VALIDATION OF A NEW SOLVER BASED ON THE VORTEX PARTICLE METHOD FOR WINGS, PROPELLERS AND ROTORS

Johan Valentin* and Luis Bernardos†
ONERA, 92190 Meudon, France

Abstract

The Vortex Particle Method (VPM) represents continuous, incompressible, unsteady, viscous flows through the use of singular, independent particles represented by small concentration of vorticity. This method is convenient for simulating vortex reconnections in the wake as well as flow mixing. This study aims to provide a mid-fidelity aerodynamic solver based on the VPM. Coupled with a Lifting Line (LL), the VPM-LL solver predicts an accurate flow solution of a lifting body in an unbounded freestream, with a significantly reduced computational cost compared to classical CFD solvers. After introducing the VPM, its implementation is detailed. The coupling with the LL is explained and validated on two wings, an axial propeller and a drone rotor, through the comparison of CFD, experimental and analytical results. A quadrotor drone from the GARTEUR Action Group 25 is studied and compared to the experimental data from DLR and numerical results from DLR and ONERA.

Nomenclature

$a(t)$	=	Vortex Section Radius (m)	$\overrightarrow{U}$	=	Total Velocity (m.s^{-1})
b	=	Wingspan (m)	$\overrightarrow{U}_\infty$	=	Freestream Velocity (m.s^{-1})
C_f, C_p	=	Friction/Pressure Coefficient	$\overrightarrow{u}_d$	=	Diffusion Velocity (m.s^{-1})
C_L, C_D	=	Lift/Drag Coefficient	$\overrightarrow{u}_\phi$	=	Solid Induced Velocity (m.s^{-1})
C_T, C_P	=	Thrust/Power Coefficient	$\overrightarrow{u}_\psi = (u, v, w)$	=	Fluid Induced Velocity (m.s^{-1})
C_s	=	Smagorinski Constant	V	=	Particle Volume (m^3)
c	=	Chord (m)	$\overrightarrow{x}$	=	Position (m)
d	=	Rotor Spacing (m)	$\overrightarrow{\alpha}$	=	Vortex Strength (m^3.s^{-1})
$\mathscr{E}_c$	=	Kinetic Energy (m^5.s^{-2})	$\alpha, \alpha_e, \alpha_i$	=	Pitch/Effective/Induced Angle (°)
$\overrightarrow{E}_{\text{trunc}}$	=	Truncature Error (s^{-2})	Γ	=	Circulation (m^2.s^{-1})
$g()$	=	Velocity Smoothing Function	γ	=	Sheet Intensity (m.s^{-1})
h	=	Resolution Scale (m)	$\delta()$	=	Dirac Delta Function (m^3)
J	=	Advance Ratio	$\mathcal{E}$	=	Divergence-Free Enstrophy (m^3.s^{-2})
k_{relax}	=	Circulation Relaxation Factor	$\mathcal{E}_f$	=	Enstrophy (m^3.s^{-2})
$\overrightarrow{L}, \overrightarrow{D}$	=	Lift/Drag (N)	$\zeta_\sigma()$	=	Vorticity Smoothing Function (m^3)
M	=	Mach Number	$\eta = U_\infty T/P$	=	Propeller Efficiency
N_p	=	Number of Particles	$\lambda_\sigma = \sigma/h$	=	Smoothing Ratio
p	=	Pressure (P$_a$)	ν, ν_t	=	Kinematic/Turbulent Viscosity (m^2.s^{-1})
R	=	Vortex Ring Radius (m)	ρ_∞	=	Fluid Density (kg.m^{-3})
R_e	=	Reynolds Number	$\rho = \|\overrightarrow{r}\|/\sigma$	=	Dimensionless Distance
$\overrightarrow{r_{ij}} = \overrightarrow{x_i} - \overrightarrow{x_j}$	=	Inter-particular Distance (m)	σ	=	Particle Core Radius (m)
s	=	Curvilinear Abscissa	$\overrightarrow{\omega}$	=	Vorticity (s^{-1})
t	=	Time (s)	ω_0	=	Vorticity Cutoff (s^{-1})

*P.h.D Student, ONERA, DAAA/H2T, johan.valentin@onera.fr
†P.h.D, ONERA, DAAA/H2T, luis.bernardos@onera.fr

1. INTRODUCTION

THE aerodynamic interactions between several rotors are arduous to accurately simulate with low fidelity methods and are also too costly to be studied with high fidelity methods such as URANS. This is especially verified during the design phase when accounting for the interactions between multiple rotors and capture small and complex vortex structures requiring refined overset meshes, as in the case of eVTOLs.

Potential methods (panels, filaments, actuator disk ..) are less costly but have trouble capturing small and intricate vortex structures, viscous interactions and/or wake mixing. The Vortex Particle Method (VPM), first introduced by Rosenhead[1], aims to represent continuous, incompressible and unsteady flows through the use of singular particles represented by small concentration of vorticity. This method is especially convenient for simulating flow mixing[2, 3] because of the independent and free motion of the particles, where prescribed wake methods fail to do so. This method also accounts for viscous effects in the flow[3–6], allowing vortex reconnections between the particles, where filaments methods can not.

Various authors managed to use the VPM to simulate lifting bodies. In that respect, propellers and other marine turbines were simulated using actuator disks[7], panel methods[8, 9] and Lifting-Lines for wings[10, 11]. The applications of the VPM covers a large variety of cases from propeller-on-wing interactions[9], rotors in hover configuration[12] to rotor wake transition and coplanar rotors wake mixing[2]. Work was also done on the hybridisation of the VPM with high-fidelity methods[13, 14] to provide a hybrid Eulerian-Lagrangian solver. The wide range of application of the VPM makes it all the more interesting to use with other solvers. Moreover, it is also a meshless method which significantly reduces computational dissipation inherent to Eulerian methods, which prevents them from accurately predicting way decay.

The objective of this work is to develop and validate a fast method for capturing the aerodynamic loads and wake decay of rotary and fixed wings. The VPM was coupled with a Lifting-Line (LL) to simulate lifting bodies in an unbounded freestream. To assess the method, several test cases are considered here, with increasing complexity level. They range from simple viscous vortex rings to fixed wings and propellers in interaction. The results are compared to CFD, experimental and analytical results.

After introducing the implementation of the VPM in section 2, its coupling with a Lifting-Line is detailed in section 3. The viscous treatment of the VPM solver is then validated with unbounded analytical vortices in section 4. The validation of the coupled solver follows in section 5. The simulation of fixed wings with variable twist and sweep angles is first examined. The loads and wake generation of an elliptical wing is then investigated. A propeller is next studied by comparing its loads and wake with a high-fidelity solver. Eventually a quadcopter drone is simulated and compared with experimental data and a high and a middle fidelity numerical solvers.

2. VISCOUS VORTEX PARTICLE METHOD

2.1. LES Filtering of the VPM

The Vortex Particle Method (VPM) aims to represent an unbounded, incompressible flow with the use of independent particles convected with the freestream and their own induced velocity from a Lagrangian point of view. Each particle is a vector element represented by a position $(\overrightarrow{x_i})$ and a strength $(\overrightarrow{\alpha_i})$ governed by the vorticity equation, obtained by taking the curl of the 3D incompressible Navier-Stokes equations:

$$\frac{\mathrm{d}\overrightarrow{\omega}}{\mathrm{d}t} = \frac{\partial \overrightarrow{\omega}}{\partial t} + \left(\overrightarrow{U} \cdot \nabla\right)\overrightarrow{\omega} = \left(\overrightarrow{\omega} \cdot \nabla\right)\overrightarrow{U} + \nu\Delta\overrightarrow{\omega} \tag{1}$$

where $\overrightarrow{\omega} = \overrightarrow{\nabla} \times \overrightarrow{U}$ is the vorticity of the flow. The material derivatives involved in this equation is well suited for a Lagrangian interpretation of the flow. The two terms on the right-hand-side respectively accounts for the stretching/compressing of the vortices and the viscous effects in the flow. The vorticity is estimated through the delta Dirac function by a distribution of N_p vortices:

$$\overrightarrow{\omega}(\overrightarrow{x}, t) \approx \sum_{j}^{N_p} \delta(\overrightarrow{x} - \overrightarrow{x_j})\overrightarrow{\alpha_j}(t) \tag{2}$$

To avoid inducing singularities in the flow, the delta Dirac function is approximated by the $\zeta_\sigma()$ function which filters the flow to the size of the particles noted σ. A Gaussian distribution filter is used and its density of probability is:

$$\zeta_{\sigma_j}(r = ||\overrightarrow{x} - \overrightarrow{x_j}||) = \frac{1}{\pi^{3/2}} \frac{1}{\left(2\sigma_j^2\right)^{3/2}} e^{-r^2/(2\sigma^2)} \tag{3}$$

Filtered quantities are noted with $\tilde{\cdot}$. Once filtered to the size of the particles, the vorticity equation becomes:

$$\frac{\mathrm{d}\widetilde{\overrightarrow{\omega}}}{\mathrm{d}t} = \left(\widetilde{\overrightarrow{\omega}} \cdot \nabla\right)\widetilde{\overrightarrow{U}} + \nu\Delta\widetilde{\overrightarrow{\omega}} - \overrightarrow{E}_{\text{trunc}} \tag{4}$$

where $\overrightarrow{E}_{\text{trunc}}$ encapsulates the error between resolved and unresolved quantities because of the filtering of the small scales of the flow. A turbulent viscosity model can be used to approximate this truncature error which represents the loss of enstrophy transfered to the small scales:

$$\overrightarrow{E}_{\text{trunc}} = -\overrightarrow{\nabla} \cdot \left(\nu_t \left(\overrightarrow{\nabla} \otimes \widetilde{\overrightarrow{\omega}}\right) + {}^t \left(\overrightarrow{\nabla} \otimes \widetilde{\overrightarrow{\omega}}\right)\right) \tag{5}$$

The second term in (5) is neglected in front of the first one[15]. The resulting expression can be merged with the viscous term of the filtered vorticity equation:

$$\frac{\mathrm{d}\widetilde{\overrightarrow{\omega}}}{\mathrm{d}t} = \left(\widetilde{\overrightarrow{\omega}} \cdot \nabla\right) \widetilde{\overrightarrow{U}} + \overrightarrow{\nabla} \cdot \left((\nu + \nu_t) \left(\overrightarrow{\nabla} \otimes \widetilde{\overrightarrow{\omega}}\right)\right) \tag{6}$$

Rather than computing the transpose term of the filtered vorticity equation as is, its transposed form is considered. Indeed one has:

$$\overrightarrow{\omega} \cdot \left(\overrightarrow{\nabla} \otimes \overrightarrow{U}\right) = \overrightarrow{\omega} \cdot \left({}^t\overrightarrow{\nabla} \otimes \overrightarrow{U}\right) \tag{7}$$

This form is more stable and allows for the total conservation of vorticity in the flow[6].

The vorticity is not transported by the particles *per se*. They only induce the vorticity while carrying their own strength. The vorticity equation has to be transformed to determine the evolution of the strength of the particles. To do so, the vorticity is integrated on the volume V_i of each particle:

$$\alpha_i(t) = \int_{V_i} \widetilde{\overrightarrow{\omega}}(\overrightarrow{x}, t) dV \approx V_i \overrightarrow{\omega_i}(t) \tag{8}$$

Integrating the vorticity equation then gives:

$$\frac{\mathrm{d}\overrightarrow{\alpha_i}}{\mathrm{d}t} = \left(\overrightarrow{\alpha_i} \cdot {}^t\nabla\right) \overrightarrow{U_i} + V_i \overrightarrow{\nabla} \cdot \left(\left(\nu + \nu_{t_i}\right) \left(\overrightarrow{\nabla} \otimes \overrightarrow{\omega_i}\right)\right) \tag{9}$$

The vorticity induced by the particles becomes:

$$\widetilde{\overrightarrow{\omega}}(\overrightarrow{x_i}, t) \approx \overrightarrow{\omega_i}(t) \approx \sum_j^{N_p} \zeta_{\sigma_j}(r_{ij}) \overrightarrow{\alpha_j}(t) \tag{10}$$

The velocity is obtained thanks to the Helmholtz Decomposition Theorem:

$$\overrightarrow{U} = \overrightarrow{U}_\infty + \overrightarrow{u}_\phi + \overrightarrow{u}_\psi$$

where $\overrightarrow{u}_\phi$ and $\overrightarrow{u}_\psi$ are respectively the velocity induced by lifting surfaces and the velocity induced by the flow. The latter is obtained from the Biot-Savart law and is the velocity that is induced by the particles. Similarly to the filtering of the vorticity, the singular Biot-Savart kernel is regularised by the smoothing function $g()$. This function can be obtained directly from the $\zeta_\sigma()$ filter used previously:

$$g(\rho = r/\sigma) = \int_0^r t^2 \zeta_\sigma(t) dt \tag{11}$$

$$= \frac{1}{4\pi} \left(\text{erf}\left(\frac{\rho}{\sqrt{2}}\right) - \frac{2}{\sqrt{\pi}} \frac{\rho}{\sqrt{2}} e^{-\rho^2/2}\right) \tag{12}$$

The velocity induced by the particles becomes:

$$\widetilde{\overrightarrow{u}}(\overrightarrow{x_i}, t) \approx \overrightarrow{u}_{\psi_i} = -\sum_j^{N_p} \frac{g(\rho_{ij})}{r_{ij}^3} \overrightarrow{r_{ij}} \times \overrightarrow{\alpha_j} \tag{13}$$

The particles being Lagrangian markers, they are transported in the flow according to the freestream velocity, the velocity induced by solid surfaces and the velocity they induce upon themselves:

$$\frac{\mathrm{d}\overrightarrow{x_i}}{\mathrm{d}t} = \widetilde{\overrightarrow{U}}(\overrightarrow{x_i}, t) \approx \overrightarrow{U_i} = \overrightarrow{U}_\infty + \overrightarrow{u}_{\phi_i} + \overrightarrow{u}_{\psi_i} \tag{14}$$

The gradient of the particle induced velocity can be expressed from the $g()$ and $\zeta_\sigma()$ smoothing functions:

$$\overrightarrow{\nabla} \otimes \overrightarrow{u_i} = -\sum_j \left(\frac{g(\rho_{ij})}{r^3} \overrightarrow{\nabla} \otimes \left(\overrightarrow{r_{ij}} \times \overrightarrow{\alpha_j}\right) + \frac{1}{r_{ij}^2} \left(\zeta_{\sigma_j}(r_{ij}) - \frac{3g(\rho_{ij})}{r^3}\right) \overrightarrow{r_{ij}} \otimes \left(\overrightarrow{r_{ij}} \times \overrightarrow{\alpha_j}\right)\right) \tag{15}$$

The stretching term of the vorticity equation can be directly computed with the expression of the velocity gradients in (15).

2.2. Diffusion Modeling

Unlike the stretching term, the diffusion term of the vorticity equation has to be approximated. A variety of methods are available for that purpose. Three of them were implemented, namely the Particle Strength Exchange (PSE), the Core Spreading Method (CSM) and the Diffusion Velocity Method (DVM).

2.2.1. Particle Strength Exchange (PSE)

The PSE[5, 6, 8, 16, 17] aims to directly approximate the Laplacian operator used for the diffusion term with a quadrature rule:

$$V_i \overrightarrow{\nabla} \cdot \left(\nu_i \overrightarrow{\nabla} \otimes \overrightarrow{\omega_i}\right) \approx V_i \int_V \left(\nu(\overrightarrow{x_i}) + \nu(\overrightarrow{y})\right) \frac{\left(\overrightarrow{x_i} - \overrightarrow{y}\right) \cdot \overrightarrow{\nabla} \zeta_{\sigma_j}\left(||\overrightarrow{x_i} - \overrightarrow{y}||\right)}{||\overrightarrow{x_i} - \overrightarrow{y}||^2} \left(\widetilde{\overrightarrow{\omega}}(\overrightarrow{x_i}) - \widetilde{\overrightarrow{\omega}}(\overrightarrow{y})\right) dV(\overrightarrow{y}) \tag{16}$$

$$\approx \sum_{j}^{N_p} (\nu_i + \nu_j) \frac{\overrightarrow{r_{ij}} \cdot \overrightarrow{\nabla}\zeta_{\sigma_j}(r_{ij})}{r_{ij}^2} V_j V_i \left(\overrightarrow{\widetilde{\omega_i}} - \overrightarrow{\widetilde{\omega_j}}\right) \quad (17)$$

$$\approx \sum_{j}^{N_p} (\nu_i + \nu_j) \frac{1}{\rho\sigma_j^2} \frac{\partial \zeta_{\sigma_j}}{\partial \rho}(r_{ij}) \left(V_j \overrightarrow{\alpha_i} - V_i \overrightarrow{\alpha_j}\right) \quad (18)$$

The main limit of this method is at the periphery of the domain. Since there are no particles beyond the domain, no vorticity can be shared beyond the space delimited by the particles. Frequent redistribution, like the one shown in 2.3.2 for example, has to follow the use of the PSE to gradually add particles beyond the boundaries of the domain to properly spread the vorticity.

A reformulation PSE (rPSE) is proposed to address the limitations of the classical PSE (cPSE). After all the terms of the vorticity equation have been computed, a cartesian grid of size h is overlapped with the particles of the domain. Each time a particle is within a cell, the four corners of this cell are removed. This eventually leaves nodes outside and within the gaps of the domain. Ghost particles with an empty strength are then positioned on each of those nodes. The PSE is then carried out a second time between the ghost particles and their closest particle neighbours.

$$\left[V_i \overrightarrow{\nabla} \cdot \left(\nu_i \overrightarrow{\nabla} \otimes \overrightarrow{\widetilde{\omega_i}}\right)\right]_{\text{loss}} \approx (\nu_i + \nu_{\text{air}}) V_{\text{ghost}} \overrightarrow{\alpha_i} \sum_{j \in \mathcal{P}_{\text{ghost}}} \frac{1}{\rho_{ij}\sigma_j^2} \frac{\partial \zeta_{\sigma_j}}{\partial \rho} \quad (19)$$

The ghost particles are then deleted. This process amounts to a transfer of the vorticity of the domain to the free-flow. This reformulation of the PSE is then non-conservative by nature but allows for a more cost-effective method than when frequently redistributing the particles.

