```
@inproceedings{scotzniovsky2026vortexad,
  author = {Luca Scotzniovsky and John T. Hwang},
  title = {VortexAD: A Graph-Based Linear Potential Flow Analysis Framework for
    Large-Scale MDO},
  booktitle = {AIAA AVIATION 2026 Forum},
  year = {2026},
  doi = {10.2514/6.2026-4642}
}
```

VortexAD: A Graph-Based Linear Potential Flow Analysis Framework for Large-Scale MDO

Luca Scotzniovsky*, and John T. Hwang†
University of California San Diego, La Jolla, California, 92023

Recent advancements in the automation of adjoint-based sensitivity analysis through graph-based modeling have enabled comprehensive, large-scale multidisciplinary design optimization (MDO) for aircraft design. Existing large-scale MDO demonstrations are currently limited to low-fidelity methods to represent crucial disciplines. Aerodynamics is one of the major disciplines for capturing key design tradeoffs of novel configurations at the conceptual design stage. However, aerodynamic methods in MDO are limited to lightweight low-fidelity methods or computationally heavy high-fidelity methods. In this paper, we address this gap by developing VortexAD, a graph-based linear potential flow analysis framework for large-scale MDO. VortexAD models arbitrary 3D surfaces using the panel method and thin 2D camber surfaces using the vortex lattice method (VLM). We employ a free-wake model with vortex dissipation to solve for unsteady potential flows. Adjoint-based sensitivity analysis is automated through CSDL, a graph-based modeling framework. We develop a novel ODE formulation to address the unsteady wake-shedding problem and apply partitioned vectorization to limit memory cost in transient flow cases. With this, we conduct verification and validation (V&V) studies of our framework with various cases in both steady and unsteady flow settings and see reasonable agreement with the data. In addition, we conduct numerical experiments using partitioned vectorization to assess the computational cost of the unsteady free-wake solver. We see noticeable memory reductions with manageable run times for unsteady flows. These studies demonstrate the efficacy of our framework to efficiently model key aerodynamic effects for aircraft conceptual design problems.

I. Introduction

Recently, sustainability initiatives in the aviation industry have emerged in response to growing concerns about the impact of aircraft emissions. Jet fuel currently accounts for close to 7% of world oil consumption, contributing to roughly 3% of overall carbon emissions and 4% of global temperature rise since the mid-20th century [1, 2]. At the current rate, the aviation industry will be responsible for roughly 25% of global carbon emissions by 2050 [3]. These initiatives have prompted the exploration of novel aircraft technologies, such as distributed propulsion, and the hybridization and electrification of aircraft propulsion systems. These technologies have been explored in a variety of novel aircraft concepts, such as the X-57 Maxwell demonstrator [4], the Parallel Electric-Gas Architecture with Synergistic Utilization Scheme (PEGASUS) concept [5], and the Electrified Powertrain Flight Demonstrator (EPFD) [6]. These technologies are relevant for subsonic low-speed aircraft operating on short-haul, regional missions. As a result, there is renewed interest in the design of propeller-driven aircraft. However, capturing the key design aspects of these aircraft configurations is difficult at the conceptual design stage, due to the high-dimensional design space and the dependence of the aerodynamic performance on the complex transient flow behavior.

Large-scale multidisciplinary design optimization (MDO), the application of numerical optimization to solve high-dimensional design problems with many disciplines, is well-suited for solving aircraft design problems. Recently, there has been significant progress towards comprehensive large-scale MDO, which considers the entire aircraft and a larger set of design conditions. Sarojini et al. demonstrated large-scale MDO of the NASA lift-plus-cruise, considering over 100 design variables [7]. Ruh et al. furthered this work by including a full mission profile, 400 design variables, and more than 200 constraints [8]. Although these studies modeled key disciplines using low-fidelity methods, they demonstrated the feasibility of large-scale MDO to capture key design trends.

A major drawback of existing aerodynamics models in MDO, however, is the bimodal distribution of fidelity. Many early-stage design studies utilize zero- to low-fidelity methods, ranging from quadratic drag polars and empirical models

*PhD Candidate, Department of Mechanical and Aerospace Engineering, lscotzni@ucsd.edu, AIAA Student Member
†Associate Professor, Department of Mechanical and Aerospace Engineering, jhwang@ucsd.edu, AIAA Member

to thin-airfoil approximations such as the lifting-line method (LLM) and vortex-lattice method (VLM) [7–10]. These approaches are not generalizable to novel configurations because they often fail to capture the relevant flow physics. LLM and VLM only model camber and cannot capture thickness effects. On the other hand, detailed design utilizes higher-fidelity methods like computational fluid dynamics (CFD) and vortex particle methods (VPM) to model the complete flow field. CFD is often used to model coupled flow phenomena, like aeroelastic [11, 12], aeroacoustic [13, 14], and aeropropulsive [15] coupling. VPM is commonly utilized for problems with unsteady, vortex-dominated flows with significant wake interactions [16, 17]. These methods can also be coupled with low-fidelity methods, like blade element momentum theory (BEMT) and immersed vorticity methods, to capture additional coupling between the flow field and surfaces [18, 19]. However, these methods require significant effort to set up and incur large run-time and memory costs. Although the computational cost bottlenecks can be addressed by parallelization through MPI, GPUs, or high-performance computing (HPC), this introduces separate difficulties with setup and accessibility. These methods also require a complex, detailed geometry representation, which often does not exist at the early conceptual design stage.

Panel methods are an attractive alternative to modeling the aerodynamic performance of full-scale configurations with reasonable computational cost. Panel methods are a class of numerical methods that compute potential flow fields around general configurations. Unlike high-fidelity approaches, these methods solve for surface quantities, which simultaneously reduces computational complexity and state size. Panel methods have been a workhorse aerodynamics analysis approach since the mid-late 20th century, with the first practical method to represent thick three-dimensional bodies being published by Hess and Smith in 1967 [20]. Since then, these methods have expanded to include higher-order distributions for improved accuracy and high-speed modeling [21, 22] and free-wake methods to capture unsteady interactions [23, 24]. More recently, panel methods have been coupled with high-fidelity viscous VPM to more accurately represent vortex structures in unsteady flows, like propeller-wing systems [25, 26] and helicopter rotors [27]. However, despite the extensive literature surrounding existing panel methods and their applications, there has been limited development in applying these methods to aircraft design optimization problems. Given that the existing tools also lack the adjoint-based sensitivity analysis needed for efficient gradient-based optimization, they are not compatible with large-scale MDO.

To address this gap, we present VortexAD, a graph-based potential flow analysis framework tailored for large-scale MDO of aircraft. Our framework is implemented using CSDL, a graph-based modeling language that automates adjoint-based sensitivity analysis [28]. This framework builds on a previously developed graph-based memory-efficient panel method [29]. This panel method is a central part of our analysis framework, which allows us to model arbitrary 3D configurations using source-doublet pairs. The framework also supports thin surface geometry representations using vortex rings. In addition, the framework supports both steady and unsteady flow fields. The unsteady flow field is resolved using a free-wake model with vortex dissipation and diffusion to model energy losses and viscosity. We developed a novel ODE approach to the free-wake shedding problem. With this framework, we present results for initial verification and validation studies and numerical experiments on the memory and run time of our unsteady free-wake solver.

The remainder of this paper proceeds as follows. Section II presents general background on potential flow theory and the unsteady free-wake model. Section III details aspects of the implementation and capabilities of our graph-based potential flow analysis framework. Section IV discusses verification and validation results for various steady and unsteady flow cases. Section V discusses results on numerical experiments conducted to assess the computational cost of our ODE-based free-wake solver. To conclude, Section VI summarizes the work presented in this paper and discusses future work.

