

SciTech Forum 2021 — Extended Abstract

Unsteady Mixed-fidelity Aerodynamics Solver for Maneuvering Multirotor Aircraft

Eduardo J. Alvarez* and Andrew Ning†

Brigham Young University, Provo, Utah, 84602

The ability to accurately and rapidly assess maneuvering multirotor aircraft is a current shortcoming and bottleneck in the design of next-generation aircraft. For instance, current models used in aircraft design fail to predict and assess configurations that may lead to the wake of a rotor impinging on another rotor or wing during the transition maneuver. In this study we present a mixed-fidelity simulation framework based on the viscous vortex particle method (VPM) for simulating maneuvering multirotor aircraft without the prohibitive computational cost of conventional mesh-based computational fluid dynamics (CFD) methods. The viscous VPM is a mesh-free CFD method for the numerical solution of the Navier-Stokes equations in their vorticity form, which is especially well fitted for resolving wake dynamics. Wings, body, and rotating blades are modeled in our framework through a combination of panel, vortex lattice, lifting line, and blade elements, while wake dynamics are solved through VPM. The framework is released as the opensource code FLOWUnsteady. Validation of predicted rotor-on-rotor, rotor-on-wing, wing-on-rotor, and wing-on-wing aerodynamic interactions is presented in this study, followed by the simulation of the eVTOL maneuver of a tilt-wing multirotor aircraft.

I. Introduction

Electric vertical takeoff and landing (eVTOL) aircraft is a newfound technology that is projected to grow into a \$1.5 trillion industry of urban air mobility by the year 2040.¹ Since the first foresight of this nascent technology² in 2010, the list of known eVTOL aircraft concepts has grown from a handful to more than 250 today.² This list includes aerospace giants like Airbus and Boeing, established car makers like Toyota and Hyundai, and tech eminence like Uber and Alphabet. The excitement around eVTOL arises from a newly opened design space with electric aircraft that can incorporate multiple rotors to takeoff and land vertically for inner-city travels, capitalizing on the dream of a “flying car”. However, these novel aircraft configurations pose technical challenges that still remain to be solved.³ A strong noise signature⁴ and a complicated transition maneuver^{5,6} are examples of the challenges encountered in eVTOL aircraft, both stemming in some

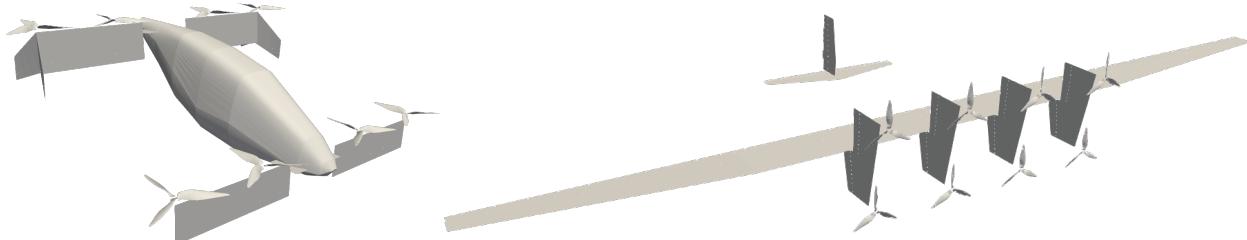


Figure 1: Examples of multirotor aircraft modeled in FLOWUnsteady. Left: A³ Vahana eVTOL aircraft. Right: Makani M600 wind-harvesting aircraft.

*Doctoral Candidate, Department of Mechanical Engineering, ealvarez@byu.edu, AIAA Member.

†Associate Professor, Department of Mechanical Engineering, aning@byu.edu, AIAA Senior Member.

²<https://evtol.news/aircraft/>, accessed May 23rd, 2020.

degree from the aerodynamic interactions between the multiple rotors and lifting surfaces.⁷ These interactions are not well understood, are not captured through conventional design tools, and need to be addressed in the early stages of design. For instance, current models used in aircraft design fail to predict and assess configurations that may lead to the wake of a rotor impinging on another rotor or wing during the transition maneuver. As an example of multirotor configurations, Fig. 1 shows two examples of next-generation electric aircraft. These interactions are conventionally analyzed through mesh-based computational fluid dynamics (CFD) methods,^{8–10} but the resolution of wake dynamics in a mesh require high-order numerical schemes with computational costs that make them prohibitive for design space exploration. Furthermore, these interactions are common across quadrotor, tilt-rotor, tilt-wing, and distributed propulsion concepts making use of multiple rotors. Thus, the ability to accurately and rapidly assess maneuvering multirotor aircraft is a current shortcoming and bottleneck in the design of next-generation aircraft.

In this paper we present a mixed-fidelity simulation framework based on the viscous vortex particle method (VPM) for simulating maneuvering eVTOL aircraft without the prohibitive computational cost of conventional CFD methods. The viscous VPM is a mesh-free CFD method for the numerical solution of the Navier-Stokes equations in their vorticity form, which is especially well fitted for resolving wake dynamics. This approach is a direct numerical simulation that efficiently preserves vortical structures in a Lagrangian scheme, eliminates the complexities of mesh generation, is absent of the numerical dissipation associated with mesh-based CFD, and is about two orders of magnitude faster than conventional CFD methods. Wings, body, and rotating blades are modeled in our framework through a combination of panel, vortex lattice, lifting line, and blade elements, while wake dynamics are solved through VPM. Validation of predicted rotor-on-rotor, rotor-on-wing, wing-on-rotor, and wing-on-wing aerodynamic interactions is hereby presented, followed by a simulation of the eVTOL maneuver of a tilt-wing multirotor aircraft.

II. Modeling Methodology

An unsteady mixed-fidelity simulation engine has been developed, called FFlight, Optimization, and Wind Unsteady, or FLOWUnsteady¹. This engine integrates a viscous vortex particle, vortex lattice, lifting line, blade element, and three-dimensional panel code into a single framework, with all codes being in-house developed. The implementation of the vortex particle method is described in Section II.A, and the framework and models developed for FLOWUnsteady are described in Section II.B. With this framework, we aim to capture all rotor-rotor, rotor-wing, and wing-wing interactions encountered by maneuvering multirotor aircraft.

