

A Stable Vortex Particle Method Formulation for Meshless Large Eddy Simulation

Eduardo J. Alvarez* and Andrew Ning[†]
Brigham Young University, Provo, UT, 84602, USA

We present a novel formulation of the vortex particle method (VPM) as a large eddy simulation (LES) in a meshless scheme that is numerically stable. A new set of VPM governing equations are derived from the LES-filtered Navier-Stokes equations. The new equations reinforce conservation of angular momentum by resizing the vortex elements subject to vortex stretching. In addition to the VPM reformulation, a new anisotropic dynamic model of subfilter-scale (SFS) vortex stretching is developed. This SFS model is well suited for turbulent flows with coherent vortical structures where the predominant cascade mechanism is vortex stretching. Mean and fluctuating components of turbulent flow and Reynolds stresses are validated through simulation of a turbulent round jet. The computational efficiency of the scheme is showcased in the simulation of an aircraft rotor in hover, showing our meshless LES to be 100x faster than a mesh-based LES with similar fidelity. The implementation of our meshless LES scheme is released as an open-source software, called [FLOWVPM](#).

Nomenclature			r	Radial position	m
<i>Physical Variables and Properties</i>			t	Time	s
ω	Vorticity field, $\omega(\mathbf{x}, t)$	1/s	$\forall$	Volume	m^3
$\mathbf{u}$	Velocity field, $\mathbf{u}(\mathbf{x}, t)$	m/s	<i>Mathematical Variables and Functions</i>		
$\mathbf{x}$	Position	m	Γ_p	Vortex strength of p -th particle, $\Gamma_p(t)$	m^3/s
ν	Kinematic viscosity	m^2/s	δ	Dirac delta	
ρ	Density	kg/m^3	$\mathbf{E}_{\text{adv}}$	SFS vorticity advection, $(\mathbf{E}_{\text{adv}})_i \equiv \partial T'_{ij}/\partial x_j$	$1/\text{s}^2$
ξ	Local or global enstrophy	$1/\text{s}^2$	$\mathbf{E}_{\text{str}}$	SFS vortex stretching, $(\mathbf{E}_{\text{str}})_i \equiv -\partial T_{ij}/\partial x_j$	$1/\text{s}^2$
C_T	Thrust coefficient, $C_T = T/\rho n^2 d^4$		$\mathbf{x}_p$	Position of p -th particle, $\mathbf{x}_p(t)$	m
d	Diameter	m	σ	Filter width (smoothing radius or core size)	m
n	Revolutions per second	1/s	σ_p	Filter width at p -th particle, $\sigma_p(t) \equiv \sigma(\mathbf{x}_p, t)$	m
R	Radius	m	ζ	Radial basis function	

*Ph.D., Currently: Aeronautical R&D Engineer, Whisper Aero Inc., AIAA Member.

[†]Associate Professor, Mechanical Engineering, AIAA Associate Fellow. Corresponding author, aning@byu.edu.

ζ_σ	Filter kernel, $\zeta_\sigma(\mathbf{x}) \equiv \zeta(\ \mathbf{x}\ /\sigma)/\sigma^3$	$1/\text{m}^3$	<i>Symbols and Operators</i>
C_d	SFS model coefficient, $C_d(\mathbf{x}, t)$		
g_σ	Regularizing function, $g_\sigma(\mathbf{x}) \equiv 4\pi \int_0^{\ \mathbf{x}\ /\sigma} \zeta(t) t^2 dt$	$\langle \rangle$	Lagrangian average operator
T_{ij}	SFS vorticity stress tensor, $T_{ij} \equiv \overline{u_i \omega_j} - \overline{u_i} \overline{\omega_j}$	m/s^2	$\overline{(\)}$ Filter operator

I. Introduction

Large eddy simulation (LES) is a class of computational fluid dynamics (CFD) that filters the Navier-Stokes equations to decompose small-scale fluctuations of the flow from large ones. The large scales are then resolved directly, while the effects of the smaller scales are modeled. The most common LES methods use a mesh or grid to discretize the space and calculate fluxes and derivatives, classified as finite volume/element/difference methods. Significant user effort is spent in the generation and manipulation of the mesh, with studies showing that about 67% of engineering time in mesh-based CFD is spent in these efforts. [1]

Vortex methods [2-4] are a class of meshless CFD solving the Navier-Stokes equations in their velocity-vorticity form. This form is solved in a Lagrangian scheme, which conserves vortical structures over long distances with minimal numerical dissipation.

We focus on the vortex particle method [5] (VPM), a particular vortex method, that uses particles to discretize the Navier-Stokes equations, with the particles representing radial basis functions that construct a continuous vorticity field [6]. While a meshless VPM is not the best approach for all problems, it can be well suited for wake-dominant flows with unbounded domains. Some advantageous characteristics of the VPM include (1) it does not suffer from the Courant–Friedrichs–Lewy (CFL) condition, (2) it has minimal numerical dissipation, (3) its velocity field derivatives are calculated analytically rather than approximated through a stencil, and (4) when used with a fast multipole method it is computationally efficient. Furthermore, the VPM is spatially second-order accurate [7], it is highly efficient since elements are placed only where vorticity exists, the spatial discretization can be automatically adapted [8-10] in the fashion of an adaptive mesh refinement, and simulations are highly-parallelizable in heterogeneous CPU and GPU architectures [11, 12]. Its major disadvantages are numerical instability (the subject of this paper) and imposing solid boundary conditions without significant computational penalty.

VPM has gained popularity in recent years due to a growing need to predict complex aerodynamic interactions in modern electric aircraft [13-16] at a computational cost fit for preliminary design [17, 18]. For example, VPM has been used for high-fidelity simulation of rotorcraft forward flight [19] and multirotor interactions [20, 21], and as a mid and low-fidelity tool for ground effect [22, 23], electrical vertical takeoff and landing (eVTOL) [24], distributed electric propulsion [25, 26], aeromechanics of unconventional rotorcraft [27-29], rotor-rotor interactions [30, 31], and wind energy [32-34]. Furthermore, a couple of open-source VPM codes have been recently released [35-37]. Fig. 1 shows

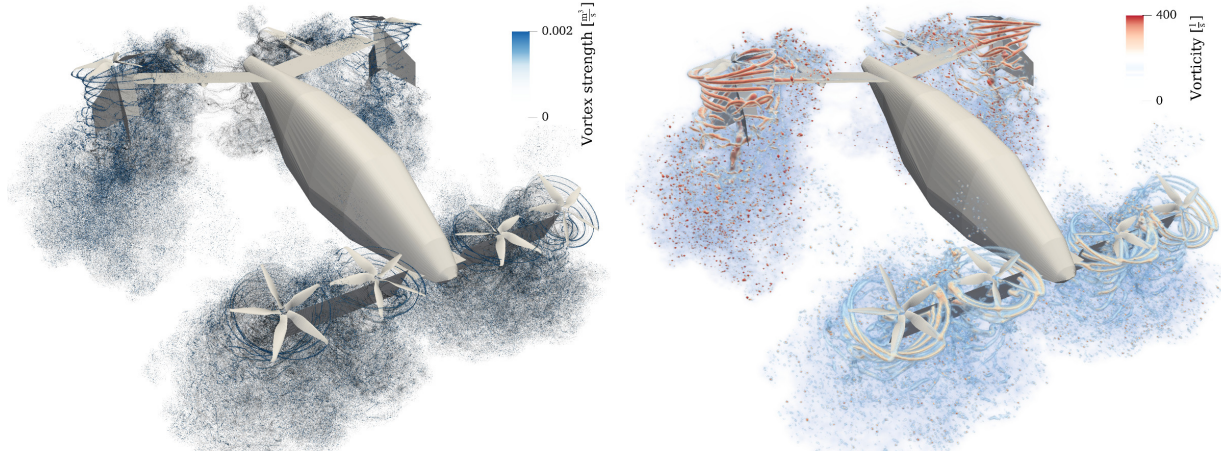


Fig. 1 Meshless LES of eVTOL aircraft using reformulated VPM: (left) computational elements (vortex particles and strength), and (right) volume rendering of vorticity field.

the simulation of an eVTOL aircraft using VPM.

In spite of its growing popularity, VPM is known to be numerically unstable when vortical structures break down close to the turbulent regime. This has limited its range of applications in the aforementioned studies to mostly benign cases with well-behaved numerics (*e.g.*, coarse simulations of rotors with axial inflow). The instability is triggered when vortex stretching gives rise to a rapid increase of local vorticity, which the low numerical dissipation of the method fails to damp out. Such numerical instability is believed to be caused by a combination of Lagrangian distortion, a vorticity field that is not divergence-free, and the absence of subfilter-scale (SFS) turbulent diffusion, among other reasons.

Multiple meshless schemes have been developed over the years to address Lagrangian distortion [9, 38, 39], divergence of the vorticity field [40, 41], and SFS effects [42, 43]. While some of these schemes are effective in two-dimensional cases, they have not succeeded at making the three-dimensional VPM numerically stable when vortex stretching arises. Cottet [2] suggested that the divergence of the vorticity arises from unphysical small scales produced by the computation, which could be properly addressed with subfilter-scale diffusion. This challenge motivated further development of the vortex-in-cell (VIC) method [44–46], also known as vortex particle-mesh method. In such method, the particles are projected onto a background mesh at every time step, and vortex stretching, viscous diffusion, and the Biot-Savart law are computed in a mesh-based scheme coupled with a diffusive SFS model. This approach has shown to be numerically stable, recently enabling the study of wake dynamics with unprecedented fidelity [47–53]. However, the introduction of this mesh also forfeits a few of the aforementioned benefits of a purely Lagrangian (meshless) scheme.

In this study, we propose a formulation of the VPM that is both numerically stable and meshless. Numerical stability is achieved formulating the VPM as an LES and capturing SFS effects. A new set of VPM governing equations are derived from the LES-filtered Navier-Stokes equations in Section III. In Section III, the classic VPM is shown to be

only one out of multiple possible formulations of these governing equations.^{*} In Section IV, we propose an alternative formulation, which uses the particle size to reinforce local conservation of angular momentum. This formulation, referred to as the *reformulated VPM*, is implemented and released as the open-source software [FLOWVPM](#).[†]

In addition to the reformulated VPM, a novel anisotropic structural model of subfilter-scale (SFS) vortex stretching is developed in Section V. The means for backscatter control are also provided, along with a dynamic procedure for the automatic computation of the model coefficient. This SFS model is apt for both meshless and mesh-based CFD, and is well suited for turbulent flows with coherent vortical structures where the predominant cascade mechanism is vortex stretching.

The scheme comprised of the reformulated VPM and SFS model is tested in Section VI, validating the scheme as an LES method. Large-scale turbulent dynamics are validated through simulation of a turbulent round jet, where predicted fluctuations and Reynolds stress are compared to experimental measurements. The numerical stability and computational efficiency of our meshless LES is showcased through the simulation of a rotor in hover. Finally, our method’s ability to resolve arbitrarily small scales is tested resolving the wake of an aircraft propeller.

II. Fundamentals of the Vortex Particle Method

A new set of VPM governing equations are derived from the LES-filtered Navier-Stokes equations in their vorticity form. Along the way, the derivation here presented introduces all the fundamental concepts necessary to bring the reader up to speed with LES and vortex methods.

A. Vorticity Form of the Navier-Stokes Equations

The vorticity equation for an incompressible fluid is obtained by taking the curl of the Navier-Stokes linear-momentum equation:

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

where D is the total or material derivative, $\mathbf{u}(\mathbf{x}, t)$ is the velocity field, ν is the kinematic viscosity, and $\boldsymbol{\omega}(\mathbf{x}, t) = \nabla \times \mathbf{u}(\mathbf{x}, t)$ is the vorticity field.

This equation depends on $\boldsymbol{\omega}$ alone since $\mathbf{u}$ can be calculated from $\boldsymbol{\omega}$ through the Biot-Savart law. Equation (1) can be interpreted as a form of conservation of angular momentum [54], which is further discussed in Section IV.A. From the right-hand side, we see that the evolution of the vorticity field is governed by vortex stretching (first term) and viscous diffusion (second term).

^{*}The particular formulation corresponding to the classic VPM is shown to locally violate conservation of mass and angular momentum under vortex stretching in the [supplementary document](#), explaining why the classic VPM tends to be numerically unstable.

[†]The code is available at <https://github.com/byuflowlab/FLOWVPM.jl>

B. Large Eddy Simulation Equation

We now derive the filtered vorticity Navier-Stokes equation—here referred to as *large-eddy simulation (LES) equation*—and define the stress tensor that transfers energy between resolved scales and subfilter scales.

Let ϕ be a field and ζ_σ a filter kernel, the filtered field is denoted by a bar and defined as

$$\bar{\phi}(\mathbf{x}) \equiv \int_{-\infty}^{\infty} \phi(\mathbf{y}) \zeta_\sigma(\mathbf{x} - \mathbf{y}) d\mathbf{y}, \quad (2)$$

where the filter ζ_σ is associated to a certain cutoff length scale σ . In this study, ζ_σ is defined as $\zeta_\sigma(\mathbf{x}) \equiv \frac{1}{\sigma^3} \zeta(\frac{\|\mathbf{x}\|}{\sigma})$ where ζ is a radial basis function, and is required to have a volume integral of unity, $\int_{-\infty}^{\infty} \zeta_\sigma(\mathbf{y}) d\mathbf{y} = 1$.

In order to derive the LES version of the vorticity equation, Eq. (1) is written in tensor notation and filtered as

$$\frac{\partial \bar{\omega}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{\omega}_i}{\partial x_j} = \bar{\omega}_j \frac{\partial \bar{u}_i}{\partial x_j} + \nu \nabla^2 \bar{\omega}_i. \quad (3)$$

In this equation, $\bar{u}_j \frac{\partial \bar{\omega}_i}{\partial x_j}$ and $\bar{\omega}_j \frac{\partial \bar{u}_i}{\partial x_j}$ are non-linear terms that cannot be calculated from resolved quantities, but are rather approximated through a tensor T_{ij} that encapsulates the error between $\overline{u_i \omega_j}$ and $\bar{u}_i \bar{\omega}_j$ as

$$\overline{u_i \omega_j} = \bar{u}_i \bar{\omega}_j + T_{ij}. \quad (4)$$

The gradient of T_{ij} and its transpose result in

$$\frac{\partial T'_{ij}}{\partial x_j} = \bar{u}_j \frac{\partial \bar{\omega}_i}{\partial x_j} - \bar{u}_j \frac{\partial \bar{\omega}_i}{\partial x_j} \quad \text{and} \quad \frac{\partial T_{ij}}{\partial x_j} = \bar{\omega}_j \frac{\partial \bar{u}_i}{\partial x_j} - \bar{\omega}_j \frac{\partial \bar{u}_i}{\partial x_j} \quad (5)$$

Replacing this into Eq. (3) and using the derivative-filter commutation property, we obtain the LES vorticity equation:

$$\frac{\partial \bar{\omega}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{\omega}_i}{\partial x_j} = \bar{\omega}_j \frac{\partial \bar{u}_i}{\partial x_j} + \nu \nabla^2 \bar{\omega}_i - \frac{\partial T'_{ij}}{\partial x_j} + \frac{\partial T_{ij}}{\partial x_j}. \quad (6)$$

The term $\frac{\partial T'_{ij}}{\partial x_j}$ represents the subfilter-scale (SFS) contributions arising from the advective term (vorticity advection), while $\frac{\partial T_{ij}}{\partial x_j}$ represents the contributions arising from vortex stretching. T_{ij} is associated to the SFS vorticity stress that encapsulates the interactions between large scale dynamics and SFS dynamics and has to be modeled in terms of resolved quantities. The accuracy of LES hinges on the modeling of this tensor. Its divergence represents the rate at which enstrophy—a measure of rotational kinetic energy—is transferred from resolved scales to subfilter scales (diffusion) and from subfilter scales to resolved scales (backscatter). In Section V we will develop a model that approximates the vortex-stretching component of this tensor, and provide the means for backscatter control.

For convenience, we write Eq. (6) in vector notation as

$$\frac{d}{dt}\bar{\omega} = (\bar{\omega} \cdot \nabla) \bar{\mathbf{u}} + \nu \nabla^2 \bar{\omega} - \mathbf{E}_{\text{adv}} - \mathbf{E}_{\text{str}}, \quad (7)$$

where $(\mathbf{E}_{\text{adv}})_i \equiv \frac{\partial T'_{ij}}{\partial x_j}$ is the SFS vorticity advection, $(\mathbf{E}_{\text{str}})_i \equiv -\frac{\partial T_{ij}}{\partial x_j}$ is the SFS vortex stretching, and the $\frac{d}{dt}$ operator is a linearized version of the filtered material derivative, $\frac{d}{dt}() \equiv \frac{\partial}{\partial t}() + (\bar{\mathbf{u}} \cdot \nabla)()$.

C. Viscous Diffusion

One common practice for solving non-linear partial differential equations (PDE), like Eq. (7), is that of splitting the PDE operator into a linear sum of its non-linear pieces. This permits discretizing and solving each non-linear piece in a separate numerical scheme. In this study we will split the PDE in Eq. (7) into inviscid and viscous pieces as

$$\frac{d}{dt}\bar{\omega} = \left(\frac{d}{dt}\bar{\omega}\right)_{\text{inviscid}} + \left(\frac{d}{dt}\bar{\omega}\right)_{\text{viscous}}, \quad (8)$$

where

$$\begin{aligned} \left(\frac{d}{dt}\bar{\omega}\right)_{\text{inviscid}} &= (\bar{\omega} \cdot \nabla) \bar{\mathbf{u}} - \mathbf{E}_{\text{adv}} - \mathbf{E}_{\text{str}} \\ \left(\frac{d}{dt}\bar{\omega}\right)_{\text{viscous}} &= \nu \nabla^2 \bar{\omega}. \end{aligned} \quad (9)$$

Over the years, multiple Lagrangian schemes have been developed that accurately resolve the viscous component, like the vortex redistribution method [55, 56], particle strength exchange [57], and core spreading [58], to name a few. In this study, viscous diffusion will be solved through the core spreading method coupled with the radial basis function (RBF) interpolation approach for spatial adaptation developed by Barba [6, 38, 59]. This viscous scheme diffuses the vorticity by thickening each particle's core size σ over time, while using an RBF interpolation to reset core sizes when they have overgrown. As shown by Rossi [58], the core spreading method has second-order spatial convergence, while showing linear convergence when coupled with spatial adaptation.

In the following sections we focus on the inviscid part of the PDE, developing a scheme for its numerical solution.

D. Lagrangian Discretization: The Vortex Particle

The material derivative in Eq. (1) and the material-conservative nature of the vorticity makes the ω field especially well fit for a Lagrangian description. We now discretize the vorticity equation with Lagrangian elements, termed *vortex particles*. Each particle represents a volume of fluid that is convected by the velocity field carrying an integral quantity of vorticity.

The unfiltered ω field is discretized with singular vortex particles of positions $\mathbf{x}_p$ and coefficients Γ_p , approximating

ω as

$$\omega(\mathbf{x}, t) \approx \sum_p \Gamma_p(t) \delta(\mathbf{x} - \mathbf{x}_p(t)), \quad (10)$$

where δ is the Dirac delta. Each particle travels with the local velocity as

$$\frac{d}{dt} \mathbf{x}_p = \mathbf{u}(\mathbf{x}_p), \quad (11)$$

where $\mathbf{x}_p$ is the position of the p -th particle. Thus, each coefficient Γ_p (termed *vortex strength*) represents the average vorticity that is carried in the volume of each particle.