II. Background

In this section, we outline the general details of linearized potential flow theory. Linearized potential theory models inviscid and irrotational flow fields, assuming the perturbation velocities of bodies in the flow are small relative to the free stream. For an irrotational flow field with velocity $\vec{U}$, we can define a velocity potential Φ such that $\nabla\Phi = \vec{U}$. In this work, we assume the flow field is also incompressible. With these assumptions, the continuity equation can be represented as

$$\nabla^2\Phi = 0, \tag{1}$$

which is satisfied everywhere in space. We aim to solve for flow fields around arbitrary 3D bodies that satisfy Equation 1. The potential field is a linear combination of two terms: the free-stream potential and the perturbation potential, which models the body's influence on the flow field. In the far-field, the perturbation potential approaches zero. Green's identity is applied to Equation 1 to formulate a general solution of the form

$$\Phi(x,y,z) = \frac{1}{4\pi}\int_{\text{body + wake}} \mu \mathbf{n}\cdot\nabla\left(\frac{1}{r}\right) dS - \frac{1}{4\pi}\int_{\text{body}} \sigma\left(\frac{1}{r}\right) dS + \Phi_\infty = \Phi^* + \Phi_\infty, \tag{2}$$

where we denote the perturbation potential Φ^* as the integral terms and σ and μ represent sources and doublets, respectively [30]. This formulation is known as a source-doublet panel method [20]. Sources represent the velocity jump across boundaries and primarily enforce thickness effects. Meanwhile, doublets represent the potential jump over boundaries and are responsible for creating circulation to generate lift. To determine the source and doublet strengths, we discretize our surface with a mesh and solve a boundary value problem. The boundary conditions are applied to the center points of a surface mesh, formally known as collocation points. Two boundary conditions can be applied: Dirichlet or Neumann. The Dirichlet problem enforces a constant potential Φ_S on and within the surface: $\Phi = \Phi_S$. The Dirichlet problem is commonly solved with a no-perturbation potential boundary condition: $\Phi^*|_S = 0$. The Neumann problem explicitly enforces a no-penetration condition: $\nabla\Phi\cdot\mathbf{n} = 0$, where $\mathbf{n}$ is the surface normal. In this work, the sources and doublets are assumed to be constant on each panel. This reduces Equation 2 to a linear system of the form

$$A\mu + B\sigma + C\mu_w = V_{BC}, \tag{3}$$

where μ_w represents the wake doublets shedding from the surface. The terms μ, σ and μ_w in our linear system refer to the distribution of these singularities across the surface and wake panels. V_{BC} represents the boundary condition value:

$$V_{BC} = \begin{cases} 0 & \text{Dirichlet} \\ -\vec{U}_c\cdot\mathbf{n} & \text{Neumann} \end{cases}, \tag{4}$$

where U_c in the Neumann condition represents the collocation velocity. The collocation velocity is defined as

$$\vec{U}_c = \vec{U}_\infty + \vec{U}_r + \vec{\omega}\times\vec{r}, \tag{5}$$

and accounts for the effects of the free-stream velocity $\vec{U}_\infty$ and unsteady effects from translation $\vec{U}_r$ and rotation $\vec{\omega}\times\vec{r}$. In steady flows, only the free-stream term is necessary. The matrices A, B, C correspond to the interactions across panel computed from the integrals in Equation 2. These matrices compute either the induced potential or velocity of each panel, depending on the chosen boundary condition. To uniquely define the flow field, the source strengths are set to

$$\sigma = -(\vec{U}_c + \vec{U}_m)\cdot\vec{n}, \tag{6}$$

to capture all relative velocities. The term $\vec{U}_m$ represents additional miscellaneous velocity influences. This can include the boundary layer transpiration velocity [31], or the wake-induced velocities from vortex wake particles [26]. The wake strengths μ_w are set to satisfy the Kutta condition, which enforces that the circulation at the trailing edge, where the upper surface, lower surface, and wake meet, must be zero. To enforce this, the wake doublet strengths closest to the surface are computed from the surface doublet strengths as

$$\mu_w = \mu_u - \mu_l, \tag{7}$$

where the subscripts u, l represent the upper and lower surfaces of the trailing edge, respectively. With these conditions, the only unknown in Equation 3 is the doublet strength vector, containing the doublet strengths on each panel of the body. The linear system in Equation 3 is solved to determine the doublet strengths on the surface.

Linearized incompressible potential flow theory can also predict flow behavior for thin, 2D surfaces. This is often referred to as the vortex lattice method. Thin surfaces are treated as flat, camber representations of closed, 3D surfaces. We apply a pure Neumann boundary condition to solve for a distribution of bound vortex rings on the surface:

$$(\vec{U}_c + \vec{u}_i)\cdot\mathbf{n} = 0, \tag{8}$$

where $\vec{u}_i$ represents the vortex ring induced velocities. Geometrically, the bound vortex rings are located at a quarter-chord shift downstream of each surface panel. The collocation point, where the boundary condition is applied, is located at the three-quarter chord of the mesh panel or the center of the bound vortex grid. We impose a no-penetration condition at the collocation points, considering the free stream velocity and the induced velocities of the bound and wake vortex rings. We can model Equation 8 as a linear system of the form

$$A\Gamma + B\Gamma_w = -V_n, \tag{9}$$

where Γ_w represents the wake vortex rings shed from the surface. The boundary condition term V_n represents the thin surface boundary condition:

$$V_n = \vec{U}_c \cdot \mathbf{n}, \tag{10}$$

which represents the same velocity components in the panel method Neumann boundary condition in Equation 4. The matrices A and B correspond to the induced velocities of the bound and wake vortex rings projected in the surface normal direction at the collocation points. To compute the vortex-ring induced velocities at a point in space P, we decompose the element into vortex line elements and use the Biot-Savart law:

$$\vec{q}_{ab} = \frac{\Gamma}{4\pi} \frac{\boldsymbol{r_a} \times \boldsymbol{r_b}}{\|\boldsymbol{r_a} \times \boldsymbol{r_b}\|^2} \boldsymbol{r_o} \cdot \left(\frac{\boldsymbol{r_a}}{\|\boldsymbol{r_b}\|} - \frac{\boldsymbol{r_b}}{\|\boldsymbol{r_b}\|} \right), \tag{11}$$

where $\boldsymbol{r_a}$ and $\boldsymbol{r_b}$ represent the vectors from the start and end points of the vortex line segment to point P, $\boldsymbol{r_0}$ represents the line vortex, and Γ is the vortex ring strength. When assembling the matrices in Equation 9, the vortex rings are assumed to have unit strength. Similar to the thick surface approach, we use the Kutta condition to determine wake vortex strength:

$$\Gamma_w = \Gamma_{TE}, \tag{12}$$

where Γ_{TE} is the strength of the trailing-edge vortex ring. The linear system in Equation 9 can then be solved for the bound vortex strengths Γ across the surface.

A. Post-Processing

The solutions of doublet, source, and vortex ring strengths are used to compute forces. Depending on the surface type, these forces are computed differently. For thick, 3D surfaces, the doublets and sources are used to compute the total velocity on the body.

$$\vec{Q} = \vec{U}_c + \nabla\mu + \sigma\mathbf{n}, \tag{13}$$

where $\mathbf{n}$ represents the outward unit panel normal vector and the doublet gradient is tangent to the surface. The pressure coefficient at each panel can then be computed as

$$C_p = 1 - \frac{Q^2}{U_\infty^2} + C_{p_{uns}} \tag{14}$$

where Q, U_∞ represent their respective vector norms, and $C_{p_{uns}}$ is the unsteady pressure term. The unsteady pressure term captures temporal variations of the surface potential and is zero in steady flows. The unsteady pressure coefficient is defined as

$$C_{p_{uns}} = -\frac{2}{U_\infty^2} \frac{d\Phi}{dt} \tag{15}$$

where $\Phi = \mu$ because μ represents the potential on the surface with a zero-perturbation potential boundary condition. To model compressibility effects, the pressure coefficient is scaled using the Prandtl-Glauert correction:

$$C_{p_c} = \frac{C_p}{\sqrt{1 - M_\infty^2}}, \tag{16}$$

where M_∞ is the freestream Mach number. Using the pressure coefficient, the force on each panel can be computed as

$$F_k = -\frac{1}{2}(C_{p_c})_k \rho U_\infty^2 S_k \mathbf{n}_k \tag{17}$$

where k represents the panel index and S represents the panel area. To compute lift and drag and their respective coefficients, this force can be decomposed into the two terms parallel and orthogonal to the flow direction.

For thin surfaces, we use the Kutta-Joukowski theorem to compute forces:

$$F_k = \rho\Gamma_k(\vec{l}_k \times \vec{V}_k) + F_{uns} \tag{18}$$

where Γ represents the net circulation strength of each bound vortex ring, $\vec{l}$ represents the dimensional bound vortex vector, F_{uns} represents the unsteady forcing due to temporal variations of the surface potential, and $\vec{V} = \vec{U}_c + \vec{v}_i$

represents the total velocity at the bound vortex. The quantity $\vec{v}_i$ represents the total induced velocity at the bound vector:

$$\vec{v}_i = A\Gamma_b + B\Gamma_w, \tag{19}$$

where A, B represent the velocity of each bound vortex and wake vortex induced at the bound vector computed using the Biot-Savart law. The unsteady forcing term is defined as

$$F_{uns} = \rho S_k \mathbf{n}_k \frac{d\Gamma_{bk}}{dt}, \tag{20}$$

where the time derivative term represents the unsteady component of the force variation. Similar to the thick surface modeling, we can utilize the Prandtl-Glauert correction:

$$F_{kc} = \frac{F_k}{\sqrt{1 - M_\infty^2}}, \tag{21}$$

where the scaling is now applied directly to the force. The panel forces F_{kc} can be integrated and decomposed into components parallel and orthogonal to the flow field to compute lift and drag.

