

Integral Velocity Sampling for Unsteady Rotor Models on Cartesian Meshes

Jonathan J. Chiew ^{*} Michael J. Aftosmis [†]

NASA Ames Research Center

Moffett Field, CA 94035

An unsteady body-force rotor model utilizing integral velocity sampling is developed for Cartesian meshes with a focus on accurate performance predictions. In this approach, the rotors are modeled as body-force source terms, and integral sampling is used to define the local freestream velocity vector. The method is verified in two dimensions for airfoils, providing accurate angle of attack and freestream Mach number estimates even on relatively coarse meshes, and demonstrating excellent mesh convergence across a variety of source region thicknesses. Extensions of the model to three dimensions for both fixed and rotating wings are considered, with numerical experiments demonstrating the need for a spanwise-continuous representation of the induced velocity or angle of attack. The model provides reasonable spanwise lift distributions without any explicit tip-loss model. Comparisons to high-fidelity simulations and experimental data show excellent agreement for predictions of an isolated DJI Phantom 3 propeller in hover. Finally, simulations of the full aircraft in edgewise forward flight also compare well with wind tunnel data across a variety of pitch angles and propeller speeds.

I. Introduction

The emergence of distributed electric propulsion has led to hundreds of novel aircraft concepts in recent years. Many of these aircraft utilize multiple propellers to enable vertical takeoff and landing (VTOL) with adequate redundancy for vehicle safety and control. These lifting propellers often generate significant unsteady aerodynamic interactions, both among the propellers and with the airframe, which can severely impact vehicle performance and noise. High-fidelity, scale-resolving computational fluid dynamics (CFD) simulations can capture these interactional effects^{1,2} but are often too expensive to utilize early in the aircraft design cycle.

The body-force approach is commonly used for reducing the computational cost of multi-rotor aircraft simulations. In this method, the physical rotor blades are replaced with representative source terms in the governing equations that model their effect on the flow field. This greatly reduces the computational cost by removing the need to capture the viscous boundary layers on the propeller blades. Early time-accurate implementations simply used structured cylindrical grids and applied the source terms along radial lines.^{3,4} Other researchers have extended the methodology to include Cartesian and hexahedral meshes,^{5,6} as well as chordwise distribution of the rotor source terms.⁷

Configuration changes are frequent early in the design cycle, making Cartesian meshes particularly compelling because they can be automatically generated for arbitrarily complex geometries. While reasonably accurate integrated performance predictions (total thrust and torque) were presented in Guntupalli's work,⁶ more recent implementations have encountered difficulties in computing accurate performance with unsteady rotor models.⁸⁻¹⁰

In this work, we explore the development of accurate unsteady rotor models, with a focus on multi-rotor aircraft and approaches suitable for non-body-fitted Cartesian meshes. In particular, freestream velocity vector computation based on the flow field is discussed in both two and three dimensions for fixed wings. Then, the extension to rotating wings is considered, including the particular subtleties that arise when computing on Cartesian meshes.

^{*}Research Aerospace Engineer, Computational Aerosciences Branch, jonathan.j.chiew@nasa.gov, AIAA Member

[†]Senior Research Scientist, Computational Aerosciences Branch, michael.aftosmis@nasa.gov, AIAA Associate Fellow

II. Numerical Method

This investigation leverages a widely used simulation framework which solves the three-dimensional inviscid flow equations on a multilevel Cartesian mesh with embedded boundaries.¹¹ The mesh consists of regular Cartesian hexahedra everywhere, except for a layer of cut cells adjacent to the body of interest; these body-intersecting cells are arbitrarily shaped polyhedra.¹² The spatial discretization employs a cell-centered, second-order accurate finite volume method with a weak imposition of boundary conditions. Time-dependent flow solutions utilize dual time stepping in which a multigrid Runge-Kutta smoother drives an implicit second-order accurate, backward difference time integration scheme similar to Jameson,¹³ using a point-implicit handling of the physical time derivative for improved stability.¹⁴

A. Governing Equations

The three dimensional equations governing unsteady compressible flow of an inviscid, calorically perfect gas for a control volume Ω with closed boundary $\partial\Omega$ may be written as

$$\iiint_{\Omega} \frac{\partial \mathbf{U}}{\partial t} dV + \oint_{\partial\Omega} \mathbf{F} \cdot \hat{\mathbf{n}} dS = \iiint_{\Omega} \mathbf{S} dV \quad (1)$$

where $\mathbf{U}$ is the vector of conserved variables,

$$\mathbf{U} = [\rho, \rho u, \rho v, \rho w, \rho E]^T \quad (2)$$

$\mathbf{F}$ is the flux density tensor,

$$\mathbf{F} = \begin{bmatrix} \rho u & \rho u^2 + p & \rho uv & \rho uw & u(\rho E + p) \\ \rho v & \rho uv & \rho v^2 + p & \rho vw & v(\rho E + p) \\ \rho w & \rho uw & \rho vw & \rho w^2 + p & w(\rho E + p) \end{bmatrix}^T \quad (3)$$

$\hat{\mathbf{n}}$ is the outward facing unit normal vector, ρ is the fluid density, u , v , and w are the Cartesian velocity components, p is the fluid pressure, E is the total energy per unit mass, and $\mathbf{S}$ is an arbitrary source-term vector density. The equations are closed with a simple equation of state:

$$p = \rho(\gamma - 1)(E - 0.5(u^2 + v^2 + w^2)) \quad (4)$$

component

B. Body-Force Model

In the governing equations (1), the source term was left unspecified. In this work, we implement body-force source terms that represent the effects of the modeled rotor blades. Leveraging blade element theory,^{15,16} the force is defined by locally two-dimensional aerodynamics for each element. In particular, consider the force proposed by Spalart^{17,18}

$$\mathbf{f} = -\rho \frac{c}{2} V_s (C_l \hat{\mathbf{e}}_z \times \mathbf{V} + C_{dp} \mathbf{V}) g(\mathbf{x}) \quad (5)$$

which naturally defines the lift component (C_l) of the force to be normal to velocity vector ($\mathbf{V}$) while the drag component (C_{dp}) is parallel. The model assumes incompressible flow, so ρ is the constant fluid density, c is the airfoil chord length, and V_s is a scalar velocity to be defined later. The real-valued function $g(\mathbf{x})$ controls the projection of $\mathbf{f}$ onto the computational mesh and integrates to unity to conserve the total force. While radial Gaussian functions are most commonly used in the existing literature, the sensitivity of the model to g will be investigated in Section III.C. Our initial simulations use a uniform distribution as a simple starting point. The sectional lift and drag coefficients in equation (5) can be either computed at run-time or interpolated from tabulated airfoil data. In either case, the freestream conditions for each element must be specified to determine the aerodynamic force coefficients. Once $\mathbf{f}$ is computed, the source terms are added to the momentum and energy equations as follows:

$$\mathbf{S} dV = \begin{bmatrix} 0 & f_x & f_y & f_z & \mathbf{f} \cdot \mathbf{V} \end{bmatrix}^T \quad (6)$$

where $f_i = \mathbf{f} \cdot \hat{\mathbf{e}}_i$, the projection of $\mathbf{f}$ into the i -th Cartesian direction. The source terms are only computed on the finest mesh and subsequently restricted to the coarser meshes with the spatial residuals as part of the multigrid solver.

Merabet & Laurendeau¹⁹ studied several velocity sampling techniques in two dimensions and found the integral velocity sampling (IVS) method^{17,18} to be superior. In this approach, the freestream velocity vector is determined by an integral of the velocity field in the source region (Equation 7). However, their simulations were run with fixed values of C_l and C_{dp} which did not vary with the sampled velocities. In practice, these coefficients will not be known a priori and must be derived from the sampled velocity, which will vary as the solution converges. Finally, we choose the simple scalar velocity $V_s = \|\mathbf{V}_\infty\|_2$ where

$$\mathbf{V}_\infty = \iint \mathbf{V}(\mathbf{x}) g(\mathbf{x}) d\mathbf{A} \quad (7)$$

III. Body-Force Aerodynamics in Two Dimensions

We begin with steady-state simulations of a NACA 0015 airfoil to verify the body-force implementation on a set of five meshes which differ in resolution by factors of two, with source region cell sizes in the range $[0.006c, 0.1c]$. The airfoil chord is of unit length and aligned with the x -direction. The far-field boundaries use a Riemann invariant formulation and are located 250 chords away. A high-resolution reference mesh was created using adaptation driven by the method of adjoint-weighted residuals,²⁰ and the body-force mesh employs a similar cell size distribution throughout the domain, only without the airfoil geometry (Figure 1).

