for the high aspect ratio wing. In Fig. 9, a plot of the pressure coefficient for both the 2D and 3D wing sections is shown for 3D wings of AR=16 and AR=8. It might be expected that the AR = 16 wing would have a coefficient of pressure similar to a 2D wing section. The AR=8 wing section would be expected to have a reduction in lift due to the increased downwash from the lower aspect ratio. Again, there is good agreement in these simple test cases.

To show the power and application of the method, we present some sample simulations of a business jet in a steady flow. One can see from these simulations that the wake roll-up is realistic, and the general solution is viable. The wake is automatically generated using the time stepping procedure outlined in this paper. In Figs. 10-12 we show the various views of the wake roll-up.

VII. 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

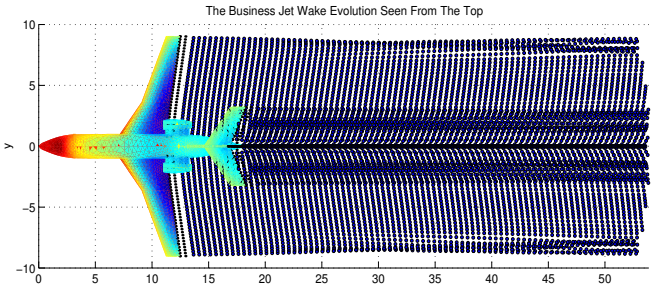

Fig. 11. The top view of the wake sheet for the business jet at 6 degrees angle of attack. The buffer region of the wake is also plotted. Note that the wake is approximated using a farfield multipole approximation model after it is cut-off (not shown here due to desire for clarity).

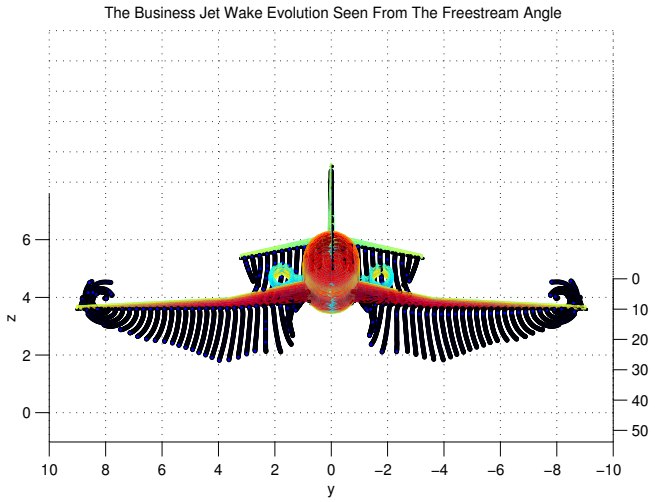

Fig. 12. The wake roll up is seen clearly for the case where the viewing axis is aligned with the free stream velocity (here the angle of attack is also 6 degrees)

the potential flow solution, while also implementing a fast multipole tree algorithm to solve the velocity and vorticity stretching influences in the domain. The approach has been demonstrated to be efficient and accurate for steady state 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 users significant setup costs and permits more automatic optimization to take place.

There are several avenues of future work to pursue. The first is to finalize and validate the simulation of unsteady flow. Second, a more efficient steady state wake solution should be implemented (requiring far fewer time steps to converge to a steady state). Finally, there are many applications which are difficult to analyze without an appropriate and relatively hands off simulation tool.

Acknowledgment

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.

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] I. H. Abbot, A. E. Von Doenhoff, Theory Of Wing Sections, Dover Publications, New York, 1959.