B. Wake-shedding

For unsteady flows, we propagate the wake in time using a free-wake method. Free-wake methods incorporate surface-and-wake-induced velocities into the wake-shedding process to capture realistic flow behavior, such as wing-tip vortices. The induced velocities are computed using the source, doublet, and wake doublet strengths. The total velocity of a wake element can be written as

$$\vec{V}_w = \vec{V}_\infty + \vec{v}_{wi}, \tag{22}$$

where the induced velocity of the wake elements is

$$\vec{v}_{wi} = A\mu + B\sigma + C\mu_w, \tag{23}$$

and the tensors A, B, C correspond to the unit-strength induced velocity of the surface doublets, surface sources, and wake doublets, respectively. We recycle the matrix/tensor notation from Equation 3 to represent induced velocities. The induced velocities for each source term in the global reference frame can be found in [31]. However, the induced velocities of doublets and vortex rings are identical. We compute the induced velocities of doublets and vortex rings using the Biot-Savart law in Equation 11, decomposing each element into vortex line elements. This computation is repeated for each edge of the panel. The total wake velocity in Equation 22 is then used to update the wake locations for the next time step.

The wake-shedding incorporates vortex diffusion and dissipation to represent viscosity and energy losses in the flow. These phenomena are not naturally captured by potential flow, so we model these effects using empirical methods previously incorporated in potential flow models [32, 33]. Vortex diffusion captures the effects of viscosity in the flow using a vortex-core model. We introduce a time dependence to the finite-core radius r_c:

$$r_c = \sqrt{r_{c0}^2 + 4\alpha\delta\nu t}, \tag{24}$$

where r_{c0} is the initial finite core size when the wake first sheds and t is the age of the wake element. The $\delta\nu$ is the average effective turbulent viscosit defined as

$$\delta\nu = \nu + a_1\Gamma, \tag{25}$$

where ν is the kinematic viscosity and Γ is the wake element circulation strength. Note that Γ in Equation 25 is a time-dependent term. In Equations 24 and 25, the empirical parameters α and a_1 represent the Oseen coefficient and Squire's coefficient, respectively. Vortex dissipation models the decay in wake circulation strength across time. In this work, we model the vortex dissipation as

$$\Gamma(t) = \Gamma_0 \exp\left(-\frac{bq}{s}t\right), \tag{26}$$

where Γ_0 is the initial wake element circulation strength when it is first shed from a surface. The term $\frac{bq}{s}$ is another empirical parameter representing the wake decay as a function of the ambient turbulence level. In this work, we set $\alpha = 1.25643$ and $\frac{bq}{s} = 2.5$, based on experimental observations from De Gregorio et al. [32]. We also ignore the effect of circulation strength on the vortex core size by setting $a_1 = 0$.

III. Graph-based Linear Potential Flow Analysis Framework

In this section, we describe VortexAD*, our novel graph-based potential flow analysis framework. First, we highlight implementation details and solver capabilities. Then, we present our method for a novel wake-shedding formulation.

A. Graph-Based Implementation

Our framework uses a low-order formulation of constant-strength vortex elements for doublets, sources, and vortex rings. The source-doublet potential and velocity influence coefficients for Equation 3 are computed using the formulation from the VSAERO panel code [31]. This formulation goes beyond the original Hess and Smith method [20] by representing induced potentials and velocities using a global reference frame, rather than the local coordinate frames centered at each panel. For doublet panels and vortex rings, as previously mentioned, we compute induced velocities of each edge using the Biot-Savart law in Equation 11 and sum the contribution of each edge for a given panel. Our solver is developed using CSDL, a graph-based modeling framework that automates adjoint-based sensitivity analysis [28]. Graph-based modeling provides speed-ups through vectorization and graph transformations to optimize sequences of operations. An image of the unsteady panel method graph is shown in Figure 1.

Our framework supports 3D surface meshes as triangulations, quadrangulations, or a mix of elements to handle arbitrary configurations. Quad-dominant meshes are generally preferred because the two cardinal in-plane vectors of quadrilateral elements align well with the directions of surface property variations. However, complex geometries with intersections or flat edges, such as a fuselage-wing mesh, require triangular elements to refine the boundary and prevent collapsed elements. We include thin surface modeling as flat meshes to capture camber. Figure 2 shows a rectangular wing using the three different grid types. Information about the trailing edge is required to model lifting behavior and satisfy the Kutta condition in Equation 7. For unstructured surface meshes, our framework has a trailing-edge detection method to automate this process, as described in previous work [29]. The method identifies elements, nodes, and edges at the trailing edge by searching for two criteria regarding the normal vectors of two adjacent panels. The first criterion is that the outward-normal vectors of adjacent panels must both point downstream. The second criterion assesses whether the two panels meet at a sharp edge by comparing the difference in normal vectors to a threshold angle θ_{TE}. Mathematically, this is represented as

$$\vec{n}_1 \cdot \vec{n}_2 < \cos(\theta_{TE}), \tag{27}$$

where an edge and corresponding panels are marked as trailing edge elements if this condition is satisfied. For structured, flat grids, the trailing edge elements is easily determined by the array structure.

1. Partitioned Vectorization

In this section, we review partitioned vectorization, a memory reduction technique developed in previous work for a steady panel code [29]. As mentioned in Section II, the potential flow solution requires solving a linear system like Equation 3 to compute doublet strengths, vortex ring strengths, or both. In this work, we focus on a direct linear system solve, which requires explicitly computing the full linear system matrix. Vectorization is generally the fastest approach to computing large arrays because it avoids iterative procedures. With vectorization, similar computations occur simultaneously rather than sequentially, as viewed from the level of the computational graph. However, vectorization poses a major memory bottleneck due to the n^2 interactions present in potential flow methods. As a result, vectorization is not viable for large problems.

Partitioned vectorization is a method that divides vectorized computations into partitions to trade off between memory and run time and alleviate the n^2 memory cost. Figure 3 illustrates this approach. We define four types of partitioning schemes: row, column, row-column, and column-row, shown from left to right in Figure 3; a and b represent the row and column size, respectively. The top left corner indices of each box represent the partition order determined by the partition type.

The benefit of partitioned vectorization is due to the reduced memory storage of intermediate computations when computing elements of large arrays. Let us first quantify the cost of the full vectorization approach. Consider a single element of our linear system matrix, which requires upwards of 100 sequential operations; we identify this cost as C, which is also the minimum allowable partition cost. In the full vectorization approach, for an $n \times n$ matrix, we store Cn^2 values, as each element is computed simultaneously. With partitioned vectorization, the vectorized operation is divided into partitions of size $a \times b$, where a and b represent row and column partition sizes, respectively. As a result, the element count stored in memory for each partition is Cab. Including the full n^2 interactions in the final explicit matrix