Using the singular particle approximation in the filtered vorticity $\bar{\omega}$,

$$\bar{\omega}(\mathbf{x}) = \int_{-\infty}^{\infty} \omega(\mathbf{y}) \zeta_{\sigma}(\mathbf{x} - \mathbf{y}) d\mathbf{y} \approx \int_{-\infty}^{\infty} \left(\sum_p \Gamma_p \delta(\mathbf{y} - \mathbf{x}_p) \right) \zeta_{\sigma}(\mathbf{x} - \mathbf{y}) d\mathbf{y}, \quad (12)$$

we obtain an approximation of the filtered vorticity field,

$$\bar{\omega}(\mathbf{x}) \approx \sum_p \Gamma_p \zeta_{\sigma}(\mathbf{x} - \mathbf{x}_p). \quad (13)$$

As seen in Eq. (13), the filter operator has the effect of spreading the vortex strength Γ_p in space, regularizing the singularity originally introduced by the Dirac delta in Eq. (10). Thus, the filter kernel takes the role of a basis function that is used to discretize and approximate the filtered vorticity field through particles. We let the filter width σ change in time and space according to the evolution of each individual particle. Here on, the filter width is referred to as *smoothing radius* or *core size*, denoted σ_p , and defined as $\sigma_p(t) \equiv \sigma(\mathbf{x}_p, t)$. We approximate the filtered vorticity as $\bar{\omega} \approx \omega_{\sigma}$, with

$$\omega_{\sigma}(\mathbf{x}, t) \equiv \sum_p \Gamma_p(t) \zeta_{\sigma_p}(\mathbf{x} - \mathbf{x}_p(t)) \quad (14)$$

and

$$\zeta_{\sigma_p}(\mathbf{x} - \mathbf{x}_p(t)) = \frac{1}{\sigma_p^3(t)} \zeta\left(\frac{\|\mathbf{x} - \mathbf{x}_p(t)\|}{\sigma_p(t)}\right). \quad (15)$$

Replacing the original filter ζ_{σ} with the variable-width filter ζ_{σ_p} deviates from the derivation of the classic VPM. The numerical properties of the classic VPM were proved through formal analysis over the first years of development of the method [60–63]. At this point, it is not well understood how those properties hold for the formulation developed in

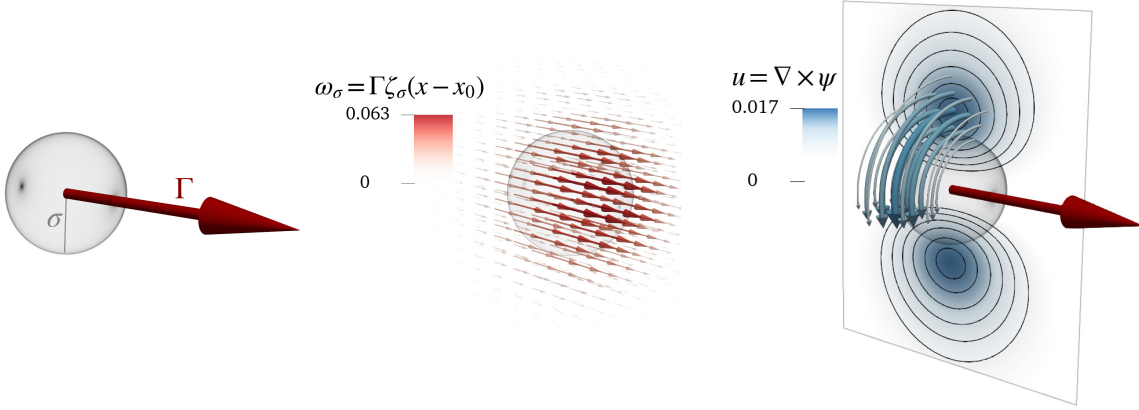


Fig. 2 Vortex particle with unit size and strength: (left) core size σ and vortex strength Γ , (middle) vorticity field, and (right) contours of velocity field and streamlines (represented as curved arrows).

this study, however, the results presented in Section VI suggest similar convergence and improved stability. Formal analysis will be needed in future work to better understand the numerical properties of the reformulated VPM. In particular, the variable-width filter introduces an error from commuting filter and differentiation operators in the LES equations that will be derived in the next section. However, the work by Ghosal and Moin [64] suggests that this error is second order in the filter width, hence it is assumed to be negligible or part of the discretization error of the VPM. Also, Eq. (13) introduces numerical issues by approximating $\bar{\omega}$ through a field that is in general not divergence-free. Cottet [2] suggested that the divergence of $\bar{\omega}$ arises from unphysical small scales produced by the computation, which could be properly addressed with subfilter-scale diffusion. The divergence of $\bar{\omega}$ will be treated in part with the relaxation proposed by Pedrizzetti [40], which, as shown in Section VI, will be sufficient to attain both numerical stability and physical accuracy with the LES formulation and subfilter-scale model developed in this study.

The rotational part of the velocity field is calculated from the particles using the Helmholtz decomposition $\bar{\mathbf{u}} = \nabla \times \psi$ where ψ is some vector potential. This is done by analytically solving the Poisson equation $\nabla^2 \psi = -\bar{\omega}$, which solution is a regularized Biot-Savart law. Fig. 2 shows a vortex particle with unit size and strength, the spreading of the vortex strength by the filter kernel, and the resulting velocity field.

E. General VPM Governing Equations

We will now use vortex particles to discretize the LES vorticity equation, Eq. (7), and derive the equations governing the evolution of the Lagrangian elements. For ease of notation, here on we denote the filtered velocity field $\bar{\mathbf{u}}$ simply $\mathbf{u}$ and use $\frac{D}{Dt}$ and $\frac{d}{dt}$ interchangeably. Also, time dependence is no longer explicitly indicated, but $\mathbf{x}_p$, Γ_p , and σ_p are time-dependent variables.

Starting from the inviscid part of the LES-filtered vorticity equation,

$$\frac{d}{dt} \bar{\omega} = (\bar{\omega} \cdot \nabla) \mathbf{u} - \mathbf{E}_{\text{adv}} - \mathbf{E}_{\text{str}}, \quad (16)$$

we write the filter operator explicitly,

$$\frac{d}{dt} \left(\int_{-\infty}^{\infty} \omega(\mathbf{y}) \zeta_{\sigma}(\mathbf{x} - \mathbf{y}) d\mathbf{y} \right) = \left[\left(\int_{-\infty}^{\infty} \omega(\mathbf{y}) \zeta_{\sigma}(\mathbf{x} - \mathbf{y}) d\mathbf{y} \right) \cdot \nabla \right] \mathbf{u}(\mathbf{x}) - \mathbf{E}_{\text{adv}}(\mathbf{x}) - \mathbf{E}_{\text{str}}(\mathbf{x}). \quad (17)$$

Using the singular particle approximation, $\omega(\mathbf{y}) \approx \sum_q \Gamma_q \delta(\mathbf{y} - \mathbf{x}_q)$, both integrals collapse resulting in

$$\frac{d}{dt} \left(\sum_q \Gamma_q \zeta_{\sigma_q}(\mathbf{x} - \mathbf{x}_q) \right) = \left[\left(\sum_q \Gamma_q \zeta_{\sigma_q}(\mathbf{x} - \mathbf{x}_q) \right) \cdot \nabla \right] \mathbf{u}(\mathbf{x}) - \mathbf{E}_{\text{adv}}(\mathbf{x}) - \mathbf{E}_{\text{str}}(\mathbf{x}). \quad (18)$$

The derivative on the left-hand side is expanded in the supplementary document as:

$$\frac{d}{dt} \left(\sum_q \Gamma_q \zeta_{\sigma_q}(\mathbf{x} - \mathbf{x}_q) \right) = \sum_q \left[\frac{d\Gamma_q}{dt} \zeta_{\sigma_q}(\mathbf{x} - \mathbf{x}_q) + \Gamma_q \frac{d}{dt} \left(\zeta_{\sigma_q}(\mathbf{x} - \mathbf{x}_q) \right) \right] \quad (19)$$

$$= \sum_q \left\{ \frac{d\Gamma_q}{dt} \zeta_{\sigma_q}(\mathbf{x} - \mathbf{x}_q) + \Gamma_q \frac{\partial \zeta_{\sigma_q}}{\partial t}(\mathbf{x} - \mathbf{x}_q) + \Gamma_q [(\mathbf{u}(\mathbf{x}) - \mathbf{u}(\mathbf{x}_q)) \cdot \nabla] \zeta_{\sigma_q}(\mathbf{x} - \mathbf{x}_q) \right\} \quad (20)$$

while vortex stretching is rearranged as

$$\left[\left(\sum_q \Gamma_q \zeta_{\sigma_q}(\mathbf{x} - \mathbf{x}_q) \right) \cdot \nabla \right] \mathbf{u}(\mathbf{x}) = \sum_q \zeta_{\sigma_q}(\mathbf{x} - \mathbf{x}_q) (\Gamma_q \cdot \nabla) \mathbf{u}(\mathbf{x}). \quad (21)$$

Evaluating at the position of the p -th particle, $\mathbf{x} = \mathbf{x}_p$, and pulling the p -th term out of each sum,

$$\frac{d}{dt} \left(\sum_q \Gamma_q \zeta_{\sigma_q}(\mathbf{x}_p - \mathbf{x}_q) \right) = \frac{d\Gamma_p}{dt} \zeta_{\sigma_p}(\mathbf{0}) + \Gamma_p \frac{\partial \zeta_{\sigma_p}}{\partial t}(\mathbf{0}) \quad (22)$$

$$+ \sum_{q \neq p} \left\{ \frac{d\Gamma_q}{dt} \zeta_{\sigma_q}(\mathbf{x}_p - \mathbf{x}_q) + \Gamma_q \frac{\partial \zeta_{\sigma_q}}{\partial t}(\mathbf{x}_p - \mathbf{x}_q) + \Gamma_q [(\mathbf{u}(\mathbf{x}_p) - \mathbf{u}(\mathbf{x}_q)) \cdot \nabla] \zeta_{\sigma_q}(\mathbf{x}_p - \mathbf{x}_q) \right\} \quad (23)$$

and

$$\left[\left(\sum_q \Gamma_q \zeta_{\sigma_q}(\mathbf{x}_p - \mathbf{x}_q) \right) \cdot \nabla \right] \mathbf{u}(\mathbf{x}_p) = \zeta_{\sigma_p}(\mathbf{0}) (\Gamma_p \cdot \nabla) \mathbf{u}(\mathbf{x}_p) + \sum_{q \neq p} \zeta_{\sigma_q}(\mathbf{x}_p - \mathbf{x}_q) (\Gamma_q \cdot \nabla) \mathbf{u}(\mathbf{x}_p). \quad (24)$$

Thus, Eq. (18) evaluated at $\mathbf{x} = \mathbf{x}_p$ becomes

$$\frac{d\Gamma_p}{dt} \zeta_{\sigma_p}(\mathbf{0}) + \Gamma_p \frac{\partial \zeta_{\sigma_p}}{\partial t}(\mathbf{0}) + \mathbf{M}_p^0 = \zeta_{\sigma_p}(\mathbf{0}) (\Gamma_p \cdot \nabla) \mathbf{u}(\mathbf{x}_p) + (\mathbf{M}_p^1 + \mathbf{M}_p^2) - (\mathbf{E}_{\text{adv}}(\mathbf{x}_p) + \mathbf{E}_{\text{str}}(\mathbf{x}_p)), \quad (25)$$

where

$$\mathbf{M}_p^0 \equiv \sum_{q \neq p} \left(\frac{d\Gamma_q}{dt} \zeta_{\sigma_q}(\mathbf{x}_p - \mathbf{x}_q) + \Gamma_q \frac{\partial \zeta_{\sigma_q}}{\partial t}(\mathbf{x}_p - \mathbf{x}_q) \right) \quad (26)$$

$$\mathbf{M}_p^1 \equiv - \sum_{q \neq p} \Gamma_q (\mathbf{u}(\mathbf{x}_p) - \mathbf{u}(\mathbf{x}_q)) \cdot \nabla \zeta_{\sigma_q}(\mathbf{x}_p - \mathbf{x}_q) \quad (27)$$

$$\mathbf{M}_p^2 \equiv \sum_{q \neq p} \zeta_{\sigma_q}(\mathbf{x}_p - \mathbf{x}_q) (\Gamma_q \cdot \nabla) \mathbf{u}(\mathbf{x}_p) \quad (28)$$

These $\mathbf{M}$ -terms pose an interdependence between the p -th particle and neighboring particles, which arises from filtering the vorticity equation. In this study we will neglect this interdependence as explained in Section III. However, we will see in Section V that $\mathbf{M}_p^1$ and $\mathbf{M}_p^2$ are closely related to advection and vortex stretching at the subfilter scale.

Recalling that ζ_σ is defined as $\zeta_\sigma(\mathbf{x}) = \frac{1}{\sigma^3} \zeta\left(\frac{\|\mathbf{x}\|}{\sigma}\right)$, its time derivative is calculated as

$$\frac{\partial \zeta_\sigma}{\partial t}(\mathbf{x}) = -3 \frac{1}{\sigma^4} \frac{\partial \sigma}{\partial t} \zeta\left(\frac{\|\mathbf{x}\|}{\sigma}\right) + \frac{1}{\sigma^3} \frac{\partial}{\partial t} \left(\zeta\left(\frac{\|\mathbf{x}\|}{\sigma}\right) \right) \quad (29)$$

$$= -3 \frac{1}{\sigma} \frac{\partial \sigma}{\partial t} \zeta_\sigma(\mathbf{x}) - \frac{1}{\sigma^3} \frac{\partial \zeta}{\partial r} \left(\frac{\|\mathbf{x}\|}{\sigma} \right) \frac{\|\mathbf{x}\|}{\sigma^2} \frac{\partial \sigma}{\partial t}. \quad (30)$$

Assuming that $\zeta(r)$ reaches a maximum at $r = 0$ (i.e., $\frac{\partial \zeta}{\partial r}(0) = 0$) and evaluating at $\mathbf{x} = \mathbf{0}$, we get

$$\frac{\partial \zeta_\sigma}{\partial t}(\mathbf{0}) = -3 \frac{1}{\sigma} \frac{\partial \sigma}{\partial t} \zeta_\sigma(\mathbf{0}). \quad (31)$$

Since viscous effects have been set aside through operator splitting, $\frac{\partial \sigma}{\partial t}$ in this derivation only accounts for inviscid effects. For clarity, this means that core spreading due to viscous diffusion must not be included in Eq. (31).

Finally, substituting Eq. (31) into Eq. (25) and assuming $\zeta_{\sigma_p}(\mathbf{0}) \neq 0$, we arrive to the equation governing the evolution of vortex strength,

$$\frac{d\Gamma_p}{dt} = (\Gamma_p \cdot \nabla) \mathbf{u}(\mathbf{x}_p) + 3\Gamma_p \frac{1}{\sigma_p} \frac{\partial \sigma_p}{\partial t} + \frac{1}{\zeta_{\sigma_p}(\mathbf{0})} (-\mathbf{M}_p^0 + \mathbf{M}_p^1 + \mathbf{M}_p^2) - \frac{1}{\zeta_{\sigma_p}(\mathbf{0})} (\mathbf{E}_{\text{adv}}(\mathbf{x}_p) + \mathbf{E}_{\text{str}}(\mathbf{x}_p)). \quad (32)$$

Thus, particle convection as in

$$\frac{d}{dt} \mathbf{x}_p = \mathbf{u}(\mathbf{x}_p), \quad (33)$$

strength evolution as in Eq. (32), some expression for $\frac{\partial \sigma_p}{\partial t}$, and the viscous diffusion equation make up the general governing equations of the vortex particle method that solve the LES-filtered Navier-Stokes vorticity equation.

We see in Eq. (32) that the evolution of the vortex strength is dictated by vortex stretching (first term), inviscid expansion/contraction of the particle size σ_p (second term), a dependence on other particles through the $\mathbf{M}$ terms, and SFS contributions through the $\mathbf{E}$ terms. One obtains different formulations of the VPM depending on how $\frac{\partial \sigma}{\partial t}$, the $\mathbf{M}$ terms, and the SFS $\mathbf{E}$ terms are handled. In particular, we will see in Section III that the classic VPM is equivalent to assuming[‡] $\frac{\partial \sigma}{\partial t} = 0$ while neglecting all $\mathbf{M}$ terms. In Section IV we propose a new formulation that uses $\frac{\partial \sigma}{\partial t} \neq 0$ to reinforce conservations laws in spherical elements. In Section V we discuss existing SFS models for meshless schemes and develop a new anisotropic dynamic model of SFS vortex stretching, $\mathbf{E}_{\text{str}}$.

F. Numerical Schemes

In FLOWVPM—the solver used for this study—vortex stretching is solved in the transposed scheme [5, 65] and the divergence of the vorticity field is treated through the relaxation scheme developed by Pedrizzetti [40]. The time integration of the governing equations is done through a low-storage third-order Runge-Kutta scheme [66]. A Gaussian kernel is used as the LES filter ζ_σ (or VPM radial basis function). The fast multipole method [67, 68] (FMM) is used for the computation of the regularized Biot-Savart law, approximating the velocity field and vortex stretching through spherical harmonics with computational complexity $O(N)$, where N is the number of particles. The FMM computation of vortex stretching is performed through an efficient complex-step derivative approximation [20], implemented in a modified version of the open-source, parallelized code ExaFMM [69, 70]. FLOWVPM is implemented in the Julia language [71], which is a modern, high-level, dynamic programming language for high-performance computing.

An in-depth description of these numerical schemes is provided in the doctoral dissertation accompanying the development of FLOWVPM [72, Chapter 4].

III. Classic Vortex Particle Method

Before further exploring the general equation governing vortex stretching, Eq. (32), we pause to analyze the equation that has been used extensively in the literature throughout the years:

$$\frac{d\Gamma_p}{dt} = (\Gamma_p \cdot \nabla) \mathbf{u}(\mathbf{x}_p). \quad (34)$$

We denote this equation as the *Classic VPM*.

Multiple variations of Eq. (34) have been used over the years. For instance, Gharakhani [41] introduced a new term accounting for the divergence of the approximated vorticity field, Winckelmans and Leonard [5] replaced the differential

[‡]The only instance that the classic VPM uses $\frac{\partial \sigma}{\partial t} \neq 0$ is in solving the viscous diffusion equation through the core spreading scheme, but even in this case the effects of $\frac{\partial \sigma}{\partial t}$ on vortex strength are neglected.

operator in vortex stretching with its transpose, and Mansfield et al. [73, 74] and Cottet [75] reintroduced the SFS contributions. However, all these variations and the classic equation have in common that they all neglect $\frac{\partial \sigma_p}{\partial t}$ and are free of the interdependence posed by the $\mathbf{M}$ -terms in Eq. (32).

We now delineate the assumption underlying the classic VPM that will justify neglecting the $\mathbf{M}$ -terms, termed *localized-vorticity assumption*, as follows.

Given a vorticity field ω that is compact in a small volume $\mathcal{V}$, the average vorticity $\langle \omega \rangle$ is calculated as

$$\langle \omega \rangle = \frac{\int_{\mathcal{V}} \omega(\mathbf{y}) \, d\mathbf{y}}{\mathcal{V}}. \quad (35)$$

The field can be approximated through a radially-symmetric field $\tilde{\omega}$ defined as

$$\tilde{\omega}(\mathbf{x}) \equiv \langle \omega \rangle \mathcal{V} \zeta_{\sigma}(\mathbf{x} - \mathbf{x}_0), \quad (36)$$

where ζ_{σ} is a radial basis function of spread σ and center $\mathbf{x}_0$ approximating the vorticity distribution of the original field ω . If ζ_{σ} is normalized such as to have a volume integral of unity, $\tilde{\omega}$ approximates ω in an average sense since

$$\int_{-\infty}^{\infty} \tilde{\omega}(\mathbf{y}) \, d\mathbf{y} = \int_{\mathcal{V}} \omega(\mathbf{y}) \, d\mathbf{y}. \quad (37)$$

Defining $\Gamma = \langle \omega \rangle \mathcal{V}$, the localized-vorticity field ω can be approximated with a single particle as

$$\omega(\mathbf{x}) \approx \Gamma \zeta_{\sigma}(\mathbf{x} - \mathbf{x}_0). \quad (38)$$

Replacing this in the inviscid part of the vorticity equation,

$$\frac{d}{dt} \omega = (\omega \cdot \nabla) \mathbf{u}, \quad (39)$$

evaluating at $\mathbf{x} = \mathbf{x}_0$, and following similar steps as in Section II.E, we get

$$\frac{d\Gamma}{dt} = (\Gamma \cdot \nabla) \mathbf{u}(\mathbf{x}_0) + 3\Gamma \frac{1}{\sigma} \frac{\partial \sigma}{\partial t}. \quad (40)$$

This is the evolution equation of the vortex strength approximating the field of localized vorticity, discretized with only one particle. In contrast, in Section II.E we showed that a more general vorticity field that is discretized using multiple

particles leads to a governing equation that is slightly different:

$$\frac{d\mathbf{\Gamma}_p}{dt} = (\mathbf{\Gamma}_p \cdot \nabla) \mathbf{u}(\mathbf{x}_p) + 3\mathbf{\Gamma}_p \frac{1}{\sigma_p} \frac{\partial \sigma_p}{\partial t} + \frac{1}{\zeta_{\sigma_p}(\mathbf{0})} (-\mathbf{M}_p^0 + \mathbf{M}_p^1 + \mathbf{M}_p^2), \quad (41)$$

after ignoring the SFS contributions. Here, the $\mathbf{M}$ -terms, as defined in Eqs. (26) to (28), pose a dependence on neighboring particles that arises from having filtered the vorticity equation.