II.A. Viscous Vortex Particle Method

FLOWUnsteady uses the VPM code FLOWVPM, previously developed by the authors.¹¹ The method and implementation are herebelow briefly described. For a more detailed description of the VPM theory and implementation, the reader is referred to previous publications by the authors.^{11–13}

II.A.1. Governing Equations

A viscous, incompressible flow dominated by vorticity can be resolved by formulating the Navier-Stokes equations into their vorticity form. This is done by taking the curl over the original momentum equation

$$\nabla \times \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = \nabla \times \left(-\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} \right),$$

leading to an expression that is not dependent on the pressure field:

$$\frac{D\boldsymbol{\omega}}{Dt} = (\boldsymbol{\omega} \cdot \nabla) \mathbf{u} + \nu \nabla^2 \boldsymbol{\omega}. \quad (1)$$

The material derivative in Eq. (1) and the material-conservative nature of the vorticity make the $\boldsymbol{\omega}$ field especially fit for a Lagrangian description. In order to do so, the field is discretized into vortex particles¹⁴

¹Code available at <https://github.com/byuflowlab/FLOWUnsteady>

through a radial basis function interpolation of basis ζ_σ and coefficients $\boldsymbol{\Gamma}_p$:

$$\boldsymbol{\omega}(\mathbf{x}, t) \approx \sum_p \boldsymbol{\Gamma}_p(t) \zeta_\sigma(\mathbf{x} - \mathbf{x}_p(t)). \quad (2)$$

Each particle represents a volume of fluid that travels with the local velocity as in Eq. (3), where \mathbf{x}_p is the position of the p -th particle. The coefficient $\boldsymbol{\Gamma}_p$, termed *vectorial circulation* or *vortex strength*, is defined as $\boldsymbol{\Gamma}_p \equiv \int_{\text{vol}_p} \boldsymbol{\omega} dV$ and approximated as $\boldsymbol{\Gamma}_p \approx \boldsymbol{\omega}_p \text{vol}_p$, where $\boldsymbol{\omega}_p$ is the vorticity associated with the p -th particle of volume vol_p . Applying this particle discretization to Eq. (1), it is obtained that the vectorial circulation of each particle evolves as in Eq. (4), where the first right-hand-side term is the vortex stretching, while the second represents a scheme for modeling the viscous diffusion $\nu \nabla^2 \boldsymbol{\omega}$.

$$\frac{d}{dt} \mathbf{x}_p(t) = \mathbf{u}(\mathbf{x}_p(t), t) \quad (3)$$

$$\frac{d}{dt} \boldsymbol{\Gamma}_p(t) = (\boldsymbol{\Gamma}_p(t) \cdot \nabla) \mathbf{u}(\mathbf{x}_p(t), t) + \left. \frac{d}{dt} \boldsymbol{\Gamma}_p(t) \right|_{\text{visc}} \quad (4)$$

II.A.2. Velocity Kernel

By the Helmholtz Decomposition Theorem, an incompressible velocity field \mathbf{u} dominated by vorticity is described by some vector potential field ψ as $\mathbf{u}(\mathbf{x}, t) = \nabla \times \psi(\mathbf{x}, t)$. Then, from the definition of the vorticity field ($\boldsymbol{\omega} = \nabla \times \mathbf{u}$), we get the three-dimensional unbounded Poisson's problem

$$\nabla^2 \psi = -\boldsymbol{\omega},$$

which can also be rewritten using the particle approximation from Eq. (2) as

$$\nabla^2 \psi(\mathbf{x}) \approx - \sum_p \boldsymbol{\Gamma}_p \zeta_\sigma(\mathbf{x} - \mathbf{x}_p).$$

The Poisson's problem yields that the velocity induced by the field of vortex particles is calculated as

$$\mathbf{u}(\mathbf{x}) = \sum_p g_\sigma(\mathbf{x} - \mathbf{x}_p) \mathbf{K}(\mathbf{x} - \mathbf{x}_p) \times \boldsymbol{\Gamma}_p, \quad (5)$$

where g_σ is a normalized smoothing function calculated from the chosen basis function ζ_σ , and $\mathbf{K}(\mathbf{x}) = -\frac{\mathbf{x}}{4\pi \|\mathbf{x}\|^3}$ is the singular Newtonian kernel resulting from the three-dimensional Green's function of the unbounded Poisson's problem. This is implemented in FLOWVPM with the Gaussian basis function

$$\zeta_\sigma(\mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{3/2}} \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2}\right).$$

II.A.3. Viscous Diffusion and Other Schemes

Viscous diffusion in FLOWVPM is incorporated into the governing equation Eq. (4) through the core spreading method¹⁵ coupled with the radial basis function (RBF) interpolation approach developed by Barba,^{16–18} which avoids the need for particle splitting while allowing the viscous VPM to be a truly meshless method. In the core spreading method, a Gaussian radial basis allows the viscous diffusion term in Eq. (1) to be solved by spreading the smoothing radius σ as

$$\frac{d\sigma^2}{dt} = 2\nu.$$

Core sizes σ are reset to their initial value after they have overgrown, while new vectorial circulations $\boldsymbol{\Gamma}_p$ are calculated through an RBF interpolation preserving the vorticity field $\boldsymbol{\omega}$.

In the FLOWVPM implementation, vortex stretching is solved in the transposed scheme as explained in Winckelmann's thesis,¹⁹ and the divergence of the vorticity field is treated through the relaxation scheme

developed by Pedrizzetti.²⁰ The time integration of the governing equations is done through a low-storage third-order Runge-Kutta scheme.²¹

In order to speed up the framework, the fast multipole method (FMM) is used for the computation of the governing equations and kernel, Eqs. (3) to (5). The FMM approximates pairwise interactions of particles through spherical harmonic functions,^{22,23} reducing the original computational complexity $\mathcal{O}(N^2)$ to a problem of order $\mathcal{O}(N)$, where N is the number of particles. The computation of the vortex stretching is performed through an efficient complex-step derivative approximation as explained in previous work,¹³ implemented in a modified version of the open-source, parallelized FMM code ExaFMM.^{24–26}

II.B. FLOWUnsteady Framework

II.B.1. State Variables and Inputs

FLOWUnsteady uses the three-dimensional geometry of the vehicle as spatial state variables (referred to as G , which is the cloud of three-dimensional points of the surfaces of the vehicle), along with the translational velocity of the vehicle \mathbf{V} and rotational velocity $\boldsymbol{\Omega}$. Using components of the geometry, the user defines sets of tilting systems (control surfaces) that tilt relative to the vehicle's orientation, rotor systems with independent RPMs, and wake-shedding systems. The time-dependent control inputs to the framework are tilting-system deflection angles and rotor-system RPMs. Additionally, the user can either prescribe the kinematics of the vehicle throughout a maneuver, or let the dynamics of the vehicle (i.e., aerodynamic, propulsion, and gravitational forces) drive the kinematics.

II.B.2. Computation Workflow

FLOWUnsteady works through multiple layers of encapsulated computation, as shown in Fig. 2. The outer layer is the VPM solver run by FLOWVPM which drives the time stepping, shown in Fig. 2 as the green block. In step (1), solid boundaries are introduced into the flow field as embedded particles that preserve the circulation of each surface and move with the surfaces. In steps (2) and (3), the particle field is updated according to convection, stretching, and viscous diffusion terms in Eqs. (3) and (4), thus solving the Navier-Stokes equations in their vorticity form (Eq. (1)), and the Pedrizzetti relaxation is applied. In step (4), all particle cores are resized if the particles have over-diffused. In step (5), FLOWVPM passes the particle field to an arbitrary user-defined function `runtime_function()` that returns a flag indicating whether to break the iteration and end the simulation. In step (6), the particle field is outputted as a file in Visualization Toolkit (VTK) format for visualization and post-processing. The solver keeps iterating in time until it reaches the end time t_f or if flagged to break in step (5).