*`https://github.com/LSDOlab/VortexAD`

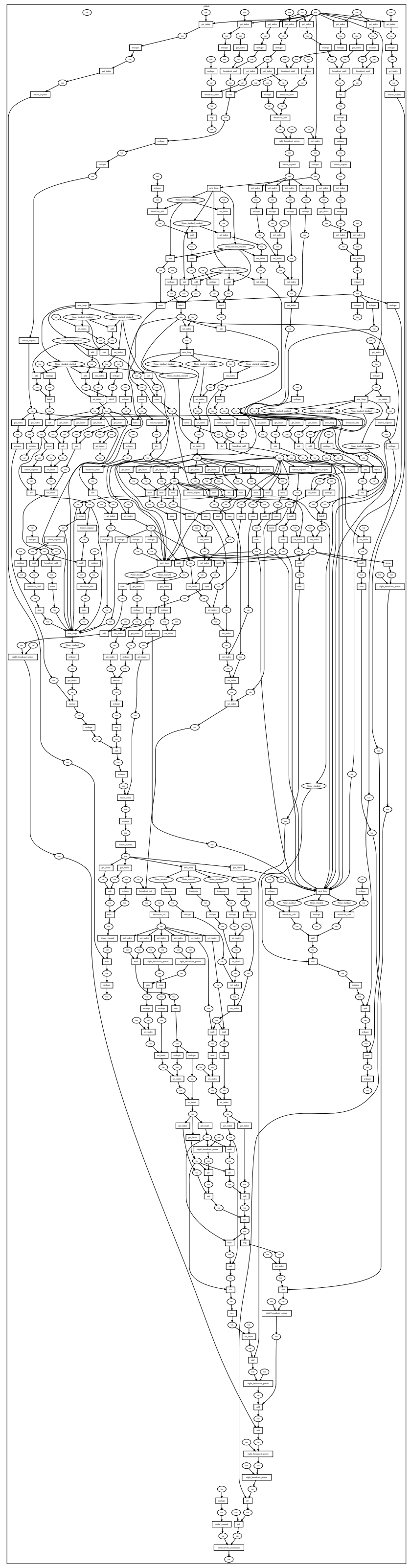
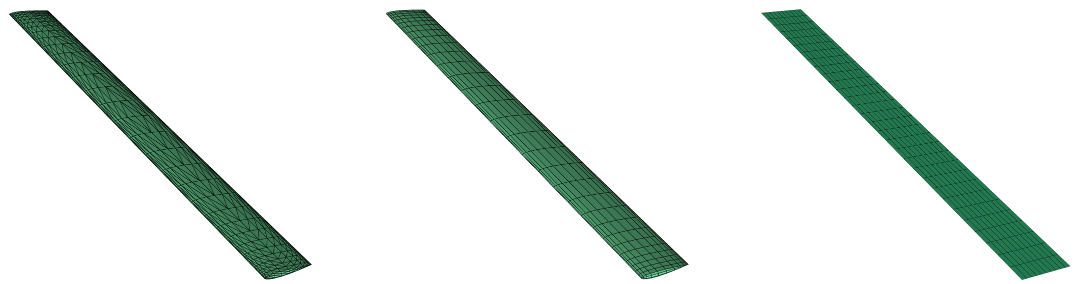


Fig. 1 Computational graph of the unsteady linear potential flow solver ODE with a free-wake method.



(a) 3D triangular surface mesh **(b) 3D quad surface mesh** **(c) Structured camber mesh**

Fig. 2 The three different grid types supported in this framework are shown here. The triangulation and quadrangulation are shown for 3D surface meshes with an NACA0012 airfoil.

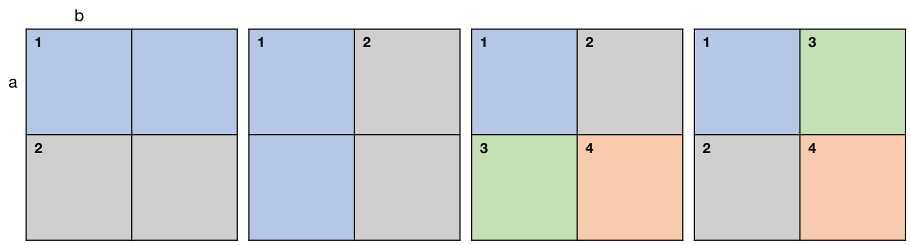


Fig. 3 A comparison of different partitioning methods for explicit matrix assembly. From left to right, the partitioning schemes shown are: row, column, row-column, and column-row. This figure was originally published for a graph-based panel code developed by Scotzniovsky et al. [29].

generation, the total data cost is $n^2 + Cab$. We see that generalized partitioning has the potential to significantly reduce memory cost. For small values of a and b, the scaling of memory cost to compute matrix entries becomes negligible. However, for $n \geq 1e4$, partitioning alone is still too prohibitive for large problems as the dominating n^2 cost remains. For problems of this scale, we must use iterative methods to solve the linear system.

Partitioned vectorization can also be applied to large matrix-vector and tensor-vector products. Many operations in our framework consist of matrix-vector or tensor-vector products, such as the free-wake induced velocity computation shown in Equation 23 and the right-hand side terms of the linear system in Equations 3 and 9. These products can be computed during assembly by partitioning across rows, effectively negating the need to generate the full tensor or matrix. This saves significant memory during run time, as these matrices and tensors exhibit n^2 memory cost.

2. Doublet Perturbation Velocity Numerical Differentiation

We use numerical differentiation to compute the doublet contribution of the perturbation velocity for thick surfaces in Equation 13. This method has also been described in our previous work for the graph-based panel method [29]. We approximate the doublet gradient with a Taylor series expansion at point j relative to point i:

$$\mu_j = \mu_i + \frac{\partial \mu}{\partial l}\Delta l + \frac{\partial \mu}{\partial m}\Delta m, \tag{28}$$

where l and m represent the two orthogonal in-plane vectors of panel i. The partial derivative terms can be combined to simplify the expression to

$$(\nabla \mu_i) \cdot r_{ij} = \mu_j - \mu_i, \tag{29}$$

where r_{ij} is a vector of displacements in the l and m directions. For a panel i, this condition must be enforced with each neighboring panel. We reformulate this as a least-squares problem of the form

$$\begin{bmatrix} \Delta l_{i1} & \Delta m_{i1} \\ \Delta l_{i2} & \Delta m_{i2} \\ \dots & \dots \\ \Delta l_{iN} & \Delta m_{iN} \end{bmatrix} \begin{bmatrix} \frac{\partial \mu}{\partial l} \\ \frac{\partial \mu}{\partial m} \end{bmatrix}_i = \begin{bmatrix} \mu_1 - \mu_i \\ \mu_2 - \mu_i \\ \dots \\ \mu_N - \mu_i \end{bmatrix}, \tag{30}$$

where N is the number of neighbors of panel i and the surface doublet perturbation velocity is $\nabla\mu = [\frac{\partial\mu}{\partial l}, \frac{\partial\mu}{\partial m}]^T$. This method is generalizable for polygons with any number of neighbors; in this work, we only consider triangular and quadrilateral elements. This system of equations is solved for each panel on the surface. For grids with many panels, the process of iteratively assembling the system of equations and solving each is time-consuming. Using vectorization, we apply the normal equation by left-multiplying Equation 30 with the transpose to analytically solve the resultant 2x2 linear system.

B. ODE Formulation of the Wake-Shedding Problem

In this subsection, we describe our ODE formulation for the wake-shedding problem. The two ODE states are the wake nodes x_w and the wake doublet strengths μ_w. Given the equivalence of the doublet and vortex ring induced velocities, the ODE formulation also applies to the wake vortex rings Γ_w. For the sake of simplicity, we denote the wake vortex strengths using μ_w.

Graph-based modeling is well-suited for efficient ODE time integration. For each iteration, the sequence of operations to compute state derivatives is the same. Thus, the computational graph can be reused. A consequence of reusing the computational graph is that the states and dynamic parameters must be preallocated across all time steps. As a result, we store inactive data for the wake states that have yet to shed into the flow. To access the active wake data at each time step, we define activation arrays that function as a continuous switch to the time derivatives. These activation arrays use zero-based indexing, like Python.

We solve the ODE using a Lagrangian perspective, where states are tracked along the wake particles and panels. The equivalent Eulerian perspective would be to shift the ODE states along an array as the wake sheds downstream. However, in developing this framework, our numerical experiments showed that the accumulation of finite difference error led to significant wake instability.

Before outlining our formulation, we present basic nomenclature. The total number of time steps is denoted nt, while the time step index is denoted n. For each timestep, the wake node derivatives are defined as:

$$\frac{dx_w}{dt}[n,:] = \dot{x}_w[n,:] = V_w[n,:]A_x[n,:], \tag{31}$$

where $V_w = V_\infty + V_i$ represents the total wake velocity vector as a sum of the freestream and induced free-wake velocities. The induced free-wake velocities are computed according to Equation 23. We define a velocity activation array:

$$A_x[n,i,:] = \begin{cases} 1 & i \in [nt-n-1, nt-1] \\ 0 & \text{else} \end{cases}, \tag{32}$$

to identify which row of wake panels has been shed, based on the time step index n. We apply a similar logic to the wake doublet strengths using activation arrays. At each timestep, the wake doublet derivatives are defined as

$$\frac{d\mu_w}{dt}[n,:] = \dot{\mu}_w[n,:] = \frac{\mu_{TE}[n,:]}{\Delta t}A_\mu[n,:] + \beta[n,:], \tag{33}$$

where changes to the wake doublet field depend on the Kutta condition and vortex dissipation. The Kutta condition derivative is responsible for generating lifting flow, while the vortex dissipation models energy losses in the wake. First, we discuss the Kutta condition derivative. We formulate a finite-difference approximation to enforce the Kutta condition, where μ_{TE} is the expected wake doublet strength defined by the Kutta condition in Equation 7. This term applies only to the newest row of wake panels. Thus, we define a sparse activation array for the Kutta condition derivative as

$$A_\Gamma[n,i] = \begin{cases} 1 & i = nt-n-2 \\ 0 & \text{else} \end{cases}, \tag{34}$$

where the only non-zero index corresponds to the index of the newest row of wake panels. In the absence of dissipation, the wake doublet strengths remain constant once they are shed. The dissipation derivatives β are applied to the remaining shed wake panels and are modeled as

$$\beta[n,:] = \begin{cases} \dot{\mu}_{diss}[n,:]B_\mu[n,:] & \text{yes dissipation} \\ 0 & \text{no dissipation} \end{cases}, \tag{35}$$

where

$$\dot{\mu}_{diss}[n,:] = -\frac{bq}{s}\mu_w[n,:] \tag{36}$$

is the time derivative of Equation 26. We define a final activation array for the dissipation derivative as

$$B_\mu[n,i] = \begin{cases} 1 & i \in [nt-n-2, nt-2] \\ 0 & \text{else} \end{cases}, \tag{37}$$

where the indices containing 1 correspond to wake rows that have already been shed. As a result, the dissipation derivative is not applied to wake elements that are pre-allocated but have yet to shed from the surface.