Discarding these $\mathbf{M}$ -terms is equivalent to assuming that the vorticity field can be approximated by the superposition of blobs of fluid with compact vorticity that evolve somewhat independently from each other.[§] This assumption, which we call *localized-vorticity assumption*, reduces Eq. (41) into Eq. (40) and is valid as long as there is no significant particle overlap.[¶] In both derivations of the classic VPM by Winckelmans and Leonard [5] and Cottet and Koumoutsakos [76], these $\mathbf{M}$ -terms are not present due to the construction of the method: In the classic derivation, the unfiltered vorticity equation (Eq. (1)) is discretized with singular particles and only the velocity field is filtered to obtain a regularized field, while in our derivation we discretized the LES-filtered vorticity equation (Eq. (7)) which leads to Eq. (41). In order to bring both approaches into agreement, in Section IV we will use the localized-vorticity assumption in the derivation of our reformulated VPM to neglect these $\mathbf{M}$ -terms.

Interestingly, discarding the $\mathbf{M}$ -terms resembles the LES decomposition approach of truncated basis functions [77]. In such an approach, the flow field is expanded using orthonormal basis functions. The summation of bases is then truncated to define the large-scale field, and the discarded modes represent the range of subfilter scales. In the localized-vorticity assumption, the sum over all the particles is truncated after the leading term, p , while neglecting the contributions of neighboring particles. Hence, the localized-vorticity assumption can be regarded as a secondary LES filter with the neglected $\mathbf{M}$ -terms becoming part of the subfilter-scale contributions.

While it is justifiable to neglect the $\mathbf{M}$ -terms through the localized-vorticity assumption, it is unclear to us what the basis is for the classic VPM to assume $\frac{\partial \sigma_p}{\partial t} = 0$. In fact, we hypothesize that this last assumption is the cause of the numerical instabilities that pervade the classic VPM.

The classic VPM simply regards σ as a numerical parameter with no physical significance. However, Leonard [78] suggested that σ should change according to conservation of mass, and Nakanishi, Ojima, and Maremoto [79, 80] suggested that σ should change according to Kelvin's theorem, which was more recently implemented by Kornev *et al.* [81]. Even though these authors let σ evolve in time, they did not include these effects back in the equation that governs $\mathbf{\Gamma}$, effectively assuming $\frac{\partial \sigma_p}{\partial t} = 0$ in the evolution of $\mathbf{\Gamma}$. In the following chapter we propose a formulation that uses $\frac{\partial \sigma_p}{\partial t} \neq 0$ in the governing equation of $\mathbf{\Gamma}$, while letting σ change as to reinforce conservation of angular momentum,

[§]The interactions neglected here are only the ones pertaining to the evolution of vortex strength, while other interactions are still accounted for in the convection and viscous diffusion of the particles.

[¶]For various reasons, most vortex particles methods in the literature have had a scheme in place to ensure that there is no significant overlap: particle splitting or remeshing to address Lagrangian distortion, or core resetting with RBF interpolation to address core spreading. In this study we use core resetting as described in Section II.C

which will be shown to lead to remarkable numerical stability.

IV. Reformulated Vortex Particle Method

The general governing equations derived in Section II.E call for an expression for $\frac{\partial \sigma_p}{\partial t}$. While the classic VPM simply assumes $\frac{\partial \sigma_p}{\partial t} = 0$, we will now go back to first principles to find some plausible expressions for this term. In Section IV.A we consider the physical implications of angular momentum conservation on our particle model, which guides our search for a candidate formulation of $\frac{\partial \sigma_p}{\partial t}$ in Section IV.B. While other options are considered in the supplementary document, this new formulation, termed *reformulated VPM* is summarized in Section IV.C which reinforces conservation of angular momentum for spherical elements subject to vortex stretching.

A. Conservation of Angular Momentum

In Section II.A we showed that the linear-momentum Navier-Stokes equation can be transformed into an expression that only depends on vorticity, namely Eq. (1). We now point our attention to conservation of angular momentum implied by Eq. (1). The discussion that follows is inspired by the writing of P. A. Davidson [82, 83].

Consider a spherical differential fluid element carrying a mean vorticity ω with moment of inertia I . Due to ω , the element is then rotating at an angular velocity of $\omega/2$ and its angular momentum $\mathbf{L}$ is calculated as [84, p. 82]

$$\mathbf{L} = \frac{I\omega}{2}. \quad (42)$$

Given that the element is spherical (and before strain distorts the element), the pressure field exerts no torque on the element and the only torque-producing forces are due to viscous effects, namely τ_{viscous} . The change of angular momentum is then calculated as

$$\frac{d}{dt}\mathbf{L} = \tau_{\text{viscous}}. \quad (43)$$

Replacing Eq. (42) in Eq. (43), we arrive to the following expression of the total derivative of vorticity^{||}

$$\frac{d}{dt}\omega = -\frac{\omega}{I} \frac{d}{dt}I + \tau_{\text{viscous}}^*, \quad (44)$$

where $\tau_{\text{viscous}}^* = 2\tau_{\text{viscous}}/I$. When viscous effects are ignored, Eq. (44) implies that, in order to conserve angular momentum, the angular velocity (or vorticity) must decrease whenever the moment of inertia increases, and vice versa.

Eqs. (1) and (44) suggest that the vorticity Navier-Stokes equation is simply an expression of the conservation of angular momentum in a spherical fluid element [54]. Furthermore, the first term in the right hand side of Eq. (1)

^{||}More generally, this equation should be expressed with an inertia tensor, but for a spherical body this tensor reduces to a scalar due to the symmetry of the body.

accounts for the change of moment of inertia as

$$(\boldsymbol{\omega} \cdot \nabla) \mathbf{u} = -\frac{\boldsymbol{\omega}}{I} \frac{d}{dt} I, \quad (45)$$

which leads to an increase/decrease of vorticity to conserve momentum as the moment of inertia decreases/increases. Hence, $(\boldsymbol{\omega} \cdot \nabla) \mathbf{u}$ is referred to as vortex stretching as it accounts for the deformation exerted by the velocity field on the fluid element, intensifying the vorticity in the direction that the element is stretched, as illustrated in Fig. 3.

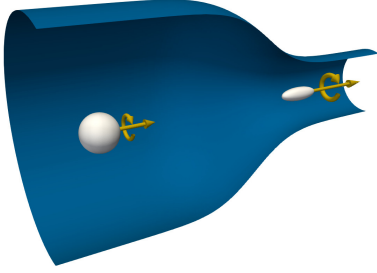


Fig. 3 Stretching of a spherical fluid element and increase in vorticity (or angular velocity).

Equation (45) establishes that vortex stretching must lead to a change in the size of a spherical fluid element. Otherwise, if the vorticity (angular velocity) has changed due to vortex stretching while the size of the element (moment of inertia) is kept constant, the conservation of angular momentum has been locally violated.

We believe that the classic VPM violates conservation of angular momentum when it assumes $\frac{d}{dt} \sigma = 0$. We also hypothesize that this is the reason for its poor numerical stability. In the following sections we will consider some possible formulations that use $\frac{d}{dt} \sigma$ as a degree of freedom to locally reinforce this conservation law.

B. Candidate Formulations

Starting from the general equation governing vortex strength (Eq. (32)), neglecting SFS terms, and using the localized-vorticity assumption, we obtain

$$\frac{d}{dt} \Gamma_p = (\Gamma_p \cdot \nabla) \mathbf{u}(\mathbf{x}_p) + 3\Gamma_p \frac{1}{\sigma_p} \frac{\partial \sigma_p}{\partial t}. \quad (46)$$

1. Sphere with Conservation of Momentum

Given that the moment of inertia of a solid sphere is $I = \frac{2}{5}mr^2$, where $m = \frac{4}{3}\rho\pi r^3$ is the mass of the sphere and r its radius, the inertial term in Eq. (45) becomes

$$\frac{1}{I} \frac{d}{dt} I = \frac{1}{r^5} \frac{d}{dt} (r^5) = \frac{5}{r} \frac{d}{dt} r. \quad (47)$$

Replacing this into Eq. (45) and taking the dot product with ω ,

$$-(\omega \cdot \omega) \frac{5}{r} \frac{d}{dt} r = [(\omega \cdot \nabla) \mathbf{u}] \cdot \omega, \quad (48)$$

results in

$$\frac{1}{r} \frac{d}{dt} r = -\frac{1}{5} \frac{1}{\|\omega\|^2} [(\omega \cdot \nabla) \mathbf{u}] \cdot \omega \quad (49)$$

$$= -\frac{1}{5} [(\hat{\omega} \cdot \nabla) \mathbf{u}] \cdot \hat{\omega}. \quad (50)$$

When ω is filtered and discretized with vortex particles, and the localized-vorticity assumption is applied, the spherical element becomes the vortex particle itself. Vortex strengths are roughly aligned with the vorticity field^{**}, meaning $\hat{\Gamma}_p \approx \hat{\omega}(\mathbf{x}_p)$, and the size of the particle can be expressed as $\sigma_p = \alpha r$ with α some scaling factor, obtaining

$$\frac{1}{\sigma_p} \frac{d}{dt} \sigma_p = -\frac{1}{5} [(\hat{\Gamma}_p \cdot \nabla) \mathbf{u}(\mathbf{x}_p)] \cdot \hat{\Gamma}_p, \quad (51)$$

or, equivalently

$$\frac{d}{dt} \sigma_p = -\frac{1}{5} \frac{\sigma_p}{\|\Gamma_p\|} [(\Gamma_p \cdot \nabla) \mathbf{u}(\mathbf{x}_p)] \cdot \hat{\Gamma}_p. \quad (52)$$

Noticing that, in a Lagrangian scheme, σ_p is only a function of time (*i.e.*, $\frac{d}{dt} \sigma_p = \frac{\partial \sigma_p}{\partial t}$), we replace Eq. (52) into Eq. (46) to obtain the equation governing Γ_p :

$$\frac{d}{dt} \Gamma_p = (\Gamma_p \cdot \nabla) \mathbf{u}(\mathbf{x}_p) - \frac{3}{5} \{[(\Gamma_p \cdot \nabla) \mathbf{u}(\mathbf{x}_p)] \cdot \hat{\Gamma}_p\} \hat{\Gamma}_p. \quad (53)$$

Thus, Eqs. (52) and (53) are the governing equations required for the particles to preserve angular momentum.

2. Other Formulations

Other possible formulations could be devised by considering the elements as cylindrical tubes rather than spheres. In this case, if we apply conservation of mass to the tubes we end up with the same governing equations as shown above.^{††} More generally, we could consider all six combinations of sphere/tube elements, momentum/mass conservation, and the assumption that mass conservation has or has not been already ensured by the convection of the elements. These latter options are derived and briefly discussed in the supplementary document. We find that some are physically implausible, and that the classic VPM lies at the threshold of overstretching, which explains its tendency to be numerically unstable.

^{**}This is especially true when using Pedrizzetti's relaxation scheme [40].

^{††}With the caveat that they are no longer the same when the SFS term is reintroduced.

C. Reformulated Governing Equations

From hereon, this formulation that preserves angular momentum in spherical chunks of the flow is referred to as the *reformulated vortex particle method*, or rVPM. In summary the rVPM equations consist of Eq. (33) resolving vorticity advection by convecting the particles, Eq. (52) governing the evolution of particle size, Eq. (53) governing the evolution of vortex strength, and the viscous part of Eq. (9) is resolved by one of the schemes mentioned in Section II.C. For simplicity, Eqs. (52) and (53) omit the SFS contributions, but they are readily re-incorporated as shown in the supplementary document.

Note that the rVPM equations do not require more computation than the classic VPM since $\frac{d\sigma_p}{dt}$ and $\frac{d\Gamma_p}{dt}$ are calculated directly and solely from vortex stretching, $(\Gamma_p \cdot \nabla) \mathbf{u}(\mathbf{x}_p)$.

V. Anisotropic Dynamic SFS Model

In the previous section we have formulated a numerical scheme for solving the large scales of the LES-filtered vorticity equation,

$$\frac{d}{dt} \bar{\omega} = (\bar{\omega} \cdot \nabla) \bar{\mathbf{u}} + \nu \nabla^2 \bar{\omega} - \mathbf{E}_{\text{adv}} - \mathbf{E}_{\text{str}}. \quad (7)$$

We will now focus on the subfilter-scale (SFS) stresses associated with advection and vortex stretching, $\mathbf{E}_{\text{adv}}$ and $\mathbf{E}_{\text{str}}$, respectively. In Section V.A we briefly discuss existing SFS models relevant to meshless vortex methods. In Section V.B we introduce a novel anisotropic structural model of SFS vortex stretching that is suitable for turbulent flows where the predominant cascade mechanism is vortex stretching. In Sections V.C and V.D we develop a dynamic procedure for computing the model coefficient while also providing the means for backscatter control. Finally, in Section V.E we show how our model can be implemented in conventional mesh-based CFD with a pressure-velocity solver.

A. Existing Models

Over the years, only a few models have been proposed that are suitable for meshless vortex methods. The most popular one is a variant of the Smagorinsky eddy-viscosity model formulated for the vorticity stress [73, 74, 85, 86],

$$\mathbf{E} = -\nabla \times (\nu_{\text{SFS}} \nabla \times \bar{\omega}), \quad (54)$$

where $\mathbf{E} \equiv \mathbf{E}_{\text{adv}} + \mathbf{E}_{\text{str}}$, $\nu_{\text{SFS}} = C_d^2 \sigma^2 \sqrt{2 S_{mn} S_{mn}}$, S_{mn} is the strain-rate tensor, σ is the filter width, and C_d is a model coefficient which is either prescribed or computed dynamically. This functional model and others alike were developed on the basis of homogeneous isotropic turbulence, which makes them overly diffusive in simulations with coherent vortical structures. In the latest developments of the vortex particle-mesh scheme [87, 88], this drawback has been avoided with the variational multiscale method [89–92], however its applicability to a meshless scheme is not clear.

In a different approach, Cottet [75, 93, 94] developed an anisotropic structural model of the advective SFS term $\mathbf{E}_{\text{adv}}$ as

$$\mathbf{E}_{\text{adv}}(\mathbf{x}_p) = C_d \sum_q \mathcal{V}_q (\omega_p - \omega_q) \left[(\mathbf{u}(\mathbf{x}_p) - \mathbf{u}(\mathbf{x}_q)) \cdot \nabla \zeta_{\sigma_q}(\mathbf{x}_p - \mathbf{x}_q) \right], \quad (55)$$

where the model coefficient C_d is usually prescribed with *ad hoc* calibration [95]. This model is reportedly [19, 96] less dissipative than Smagorinsky-type models, however, it was developed on the basis of 2D flow which is absent of vortex stretching. Thus, this model neglects $\mathbf{E}_{\text{str}}$ even though vortex stretching is known to be one of the main mechanisms for enstrophy production in the energy cascade in three dimensions.

To address the need for a low-dissipation SFS model that captures vortex stretching as the physical mechanism for turbulence, in the following sections we develop an anisotropic structural model of SFS vortex stretching with a dynamic model coefficient. To isolate the effects of SFS vortex stretching, in this study we neglect the SFS advection term $\mathbf{E}_{\text{adv}}$ and attempt to only capture the effects of the SFS stretching $\mathbf{E}_{\text{str}}$. In Section VI we will show that $\mathbf{E}_{\text{str}}$ alone is sufficient to obtain stable and accurate simulations in cases that are dominated by vortex stretching and with coherent vortical structures, but it is recommended that $\mathbf{E}_{\text{adv}}$ is added in other cases where advection is the physical mechanism for the development of turbulence.

B. SFS Vortex Stretching Model

Starting from the definition of SFS vortex stretching,

$$E_i^{\text{str}} = -\frac{\partial T_{ij}}{\partial x_j} \equiv -\left(\overline{\omega_j \frac{\partial u_i}{\partial x_j}} - \overline{\omega_j} \frac{\partial \overline{u_i}}{\partial x_j} \right), \quad (56)$$

we group the filter operator as

$$E_i^{\text{str}}(\mathbf{x}) = -\int \omega_j(\mathbf{y}) \left(\frac{\partial u_i}{\partial x_j}(\mathbf{y}) - \frac{\partial \overline{u_i}}{\partial x_j}(\mathbf{x}) \right) \zeta_{\sigma}(\mathbf{x} - \mathbf{y}) d\mathbf{y}. \quad (57)$$

Using the singular particle approximation, $\omega(\mathbf{x}) \approx \sum_q \Gamma_q \delta(\mathbf{x} - \mathbf{x}_q)$, the integral collapses to

$$E_i^{\text{str}}(\mathbf{x}) \approx -\sum_q \Gamma_q^q \left(\frac{\partial u_i}{\partial x_j}(\mathbf{x}_q) - \frac{\partial \overline{u_i}}{\partial x_j}(\mathbf{x}) \right) \zeta_{\sigma}(\mathbf{x} - \mathbf{x}_q). \quad (58)$$

Assuming $\frac{\partial u_i}{\partial x_j} \approx \frac{\partial \overline{u_i}}{\partial x_j}$ and writing in vector notation, the model then becomes

$$\mathbf{E}_{\text{str}}(\mathbf{x}) \approx \sum_q \zeta_{\sigma}(\mathbf{x} - \mathbf{x}_q) (\mathbf{\Gamma}_q \cdot \nabla) (\overline{\mathbf{u}}(\mathbf{x}) - \overline{\mathbf{u}}(\mathbf{x}_q)). \quad (59)$$

Interestingly, both Cottet's model and our model are strikingly similar to the non-localized vorticity terms $\mathbf{M}_p^1$ and $\mathbf{M}_p^2$, respectively (defined in Eqs. (27) and (28)), which are neglected (or filtered out) by the localized-vorticity assumption. Thus, these models can be thought of as soft deconvolutions of the localized-vorticity assumption. Furthermore, introducing a model coefficient, C_d , they can approximate full deconvolutions encompassing the entire spectrum of subfilter scales. This model coefficient is computed dynamically as follows.