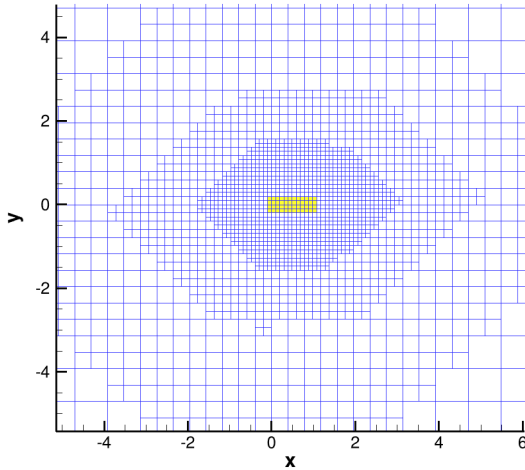


Figure 1. Coarse mesh for 2-D body-force airfoil simulations with source region in yellow

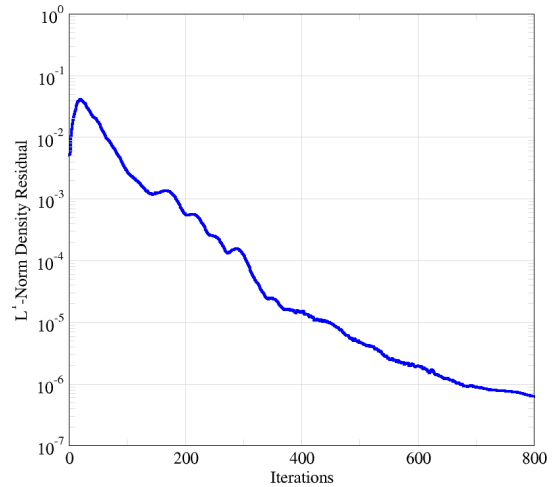


Figure 2. Sample convergence for 2-D body-force airfoil simulation

A. Integral Velocity Sampling Verification

In order to verify the fully-coupled IVS implementation, we begin with simulations at $\alpha = 2^\circ$, $M_\infty = 0.1$. In addition, the airfoil drag coefficient is explicitly set to zero, thereby matching the theoretical assumptions as closely as possible: steady, incompressible, inviscid flow with no airfoil drag. Under these conditions, we find that a fully-coupled IVS method accurately predicts the freestream conditions, even on the coarsest mesh, as shown in Figures 3 & 4. For angle of attack predictions, the errors are generally smaller than 10% across the range of cell spacings and thicknesses examined. Mach number predictions are an order of magnitude better, with discrepancies of less than 1%. In both angle of attack and Mach number, we see that thicker source regions generally reduce prediction error, especially on the coarse meshes. For Mach number, we find that only thicker source regions asymptotically converge to the correct value. This is because the weighted velocity sampling depends on an accurate computation of the airfoil circulation,¹⁸ and the thinner source regions ($<40\%c$) do not sufficiently capture the generated vorticity to converge to the exact value, although the resulting errors are quite small.

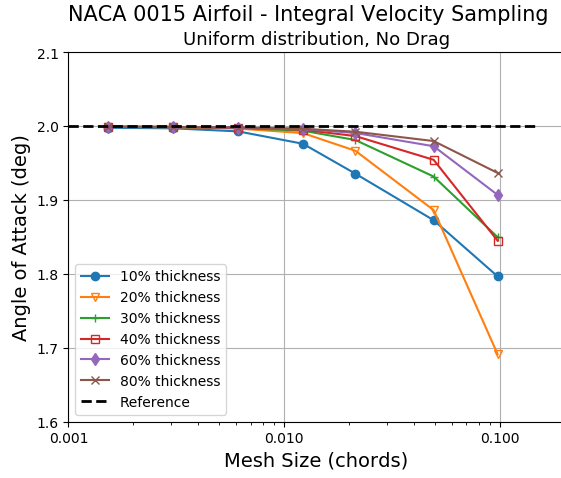


Figure 3. Mesh convergence of freestream angle of attack prediction from IVS for various thicknesses (zero drag)

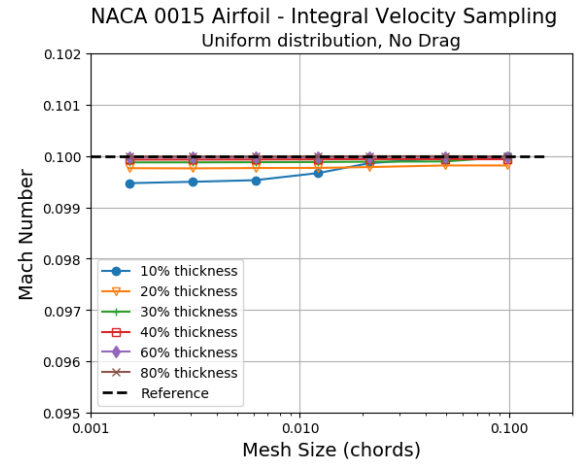


Figure 4. Mesh convergence of freestream Mach number prediction from IVS for various thicknesses (zero drag)

With preliminary verification complete, it is important to consider the effect of airfoil drag, since any practical use case will involve viscous effects. Figures 5 & 6 depict the results of a similar mesh convergence study, this time including airfoil drag ($C_d > 0$) for a source region of 40% thickness with a cell spacing of 1% chord. The results show negligible change in the prediction of angle of attack, and although the computed Mach numbers show slightly increased errors, they are still accurately predicted to within 1% of the freestream value. Furthermore, since the sectional force coefficients are expected to be reasonably insensitive to Mach number changes of this magnitude, the resulting values of $\mathbf{V}_\infty$ should provide reasonable values for the body force (Equation 5).

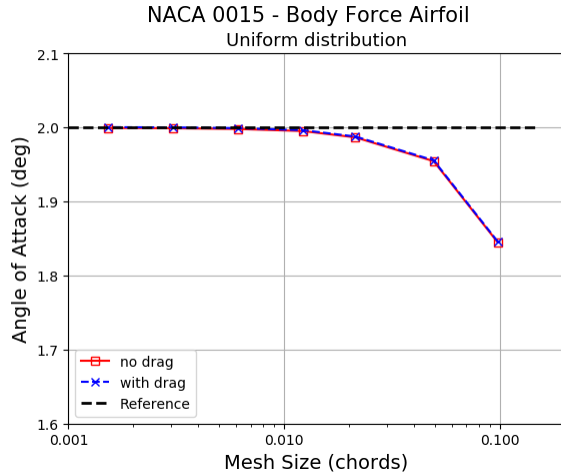


Figure 5. Mesh convergence of freestream angle of attack prediction from IVS for various thicknesses

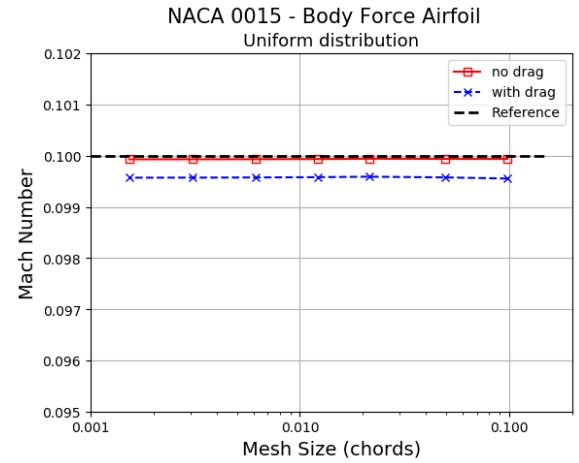


Figure 6. Mesh convergence of freestream Mach number prediction from IVS for various thicknesses

B. Angle of Attack and Mach Number Sweeps

Next, we examine a different ambient condition which relaxes the restriction of incompressible flow. Many passenger-carrying multi-rotor aircraft designs employ rotor tip speeds in the range Mach 0.5-0.6, avoiding transonic tip Mach numbers to reduce the vehicle's acoustic signature.²¹ However, compressibility effects can still be significant at these flow speeds, and so we repeat the previous mesh convergence study at a new flow condition: $\alpha = 6^\circ$, $M_\infty = 0.5$.

Figure 7 shows that angle of attack is still estimated accurately, especially with larger ($\geq 40\%c$) source region thicknesses, with errors below 1% on the finer meshes. For Mach number, we see that compressibility introduces errors of up to 2%, as seen in Figure 8. Note that these simulations were run without any airfoil drag to isolate the compressibility effects.

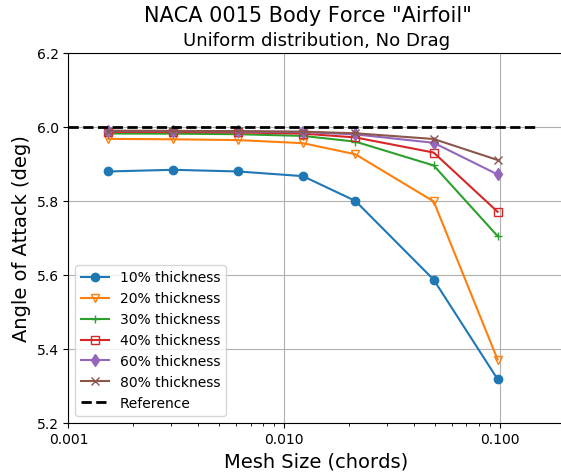


Figure 7. Mesh convergence of freestream angle of attack prediction from IVS for various thicknesses in compressible flow ($M=0.5$, zero drag)

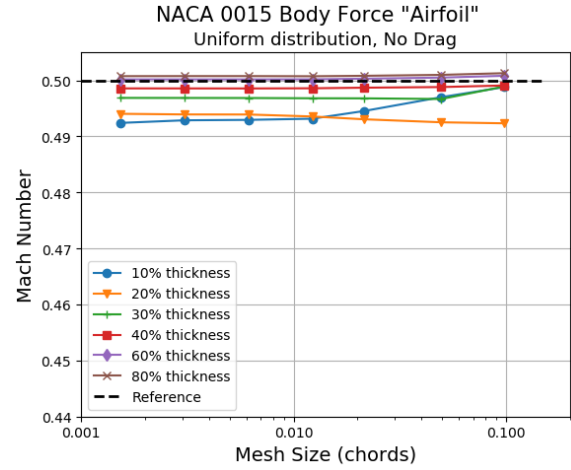


Figure 8. Mesh convergence of freestream Mach number prediction from IVS for various thicknesses in compressible flow ($M=0.5$, zero drag)

The actual cases of interest will of course encounter compressibility (moderately high tip Mach numbers) and viscous effects, since prediction of rotor power required involves airfoil drag. We now include the airfoil drag force at this Mach number to assess the accuracy of integral velocity sampling for a “realistic” case. Figures 9 & 10 show how the inclusion of airfoil drag behaves similarly to the incompressible case, repeated again with a thickness of $0.4c$. Angle of attack predictions are nearly identical to the inviscid case, and the freestream Mach number is very slightly underpredicted with about 1% error compared to the far-field reference value.