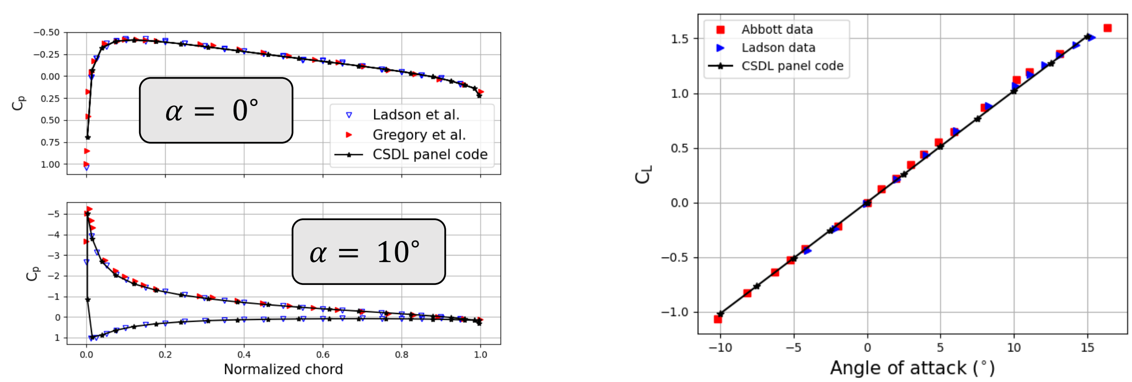
We use Ozone, †, an open-source Python library for solving ordinary differential equations, to solve our ODE [34]. Ozone is developed using CSDL, so adjoint-based sensitivities are automatically propagated through the ODE solver. Ozone offers various integration techniques, such as time-marching and collocation methods. In addition, Ozone offers checkpointing for efficient memory management. We use a time-marching procedure for the wake-shedding problem. Besides the Kutta condition derivative for the wake doublet strengths, the ODE formulation uses analytical derivatives to update the ODE states.

IV. Verification and Validation

In this section, we present verification and validation (V&V) results of our general potential flow analysis framework against canonical solutions and experimental data.

A. NACA0012 Rectangular Wing Validation

We compare the steady formulation of our panel code against experimental data using a rectangular wing with a NACA0012 airfoil [35, 36]. This validation case was published previously for a steady graph-based panel code, developed as part of this framework [29]. The NACA0012 wing was modeled with a 10-meter wingspan and an aspect ratio of 10. As shown in Figure 4, we see good agreement between experimental data and the VortexAD pressure coefficient distributions at $\alpha = 0°$ and $\alpha = 10°$ for the NACA0012 wing. In addition, the lift coefficients also show good agreement, with a maximum error of around 5%.



(a) Pressure coefficient comparison **(b) Lift vs. angle of attack**

Fig. 4 Comparison of NACA0012 pressure coefficient distributions and lift coefficient with experimental data. We see good agreement of pressure distributions and lift coefficients, with a maximum lift coefficient error of around 5%. This validation study was published in previous work, focused on developing the panel code for the VortexAD framework [29].

B. Sudden Acceleration of a Wing From Rest

In this study, we assess the aerodynamic properties of a wing upon a sudden acceleration from rest. This is a canonical verification case for free-wake methods to test how unsteady flows converge to steady-state operation. We

† https://github.com/LSDOlab/ozone

model a flat plate across many different aspect ratios, up to 1000. In addition, we also consider a wing with an NACA0012 airfoil. We solve for the aerodynamic coefficients at $\alpha = 5°$. The time evolution of C_L and C_{Di} is shown in Figure 5. We see that, as the aspect ratio increases, our flat plate lift coefficient approaches the theoretical 2D flat plate

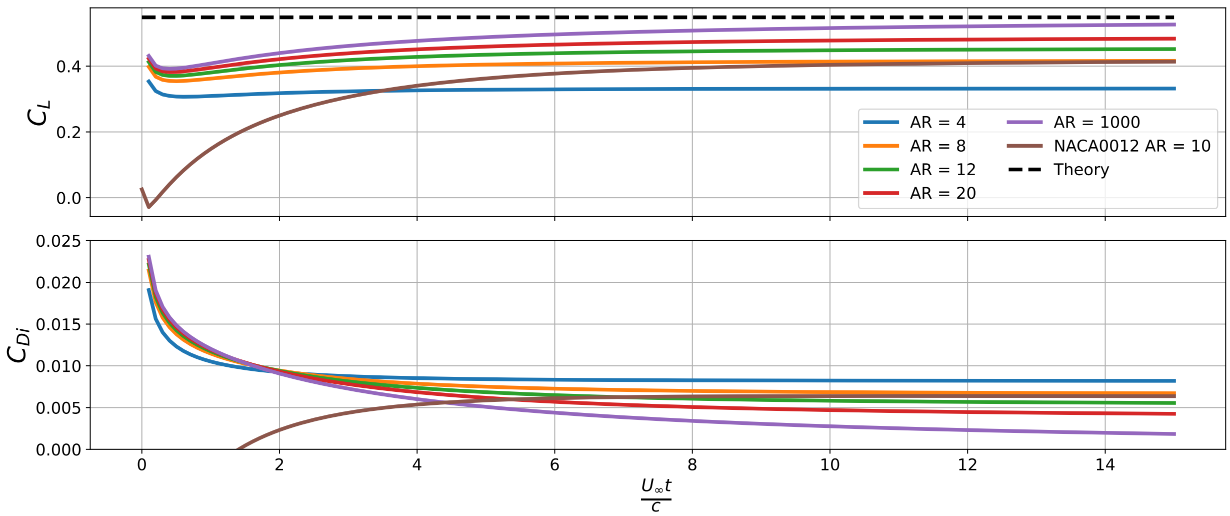


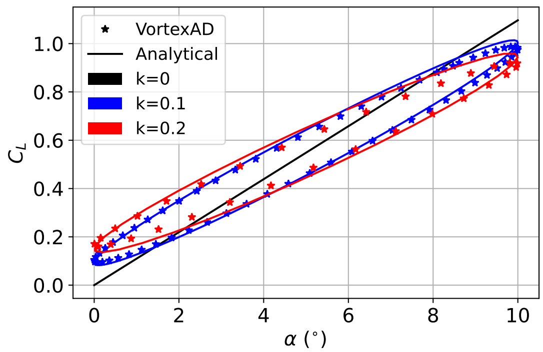
Fig. 5 Trends of C_L and C_{Di} across time. We compare a flat plate rectangular wing at various aspect ratios, along with a rectangular wing with a NACA0012 airfoil at an aspect ratio of 10.

lift coefficient of 0.548 at $\alpha = 5°$. Meanwhile, the induced drag coefficient monotonically decreases with increasing aspect ratio. This behavior is expected, as the downwash effect of wing-tip vortices is reduced as the aspect ratio increases. The downwash effect is the driver behind lift-induced drag and lowers the effective angle of attack, which explains why the lift coefficient approaches but never reaches the theoretical 2D lift coefficient. For wings that approach infinite span, we expect the induced drag coefficient to approach zero. We see this trend in our numerical results as the aspect ratio is increased.