2.2.2. Core Spreading Method (CSM)

The CSM[18–21] forces the particles to grow in order to follow the solution of a purely diffusive flow. This is achieved by modifying σ so the vorticity the particles induce becomes a solution of the following equation:

$$\frac{\mathrm{d}\omega_i}{\mathrm{d}t} = \nu_i \Delta \overrightarrow{\omega_i} \quad (20)$$

This is verified when using the Gaussian filter as long as:

$$\frac{\mathrm{d}\sigma_i}{\mathrm{d}t} = \frac{\nu_i}{\sigma_i} \quad (21)$$

The vorticity equation is solved in two stages. It is first solved using only the stretching term, then the cores of the particles are updated following (21).

2.2.3. Diffusion Velocity Method (DVM)

The DVM[22–25] rewrites the vorticity equation as an advection equation by incorporating the diffusion term inside the material derivatives. The implementation that follows comes from the developments made by Mycek[22, 23]:

$$\frac{\partial \overrightarrow{\widetilde{\omega}}}{\partial t} + \left(\overrightarrow{\widetilde{U}} \cdot \overrightarrow{\nabla}\right)\overrightarrow{\widetilde{\omega}} = \left(\overrightarrow{\widetilde{\omega}} \cdot \overrightarrow{\nabla}\right)\overrightarrow{\widetilde{U}} + \overrightarrow{\nabla} \cdot \left(\nu \overrightarrow{\nabla} \otimes \overrightarrow{\widetilde{\omega}}\right) \quad (22)$$

$$\frac{\partial \overrightarrow{\widetilde{\omega}}}{\partial t} + \overrightarrow{\nabla} \cdot \left(\left(\overrightarrow{\widetilde{U}} - \nu \overrightarrow{\nabla}\right) \otimes \overrightarrow{\widetilde{\omega}}\right) = \left(\overrightarrow{\widetilde{\omega}} \cdot \overrightarrow{\nabla}\right)\overrightarrow{\widetilde{U}} \quad (23)$$

$$\frac{\partial \overrightarrow{\widetilde{\omega}}}{\partial t} + \overrightarrow{\nabla} \cdot \left(\left(\overrightarrow{\widetilde{U}} + \overrightarrow{u}_d\right) \otimes \overrightarrow{\widetilde{\omega}}\right) = \left(\overrightarrow{\widetilde{\omega}} \cdot \overrightarrow{\nabla}\right)\overrightarrow{\widetilde{U}} \quad (24)$$

The expression of the diffusion velocity $\overrightarrow{u}_d$ is obtained by minimising the residual sum of squares of the tensor:

$$\overline{\overline{B}} = \overrightarrow{u}_d \otimes \overrightarrow{\omega} - \left(-\nu \overrightarrow{\nabla} \otimes \overrightarrow{\omega}\right) \quad (25)$$

Eventually the diffusion velocity can be written as:

$$\overrightarrow{u_{d_i}} = -(\nu + \nu_{t_i}) \frac{\sum_j ||\overrightarrow{\alpha_j}|| \overrightarrow{\nabla}\zeta_{\sigma_j}(r_{ij})}{\sum_j ||\overrightarrow{\alpha_j}|| \zeta_{\sigma_j}(r_{ij})} \quad (26)$$

Using this approach leaves a residual error which can be written as:

$$V_i \overrightarrow{\nabla} \cdot \overline{\overline{B}}_i = \sum_j \frac{\nu_i ||\overrightarrow{\alpha_i}|| V_j + \nu_j ||\overrightarrow{\alpha_j}|| V_j}{\rho\sigma^2} \frac{\partial \zeta_{\sigma_j}}{\partial \rho}(r_{ij}) \left(\frac{\overrightarrow{\alpha_i}}{||\overrightarrow{\alpha_i}||} - \frac{\overrightarrow{\alpha_j}}{||\overrightarrow{\alpha_j}||}\right) \quad (27)$$

The advection vorticity equation becomes:

$$\frac{\mathrm{d}\overrightarrow{\alpha_i}}{\mathrm{d}t} = \frac{\partial \overrightarrow{\alpha_i}}{\partial t} + \overrightarrow{\nabla} \cdot \left(\left(\overrightarrow{U_i} + \overrightarrow{u_{d_i}}\right) \otimes \overrightarrow{\alpha_i}\right) = \left(\overrightarrow{\alpha_i} \cdot \overrightarrow{\nabla}\right)\overrightarrow{U_i} + V_i \overrightarrow{\nabla} \cdot \overline{\overline{B_i}} \quad (28)$$

The modification of the material derivative is reflected upon the transport of the particles:

$$\frac{\mathrm{d}\overrightarrow{x_i}}{\mathrm{d}t} = \overrightarrow{U}_\infty + \overrightarrow{u}_{\phi_i} + \overrightarrow{u}_{\psi_i} + \overrightarrow{u_{d_i}} \quad (29)$$

While the induced velocities are divergence-free by construction, it is not the case of the diffusion velocity. This means that the volume of the particles, and thus their core size ($V_i \propto \sigma_i^3$) have to evolve according to:

$$\frac{\mathrm{d}V_i}{\mathrm{d}t} = \int_{V_i} \overrightarrow{\nabla} \cdot \overrightarrow{u}_d dV \approx V_i \overrightarrow{\nabla} \cdot \overrightarrow{u_{d_i}} \quad (30)$$

$$i.e. \quad \frac{\mathrm{d}\sigma_i}{\mathrm{d}t} = \frac{\sigma_i}{3} \overrightarrow{\nabla} \cdot \overrightarrow{u_{d_i}} \quad (31)$$

where:

$$\overrightarrow{\nabla} \cdot \overrightarrow{u_{d_i}} = \frac{1}{\nu_i}\left(\overrightarrow{\nabla}\nu_i + \overrightarrow{u_{d_i}}\right) \cdot \overrightarrow{u_{d_i}} - \nu_i \frac{\sum_j ||\overrightarrow{\alpha_j}||\Delta\zeta_{\sigma_j}(r_{ij})}{\sum_j ||\overrightarrow{\alpha_j}||\zeta_{\sigma_j}(r_{ij})} \quad (32)$$

$$\approx \frac{1}{\nu_i}||\overrightarrow{u_{d_i}}||^2 - \nu_i \frac{\sum_j ||\overrightarrow{\alpha_j}||\Delta\zeta_{\sigma_j}(r_{ij})}{\sum_j ||\overrightarrow{\alpha_j}||\zeta_{\sigma_j}(r_{ij})} \quad (33)$$

2.3. Other Modules

2.3.1. Control of the divergence

The major drawback of the VPM is that even though the velocity is divergence-free, it is not necessarily the case of the vorticity[6]. This means that $\overrightarrow{\omega_i} = \overrightarrow{\nabla} \times \overrightarrow{u_i}$ is not necessarily verified. Different methods exist to mitigate those effects[6, 18, 20, 26] but they are either marginally effective or rather costly. A new approach is proposed in this work using the Enstrophy of the particles:

$$\mathcal{E} = \int \overrightarrow{\omega} \cdot \overrightarrow{\omega} dV \quad (34)$$

Eq. (34) can either be computed using the divergence-free vorticity obtained from the curl of the velocity $\overrightarrow{\nabla} \times \overrightarrow{u_i}$ or use the expression of the filtered vorticity induced by the particles in Eq. (10). The difference between the values of the real Enstrophy (noted $\mathcal{E}$) and the non divergence-free Enstrophy (noted $\mathcal{E}_f$) gives an estimate of the behaviour of the simulation and whether or not the flow remains divergence-free.

A rough approximation of the Enstrophy when only considering the contribution of the neighbouring particles is:

$$\mathcal{E}_i \propto \frac{\overrightarrow{\alpha_i} \cdot \overrightarrow{\alpha_i}}{\sigma^3} \quad (35)$$

The ratio between the two Enstrophy then gives a measure of the error of the magnitude of the strength of the particles:

$$\frac{\mathcal{E}_i}{\mathcal{E}_{f_i}} \approx \frac{||\overrightarrow{\alpha_i}||^2_{\text{new}}}{||\overrightarrow{\alpha_i}||^2_{\text{old}}} \quad (36)$$

The relaxation of the magnitude of the strength of the particles through the Enstrophy allows for a better control of the divergence of the flow and contributes to its stabilisation:

$$||\overrightarrow{\alpha}_{i\text{new}}|| = \left(1 + f\left(\sqrt{\frac{\mathcal{E}_i}{\mathcal{E}_{f_i}}} - 1\right)\right)||\overrightarrow{\alpha}_{i\text{old}}|| \quad (37)$$

It does not, however, conserve the total circulation. This expression resembles the particle reorientation done by Pedrizzetti[26], where the direction of the particles is similarly modified to match the direction of $\overrightarrow{\nabla} \times \overrightarrow{U_i}$.

As highlighted in other studies[9, 17], the expression of $\mathcal{E}$ given in [6] is not dimensionally consistent. The calculations of the general expression of the Enstrophy is given in Appendix A.

2.3.2. Control of the Lagrangian Distortion

Eulerian methods require the mesh to envelop the whole domain to properly simulate the regions of interest. The VPM however only require particles to be present in the region of interest where the vorticity is strong enough, since the particle domain shapes itself depending on the velocity gradients. On the other hand this advantage can lead to Lagrangian distortion because of the gathering or dispersion of particles due to high velocity gradients and strong vorticity strains. This tends to destabilise the flow and to create gaps in the flow field, making the representation of the concerned regions inaccurate. This calls for a periodic redistribution by locally merging or splitting the particles. When two particles are close enough and sufficiently aligned with each other, they can be merged into a bigger particle:

$$\overrightarrow{\alpha}_{\text{new}} = \overrightarrow{\alpha_1} + \overrightarrow{\alpha_2} \quad (38)$$

$$\overrightarrow{x}_{\text{new}} = \frac{||\overrightarrow{\alpha_1}||\overrightarrow{x_1} + ||\overrightarrow{\alpha_2}||\overrightarrow{x}_2}{||\overrightarrow{\alpha_1}|| + ||\overrightarrow{\alpha_2}||} \quad (39)$$

However if a particle becomes too strong, it can be split into two smaller particles:

$$\overrightarrow{\alpha}_{\text{new}} = \frac{1}{2}\overrightarrow{\alpha}_{\text{old}} \quad (40)$$

$$\overrightarrow{x}_{\text{new}} = \overrightarrow{x}_{\text{old}} \pm \frac{\sigma_{\text{old}}}{2\lambda_\sigma||\overrightarrow{\alpha}_{\text{old}}||}\overrightarrow{\alpha}_{\text{old}} \quad (41)$$

2.3.3. Turbulent Viscosity Control

The turbulent viscosity used to account for the truncature error and close the VPM equations has to be computed through a model. While a variety of LES turbulent viscosity models are available in the literature, the limited number of resolved quantities available with the VPM restrict the panel of models available. Three models were implemented in the solver, namely the Smagorinski model, the Mansour model[27] and the Vreman model[28]. Unfortunately they all come from the hypothesis of homogeneous and isotropic turbulence. One of the issue of those models is their dependence on the Smagorinski constant, C_s, typically ranging from 0.1 to 0.2, which can emphasise or downplay the role of the turbulence in the flow. One way to dynamically adjust this constant is through the Germano identity[29].

However, this method is costly and difficult to implement, especially for Lagrangian frameworks.

An alternative is proposed where the Smagorinski constant is adjusted according to the loss of Enstrophy in the flow. The computation of the following equation is detailed in Appendix B:

$$\frac{\partial \widetilde{\mathscr{E}_c}}{\partial t} = -\nu \widetilde{\mathcal{E}} - \int \nu_t ||\overrightarrow{\nabla} \otimes \overrightarrow{\widetilde{U}}||^2 dV \tag{42}$$

Eq. (42) shows how the loss of Kinetic Energy is linked to the Enstrophy, $i.e.$ the transfer of energy to the small scales because of viscous effects. When no turbulent viscosity is used, the loss of Kinetic Energy is driven by the molecular viscosity.

$$\frac{\partial \mathscr{E}_c}{\partial t} = -\nu \mathcal{E} \tag{43}$$

To avoid reverse energy cascade from the small scales to the resolved scales, the tendency of the Kinetic Energy loss can be controlled so as to have:

$$0 < \frac{\partial^2 \mathscr{E}_c}{\partial t^2} = -\nu \frac{\partial \mathcal{E}}{\partial t} \tag{44}$$

$$i.e. \qquad \frac{\partial \mathcal{E}}{\partial t} < 0 \tag{45}$$

The Kinetic Energy loss can then be controlled by increasing or decreasing the Smagorinski constant to emphasise or dampen the viscous effects in the flow and adjust the loss of Enstrophy. If the derivative of the Enstrophy becomes positive (resp. negative), the Smagorinski constant is increased (resp. decreased) by a given step. The minimum and maximum values of C_s are caped in order to avoid having negative viscosity or nonphysical over diffusion.

2.3.4. Acceleration Method

Direct resolution of the aforementioned VPM governing equations is costly. The inter-particular interactions makes it a $O(N^2)$ problem, preventing in practice from efficiently using more than 10^5 particles when using modern CPUs. The Fast Multipole Method (FMM) can be used to efficiently reduce the number of inter-particular interactions to reduce the cost to $O(N)$. The open source module ExaFMM[30] was chosen for this purpose. It was fully optimised, parallelised with OpenMP, vectorised and implemented into the VPM solver. This module regroups particles in clusters depending on their distribution in space. The clusters are subdivided until a given amount of particles is in each cluster. Interactions between two clusters far apart can be summed up as a single interaction, thus reducing inter-particular interactions to particles close to each other, as shown in Fig. (1). The readers can refer to [30] for more details about the FMM.

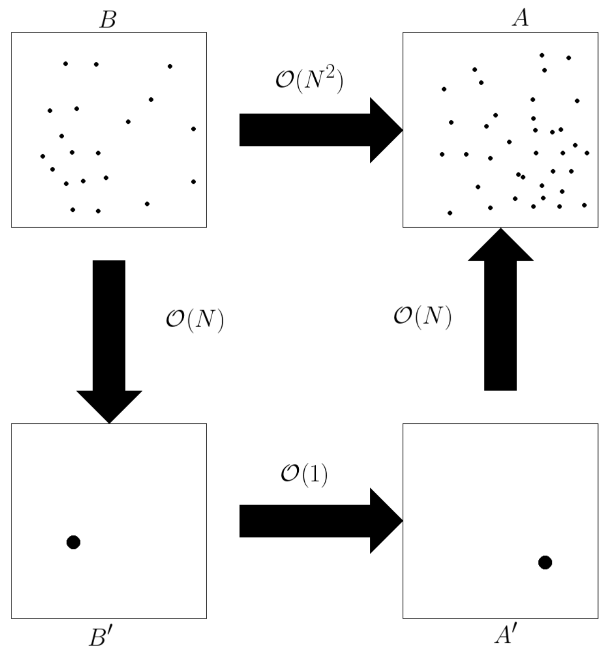


Fig. 1 Concept of the FMM.

Fig. (2) compares the CPU time taken to compute the induced velocity and its gradients, the induced vorticity, the diffusion velocity and its divergence and the DVM error tensor used by the VPM equations. The particles are randomly disposed in a 1m-long cube and are assigned a random strength.

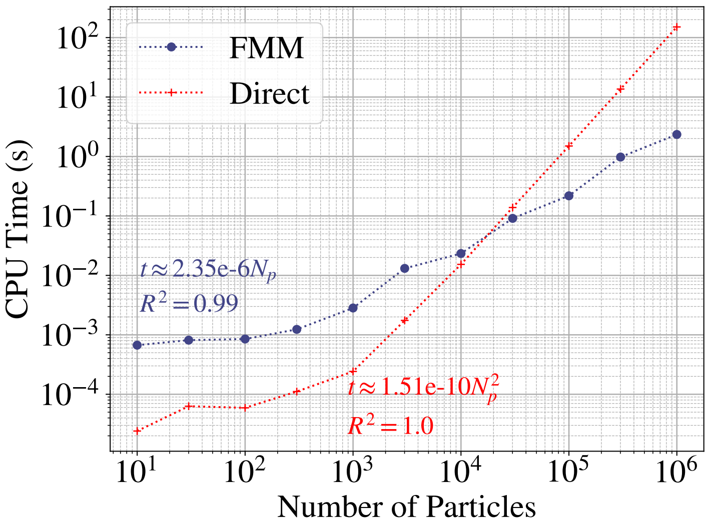
The computations were done with a $2.10GHz$ Intel CPU with an AVX architecture, allowing 256bits to the width of the SIMD registers for the vectorization.

The Third-Order Low-Storage Runge-Kutta integration scheme is used to avoid overburdening the memory of the simulation while retaining a high enough order of precision[31].

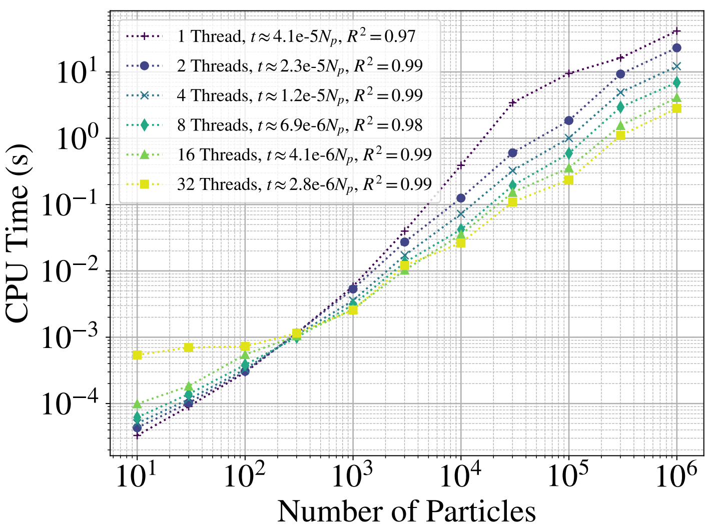
3. Lifting-Line COUPLING

3.1. Generation of Particles

A Lifting-Line (LL) module was added to the VPM solver to represent the influence of a lifting body in an unbounded free flow. This century-old method remains relevant to this day thanks to its low computational cost and simplicity. This section explains how the LL sheds



(a) FMM and Direct CPU time with 44 threads.