C. Dynamic Procedure

1. Derivative Balance

Since SFS vortex stretching needs to be modeled from resolved quantities, the initial equality given in Eq. (56), written here in vector notation

$$\mathbf{E}_{\text{str}} = - \left[\overline{(\boldsymbol{\omega} \cdot \nabla) \mathbf{u}} - (\overline{\boldsymbol{\omega}} \cdot \nabla) \overline{\mathbf{u}} \right], \quad (60)$$

becomes only an approximation once the $\mathbf{E}_{\text{str}}$ model is introduced,

$$\mathbf{E}_{\text{str}} \approx - \left[\overline{(\boldsymbol{\omega} \cdot \nabla) \mathbf{u}} - (\overline{\boldsymbol{\omega}} \cdot \nabla) \overline{\mathbf{u}} \right]. \quad (61)$$

To recover the equality (or at least improve the approximation), we introduce a dynamic model coefficient $C_d(\mathbf{x}, t)$ satisfying

$$C_d \mathbf{E}_{\text{str}} = - \left[\overline{(\boldsymbol{\omega} \cdot \nabla) \mathbf{u}} - (\overline{\boldsymbol{\omega}} \cdot \nabla) \overline{\mathbf{u}} \right]. \quad (62)$$

However, this equation is not useful *per se* since, in order to determine C_d , Eq. (62) requires knowing the SFS quantity $\overline{(\boldsymbol{\omega} \cdot \nabla) \mathbf{u}} - (\overline{\boldsymbol{\omega}} \cdot \nabla) \overline{\mathbf{u}}$, which is exactly what we are trying to model with $\mathbf{E}_{\text{str}}$. However, assuming scale similarity (meaning that C_d is independent of the filter width), we differentiate this equation with respect to the filter width σ to obtain a more useful relation:

$$C_d \frac{\partial \mathbf{E}_{\text{str}}}{\partial \sigma} = - \frac{\partial}{\partial \sigma} \left[\overline{(\boldsymbol{\omega} \cdot \nabla) \mathbf{u}} - (\overline{\boldsymbol{\omega}} \cdot \nabla) \overline{\mathbf{u}} \right]. \quad (63)$$

Using the singular particle approximation, the right-hand side of Eq. (63) before differentiation becomes

$$\left[\overline{(\boldsymbol{\omega} \cdot \nabla) \mathbf{u}} - (\overline{\boldsymbol{\omega}} \cdot \nabla) \overline{\mathbf{u}} \right]_{\mathbf{x}} \approx \sum_q \zeta_{\sigma}(\mathbf{x} - \mathbf{x}_q) (\boldsymbol{\Gamma}_q \cdot \nabla) (\mathbf{u}(\mathbf{x}_q) - \overline{\mathbf{u}}(\mathbf{x})), \quad (64)$$

and after differentiation,

$$\frac{\partial}{\partial \sigma} \left[\overline{(\boldsymbol{\omega} \cdot \nabla) \mathbf{u}} - (\overline{\boldsymbol{\omega}} \cdot \nabla) \overline{\mathbf{u}} \right]_{\mathbf{x}} \approx \sum_q \frac{\partial \zeta_\sigma}{\partial \sigma} (\mathbf{x} - \mathbf{x}_q) (\boldsymbol{\Gamma}_q \cdot \nabla) (\mathbf{u}(\mathbf{x}_q) - \overline{\mathbf{u}}(\mathbf{x})) - \sum_q \zeta_\sigma (\mathbf{x} - \mathbf{x}_q) (\boldsymbol{\Gamma}_q \cdot \nabla) \frac{\partial \overline{\mathbf{u}}}{\partial \sigma}(\mathbf{x}). \quad (65)$$

Recalling that ζ_σ is defined as $\zeta_\sigma(\mathbf{x}) = \frac{1}{\sigma^3} \zeta \left(\frac{\|\mathbf{x}\|}{\sigma} \right)$, its width-derivative is

$$\frac{\partial \zeta_\sigma}{\partial \sigma}(\mathbf{x}) = -\frac{3}{\sigma} \zeta_\sigma(\mathbf{x}) - \frac{\|\mathbf{x}\|}{\sigma^5} \frac{\partial \zeta}{\partial r} \left(\frac{\|\mathbf{x}\|}{\sigma} \right), \quad (66)$$

and assuming that $\zeta_\sigma(\mathbf{x})$ reaches a maximum at $\mathbf{x} = \mathbf{0}$ (meaning, $\frac{\partial \zeta}{\partial r}(0) = 0$),

$$\frac{\partial \zeta_\sigma}{\partial \sigma}(\mathbf{0}) = -\frac{3}{\sigma} \zeta_\sigma(\mathbf{0}). \quad (67)$$

Then, evaluating at $\mathbf{x} = \mathbf{x}_p$, assuming $\zeta_\sigma(\mathbf{0}) \neq 0$, and using the localized-vorticity assumption to neglect all terms $q \neq p$, Eq. (63) becomes

$$C_d(\mathbf{x}_p) \frac{1}{\zeta_\sigma(\mathbf{0})} \frac{\partial \mathbf{E}_{\text{str}}}{\partial \sigma}(\mathbf{x}_p) = \frac{3}{\sigma} (\boldsymbol{\Gamma}_p \cdot \nabla) (\mathbf{u}(\mathbf{x}_p) - \overline{\mathbf{u}}(\mathbf{x}_p)) + (\boldsymbol{\Gamma}_p \cdot \nabla) \frac{\partial \overline{\mathbf{u}}}{\partial \sigma}(\mathbf{x}_p), \quad (68)$$

or

$$C_d \mathbf{m} = \mathbf{L}, \quad (69)$$

with

$$\mathbf{m} \equiv \frac{\sigma^3}{\zeta(0)} \frac{\partial \mathbf{E}_{\text{str}}}{\partial \sigma}(\mathbf{x}_p) \quad (70)$$

$$\mathbf{L} \equiv \frac{3}{\sigma} (\boldsymbol{\Gamma}_p \cdot \nabla) (\mathbf{u}(\mathbf{x}_p) - \overline{\mathbf{u}}(\mathbf{x}_p)) + (\boldsymbol{\Gamma}_p \cdot \nabla) \frac{\partial \overline{\mathbf{u}}}{\partial \sigma}(\mathbf{x}_p). \quad (71)$$

The relation in Eq. (69) will be the basis for our dynamic procedure. This procedure differs from the classic dynamic procedure developed by Germano *et al.* [97, 98] in that the former is motivated by the balance of derivatives between true and modeled SFS contributions given in Eq. (63), while the latter is based on the Germano identity.

2. Enstrophy-Production Balance

The procedure aims at calculating C_d such as to impose the relation given in Eq. (69), however, this is an overdetermined system as there are three equations (one for each spatial dimension) and only one unknown, C_d . Thus, the relation is now contracted by also imposing a balance of enstrophy production between true and modeled SFS

contributions as follows.

Enstrophy, a measure of the rotational kinetic energy of the flow, is defined locally as $\xi \equiv \frac{1}{2} \boldsymbol{\omega} \cdot \boldsymbol{\omega}$. The rate of local enstrophy production is then calculated as

$$\frac{d}{dt} \xi = \boldsymbol{\omega} \cdot \left(\frac{d}{dt} \boldsymbol{\omega} \right), \quad (72)$$

which can be decomposed between resolved and unresolved domains as

$$\frac{d}{dt} \xi = \boldsymbol{\omega} \cdot \left(\frac{d}{dt} \bar{\boldsymbol{\omega}} \right) + \boldsymbol{\omega} \cdot \left[\frac{d}{dt} (\boldsymbol{\omega} - \bar{\boldsymbol{\omega}}) \right], \quad (73)$$

The local enstrophy production in the resolved domain is then defined as

$$\frac{d}{dt} \xi_r \equiv \boldsymbol{\omega} \cdot \left(\frac{d}{dt} \bar{\boldsymbol{\omega}} \right), \quad (74)$$

and the global enstrophy production in the resolved domain is then calculated by integration,

$$\int_{-\infty}^{\infty} \frac{d}{dt} \xi_r \, d\mathbf{y} = \int_{-\infty}^{\infty} \boldsymbol{\omega} \cdot \left(\frac{d}{dt} \bar{\boldsymbol{\omega}} \right) \, d\mathbf{y}. \quad (75)$$

The SFS contribution to the rate of enstrophy production is isolated through operator splitting as

$$\left(\int_{-\infty}^{\infty} \frac{d}{dt} \xi_r \, d\mathbf{y} \right)_{\text{SFS}} \equiv \int_{-\infty}^{\infty} \boldsymbol{\omega} \cdot \left(\frac{d}{dt} \bar{\boldsymbol{\omega}} \right)_{\text{SFS}} \, d\mathbf{y}. \quad (76)$$

Now, we constrain the SFS model to match the enstrophy production of the true SFS contribution,

$$\int_{-\infty}^{\infty} \boldsymbol{\omega} \cdot \left(\frac{d}{dt} \bar{\boldsymbol{\omega}} \right)_{\text{SFS model}} \, d\mathbf{y} = \int_{-\infty}^{\infty} \boldsymbol{\omega} \cdot \left(\frac{d}{dt} \bar{\boldsymbol{\omega}} \right)_{\text{true SFS}} \, d\mathbf{y}. \quad (77)$$

Using the singular particle approximation, $\boldsymbol{\omega}(\mathbf{x}) \approx \sum_p \boldsymbol{\Gamma}_p \delta(\mathbf{x} - \mathbf{x}_p)$, the integrals collapse into

$$\sum_p \boldsymbol{\Gamma}_p \cdot \left(\frac{d}{dt} \bar{\boldsymbol{\omega}} \right)_{(\mathbf{x}_p)}^{\text{SFS model}} = \sum_p \boldsymbol{\Gamma}_p \cdot \left(\frac{d}{dt} \bar{\boldsymbol{\omega}} \right)_{(\mathbf{x}_p)}^{\text{true SFS}}. \quad (78)$$

One instance that this equality is satisfied is when each term in the sum satisfies

$$\boldsymbol{\Gamma}_p \cdot \left(\frac{d}{dt} \bar{\boldsymbol{\omega}} \right)_{(\mathbf{x}_p)}^{\text{SFS model}} = \boldsymbol{\Gamma}_p \cdot \left(\frac{d}{dt} \bar{\boldsymbol{\omega}} \right)_{(\mathbf{x}_p)}^{\text{true SFS}}. \quad (79)$$

Differentiating with respect to the filter width, we arrive to

$$\mathbf{\Gamma}_p \cdot \frac{\partial}{\partial \sigma} \left(\frac{d}{dt} \bar{\omega} \right)_{(\mathbf{x}_p)}^{\text{SFS model}} = \mathbf{\Gamma}_p \cdot \frac{\partial}{\partial \sigma} \left(\frac{d}{dt} \bar{\omega} \right)_{(\mathbf{x}_p)}^{\text{true SFS}}. \quad (80)$$

When SFS advection is neglected, the SFS model and the true SFS become

$$\left(\frac{d}{dt} \bar{\omega} \right)^{\text{SFS model}} = -C_d \mathbf{E}_{\text{str}} \quad (81)$$

$$\left(\frac{d}{dt} \bar{\omega} \right)^{\text{true SFS}} = \overline{(\omega \cdot \nabla) \mathbf{u}} - (\bar{\omega} \cdot \nabla) \bar{\mathbf{u}}, \quad (82)$$

thus Eq. (80) is simply the dot-product of Eq. (63) with $\mathbf{\Gamma}_p$,

$$C_d \mathbf{\Gamma} \cdot \frac{\partial \mathbf{E}_{\text{str}}}{\partial \sigma} = -\mathbf{\Gamma} \cdot \frac{\partial}{\partial \sigma} \left[\overline{(\omega \cdot \nabla) \mathbf{u}} - (\bar{\omega} \cdot \nabla) \bar{\mathbf{u}} \right]. \quad (83)$$

Finally, following the same steps that led from Eq. (63) to Eq. (69), the enstrophy balance in Eq. (80) becomes

$$C_d \mathbf{\Gamma}_p \cdot \mathbf{m} = \mathbf{\Gamma}_p \cdot \mathbf{L}, \quad (84)$$

and C_d is then calculated dynamically as

$$C_d = \frac{\mathbf{\Gamma}_p \cdot \mathbf{L}}{\mathbf{\Gamma}_p \cdot \mathbf{m}}. \quad (85)$$

Thus, this C_d calculated at the position of every particle is the coefficient that approximates the derivative balance while satisfying a local balance of enstrophy production between the model and the true SFS contribution.

3. Lagrangian Average

Similar to the classic procedure based on the Germano identity [97], Eq. (85) poses numerical issues when the denominator is close to zero, leading to large fluctuations. In the classic procedure, Meneveau *et al.* [99] addressed this issue by integrating both numerator and denominator along Lagrangian trajectories (pathlines), effectively building ensemble averages. Applying this technique, our dynamic procedure becomes

$$C_d = \frac{\langle \mathbf{\Gamma}_p \cdot \mathbf{L} \rangle}{\langle \mathbf{\Gamma}_p \cdot \mathbf{m} \rangle}, \quad (86)$$

where $\langle \cdot \rangle$ denotes the Lagrangian integration. As suggested by Meneveau *et al.*, the integration is performed as a relaxation process at every time step of the form

$$\langle \phi \rangle_{\text{new}} = (1 - \alpha) \langle \phi \rangle_{\text{old}} + \alpha \phi, \quad (87)$$

where α is calculated as $\alpha = \Delta t / T \leq 1$, Δt is the time step of the simulation, and T is the time length of the ensemble average. Meneveau *et al.* recommends judiciously choosing the value of T (or equivalently α) such that the instantaneous fluctuations of C_d are smoothed out. In this study we have chosen $\alpha = 0.005$ (roughly equivalent to an averaging window of 200 time steps), which successfully smooths out the large fluctuations of C_d in the simulations presented in Section VI

D. Backscatter Control

For stability reasons, it is common in LES to control the amount of energy being backscattered from the unresolved scales into the resolved scales. The SFS term is purely-dissipative if its contributions to the enstrophy budget decrease the total enstrophy. This is,

$$\left(\int_{-\infty}^{\infty} \frac{d}{dt} \xi_r \, dy \right)_{\text{SFS}} \leq 0. \quad (88)$$

Following the derivation in Section V.C.2, this condition is satisfied if

$$\mathbf{\Gamma}_p \cdot \left(\frac{d}{dt} \bar{\omega} \right)_{(\mathbf{x}_p)}^{\text{SFS model}} \leq 0 \quad (89)$$

for each particle, or equivalently,

$$C_d \mathbf{\Gamma}_p \cdot \mathbf{E}_{\text{str}}(\mathbf{x}_p) \geq 0 \quad (90)$$

Therefore, enstrophy backscatter can be filtered out at each particle by clipping the model coefficient as $C_d = 0$ whenever the condition shown above is not satisfied.

E. Usage in Conventional Mesh-Based CFD

Even though our SFS model is tailored for vortex methods as it only uses the primitive variables of the VPM, the model can also be readily applied to conventional mesh-based CFD as follows. The vortex strength is expressed from mesh quantities as $\mathbf{\Gamma}_q = \langle \omega \rangle_q \mathbf{V}_q$, where $\langle \omega \rangle_q$ and $\mathbf{V}_q$ are the average vorticity and volume associated to each element in the grid (*i.e.*, cells in a finite volume method or nodes in a finite difference method). The model in the vorticity

transport equation then becomes

$$\frac{\partial T_{ij}}{\partial x_j}(\mathbf{x}) = C_d \Delta^{-3} \sum_q \langle \omega \rangle_j^q \mathbb{V}_q \left(\frac{\partial \bar{u}_i}{\partial x_j}(\mathbf{x}_q) - \frac{\partial \bar{u}_i}{\partial x_j}(\mathbf{x}) \right) \zeta \left(\frac{\|\mathbf{x} - \mathbf{x}_q\|}{\Delta} \right), \quad (91)$$

where the index q iterates over each element in the grid, Δ is the width of the grid filter, and ζ is the filter kernel. Furthermore, the model can be used in the pressure-velocity form of the momentum equation noticing that the SFS terms in the vorticity transport equation are simply the curl of the SFS term in the pressure-velocity equation, $\frac{\partial \tau_{ij}}{\partial x_j}$, as

$$\epsilon_{ijk} \frac{\partial}{\partial x_j} \frac{\partial \tau_{kl}}{\partial x_l} = \frac{\partial T'_{ij}}{\partial x_j} - \frac{\partial T_{ij}}{\partial x_j}, \quad (92)$$

where ϵ_{ijk} is the Levi-Civita tensor. Hence, the SFS models of the vorticity transport equation can be used to model $\frac{\partial \tau_{ij}}{\partial x_j}$ by “un-curling” the equation above.

VI. Results

In the preceding sections we have developed a scheme for numerically solving the LES-filtered Navier-Stokes equations in their vorticity form. The proposed scheme uses a reformulation of the VPM and a novel model of SFS vortex stretching to achieve a meshless large eddy simulation.

We now proceed to test and validate both the VPM reformulation and the vortex-stretching SFS model that comprise our meshless LES. Simulations are compared to results reported in the literature from experimental work, large and detached eddy simulation, and unsteady Reynolds-average Navier-Stokes (URANS) simulation. These validation cases also serve as examples on how to impose initial and boundary conditions in our meshless scheme.

In Section [VI.A](#), a turbulent round jet is simulated to test the accuracy and numerical stability of the LES scheme. The evolution of the jet and predicted Reynolds stress are compared to experimental measurements, validating the scheme as an LES method accurately resolving large-scale features of turbulent flow. In Section [VI.B](#), a rotor in hover is simulated with our meshless LES and benchmarked against conventional mesh-based CFD. The computational time is compared to mesh-based CFD to highlight the lightweight cost of our meshless approach. In Section [VI.C](#), the wake of an aircraft propeller is resolved with various levels of refinement to test the ability to resolve arbitrarily small scales. The supplementary document contains additional verification cases for an isolated vortex ring to test convergence, and a leapfrogging vortex ring case to test vortex stretching.

A. Turbulent Round Jet

A jet discharging into a quiescent environment poses a canonical case for the study of turbulence, encompassing a laminar region near the nozzle that breaks down into turbulence away from the nozzle. Experimental measurements

on a round jet were used to test that the reformulated VPM is able to resolve the mean and fluctuating components of turbulent flow, while resolving Reynolds stress directly. The simulation replicated the experiment by Quinn and Militzer [100] which used a contoured nozzle with an exit diameter d of 45.4 mm, discharging air into stagnant ambient air with a centerline velocity U_c of 60 m/s. This corresponds to a Mach number of 0.18 and a diameter-based Reynolds number $Re = U_c d / \nu$ of 2×10^5 .

The simulation assumed a top-hat velocity profile with smooth edges at the nozzle exit given by

$$u_e(r) = U_c \tanh\left(\frac{\frac{d}{2} - r}{\theta}\right), \quad (93)$$

where θ is the momentum thickness of the shear layer, assumed to be $\theta = 0.025d$. As depicted in Fig. 4, this velocity profile was imposed as a boundary condition at $z = 0$ by defining a volumetric cylinder of length h with particles spanning from $z = -h/2$ to $z = h/2$ and performing an RBF fit to the vorticity profile $\omega_\theta = \frac{du_e}{dr}$. This computes the vortex strengths that induce the velocity profile u_e inside the cylinder. The $+z$ half of the cylinder was then removed, the $-z$ half was kept in the computational domain as static particles throughout the simulation, while the set of particles computed at $z = 0$ were injected at each time step as free particles. The resulting boundary condition showed sensitivity to the cylinder length h , leading to a straight jet when $h \geq 2d$, a contracting jet when $h < 2d$, and an actuator disk when $h \rightarrow 0$. Since Quinn and Militzer reported a jet that was slightly contracting, the length of the boundary-condition cylinder was tailored to match the streamwise and radial velocity that they measured at the nozzle exit plane shown in Fig. 5, finding sufficient agreement when $h = 1.7d$.