The second layer of encapsulation is the aerodynamics engine, shown in Fig. 2 as the blue block, which is seen and called by the VPM solver as a user-defined function (step (5) in green block). In step (1), the

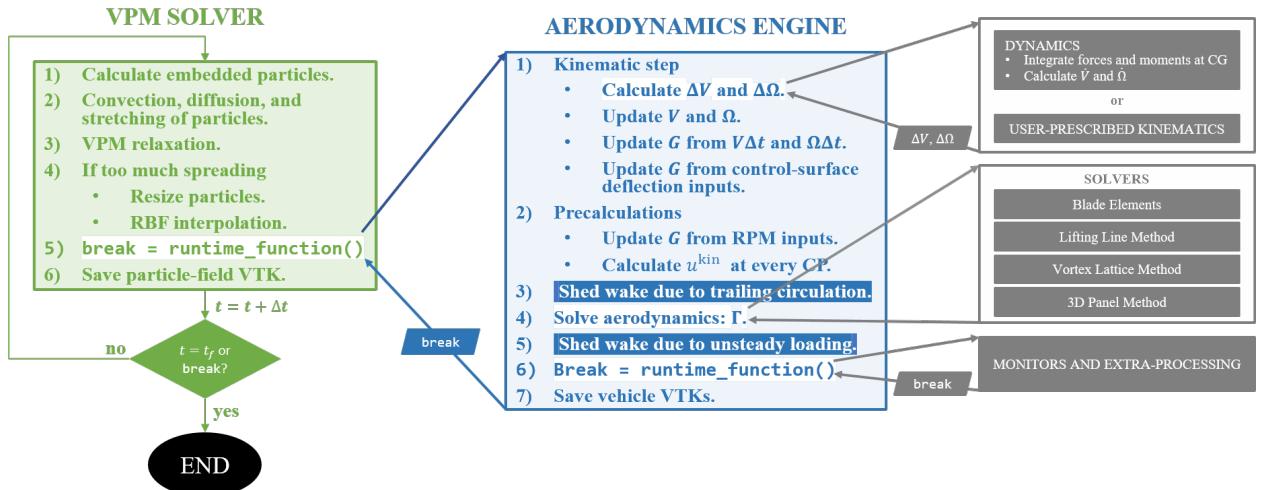


Figure 2: Flowchart of FLOWUnsteady framework.

vehicle takes a kinematic step where the geometry is translated and rotated according to its translational and rotational velocity. First, velocities changes— $\Delta \mathbf{V}$ and $\Delta \boldsymbol{\Omega}$ —are either calculated from the kinematics prescribed by the user or directly from the dynamics of the vehicle. The state variables \mathbf{V} and $\boldsymbol{\Omega}$ are then updated, and the whole geometry is translated and rotated according to \mathbf{V} and $\boldsymbol{\Omega}$. Tilting systems (control surfaces) are deflected according to control inputs. In the precalculation step (2), rotors are rotated according to RPM control inputs and the kinematic velocity \mathbf{u}^{kin} of each control point (CP) is calculated². In step (3), vortex particles are shed from trailing edges of all wake-shedding systems stemming from the trailing circulation obtained in the previous time step. In step (4), the velocity induced by the VPM wake is added to the kinematic velocity \mathbf{u}^{kin} and freestream \mathbf{u}_∞ to obtain the effective velocity \mathbf{u}^{eff} at every control point. The aerodynamic solvers are then sequentially called to cancel the normal component of \mathbf{u}^{eff} along all surfaces. Thus, all aerodynamic solvers are loosely coupled with each other and VPM as the simulation steps in time, solving for the surface circulation Γ needed to satisfy the no-flow-through boundary condition at each control point. In step (5), vortex particles are shed from trailing edges of all wake-shedding systems from the unsteady loading obtained from the change in circulation. In step (6), the engine passes the particle field and vehicle to an arbitrary user-defined function `runtime_function()` that returns a break flag. Through this function, the user has access to the state of the simulation at every step, making it possible to define monitors, add processing functions, or alter the state of the simulation in the loop. Also, notice that this `runtime_function()` defines a third layer of encapsulation where the user can connect an aeroacoustics solver, structural solver, controls framework, or any other solver to FLOWUnsteady. Finally, in step (7) the vehicle is outputted as VTKs for visualization and post-processing.

The schemes for embedded particles, trailing and unsteady circulation wake shedding, and the no-flow-through condition are further explained in the following two sections.

II.B.3. Rotor Model

The rotor model used in FLOWUnsteady is formulated in the same numerical scheme of the VPM, capturing unsteady dynamics and viscous effects while benefiting from the computational speed and linear scaling of FLOWVPM.¹¹ The model is here briefly described, while an in-depth description and a thorough convergence and validation study are given in previous work by the authors.¹³ Each rotor blade is modeled as a rotary lifting surface, and the physics of interest are broken down into three aspects: load distribution, blade-induced velocity, and wake-induced velocity. The load distribution is calculated through blade elements, and is used for deriving the circulation along the lifting surface. In turn, the blade-induced velocity is obtained from the circulation distribution by embedding vortex particles along the surface that preserve such circulation, as shown in Fig. 3. As the blade moves, vorticity is shed off the trailing edge as free particles stemming from both unsteady loading and trailing circulation. Thus, blade and wake-induced velocities are all computed

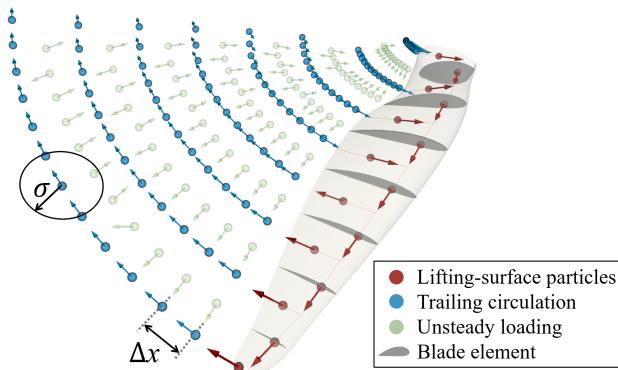


Figure 3: Blade and wake deployment in the rotor model. Particles are colored by their respective source of vorticity; arrows indicate the direction of vortex strength Γ_p .

²Notice that \mathbf{u}^{kin} is different than \mathbf{V} . The latter is the translational velocity of the vehicle, while \mathbf{u}^{kin} is the velocity of each control point in the vehicle resulting from translation and rotation of the vehicle in addition to control-surface tilting and rotor rotation of the control point if the point is in a tilting or rotor system, or both. This is calculated from the changes in G , as shown in Eq. (7).