(b) FMM CPU time based on the number of threads.

Fig. 2 CPU time for the VPM

vortices in the wake and how it interacts with the wake it generates.

The LL is first split into small segments. Depending on the lifting body, (wings, propellers, rotors ...), the geometrical discretisation of the segments will vary along the wingspan to better discretise the tips and capture the wingtip vortices.

The LL is moved according to its own solid translation and rotation. The effective angle of attack, $\alpha_e = \alpha - \alpha_i$, is computed by summing the contribution of the kinematic velocity of the LL, the free stream and the velocities induced by the LL and the particles. The aerodynamic loads on the LL are then computed on each segment thanks to a database of 2D polars linking α_e to the aerodynamic coefficients of the LL. The polars can be obtained either from 2D CFD computations, experimental data, XFoil ... Fig. (3) shows the loads and angles for each 2D sections of the LL.

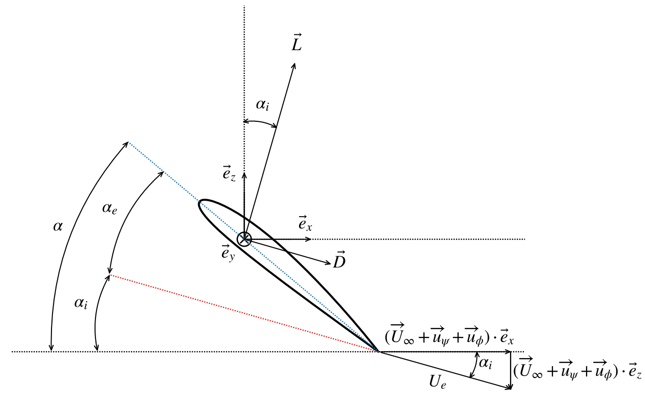


Fig. 3 Aerodynamic loads on a 2D airfoil.

$$\begin{cases} L = \frac{1}{2}\rho_\infty \int U^2 c C_L(\alpha_e, R_e, M) dy \\ D = \frac{1}{2}\rho_\infty \int U^2 c C_D(\alpha_e, R_e, M) dy \end{cases} \tag{46}$$

The circulation is then recovered on each airfoil with the Kutta-Joukowski theorem:

$$\vec{L} = \rho_\infty \int \vec{U} \times \vec{\Gamma} dy = \rho_\infty \int \Gamma \vec{U} \times \vec{dy} \tag{47}$$

$$\text{i.e.} \quad \vec{\Gamma} = \frac{1}{2} c U C_L \vec{e}_y = \Gamma \vec{e}_y \tag{48}$$

In their current form, the airfoil polars and the LL components used in the solver do not use any specific corrections for post-stall conditions[32], 3D flow or unsteady loading[33].

According to Kelvin's theorem, the introduction of a solid in the flow has to be balanced with the generation of vorticity because of the variations of circulation on the LL. Two contributions can be identified,

$$\frac{\mathrm{d}\Gamma}{\mathrm{d}t} = \frac{\partial \Gamma}{\partial t} + \vec{U} \cdot \nabla \Gamma = 0 \tag{49}$$

These two terms contribute to the generation of vorticity because of the temporal rate of change and the spanwise variation of the circulation on the LL. To respect this, particles are shed along the LL and are set free in the wake[8, 17, 20, 34]. The spatial (resp. temporal) variation of circulation contribute to the generation of vorticity perpendicular (resp. parallel) to the LL. Particles are then generated on N_s stations on the LL, independently of its geometrical discretisation. The shedding procedure proposed in [35] is here declined for LL applications. Let us consider the coordinate system on each section of the LL where $\overrightarrow{i} = \overrightarrow{e}_y$ is the normal of the section, $\overrightarrow{j}$ is the direction of the projection of the total velocity $\overrightarrow{U}$ on the airfoil section and $\overrightarrow{n} = \overrightarrow{i} \times \overrightarrow{j}$. The strength of the particle to generate can be linked to the intensity of an equivalent vortex sheet that is to be shed in the wake along the $\overrightarrow{j}$ direction:

$$\overrightarrow{\alpha_s} = \int_s \overrightarrow{\gamma} dS \tag{50}$$

$$= \int_s \left(\overrightarrow{n} \times \overrightarrow{\nabla}_s \Gamma\right) dS \tag{51}$$

$$= \int_s \left(-\frac{\partial \Gamma}{\partial x_j}\overrightarrow{i} + \frac{\partial \Gamma}{\partial x_i}\overrightarrow{j}\right) dS \tag{52}$$

Using Eq. (49), one gets:

$$\overrightarrow{\alpha_s} = \int_s \left(\frac{1}{\overrightarrow{U} \cdot \overrightarrow{e_j}} \left(\frac{\partial \Gamma}{\partial t} + \overrightarrow{U} \cdot \overrightarrow{e_i} \frac{\partial \Gamma}{\partial x_i}\right)\overrightarrow{i} + \frac{\partial \Gamma}{\partial x_i}\overrightarrow{j}\right) dS \tag{53}$$

$$\approx \left(\frac{1}{U_{s_j}} \left(\frac{\partial \Gamma_s}{\partial t} + U_{s_i} \frac{\partial \Gamma_s}{\partial x_i}\right)\overrightarrow{i}_s + \frac{\partial \Gamma_s}{\partial x_i}\overrightarrow{j}_s\right) dx_{s_i} dx_{s_j} \tag{54}$$

Since $U_{s_i} \ll U_{s_j}$ and the length of the sheet is $dx_{s_j} \approx U_{s_j} dt$:

$$\overrightarrow{\alpha_s} \approx dx_{s_i} dt \frac{\partial \Gamma_s}{\partial t}\overrightarrow{i}_s + dx_{s_j} dx_{s_i} \frac{\partial \Gamma_s}{\partial x_i}\overrightarrow{j}_s \tag{55}$$

where:

$$\begin{cases} dt \frac{\partial \Gamma_s}{\partial t} \approx \Gamma_s(t) - \Gamma_s(t - \Delta t) \\ dx_{s_i} \frac{\partial \Gamma_s}{\partial x_i} \approx \frac{1}{12}\left(-\Gamma_{s+2} + 8\Gamma_{s+1} - 8\Gamma_{s-1} + \Gamma_{s-2}\right) \end{cases} \tag{56}$$

and the circulation outside the LL is null.

In turn, the particles that are generated in the wake will induce velocities on the blade elements of the LL which will affect the effective angles of attack, thus the loads, thus the circulation. A sub-iteration process is used at each timestep until convergence of the circulation is reached. To do so, a relaxation scheme is introduced when it is updated:

$$\Gamma_{n_{\text{new}}} = \Gamma_{n-1_{\text{new}}} + k_{\text{relax}}(\Gamma_n ofvorticitythat - \Gamma_{n-1_{\text{new}}}) \tag{57}$$

at the n^{th} sub-iteration. Eventually the particles are set free in the wake to form the wake behind the LL.

Increasing the time step will increase the area swiped by the LL and generating a single line of particles behind the LL will leave gaps in the wake. To avoid that, the number of particles generated at each station is also increased in order to entirely fill the gap between the previously generated line of particles and the LL, as shown in Fig. (4). The particles generated behind a station will see their strength shared equally.

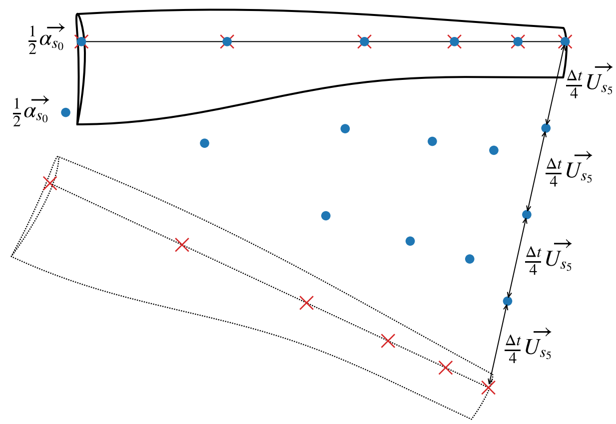


Fig. 4 Decoupled generation of particles.

This decoupling process is especially convenient for propeller applications. It allows for bigger time steps in order to save computation time while properly shedding vorticity in the wake and having the particles overlapping with each other.

3.2. Propeller Induction

The presence of a solid in the flow generates vorticity as shown in the previous section, but the solid also induce velocities on itself and in the wake. Each segment of the LL has its own bound circulation which will induce velocities in the surrounding flow as shown in Fig. (5).

The velocities induced by the solid are obtained with the Biot-Savart law as proposed by Vatistas[36, 37]:

$$\overrightarrow{u}_{\phi_i} = \sum_{j}^{Ns-1} \frac{\Gamma_0}{4\pi} \frac{\overrightarrow{r_0} \cdot \left(\frac{\overrightarrow{r}_1}{||\overrightarrow{r}_1||} - \frac{\overrightarrow{r}_2}{||\overrightarrow{r}_2||}\right)}{\left(||\overrightarrow{r}_1 \times \overrightarrow{r}_2||^{2n} + ||\sigma \overrightarrow{r_0}||^{2n}\right)^{1/n}} \overrightarrow{r}_1 \times \overrightarrow{r}_2 \tag{58}$$

where $\Gamma_0 = (\Gamma_{j+1} + \Gamma_j)/2$, $\overrightarrow{r}_2 = \overrightarrow{x} - \overrightarrow{x}_{j+1}$, $\overrightarrow{r}_1 = \overrightarrow{x} - \overrightarrow{x}_j$ and $\overrightarrow{r}_0 = \overrightarrow{x}_{j+1} - \overrightarrow{x}_j = \overrightarrow{r}_j - \overrightarrow{r}_{j+1}$. The velocity induction of the solid also comes with the induction of velocity

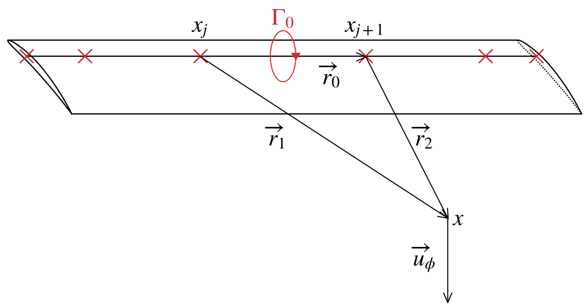


Fig. 5 Velocity induction of the solid in the flow.

gradients which are incorporated in the stretching term of the vorticity equation. Indeed on has:

$$\vec{\nabla} \otimes \vec{U} = \vec{\nabla} \otimes \vec{u}_\psi + \vec{\nabla} \otimes \vec{u}_\phi \tag{59}$$

The computation of the gradients of this induced velocity is detailed in Appendix C:

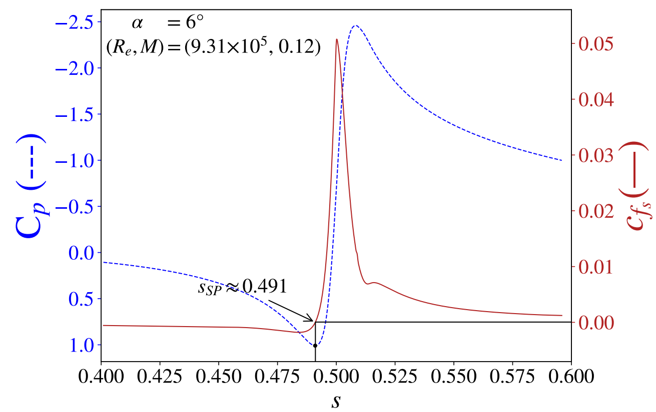
$$\begin{aligned}
\vec{\nabla} \otimes \vec{u}_{\phi_i} = \sum_{j}^{N_s-1} \frac{\Gamma_0}{4\pi} \frac{1}{\left(||\vec{r}_1 \times \vec{r}_2||^{2n} + ||\sigma \vec{r_0}||^{2n}\right)^{1/n}} \Bigg[\\
\vec{r_0} \cdot \left(\frac{\vec{r}_1}{||\vec{r}_1||} - \frac{\vec{r}_2}{||\vec{r}_2||} \right) \vec{\nabla} \otimes \left(\vec{r}_1 \times \vec{r}_2 \right) \\
+ \left(\left(\frac{\vec{r}_1 \cdot \vec{r}_2}{r_1^3} - \frac{1}{r_2} \right) \vec{r}_1 + \left(\frac{\vec{r}_1 \cdot \vec{r}_2}{r_2^3} - \frac{1}{r_1} \right) \vec{r}_2 \right. \\
\left. + \frac{2||\vec{r}_1 \times \vec{r}_2||^{2(n-1)} \vec{r_0} \cdot \left(\frac{\vec{r}_1}{||\vec{r}_1||} - \frac{\vec{r}_2}{||\vec{r}_2||} \right)}{||\vec{r}_1 \times \vec{r}_2||^{2n} + ||\sigma \vec{r_0}||^{2n}} \vec{r_0} \times \left(\vec{r}_1 \times \vec{r}_2 \right) \right) \\
\otimes \left(\vec{r}_1 \times \vec{r}_2 \right) \Bigg]
\end{aligned}$$

The value of $n = 1$ was chosen for this study.

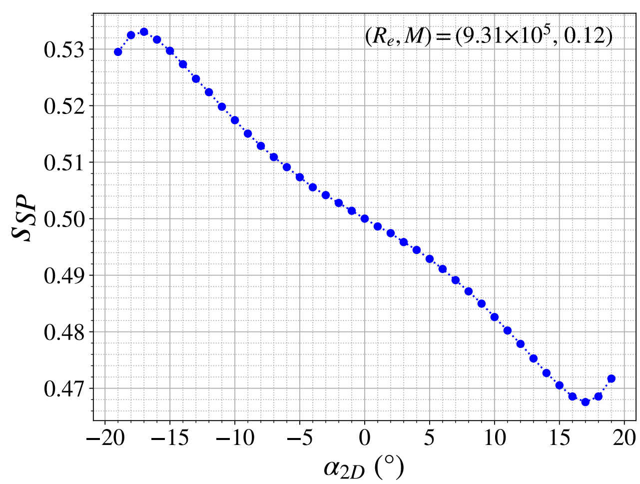
3.3. Equivalent 2D Angle of Attack

While the effective angle of attack seen by the lifting body can be directly retrieved from the Lifting Line ($\alpha_e = \alpha - \alpha_i$), this is not the case for the CFD or other 3D lifting bodies. The following section introduces a method for approximating the 2D effective angle of attack of a 3D lifting body. This will allow for useful comparisons between the CFD and the VPM-LL solver in the validation section.

First the pressure and friction coefficients distributions around the walls of a 2D airfoil can be retrieved from the polar database used for the VPM-LL computations. These coefficients are used to infer the curvilinear abscissa of the stagnation point, s_{SP}, for each angle of attack based on the chord-based Reynolds and Mach numbers. The stagnation point can be retrieved from the maximum of pressure or the change of the sign of the friction along the wall of a 2D airfoil. In this manner, a function $f(s_{SP}, R_e, M) = \alpha_e$ can be obtained by linear interpolation as in Fig. (6).



(a) Position of the stagnation point.



(b) Stagnation point as a function of α.

Fig. 6 Computation of the curvilinear abscissa of the stagnation point as a function of α, R_e and M for the NACA 0012 airfoil.

The stagnation points along the 3D lifting body are then computed to be used by the interpolation function to approximate its equivalent 2D angles of attack.

4. VALIDATION OF THE VPM SOLVER

The Vortex Particle Method (VPM) solver is first validated using flow configurations with known analytical solutions. The Lamb-Oseen Vortex is first simulated to compare the different diffusion methods available and validate the diffusion term of the vorticity equation. An Inviscid Vortex Ring is studied to validate the stretching term of the vorticity equation. Eventually Viscous Vortex Rings are used to validate both terms of the vorticity equation.

4.1. The Lamb-Oseen Vortex

The Lamb-Oseen Vortex is a 2D, cylindrical, unbounded vortex solution of the unsteady, incompressible Navier-Stokes equations[38]. This vortex is characterised by a concentration of vorticity that decays and spreads due solely to viscous effects. This allows for the study of the diffusion term of the vorticity equation independently of the stretching term. Let us note $a(t)$ the core size of the vortex:

$$\begin{cases} \vec{\omega}(r,t) &= \frac{\Gamma}{\pi a(t)^2}\left(1 - e^{-r^2/a(t)^2}\right)\vec{e_z} \\ \vec{U}(r,t) &= \frac{\Gamma}{2\pi r}\left(1 - e^{-r^2/a(t)^2}\right)\vec{e_\theta} \\ a(t) &= \sqrt{4\nu t + a_0^2} \end{cases} \tag{60}$$

The Lamb-Oseen Vortex is here generalised in 3D by being elongating into a tube. To approximate an infinitely long tube, the changes of strength of the particle on the section at the middle of the finite tube are passed on the other sections. The simulation is started at $a(t_0 = 0) = a_0 = 0.1$ m with $\nu = 10^{-3}$ m^2.s^{-1} and $\Gamma = 1$ m^2.s^{-1}. Particles are initialised up to $r = 2a_0$ and the resolution scale of the VPM is $h = 0.1a_0$ with $\lambda_\sigma = 2$.