The simulation was first attempted with the classic VPM, but it quickly ended in numerical blow up at the initial stage of the jet. This is shown in Fig. 6 through an abrupt jump in global enstrophy, where enstrophy is computed as described by Winckelmans [85]. The reformulated VPM proved to be numerically stable in the initial and transition

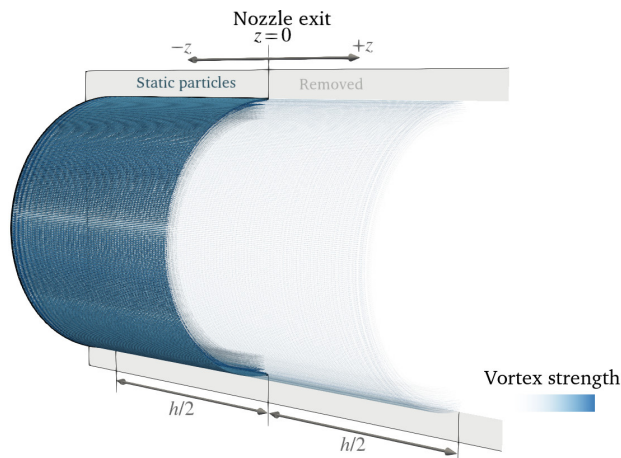


Fig. 4 Boundary condition of round jet simulation defined with particles.

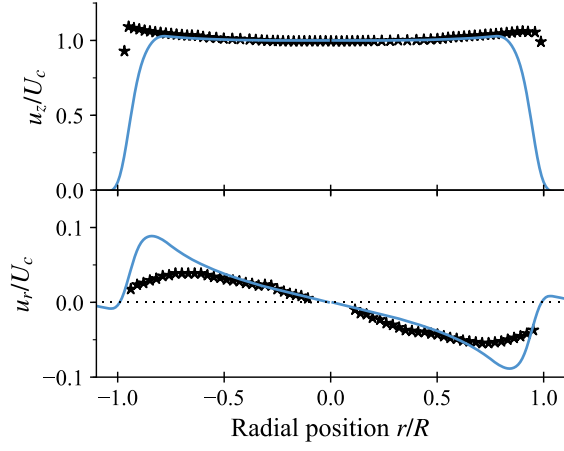


Fig. 5 Streamwise (top) and radial (bottom) velocity close to nozzle exit in experiment by Quinn and Militzer [100] (markers), compared to boundary condition of simulation (solid line) probed at $z/d = 0.1$.

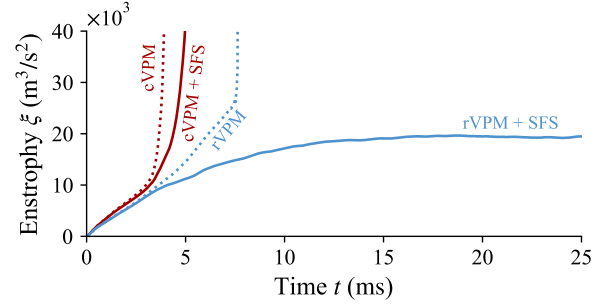


Fig. 6 Global enstrophy of turbulent round jet with classic VPM (cVPM) and reformulated VPM (rVPM), with and without SFS model.

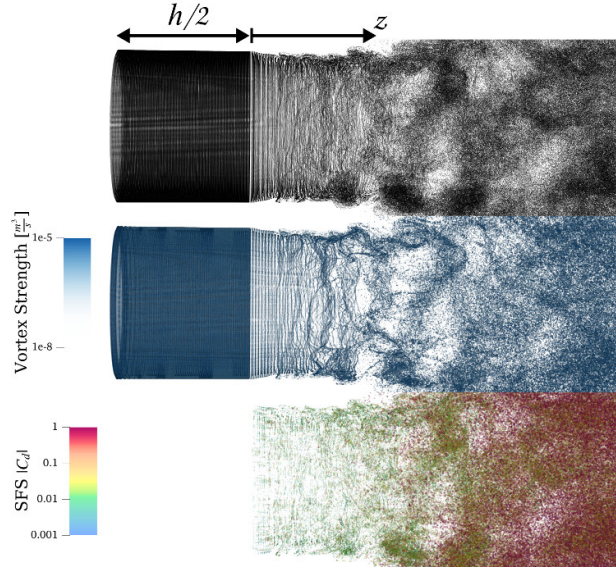


Fig. 7 Vortex particles (top), vortex strength (middle), and SFS model coefficient (bottom) close to jet nozzle at $t = 48$ ms.

stages of the jet, however, the simulation becomes unstable in the fully-developed turbulent regime as enstrophy builds up in the absence of SFS turbulent diffusion. Introducing the SFS model, the enstrophy production of the inlet balances out with the forward-scatter of the fully-developed region and the rVPM simulation becomes indefinitely stable, as shown in Fig. 6. Interestingly, the SFS model did not stabilize the classic VPM simulation, which indicates that both the reformulated VPM and the SFS model are needed to achieve numerical stability.

Once proven stable, the simulation was run for 50 ms with time step $\Delta t = 0.02$ ms until achieving a fully-developed

region that was statistically stationary. This was run on a single node with 128 CPU cores (dual 64-core AMD EPYC 7702, 2.0 GHz) resulting on a wall-clock time of two and a half days using up to 3×10^6 particles. Fig. 7 shows the particle field, vortex strengths, and the SFS model coefficient C_d close to the nozzle at $t = 48$ ms, while Fig. 8 shows the vorticity of the entire field. As seen in Fig. 8, coherent vortical structures form in the initial region ($z < 1d$), which leapfrog and mix transitioning to fully-developed turbulent flow by $z > 3d$. Notice in Fig. 7 that the model coefficient is negligibly small in the initial region and increases to $0.1 < C_d < 1$ in the fully-developed regime. This shows that the dynamic procedure succeeds at automatically calibrating C_d , hindering SFS turbulent diffusion in the laminar regime while facilitating it in turbulent regions.

The formation and time evolution of the jet is available as a video in the supplemental content. Initially, a vortex ring forms at the head of the jet. As the ring travels downstream, a vortex sheet is deployed forming the shear layer. The vortex sheet is stretched as the flow develops, eventually rolling up and forming filaments. The filaments pair up, leapfrog, merge, and breakdown, eventually forming the fully-developed turbulent region. Fig. 9 shows the history of C_d throughout this process, calculated at each time step as the average $|C_d|$ over all the particles where $C_d \neq 0$. Here we see that the average C_d is negligibly small in the initial development of the jet, but it ramps up and converges to a value of 0.43 as the flow becomes fully developed, again confirming the ability of the dynamic procedure to automatically calibrate the SFS model at each flow regime.

To validate the dynamics predicted by the simulation, the velocity was probed at five stations along the jet and statistical properties were compared to the experimental measurements reported by Quinn and Militzer [100]. The data was also supplemented with other experiments of similar round jets compiled by Ball *et al.* [101]. Statistical properties were calculated through temporal and spatial ensemble averages after the flow became statistically stationary in the region $0 \leq z \leq 5d$. Averaging was performed over the time interval $40 \text{ ms} \leq t < 50 \text{ ms}$.

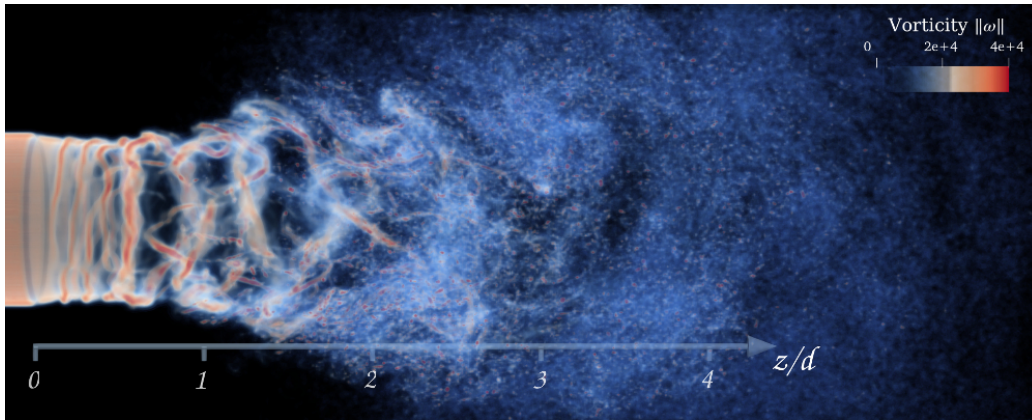


Fig. 8 Volume rendering of vorticity field in turbulent jet at $t = 48$ ms showing distinct flow features: Coherent structures form in the initial region ($z < 1d$) that mix and break down by $z > 3d$.

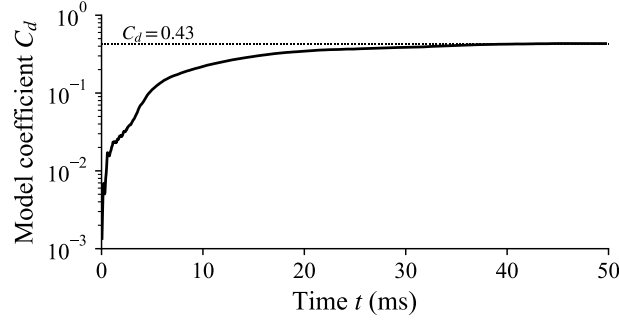


Fig. 9 Average SFS model coefficient in turbulent jet simulation as computed through the dynamic procedure.

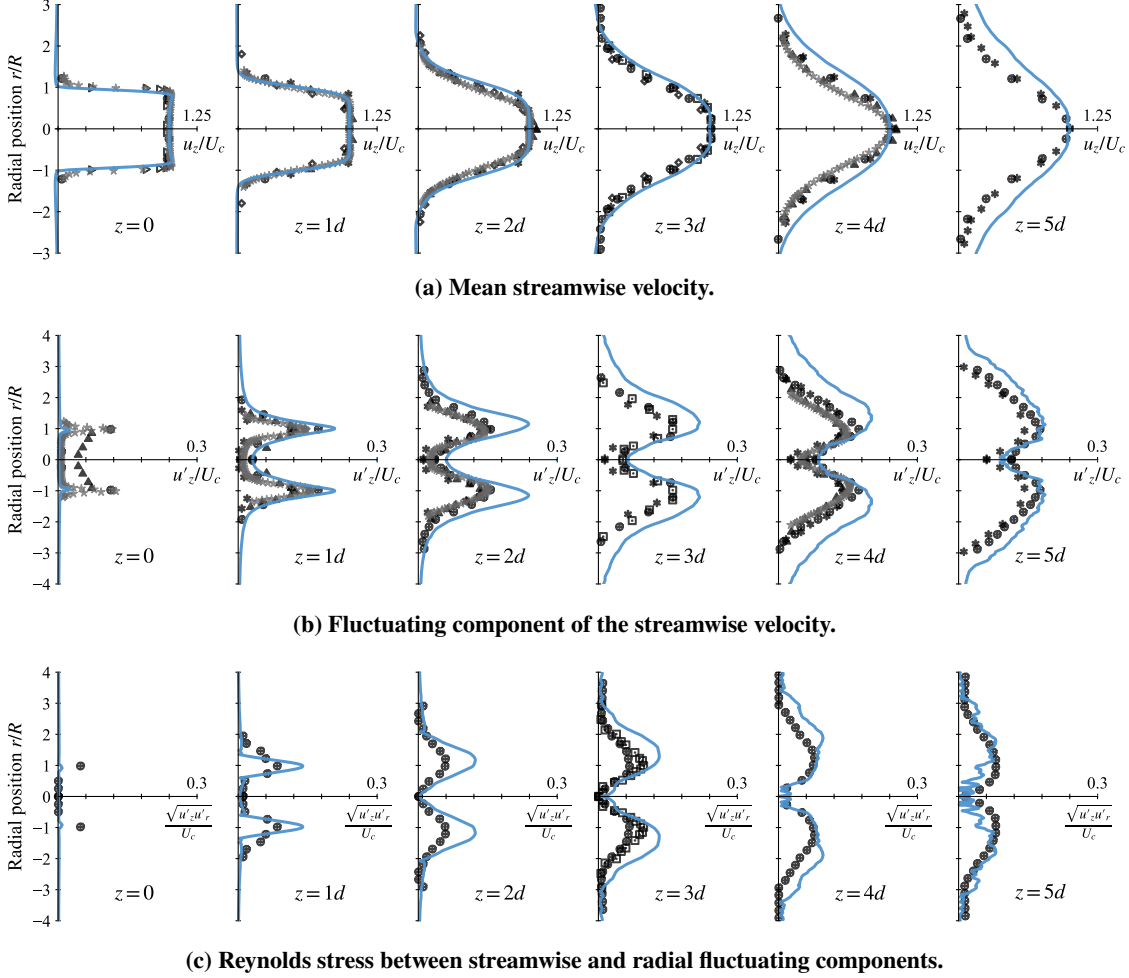


Fig. 10 Profiles along turbulent round jet. Simulation: - rVPM. Experimental: $\star$ Quinn and Militzer [100]; $\oplus$ Fellouah and Pollard [102]; $\triangleright$ Mi *et al.*, contraction nozzle [103]; $\ast$ Iqbal and Thomas [104]; $\diamond$ Romano [105]; $\boxtimes$ Xu and Antonia, contraction nozzle [106].

Fig. 10a shows the mean component of the streamwise velocity profile, u_z , normalized by the centerline velocity at each station. The mean component shows excellent agreement with the experiments in the initial and transition regions ($0 \leq z \leq 3d$) and reasonable agreement in the fully-developed turbulent region ($z \geq 4d$) though slightly overexpanded. Fig. 10b shows the fluctuating component of the streamwise velocity, u'_z , or standard deviation, while Fig. 10c shows the Reynolds stress between streamwise and radial velocity, defined as the square root of the covariance, $\sqrt{u'_z u'_r}$. At $z = 0$, the simulation shows only a small fluctuation and Reynolds stress as the flow is dominated by the boundary condition. Away from the nozzle exit plane ($z > 0$), fluctuations and Reynolds stress concentrate at the shear layer ($r/R = 1$) in the laminar region ($z \approx 1d$), and gradually spread as the jet breaks down in the turbulent regime ($z \geq 4d$). Fluctuations are overpredicted (and, as a consequence, also the Reynolds stress), however this is in within reasonable agreement with the experiments. These predictions can be further improved in future work with the addition of an SFS model of vorticity advection increasing the turbulent diffusion (and damping out fluctuations), or implementing a spatial adaptation strategy to better resolve small scales. However, the current agreement with the experiments suffices to confirm that our scheme is an LES able to resolve mean and fluctuating large-scale features of the flow. Also, its ability to directly resolve Reynolds stress potentially makes the reformulated VPM a higher fidelity approach than Reynolds-average approaches like RANS and URANS (where Reynolds stresses are rather modeled), while being completely meshless.

B. Rotor in Hover

The rotation of blades in static air drives a strong axial flow that is caused by the shedding of tip vortices. This is challenging to simulate since, in the absence of a freestream, the wake quickly becomes fully turbulent and breaks down as tip vortices leapfrog and mix close to the rotor. Thus, a rotor in hover is a good engineering application to showcase the accuracy, numerical stability, and computational efficiency of the reformulated VPM.

In this test case, we simulated the experiment by Zawodny *et al.* [107] consisting of a DJI 9443 rotor in hover at 5400 RPM. This two-bladed rotor is 9.4 inches in diameter, resulting in a tip Mach number of 0.20 and chord and diameter-based Reynolds numbers at 70% of the blade span of 6×10^4 and 7×10^5 , respectively. Our simulations were also compared to unsteady Reynolds-average Navier-Stokes (URANS) results reported by Schenk, [108] obtained with the commercial software STAR-CCM+. The URANS is an unsteady compressible solver with an SST $k-\omega$ turbulence model, resolving the blades with an all- y^+ wall treatment on a rotating mesh surrounding the rotor. It used an unstructured mesh with 14 million cells, mesh refinement down to a y^+ of 30 using a wall model, and time steps equivalent to 3° of rotation on a first-order time integration scheme, which is a rather coarse simulation, but requiring very low computational resources.

The rotating blades are computed in our VPM through an actuator line model (ALM), which is a common practice in LES [109]. Our ALM discretizes the geometry into blade elements, using a two-dimensional viscous panel method to compute forces and circulation along each blade cross section as the blades move, as described in previous work [20].

The vorticity of each blade is introduced in the fluid domain by embedding static particles along the surface that capture the blade's circulation distribution, while shedding free particles at the trailing edge associated with unsteady loading and trailing circulation, as shown in Fig. 11. The frequency of particle shedding per revolution determines the initial spacing Δx in between particles, which, along with the core size σ , determines the spatial resolution at which the wake is being resolved. The initial particle size σ was set as to provide a particle overlap $\lambda = \frac{\sigma}{\Delta x}$ of 2.125 at the blade tip. The VPM simulations used a time step equivalent to a rotation of 1° while shedding particles every 0.5° with 50 blade elements.

First, the simulation was attempted on the classic VPM without the SFS model, which quickly ended in numerical blow up after one revolution, shown in Fig. 12. Introducing the SFS model made the classic VPM noticeably more stable, however, the simulation still blew up before the wake became fully developed. Introducing the reformulated VPM made the simulation completely stable, using up to 1×10^6 particles after 16 revolutions. As shown in Fig. 12, the rate of enstrophy produced by the rotor eventually balances out with the forward scatter of the SFS model. This further asserts the numerical stability gained with the reformulated VPM and SFS model.

Next, the thrust coefficient C_T predicted with rVPM was compared to the experimental coefficient reported by Zawodny *et al.* [107]. We define C_T in the propeller convention as $C_T = \frac{T}{\rho n^2 d^4}$, where T is the dimensional thrust, ρ is air density, and n is rotations per seconds. The experiment reported a mean C_T of 0.072 (Zawodny *et al.* reported the thrust coefficient defined as $C_T = \frac{T}{\rho \pi R^2 (\Omega R)^2}$ which is typical in the rotorcraft community, while here we have converted their measurement to C_T as defined in the propeller convention). In our simulation, the thrust is calculated integrating the force computed by the ALM at each blade element that is immersed in the fluid domain. As shown in Fig. 13, the VPM simulation shows excellent agreement with the experiment, predicting a mean value within 2% of the experimental mean value. This illustrates the capacity of our method to provide accurate predictions in a real engineering application.

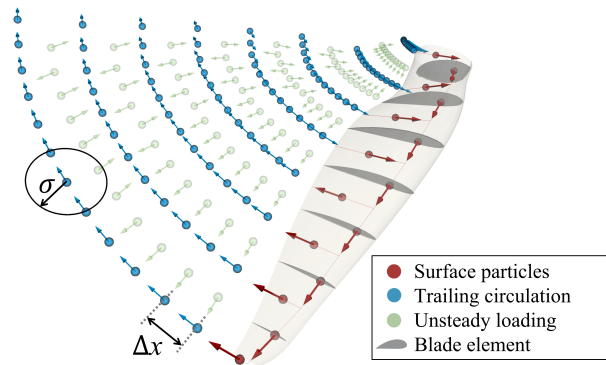


Fig. 11 Actuator line model in rotor simulation. Particles colored by their source of vorticity; arrows indicate direction of vortex strength.

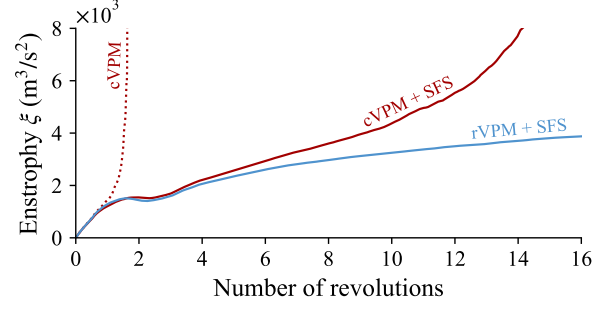


Fig. 12 Global enstrophy in rotor simulation with classic VPM (cVPM) and reformulated VPM (rVPM).

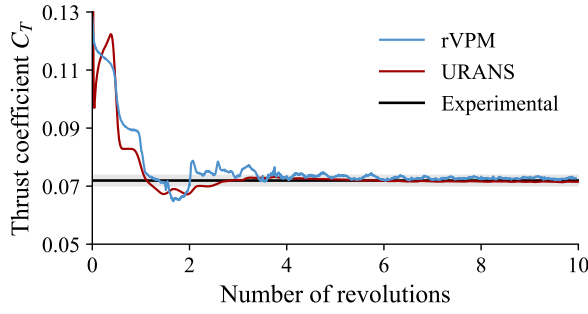


Fig. 13 Thrust history in rotor simulations compared to experimental mean C_T . Shaded region encompasses the 95%-confidence interval of the experiment.