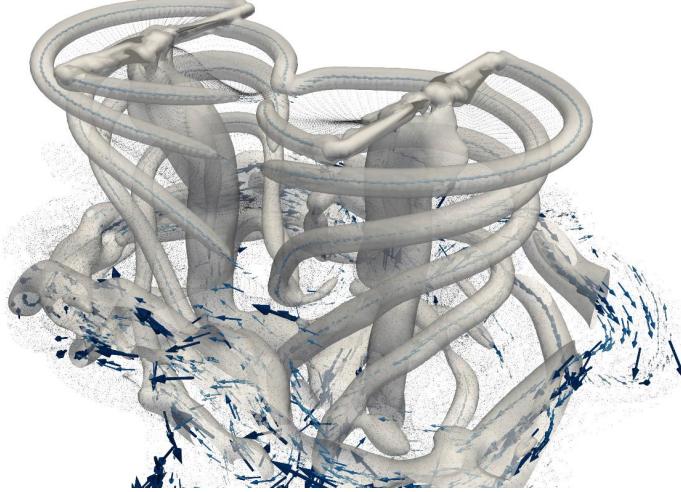


Figure 4: Simulation of hovering rotors, with particles and vortex strengths shown as dots and arrows respectively, superimposed with isosurfaces of the vorticity field.

through the fast-multipole approximation achieving an efficient scaling of the simulation to the order of millions of particles, while all unsteady dynamics are resolved as the VPM steps in time. In this scheme, rotor-on-rotor interactions are captured as all blades are affected by each other and wake mixing. Fig. 4 shows the simulation of two counter-rotating rotors in hover, exemplifying the unsteady dynamics and wake mixing during rotor-on-rotor interactions captured with this rotor model.

II.B.4. Wing and Body Models

FLOWUnsteady integrates three in-house-developed aerodynamics solvers for the modeling of lifting and non-lifting surfaces (wings and bodies, respectively). These solvers implement a lifting line method, a vortex lattice method, and a three-dimensional panel method. The role of every solver is to cancel the normal component of the effective velocity \mathbf{u}^{eff} experienced by every control point in the geometry. It must be pointed out that, in the unsteady framework of FLOWUnsteady, none of the solvers include the rigid-wake models that are usually associated with these methods, but rather the wake models are replaced with the VPM.

The effective velocity $\mathbf{u}_i^{\text{eff}}$ experienced by the i -th control point of position \mathbf{x}_i includes the freestream velocity \mathbf{u}_∞ , the wake-induced velocity \mathbf{u}_{VPM} calculated through the VPM, and the kinematic velocity $\mathbf{u}_i^{\text{kin}}$ of the control point,

$$\mathbf{u}_i^{\text{eff}}(t) = \mathbf{u}_\infty + \mathbf{u}_{\text{VPM}}(\mathbf{x}_i, t) + \mathbf{u}_i^{\text{kin}}(t). \quad (6)$$

The kinematic velocity of the control point³ is calculated from the change in position between time steps as

$$\mathbf{u}_i^{\text{kin}}(t) = -\frac{\mathbf{x}_i(t) - \mathbf{x}_i(t - \Delta t)}{\Delta t}. \quad (7)$$

Finally, the velocity \mathbf{u}_{ind} induced by the superposition of all rotor blades, wings, and bodies is used to solve for the circulation of each wing and body as to satisfy the no-flow-through condition at every control point

$$(\mathbf{u}_{\text{ind}}(\mathbf{x}_i, t) + \mathbf{u}_i^{\text{eff}}(t)) \cdot \hat{\mathbf{n}}_i = 0, \quad (8)$$

where $\hat{\mathbf{n}}_i$ is the vector normal to the surface at the i -th control point. Identical to the rotor model, velocities induced by all surfaces are reintroduced into the governing equations of the VPM by placing embedded particles that preserve the circulation of every surface, while the VPM wake is shed from all trailing edges due to trailing and unsteady circulation. Fig. 5 shows the simulation of a swept wing in cruise with two

³It must be mentioned that this kinematic velocity is also accounted for in the rotor model and is calculated identically at every blade element.

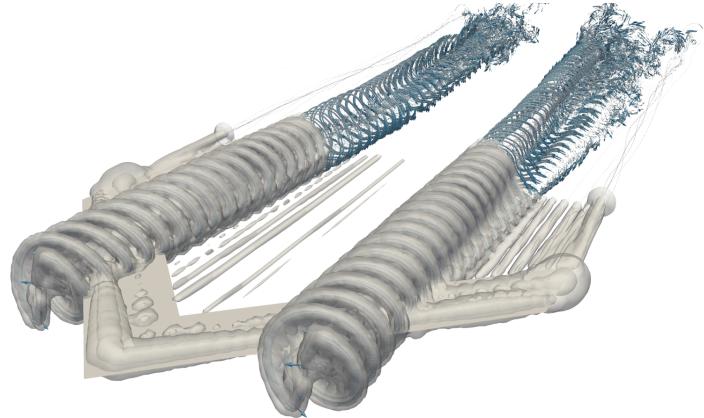


Figure 5: Simulation of a blown wing in cruise, with vortex strengths shown as arrows and superimposed with isosurfaces of the vorticity field.

co-rotating rotors, exemplifying the unsteady dynamics and wake mixing due to rotor-wing interactions captured through this scheme. Here the wing is modeled as a vortex lattice of only one chord-wise element, where the placement of embedded particles along the wing is evident through the concentration of vorticity at the quarter-chord line.

In this scheme, rotor-on-wing interactions are captured through the term \mathbf{u}_{VPM} in Eq. (6), accounting for the effects of a rotor wake impinging on a wing and the velocity induced by each blade included as embedded particles. Similarly, wing-on-rotor interactions are captured through the VPM-induced velocity given to the blade elements by both free and embedded particles, accounting for the effects of a rotor operating in the wake of a wing and freestream deflection in the vicinity of the wing. Wing-on-wing interactions are captured through the term \mathbf{u}_{VPM} in Eq. (6), accounting for the effects of a wing wake impinging on another wing, and the freestream deflection in the vicinity of the wings included in Eq. (8) through \mathbf{u}_{ind} . Thus, in this framework all unsteady interactional aerodynamics are resolved as the vehicle moves in space performing an arbitrary maneuver.

III. Preliminary Verification and Validation

In this section we breakdown and validate each phenomenon of interest: steady loads of an isolated wing are validated in Section III.A, unsteady dynamics of a maneuvering wing are verified in Section III.B, wing-on-wing interactions are verified in Section III.C, and validation of rotor-on-rotor interactions is shown in Section III.D. In the current state of this work, the results shown in Section III.B and Section III.C portray our verification efforts rather than a decisive validation of the models, but a more rigorous validation will be included in the final version of this manuscript. The validation of rotor-on-wing and wing-on-rotor interactions is also a matter of ongoing work.