The particles of each sections are initialised by solving (10) with the Richardson method:

$$\vec{\alpha}_i^{\,n+1} = \vec{\alpha}_i^{\,n} + V_i\left(\vec{\omega}(r_i, t_0) - \vec{\omega}_i^{\,n}\right) \tag{61}$$

Fig. (7) and Fig. (8) show the evolution of the vorticity and the radius of the vortex for the analytical case as in Eq. (60), for the classical Particle Strength Exchange (cPSE), for the reformulated Particle Strength Exchange (rPSE), for the Diffusion Velocity Method (DVM) and for the Core Spreading Method (CSM). The radius of the vortex is computed from the radius where the velocity is maximum. Using the analytical expressions in (60), one has:

$$r^2_{U_{\max}(t)} \approx 1.25643121a(t)^2 \tag{62}$$

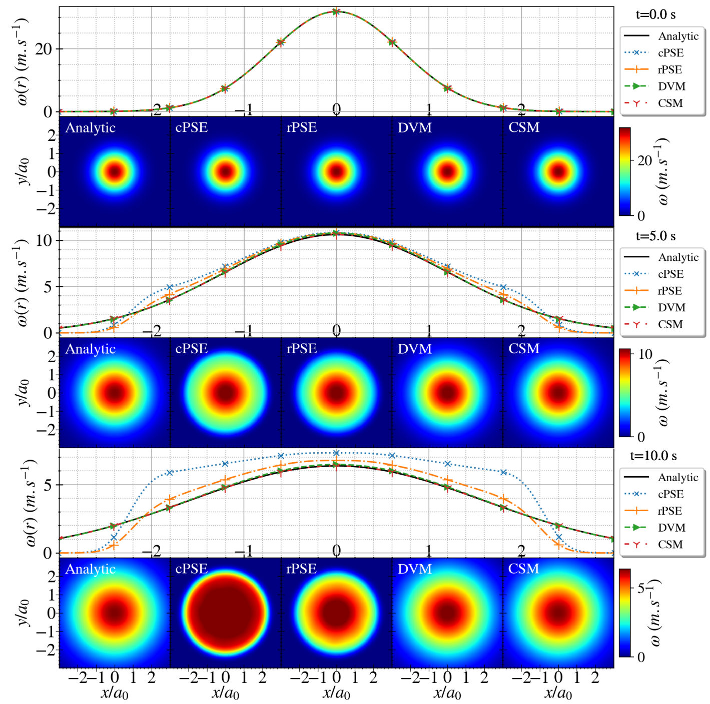


Fig. 7 Lamb-Oseen Vortex vorticity evolution for different viscosity methods.

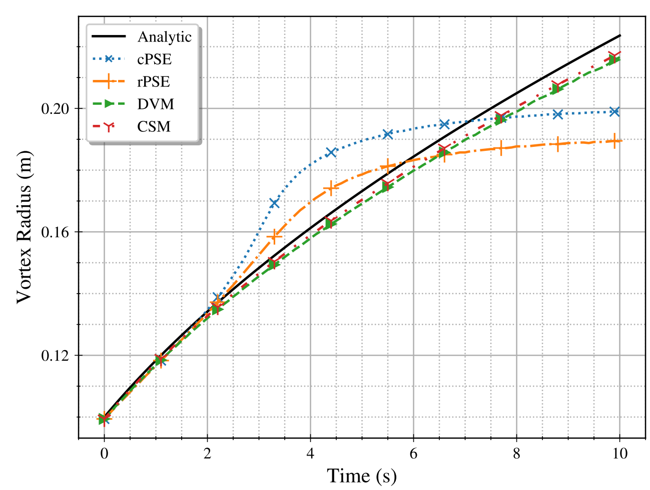


Fig. 8 Lamb-Oseen Vortex radius evolution for different viscosity methods.

After ≈ 3 s, the vorticity reaches the edges of the initial discretisation of the tube $(2a_0)$. After this point, both PSE methods have no more particles to share the vorticity with and the vorticity begins homogenising in the vortex. This shows that the PSE has to be used alongside a frequent redistribution of the particles at the edges of the domain to represent the spread of the vorticity and growth of the vortex. This is also shown in Fig. (8) where the core size of the vortex does not grow beyond ≈ 5 s.

The rPSE allows for a more accurate loss of vorticity in the freeflow. Indeed a step down of vorticity can be seen at $r \approx 2a_0$. Beyond this point the vorticity and velocity are off however since no particle are present.

However both the DVM and the CSM properly simulate the spread of vorticity and decay of velocity in the vortex. The CSM however, because the viscosity is important, makes the particles grow fast in size. This means that the inter-particular approximation made by the FMM are less and less accurate. This means that the size of the particles has to be reset. Solving (10) is too costly and initialising the particles by using:

$$\overrightarrow{\alpha_i} \approx V_i \overrightarrow{\omega_i} \tag{63}$$

amounts to an over-diffusion of the vorticity[18]. In most cases however, the viscosity remains small enough so that the particles do not overgrow too much when using the CSM and the FMM remains accurate.

On the other hand the DVM can become singular when the vorticity becomes null, which tends to happen when vortex structures realign themselves with the freeflow after a long time. This can be avoided using:

$$\overrightarrow{u_{d_i}} = \frac{\left(\sum_j ||\overrightarrow{\alpha_j}|| \zeta_{\sigma_j}(r_{ij})\right)^2}{\omega_0^2 + \left(\sum_j ||\overrightarrow{\alpha_j}|| \zeta_{\sigma_j}(r_{ij})\right)^2} \overrightarrow{u_{d_i}}_{\text{sing}} \tag{64}$$

The ω_0 cutoff is set as a small fraction of the highest vorticity shed in the wake.

4.2. Vortex Rings

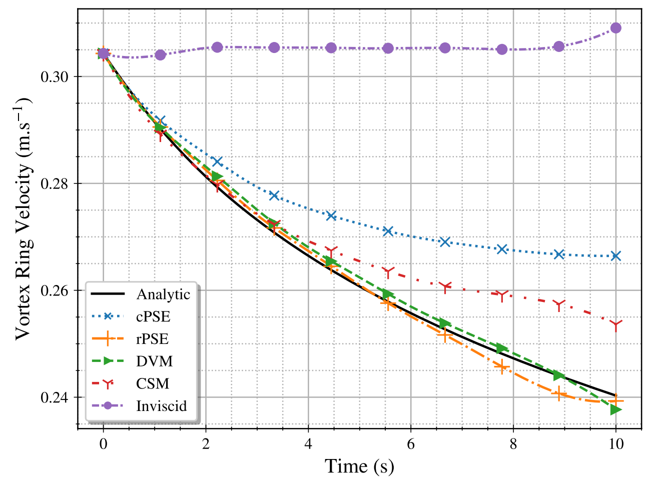
The Vortex Ring is a 3D unsteady vortex structure that self-induce velocities in order to translate itself. The same repartition of vorticity as with the Lamb-Oseen Vortex is used. In fact the ring can be seen as a bent tube closed upon itself. Since the ring is 3D and not infinitely thin, it is flattened and the vorticity and velocity do no longer follow the equations in (60). The translation velocity of the ring can however be estimated by [39]:

$$U(t) = \frac{\Gamma}{4\pi R}\left(\ln\left(\frac{8R}{a(t)}\right) + \frac{\gamma_{\mathrm{E}} - 1 - \ln 2}{2}\right) \tag{65}$$

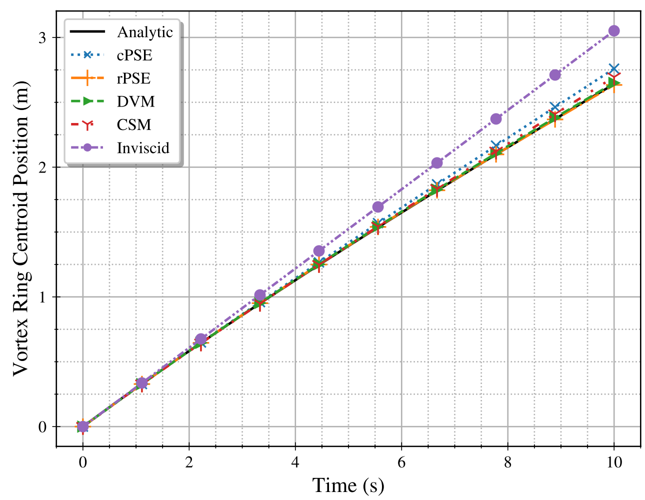
where γ_{E} is the Euler constant. The position of the ring can then be obtained:

$$X(t) = \int_{t_0=0}^{t} U(T)dT \tag{66}$$

$$= \frac{\Gamma}{4\pi R}\left(\left(\ln\left(\frac{8R}{a(t)}\right) + \frac{\gamma_{\mathrm{E}} - \ln 2}{2}\right)t + \frac{a_0^2}{4\nu}\ln\left(\frac{a_0}{a(t)}\right)\right) \tag{67}$$



(a) Velocity of the centroid.



(b) Position of the centroid.

Fig. 9 Translation of the Vortex Ring.

The case of the Inviscid Vortex Ring allows for the validation of the stretching term. Fig. (9a) and (10) show that the velocity and radius of the inviscid case remains constant through time and close to the theoretical values.

The Viscous Vortex Ring shows the interaction between both the stretching and diffusion terms. The viscous case leads to similar conclusion as with the Lamb-Oseen case.

The ring radius shown in Fig. (10) is harder to approximate than with the Lamb-Oseen case. This is due to the flattening of the ring which gives an elliptical shape to the ring section. The mean radius is computed by averaging the bigger and the smaller radii of the section. This gives the general trend of the section radius.

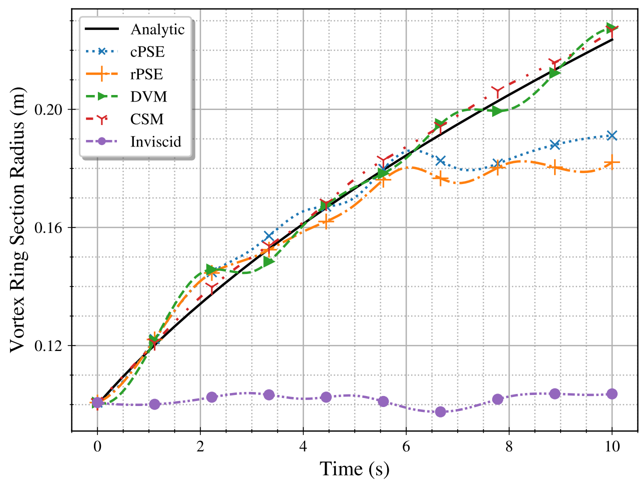


Fig. 10 Vortex Ring radius evolution for different viscosity methods.

The cPSE saturates and fail to properly diffuse the vorticity which eventually gives a higher velocity.

The rPSE manages to diffuse enough to the freeflow to correctly translate the ring. Since the sections of the ring do not expand, the ring itself do not grow. The core size of both PSE methods eventually stops growing.

The CSM produces accurate predictions up to 4s of simulation. Then, the vortex ring velocity is overestimated. This overestimation is probably caused by the significant errors during the FMM approximation, since this approximation is less accurate for overgrown particle cores which tend to more importantly interact with far away particles.

The DVM on the other hand manages to translate the ring and expand its core according to the theoretical values.

For the cases of moderate kinematic viscosity, both the CSM and the DVM appear to suitably represent vortex decay and vorticity diffusion. The DVM is however prefered for long simulation and wake decay analysis while the CSM is prefered for shorter simulation to benefit from its low computational cost. Nonetheless, the DVM will be used for the validation cases that follow.

5. VALIDATION OF THE COUPLED SOLVER

The coupled Vortex Particle Method - Lifting Line (VPM-LL) solver is validated in this section. Several test cases of wings and propellers are investigated and the results are compared to other numerical solvers, analytical theory and experimental data. The loads on the LL and the wake it generates are studied to ascertain the correct generation of vortex structures, their interaction with the lifting bodies from where they are shed, and their transport in the wake. The industrial CFD code used is the elsA solver[40] with the Wilcox (2006) $k-\omega$ turbulence model[41] on an overset cartesian mesh.

5.1. Twisted and Swept Wings

This section aims to observe the effects of the twist and sweep angles of the LL on the VPM-LL solver. To do so a rectangular wing at $\alpha = 8°$, with a constant chord of 0.3 m, an aspect ratio of 10 and NACA 4412 airfoil all along the wingspan is used. The freestream is set to 50 m.s^{-1}, which gives a chord-base Reynolds and Mach numbers of $\approx 1.03\times10^6$ and 0.147. The VPM inputs are $h = \frac{b}{50} = 0.06$ m, $\lambda_\sigma = 2$, $\Delta t = 3h/U_\infty$, $k_{\text{relax}} = 1/3$ and $C_s = 0.15$ with the Vreman viscosity model[28].

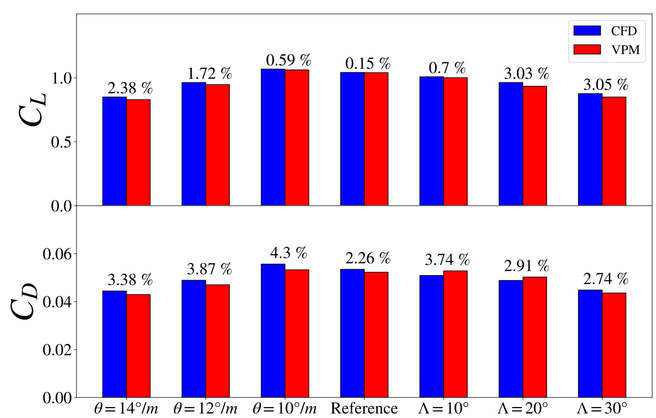


Fig. 11 Aerodynamic coefficients comparison for different twist (θ) and sweep (Λ) angles.

The Reference case shows good agreements with the CFD. This wing is then modified so as to have $\alpha = 7.5°$ at the middle of the wing, and $\alpha = -7.5°$, $-10°$ or $-13.5°$ at the tips. As shown, the VPM is in good agreement with the CFD concerning the influence of the twist angle θ, even at relatively high values of twist.

Despite the 2D hypothesis made by the LL, the integrated loads predicted by the VPM are also close to the CFD for the swept wing, even for relatively high values of sweep. It has indeed been observed that good results are obtained when the wing self-induction is taken into account for important sweep angles.

Overall the LL-VPM is accurate within respectively 3.05% and 4.3% regarding the 3D lift and drag coefficients compared to the CFD for all configurations.

5.2. The Elliptical Wing

The VPM-LL is then studied on an elliptical wing. This peculiar geometry offers convenient analytical results providing validation for the LL module.

The elliptical wing of this study has a reference chord of $c(0) = c_0 = 0.34$ m, a wingspan of $b = 1.6$ m and its chord repartition is $c(y) = c_0\sqrt{1-\left(\frac{2y}{b}\right)^2}$. The NACA 0012 airfoil is used for the whole wing and the airfoils are positioned so that the LL, located at the quarter chord of each airfoil, is straight. The wing is untwisted and has neither dihedral nor sweep. Its surface is given by $S = \frac{\pi}{4}c_0 b \approx 0.427m^2$. The specificity of this renowned wing, is that its shape minimises its induced drag and gives elliptical circulation and lift[42]:

$$\Gamma(y) = \Gamma_0\sqrt{1-\left(\frac{2y}{b}\right)^2} = \Gamma_0\frac{c(y)}{c_0} \tag{68}$$

The analytical 3D aerodynamic coefficients are provided by the Thin Airfoil Theory. The Prandtl–Glauert correction is used to account for compressible and 3D effects on the C_L. The drag coefficient is split into the parasitic and the induced drag. The former is obtained from the drag coefficients of the 2D polar database.

$$C_L = \frac{2\pi\alpha}{\sqrt{1-M^2}+2/AR} \tag{69}$$

$$C_D = \frac{1}{S}\int Cd(y)c(y)dy + \frac{C_L^2 S}{\pi b^2} \tag{70}$$

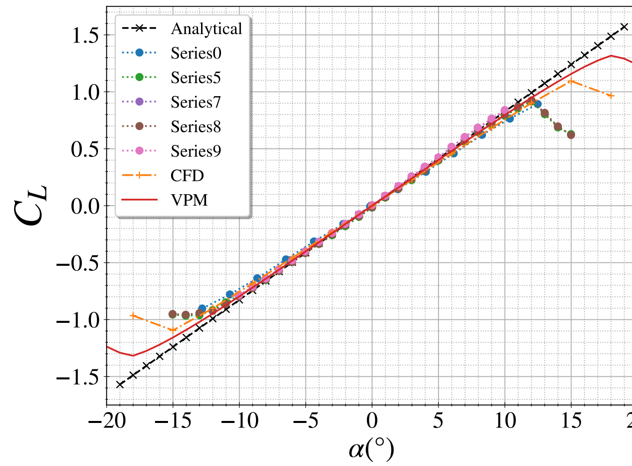
The velocity induced on the LL can be obtained from the circulation[42]:

$$w(y) = \frac{\Gamma_0}{2b} \tag{71}$$

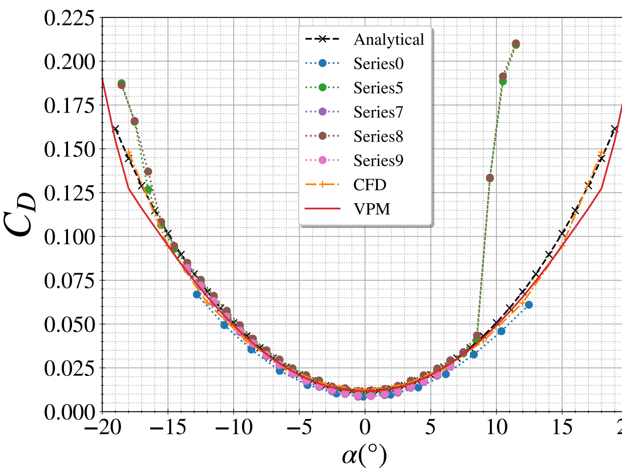
Since the induced velocity is constant on the whole wing, so is the induced angle $\alpha_i = \arctan(\frac{w}{U_\infty})$. Eventually, Γ_0 is recovered with the Kutta-Joukowski theorem[42]:

$$\Gamma_0 = \frac{1}{2}c_0 U C_L \tag{72}$$

Polars for the 3D wing were obtained from analytical results, the VPM-LL solver, CFD computations and series of experiments conducted internally at ONERA Lille L1 wind tunnel. This case is set at $U_\infty = 40$ m.s^{-1}. The air density was taken from the ideal gas law at a temperature of 19℃ and at standard atmospheric pressure to respect the experimental conditions. This gives $\nu_{air} = 1,46\cdot10^{-5}$m^2.s^{-1} and a chord-based Reynolds and Mach numbers of $\approx 0.93\times10^6$ and ≈ 0.118. The VPM inputs are $h = \frac{b}{50} = 0.032$ m, $\lambda_\sigma = 2$, $\Delta t = 3h/U_\infty$, $k_{\text{relax}} = 1/3$ and $C_s = 0.15$ with the Vreman viscosity model[28].