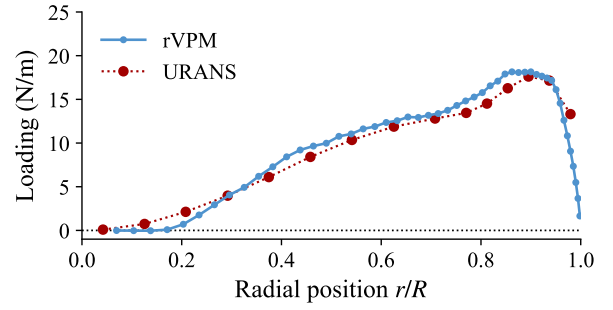


Fig. 14 Time-average blade loading distribution in rotor simulations.

In order to illustrate the low computational cost of our meshless LES, we now compare our simulation to the URANS simulation by Schenk. As previously pointed out, for the sake of minimizing computational effort, the URANS simulation is both spatially and temporally coarse (using only 14 million cells down to a y^+ of only 30, with time steps of 3°), while using a first-order time integration. Thus, this URANS simulation represents the minimum computational effort that is plausible for a simulation of this kind while still being accurate (the C_T predicted by URANS is still within 2% of the experimental mean value, as shown in Fig. 13). Fig. 14 compares the time-average loading distribution along the blade as predicted with rVPM and URANS, showing that both methods can resolve the blade loading with similar accuracy. However, the rVPM has the advantage of being able to accurately preserve the vortical structure of the wake, shown in Fig. 15 with minimal computational effort. The URANS simulation reportedly took about 10 wall-clock hours to resolve 10 revolutions, using 192 CPU cores across 12 nodes on the BYU Fulton supercomputer. This is equivalent to about 1800 core-hours. On the other hand, the rVPM simulation took about 4 wall-clock hours to resolve 10 revolutions on a single node of the BYU Fulton supercomputer with 32 CPU cores, equivalent to about 140 core-hours. Hence, rVPM is 13 times faster (or one order of magnitude faster) than this coarse URANS simulation, while providing LES accuracy.

Table 1 Description of CFD solvers in rotor benchmark.

Simulation	Software	Blade Scheme	Turbulence Model	Computational Elements
rVPM (meshless LES)	FLOWVPM	Actuator line model	Anisotropic dynamic SFS	1M vortex elements
URANS	STAR-CCM+	Blade resolved ($y^+ = 30$, all- y^+ wall treatment)	SST $k-\omega$	14M grid cells
LES-ALM	MIRACLES	Actuator line model	None (numerical dissipation)	50M grid points
LES-IBM	MIRACLES	Immersed boundary method	None (numerical dissipation)	216M grid points
DES	OVERFLOW2	Blade resolved	Spalart-Allmaras DES	260M grid points

While Schenk’s URANS simulation represents the low-fidelity end of mesh-based CFD, Zawodny *et al.* [107] reported a high-fidelity mesh-based simulation using OVERFLOW2. OVERFLOW2 is a detached-eddy simulation (DES) code using a URANS solver with the Spalart-Allmaras turbulence model near solid surfaces, while switching to a subgrid scale formulation in regions fine enough for large eddy simulation. Their simulation of the DJI 9443 rotor used 260 million grid cells with time steps corresponding to 0.25° on a second-order time integration scheme, reportedly predicting a mean C_T within 2.5% of the experimental value. One rotor revolution reportedly took approximately 30 wall-clock hours using 1008 CPU cores on the NAS Pleiades supercomputer. Extrapolating this to 10 revolutions, the DES takes 300 wall-clock hours and 300k core-hours. Thus, recalling that rVPM took only 140 core-hours, the rVPM is 2200x faster (or three orders of magnitude faster) than this DES simulation.

While Schenk’s URANS and Zawodny’s DES represent the low and high end of fidelity in mesh-based CFD, respectively, Delorme *et al.* [110] reported two LES that lay somewhere in between both ends. One LES used an actuator line model (ALM) similar to our meshless LES, while the other used an immersed boundary method (IBM). All solvers previously mentioned are summarized in Table 1 and their computational costs are compared to our meshless LES in Table 2. To be clear, the LES-IBM and DES comparisons are not directly comparable. These latter methods are

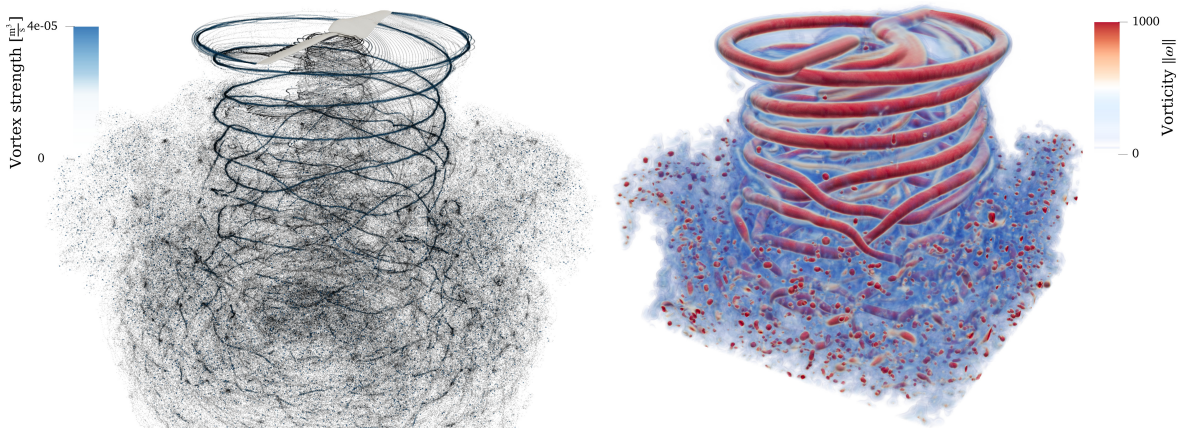


Fig. 15 Meshless LES of rotor in hover after 15 revolutions using rVPM: (left) computational elements (vortex particles and strength), and (right) volume rendering of vorticity field. Video available in supplemental content.

Table 2 Rotor benchmark resolving 10 rotor revolutions. The faded times are blade-resolved simulations that are not directly comparable.

Simulation	CPU Cores	Wall-Clock	Core-Hours	rVPM Speedup
rVPM	32	4.3 hours	140	—
URANS	192	9.6 hours	1.8k	~10x faster
LES-ALM	845	24 hours	20k	~100x faster
LES-IBM	1000	96 hours	96k	~500x faster
DES	1008	300 hours	300k	~1000x faster

blade-resolved simulations and so capture higher-fidelity blade surface loading, important for unsteady aerodynamics and acoustics. Their timings are only included for completeness, with the caution that a direct comparison is not meaningful. The LES by Delorme *et al.*, shown in Fig. 16, is an implicit LES that relies on numerical dissipation to approximate subgrid-scale turbulent diffusion, reportedly predicting a mean C_T within 2% of the experimental value. Out of the aforementioned mesh-based simulations, their LES-ALM simulation is the most akin to our meshless LES since both use an ALM while resolving vortical structures with comparable fidelity, shown in Fig. 15 (right) and Fig. 16 (right). Fig. 17 compares the wake velocity predicted by both simulations and the experiment by Ning [11], showing good agreement (except towards the hub where ALM relies heavily on a hub loss correction to account for the effects of the hub on blade loading). This confirms that both methods lead to comparable accuracy. The main computational advantage of our meshless LES lays in that computational elements are only placed where vorticity is originated and are automatically convected by the flow field, as shown in Fig. 15 (left), while mesh-based LES requires meshing the entire space, as shown in Fig. 16 (left). Their LES-ALM simulation reportedly took a wall-clock time of 24 hours per every 10

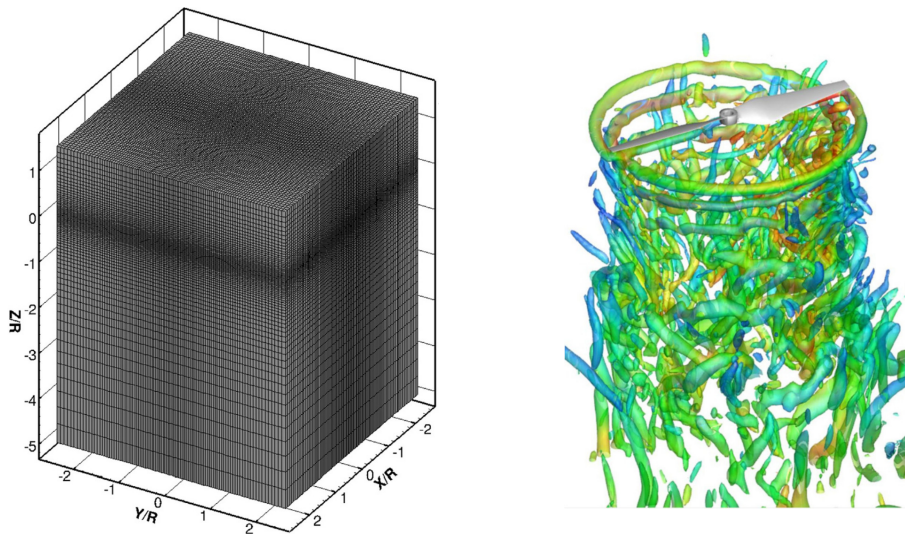


Fig. 16 Mesh-based LES of rotor in hover reported by Delorme *et al.*: (left) computational elements, and (right) iso-surface of Q-criterion colored by velocity magnitude. Reprinted from Aerospace Science and Technology, Vol 108, Delorme *et al*, *Application of Actuator Line Model for Large Eddy Simulation of Rotor Noise Control*, Copyright 2020 Elsevier Masson SAS, with permission from Elsevier.

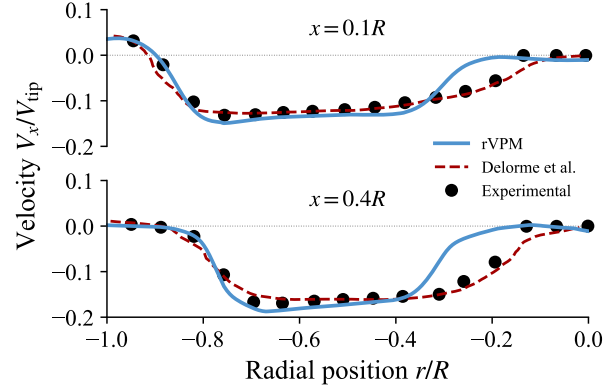


Fig. 17 Time-average velocity in wake of rotor in hover as predicted by our meshless LES (rVPM) and compared to mesh-based LES-ALM by Delorme et al. [110] and experiment by Ning [111]. Our simulation does not include the hub so we expect more significant discrepancies in that region.

rotor revolutions using 845 CPUs, equivalent to 20k core-hours. Compared to the 140 core-hours of the rVPM, our meshless LES is 145x faster (or two orders of magnitude faster) than this mesh-based LES.

In summary, as shown in Table 2 our meshless LES appears to be two orders of magnitude faster than a mesh-based LES with similar fidelity, while being one order of magnitude faster than a low-fidelity URANS simulation. It is difficult to justify an exact comparison between the computational cost of each simulation since each solver used different order-of-convergence schemes, spatial and temporal resolution, and computing hardware. However, the comparison is still qualitatively insightful: our meshless LES can be 10x to 100x faster than conventional mesh-based CFD approaches for unbounded, wake dominant problems.

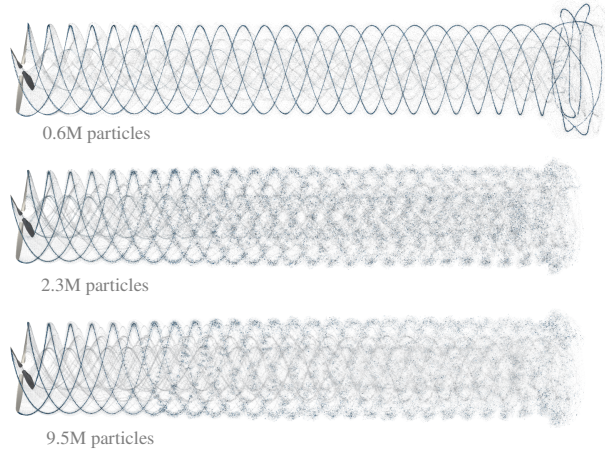


Fig. 18 Computational elements (vortex particles) in propeller case as spatial resolution is increased.

C. Aircraft Propeller

In order to test the ability of our LES to resolve arbitrarily small scales, the wake of an aircraft propeller was simulated with increasingly finer discretization. In this test, we simulated the experimental work by Sinnige et al. [112] and Stokkermans et al. [113], which used the Beaver propeller at an advance ratio of 0.8 and freestream velocity of 40 m/s. This four-bladed propeller is 0.237 m in diameter, resulting in a diameter-based Reynolds numbers of 1.8×10^6 .

The spatial discretization was refined from 0.6M to 9.5M particles by simultaneously increasing the number of blade elements and the frequency that particles are shed off the blades, shown in Fig. 18. In all cases, thrust and torque coefficients predicted by the simulations were within 2% of the measurements reported by Sinnige et al. [112]. The flow field in the finest simulation (9.5M particles) is shown in Fig. 19, with (top) a volume rendering of the instantaneous vorticity and (bottom) a slice of the ensemble-average in-plane vorticity taken as blades intersect the plane. The flow field shows the simulation capturing the fine vortical structure of the wake and its turbulent breakdown as the wake evolves. In between the plane of rotation and $x/R = 3$, the ensemble average shows the inner vortex sheet stretching and folding around tip vortices. At $x/R > 3$, the inner sheet approaches the preceding tip vortex, causing the vortex to deform and break down.

The vortical structure of the second, third, and fourth tip vortices are shown in Fig. 20 as the spatial resolution is increased from 0.6M to 9.5M particles. The simulations are also compared to the experimental observations by Sinnige et al. [112] measured through particle image velocimetry (PIV). In the coarse simulation, the positions of the tip vortices are accurately predicted, however, the vortices and inner sheet are overly diffused. Refining the simulation to 2.3M particles, the size of tip vortices reaches qualitative agreement with the experiment. This is confirmed in Fig. 21 where

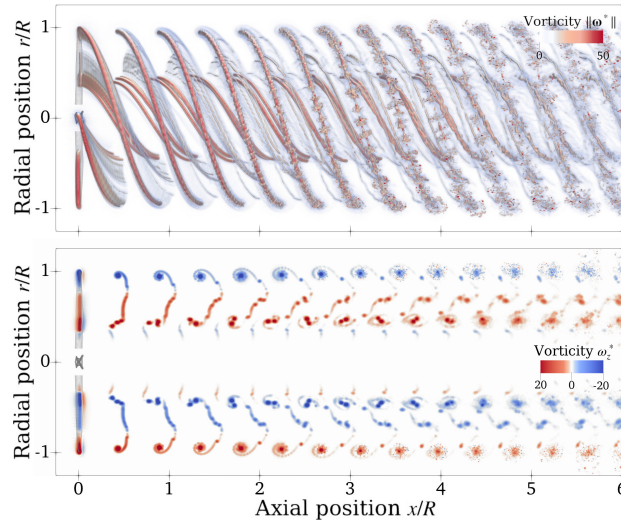


Fig. 19 Flow field in propeller simulation: (top) instantaneous volume rendering of vorticity field and (bottom) slice of ensemble-average in-plane vorticity. Vorticity nondimensionalized as $\omega^* \equiv \omega d / V_{\text{disk}}$, where V_{disk} is the equivalent actuator-disk velocity $2V_{\text{disk}} = V_{\infty} + \sqrt{V_{\infty}^2 + 8T/\rho\pi d^2}$.

the velocity induced by the second tip vortex is shown to converge to the experimental measurement within reasonable agreement. Further refining to 9.5M particles, the simulation accurately resolves the inner sheet stretching and folding around the tip vortices as observed experimentally. This confirms the ability of our LES to resolve increasingly smaller scales as its spatial resolution is increased.

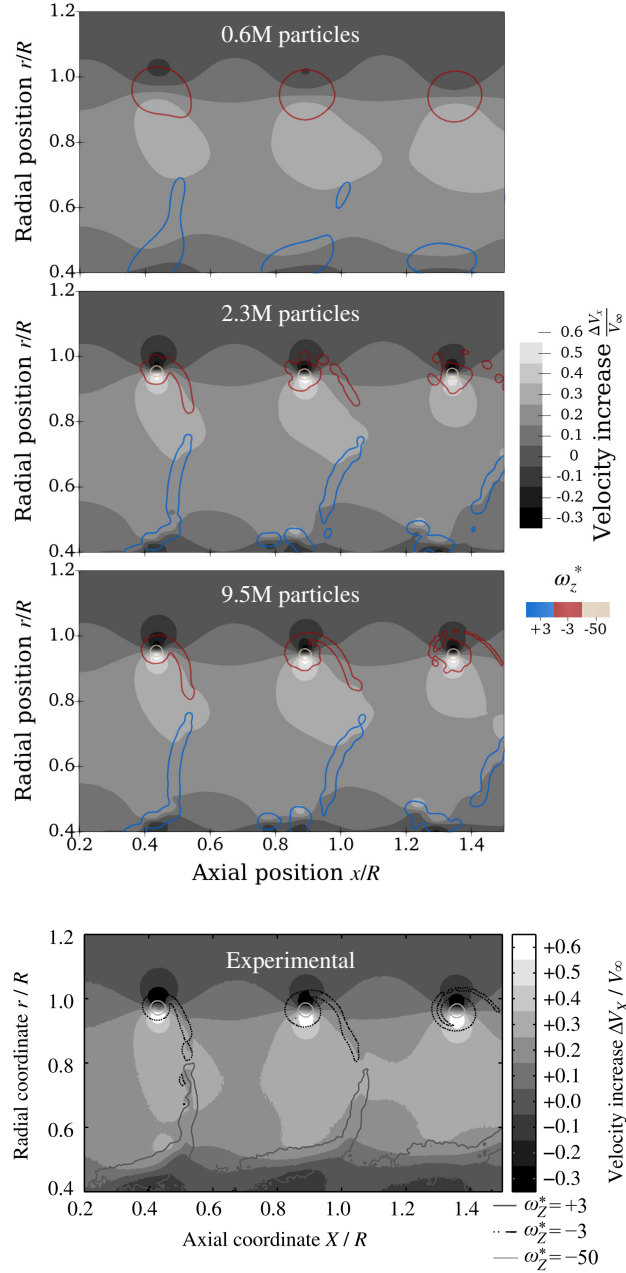


Fig. 20 Ensemble-average flow field as spatial discretization is refined, compared to experimental PIV. Color contour levels show the second, third, and fourth tip vortices and the inner vortex sheet. Black and white contour levels show the induced velocity. Experimental (bottom) reprinted from Sinnige et al. [112] with permission from copyright holder.

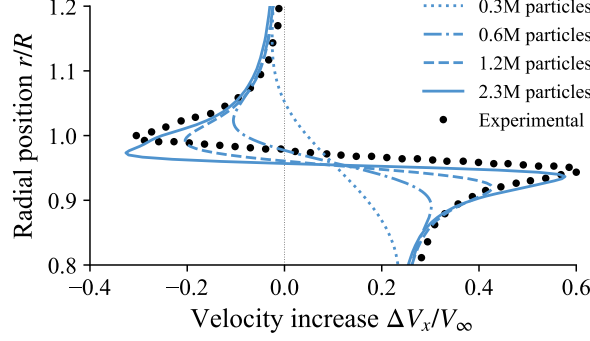


Fig. 21 Ensemble-average axial velocity probed across the second tip vortex as spatial refinement is increased. Experimental measurement reported by Stokkermans et al. [113].