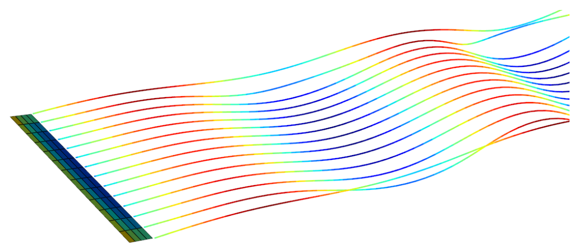
At the initial time steps, the induced drag of the NACA0012 airfoil wing is negative, which is not physically realistic. This is a consequence of computing drag using the pressure distribution before the wake has fully developed. In the future, we will explore a Trefftz-plane approach to compute induced drag to avoid unphysical negative drag.

C. Theodorsen Heaving Wing Solution

In this study, we assess the time-varying lift coefficient of a heaving wing and compare it to the canonical Theodorsen solution [37]. The purpose of this study is to assess the solver's capability to capture the effects of added mass and phase lag of the wake. We compared the VortexAD solution against the analytical solution, given a sinusoidal angle of attack input of the form $\alpha(t) = \alpha_0 + \alpha_a \sin(\omega t)$. The terms α_0 and α_a represent the mean and amplitude of the time-varying angle of attack, respectively. The wake of the lifting surface is propagated far enough such that the oscillatory angle of attack achieves four cycles. We assessed reduced frequencies of $k = 0.1$ and $k = 0.2$, where $k = \frac{\omega c}{2V_\infty}$ and c is the chord. The results for the heaving wing simulation are shown in Figure 6. We see that VortexAD captures the general trends of the oscillatory lifting response. There is reasonable agreement between the analytical data and VortexAD for the reduced frequency of $k = 0.1$. As the frequency doubles to $k = 0.2$, the VortexAD solution does not accurately capture the phase lag effects. However, the resultant error is likely a consequence of too large a time step. From Figure 6a, we see that, when the frequency is doubled, about half of the points are sampled in one cycle. The total simulation time is unchanged, but the angle of attack oscillates twice as fast. Without enough refinement of the wake, the influence of the wake on the wing will be underestimated. In the future, this study will be repeated with a timestep that depends on the frequency. In addition, the grid can be refined to improve the agreement between the analytical solution and the VortexAD simulation results.



(a) The heaving wing lift coefficient.



(b) The heaving wing force-free wake.

Fig. 6 We see reasonable agreement between the VortexAD numerical results and the analytical solution of the heaving wing transient lift coefficient. We see better agreement for the lower-frequency case, likely due to the timestep. The $k = 0$ solution describes the steady C_L variation across angle of attack.

D. Verification of Wake Vortex Dissipation Model

In this study, we verified the wake vortex dissipation in our ODE formulation in Equation 36 with the closed-form solution of Equation 26. We conducted this study to assess the validity of the ODE approach to model dissipation effects with the activation arrays. The analytical formulation of the wake vortex dissipation prescribes a degree of dissipation at each time instance. With the ODE formulation, however, we provide the expected *change* in time to the dissipation at each time instance. As a result, we expect a certain degree of numerical error to accumulate with the time integration process. To assess this error, we simulated a wing in forward flight and compared the relative wake strength to the closed-form solution in Equation 26. We define the relative wake strength as $\bar{\mu} = \mu_w(t)/\mu_{w0}$, which is an indicator of how much the wake vortex element strength reduces in time. The wake is propagated downstream to just under 5 chord lengths. We sampled three different time steps to assess the error in the relative wake strength. Figure 7 shows the relative wake vortex strength and error of the dissipation term in the ODE formulation across time. Qualitatively, applying the dissipation derivatives to the wake element strengths in the ODE formulation matches well

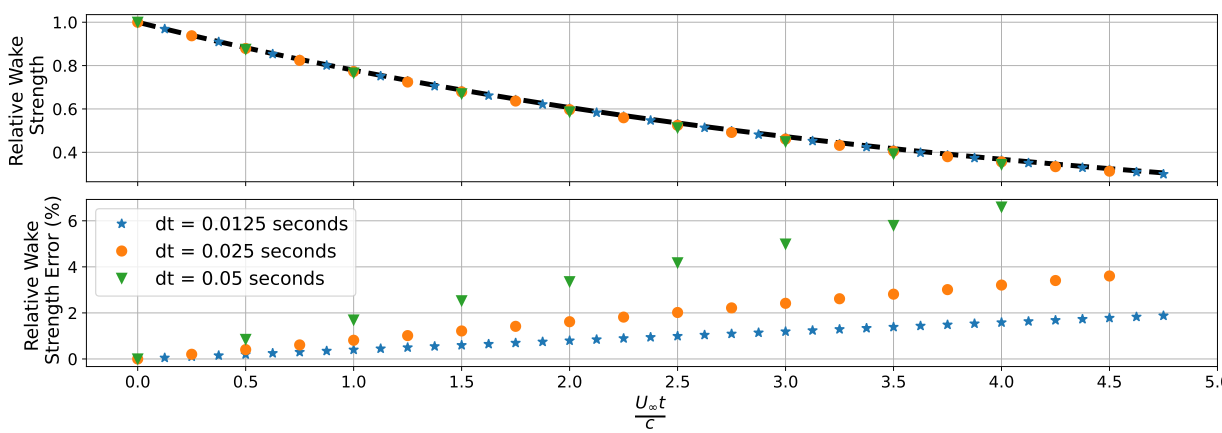


Fig. 7 Wake dissipation across various time steps.

with the analytical closed-form solution. We see a direct correlation between the accumulated error in the relative wake strength and the time step. As we cut the time step in half, the error is also halved. This is expected, given that forward Euler, a first-order method, is used to integrate across time. We anticipate that the error can be further controlled using higher-order time integration schemes. Although there are noticeable errors in the relative wake strength across time, it is important to note the overall impact of the far-field wake elements on the solution. Induced potentials and velocities scale with $1/r$ and $1/r^2$, respectively. As a result, the solution will be more heavily influenced by the near-field elements.

E. Propeller Experimental Validation

We validate our unsteady VLM model using experimental data from a study conducted at the Low Speed Acoustics Wind Tunnel (LSAWT) at the NASA Langley Research Center [38]. We compare our VortexAD results against the experimental data of the C24ND proprotor. The C24ND propeller design parameters are shown in Table 1. The C24ND

Table 1 Propeller parameters for the LSAWT test

Parameter	Value
Blades	3
Diameter (inch)	24
Hub diameter (inch)	4.8
Chord (inch)	1.5
Pitch (inch)	16
Twist (rad)	$\arctan\left(\frac{P}{\pi D \frac{r}{R}}\right)$

propeller uses a helical twist distribution, as shown in Table 1. P is the propeller pitch, D is the diameter, and $\frac{r}{R}$ is the nondimensional radial station of the blade. We use a numerical setup similar to that of Ingraham et al., modeling each blade with 20 spanwise panels and 6 chordwise panels [33]. The propeller was simulated with four full revolutions. The VortexAD results are shown in comparison to the LSAWT experimental data in Figure 8. We see reasonable

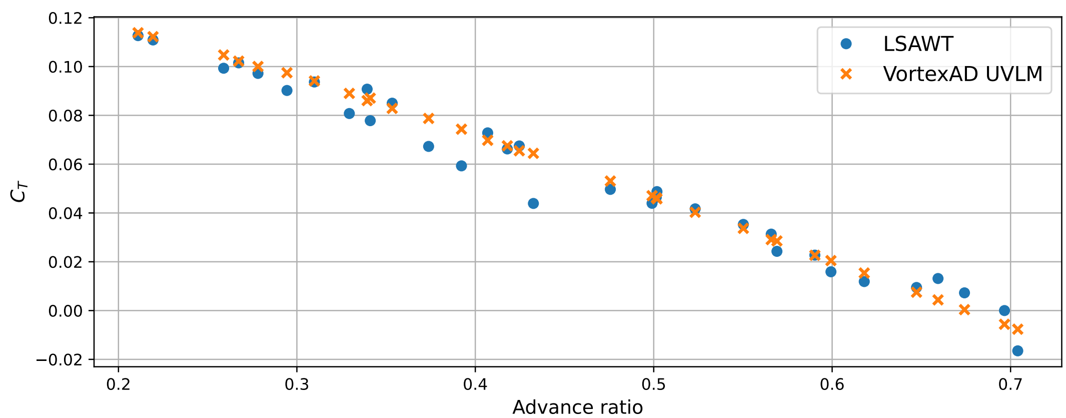
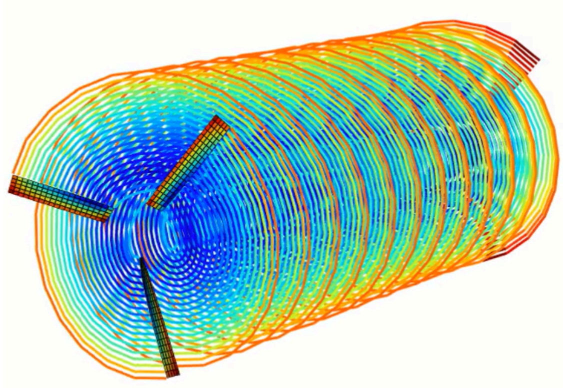
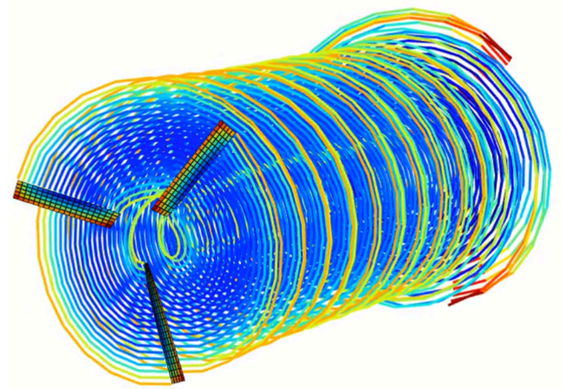