III.A. Isolated Wing

In order to validate the lifting-surface model, a 45-deg swept-back wing of aspect ratio 5 is simulated matching the experimental conditions reported by Weber and Brebner.²⁷ In this simulation, the wing moves at a constant velocity of 49.7 m/s and 4.2° angle of attack in a still freestream, and is modeled as a vortex lattice of only one chord-wise element. Fig. 6 shows the rolled-up wake by the end of the simulation, while Fig. 7 shows the predicted lift coefficient C_L converging to experimental value (black line) as the wake is being deployed. FLOWUnsteady predicts lift and drag coefficients (C_L and C_D) of 0.2381 and 0.0024, respectively, meanwhile Weber and Brebner report C_L and C_D values of 0.238 and 0.005, respectively. The error between predicted and measured C_L is smaller than 0.5%, meanwhile C_D is largely under-predicted, presumably due to the effects of viscous drag not captured in the vortex-lattice solver. Fig. 8 shows the sectional lift and induced drag along the semi-span of the wing compared to chord-wise loads integrated from Pitot-tube measurements,²⁷ evidencing reasonable agreement between predicted and measured distributions.

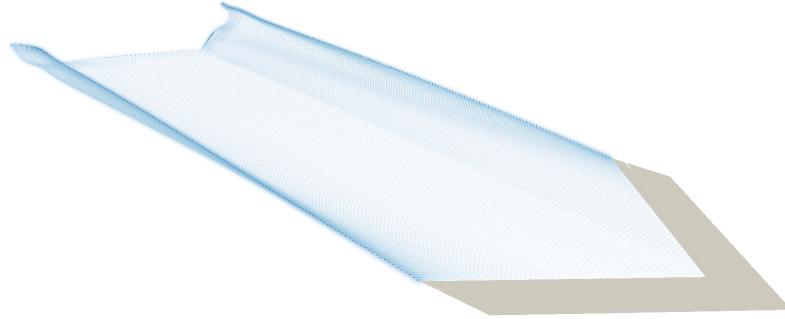


Figure 6: Simulation of isolated, 45-deg swept-back wing.

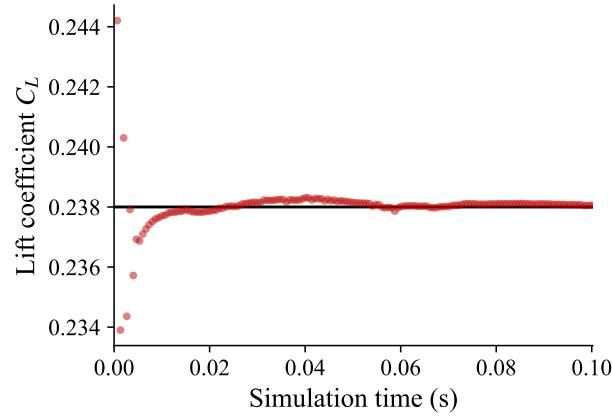


Figure 7: Time history of C_L in simulation of isolated 45-deg swept-back wing (red markers), compared to experimental C_L ²⁷ (black line).

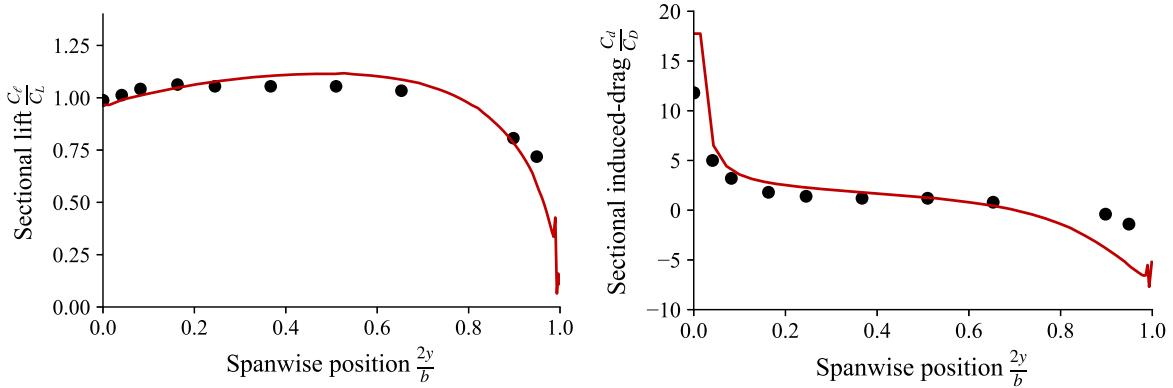


Figure 8: Normalized lift and induced-drag distribution (red lines) at end of simulation of isolated 45-deg swept-back wing, compared to experimental measurements²⁷ (black markers).

III.B. Maneuvering Wing

In order to verify the capability of FLOWUnsteady to capture unsteady loads during maneuvers, the swept-back wing from Section III.A is now placed in heaving motion as it translates in space. In addition to the 4.2° geometric angle of attack, the wing is tilted at an angle θ about the nose, with θ changing in time as

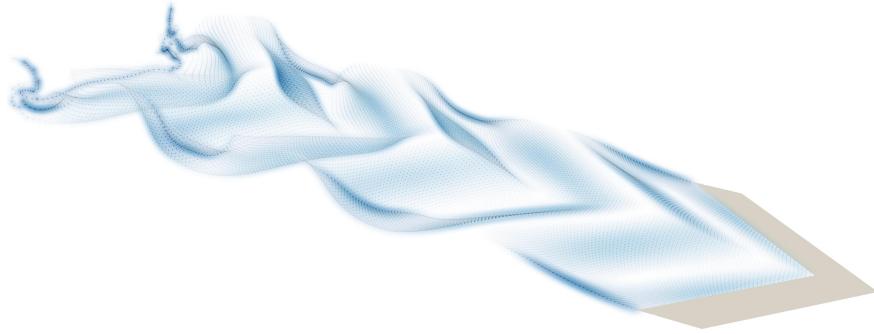


Figure 9: Simulation of wing in heaving maneuver.

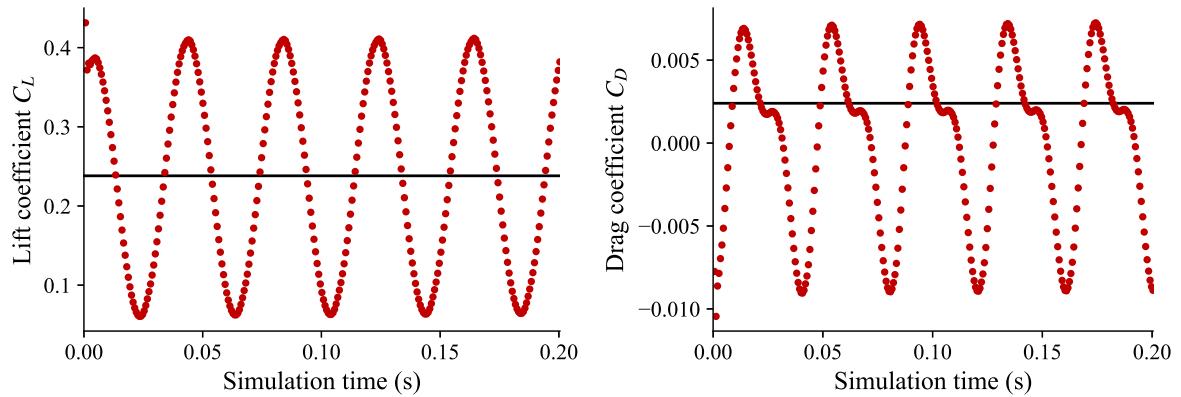


Figure 10: Time history of C_L and C_D during heaving maneuver (red markers), compared to expected C_L and C_D without heaving (black lines).