VII. Conclusions

In this study we have developed a meshless scheme for large eddy simulation that is accurate, stable, and computationally efficient. We started off by deriving a general set of VPM governing equations that solve the LES-filtered Navier-Stokes equations. The classic VPM turns out to be one of the formulations arising from these general equations, obtained by assuming $\frac{\partial \sigma_p}{\partial t} = 0$. We then derived a new VPM formulation that uses the particle shape to reinforce local conservation of angular momentum.

In addition to the reformulated VPM, we developed a low-dissipation anisotropic SFS model that uses vortex stretching as the physical mechanism for turbulence. We proposed a novel dynamic procedure for calculating the model coefficient and a strategy for backscatter control. The dynamic procedure is based on a simultaneous balance of enstrophy-production and derivatives between true and modeled SFS contributions. This SFS model is apt for both meshless and conventional mesh-based CFD, and is well suited for flows with coherent vortical structures where the predominant cascade mechanism is vortex stretching.

Validation was presented, asserting the scheme comprised of the reformulated VPM and SFS model as a meshless LES accurately resolving large-scale turbulent dynamics. In the simulation of a turbulent round jet, a favorable agreement with experiments showed that our scheme is able to resolve the mean and fluctuating components of turbulent flow, while also resolving Reynolds stress directly.

Numerical stability and computational efficiency were demonstrated through the simulation of an aircraft rotor in hover. Our meshless LES showed to be two orders of magnitude faster than a mesh-based LES with similar fidelity, while being one order of magnitude faster than a low-fidelity URANS simulation. Finally, the ability of our LES to resolve arbitrarily small scales was verified resolving the wake of an aircraft propeller as the spatial resolution was increased.

While the ideas of this paper show promise in addressing stability issues, multiple areas of future work remain. The approach used here was motivated by simple arguments of angular momentum conservation. While the methodology

has been effective in improving stability, it is unclear what the impacts are on convergence error both spatially and temporally. More rigorous convergence error analyses are needed to quantify these potential errors and improve the methodology. Detailed studies on the relative role of viscosity and the subfilter-scale terms are also needed. For example, to quantify the relative contributions to turbulent transition.

Another major limitation of particle methods is handling solid boundary conditions. Future work should explore a wider variety of boundary conditions aimed at introducing solid boundaries without a mesh. This is important not just for non-lifting bodies and objects like fuselages and the ground, but also to be able to resolve wings, blades, etc., as an alternative to actuator line and actuator surface methods.

VPMs naturally fit with the structure of the incompressible Navier-Stokes equations. However, efforts have extended the VPM to some limited compressible flows [114, 115]. It is yet unclear how the methodology discussed in this paper might apply in these efforts.

Finally, this study limited its computation to a conventional CPU paradigm, but previous work [11, 12] shows a strong potential for massive parallelization and speed up in heterogeneous GPU architectures.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 2006219. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

FLOWVPM uses a modified version of the open-source code ExaFMM [70] originally developed by Lorena Barba and Rio Yokota. FLOWVPM is implemented in the Julia programming language [71] and integrates the open-source software ParaView [116, 117] as its visualization engine. The authors wish to thank Prof. Adrin Gharakhani for the insightful conversations on vortex methods and his contributions to the field during his life.

References

- [1] Hardwick, M. F., Clay, R. L., Boggs, P. T., Walsh, E. J., Larzelere, A. R., and Altshuler, A., “DART System analysis,” Tech. rep., Sandia National Laboratories, 2005. <https://doi.org/SAND2005-4647>.
- [2] Cottet, G.-H., and Koumoutsakos, P. D., *Vortex Methods Theory and Practice*, Cambridge University Press, 2000. <https://doi.org/10.1017/CBO9780511526442>, URL <https://www.cambridge.org/core/product/identifier/9780511526442/type/book>.
- [3] Winckelmans, G. S., “Vortex methods,” *Encyclopedia of Computational Mechanics*, Vol. 3, 2004, pp. 129–153. <https://doi.org/10.1002/0470091355>.
- [4] Mimeau, C., and Mortazavi, I., “A Review of Vortex Methods and Their Applications: From Creation to Recent Advances,” *Fluids*, Vol. 6, No. 2, 2021, p. 68. <https://doi.org/10.3390/fluids6020068>, URL <https://www.mdpi.com/2311-5521/6/2/68>.

- [5] 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, 1993, pp. 247–273. <https://doi.org/10.1006/jcph.1993.1216>, URL <http://linkinghub.elsevier.com/retrieve/pii/S0021999183712167>
- [6] Barba, L. A., “Vortex Method for computing high-Reynolds number Flows: Increased accuracy with a fully mesh-less formulation,” Ph.D. thesis, California Institute of Technology, 2004. <https://doi.org/10.7907/TSR5-DE67>
- [7] Yokota, R., and Barba, L. A., “FMM-based vortex method for simulation of isotropic turbulence on GPUs , compared with a spectral method,” *Computers and Fluids*, Vol. 80, 2013, pp. 17–27. <https://doi.org/10.1016/j.compfluid.2012.08.002>, URL <http://dx.doi.org/10.1016/j.compfluid.2012.08.002>
- [8] Barba, L. A., Leonard, A., and Allen, C. B., “Vortex method with meshless spatial adaption for accurate simulation of viscous, unsteady vortical flows,” *International Journal for Numerical Methods in Fluids*, Vol. 47, No. 8-9, 2005, pp. 841–848. <https://doi.org/10.1002/flid.842>
- [9] Lakkis, I., and Ghoniem, A., “A high resolution spatially adaptive vortex method for separating flows. Part I: Two-dimensional domains,” *Journal of Computational Physics*, Vol. 228, No. 2, 2009, pp. 491–515. <https://doi.org/10.1016/j.jcp.2008.09.025>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0021999108005093>
- [10] Stock, M. J., and Gharakhani, A., “Solution-Responsive Particle Size Adaptivity in Lagrangian Vortex Particle Methods,” *Volume 1: Aerospace Engineering Division Joint Track; Computational Fluid Dynamics*, American Society of Mechanical Engineers, 2021, pp. 1–11. <https://doi.org/10.1115/FEDSM2021-65621>, URL <https://asmedigitalcollection.asme.org/FEDSM/proceedings/FEDSM2021/85284/V001T02A033/1120902>
- [11] Yokota, R., Barba, L. A., Narumi, T., and Yasuoka, K., “Petascale turbulence simulation using a highly parallel fast multipole method on GPUs,” *Computer Physics Communications*, Vol. 184, No. 3, 2013, pp. 445–455. <https://doi.org/10.1016/j.cpc.2012.09.011>, URL <http://dx.doi.org/10.1016/j.cpc.2012.09.011>
- [12] Hu, 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, 2014, pp. 966–975. <https://doi.org/10.1109/IPDPSW.2014.110>, URL <http://ieeexplore.ieee.org/document/6969486/>
- [13] Kasliwal, A., Furbush, N. J., Gawron, J. H., McBride, J. R., Wallington, T. J., De Kleine, R. D., Kim, H. C., and Keoleian, G. A., “Role of flying cars in sustainable mobility,” *Nature Communications*, Vol. 10, No. 1, 2019, p. 1555. <https://doi.org/10.1038/s41467-019-09426-0>, URL <http://www.nature.com/articles/s41467-019-09426-0>
- [14] “Advancing Aerial Mobility: A National Blueprint,” Tech. rep., National Academies of Sciences, Engineering, and Medicine, Washington, D.C., 2020. <https://doi.org/10.17226/25646>, URL <https://www.nap.edu/catalog/25646>
- [15] Schwab, A., Thomas, A., Bennett, J., Robertson, E., Cary, S., Schwab, A., Thomas, A., Bennett, J., Robertson, E., and Cary, S., “Electrification of Aircraft : Challenges , Barriers , and Potential Impacts Electrification of Aircraft : Challenges , Barriers , and

- Potential Impacts,” Tech. Rep. October, National Renewable Energy Laboratory, 2021. <https://doi.org/NREL/TP-6A20-80220>, URL <https://www.nrel.gov/docs/fy22osti/80220.pdf>.
- [16] McDonald, R. A., German, B. J., Takahashi, T., Bil, C., Anemaat, W., Chaput, A., Vos, R., and Harrison, N., “Future aircraft concepts and design methods,” *The Aeronautical Journal*, , No. 801, 2021, pp. 1–33. <https://doi.org/10.1017/aer.2021.110>, URL https://www.cambridge.org/core/product/identifier/S000192402100110X/type/journal_article.
- [17] Silva, C., Johnson, W. R., Antcliff, K. R., and Patterson, M. D., “VTOL Urban Air Mobility Concept Vehicles for Technology Development,” *2018 Aviation Technology, Integration, and Operations Conference*, 2018, pp. 1–16. <https://doi.org/10.2514/6.2018-3847>, URL <https://doi.org/10.2514/6.2018-3847>.
- [18] 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*, Vol. 104, 2018, pp. 1–19. <https://doi.org/10.1016/j.paerosci.2018.06.004>, URL <https://doi.org/10.1016/j.paerosci.2018.06.004>.
- [19] Stock, M., Gharakhani, A., and Stone, C., “Modeling Rotor Wakes with a Hybrid OVERFLOW-Vortex Method on a GPU Cluster,” *28th AIAA Applied Aerodynamics Conference*, , No. July, 2010, pp. 1–12. <https://doi.org/10.2514/6.2010-4553>, URL <http://arc.aiaa.org/doi/abs/10.2514/6.2010-4553>.
- [20] Alvarez, E. J., and Ning, A., “High-Fidelity Modeling of Multirotor Aerodynamic Interactions for Aircraft Design,” *AIAA Journal*, Vol. 58, No. 10, 2020, pp. 4385–4400. <https://doi.org/10.2514/1.J059178>, URL <https://arc.aiaa.org/doi/10.2514/1.J059178>.
- [21] Alvarez, E. J., Schenk, A., Critchfield, T., and Ning, A., “Rotor-on-Rotor Aeroacoustic Interactions of Multirotor in Hover,” 2020.
- [22] Tan, J. F., Cai, J. G., Barakos, G. N., Wang, C., and Huang, M. Q., “Computational Study on the Aerodynamic Interference Between Tandem Rotors and Nearby Obstacles,” *Journal of Aircraft*, Vol. 57, No. 3, 2020, pp. 456–468. <https://doi.org/10.2514/1.C035629>, URL <https://arc.aiaa.org/doi/10.2514/1.C035629>.
- [23] Tan, J. F., er Gao, J., Barakos, G. N., Lin, C. L., Zhang, W. G., and Huang, M. Q., “Novel approach to helicopter brownout based on vortex and discrete element methods,” *Aerospace Science and Technology*, Vol. 116, 2021, p. 106839. <https://doi.org/10.1016/j.ast.2021.106839>, URL <https://doi.org/10.1016/j.ast.2021.106839https://linkinghub.elsevier.com/retrieve/pii/S1270963821003497>.
- [24] Yücekayalı, A., and Ortakaya, Y., “Viscous Vortex Particle Method Explored for Main Rotor-Tail Rotor Interaction,” *8th Asian/Australian Rotorcraft Forum*, 2019.
- [25] 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, 2018, pp. 1–22. <https://doi.org/10.2514/6.2018-3646>, URL <https://arc.aiaa.org/doi/10.2514/6.2018-3646>.

- [26] Teixeira, P. C., and Cesnik, C. E. S., "Propeller Effects on the Response of High-Altitude Long-Endurance Aircraft," *AIAA Journal*, Vol. 57, No. 10, 2019, pp. 4328–4342. <https://doi.org/10.2514/1.j057575>.
- [27] Singh, P., and Friedmann, P. P., "Aeromechanics and Aeroelastic Stability of Coaxial Rotors," *Journal of Aircraft*, 2021, pp. 1–20. <https://doi.org/10.2514/1.C036291> URL <https://arc.aiaa.org/doi/10.2514/1.C036291>
- [28] Jacobellis, G., Singh, R., Johnson, C., Sirohi, J., and McDonald, R., "Experimental and computational investigation of stacked rotor performance in hover," *Aerospace Science and Technology*, Vol. 1, 2021, p. 106847. <https://doi.org/10.1016/j.ast.2021.106847>, URL <https://doi.org/10.1016/j.ast.2021.106847https://linkinghub.elsevier.com/retrieve/pii/S1270963821003576>
- [29] Corle, E., Floros, M., and Schmitz, S., "On the Influence of Inflow Model Selection for Time-Domain Tiltrotor Aeroelastic Analysis," *Journal of the American Helicopter Society*, Vol. 66, No. 3, 2021, pp. 1–12. <https://doi.org/10.4050/JAHS.66.032009>, URL <https://www.ingentaconnect.com/content/10.4050/JAHS.66.032009>
- [30] 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, 2019. <https://doi.org/10.2514/6.2019-2827>, URL <https://arc.aiaa.org/doi/10.2514/6.2019-2827>.
- [31] Lee, H., and Lee, D. J., "Rotor interactional effects on aerodynamic and noise characteristics of a small multirotor unmanned aerial vehicle," *Physics of Fluids*, Vol. 32, No. 4, 2020. <https://doi.org/10.1063/5.0003992>, URL <https://doi.org/10.1063/5.0003992>
- [32] Lee, H., and Lee, D.-J., "Wake impact on aerodynamic characteristics of horizontal axis wind turbine under yawed flow conditions," *Renewable Energy*, Vol. 136, 2019, pp. 383–392. <https://doi.org/10.1016/j.renene.2018.12.126>, URL <https://doi.org/10.1016/j.renene.2018.12.126https://linkinghub.elsevier.com/retrieve/pii/S0960148118315829>
- [33] Lee, H., and Lee, D.-J., "Effects of platform motions on aerodynamic performance and unsteady wake evolution of a floating offshore wind turbine," *Renewable Energy*, Vol. 143, 2019, pp. 9–23. <https://doi.org/10.1016/j.renene.2019.04.134>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0960148119306184>
- [34] Mehr, J. A., Alvarez, E. J., and Ning, A., "Unsteady Aerodynamic Analysis of Wind Harvesting Aircraft," *AIAA AVIATION 2020 FORUM*, Vol. 1 PartF, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2020, pp. 1–13. <https://doi.org/10.2514/6.2020-2761>, URL <https://arc.aiaa.org/doi/10.2514/6.2020-2761>
- [35] Kuzmina, K. S., Marchevsky, I. K., and Ryatina, E. P., "The VM2D Open Source Code for Incompressible Flow Simulation by Using Meshless Lagrangian Vortex Methods on CPU and GPU," *2019 Ivannikov Memorial Workshop (IVMEM)*, IEEE, 2019, pp. 86–92. <https://doi.org/10.1109/IVMEM.2019.00020>, URL <https://ieeexplore.ieee.org/document/8880756/>
- [36] Stock, M. J., and Gharakhani, A., "Open-Source Accelerated Vortex Particle Methods for Unsteady Flow Simulation," *Proceedings of the ASME 2020 Fluids Engineering Division Summer Meeting*, 2020. <https://doi.org/10.1115/FEDSM2020-20486>

- [37] Tugnoli, M., Montagnani, D., Syal, M., Droandi, G., and Zanotti, A., “Mid-fidelity approach to aerodynamic simulations of unconventional VTOL aircraft configurations,” *Aerospace Science and Technology*, Vol. 115, 2021, p. 106804. <https://doi.org/10.1016/j.ast.2021.106804>, URL <https://doi.org/10.1016/j.ast.2021.106804https://linkinghub.elsevier.com/retrieve/pii/S127096382100314X>.
- [38] 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. <https://doi.org/10.1002/fld.811>.
- [39] Kirchhart, M., and Rieger, C., “Discrete Projections: A Step Towards Particle Methods on Bounded Domains without Remeshing,” *SIAM Journal on Scientific Computing*, Vol. 43, No. 1, 2021, pp. A609—A635. <https://doi.org/10.1137/19M1299864>, URL <https://epubs.siam.org/doi/10.1137/19M1299864>.
- [40] Pedrizzetti, G., “Insight into singular vortex flows,” *Fluid Dynamics Research*, Vol. 10, No. 2, 1992, pp. 101–115. [https://doi.org/10.1016/0169-5983\(92\)90011-K](https://doi.org/10.1016/0169-5983(92)90011-K).
- [41] Gharakhani, A., “A Survey of Grid-Free Methods for the Simulation of 3-D Incompressible Flows in Bounded Domains,” Tech. Rep. SAND97-2256, Sandia National Laboratories, 1997.
- [42] Gharakhani, A., “A Grid-Free Method for LES of Incompressible Flow,” *Volume 2: Symposia and General Papers, Parts A and B*, Vol. 257, ASMEDC, 2002, pp. 597–604. <https://doi.org/10.1115/FEDSM2002-31370>, URL <https://asmedigitalcollection.asme.org/FEDSM/proceedings/FEDSM2002/36169/597/296541>.
- [43] Gharakhani, A., “A Lagrangian Vortex Method for Grid-Free Dynamic LES,” *17th AIAA Computational Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2005, pp. 1–13. <https://doi.org/10.2514/6.2005-4625>, URL <https://arc.aiaa.org/doi/10.2514/6.2005-4625>.
- [44] Christiansen, J., “Numerical Simulation of Hydrodynamics by the Method of Point Vortices,” *Journal of Computational Physics*, Vol. 135, No. 2, 1997, pp. 189–197. <https://doi.org/10.1006/jcph.1997.5701>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0021999197957016>.
- [45] Cottet, G.-H., and Poncet, P., “Advances in direct numerical simulations of 3D wall-bounded flows by Vortex-in-Cell methods,” *Journal of Computational Physics*, Vol. 193, No. 1, 2004, pp. 136–158. <https://doi.org/10.1016/j.jcp.2003.08.025>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0021999103004248>.
- [46] Koumoutsakos, P., “Multiscale flow simulations using particles,” *Annual Review of Fluid Mechanics*, Vol. 37, No. 1, 2005, pp. 457–487. <https://doi.org/10.1146/annurev.fluid.37.061903.175753>.
- [47] Chatelain, P., Curioni, A., Bergdorf, M., Rossinelli, D., Andreoni, W., and Koumoutsakos, P., “Billion vortex particle direct numerical simulations of aircraft wakes,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 197, No. 13-16, 2008, pp. 1296–1304. <https://doi.org/10.1016/j.cma.2007.11.016>.