(a) Lift coefficient polar.



(b) Drag coefficient polar.

Fig. 12 Analytical, experimental, elsA and VPM elliptical wing polars comparison.

As can be seen in Fig. (12), the VPM matches with the other polars as long as the angle of attack is in the linear range of C_L (between $-13°$ and $13°$). One of the requirement of the LL is indeed to have small angles of attack. All the 3D dynamics of the stall are indeed not taken into account, since the polar database used is 2D. Moreover the radial velocities are not taken into account when updating the velocities induced on the LL. Notwithstanding the stall phenomenon, the lift predicted by the VPM closely agrees with both experiments and analytical results.

A focus is made on the case where the pitch angle is set at $\alpha = 6°$ to confront this result with Prandtl's theory.

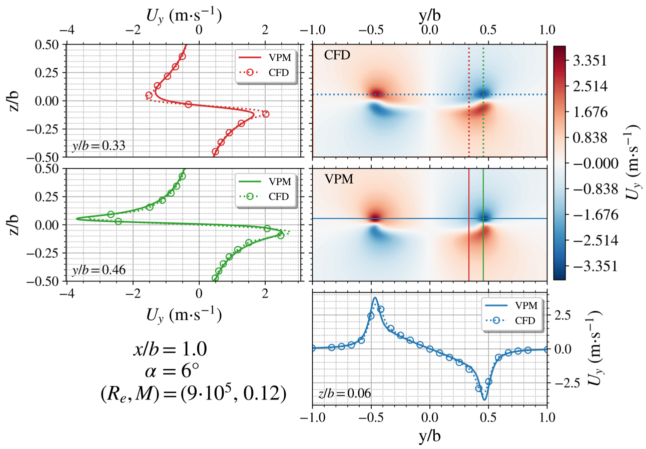


Fig. 13 Induced radial velocity fields in the wake of the elliptical wing.

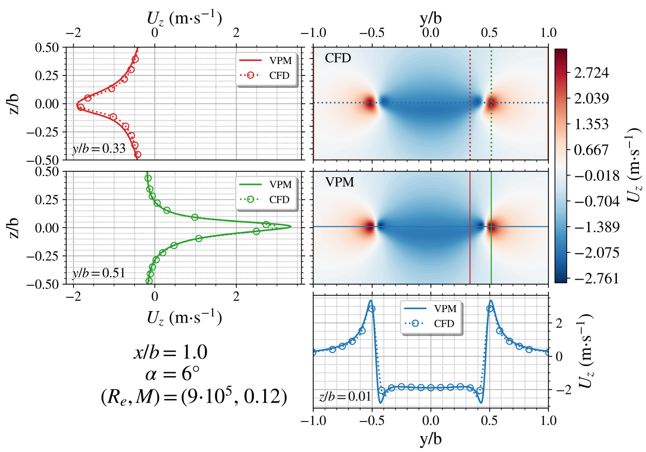


Fig. 14 Induced downwash velocity fields in the wake of the elliptical wing.

Fig. (13 - 15) show a the evolution of the wake on a plane perpendicular, behind the wing. The three graphics (red, green and blue) on the left and bottom sides show the evolution of the velocity or the vorticity along the three coloured segments which cut the planes of the wake. Despite the 2D assumption made by the LL theory, Fig. (13) shows that the radial velocities in the wake match the CFD. As can be seen in Fig. (14) in the blue graph, the induced velocity w (and thus the induced angle α_i) remains constant for the elliptical wing, as expected from Eq. (71). Wingtip vortices play a major role in the loads of the wing and the wake it generates. The effects of those vortices can be seen in the induced velocities at the tips of the wing. The infinitely big aspect ratio assumed by the analytical results is not respected with an actual physical wing. The wingtips vortices and the wake roll-up are represented by the VPM and are the cause of the variations in the induced velocity at the wingtips. The LL itself depends on having a high aspect ratio to avoid such issues, which in our case is rather small ($AR \approx 6$) compared to modern aircrafts ($AR \in [7, ..., 10]$). Another limitations of the LL approach is the lack of induced drag in the wake that should be caused by a 3D bluff body. This means that the wake is not slowed down as much as it should be since the particles do not induce enough axial velocities. Fig. (15) shows the presence of the viscous drag for the CFD with the two counter-rotating vortex sheet at the centre of the wake. However, it also shows that thanks to its Lagrangian framework, the VPM better conserves the wingtips vortices compared to the CFD.

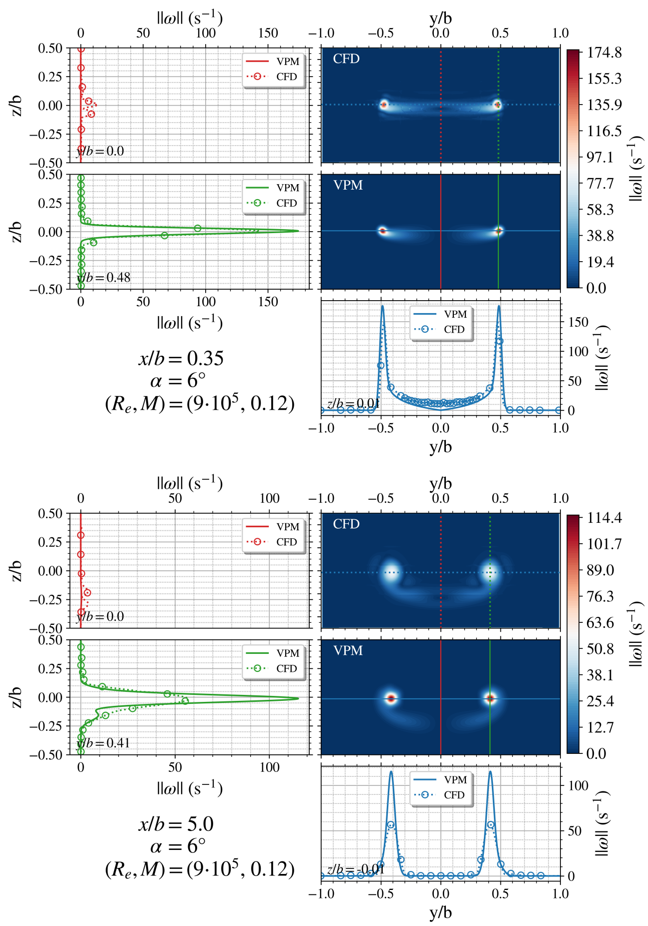


Fig. 15 Vorticity fields in the wake of the elliptical wing.

As can be seen in Fig. (16), the effective angles of attack of the VPM-LL method are in very good agreement with CFD and analytical results ($\alpha_e \approx 4.49 = 6°–1.51°$).

Indeed, notwithstanding the very end of the wingtips, the effective angle of attack remains constant for the whole wing.

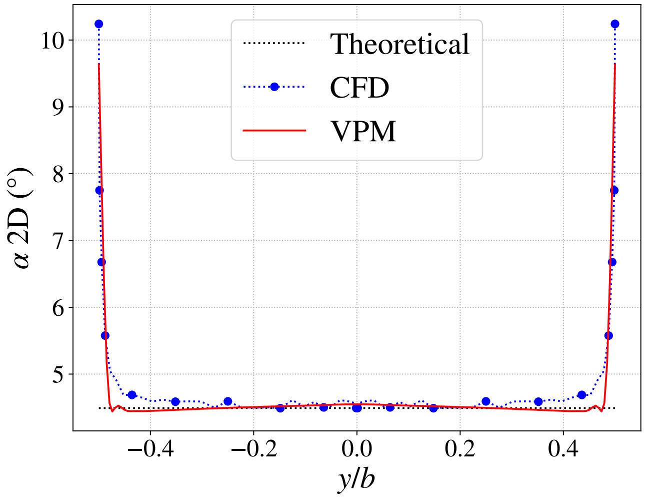


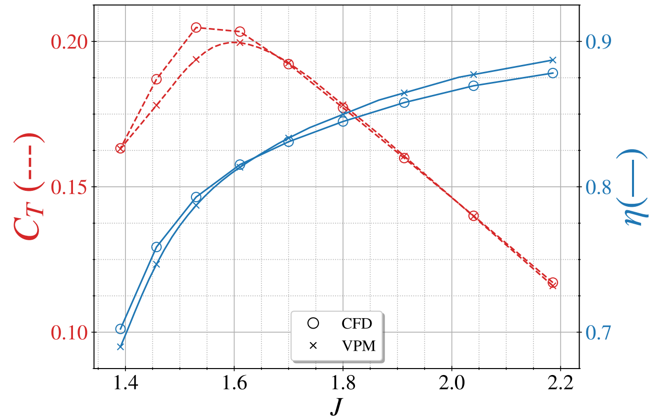
Fig. 16 2D angle of attack for the Elliptical wing.

Overall, Fig. (12-16) demonstrate the capacity of the coupled VPM-LL solver to estimate the loads on the wing and to generate, transport and conserve vortices in the wake.

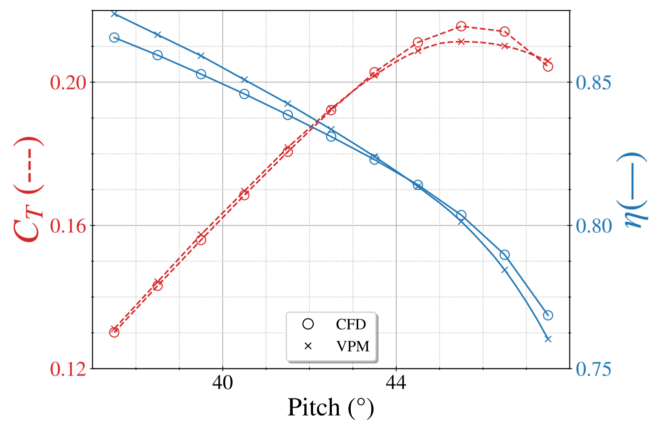
5.3. MRP1 Propeller

Now that fixed wings have been validated, the Multifidelity Rectangular Propeller 1 (MRP1) with 3 blades, constant chord $c = 0.09$ m, linear twist ($35°$ at the root and $-8°$ at the tip), composed of NACA 4412 airfoils, with a diameter of $D = 1.6$ m is used. The case is axisymmetric with a freestream at Mach 0.3. The CFD computations of this propeller are done with a spinner while the VPM simulations represent the hub-less propeller. The VPM inputs are $h = \frac{R-r}{50} = 0.0135$ m, $\lambda_\sigma = 2$, $\Delta t = 5/(6\text{RPM})$, $k_{\text{relax}} = 1/3$ and $C_s = 0.15$ with the Vreman viscosity model[28].

Pitch and RPM Polars of the MRP1 propeller can be seen on Fig. (17a) and Fig. (17b). The results obtained from the VPM agree with those of elsA for moderate Pitch and RPM. Indeed for a Pitch greater than $44°$ or a rotation greater than 2400 RPM ($J \approx 1.6$), the blades start to stall. This limitation comes primarily from the LL. Indeed for sectional Mach numbers greater than 0.7, the stall is missed by the 2D polars. Notwithstanding the stall effect however the VPM gives satisfactory results and manages to give the general trend of the thrust and the efficiency of the propeller.



(a) Advance ratio polars.



(b) Pitch polars.

Fig. 17 elsA and VPM MRP1 polars comparison.

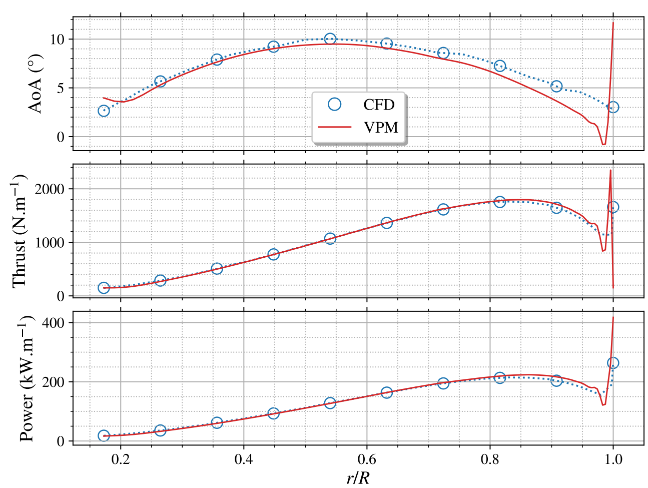


Fig. 18 MRP1 radial angles of attack and loads.

Let us consider the case where the propeller rotates at 2250 RPM with a pitch of $42.5°$. The Fig. (18) shows the effective angles of attack and the repartition of loads of the blades. The thrust and power closely agree with the CFD along the blade up until the tip. In the near-tip region, we found that a fine level resolution of VPM is required in order to obtain accurate predictions in terms of total thrust and power. However this leads to strong oscillations of angle of attack, and thus sectional thrust and power at the tips. A light smoothing of the angles of attack at the tips could be done to avoid such oscillations and obtain smoother loads at the tips. Both methods remain close however, showing the aptitude of the solver to tackle rotary lifting bodies.

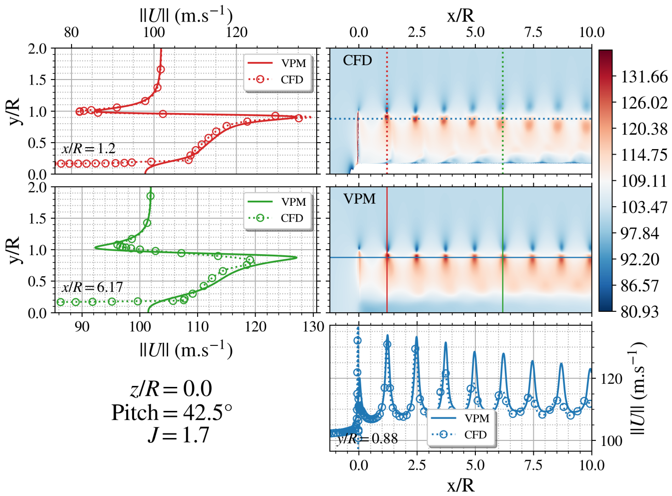


Fig. 19 Velocity fields in the wake of MRP1.

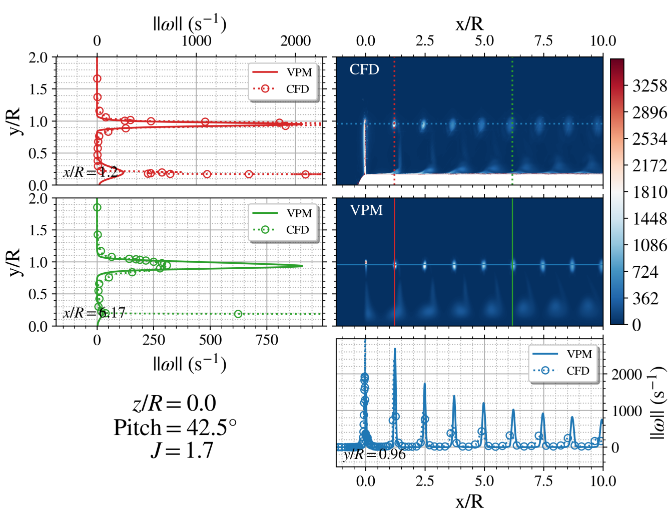


Fig. 20 Vorticity fields in the wake of MRP1.

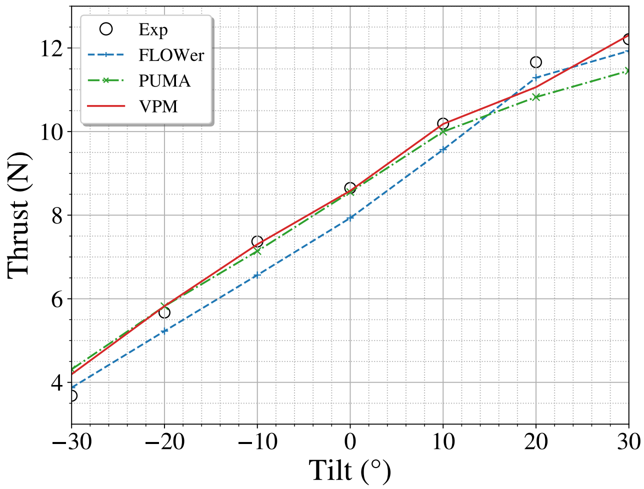
A more thorough analysis is made by comparing the wake of the CFD computation (propeller with spinner) with the VPM. A transversal plane of the top half of the wake is shown for the CFD and the VPM on the right-hand-side of Fig. (19) and Fig. (20). The tip vortices shed in the wake almost overlap with the CFD, showing the capacity of the VPM to properly convect them. The VPM also generates small hub vortices, similar to the ones shed in the CFD. Moreover, the vortex structures generated are better preserved by the VPM as can be shown by the stronger vorticity at their core in Fig. (20). This leads to the vorticies being transported a little further with the VPM than with the CFD.