Fig. 8 Comparison of the C24ND thrust coefficient prediction from the VortexAD UVLM solver vs. experimental data across a range of advance ratios. The experimental setup and data are taken from an experimental study conducted by Zawodny et al. [38].

agreement between the VortexAD UVLM results and the experimental data, especially for cases with higher advance ratios. At higher advance ratios, there is less potential for wake mixing and interaction around the blade surfaces due to the higher free-stream velocity. We can also see qualitative variations in the wake for different test cases. The propeller wakes for two of the experimental cases are shown in Figure 9. The higher-thrust solution in Figure 9b experiences a more noticeable slipstream contraction, compared to the low-thrust solution in Figure 9a. Qualitatively, this agrees with momentum theory, which says that the streamtube should contract as the flow accelerates behind the propeller, producing thrust. This is further justification that our free-wake method can reliably capture the relevant unsteady flow physics. In the future, we aim to conduct studies comparing the numerical torque coefficient to experimental data as well, as done in [33]. Viscous drag effects must be accounted for to accurately estimate the torque coefficient; however, potential flow models alone cannot capture this effect. To compute viscous drag effects at each blade section, an airfoil model is required. We aim to continue these studies in the future by coupling an airfoil model with the framework.



(a) RPM = 7487, J = 0.647, C_T = 0.009



(b) RPM = 7199, J = 0.523, C_T = 0.042

Fig. 9 We see qualitative differences in the propeller wakes for two LSAWT test cases with different operating conditions. The colors represent vortex strength. We see a more pronounced slip-stream contraction for the higher thrust-coefficient case.

V. Numerical Experiments of the Free-Wake Model Using Partitioned Vectorization

In this section, we conduct numerical experiments to assess the computational cost of our general potential flow framework, comparing the effects of using full and partitioned vectorization. As mentioned in Section III, partitioned vectorization is a memory-reduction method previously developed for a steady panel code [29]. The previous work has quantified the benefits of partitioned vectorization in a steady setting. Therefore, in this study, we focus on the impact on the memory and run time of the unsteady free-wake solver.

Table 2 summarizes memory and time data for the unsteady free-wake solver with full and partitioned vectorization (with a partition size of 1) across two different grids. For both grid sizes, the PV forward evaluation time is roughly

Table 2 We compare the forward evaluation memory and time of full vectorization vs. partitioned vectorization. The free-wake simulations were conducted with 30 time steps to propagate the wake to a steady state.

Panels	Full vectorization		Partitioned vectorization		Ratios (PV/FV)	
	Memory (GB)	Time (s)	Memory (GB)	Time (s)	Memory ratio	Time ratio
808	0.2887	12.59	0.0253	23.74	0.088	1.886
2649	2.0258	71.15	0.3491	133.04	0.172	1.870

double that of the FV run time; these results agree well with data from our previous numerical experiments on the steady panel method [29]. For the smaller grid size, we see more than an order of magnitude reduction in total memory. However, this ratio worsens with larger grid sizes. This is expected, given that partitioned vectorization helps alleviate the influence of the coefficient on the quadratic scaling that arises from storing intermediate operations in the linear system matrix assembly.

Generally, we expect the memory and time of the unsteady solver to vary more significantly between full vectorization (FV) and partitioned vectorization (PV) than the steady solver for two main reasons. First, the unsteady method requires a linear solve at each time step; as a result, the linear system is generated nt times. We then expect the run time to increase linearly with the number of time steps. Second, the free-wake computes the velocity at the wake nodes induced by the body sources and doublets and the wake doublets; this reintroduces an additional quadratic scaling, now relative to the number of time steps. Fully developed flows require substantial time steps to model the wake completely. As a result, we expect that the quadratic scaling of the free-wake induced velocity computation will dominate the memory cost. This also motivates conducting additional studies to explore the full memory and time Pareto front by varying the partition size.

VI. Conclusion

In this work, we have presented VortexAD, a novel graph-based linear potential flow analysis framework for large-scale MDO. VortexAD models general 3D configurations using a panel method and thin 2D surfaces with a vortex lattice method. We implement our framework in CSDL, a graph-based modeling paradigm, to automate adjoint sensitivity analysis for gradient-based optimization. We utilize a free-wake method with vortex diffusion and dissipation to model unsteady flows. We developed a novel ODE formulation for the free-wake-shedding problem. Our ODE formulation is implemented using Ozone, an open-source graph-based ODE integration framework. With our framework, we conducted verification and validation (V&V) studies across a set of steady and unsteady cases and found reasonable agreement between our numerical results and the data. We then conducted numerical experiments to quantify the benefits of applying partitioned vectorization to our framework to efficiently handle memory usage for unsteady problems. We see noticeable memory reductions with partitioned vectorization, with run time scaling similar to the steady panel method from our prior work [29]. Overall, these studies demonstrate the capability of our potential flow analysis framework to efficiently and accurately model key aerodynamic phenomena for conceptual aircraft design. Our framework is limited to modeling flows that ignore the effects of viscosity. Going forward, we aim to incorporate a viscous model through an integral boundary layer (IBL) approach. In addition, we aim to validate our framework for unsteady flow cases with significant surface-wake interactions, such as propeller-wing configurations, and demonstrate aircraft design optimization.

Acknowledgments

The material presented in this paper is, in part, based upon work supported by NASA under award No. 80NSSC23M0217. The authors would like to thank Dr. Daniel Ingraham from the NASA Glenn Research Center for his help with ideation on the vortex dissipation model and for providing access to experimental data.

References

[1] Schäfer, A. W., Barrett, S. R., Doyme, K., Dray, L. M., Gnadt, A. R., Self, R., O'Sullivan, A., Synodinos, A. P., and Torija, A. J., "Technological, economic and environmental prospects of all-electric aircraft," *Nature Energy*, Vol. 4, No. 2, 2019, pp. 160–166. https://doi.org/https://doi.org/10.1038/s41560-018-0294-x.

[2] Klöwer, M., Allen, M. R., Lee, D. S., Proud, S. R., Gallagher, L., and Skowron, A., "Quantifying aviation's contribution to global warming," *Environmental Research Letters*, Vol. 16, No. 10, 2021, p. 104027.

[3] Graver, B., Zhang, K., and Rutherford, D., "CO2 emissions from commercial aviation," *International Council on Clean Transportation*, 2018.

[4] Borer, N. K., Derlaga, J. M., Deere, K. A., Carter, M. B., Viken, S., Patterson, M. D., Litherland, B., and Stoll, A., "Comparison of aero-propulsive performance predictions for distributed propulsion configurations," *55th AIAA aerospace sciences meeting*, 2017, p. 0209.

[5] Antcliff, K. R., and Capristan, F. M., "Conceptual design of the parallel electric-gas architecture with synergistic utilization scheme (PEGASUS) concept," *18th AIAA/ISSMO multidisciplinary analysis and optimization conference*, 2017, p. 4001.

[6] Jorgensen, M., and Finazzo, T., "Electrified Powertrain Flight Demonstration (EPFD) Project Executive Summary," Tech. rep., National Aeronautics and Space Administration, 2025.