$\theta(t) = 1.4^\circ \sin(157t)$. Fig. 9 shows the wake after about three heaving cycles ($t = 0.13$ s), while Fig. 10 shows the time history of lift and drag coefficients throughout the simulation (five heaving cycles). Black lines denote the converged C_L and C_D values from the non-heaving simulation obtained in Section III.A. The drag history indicates some propulsive characteristics due to the heaving motion, achieving a negative drag (or propulsive thrust) in parts of the heaving stroke even though the lift stays always positive. This behavior is characteristic of the unsteady dynamics encountered in flapping flight, which lends confidence at the capacity of FLOWUnsteady to successfully capture unsteady dynamics during this maneuver.

III.C. Wing-on-wing Interactions

In order to verify the capability of FLOWUnsteady to capture wing-on-wing interactions, the swept-back wing from Section III.A is now placed in a tandem configuration with a canard in heaving motion. The swept-back wing is at a constant 4.2° angle of attack, while the canard is tilted at an angle θ changing in time as $\theta(t) = 2.1^\circ \sin(157t)$. The canard has a span two times smaller than the span of the main wing, and is placed upstream at a distance of 75% the length of the main wing's span. The vehicle (main wing and canard) travels at a constant velocity of 49.7 m/s in a still freestream. Fig. 11 shows the mixed wake and load distribution along the main wing in two instants as the wake of the canard impinges on the main wing. Fig. 12 shows that, as the wake is being deployed ($t < 0.3$ s), the main wing's C_L initially converges to the C_L of the isolated wing obtained in Section III.A. Once the wake of the heaving canard reaches the main wing ($t \approx 0.4$ s), the lift coefficient starts to oscillate at the same frequency that the canard's heaving motion.

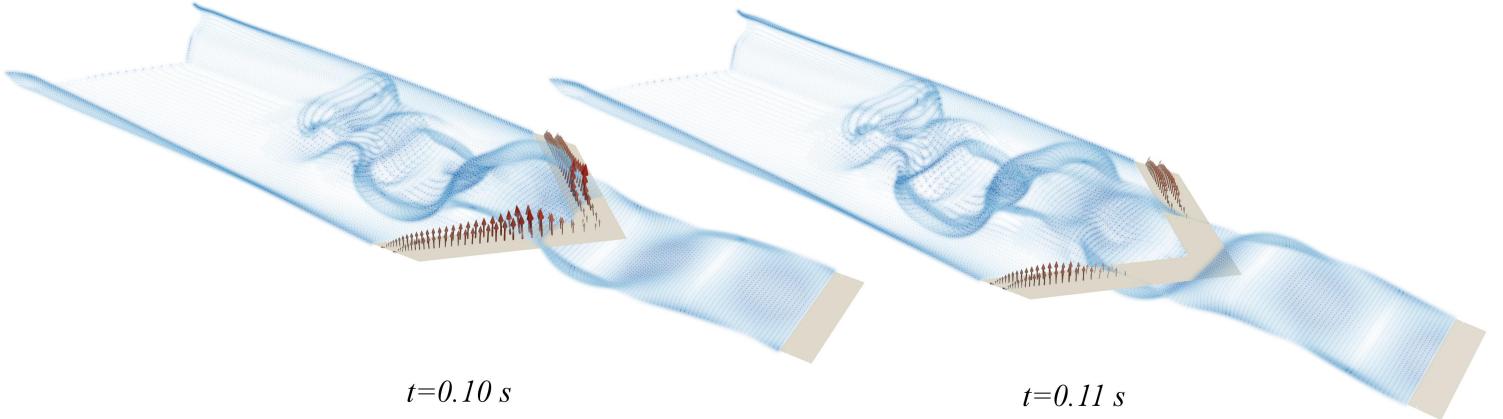


Figure 11: Simulation of tandem configuration with a heaving canard. Arrows shows the loading distribution along the main wing.

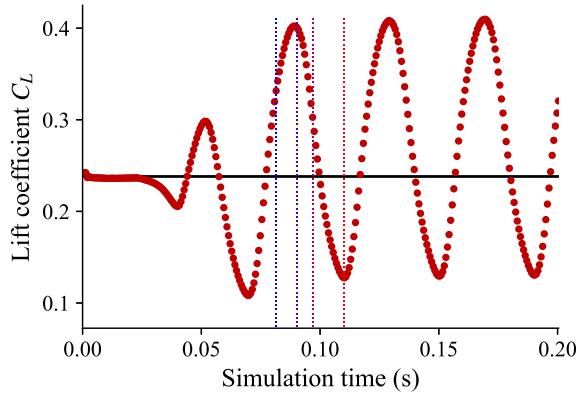


Figure 12: Time history of main wing's C_L (red markers) in heaving-canard simulation, compared to the C_L of the isolated wing (black line).

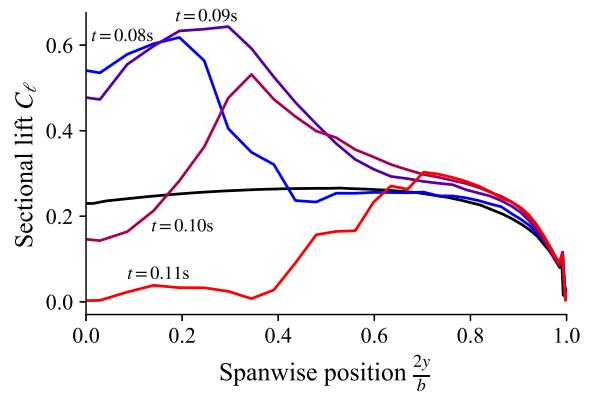


Figure 13: Lift distribution along the main wing at different instants of the simulation, compared to the distribution of the isolated wing (black line).

Fig. 13 shows the lift distribution along the semi-span of the main wing at the four instants indicated as vertical dotted lines in Fig. 12, and they are superimposed with the lift distribution of the isolated wing (black line) obtained in Section III.A. Here we see that the sectional lift stays relatively constant on the outboard portion of the span ($2y/b > 0.7$), while inboard it alternates between enhancement and attenuation as the wake impinges on the wing. These oscillations lend confidence at the capability of FLOWUnsteady to successfully capture wing-on-wing interactions.