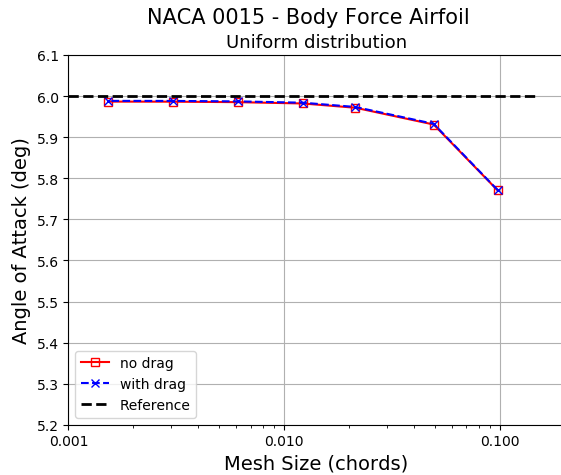


Figure 9. Mesh convergence of freestream angle of attack prediction from IVS for various thicknesses

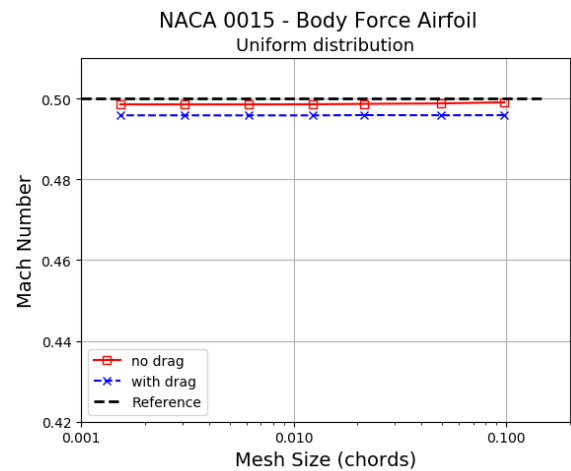


Figure 10. Mesh convergence of freestream Mach number prediction from IVS for various thicknesses

Finally, we perform two final studies to assess the accuracy of integral velocity sampling across a range of flow conditions. All simulations include airfoil drag and use best practices from the previous section, namely, a source region with 40% thickness and a cell size of $\approx 1\%c$. The results of an angle of attack sweep between -8 and 14 degrees are shown in Figure 11 with a freestream velocity of $M_\infty=0.5$. We find excellent agreement ($<1\%$ error) across all the angles of attack, except at 14° where the airfoil (table) is stalled. In this case,

the net result is that while interpolated values for C_l should be quite reasonable, C_d will be underpredicted, producing optimistic efficiency estimates at high angles of attack. Figure 12 highlights how the freestream Mach number is also predicted nearly exactly for all cases at an angle of attack of 2 degrees.

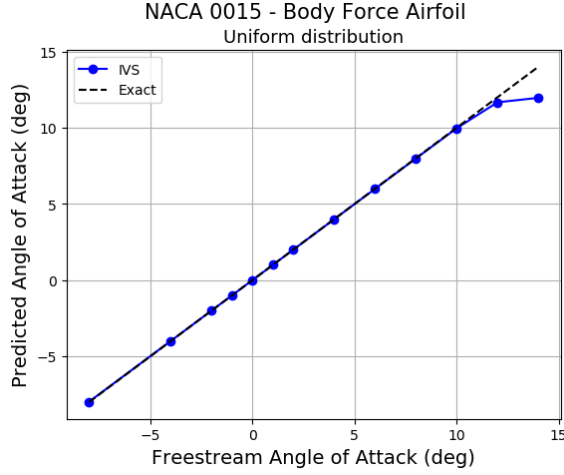


Figure 11. Accurate prediction of freestream angle of attack with IVS at a fixed Mach ($M_\infty = 0.5$)

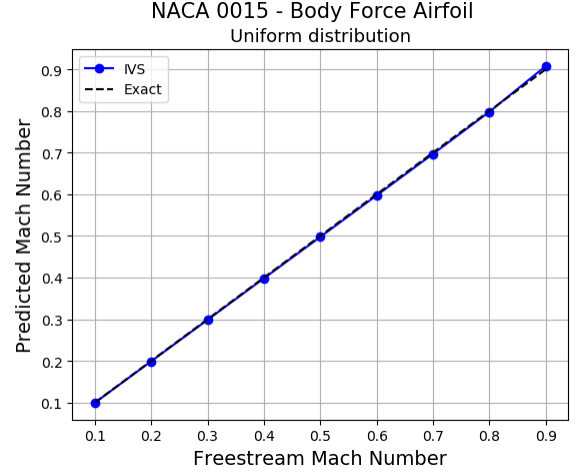


Figure 12. Excellent prediction of freestream Mach number with IVS at a fixed angle of attack (2 deg)

C. Force Projection Functional Forms

In Section III.A, a uniform distribution was chosen for the force projection function, $g(\mathbf{x})$, purely for simplicity. In fact, the derivation of the integral velocity sampling method only assumes that g integrates to unity. Spherically isotropic Gaussians are most commonly used for the force projection, however some researchers have started to consider anisotropic projections.^{18,22}

In this section, we explore force projection functions which are a product of orthogonal distributions in the x and y directions

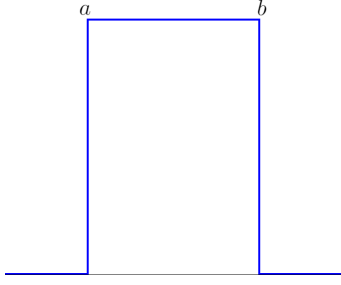
$$g(x, y) = g_x(x)g_y(y) \quad (8)$$

subject to

$$\int_{-\infty}^{\infty} g_\xi(\xi) d\xi = 1 \quad \xi = x, y \quad (9)$$

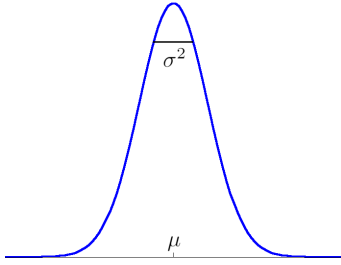
for two reasons. First, this permits a very natural shaping of the force distribution in the chordwise and thickness directions independently, which could lead to more accurate near-field pressure distributions. Secondly, this approach is also well suited to the cylindrical coordinate system to be employed in the rotary-wing implementation. For an unsteady rotor model, the body force volume changes at each timestep, which then requires an updated computation of the force projection function on the rotor hexes. Utilizing an orthogonal distribution (in cylindrical coordinates) in conjunction with the assumption that the unsteady motion is in the azimuthal direction simplifies the force projection update - only the scalar azimuthal function (g_ψ) would require re-evaluation. As a first step, we investigate simple statistical distributions (which integrate to unity by definition) to determine the effect on the prediction of $\mathbf{V}_\infty$, although others have tried more complex approaches.²³

1. *Uniform*



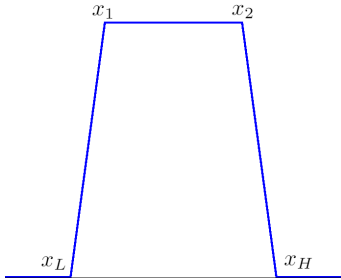
$$g = \begin{cases} \frac{1}{b-a} & : x \in [a, b] \\ 0 & : \text{otherwise} \end{cases} \quad (10)$$

2. *Gaussian*



$$g = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (11)$$

3. *Trapezoidal*



$$g = \begin{cases} \kappa \frac{x-x_L}{x_1-x_L} & : x \in [x_L, x_1] \\ \kappa & : x \in [x_1, x_2] \\ \kappa \frac{x_H-x}{x_H-x_2} & : x \in [x_2, x_H] \\ 0 & : \text{otherwise} \end{cases} \quad (12)$$

$$\kappa = \frac{2}{x_H - x_L + x_2 - x_1} \quad (13)$$

The uniform distribution implies that the computed $\mathbf{V}_\infty$ is simply the average velocity in the source region. The trapezoidal distribution can be thought of as a uniform distribution with a linear transition near the boundaries. In addition, if $x_1 = x_2$, the trapezoidal distribution becomes triangular. The Gaussian distribution is included for comparison to other actuator line methods.^{18, 24}

In order to ensure that the method maintains freestream values, the integral weights $g(\mathbf{x})$ are corrected so that $\Sigma g(\mathbf{x}) dA = 1$ over the source region cells, assuming a homogeneous error distribution. This correction results in $\mathbf{V}_\infty$ being correctly predicted in uniform flow no matter how coarse the mesh. Of course the quadrature error is not equidistributed, but this is a reasonable first-order correction.

Our previous work on the IVS method²⁵ found that the uniform distribution was the most accurate, followed by the triangular, trapezoidal ($x_1 = 1/3, x_2 = 2/3$), and Gaussian ($\sigma = t/6$) distributions in that order. We repeat that analysis at $M_\infty = 0.1$ to create a baseline set of results matching the flow condition from Section III.A (Figures 13 & 14). We have already shown that using thicker source regions and finer meshes removes much of the error when utilizing a uniform distribution.