5.4. KDE Drone

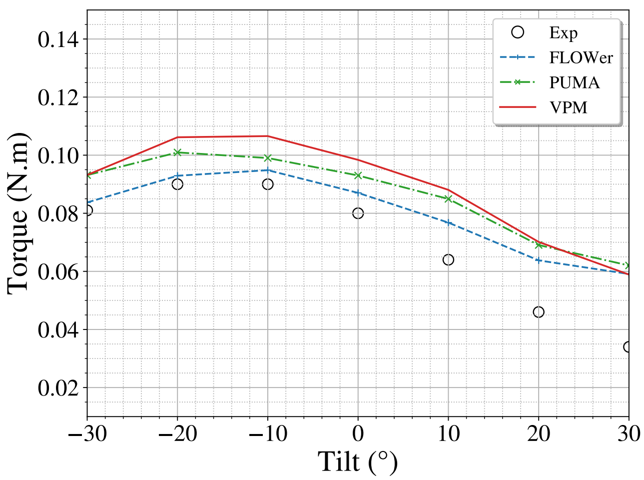
The studies made in [43] and [44] compares several solvers, *inter alia*: FLOWer (a CFD solver) and PUMA (LL and panels); on the KDE quadcopter with 2-bladed 12.5x4.3" rotors. The reader can refer to those studies for the comparison of the other solvers. The rotor undergoes an edgewise flight with a side freestream of 12.9 m.s^{-1} while rotating at 5400 RPM. The rotor is then tilted to the side, mimicking a forward flight (negative tilt) or a brake configuration (positive tilt). The simulation of this drone is tackled with our solver thanks to the polars and geometrical data provided by the DLR. The VPM inputs are $h = \frac{R-r}{30} = 0.00374$ m, $\lambda_\sigma = 2$, $\Delta t = 2.5/(6\text{RPM})$, $k_{\text{relax}} = 1/3$ and $C_s = 0.15$ with the Vreman viscosity model[28].

Let us first consider the case of the isolated rotor[43]. The Fig. (21a) and (21b) shows the thrust and torque of the rotor for different tilt angles. The predictions of thrust of the VPM-LL solver are close to the experiments for most tilt angles. The predictions of PUMA match the VPM-LL for tilt angles below $20°$. This is mainly because for positive tilt angles, the retreating blade of the rotor meets the wake generated by the forwarding blade, creating vortex interactions with important 3D effects. This is especially the case the +20° and +30° tilts. The torque predicted by all three solvers follow the same trend but are above the experimental data. Overall, the integrated loads are close to the predictions of the other solvers and the experiment.

A close-up of the thrust variation is made for tilt values of -10° and +20° for one whole rotation in Fig. (22a) and (22b). It can be seen that for negative tilt, the thrust variations of the VPM and PUMA match. The two methods indeed have similar fidelity and operate with the same 2D polars. For positive tilt, the free motion of the particles are better suited than panels for the important

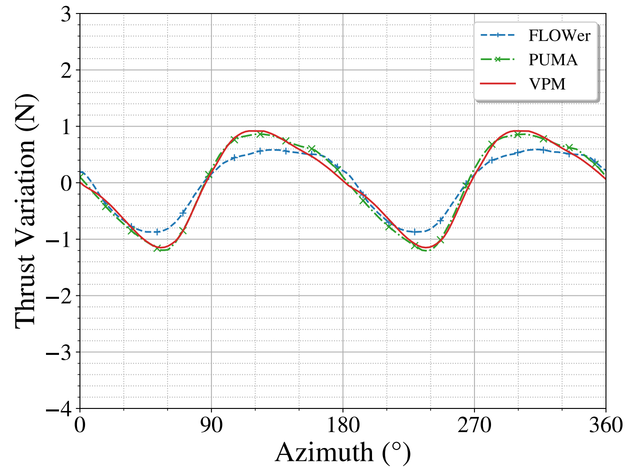


(a) Thrust according to the tilt angle.

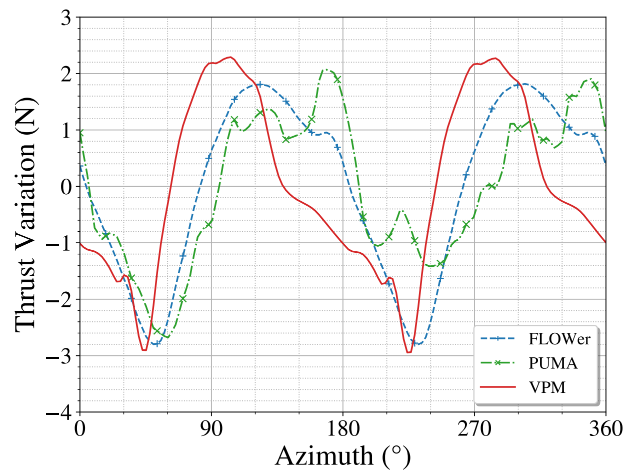


(b) Torque according to the tilt angle.

Fig. 21 Isolated KDE rotor integrated loads.



(a) Thrust variations for -10° tilt according to the azimuth.



(b) Thrust variations for +20° tilt according to the azimuth.

Fig. 22 Isolated KDE rotor thrust variation for one rotation.

deformation of the wake due its interaction with the retreating blade. The thrust variations are periodic and close to the CFD while slightly in advance. The use of unsteady loads, as proposed in [33], might be able to correct this trend.

The case of the whole quadcopter drone is now considered[44]. The rotors are positioned in a square of size d. The right-front, right-rear, left-front and left-rear rotors are respectively numbered 1, 2, 3 and 4. The breaststroke configuration was chosen for this study (positive rotation for rotors 2 and 3 and negative for 1 and 4) with the drone tilted at -10° relatively to the horizontal plane.

Fig. (24) gives the ratio of thrust and torque between the KDE quadrotor and the isolated rotor for different rotor spacing distances d. For $1.2D \leq d$, the thrust and torque ratio of the VPM agrees with FLOWer and PUMA predictions. For small rotor spacing however, the rotor interactions become more important, especially between the front rotors 1 and 3. Those turbulent interactions are slightly downplayed by the VPM that does not properly render these vortex 3D dynamics on the 2D LL. This leads to an increase of thrust for the rear rotors that are less affected by the breaststroke interactions of the front rotors. The case of $d \approx D$, gives the highest breaststroke interactions and because of that the higher efficiency in the front rotors, as showed by all solvers. The wingtip vorticies shed by rotor 1 (resp. 2) directly impact the tips of rotor 2, 3 and 4 (resp. 1, 3 and 4). Those strong

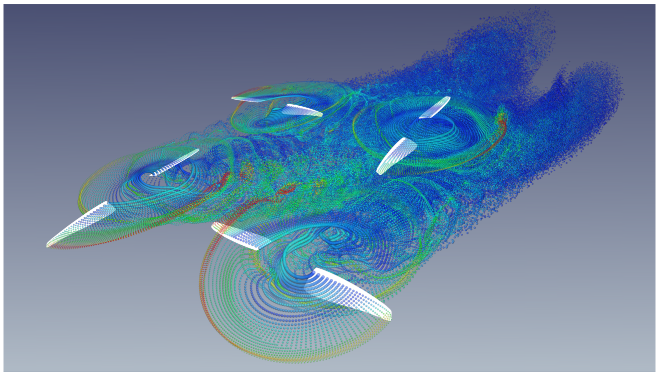
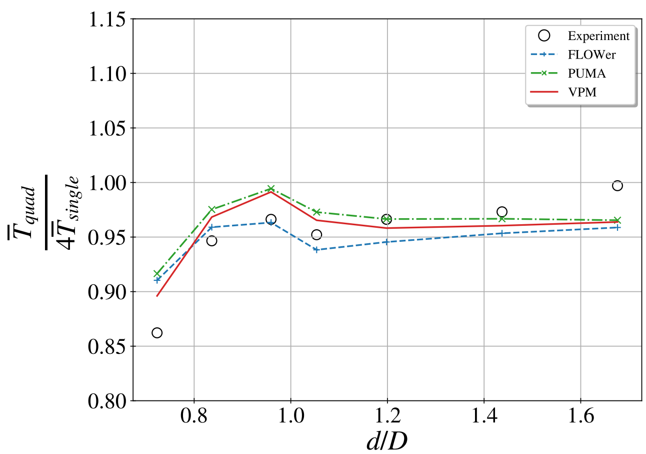
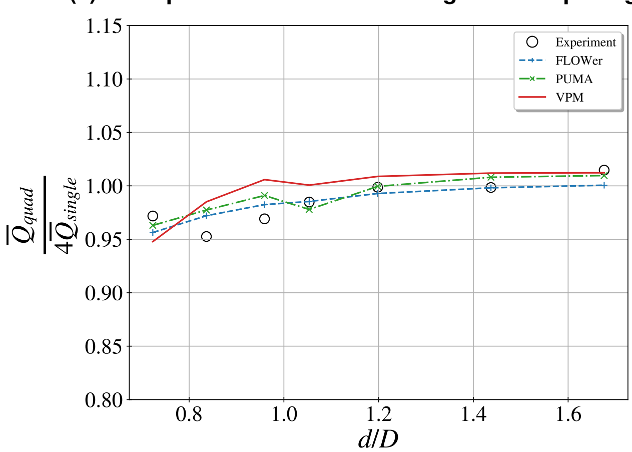


Fig. 23 Full KDE quadrotor for $d = 1.2D$.



(a) KDE quadrotor thrust according to rotor spacing.



(b) KDE quadrotor torque according to rotor spacing.

Fig. 24 KDE quadrotor loads according to rotor spacing.

interactions between the rotors lead to rapidly varying loads. Below $0.96D$ however, the tip vorticies impact the middle of the blades of the other rotors instead of the tips. While the tip vorticies still impact the rest of the blades rotors, those configurations are less turbulent but also offer less efficiency.

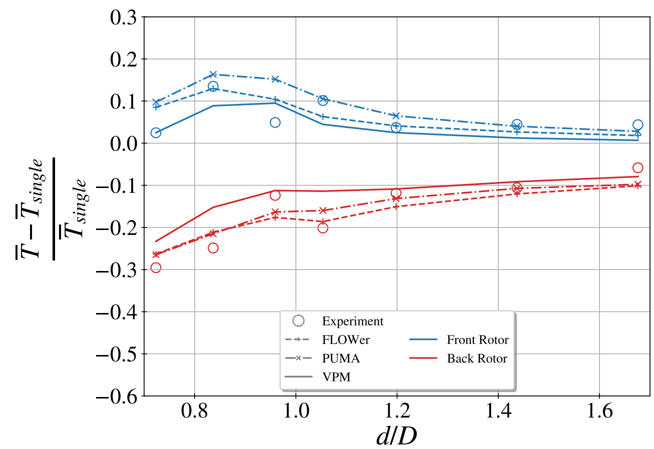


Fig. 25 KDE quadrotor loads for the front and back rotors according to rotor spacing.

Fig. (26) shows the normal velocities on a plane located at 75mm below the drone. The velocity field of the VPM was time-averaged for three complete rotor rotations to obtain a statistically steady field to be compared with the RANS and experimental fields. The comparison of the experiment, FLOWer and the VPM give similar order of magnitude of the downwash velocity and similar wake propagation. The main difference resides at around $(x, y) = (0.1\text{m}, \pm 0.1\text{m})$ where the VPM fails to properly stretch and transport far enough the wake generated by the advancing blade. The back rotors undergo less downwash and thus have a higher thrust. The upwash velocity between the front rotors estimated by the VPM is similar to the one measured by the experiment while being a little less than the one estimated by FLOWer. Those combined effects might explain the increase of efficiency of the back rotors estimated by the VPM.

6. CONCLUSION

The Vortex Particle Method (VPM) solver was successfully implemented. The acceleration of the VPM using the FMM approach proved very satisfactory. The linear scalability of the code with respect to the number of particles and the number of computational threads, makes it suitable for computing complex flows with a relatively large number of particles. The VPM was first

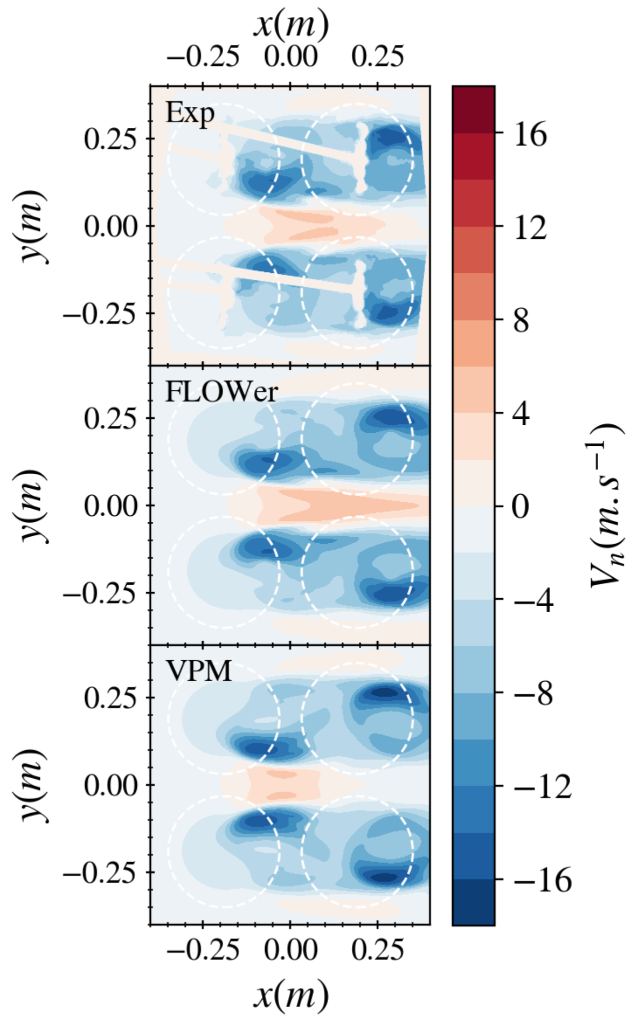


Fig. 26 Downwash velocity fields comparison of the KDE quadrotor for $d = 1.2D$ between the experiment, FLOWer and the VPM.

validated with a Lamb-Oseen viscous vortex and 3D vortex rings to investigate different viscous methods and validate the solver. The VPM was able to reproduce the evolution and viscous decay of those vortex structures. The Diffusion Velocity Method was eventually chosen for the solver because of its capacity to adapt the shape of the domain to match the growth of the vortex structures due to their viscous spread.

The comparison with respect to theoretical, numerical and experimental results provided satisfactory agreement and allowed to validate the coupled solver. The VPM proved capable of properly predicting the loads on a lifting surface and to generate, transport and conserve vortex structures in the wake. This study shows that the new VPM-LL solver is well suited for fast computations of lifting surfaces and viscous vortex structures interactions in a robust and reasonably accurate manner. This is of practical interest for mid-fidelity simulations of aircraft, propellers and rotorcraft.

Further validation of increasingly complex configurations would be required in order to assess the code's applicability to practical use scenarios. The main issues of the solver primarily come from the limitations linked to the Lifting-Line solver, especially its 2D representation of the flow. The difficulty behind the coupling of the latter method with the VPM came from the initialisation of the vortex particles in the wake to properly generate the vortex structures shed by the lifting surfaces. Other methods might be investigated to more accurately represent 3D lifting bodies and better initialise the vortex structures they generate. An Eulerian-Lagrangian method, which would bind a high-fidelity CFD solver with the VPM, might be a candidate for such an application.

Very large computations involving high number of particles (several million) are beyond the scope of this work. In the authors opinion, it is unlikely that very large computations can be efficiently accomplished with current OpenMP parallelism framework. MPI parallelisation or GPU implementation would be required for better scalability.

Further investigation of the theoretical background regarding the control of the divergence through the Enstrophy and the divergence-free Enstrophy and the dynamic modification of the Smagorinski constant are also needed to ascertain the validity of both methods.

Author contact: johan.valentin@onera.fr

ACKNOWLEDGEMENTS

I wish to thank first and foremost the Direction Générale de l'Aviation Civile (DGAC) and l'Office National d'Études et de Recherches Aérospatiales (ONERA) for funding this Ph.D doctorate through the European project INPRO. I also want to thank the DLR for providing their experimental data and the polars of the KDE drone which made possible the validation of the coupled VPM solver.

Last but not least I wish to thank Élie Rivoalen, Gregory Pinon and Paul Mycek for their guidance feedback on the VPM and its intricacies.