III.D. Rotor-on-rotor Interactions

Previous work by the authors¹³ has shown excellent agreement between experimental measurements and predictions using the rotor model described in Section II.B.3 across Reynolds numbers from hover to moderately high advance ratios. The wake of an isolated DJI rotor in hover is shown in Fig. 14 (left), where features of the topology can be clearly identified: a well-defined structure in the near field ($z < 0.5D$), an onset of leapfrogging at $z \approx 0.5D$, and a transition into turbulent breakdown at $z \approx 1.0D$. The multirotor configuration is also shown in Fig. 14 (right), where a region of wake mixing is observed in between the rotors, which moves the turbulent-breakdown region closer to the plane of rotation. The effects of tip-to-tip distance s on mean thrust and thrust fluctuation (standard deviation) are shown in Fig. 15. Here is observed that the mean thrust drops and the fluctuation increases as the distance between rotors becomes small, showing reasonable agreement with the experimental measurements reported by Zhou et al.²⁸

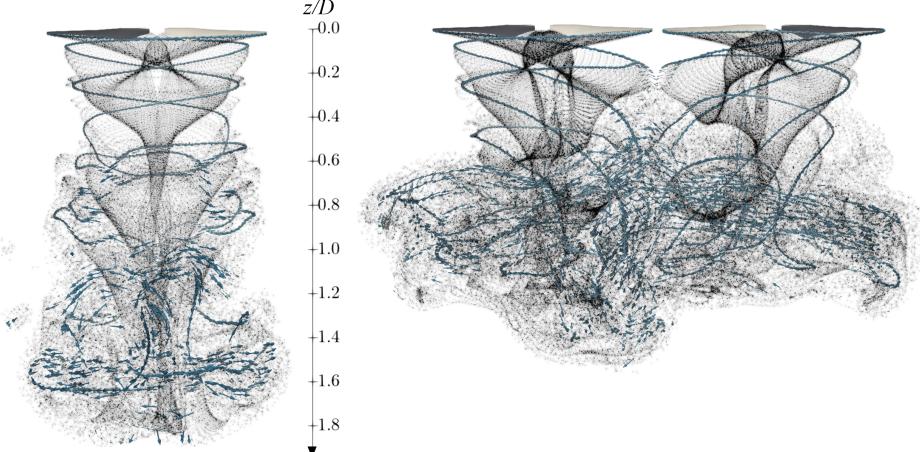


Figure 14: Wake of single-rotor and multirotor hover simulations. Arrows scaled by the vortex strength of every particle.

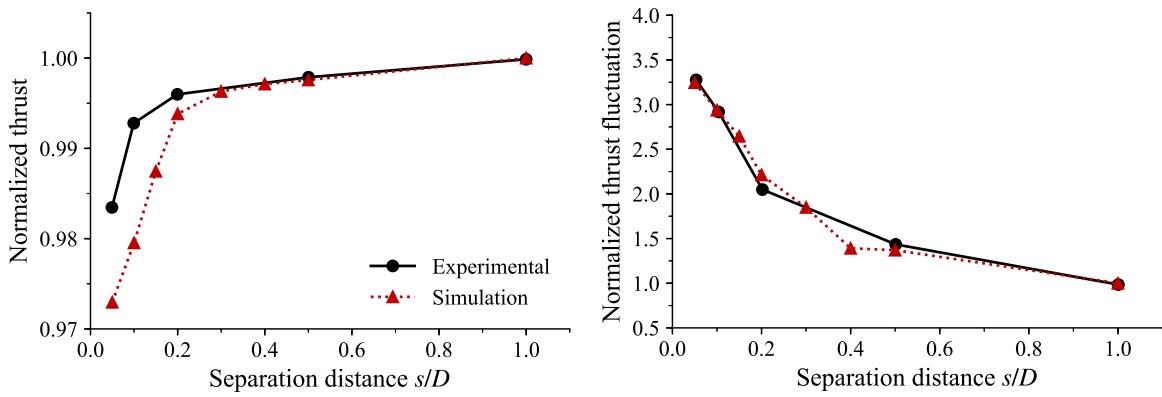


Figure 15: Effects of rotor-on-rotor interactions on thrust as separation between rotors is decreased, normalized by their respective values in single-rotor configuration.

IV. Expected Results

The current efforts of this research focus on validating the predicted rotor-on-wing and wing-on-rotor interactions. In the final manuscript, a complete validation of rotor-on-wing interactions will be shown by comparing the simulation of a blown wing with experimental measurements reported in the literature. It is also expected that the wing-on-wing interactions shown in Section III.C will be supplemented with a experimental study available in the literature.

Following the validation of the simulation engine, FLOWUnsteady will be used for analyzing the eVTOL transition maneuver of a tilt-wing multirotor aircraft. To illustrate the expected results, Fig. 16 shows a preliminary simulation capturing rotor-on-rotor interactions of an eVTOL aircraft using FLOWUnsteady.

Acknowledgments

This work is funded by the High Impact Doctoral Research fellowship granted by Brigham Young University. FLOWVPM uses a modified version of the open-source code ExaFMM originally developed by Lorena Barba and Rio Yokota. FLOWUnsteady integrates the open-source software Paraview as its visualization engine. FLOWUnsteady integrates the code PSU-WOPWOP for aeroacoustic noise predictions, developed by Kenneth Brentner.

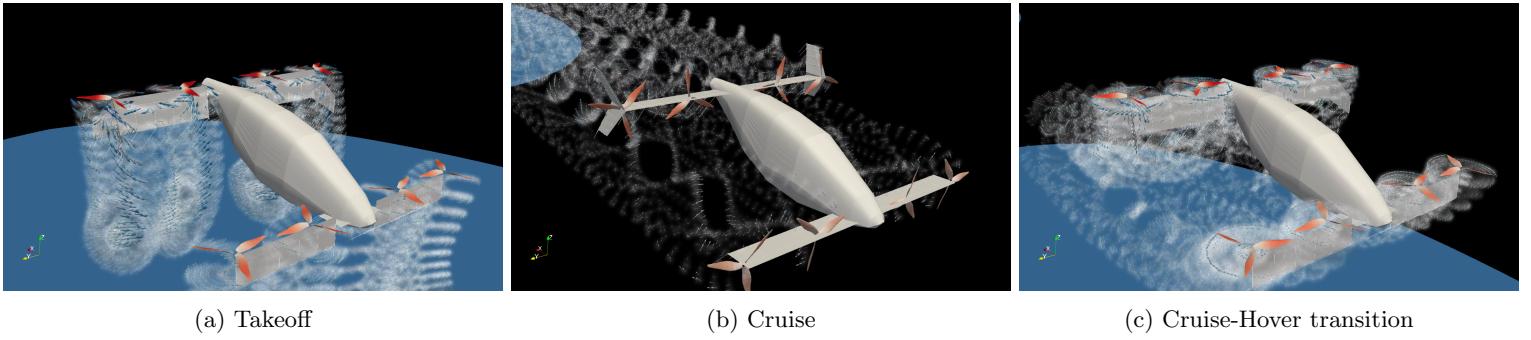


Figure 16: Preliminary results of eVTOL aircraft simulation completing mission profile.