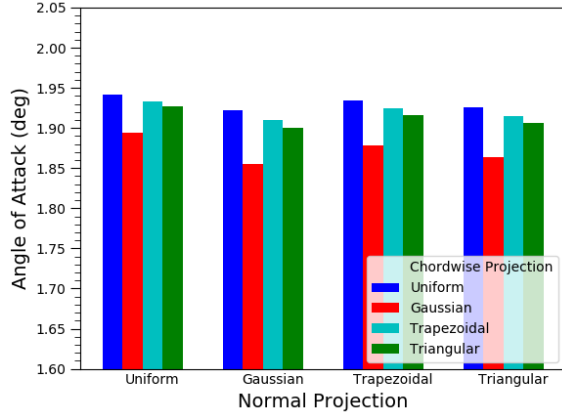


Figure 13. Predicted angle of attack sensitivity to force projection function

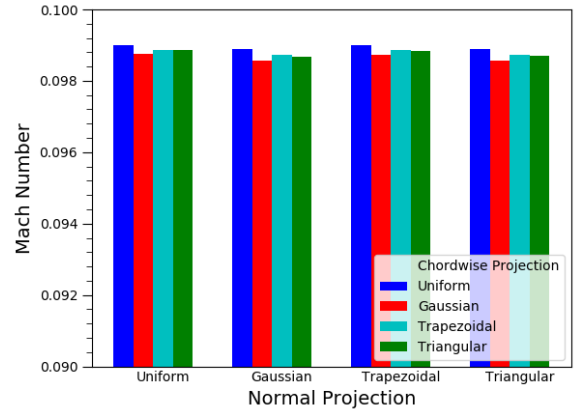


Figure 14. Predicted Mach number sensitivity to force projection function

The predicted freestream velocity vectors in Figures 15 & 16 show similar behavior across all projection functions analyzed, using the same mesh spacing ($1\%c$) and source region thickness ($40\%c$) as in the previous section. The absolute error is much smaller and differences among projection functions have all but disappeared. For the remainder of this work, we retain the uniform distribution due to its simplicity and improved accuracy on coarse meshes.

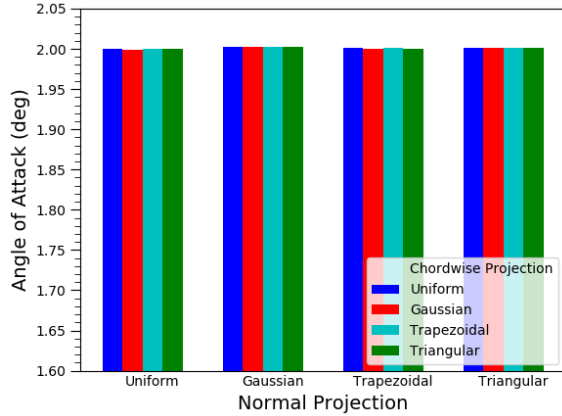


Figure 15. Predicted angle of attack sensitivity to force projection function with finer mesh and thicker source region

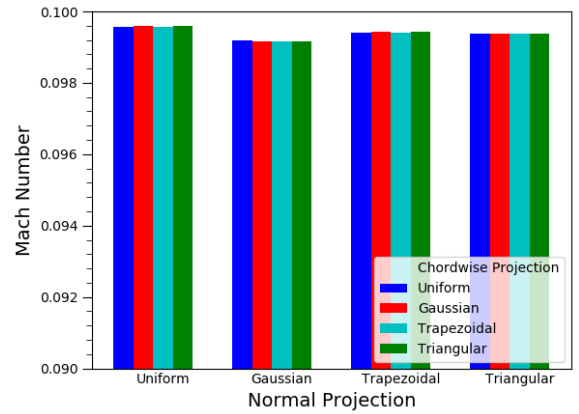


Figure 16. Predicted Mach number sensitivity to force projection function with finer mesh and thicker source region

IV. Body-Force Aerodynamics in Three Dimensions

In this section, an extension of the IVS approach to three dimensions is discussed. As mentioned previously, typical models employ a spherically isotropic Gaussian function to project the blade element force onto the computational mesh. This approach constrains the force projection to be governed by one length scale, whereas the airfoil chord, thickness, and element width are in fact all independent parameters. In addition, the spanwise smoothing of $\mathbf{f}$ by $g(\mathbf{x})$ seems to be at odds with the assumptions of blade element theory and two-dimensional aerodynamics. Instead, we choose to extend the anisotropic distribution used in Section III.C with an orthogonal spanwise projection over each wing or blade element. For simplicity, we again utilize a uniform distribution. This approach restricts the body-force to only the element-intersecting grid cells and evenly distributes the aerodynamic force onto those hexahedra.

A. Fixed-Wing

We begin with a discussion for finite-span fixed wings before including the complexities of rotation, considering only the case where the wing axis is in the z -direction. We allow arbitrary spacing of the wing elements, but initially only consider uniform spacings in this paper. Non-overlapping elements (S_i) are chosen such that

$$S_w = \bigcup_{i=1}^N S_i, \quad \bigcap_{i=1}^N S_i = \emptyset \quad (14)$$

where S_w is the wing planform. While the mesh and element spacings are distinct, we focus our attention on the case where the grid cells are smaller than the partition widths. In the case where the mesh is much coarser than the element partitioning, $\mathbf{V}_\infty$ will be approximately constant over each partition since $\mathbf{V}(\mathbf{x})$ is the cell-centered value and thus invariant over the source region. Thus $\mathbf{f}$ is also constant across the element partitions, and the fine element spacing is largely inconsequential.

We solve the governing equations (1) in the wing's frame of reference so there is no motion of the source terms relative to the computational mesh. This enables many of the geometric computations to be performed in a pre-processing step, while still maintaining full coupling of the body-force source terms. The Cartesian mesh and wing axis assumption simplify the search for cells within each wing element to rapid axis-aligned bounding box intersection tests. The wing bounding box is defined by the wingspan, chord, and thickness, while the Cartesian hexahedra are their own bounding boxes. For intersecting hexahedra, the projection function, $g(\mathbf{x})$, is then computed at the cell centroid. Cells intersecting the partition boundaries are simply split in two, and a new centroid is computed for each split element along with a corresponding value for g . Then $\mathbf{V}_\infty$ is simply computed as the g -weighted summation over all cells intersecting the j -th wing element

$$\mathbf{V}_{\infty,j} = \sum_{i=1}^{N_j} g_i \mathbf{V}_i dV \quad (15)$$

where g_i is the cell centered value of the projection function computed during preprocessing.

1. Spanwise Interpolation

The aerodynamic source terms computed from the element freestream velocities (Equation (15)) are disjoint because we chose a non-overlapping spanwise force projection. Figure 17 demonstrates how this creates spurious vorticity along element boundaries. While this effect will vanish in the limit of infinite elements, we develop a method to remove this spurious vorticity independent of the number of source elements (N).

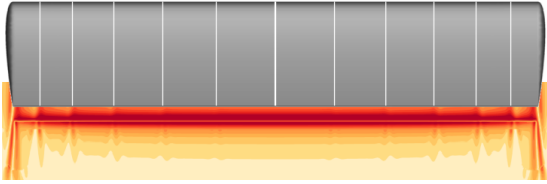


Figure 17. Top view showing discontinuous vorticity distribution (red=high) at the trailing edge without spanwise interpolation (wing shown for reference only)

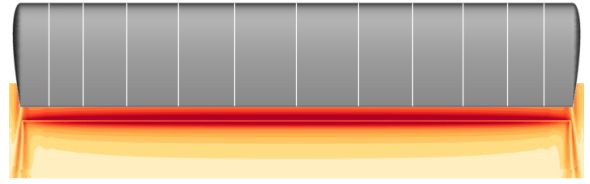


Figure 18. Top view showing continuous vorticity distribution (red=high) at the trailing edge with spanwise interpolation (wing shown for reference only)

We create a C^0 -continuous force distribution by interpolating the element values. In the fixed wing case, it is sufficient to interpolate angle-of-attack only, as shown in Figure 18. A quadratic reconstruction on each element is generated by first linearly interpolating neighboring element values to the element boundary (B):

$$\alpha_B = \alpha_i + \frac{\alpha_{i+1} - \alpha_i}{x_{i+1} - x_i} (x_B - x_i) \quad (16)$$

Then, each interpolating polynomial is computed about its element midpoint in Newton form:

$$\alpha(x) = \alpha_M + c_1(x - x_M) + c_2(x - x_M)(x - x_L) \quad (17)$$

where

$$c_1 = \frac{\alpha_L - \alpha_M}{x_L - x_M} \quad (18)$$

$$c_2 = \frac{1}{x_H - x_M} \left(\frac{\alpha_H - \alpha_L}{x_H - x_L} - c_1 \right) \quad (19)$$

and the subscripts M , L , H denote the element's midpoint, low boundary, and high boundary, respectively. The first and last elements use linear extrapolation to the wingtips, resulting in a first-order accurate reconstruction on those elements.

Figure 19 shows how the interpolated spanwise angle-of-attack distribution converges as the number of spanwise elements is increased for a rectangular wing with aspect ratio 8 and NACA 0015 airfoil section. The spanwise variation of lift coefficient is plotted in Figure 20 and compared to a reference loading computed with the actual wing geometry in inviscid flow. The angle of attack is varied for the IVS model to exactly match the total lift, and we see that the body-force approach provides reasonable span loadings but underpredicts the loss of lift at the wingtips.

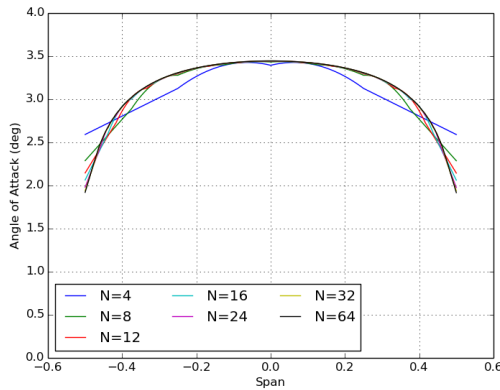


Figure 19. Spanwise distribution of interpolated angle of attack for various numbers of wing elements (N)

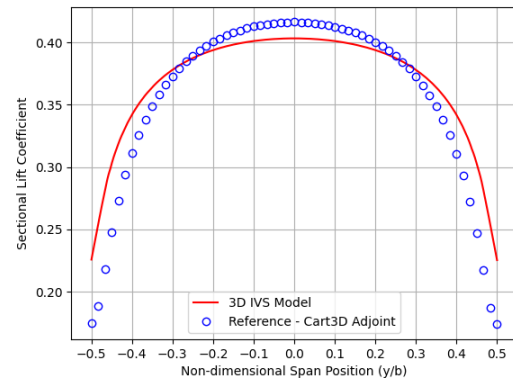


Figure 20. Spanwise lift distribution demonstrating how the IVS model overpredicts lift at the wingtips

2. Tip-Loss Model

Next, we investigate the effect of simple tip-loss modeling on the span loading. Since the loading must drop to zero at the wing tip, we can force zero lift by explicitly constraining the angle of attack to be zero at the wingtips, rather than linearly extrapolated from the interior values. In the general case of a cambered wing section, the angle of attack would be set to the zero lift angle defined by the tabulated airfoil data.