APPENDIX

A. Enstrophy Computation

The calculation of the real and non divergence-free Enstrophy, noted $\mathcal{E}$ and $\mathcal{E}_f$, are detailed here. Since the flow is not necessarily divergence-free ($\overrightarrow{\omega} \neq \overrightarrow{\nabla} \times \overrightarrow{u}$), this leads us to consider the curl of the velocity rather than the vorticity itself, obtained from Eq. (13):

$$\overrightarrow{\nabla} \times \overrightarrow{\widetilde{u}}(\overrightarrow{x}, t) = \sum_{j}^{N_p} \left[\zeta_\sigma(r)\overrightarrow{\alpha_j} + \overrightarrow{\nabla} \left(\overrightarrow{\alpha_j} \cdot \overrightarrow{\nabla} \chi_\sigma(r) \right) \right] \quad (73)$$

where $\chi_\sigma(\rho) = -\int_0^\rho g(t)/(\sigma t^2)dt$. Its non regularised version is the Green function $G(r) = 1/(4\pi r)$. Eq. (34) diverges when using the singular vorticity in Eq. (2) and can not be written with basic mathematical function using the filtered vorticity in Eq. (10).The complexity of integrating Eq. (34) with both regularised velocities forces us to regularise only one and keep the other singular as proposed in [6]:

$$\begin{aligned}
\widetilde{\mathcal{E}} &= \int \overrightarrow{\widetilde{\omega}} \cdot \overrightarrow{\omega} dx \\
&= \int \left(\overrightarrow{\nabla} \times \overrightarrow{\widetilde{u}} \right) \cdot \left(\overrightarrow{\nabla} \times \overrightarrow{u} \right) dx \\
&= \int \Bigg(\sum_{p}^{N_p} \underbrace{\zeta_\sigma(||\overrightarrow{x} - \overrightarrow{x_p}||)}_{\zeta_\sigma^p} \overrightarrow{\alpha_p} \\
&\qquad + \overrightarrow{\nabla}\Big(\overrightarrow{\alpha_p} \cdot \overrightarrow{\nabla} \underbrace{\chi_\sigma(||\overrightarrow{x} - \overrightarrow{x_p}||)}_{\chi_\sigma^p} \Big) \Bigg) \cdot \Bigg(\sum_{q}^{N_p} \delta(||\overrightarrow{x} - \overrightarrow{x_q}||)\overrightarrow{\alpha_q} \\
&\qquad + \overrightarrow{\nabla}\Big(\overrightarrow{\alpha_q} \cdot \overrightarrow{\nabla} \underbrace{G(||\overrightarrow{x} - \overrightarrow{x_q}||)}_{G^q} \Big) \Bigg) dx \\
&= \sum_{p,q}^{N_p} \Bigg[\zeta_\sigma(||\overrightarrow{x_p} - \overrightarrow{x_q}||)\overrightarrow{\alpha_p} \cdot \overrightarrow{\alpha_q} \\
&\qquad + \overrightarrow{\alpha_q} \cdot \overrightarrow{\nabla} \left(\overrightarrow{\alpha_p} \cdot \overrightarrow{\nabla} \chi_\sigma(||\overrightarrow{x_p} - \overrightarrow{x_q}||) \right) \\
&\qquad + \int \Big[\zeta_\sigma^p \overrightarrow{\alpha_p} \cdot \overrightarrow{\nabla} \left(\overrightarrow{\alpha_q} \cdot \overrightarrow{\nabla} G^q \right) \\
&\qquad + \overrightarrow{\nabla} \left(\overrightarrow{\alpha_p} \cdot \overrightarrow{\nabla} \chi_\sigma^p \right) \cdot \overrightarrow{\nabla} \left(\overrightarrow{\alpha_q} \cdot \overrightarrow{\nabla} G^q \right) \Big] dx \Bigg] \\
&= \sum_{p}^{N_p} \overrightarrow{\alpha_p} \cdot \overrightarrow{\nabla} \times \overrightarrow{\widetilde{u}}(\overrightarrow{x_p}, t) + \underbrace{\sum_{p,q}^{N_p} \int \left[\zeta_\sigma^p \overrightarrow{\alpha_p} \cdot \overrightarrow{\nabla} \left(\overrightarrow{\alpha_q} \cdot \overrightarrow{\nabla} G^q \right) \right] dx}_{I} \\
&\qquad + \underbrace{\sum_{p,q}^{N_p} \int \left[\overrightarrow{\nabla} \left(\overrightarrow{\alpha_p} \cdot \overrightarrow{\nabla} \chi_\sigma^p \right) \cdot \overrightarrow{\nabla} \left(\overrightarrow{\alpha_q} \cdot \overrightarrow{\nabla} G^q \right) \right] dx}_{J}
\end{aligned}$$

Let us prove that $I + J = 0$. Since $\zeta_\sigma()$ goes to zero at infinity, integrating I by parts comes easy. The symmetry of $\zeta_\sigma()$, $\chi_\sigma()$ and $G()$ also allows us to use $\frac{\partial}{\partial x_i}\left(f(||\overrightarrow{x} - \overrightarrow{x_p}||) \right) = -\frac{\partial}{\partial x_i^p}\left(f(||\overrightarrow{x} - \overrightarrow{x_p}||) \right)$:

$$\begin{aligned}
I + J &= \sum_{p,q}^{N_p} \int \left[\zeta_\sigma^p \alpha_i^p \frac{\partial}{\partial x_i} \left(\alpha_j^q \frac{\partial}{\partial x_j} G^q \right) \right] dx \\
&\qquad + \int \left[\frac{\partial}{\partial x_i} \left(\alpha_k^p \frac{\partial}{\partial x_k} \chi_\sigma^p \right) \frac{\partial}{\partial x_i} \left(\alpha_m^q \frac{\partial}{\partial x_m} G^q \right) \right] dx \\
&= \sum_{p,q}^{N_p} \alpha_i^p \alpha_j^q \int \left[\zeta_\sigma^p \frac{\partial}{\partial x_i} \left(\frac{\partial}{\partial x_j} G^q \right) \right] dx \\
&\qquad + \alpha_k^p \alpha_m^q \int \left[\frac{\partial}{\partial x_i} \left(\frac{\partial}{\partial x_k} \chi_\sigma^p \right) \frac{\partial}{\partial x_i} \left(\frac{\partial}{\partial x_m} G^q \right) \right] dx \\
&= \sum_{p,q}^{N_p} -\alpha_i^p \alpha_j^q \int \left[\frac{\partial}{\partial x_i^p} \left(\zeta_\sigma^p \right) \frac{\partial}{\partial x_j^q} \left(G^q \right) \right] dx \\
&\qquad + \alpha_k^p \alpha_m^q \int \left[\frac{\partial}{\partial x_i^p} \left(\frac{\partial}{\partial x_k^p} \chi_\sigma^p \right) \frac{\partial}{\partial x_i^q} \left(\frac{\partial}{\partial x_m^q} G^q \right) \right] dx \\
&= \sum_{p,q}^{N_p} -\alpha_i^p \alpha_j^q \frac{\partial^2}{\partial x_i^p x_j^q} \underbrace{\int \zeta_\sigma^p G^q dx}_{I_0} \\
&\qquad + \alpha_k^p \alpha_m^q \frac{\partial^4}{\partial x_i^p x_k^p x_i^q x_m^q} \underbrace{\int \chi_\sigma^p G^q dx}_{J_0}
\end{aligned}$$

The integrations of I_0 and J_0 are carried out in spherical coordinates $(z,\ \theta,\ \psi)$, centred around $\overrightarrow{x_p}$ so that $z = ||\overrightarrow{x} - \overrightarrow{x_p}||$ and $r = ||\overrightarrow{x_p} - \overrightarrow{x_q}||$ where the law of cosines gives $||\overrightarrow{x} - \overrightarrow{x_q}||^2 = ||\overrightarrow{x} - \overrightarrow{x_p}||^2 - 2||\overrightarrow{x} - \overrightarrow{x_p}||||\overrightarrow{x_p} - \overrightarrow{x_q}|| \cos\theta + ||\overrightarrow{x_p} - \overrightarrow{x_q}||^2 = z^2 - 2zr\cos\theta + r^2$. Let us focus on I_0:

$$\begin{aligned}
I_0 &= \int \zeta_\sigma \left(||\overrightarrow{x} - \overrightarrow{x_p}|| \right) G \left(||\overrightarrow{x} - \overrightarrow{x_q}|| \right) z dz z d\theta \sin\theta d\psi \\
&= \int_{\psi=0}^{2\pi} \frac{d\psi}{4\pi} \int_{z=0}^{+\infty} z\zeta_\sigma(z) \left(\int_{\theta=0}^{\pi} \frac{z\sin\theta d\theta}{\sqrt{z^2 - 2zr\cos\theta + r^2}} \right) dz \\
&= \frac{1}{2} \int_{z=0}^{+\infty} z\zeta_\sigma(z) \left(\int_{\theta=0}^{\pi} \frac{z\sin\theta d\theta}{\sqrt{z^2 - 2zr\cos\theta + r^2}} \right) dz
\end{aligned}$$

Let us change variables by taking $\cos\theta = \frac{r^2+z^2-t^2}{2rz}$ so that $\sin\theta d\theta = \frac{tdt}{rz}$:

$$\begin{aligned} I_0 &= \frac{1}{2}\int_{z=0}^{+\infty} z\zeta_\sigma(z)\left(\int_{t=\sqrt{z^2-2rz+r^2}}^{\sqrt{z^2+2rz+r^2}} \frac{z}{|t|}\frac{tdt}{rz}\right)dz \\ &= \frac{1}{2}\int_{z=0}^{+\infty} z\zeta_\sigma(z)\left(|\frac{z}{r}+1| - |\frac{z}{r}-1|\right)dz \end{aligned}$$

The integration of J_0 is done likewise:

$$J_0 = \frac{1}{2}\int_{z=0}^{+\infty} z\chi_\sigma(z)\left(|\frac{z}{r}+1| - |\frac{z}{r}-1|\right)dz$$

However:

$$|\frac{z}{r}+1| - |\frac{z}{r}-1| = \begin{cases} 2z/r & 0 \le z/r \le 1 \\ 2 & 1 \le z/r \end{cases}$$

It now appears that J_0 does not converge since $\lim_{z\to\infty}\chi_\sigma(z) \neq 0$. However only its derivatives regarding to $\rho = r/\sigma$ are of interest. This means that another integral, independent of ρ, can be subtracted to J_0 to make it converge. Let us consider the integral J_1 so that $\frac{\partial J_0}{\partial\rho} = \frac{\partial J_1}{\partial\rho}$:

$$\begin{aligned} J_1 &= J_0 - \int_{z=0}^{+\infty} z\chi_\sigma(z)dz \\ &= \frac{1}{2}\int_{z=0}^{r} z\chi_\sigma(z)\left(2\frac{z}{r}-2\right)dz \\ &= \frac{\sigma}{\rho}\underbrace{\int_{t=0}^{\rho} t^2\sigma\chi_\sigma(t\sigma)dt}_{J_2} - \sigma\int_{t=0}^{\rho} t\sigma\chi_\sigma(t\sigma)dt \end{aligned}$$

The same can be done to I_0 to ease its computation:

$$\begin{aligned} I_1 &= I_0 - \int_{z=0}^{+\infty} z\zeta_\sigma(z)dz \\ &= \frac{\sigma^2}{\rho}\underbrace{\int_{t=0}^{\rho} t^2\zeta_\sigma(t\sigma)dt}_{g(\rho)/\sigma^3} - \sigma^2\int_{t=0}^{\rho} t\zeta_\sigma(t\sigma)dt \end{aligned}$$

Let us now calculate the derivatives of I_1 and J_1 according to ρ:

$$\frac{\partial I_0}{\partial\rho} = \frac{\partial I_1}{\partial\rho} = -\frac{g(\rho)}{\rho^2\sigma} + \frac{1}{\rho\sigma}\rho^2\sigma^3\zeta_\sigma(r) - \sigma^2\rho\zeta_\sigma(r) = -\frac{g(\rho)}{\rho^2\sigma}$$

$$\frac{\partial^2 I_0}{\partial\rho^2} = \frac{2g(\rho)}{\rho^3\sigma} - \frac{1}{\rho^2\sigma}\rho^2\sigma^3\zeta_\sigma(r) = \frac{1}{\sigma}\left(\frac{2g(\rho)}{\rho^3} - \sigma^3\zeta_\sigma(r)\right)$$

And:

$$\frac{\partial J_0}{\partial\rho} = \frac{\partial J_1}{\partial\rho} = -\frac{\sigma}{\rho^2}J_2 + \sigma^2\rho\chi_\sigma(r) - \sigma^2\rho\chi_\sigma(r) = -\frac{\sigma}{\rho^2}J_2$$

$$\frac{\partial^2 J_0}{\partial\rho^2} = \frac{2\sigma}{\rho^3}J_2 - \sigma^2\chi_\sigma(r)$$

$$\frac{\partial^3 J_0}{\partial\rho^3} = -\frac{6\sigma}{\rho^4}J_2 + \frac{2\sigma^2}{\rho}\chi_\sigma(r) + \sigma\frac{g(\rho)}{\rho^2}$$

$$\frac{\partial^4 J_0}{\partial\rho^4} = \frac{24\sigma}{\rho^5}J_2 - \frac{8\sigma^2}{\rho^2}\chi_\sigma(r) - 4\sigma\frac{g(\rho)}{\rho^3} + \sigma^4\zeta_\sigma(r)$$

Let us write the total derivatives of both integrals:

$$\begin{aligned} \frac{\partial^2 I_0}{\partial x_i^p \partial x_j^q} &= \frac{\partial}{\partial x_i^p}\left(\frac{\partial\rho}{\partial x_j^q}\frac{\partial I_0}{\partial\rho}\right) \\ &= \frac{\partial}{\partial x_i^p}\left(\frac{-r_j}{r\sigma}\frac{\partial I_0}{\partial\rho}\right) \\ &= \left(\frac{-\delta_{ij}}{r\sigma} + \frac{r_i r_j}{r^3\sigma}\right)\frac{\partial I_0}{\partial\rho} - \frac{r_j}{r\sigma}\frac{\partial\rho}{\partial x_i^p}\frac{\partial^2 I_0}{\partial\rho^2} \\ &= -\frac{1}{r\sigma}\frac{\partial I_0}{\partial\rho}\delta_{ij} + \left(\frac{1}{r\sigma}\frac{\partial I_0}{\partial\rho} - \frac{1}{\sigma^2}\frac{\partial^2 I_0}{\partial\rho^2}\right)\frac{r_i r_j}{r^2} \end{aligned}$$

Hence:

$$\begin{aligned} \alpha_i^p\alpha_j^q\frac{\partial^2 I_0}{\partial x_i^p x_j^q} &= -\frac{1}{\rho\sigma^2}\frac{\partial I_0}{\partial\rho}\overrightarrow{\alpha_p}\cdot\overrightarrow{\alpha_q} \\ &\quad + \left(\frac{1}{\rho\sigma^2}\frac{\partial I_0}{\partial\rho} - \frac{1}{\sigma^2}\frac{\partial^2 I_0}{\partial\rho^2}\right)\frac{\left(\overrightarrow{\alpha_p}\cdot\overrightarrow{r}\right)\left(\overrightarrow{\alpha_q}\cdot\overrightarrow{r}\right)}{r^2} \\ &= \frac{1}{\sigma^3}\left(\frac{g(\rho)}{\rho^3}\overrightarrow{\alpha_p}\cdot\overrightarrow{\alpha_q}\right. \\ &\quad \left. + \left(\sigma^3\zeta_\sigma(r_{pq}) - \frac{3g(\rho)}{\rho^3}\right)\frac{\left(\overrightarrow{\alpha_p}\cdot\overrightarrow{r}\right)\left(\overrightarrow{\alpha_q}\cdot\overrightarrow{r}\right)}{r^2}\right) \end{aligned}$$

And:

$$\begin{aligned} \frac{\partial^4 J_0}{\partial x_i^p \partial x_k^p \partial x_i^q \partial x_m^q} &= \frac{\partial^2}{\partial x_k^p \partial x_m^q}\left(\frac{\partial^2 J_0}{\partial x_i^p \partial x_i^q}\right) \\ &= \frac{\partial^2}{\partial x_k^p \partial x_m^q}\left(\frac{\partial}{\partial x_i^p}\left(-\frac{r_i}{r\sigma}\frac{\partial J_0}{\partial\rho}\right)\right) \\ &= \frac{\partial^2}{\partial x_k^p \partial x_m^q}\left(\left(-\frac{\delta_{ii}}{r\sigma} + \frac{r_i^2}{r^3\sigma}\right)\frac{\partial J_0}{\partial\rho} - \frac{r_i^2}{r^2\sigma^2}\frac{\partial^2 J_0}{\partial\rho^2}\right) \\ &= \frac{\partial^2}{\partial x_k^p \partial x_m^q}\left(\left(-\frac{3}{r\sigma} + \frac{r^2}{r^3\sigma}\right)\frac{\partial J_0}{\partial\rho} - \frac{r^2}{r^2\sigma^2}\frac{\partial^2 J_0}{\partial\rho^2}\right) \\ &= \frac{\partial^2}{\partial x_k^p \partial x_m^q}\left(-\frac{2}{r\sigma}\frac{\partial J_0}{\partial\rho} - \frac{1}{\sigma^2}\frac{\partial^2 J_0}{\partial\rho^2}\right) \\ &= \frac{\partial}{\partial x_k^p}\left(-2\frac{r_m}{r^3\sigma}\frac{\partial J_0}{\partial\rho} + 2\frac{r_m}{r^2\sigma^2}\frac{\partial^2 J_0}{\partial\rho^2} + \frac{r_m}{r\sigma^3}\frac{\partial^3 J_0}{\partial\rho^3}\right) \end{aligned}$$

$$= \left(-2\frac{\delta_{km}}{r^3\sigma} + 6\frac{r_k r_m}{r^5\sigma}\right)\frac{\partial J_0}{\partial\rho} - 2\frac{r_m}{r^3\sigma}\frac{\partial\rho}{\partial x_k^p}\frac{\partial^2 J_0}{\partial\rho^2}$$
$$+ \left(2\frac{\delta_{km}}{r^2\sigma^2} - 4\frac{r_k r_m}{r^4\sigma^2}\right)\frac{\partial^2 J_0}{\partial\rho^2} + 2\frac{r_m}{r^2\sigma^2}\frac{\partial\rho}{\partial x_k^p}\frac{\partial^3 J_0}{\partial\rho^3}$$
$$+ \left(\frac{\delta_{km}}{r\sigma^3} - \frac{r_k r_m}{r^3\sigma^3}\right)\frac{\partial^3 J_0}{\partial\rho^3} + \frac{r_m}{r\sigma^3}\frac{\partial\rho}{\partial x_k^p}\frac{\partial^4 J_0}{\partial\rho^4}$$
$$= \frac{1}{\sigma^3}\left[\left(-\frac{2}{\rho^3\sigma}\frac{\partial J_0}{\partial\rho} + \frac{2}{\rho^2\sigma}\frac{\partial^2 J_0}{\partial\rho^2} + \frac{1}{\rho\sigma}\frac{\partial^3 J_0}{\partial\rho^3}\right)\delta_{km} + \left(\frac{6}{\rho^3\sigma}\frac{\partial J_0}{\partial\rho} - \frac{6}{\rho^2\sigma}\frac{\partial^2 J_0}{\partial\rho^2} + \frac{1}{\rho\sigma}\frac{\partial^3 J_0}{\partial\rho^3} + \frac{1}{\sigma}\frac{\partial^4 J_0}{\partial\rho^4}\right)\frac{r_k r_m}{r^2}\right]$$

Hence

$$\alpha_k^p \alpha_m^q \frac{\partial^4 J_0}{\partial x_i^p x_k^p x_i^q x_m^q} =$$
$$\frac{1}{\sigma^3}\left[\left(-\frac{2}{\rho^3\sigma}\frac{\partial J_0}{\partial\rho} + \frac{2}{\rho^2\sigma}\frac{\partial^2 J_0}{\partial\rho^2} + \frac{1}{\rho\sigma}\frac{\partial^3 J_0}{\partial\rho^3}\right)\overrightarrow{\alpha_p}\cdot\overrightarrow{\alpha_q} + \left(\frac{6}{\rho^3\sigma}\frac{\partial J_0}{\partial\rho} - \frac{6}{\rho^2\sigma}\frac{\partial^2 J_0}{\partial\rho^2} + \frac{1}{\rho\sigma}\frac{\partial^3 J_0}{\partial\rho^3} + \frac{1}{\sigma}\frac{\partial^4 J_0}{\partial\rho^4}\right)\frac{\left(\overrightarrow{\alpha_p}\cdot\overrightarrow{r}\right)\left(\overrightarrow{\alpha_q}\cdot\overrightarrow{r}\right)}{r^2}\right]$$
$$= \frac{1}{\sigma^3}\left[\frac{g(\rho)}{\rho^3}\overrightarrow{\alpha_p}\cdot\overrightarrow{\alpha_q} + \left(\sigma^3\zeta_\sigma(r_{pq}) - 3\frac{g(\rho)}{\rho^3}\right)\frac{\left(\overrightarrow{\alpha_p}\cdot\overrightarrow{r}\right)\left(\overrightarrow{\alpha_q}\cdot\overrightarrow{r}\right)}{r^2}\right]$$
$$= \alpha_i^p \alpha_j^q \frac{\partial^2 I_0}{\partial x_i^p x_j^q}$$