- [48] Chatelain, P., Duponcheel, M., Caprace, D.-G., Marichal, Y., and Winckelmans, G., “Vortex particle-mesh simulations of vertical axis wind turbine flows: from the airfoil performance to the very far wake,” *Wind Energy Science*, Vol. 2, No. 1, 2017, pp. 317–328. <https://doi.org/10.5194/wes-2-317-2017>, URL <http://stacks.iop.org/1742-6596/753/i=3/a=032007?key=crossreffc2b2fe9996c69d4a419caec4e075875https://www.wind-energ-sci.net/2/317/2017/>
- [49] Caprace, D.-G., Chatelain, P., and Winckelmans, G., “Wakes of rotorcraft in advancing flight: A large-eddy simulation study,” *Physics of Fluids*, Vol. 32, No. 8, 2020, p. 087107. <https://doi.org/10.1063/5.0015162>, URL <https://doi.org/10.1063/5.0015162http://aip.scitation.org/doi/10.1063/5.0015162>
- [50] Nguyen, V. L., “Deformation of a vortex ring caused by its impingement on a sphere,” *Physics of Fluids*, Vol. 31, No. 10, 2019, p. 107108. <https://doi.org/10.1063/1.5122260>, URL <https://doi.org/10.1063/1.5122260http://aip.scitation.org/doi/10.1063/1.5122260>
- [51] Nguyen, V. L., and Duong, V. D., “Vortex ring-tube reconnection in a viscous fluid,” *Physics of Fluids*, Vol. 33, No. 1, 2021. <https://doi.org/10.1063/5.0037930>, URL <https://doi.org/10.1063/5.0037930>
- [52] Colognesi, V., Ronsse, R., and Chatelain, P., “Model coupling biomechanics and fluid dynamics for the simulation of controlled flapping flight,” *Bioinspiration and Biomimetics*, Vol. 16, No. 2, 2021. <https://doi.org/10.1088/1748-3190/abdd9c>
- [53] Ramos-García, N., Kontos, S., Pegalajar-Jurado, A., Horcas, S. G., and Bredmose, H., “Investigation of the floating IEA Wind 15 MW RWT using vortex methods Part II: Wake impact on downstream turbines under turbulent inflow. In review,” *Wind Energy*, , No. July, 2021, pp. 1–37. <https://doi.org/10.1002/we.2682>
- [54] Chatwin, P. C., “The vorticity equation as an angular momentum equation,” *Mathematical Proceedings of the Cambridge Philosophical Society*, Vol. 74, No. 2, 1973, pp. 365–367. <https://doi.org/10.1017/S0305004100048131>, URL https://www.cambridge.org/core/product/identifier/S0305004100048131/type/journal_article
- [55] Shankar, S., and Dommelen, L., “A New Diffusion Procedure for Vortex Methods,” *Journal of Computational Physics*, Vol. 127, No. 1, 1996, pp. 88–109. <https://doi.org/10.1006/jcph.1996.0160>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0021999196901606>
- [56] Gharakhani, A., “Grid-free simulation of 3-D vorticity diffusion by a high-order vorticity redistribution method,” *15th AIAA Computational Fluid Dynamics Conference*, , No. June 2001, 2001. <https://doi.org/doi:10.2514/6.2001-2640>, URL <http://dx.doi.org/10.2514/6.2001-2640>
- [57] Degond, P., and Mas-Gallic, S., “The Weighted Particle Method for Convection-Diffusion Equations. Part 1: The Case of an Isotropic Viscosity,” *Mathematics of Computation*, Vol. 53, No. 188, 1989, p. 485. <https://doi.org/10.2307/2008716>, URL <http://www.jstor.org/stable/2008717?origin=crossrefhttp://www.jstor.org/stable/2008716?origin=crossref>
- [58] 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, 1996, pp. 370–397. <https://doi.org/10.1137/S1064827593254397>, URL <http://epubs.siam.org/doi/10.1137/S1064827593254397>

- [59] 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. <https://doi.org/10.1016/j.jcp.2009.03.007>, URL <http://dx.doi.org/10.1016/j.jcp.2009.03.007>.
- [60] Beale, J. T., and Majda, A., “Vortex Methods. II: Higher Order Accuracy in Two and Three Dimensions,” *Mathematics of Computation*, Vol. 39, No. 159, 1982, p. 29. <https://doi.org/10.2307/2007618>.
- [61] Anderson, C., and Greengard, C., “On Vortex Methods,” *SIAM Journal on Numerical Analysis*, Vol. 22, No. 3, 1985, pp. 413–440. URL <http://www.jstor.org/stable/2157074>.
- [62] Raviart, P. A., “An analysis of particle methods,” *Numerical Methods in Fluid Dynamics*, edited by F. Brezzi, Springer Berlin Heidelberg, Berlin, Heidelberg, 1985, pp. 243–324.
- [63] Beale, J. T., “A convergent 3-D vortex method with grid-free stretching,” *Mathematics of Computation*, Vol. 46, No. 174, 1986, pp. 401–424. <https://doi.org/10.1090/S0025-5718-1986-0829616-6>, URL <https://www.ams.org/mcom/1986-46-174/S0025-5718-1986-0829616-6/>.
- [64] Ghosal, S., and Moin, P., “The Basic Equations for the Large Eddy Simulation of Turbulent Flows in Complex Geometry,” *Journal of Computational Physics*, Vol. 118, No. 1, 1995, pp. 24–37. <https://doi.org/10.1006/jcph.1995.1077>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0021999185710777>.
- [65] Winckelmans, G. S., “Topics in vortex methods for the computation of three and two dimensional incompressible unsteady flows,” Ph.D. thesis, California Institute of Technology, 1989. <https://doi.org/10.7907/19HD-DF80>, URL <http://core.kmi.open.ac.uk/download/pdf/11813390.pdf>.
- [66] Williamson, J. H., “Low-storage Runge-Kutta schemes,” *Journal of Computational Physics*, Vol. 35, No. 1, 1980, pp. 48–56. [https://doi.org/10.1016/0021-9991\(80\)90033-9](https://doi.org/10.1016/0021-9991(80)90033-9).
- [67] Greengard, L. F., “The Rapid Evaluation Of Potential Fields In Particle Systems,” Ph.D. thesis, Yale University, 1987.
- [68] 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. <https://doi.org/10.1006/jcph.1999.6355>, URL <http://www.sciencedirect.com/science/article/pii/S0021999199963556>, <http://linkinghub.elsevier.com/retrieve/pii/S0021999199963556>.
- [69] 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. <https://doi.org/10.1016/B978-0-12-384988-5.00009-7>.
- [70] Wang, T., Yokota, R., and Barba, L., “ExaFMM: a high-performance fast multipole method library with C++ and Python interfaces,” *Journal of Open Source Software*, Vol. 6, No. 61, 2021, p. 3145. <https://doi.org/10.21105/joss.03145>, URL <https://joss.theoj.org/papers/10.21105/joss.03145>.
- [71] Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B., “Julia: A Fresh Approach to Numerical Computing,” *SIAM Review*, Vol. 59, No. 1, 2017, pp. 65–98. <https://doi.org/10.1137/141000671>, URL <https://epubs.siam.org/doi/10.1137/141000671>.

- [72] Alvarez, E. J., “Reformulated Vortex Particle Method and Meshless Large Eddy Simulation of Multirotor Aircraft,” Ph.D. thesis, Brigham Young University, 2022. URL <https://scholarsarchive.byu.edu/etd/9589>
- [73] Mansfield, J. R., Knio, O. M., and Meneveau, C., “A Dynamic LES Scheme for the Vorticity Transport Equation: Formulation and a Priori Tests,” *Journal of Computational Physics*, Vol. 145, No. 2, 1998, pp. 693–730. <https://doi.org/10.1006/jcph.1998.6051>
- [74] Mansfield, J. R., Knio, O. M., and Meneveau, C., “Dynamic LES of Colliding Vortex Rings Using a 3D Vortex Method,” *Journal of Computational Physics*, Vol. 152, No. 1, 1999, pp. 305–345. <https://doi.org/10.1006/jcph.1999.6258>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0021999199962587>
- [75] Cottet, G.-H., “Artificial Viscosity Models for Vortex and Particle Methods,” *Journal of Computational Physics*, Vol. 127, No. 2, 1996, pp. 299–308. <https://doi.org/10.1006/jcph.1996.0176>, URL <https://linkinghub.elsevier.com/retrieve/pii/S002199919690176X>
- [76] Cottet, G.-H., and Koumoutsakos, P. D., *Vortex Methods Theory and Practice*, Cambridge University Press, 2000. <https://doi.org/10.1017/CBO9780511526442>, URL <http://stacks.iop.org/0957-0233/12/i=3/a=704?key=crossref337f3af0cd1c6d71f7bf7e2f021fc491https://www.cambridge.org/core/product/identifier/9780511526442/type/book>
- [77] Meneveau, C., and Katz, J., “Scale-Invariance and Turbulence Models for Large-Eddy Simulation,” *Annual Review of Fluid Mechanics*, Vol. 32, No. 1, 2000, pp. 1–32. <https://doi.org/10.1146/annurev.fluid.32.1.1>, URL <https://www.annualreviews.org/doi/10.1146/annurev.fluid.32.1.1>
- [78] Leonard, A., “Vortex methods for flow simulation,” *Journal of Computational Physics*, Vol. 37, No. 3, 1980, pp. 289–335. [https://doi.org/10.1016/0021-9991\(80\)90040-6](https://doi.org/10.1016/0021-9991(80)90040-6), URL <https://linkinghub.elsevier.com/retrieve/pii/0021999180900406>
- [79] Nakanishi, Y., and Kamemoto, K., “Numerical Simulation of Flow around a Sphere with Vortex Blobs,” *Computational Wind Engineering I*, Vol. 47, Elsevier, 1993, pp. 363–369. <https://doi.org/10.1016/B978-0-444-81688-7.50040-1>, URL <http://dx.doi.org/10.1016/B978-0-444-81688-7.50040-1https://linkinghub.elsevier.com/retrieve/pii/B9780444816887500401>
- [80] Ojima, A., and Kamemoto, K., “Numerical Simulation of Unsteady Flow around Three Dimensional Bluff Bodies by an Advanced Vortex Method,” *JSME International Journal Series B*, Vol. 43, No. 2, 2000, pp. 127–135. <https://doi.org/10.1299/jsmeb.43.127>, URL http://www.jstage.jst.go.jp/article/jsmeb1993/43/2/43_{_}2_{_}127/{_}article
- [81] Kornev, N., Denev, J., and Samarbakhsh, S., “Theoretical Background of the Hybrid VpiLES Method for Flows with Variable Transport Properties,” *Fluids*, Vol. 5, No. 2, 2020, p. 45. <https://doi.org/10.3390/fluids5020045>, URL <https://www.mdpi.com/2311-5521/5/2/45>
- [82] Davidson, P. A., *An Introduction to Magnetohydrodynamics*, Cambridge University Press, 2001. <https://doi.org/10.1017/CBO9780511626333>, URL <https://www.cambridge.org/core/product/identifier/9780511626333/type/book>
- [83] Davidson, P. A., *An Introduction to Electrodynamics*, Oxford University Press, 2019.

- [84] Batchelor, G. K., *An Introduction to Fluid Dynamics*, Cambridge University Press, 1967. <https://doi.org/10.1017/CBO9780511800955>, URL <https://www.cambridge.org/core/product/identifier/9780511800955/type/book>.
- [85] Winckelmans, G. S., "Some Progress in Large-Eddy Simulation Using the 3D Vortex Particle Method," *CTR Annual Research Briefs*, 1995, pp. 391–415.
- [86] Mansfield, J. R., Knio, O. M., and Meneveau, C., "Towards lagrangian large vortex simulation," *ESAIM: Proceedings*, Vol. 1, 1996, pp. 49–64. <https://doi.org/10.1051/proc:1996019>, URL <http://www.esaim-proc.org/10.1051/proc:1996019>.
- [87] Caprace, D.-G., Winckelmans, G., and Chatelain, P., "An Immersed Lifting and Dragging Line Model for the Vortex Particle-Mesh Method," *Theoretical and Computational Fluid Dynamics*, 2020. <https://doi.org/10.1007/s00162-019-00510-1>, URL <https://doi.org/10.1007/s00162-019-00510-1> <http://link.springer.com/10.1007/s00162-019-00510-1>
- [88] Caprace, D.-G., Winckelmans, G., and Chatelain, P., "Assessment of the Vortex Particle-Mesh Method for Efficient LES of Hovering Rotors and their Wakes," *AIAA Scitech 2021 Forum*, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2021, pp. 1–17. <https://doi.org/10.2514/6.2021-0738>, URL <https://arc.aiaa.org/doi/10.2514/6.2021-0738>
- [89] Hughes, T. J., Mazzei, L., and Jansen, K. E., "Large Eddy Simulation and the variational multiscale method," *Computing and Visualization in Science*, Vol. 3, No. 1-2, 2000, pp. 47–59. <https://doi.org/10.1007/s007910050051>, URL <http://link.springer.com/10.1007/s007910050051>
- [90] Vreman, A. W., "The filtering analog of the variational multiscale method in large-eddy simulation," *Physics of Fluids*, Vol. 15, No. 8, 2003, pp. L61–L64. <https://doi.org/10.1063/1.1595102>, URL <http://aip.scitation.org/doi/10.1063/1.1595102>
- [91] Jeanmart, H., and Winckelmans, G., "Investigation of eddy-viscosity models modified using discrete filters: A simplified "regularized variational multiscale model" and an "enhanced field model"," *Physics of Fluids*, Vol. 19, No. 5, 2007. <https://doi.org/10.1063/1.2728935>
- [92] Cocle, R., Bricteux, L., and Winckelmans, G., "Scale dependence and asymptotic very high Reynolds number spectral behavior of multiscale subgrid models," *Physics of Fluids*, Vol. 21, No. 8, 2009, p. 085101. <https://doi.org/10.1063/1.3194302>, URL <http://aip.scitation.org/doi/10.1063/1.3194302>
- [93] Cottet, G.-H., "Anisotropic sub-grid scale numerical schemes for Large Eddy Simulations of turbulent flows," Tech. rep., 1998. <https://doi.org/10.13140/RG.2.1.1901.8323>
- [94] Cottet, G.-H., Jiroveanu, D., and Michaux, B., "Vorticity dynamics and turbulence models for Large-Eddy Simulations," *ESAIM: Mathematical Modelling and Numerical Analysis*, Vol. 37, No. 1, 2003, pp. 187–207. <https://doi.org/10.1051/m2an:2003013>, URL <http://www.esaim-m2an.org/10.1051/m2an:2003013>
- [95] Mimeau, C., Cottet, G.-H., and Mortazavi, I., "A Hybrid Vortex Method for the Simulation of 3D Incompressible Flows," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 11539 LNCS, 2019, pp. 611–622. https://doi.org/10.1007/978-3-030-22747-0_46, URL http://link.springer.com/10.1007/978-3-030-22747-0_{_}46

- [96] Mimeau, C., Cottet, G.-H., and Mortazavi, I., "An Artificial Viscosity Model for 3D Simulations with Vortex Methods," *10th International Conference on Computational Fluid Dynamics, ICCFD 2018 - Proceedings*, 2018, pp. 1–6.
- [97] Germano, M., Piomelli, U., Moin, P., and Cabot, W. H., "A dynamic subgrid-scale eddy viscosity model," *Physics of Fluids A*, Vol. 3, No. 7, 1991, pp. 1760–1765. <https://doi.org/10.1063/1.857955>.
- [98] Germano, M., "Turbulence: the filtering approach," *Journal of Fluid Mechanics*, Vol. 238, 1992, pp. 325–336. <https://doi.org/10.1017/S0022112092001733> URL https://www.cambridge.org/core/product/identifier/S0022112092001733/type/journal_article
- [99] Meneveau, C., Lund, T. S., and Cabot, W. H., "A Lagrangian dynamic subgrid-scale model of turbulence," *Journal of Fluid Mechanics*, Vol. 319, No. -1, 1996, p. 353. <https://doi.org/10.1017/S0022112096007379>, URL http://www.journals.cambridge.org/abstract/_/S0022112096007379.
- [100] Quinn, W. R., and Militzer, J., "Effects of nonparallel exit flow on round turbulent free jets," *International Journal of Heat and Fluid Flow*, Vol. 10, No. 2, 1989, pp. 139–145. [https://doi.org/10.1016/0142-727X\(89\)90008-8](https://doi.org/10.1016/0142-727X(89)90008-8)
- [101] Ball, C. G., Fellouah, H., and Pollard, A., "The flow field in turbulent round free jets," *Progress in Aerospace Sciences*, Vol. 50, 2012, pp. 1–26. <https://doi.org/10.1016/j.paerosci.2011.10.002> URL <http://dx.doi.org/10.1016/j.paerosci.2011.10.002>
- [102] Fellouah, H., and Pollard, A., "The velocity spectra and turbulence length scale distributions in the near to intermediate regions of a round free turbulent jet," *Physics of Fluids*, Vol. 21, No. 11, 2009, pp. 1–9. <https://doi.org/10.1063/1.3258837>
- [103] Mi, J., Nobes, D. S., and Nathan, G. J., "Influence of jet exit conditions on the passive scalar field of an axisymmetric free jet," *Journal of Fluid Mechanics*, Vol. 432, 2001, pp. 91–125. <https://doi.org/10.1017/S0022112000003384>
- [104] Iqbal, M. O., and Thomas, F. O., *Coherent structure in a turbulent jet via a vector implementation of the proper orthogonal decomposition*, Vol. 571, 2007. <https://doi.org/10.1017/S0022112006003351>
- [105] Romano, G., "The effect of boundary conditions by the side of the nozzle of a low Reynolds number jet," *Experiments in Fluids*, Vol. 33, No. 2, 2002, pp. 323–333. <https://doi.org/10.1007/s00348-002-0439-2>
- [106] Xu, G., and Antonia, R. A., "Effect of different initial conditions on a turbulent round free jet," *Experiments in Fluids*, Vol. 33, No. 5, 2002, pp. 677–683. <https://doi.org/10.1007/s00348-002-0523-7>.
- [107] Zawodny, N. S., and Boyd, Jr., D. D., "Acoustic Characterization and Prediction of Representative, Small-scale Rotary-wing Unmanned Aircraft System Components," *72nd American Helicopter Society (AHS) Annual Forum*, 2016.
- [108] Schenk, A. R., "Computational Investigation of the Effects of Rotor-on-Rotor Interactions on Thrust and Noise," Master's thesis, Brigham Young University, 2020.
- [109] Churchfield, M. J., Schreck, S. J., Martinez, L. A., Meneveau, C., and Spalart, P. R., "An Advanced Actuator Line Method for Wind Energy Applications and Beyond," *35th Wind Energy Symposium*, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2017. <https://doi.org/10.2514/6.2017-1998> URL <https://arc.aiaa.org/doi/10.2514/6.2017-1998>.

- [110] Delorme, Y., Stanly, R., Frankel, S. H., and Greenblatt, D., “Application of Actuator Line Model for Large Eddy Simulation of Rotor Noise Control,” *Aerospace Science and Technology*, Vol. 108, 2021, p. 106405. <https://doi.org/10.1016/j.ast.2020.106405>, URL <https://linkinghub.elsevier.com/retrieve/pii/S1270963820310877>.
- [111] Ning, Z., “Experimental investigations on the aerodynamic and aeroacoustic characteristics of small UAS propellers,” Ph.D. thesis, Iowa State University, Digital Repository, Ames, 2018. <https://doi.org/10.31274/etd-180810-6057>, URL <http://arc.aiaa.org/doi/10.2514/6.2016-1785https://lib.dr.iastate.edu/etd/16427/>.
- [112] Sinnige, T., de Vries, R., Corte, B. D., Avallone, F., Ragni, D., Eitelberg, G., and Veldhuis, L. L. M., “Unsteady Pylon Loading Caused by Propeller-Slipstream Impingement for Tip-Mounted Propellers,” *Journal of Aircraft*, Vol. 55, No. 4, 2018, pp. 1605–1618. <https://doi.org/10.2514/1.C034696>, URL <https://arc.aiaa.org/doi/10.2514/1.C034696>.
- [113] Stokkermans, T. C. A., van Arnhem, N., Sinnige, T., and Veldhuis, L. L. M., “Validation and Comparison of RANS Propeller Modeling Methods for Tip-Mounted Applications,” *AIAA Journal*, Vol. 57, No. 2, 2019, pp. 566–580. <https://doi.org/10.2514/1.J057398>, URL <https://arc.aiaa.org/doi/10.2514/1.J057398>.
- [114] Parmentier, P., Winckelmans, G., and Chatelain, P., “A Vortex Particle-Mesh method for subsonic compressible flows,” *Journal of Computational Physics*, Vol. 354, 2018, pp. 692–716. <https://doi.org/10.1016/j.jcp.2017.10.040>, URL <https://doi.org/10.1016%2Fj.jcp.2017.10.040>.
- [115] Eldredge, J. D., Colonius, T., and Leonard, A., “A Vortex Particle Method for Two-Dimensional Compressible Flow,” *Journal of Computational Physics*, Vol. 179, No. 2, 2002, pp. 371–399. <https://doi.org/10.1006/jcph.2002.7060>, URL <https://doi.org/10.1006%2Fjcph.2002.7060>.
- [116] Ahrens, J., Geveci, B., and Law, C., “ParaView: An End-User Tool for Large Data Visualization,” *Visualization Handbook*, Elsevier, 2005.
- [117] Ayachit, U., *The ParaView Guide: A Parallel Visualization Application*, Kitware, 2015.