Figure 21 shows how this approach does reduce the angle of attack to zero at the wingtip, but the inboard distribution is largely unchanged when compared to Figure 19. As a result, the span loading on the wing is also largely unchanged (Figure 22), even with fewer ($N=12$) wing elements which emphasizes the tip-loss effects. We also note that the quadratic interpolation on the wingtip element is noticeable and perhaps a linear interpolation would provide better results. In the future, we hope to investigate this effect as well as more complex tip-loss models; nevertheless, the extension to three dimensions provides reasonable lift distributions as-is, and we proceed to implementing an IVS approach for propellers.

B. Rotating-Wing

1. Cylindrical Coordinates

Having established reasonable confidence in the model fundamentals by exploring its performance for the simpler airfoil and wing examples, we turn our attention to the primary case of interest, propellers and rotors. This is considerably more complex because the problem is inherently time-dependent, and the source terms will rotate through the mesh as the simulation progresses. Furthermore, the (rotating) blade axis

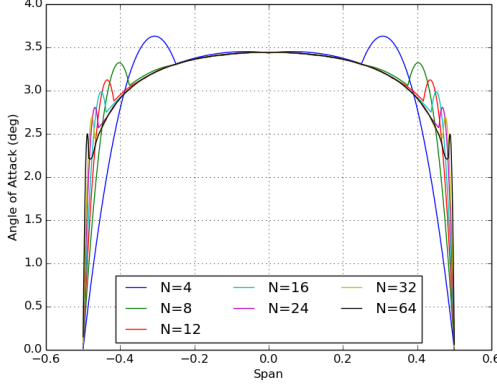


Figure 21. Spanwise angle of attack distribution for IVS model with explicit tip-loss over the last wing element

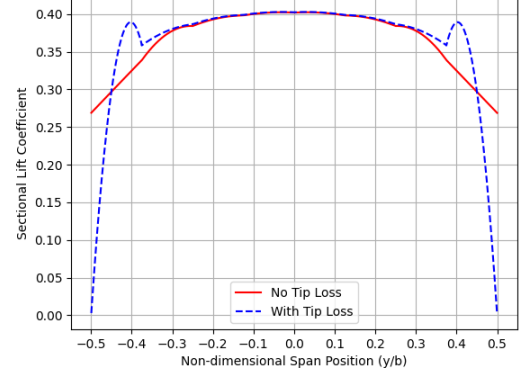


Figure 22. Comparison of lift distributions demonstrating the spanwise extent of tip-loss effects ($N=12$)

cannot be assumed to be aligned with a Cartesian axis. Therefore, it is natural to switch to a cylindrical coordinate system, where the source term motion (to first order) is in the azimuthal direction only. This coordinate system has its origin at the rotor's center of rotation and its longitudinal axis aligned with the rotor shaft. The polar axis (zero azimuth angle) is chosen to point aft, a typical rotorcraft convention, or starboard if the rotor shaft points exactly forward or aft. For an arbitrary point ($\mathbf{x}_i$), given in Cartesian coordinates, the conversion to cylindrical coordinates is as follows:

$$\mathbf{d} = \mathbf{x}_i - \mathbf{x}_c \quad (20)$$

$$k_1 = d_x \sqrt{a_y^2 + a_z^2} - a_x d_z \quad (21)$$

$$k_2 = d_y \sqrt{a_x^2 + a_z^2} - a_y d_z \quad (22)$$

$$\psi = \text{atan2}(k_2, k_1) \quad (23)$$

$$r = \sqrt{k_1^2 + k_2^2} \quad (24)$$

$$h = \mathbf{d} \cdot \mathbf{a} \quad (25)$$

where $\mathbf{d}$ is the displacement vector relative to the rotor center ($\mathbf{x}_c$) in the Cartesian coordinate system, and r, ψ, h are the standard cylindrical coordinates, reserving θ for the blade twist angle distribution. $\mathbf{a}$ is the normalized shaft axis vector, with Cartesian vector components (x, y, z) denoted by subscripts.

2. Rotor Model Preprocessor

Given this coordinate system, it is trivial to define blade elements by simply specifying their radial and thickness extents. We assume that the simulation's temporal domain will exceed one rotor rotation period, requiring $\psi \in [0, 2\pi]$, which means that the swept space-time volume of any blade element is a cylindrical shell (Figure 23). We now need to connect these elements to their associated Cartesian mesh cells to add the body force source terms to the governing equations. The element boundaries of arbitrarily oriented propellers can create complex intersections with the hexahedral mesh; however, Cartesian grid algorithms in the literature already handle arbitrarily-oriented, tessellated geometry with ease.¹² We leverage these algorithms by constructing simple triangulations enclosing the swept volume of each blade element (Figure 23) and employ rapid triangle-cube intersection algorithms to determine the mapping of hexahedral cells to blade elements (Figure 24). The blades are assumed to be rigid and hingeless which is reasonable for carbon fiber propellers used on multi-rotor aircraft today.

Each cylindrical shell triangulation is generated with the following procedure. First, an orthonormal basis ($\mathbf{v}_1, \mathbf{v}_2$) spanning the plane of rotation is computed. If the rotor plane is aligned with a Cartesian axis, then $\mathbf{v}_1$ is defined to be the unit vector normal to that plane pointing in the positive Cartesian direction. Otherwise, a random vector is chosen and orthonormalized with the Gram-Schmidt process to create $\mathbf{v}_1$.

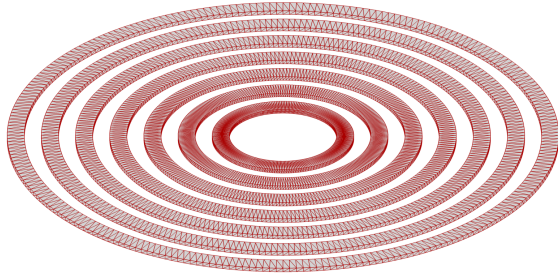


Figure 23. Cylindrical shell triangulations for determining blade element mapping (every other shell removed for clarity)

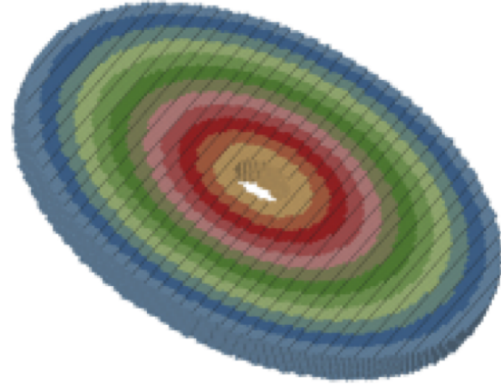


Figure 24. Resulting cut “rotor hexes” from triangle-cube intersection with colors denoting individual blade elements

If the angle between the random vector and rotor shaft axis does not exceed a minimum tolerance, a new random vector is chosen. With $\mathbf{v}_1$ in hand, the basis is completed with $\mathbf{v}_2 = \mathbf{a} \times \mathbf{v}_1$. Given this orthonormal basis and a radius (r_j) for a blade element boundary (Figure 25), the vertices on the corresponding circle are computed as

$$\mathbf{x}_i = \mathbf{x}_c + r_j (\mathbf{v}_1 \cos \psi_i + \mathbf{v}_2 \sin \psi_i) \quad (26)$$

where ψ_i is an element of the set Ψ discretizing the angular interval $[0, 2\pi]$. We find that approximately 2000 points (uniformly distributed) provides good results across a range of cases, noting that the cell spacing near the blade tip is typically $<1\%$ of the rotor radius in order to capture the tip vortex reasonably well. Nevertheless, we keep the angular discretization as an adjustable parameter for flexibility. Each triangulation consists of 4 circular sets of vertices: two sets at each of the inner and outer radii, offset by the source region thickness along the shaft axis. Finally, we note that this procedure is embarrassingly parallel once the basis vectors are computed since there are no dependencies between adjacent triangulations.

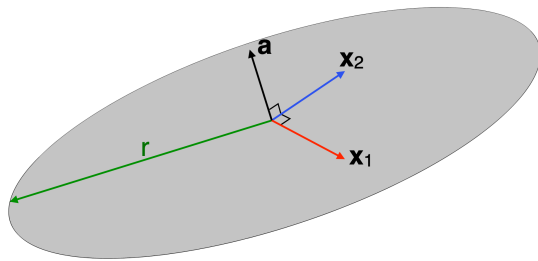


Figure 25. Orthonormal basis ($\mathbf{x}_1, \mathbf{x}_2$) for creating the vertices of the cylindrical shell triangulations. Rotor shaft axis shown in black with radius depicted in green.