[7] Sarojini, D., Ruh, M. L., Joshy, A. J., Yan, J., Ivanov, A. K., Scotzniovsky, L., Fletcher, A. H., Orndorff, N. C., Sperry, M., Gandarillas, V. E., Asher, I., Chambers, J. T., Gill, H., Lee, S., Cheng, Z., Rodriguez, G., Zhao, S., Mi, C., Nascenzi, T., Cuatt, T., Winter, T. F., Guibert, A. T., Cronk, A., Kim, H. A., Meng, S., and Hwang, J. T., *Large-Scale Multidisciplinary Design Optimization of an eVTOL Aircraft using Comprehensive Analysis*, 2023. https://doi.org/10.2514/6.2023-0146, URL https://arc.aiaa.org/doi/abs/10.2514/6.2023-0146.

[8] Ruh, M. L., Fletcher, A., Sarojini, D., Sperry, M., Yan, J., Scotzniovsky, L., van Schie, S. P., Warner, M., Orndorff, N. C., Xiang, R., Joshy, A. J., Zhao, H., Krokowski, J., Gill, H., Lee, S., Cheng, Z., Cao, Z., Mi, C., Silva, C., Wolfe, L., Chen, J.-S., and Hwang, J. T., *Large-scale multidisciplinary design optimization of a NASA air taxi concept using a comprehensive physics-based system model*, 2024. https://doi.org/10.2514/6.2024-0771, URL https://arc.aiaa.org/doi/abs/10.2514/6.2024-0771.

[9] Adler, E. J., and Martins, J. R. R. A., "Efficient Aerostructural Wing Optimization Considering Mission Analysis," *Journal of Aircraft*, 2022. https://doi.org/10.2514/1.c037096.

[10] Jasa, J. P., Hwang, J. T., and Martins, J. R., “Open-source coupled aerostructural optimization using Python,” *Structural and Multidisciplinary Optimization*, Vol. 57, 2018, pp. 1815–1827.

[11] Gray, A. C., Riso, C., Jonsson, E., Martins, J. R. R. A., and Cesnik, C. E. S., “High-Fidelity Aerostructural Optimization with a Geometrically Nonlinear Flutter Constraint,” *AIAA Journal*, Vol. 61, No. 6, 2023, pp. 2430–2443. https://doi.org/10.2514/1.J062127, URL https://doi.org/10.2514/1.J062127.

[12] McDonnell, T. G., “Gradient-Based Optimization of Highly Flexible Aeroelastic Structures,” Ph.D. thesis, 2023. URL https://www.proquest.com/dissertations-theses/gradient-based-optimization-highly-flexible/docview/2866082369/se-2.

[13] Economon, T., Palacios, F., and Alonso, J., “A coupled-adjoint method for aerodynamic and aeroacoustic optimization,” *12th AIAA Aviation Technology, Integration, and Operations (ATIO) conference and 14th AIAA/ISSMO multidisciplinary analysis and optimization conference*, 2012, p. 5598.

[14] Fabiano, E., and Mavriplis, D., “Adjoint-based aeroacoustic design-optimization of flexible rotors in forward flight,” *Journal of the American Helicopter Society*, Vol. 62, No. 4, 2017, pp. 1–17.

[15] Abdul-Kaiyoom, M. A. S., Yildirim, A., and Martins, J. R. R. A., “Coupled Aeropropulsive Design Optimization of an Over-wing Nacelle Configuration,” *Journal of Aircraft*, Vol. 0, No. 0, 0, pp. 1–23. https://doi.org/10.2514/1.C037678, URL https://doi.org/10.2514/1.C037678.

[16] He, C., and Zhao, J., “Modeling rotor wake dynamics with viscous vortex particle method,” *AIAA journal*, Vol. 47, No. 4, 2009, pp. 902–915.

[17] Tagg, A. C., Anderson, R., Joseph, C., and Ning, A., “Trajectory Optimization of eVTOL and Conventional Aircraft: A Comparative Analysis of Vortex Particle Method and Vortex Lattice+ Blade Element Momentum Theory,” *AIAA AVIATION FORUM AND ASCEND 2024*, 2024, p. 3587.

[18] Alvarez, E. J., Mehr, J., and Ning, A., “FLOWUnsteady: an interactional aerodynamics solver for multirotor aircraft and wind energy,” *AIAA Aviation 2022 Forum*, 2022, p. 3218.

[19] Chauhan, S. S., and Martins, J. R. R. A., “RANS-Based Aerodynamic Shape Optimization of a Wing Considering Propeller–Wing Interaction,” *Journal of Aircraft*, Vol. 58, No. 3, 2021, pp. 497–513. https://doi.org/10.2514/1.C035991, URL https://doi.org/10.2514/1.C035991.

[20] Hess, J. L., and Smith, A. O., “Calculation of potential flow about arbitrary bodies,” *Progress in Aerospace Sciences*, Vol. 8, 1967, pp. 1–138.

[21] Johnson, F. T., “A general panel method for the analysis and design of arbitrary configurations in incompressible flows,” Tech. rep., NASA, 1980.

[22] Morino, L., “Steady, oscillatory, and unsteady subsonic and supersonic aerodynamics, production version (soussa-p 1.1). volume 1: Theoretical manual,” Tech. rep., NASA, 1980.

[23] Ashby, D. L., “Potential flow theory and operation guide for the panel code PMARC,” Tech. rep., NASA, 1999.

[24] Wachspress, D. A., Quackenbush, T. R., and Boschitsch, A. H., “Rotorcraft interactional aerodynamics calculations with fast vortex/fast panel methods,” *ANNUAL FORUM PROCEEDINGS-AMERICAN HELICOPTER SOCIETY*, Vol. 56, AMERICAN HELICOPTER SOCIETY, INC, 2000, pp. 51–71.

[25] Wang, H., Zhou, Z., Xu, X., and Zhu, X., “Influence analysis of propeller location parameters on wings using a panel/viscous vortex particle hybrid method,” *The Aeronautical Journal*, Vol. 122, No. 1247, 2018, pp. 21–41.

[26] Calabretta, J., and McDonald, R., “A Three-Dimensional Vortex Particle-Panel Method for Modeling Propulsion-Airframe Interaction,” *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 2010, p. 679.

[27] Tan, B. T., “Proper orthogonal decomposition extensions and their applications in steady aerodynamics,” 2003.

[28] Gandarillas, V., Joshy, A. J., Sperry, M. Z., Ivanov, A. K., and Hwang, J. T., “A graph-based methodology for constructing computational models that automates adjoint-based sensitivity analysis,” *Structural and Multidisciplinary Optimization*, Vol. 67, No. 5, 2024, p. 76.

[29] Scotzniovsky, L., and Hwang, J. T., *A Fast, Memory-Efficient Panel Method for Large-Scale Multidisciplinary Design Optimization Under Uncertainty Using Graph-Based Modeling*, AIAA AVIATION FORUM AND ASCEND 2025, 2025, p. 3021. https://doi.org/10.2514/6.2025-3021, URL https://arc.aiaa.org/doi/abs/10.2514/6.2025-3021.

[30] Katz, J., and Plotkin, A., *Low-speed aerodynamics*, Vol. 13, Cambridge university press, 2001.

[31] Maskew, B., "Program VSAERO theory Document: a computer program for calculating nonlinear aerodynamic characteristics of arbitrary configurations," Tech. rep., NASA, 1987.

[32] De Gregorio, F., Visingardi, A., and Iuso, G., "An experimental-numerical investigation of the wake structure of a hovering rotor by PIV combined with a Γ2 vortex detection criterion," *Energies*, Vol. 14, No. 9, 2021, p. 2613.

[33] Ingraham, D., "Low-noise propeller design with the vortex lattice method and gradient-based optimization," *AIAA SCITECH 2023 Forum*, 2023, p. 2039.

[34] Sperry, M. Z., and Hwang, J. T., "Ozone: an open-source ordinary differential equation solver for gradient-based optimization," *Optimization and Engineering*, 2025, pp. 1–39.

[35] Ladson, C. L., Hill, A. S., and Johnson Jr, W. G., "Pressure distributions from high Reynolds number transonic tests of an NACA 0012 airfoil in the Langley 0.3-meter transonic cryogenic tunnel," Tech. rep., NASA, 1987.

[36] Gregory, N., and O'reilly, C., "Low-speed aerodynamic characteristics of NACA 0012 aerofoil section, including the effects of upper-surface roughness simulating hoar frost," 1970.

[37] Theodorsen, T., "General theory of aerodynamic instability and the mechanism of flutter," Tech. rep., 1949.

[38] Zawodny, N. S., Pettingill, N. A., Lopes, L. V., and Ingraham, D. J., "Experimental validation of an acoustically and aerodynamically optimized UAM proprotor part 1: test setup and results," 2023.