References

- ¹ "Flying Cars: Investment Implications of Autonomous Urban Air Mobility," Tech. rep., Morgan Stanley Research, 2018.
- ² Moore, M., "NASA Puffin Electric Tailsitter VTOL Concept," *10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, American Institute of Aeronautics and Astronautics, Reston, Virginia, sep 2010, pp. 1–12.
- ³ Brelje, B. J. and Martins, J. R. R. A., "Electric, hybrid, and turboelectric fixed-wing aircraft: A review of concepts, models, and design approaches," *Progress in Aerospace Sciences*, , No. June, 2018, pp. 1–19.
- ⁴ Pascioni, K. and Rizzi, S. A., "Tonal Noise Prediction of a Distributed Propulsion Unmanned Aerial Vehicle," *2018 AIAA/CEAS Aeroacoustics Conference*, American Institute of Aeronautics and Astronautics, Reston, Virginia, jun 2018, pp. 1–18.
- ⁵ Persson, L. and Lawrence, B., "Design and Control of an Experimental Tiltwing Aircraft," Tech. rep., NASA, 2017.
- ⁶ Droandi, G., Syal, M., and Bower, G., "Tiltwing Multi-Rotor Aerodynamic Modeling in Hover, Transition and Cruise Flight Conditions," *AHS International 74th Annual Forum & Technology Display*, 2018, pp. 2018.
- ⁷ Droandi, G., Gibertini, G., Grassi, D., Campanardi, G., and Liprino, C., "Proprotorwing aerodynamic interaction in the first stages of conversion from helicopter to aeroplane mode," *Aerospace Science and Technology*, Vol. 58, nov 2016, pp. 116–133.
- ⁸ Yoon, S., Lee, H. C., and Pulliam, T. H., "Computational Analysis of Multi-Rotor Flows," *54th AIAA Aerospace Sciences Meeting*, , No. January, 2016, pp. 1–11.
- ⁹ Ventura, P., Yoon, S., and Theodore, C. R., "High-Fidelity Computational Aerodynamics of the Elytron 4S UAV," *American Helicopter Society International - Future Vertical Lift Aircraft Design Conference, San Francisco CA*, 2018.
- ¹⁰ Mankbadi, R. R., Afari, S., and Golubev, V. V., "Towards High-fidelity Analysis of Noise Radiation and Control of Propeller-driven UAV," *25th AIAA/CEAS Aeroacoustics Conference*, No. May, American Institute of Aeronautics and Astronautics, Reston, Virginia, may 2019, pp. 1–14.
- ¹¹ Alvarez, E. J. and Ning, A., "Development of a Vortex Particle Code for the Modeling of Wake Interaction in Distributed Propulsion," *2018 Applied Aerodynamics Conference*, American Institute of Aeronautics and Astronautics, jun 2018, pp. 1–22.
- ¹² Alvarez, E. J. and Ning, A., "Modeling Multirotor Aerodynamic Interactions Through the Vortex Particle Method," *AIAA Aviation 2019 Forum*, American Institute of Aeronautics and Astronautics, Dallas, Texas, jun 2019.
- ¹³ Alvarez, E. J. and Ning, A., "High-fidelity Modeling of Multirotor Aerodynamic Interactions for Aircraft Design," *AIAA Journal (accepted)*, 2020.
- ¹⁴ Winckelmans, G. and Leonard, A., "Contributions to Vortex Particle Methods for the Computation of Three-Dimensional Incompressible Unsteady Flows," *Journal of Computational Physics*, Vol. 109, No. 2, dec 1993, pp. 247–273.
- ¹⁵ Rossi, L. F., "Resurrecting Core Spreading Vortex Methods: A New Scheme that is Both Deterministic and Convergent," *SIAM Journal on Scientific Computing*, Vol. 17, No. 2, mar 1996, pp. 370–397.
- ¹⁶ Barba, L. A., Leonard, A., and Allen, C. B., "Advances in viscous vortex methods - Meshless spatial adaption based on radial basis function interpolation," *International Journal for Numerical Methods in Fluids*, Vol. 47, No. 5, 2005, pp. 387–421.
- ¹⁷ Barba, L. a., "Vortex Method for computing high-Reynolds number Flows: Increased accuracy with a fully mesh-less formulation," *California Institute of Technology*, Vol. 2004, 2004.
- ¹⁸ Torres, C. E. and Barba, L. A., "Fast radial basis function interpolation with Gaussians by localization and iteration," *Journal of Computational Physics*, Vol. 228, No. 14, 2009, pp. 4976–4999.
- ¹⁹ Winckelmans, G. S., "Topics in vortex methods for the computation of three and two dimensional incompressible unsteady flows," 1989.
- ²⁰ Pedrizzetti, G., "Insight into singular vortex flows," *Fluid Dynamics Research*, Vol. 10, No. 2, 1992, pp. 101–115.
- ²¹ Williamson, J. H., "Low-storage Runge-Kutta schemes," *Journal of Computational Physics*, Vol. 35, No. 1, 1980, pp. 48–56.
- ²² Greengard, L. F., *The Rapid Evaluation Of Potential Fields In Particle Systems*, Phd thesis, Yale University, 1987.
- ²³ Cheng, H., Greengard, L., and Rokhlin, V., "A Fast Adaptive Multipole Algorithm in Three Dimensions," *Journal of Computational Physics*, Vol. 155, No. 2, 1999, pp. 468–498.
- ²⁴ Yokota, R. and Barba, L. A., "Treecode and fast multipole method for N-body simulation with CUDA," *GPU Computing Gems Emerald Edition*, 2011, pp. 113–132.

- ²⁵Yokota, R. and Barba, L. A., “Comparing the treecode with FMM on GPUs for vortex particle simulations of a leapfrogging vortex ring,” *Computers and Fluids*, Vol. 45, No. 1, 2011, pp. 155–161.
- ²⁶Hu, Q., Gumerov, N. A., Yokota, R., Barba, L., and Duraiswami, R., “Scalable Fast Multipole Accelerated Vortex Methods,” *2014 IEEE International Parallel & Distributed Processing Symposium Workshops*, IEEE, may 2014, pp. 966–975.
- ²⁷Weber, J. and Brebner, G., “Low-Speed Tests on 45-deg Swept-Back Wings,” *Part I, Her Majesty’s Stationery Office (HMSO), Reports . . .*, Vol. 2882, No. 2882, 1958.
- ²⁸Zhou, W., Ning, Z., Li, H., and Hu, H., “An Experimental Investigation on Rotor-to-Rotor Interactions of Small UAV Propellers,” *35th AIAA Applied Aerodynamics Conference*, , No. June, 2017, pp. 1–16.