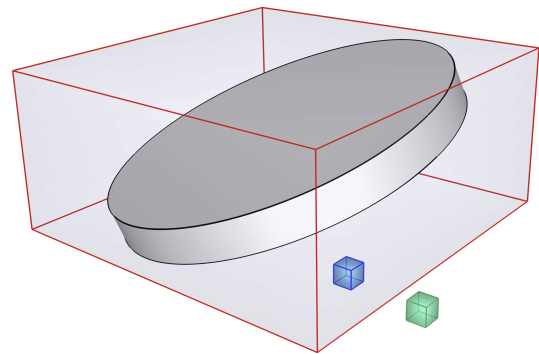


Figure 26. Axis-aligned bounding box (red) of the rotor source volume (gray). Most hexahedra (green) will be outside the source region, but others (blue) pass the bounding box check yet are still outside the source volume

Once these triangulations are constructed, the mapping between Cartesian hexahedra and the blade elements can be computed. We utilize multiple rapid intersection tests to exclude as many hexahedra as possible before computing the actual triangle-cube intersections, as described in Algorithm 1. We denote

the hexahedra that intersect the blade elements as “rotor hexes.”

Algorithm 1: Mapping blade elements to Cartesian hexahedra

Result: Intersected Polyhedra
Input: Cartesian mesh, cylindrical shell triangulations

```

for  $i \leftarrow 0$  to  $N_{hexes}$  do
  for  $j \leftarrow 0$  to  $N_{verts}$  do
    Set hexahedron bounding box boundary value
    Transform vertex to cylindrical coordinates
  end
  Compute hexahedron bounding box in cylindrical coordinates
  if  $hexahedron \cap rotor \text{ bounding box} = \emptyset$  then
    continue
  end
  for  $j \leftarrow 0$  to  $N_{shells}$  do
    for  $k \leftarrow 0$  to  $N_{tris,j}$  do
      if  $hexahedron \cap triangle \text{ bounding box} = \emptyset$  then
        continue
      else
        Append to triangle list ( $L_j$ )
      end
    end
    for  $k \leftarrow 0$  to  $N_{L_j}$  do
      Intersect triangle  $k$  and hexahedron  $i$ 
    end
    Assemble cut polyhedra
    Compute polyhedra centroids and volumes
  end
end
end

```

We begin with tests that are independent of the cylindrical shell triangulations. The first test determines whether a hexahedron intersects the Cartesian axis-aligned bounding box enclosing the entire rotor, as depicted in Figure 26 where it is clear that the green hexahedron cannot be a rotor hex. The blue hexahedron passes this first test but in fact, is not a rotor hex either. Note that a finite thickness rotor sweeps through a cylindrical volume in space, and the bounding box can be computed using only the cylinder’s end caps. The bounding intervals (B_i) of an arbitrarily oriented disk (end cap) are

$$B_i = x_{c,i} \pm r\sqrt{1 - a_i^2} \quad (27)$$

where i denotes the index of a particular Cartesian axis, $x_{c,i}$ is the i -th coordinate of disk center, and a_i is the i -th component of the normalized rotor shaft (cylindrical) axis.²⁶ $\mathbf{x}_i$ is easily computed from the model inputs as

$$\mathbf{x}_i = \mathbf{x}_c + t_r \mathbf{a} \quad (28)$$

where t_r is the semi-thickness of the rotor source volume.

After this we perform a similar test, but this time against a rotor-aligned bounding box. The height of each vertex above the rotor plane is computed with Equation (25), and the minimum and maximum heights are stored. If the maximum vertex height is below the rotor volume’s minimum height, or vice versa, then that hexahedron will not intersect the rotor source region and is excluded, like the blue hexahedron in Figure 26. Given our previous assumptions, the azimuthal bounds are $[0, 2\pi]$ and of course all hexahedra will lie in that interval. In our current implementation, we do not apply a radial coordinate test at this point in the algorithm. Due to the curvature of the radial coordinate, vertex testing is insufficient for determining intersection and more complex tests are deferred until later in the algorithm.

Now that the initial tests have excluded many of the non-intersecting hexahedra, we check if the remaining cells intersect any of the cylindrical shells. First, an axis-aligned bounding box test is performed against each of the shell’s triangles. Bounding box intersecting triangles, denoted as spanning triangles, are accumulated

in a list. Note that a hexahedron with zero spanning triangles can be a rotor hex that is embedded entirely inside of a cylindrical shell. Radial coordinate testing is sufficient to determine if this is the case since the vertex heights have already been analyzed.

After identifying all spanning triangles, we then determine the subset that actually intersect the hexahedron, using algorithms found in the literature.¹² If none of the spanning triangles are intersecting, we again determine if the cell is embedded inside the triangulation. Otherwise, the hexahedron is cut by the intersecting triangles to yield polyhedra whose centroid and volume are then computed.

Finally, with all (cut and uncut) rotor hexes identified, along with their respective centroids and volumes, two components of the force projection function are computed: g_h and g_r . Note that g_ψ cannot be pre-computed because in this approach, the angular extent of the rotor source volume is only known at run-time, as it is determined by the physical timestep Δt .

In the Cart3D flow solver, parallelization and load balancing is accomplished with domain decomposition via space-filling curves. Each outer loop iteration of Algorithm 1 is independent because we do not require any connectivity information between adjacent hexahedra. However, the computational work per iteration is highly variable - some hexahedra will be ruled out by the first bounding box test, while others will pass all tests and intersect multiple cylindrical shells. Therefore, rather than utilizing the flow solver mesh partitioning, we instead use dynamically scheduled loop-level parallelism to accelerate (and load balance) the mapping algorithm. Even the coarsest meshes will typically contain at least 50,000 cells, and we find that a chunk size of 128 yields good parallel scaling while helping mitigate the overhead of dynamic scheduling (Figure 27).

The final step of the rotor preprocessor is writing a binary file containing the rotor inputs and rotor hexahedra metadata, including the cell's Cartesian integer coordinates¹² or "name." The flow solver reads this file on startup and generates a hash table of rotor hexahedra indices keyed by Cartesian name. In a parallel lookup via domain decomposition, each Cartesian cell searches the hash table (by name) to see if it is a rotor hex, and if so, connects each rotor hex to the appropriate index in the flow solver's state and source-term vectors. One small drawback of this approach is that it limits every Cartesian cell to only one associated rotor hex. However, in tandem or intermeshing rotor systems, blades from different rotors will pass through the same Cartesian cell at different points in time, generating multiple rotor hexes for the same Cartesian cell. This is not problematic for our target use case, since aircraft with distributed electric propellers do not typically employ tandem or intermeshing rotor systems that run at constant propeller RPM, although the overlapped lateral tandem design has recently garnered some interest.²⁷ One potential workaround is to replace the rotor hex index with a more complex data structure in the hash table, which would map to all rotor hexes identified for each Cartesian cell. We defer this enhancement to future work and focus on non-overlapped rotors in this paper.

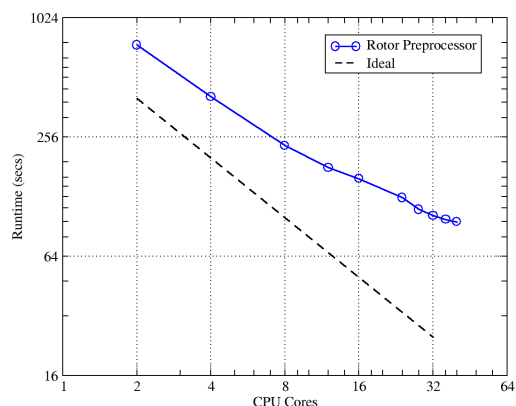


Figure 27. Near-linear strong scaling of rotor preprocessor using dynamically scheduled loop-level parallelism

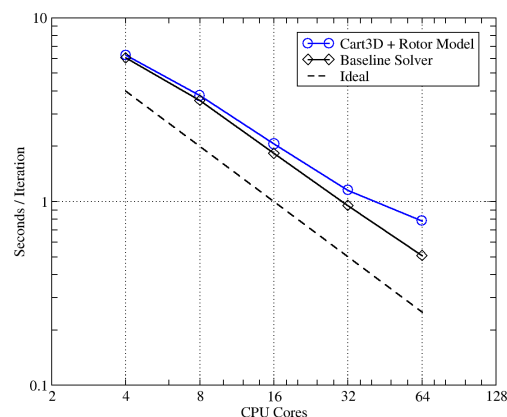


Figure 28. Strong scaling of unsteady rotor model showing near-linear parallel speedup with only a modest increase in work each iteration

3. Flow Solver Partitioning

As mentioned above, the flow solver uses domain decomposition to parallelize the multigrid solver, and the rotor model also needs to be parallelized to fully leverage modern compute architectures. Only a portion of those subdomains will contain rotor hexes, as significant mesh resolution is needed to resolve off-body flow features (e.g. tip vortices) or complex geometry and sharp flow gradients outside the rotor source volume. This would create a sizeable load imbalance between subdomains and severely degrade parallel scalability if rotor hex source terms were computed by their parent partitions.

Instead, we distribute the rotor hexahedra evenly among the partitions. This approximately balances the computational work but introduces another level of communication since one partition will own a rotor hex that accesses a state (or source-term) vector on a different partition. However, this is not unlike the existing multi-grid restriction algorithm, and we show near-linear parallel scalability on up to 64 cores as seen in Figure 28. The slight degradation in scalability at higher core counts is believed to arise from certain datastructures and will be re-architected in a future release.

V. Computational Results

We validate our unsteady body-force rotor model with simulations of the DJI Phantom 3. First, the isolated propeller is simulated in hovering flight for a range of rotational velocities and compared to both existing wind tunnel data²⁸ as well as high-fidelity, discrete blade CFD simulations.¹ After investigating the model’s performance for an isolated propeller, we proceed to study the model’s performance for the full Phantom 3 aircraft in forward flight and compare to the experimental data of Russell et al.

A. Isolated DJI Phantom 3 Propeller in Hover

The two-bladed DJI Phantom 3 propeller airfoils and blade planform were determined via 3-D laser scanning and published along with the performance data in the wind tunnel test report.²⁸ The hover testing was performed in a lab, with the nearest wall 30 feet away to mitigate any issues with flow recirculation. The high-fidelity CFD simulations were performed with Helios, using a wake spacing of about $7\%c_{tip}$ and a timestep ($\Omega\Delta t$) equivalent to 0.25° of rotation. The near-body mesh had approximately 9 million cells, while the off-body Cartesian mesh was about 5 times larger.

In our work, we use a similar time-stepping methodology where the simulations are initialized with a large timestep ($\Omega\Delta t = 30^\circ$) for 10 revolutions to set up the flow structure in the rotor wake, and then run for 3 more revolutions with a finer timestep ($\Omega\Delta t = 1^\circ$). Propeller forces and moments are averaged over the final revolution. A generic centerbody is included to help suppress the root vortices and the mesh has a total of 14.2 million cells. The far-field boundaries are placed 60 rotor radii away and the finest mesh spacing is approximately $5.8\%c_{tip}$. Each simulation took 2.2 hours on a single node of the NASA Electra supercomputer with 2 Intel Xeon Gold 6148 processors.

Figure 29 shows how the propeller hover performance is accurately predicted at all speeds. The model predicts thrust and torque to well within the experimental uncertainty and closely matches the high-fidelity Helios results. The hover wake is visualized in Figure 30 as an iso-surface of q-criterion, where the propeller is rotating at 7000RPM. The propeller “blades” are represented by every rotor hex with a non-zero source term, and the source region thickness is approximately 40% of the tip chord.

B. DJI Phantom 3 Aircraft in Edgewise Flight

The final validation case included in this paper is the simulation of the DJI Phantom 3 aircraft in edgewise forward flight at various pitch angles. Only the aircraft fuselage geometry was included in the model, neglecting any effects from the wind tunnel walls or mounting hardware. We consider cases where the freestream velocity is 40 feet per second, and the propeller rotation speeds are uniform, corresponding to Table 19 in the test report.²⁸

The mesh was generated with the following procedure. First, the fuselage (without rotors) was simulated at the specified freestream condition, generating a mesh adapted for accurate force predictions. This mesh has no knowledge of the expected off-body flow features from the propellers and needs finer cells in those wake regions. However, in forward flight the wake trajectory is not known a priori, especially if there are significant interactions with the fuselage or other rotors. In order to provide reasonable mesh refinement in

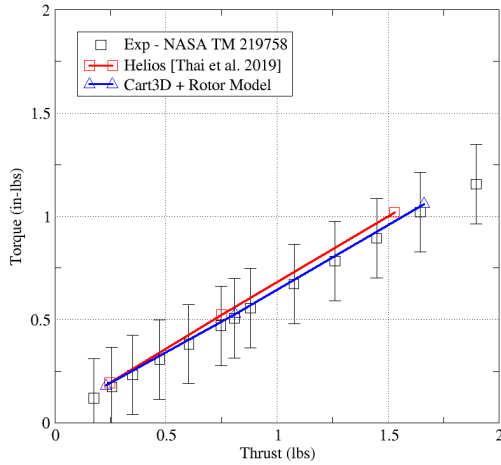


Figure 29. Comparison of integrated torque vs thrust for DJI Phantom 3 propeller in hover to experiment²⁸ and high-fidelity CFD simulation¹

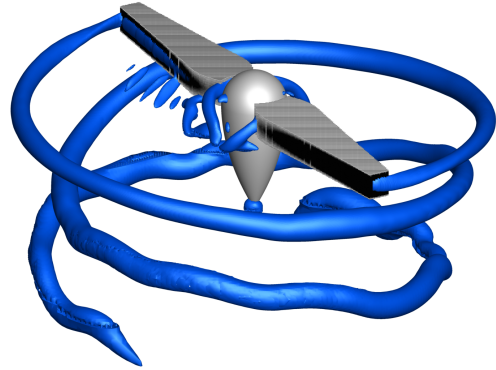


Figure 30. DJI Phantom 3 propeller wake visualization using q-criterion iso-surfaces in hovering flight (7000 RPM)

those regions, the steady-state formulation¹⁰ of the rotor model was employed to iteratively adapt the mesh in regions with large velocity gradients. The final mesh was generated using the last adapted mesh as a reference but included an extra buffer layer to smooth out the mesh interfaces. Typically, the final adapted mesh contained approximately 45 million cells. A cutting plane through the mesh at the port rotor hubs is presented in Figure 31, showing the mesh refinement in the steady-state rotor wake. Figure 32 shows a cutting plane through the rotors' plane of rotation with the rotor hexes shown in orange. All of the blades are assumed to spin in phase and with zero cant angle.

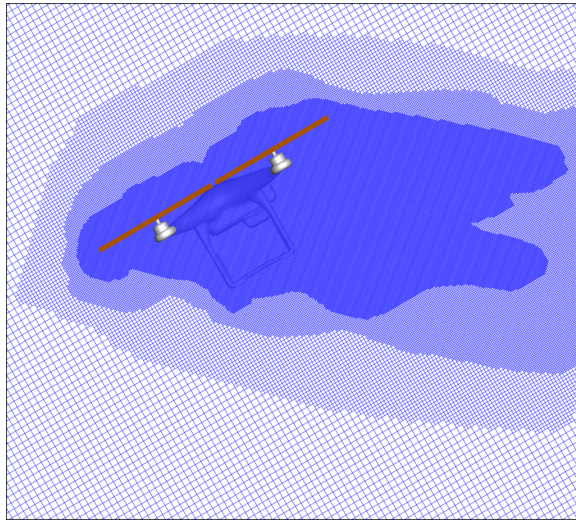


Figure 31. Adapted mesh (blue) on cutting plane through port rotor hub centers with rotor hexes shown in orange for -30° pitch angle.

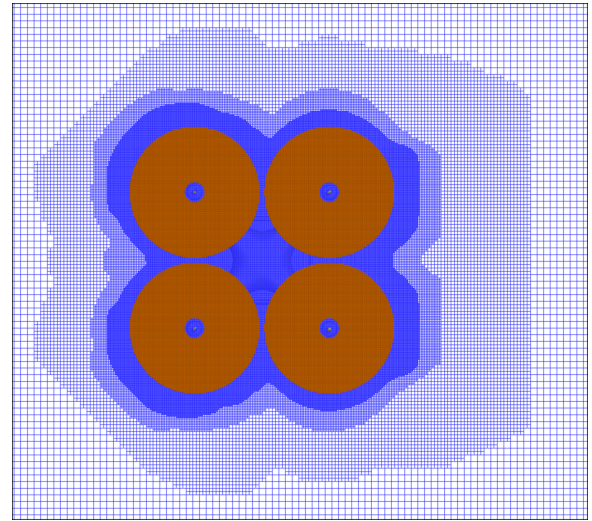


Figure 32. Adapted mesh (blue) on horizontal cutting plane through all rotor hub centers with rotor hexes shown in orange for -30° pitch angle

Finally, the simulation is restarted from the steady-state solution with the rotor model in unsteady mode. A timestep corresponding to 1° of rotor rotation is chosen for all cases, and the simulations reach a time-stationary state in approximately 7 rotor revolutions, converging the density residual over two orders of magnitude in 12 subiterations. On average, the simulations required approximately 65 hours on a single node of the NASA Aitken supercomputer with 2 AMD Epyc 7742 processors. Figures 33 & 34 show side and top views of the rotor wakes, with the tip vortices depicted as q-criterion isosurfaces colored by vorticity

magnitude.

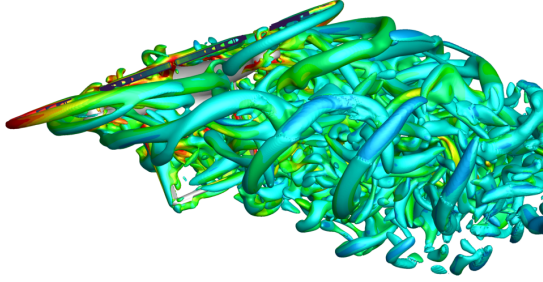


Figure 33. Side view of rotor wake visualized with q-criterion colored by vorticity magnitude (red high) showing complex aerodynamic interactions. $q_\infty = 2$ psf, 6400RPM, -20° pitch angle

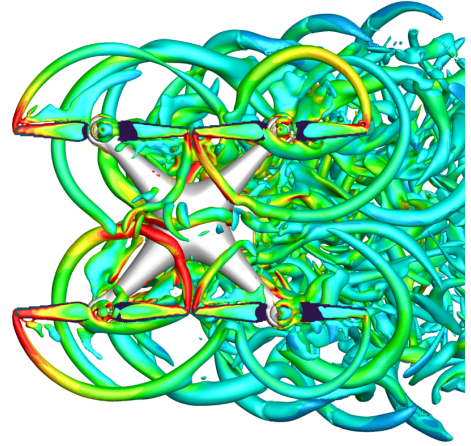


Figure 34. Top view of rotor wake visualized with q-criterion colored by vorticity magnitude (red high) showing complex aerodynamic interactions. $q_\infty = 2$ psf, 6400RPM, -20° pitch angle

The total aircraft lift and drag are shown in Figures 35 & 36, respectively. Lift is accurately predicted at all flight conditions and generally to within the experimental uncertainty, except at -30° pitch angle for low RPM. Unfortunately, only total forces were measured in the experiment, making it difficult to isolate individual rotor or fuselage contributions. Although aircraft drag is generally underpredicted, its variation with pitch angle and propeller RPM is captured by the model. The discrepancy in drag is likely caused by the assumption of inviscid flow and lack of rotor hub drag. Overall, the trends are very promising and give confidence to complex multi-rotor simulations leveraging Cartesian meshes and an IVS-based rotor model.

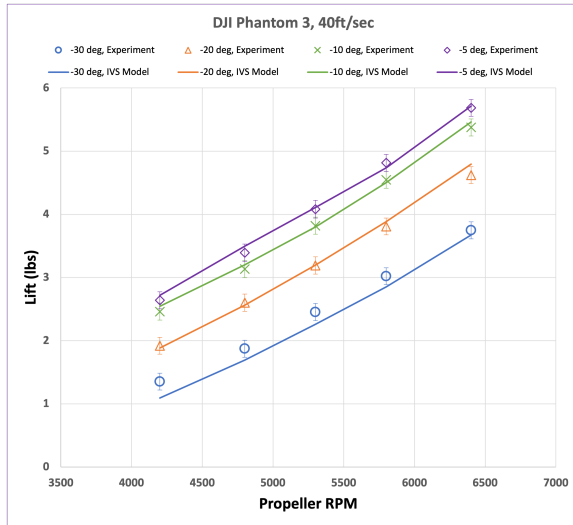


Figure 35. Comparison of lift for DJI Phantom 3 in forward flight with uniform propeller RPM. Experimental data from MUAS wind tunnel test at NASA Ames.

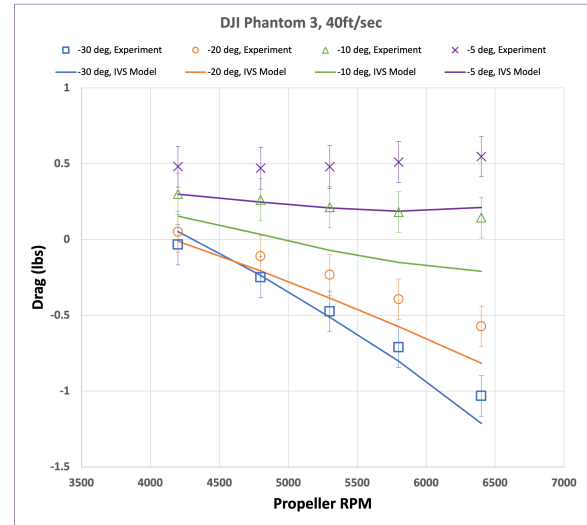


Figure 36. Comparison of drag for DJI Phantom 3 in forward flight with uniform propeller RPM. Experimental data from MUAS wind tunnel test at NASA Ames.

VI. Conclusions

In this paper, we briefly review the recent developments in unsteady body-force rotor models as well as the integral velocity sampling (IVS) technique for estimating the freestream velocity vector. When fully coupled to the flow solver, IVS is shown to predict angle of attack and Mach number with less than 20% error on even the coarsest meshes, and to within a fraction of a percent on only modestly refined grids. Furthermore, its performance is investigated for a variety of freestream flow conditions, and found to be accurate even when the assumptions of incompressible flow and zero airfoil drag are relaxed. The methodology is found to be generally independent of the choice of force projection function if sufficiently refined meshes are employed (1% c) along with source region thickness around 40% c . A straightforward extension of the model to three dimensions revealed the need for C^0 -continuity of the body-force to suppress spurious vorticity generation, and quadratic interpolation provided reasonable loadings when using at least 16 elements across the wingspan. First order tip-loss effects are naturally captured by the method, and an investigation into a simple tip-loss model did not reveal significant improvements to the spanwise lift distribution. Nevertheless, we still obtain accurate thrust and power estimates for an isolated propeller in hover that compare well to high-fidelity CFD simulations as well as experimental data for several propeller speeds. Finally, simulations of a DJI Phantom 3 aircraft in forward flight at multiple pitch angles provide reasonable performance estimates for this complex configuration, closely matching lift and while slightly underpredicting vehicle drag, demonstrating the applicability of this method for multi-rotor aircraft design and analysis.

Acknowledgments

Funding for this work was provided by the NASA Aeronautics Research Mission Directorate's Transformational Tools and Technologies (T³) project. Computer time was provided by the NASA High-End Computing (HEC) program in the NASA Advanced Supercomputing (NAS) Division at Ames Research Center. The authors also thank Philippe Spalart for his helpful conversations in understanding the IVS model.

References

- ¹Thai, A. and Grace, S. M., "Prediction of small quadrotor blade induced noise," *25th AIAA/CEAS Aeroacoustics Conference*, 2019, AIAA 2019-2684.
- ²Ventura Diaz, P. and Yoon, S., "High-fidelity computational aerodynamics of multi-rotor unmanned aerial vehicles," *2018 AIAA Aerospace Sciences Meeting*, 2018, AIAA 2018-1266.
- ³Tadghighi, H., "Simulation of Rotor-body Interactional Aerodynamics - An Unsteady Rotor Source Distributed Disk Model," *Annual Forum Proceedings*, Vol. 1, American Helicopter Society, 2001, pp. 350–361.
- ⁴Boyd Jr, D. D., *Rotor/fuselage unsteady interactional aerodynamics: a new computational model*, Ph.D. thesis, Virginia Tech, 1999.
- ⁵Guntupalli, K. and Rajagopalan, R., "Momentum Source Model for Discrete Blades," *International Powered Lift Conference*, American Helicopter Society, 2010, pp. 314–329.
- ⁶Guntupalli, K., *Development, validation and verification of the Momentum Source Model for discrete rotor blades*, Master's thesis, Iowa State University, 2011.
- ⁷O'Brien Jr, D. M., *Analysis of computational modeling techniques for complete rotorcraft configurations*, Ph.D. thesis, Georgia Institute of Technology, 2006.
- ⁸Koning, W. J., "Wind Tunnel Interference Effects on Tilt Rotor Testing Using Computational Fluid Dynamics," Tech. Rep. CR-2015-219086, NASA, 2016.
- ⁹Wissink, A., Lakshminarayan, V., Jude, D., Jayaraman, B., and Sitaraman, J., "Assessment of Different CFD Download Predictions with Helios," *VFS Transformative Vertical Flight Conference*, 2020.
- ¹⁰Chiew, J. J. and Aftosmis, M., "A Conservative, Scalable, Space-Time Blade Element Rotor Model for Multi-rotor Vehicles," *AHS Aeromechanics Design for Transformative Vertical Flight Conference*, 2018.
- ¹¹Aftosmis, M. J., Berger, M. J., and Adomavicius, G., "A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries," *38th Aerospace Sciences Meeting and Exhibit*, 2000, AIAA 2000-0808.
- ¹²Aftosmis, M. J., Berger, M. J., and Melton, J. E., "Robust and Efficient Cartesian Mesh Generation for Component-Based Geometry," *AIAA Journal*, Vol. 36, No. 6, 1998, pp. 952–960.
- ¹³Jameson, A., "Time dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings," *10th Computational Fluid Dynamics Conference*, Honolulu, HI, Jun 1991, AIAA 1991-1596.
- ¹⁴Melson, N., Sanetrik, M., and Atkins, H., "Time-accurate Navier-Stokes Calculations with Multigrid Acceleration," *Sixth Copper Mountain Conference on Multigrid Methods*, Copper Mountain, CO, Apr 1993.

- ¹⁵Froude, W., “On the elementary relation between pitch, slip, and propulsive efficiency,” *Nineteenth Session of the Institution of Naval Architects*, 1878, pp. 47–57.
- ¹⁶Taylor, D. W., *Resistance of ships and screw propulsion*, The Macmillan Company, 1893.
- ¹⁷Forsythe, J. R., Lynch, E., Polsky, S., and Spalart, P., “Coupled Flight Simulator and CFD Calculations of Ship Airwake using Kestrel,” *53rd Aerospace Sciences Meeting*, 2015, AIAA 2015-0556.
- ¹⁸Churchfield, M. J., Schreck, S. J., Martinez, L. A., Meneveau, C., and Spalart, P. R., “An advanced actuator line method for wind energy applications and beyond,” *35th Wind Energy Symposium*, 2017, AIAA 2017-1998.
- ¹⁹Merabet, R. and Laurendeau, E., “Parametric study on the velocity sampling techniques for the actuator line method in 2D,” *AIAA Scitech Forum*, 2019, AIAA 2019-1797.
- ²⁰Nemec, M. and Aftosmis, M., “Aerodynamic shape optimization using a Cartesian adjoint method and CAD geometry,” *24th AIAA Applied Aerodynamics Conference*, 2006, AIAA 2006-3456.
- ²¹Johnson, W., Silva, C., and Solis, E., “Concept Vehicles for VTOL Air Taxi Operations,” *AHS Aeromechanics Design for Transformative Vertical Flight Conference*, 2018.
- ²²Jha, P. K. and Schmitz, S., “Actuator curve embedding—an advanced actuator line model,” *Journal of Fluid Mechanics*, Vol. 834, 2018.
- ²³Kim, T., Oh, S., and Yee, K., “Improved actuator surface method for wind turbine application,” *Renewable Energy*, Vol. 76, 2015, pp. 16–26.
- ²⁴Martínez-Tossas, L. A., Churchfield, M. J., and Meneveau, C., “Optimal smoothing length scale for actuator line models of wind turbine blades based on Gaussian body force distribution,” *Wind Energy*, Vol. 20, No. 6, 2017, pp. 1083–1096.
- ²⁵Chiew, J. J., *Rapid Computational Aerodynamic Analysis for Multi-Rotor Aircraft*, Ph.D. thesis, Stanford University, 2020.
- ²⁶Quilez, I., <https://www.iquilezles.org/www/articles/diskbbo/diskbbo.htm>, Accessed: 2021-06-11.
- ²⁷Silva, C., Johnson, W., and Solis, E., “Multidisciplinary Conceptual Design for Reduced-Emission Rotorcraft,” *AHS Aeromechanics Design for Transformative Vertical Flight Conference*, 2018.
- ²⁸Russell, C. R., Willink, G., Theodore, C., Jung, J., and Glasner, B., “Wind tunnel and hover performance test results for multicopter UAS vehicles,” Tech. Rep. TM-2018-219758, NASA, 2018.