So that eventually:

$$I + J = \sum_{p,q}^{N_p} -\alpha_i^p \alpha_j^q \frac{\partial^2 I_0}{\partial x_i^p x_j^q} + \alpha_k^p \alpha_m^q \frac{\partial^4 J_0}{\partial x_i^p x_k^p x_i^q x_m^q} = 0$$

The Enstrophy becomes:

$$\widetilde{\mathcal{E}} = \sum_{p,q}\zeta_\sigma(r_{pq})\overrightarrow{\alpha_p}\cdot\overrightarrow{\alpha_q} + \overrightarrow{\alpha_q}\cdot\overrightarrow{\nabla}\left(\overrightarrow{\alpha_p}\cdot\overrightarrow{\nabla}\chi_\sigma(\rho)\right)$$
$$= \sum_{p,q}\zeta_\sigma(r_{pq})\overrightarrow{\alpha_p}\cdot\overrightarrow{\alpha_q} + \overrightarrow{\alpha_q}\cdot\overrightarrow{\nabla}\left(\frac{\partial\chi_\sigma(\rho)}{\partial\rho}\overrightarrow{\alpha_p}\cdot\overrightarrow{\nabla}\rho\right)$$
$$= \sum_{p,q}\zeta_\sigma(r_{pq})\overrightarrow{\alpha_p}\cdot\overrightarrow{\alpha_q} - \overrightarrow{\alpha_q}\cdot\overrightarrow{\nabla}\left(\frac{1}{\rho^2\sigma}g(\rho)\overrightarrow{\alpha_p}\cdot\left(\frac{1}{\rho\sigma^2}\overrightarrow{r}\right)\right)$$
$$= \sum_{p,q}\zeta_\sigma(r_{pq})\overrightarrow{\alpha_p}\cdot\overrightarrow{\alpha_q} - \overrightarrow{\alpha_q}\cdot\left(\overrightarrow{\nabla}\left(\frac{g(\rho)}{r3}\right)\left(\overrightarrow{\alpha_p}\cdot\overrightarrow{r}\right) + \frac{g(\rho)}{r3}\overrightarrow{\nabla}\left(\overrightarrow{\alpha_p}\cdot\overrightarrow{r}\right)\right)$$

where:

$$\overrightarrow{\nabla}\left(\frac{g(\rho)}{r^3}\right) = \frac{1}{\sigma^3}\frac{\partial}{\partial\rho}\left(\frac{g(\rho)}{\rho^3}\right)\overrightarrow{\nabla}\rho$$
$$= \frac{1}{r^2}\left(\zeta_\sigma(r_{pq}) - 3\frac{g(\rho)}{r^3}\right)\overrightarrow{r}$$

The general formulation of the Enstrophy, regardless of the smoothing function is:

$$\widetilde{\mathcal{E}} = \sum_p \overrightarrow{\alpha_p}\cdot\overrightarrow{\nabla}\times\overrightarrow{\widetilde{u}}(\overrightarrow{x_p},t)$$
$$= \sum_{p,q}\left(\zeta_\sigma(r_{pq}) - \frac{g(\rho)}{r3}\right)\overrightarrow{\alpha_p}\cdot\overrightarrow{\alpha_q} + \left(3\frac{g(\rho)}{r^3} - \zeta_\sigma(r_{pq})\right)\frac{\left(\overrightarrow{\alpha_q}\cdot\overrightarrow{r}\right)\left(\overrightarrow{\alpha_p}\cdot\overrightarrow{r}\right)}{r^2}$$

When the flow is supposed divergence-free, $i.e.$ when $\overrightarrow{\widetilde{\omega}} = \overrightarrow{\nabla}\times\overrightarrow{\widetilde{u}}$, the Enstrophy can be written as:

$$\widetilde{\mathcal{E}}_f = \sum_p \overrightarrow{\alpha_p}\cdot\overrightarrow{\widetilde{\omega}}(\overrightarrow{x_p},t) = \sum_{p,q}\zeta_\sigma(r_{pq})\overrightarrow{\alpha_p}\cdot\overrightarrow{\alpha_q}$$

The contribution to the Enstrophy from neighbouring particles can also be computed. In the case of the Gaussian filter and for small ρ, one has:

$$\frac{g(\rho)}{\rho^3} = \frac{1}{3}\frac{1}{(2\pi)^{3/2}} + O\left(\rho^2\right)$$

And:

$$\sigma^3\zeta_\sigma(r_{pq}) = \frac{1}{(2\pi)^{3/2}} + O\left(\rho^2\right)$$

This approximation of the Enstrophy for the Gaussian filter becomes:

$$\begin{cases} \widetilde{\mathcal{E}}_0 = \dfrac{2||\overrightarrow{\alpha}||^2}{3(2\pi)^{3/2}\sigma^3} \\ \widetilde{\mathcal{E}}_{f_0} = \dfrac{||\overrightarrow{\alpha}||^2}{(2\pi)^{3/2}\sigma^3} \end{cases}$$

B. Kinetic Energy Loss

Let us start by filtering the incompressible Navier-Stokes equation:

$$\widetilde{\frac{\partial\overrightarrow{U}}{\partial t} + \left(\overrightarrow{U}\cdot\overrightarrow{\nabla}\right)\overrightarrow{U}} = \widetilde{-\frac{1}{\rho_\infty}\overrightarrow{\nabla}p + \nu\overrightarrow{\Delta}\overrightarrow{U}}$$
$$\frac{\partial\overrightarrow{\widetilde{U}}}{\partial t} + \left(\overrightarrow{\widetilde{U}}\cdot\overrightarrow{\nabla}\right)\overrightarrow{\widetilde{U}} = -\frac{1}{\rho_\infty}\overrightarrow{\nabla}\widetilde{p} + \nu\overrightarrow{\Delta}\overrightarrow{\widetilde{U}}$$

$$
\underbrace{-\left(\widetilde{\left(\vec{U}\cdot\vec{\nabla}\right)\vec{U}}-\left(\widetilde{\vec{U}}\cdot\vec{\nabla}\right)\widetilde{\vec{U}}\right)}_{\vec{E}_{\text{trunc}}}
$$

The truncature error can now be written with a turbulent viscosity model so that $\vec{E}_{\text{trunc}} \approx -\vec{\nabla}\cdot\left(\nu_t\vec{\nabla}\otimes\widetilde{\vec{U}}\right)$. Let us now integrate the product of the above equation with the velocity:

$$
\int\left(\frac{\partial\widetilde{\vec{U}}}{\partial t}+\left(\widetilde{\vec{U}}\cdot\vec{\nabla}\right)\widetilde{\vec{U}}\right)\cdot\widetilde{\vec{U}}dV =
$$
$$
\int\left(-\frac{1}{\rho_\infty}\vec{\nabla}\widetilde{p}+\vec{\nabla}\cdot\left((\nu+\nu_t)\,\vec{\nabla}\otimes\widetilde{\vec{U}}\right)\right)dV
$$
$$
\frac{\partial}{\partial t}\int\frac{1}{2}||\widetilde{\vec{U}}||^2dV+\int\left(\left(\widetilde{\vec{U}}\cdot\vec{\nabla}\right)\widetilde{\vec{U}}\right)\cdot\widetilde{\vec{U}}dV =
$$
$$
\int -\frac{1}{\rho_\infty}\left(\widetilde{\vec{U}}\cdot\vec{\nabla}\right)\widetilde{p}dV
$$
$$
+\int\left(\vec{\nabla}\cdot\left((\nu+\nu_t)\,\vec{\nabla}\otimes\widetilde{\vec{U}}\right)\right)\cdot\widetilde{\vec{U}}dV
$$

Let us write each term of the equation above:

$$
\frac{\partial}{\partial t}\int\frac{1}{2}||\widetilde{\vec{U}}||^2dV=\frac{\partial}{\partial t}\widetilde{\mathcal{E}_c}
$$

where $\widetilde{\mathcal{E}_c}$ is the filtered Kinetic Energy.

$$
\begin{aligned}
\int\left(\left(\widetilde{\vec{U}}\cdot\vec{\nabla}\right)\widetilde{\vec{U}}\right)\cdot\widetilde{\vec{U}}dV &= \int\widetilde{U_j}\frac{\partial\widetilde{U_i}}{\partial x_j}\widetilde{U_i}dV\\
&=\frac{1}{2}\int\widetilde{U_j}\frac{\partial\widetilde{U_i}^2}{\partial x_j}dV\\
&=\frac{1}{2}\int\widetilde{U_j}\frac{\partial\widetilde{U_i}^2}{\partial x_j}dV\\
&=\frac{1}{2}\cancelto{0}{[\widetilde{U_j}\widetilde{U_i}^2]_{-\infty}^{+\infty}}-\int\cancelto{0}{\frac{\partial\widetilde{U_j}}{\partial x_j}}\widetilde{U_i}^2dV
\end{aligned}
$$

The first term above goes to zero for an unbounded domain since $\lim_{x\to\pm\infty}\widetilde{U}=0$. Note that this is also true in a bounded domain as long as the velocity is null at the boundaries. The second term goes to zero because of the conservation of mass.

$$
\begin{aligned}
\int -\frac{1}{\rho_\infty}\left(\widetilde{\vec{U}}\cdot\vec{\nabla}\right)\widetilde{p}dV &= -\frac{1}{\rho_\infty}\int\widetilde{U_i}\frac{\partial\widetilde{p}}{\partial x_i}dV\\
&=-\frac{1}{\rho_\infty}\left(\cancelto{0}{[\widetilde{p}\widetilde{U_i}]_{-\infty}^{+\infty}}-\int p\cancelto{0}{\frac{\partial\widetilde{U_i}}{\partial x_i}}dV\right)
\end{aligned}
$$

Both terms above go to zero for the same reason as in the previous equation.

$$
\begin{aligned}
&\int\left(\vec{\nabla}\cdot\left((\nu+\nu_t)\,\vec{\nabla}\otimes\widetilde{\vec{U}}\right)\right)\cdot\widetilde{\vec{U}}dV\\
&=\int\frac{\partial}{\partial x_j}\left((\nu+\nu_t)\,\frac{\partial\widetilde{U_i}}{\partial x_j}\right)\widetilde{U_i}dV\\
&=\cancelto{0}{\left[(\nu+\nu_t)\,\frac{\partial\widetilde{U_i}}{\partial x_j}\widetilde{U_i}\right]_{-\infty}^{+\infty}}-\int(\nu+\nu_t)\left(\frac{\partial\widetilde{U_i}}{\partial x_j}\right)^2dV\\
&=-\int(\nu+\nu_t)\,||\vec{\nabla}\otimes\widetilde{\vec{U}}||^2dV\\
&=-\nu\widetilde{\mathcal{E}}-\int\nu_t||\vec{\nabla}\otimes\widetilde{\vec{U}}||^2dV
\end{aligned}
$$

where $\widetilde{\mathcal{E}}$ is the filtered Enstrophy. Eventually one has:

$$
\frac{\partial\widetilde{\mathcal{E}_c}}{\partial t}=-\nu\widetilde{\mathcal{E}}-\int\nu_t||\vec{\nabla}\otimes\widetilde{\vec{U}}||^2dV
$$

C. Solid Induced Velocity Gradients

Let us derive the expression of the following induced velocity:

$$
\vec{u}_\phi=\sum_j^{N_s-1}\frac{\Gamma_0}{4\pi}\overbrace{\frac{1}{\left(||\vec{r}_1\times\vec{r}_2||^{2n}+||\sigma\vec{r_0}||^{2n}\right)^{1/n}}}^{K_1}\underbrace{\vec{r_0}\cdot\left(\frac{\vec{r}_1}{||\vec{r}_1||}-\frac{\vec{r}_2}{||\vec{r}_2||}\right)}_{K_2}\vec{r}_1\times\vec{r}_2
$$

where $\vec{r_0}$, $\vec{r}_1$ and $\vec{r}_2$ are detailed in section 3.2. The gradient of the above expression is:

$$
\vec{\nabla}\otimes\vec{u}_\phi=\sum_j^{N_s-1}\frac{\Gamma_0}{4\pi}\left[K_1K_2\vec{\nabla}\otimes\left(\vec{r}_1\times\vec{r}_2\right)\right.
$$
$$
\left.+\left(K_2\vec{\nabla}K_1+K_1\vec{\nabla}K_2\right)\otimes\left(\vec{r}_1\times\vec{r}_2\right)\right]
$$

Let us compute the gradient of K_1:

$$
\begin{aligned}
\vec{\nabla}K_1&=-\frac{1}{n}K_1^{n+1}\vec{\nabla}\left(||\vec{r}_1\times\vec{r}_2||^{2n}\right)\\
&=-K_1^{n+1}||\vec{r}_1\times\vec{r}_2||^{2(n-1)}\vec{\nabla}\left(||\vec{r}_1\times\vec{r}_2||^2\right)
\end{aligned}
$$

Since:

$$
||\vec{r}_1\times\vec{r}_2||^2=r_1^2r_2^2\sin^2(\theta)
$$

$$= r_1^2 r_2^2 \left(1 - \cos^2(\theta)\right)$$
$$= r_1^2 r_2^2 \left(1 - \left(\frac{\vec{r}_1 \cdot \vec{r}_2}{r_1 r_2}\right)^2\right)$$

And:

$$\begin{cases} \vec{\nabla} r = \vec{r}/r \\ \vec{\nabla} 1/r = \vec{r}/r^3 \end{cases}$$

We have:

$$\vec{\nabla}\left(||\vec{r}_1 \times \vec{r}_2||^2\right) = \vec{\nabla}\left(r_1^2 r_2^2 \left(1 - \left(\frac{\vec{r}_1 \cdot \vec{r}_2}{r_1 r_2}\right)^2\right)\right)$$
$$= 2\left(\vec{r}_1 r_2^2 + r_1^2 \vec{r}_2\right)\left(1 - \left(\frac{\vec{r}_1 \cdot \vec{r}_2}{r_1 r_2}\right)^2\right) - r_1^2 r_2^2 \vec{\nabla}\left(\left(\frac{\vec{r}_1 \cdot \vec{r}_2}{r_1 r_2}\right)^2\right)$$
$$= 2\left(\vec{r}_1 r_2^2 + r_1^2 \vec{r}_2\right)\left(1 - \left(\frac{\vec{r}_1 \cdot \vec{r}_2}{r_1 r_2}\right)^2\right)$$
$$- 2 r_1 r_2 \vec{r}_1 \cdot \vec{r}_2 \left(\frac{\vec{r}_1 + \vec{r}_2}{r_1 r_2} - \vec{r}_1 \cdot \vec{r}_2 \left(\frac{\vec{r}_1}{r_1^3 r_2} + \frac{\vec{r}_2}{r_1 r_2^3}\right)\right)$$
$$= 2\left(r_2^2 - \vec{r}_1 \cdot \vec{r}_2\right)\vec{r}_1 + 2\left(r_1^2 - \vec{r}_1 \cdot \vec{r}_2\right)\vec{r}_2$$
$$= -2\left(\vec{r_0} \cdot \vec{r}_2\right)\vec{r}_1 + 2\left(\vec{r_0} \cdot \vec{r}_1\right)\vec{r}_2$$
$$= -2\vec{r_0} \times \left(\vec{r}_1 \times \vec{r}_2\right)$$

The gradients of K_1 now becomes:

$$\vec{\nabla} K_1 = 2 K_1^{n+1} ||\vec{r}_1 \times \vec{r}_2||^{2(n-1)} \vec{r_0} \times \left(\vec{r}_1 \times \vec{r}_2\right)$$

Let us now calculate the gradients of K_2:

$$\vec{\nabla} K_2 = \vec{\nabla}\left(\frac{\vec{r_0} \cdot \vec{r}_1}{r_1} - \frac{\vec{r_0} \cdot \vec{r}_2}{r_2}\right)$$
$$= \vec{\nabla}\left(r_1 + r_2 - \vec{r}_1 \cdot \vec{r}_2 \left(\frac{1}{r_1} + \frac{1}{r_2}\right)\right)$$
$$= \frac{\vec{r}_1}{r_1} + \frac{\vec{r}_2}{r_2} - \left(\vec{r}_1 + \vec{r}_2\right)\left(\frac{1}{r_1} + \frac{1}{r_2}\right) + \vec{r}_1 \cdot \vec{r}_2 \left(\frac{\vec{r}_1}{r_1^3} + \frac{\vec{r}_2}{r_2^3}\right)$$
$$= \left(\frac{\vec{r}_1 \cdot \vec{r}_2}{r_1^3} - \frac{1}{r_2}\right)\vec{r}_1 + \left(\frac{\vec{r}_1 \cdot \vec{r}_2}{r_2^3} - \frac{1}{r_1}\right)\vec{r}_2$$

Eventually the expression of the gradients of the velocity is:

$$\vec{\nabla} \otimes \vec{u}_\phi = \sum_j^{N_s - 1} \frac{\Gamma_0}{4\pi} \frac{1}{\left(||\vec{r}_1 \times \vec{r}_2||^{2n} + ||\sigma \vec{r_0}||^{2n}\right)^{1/n}} \Bigg[$$
$$\vec{r_0} \cdot \left(\frac{\vec{r}_1}{||\vec{r}_1||} - \frac{\vec{r}_2}{||\vec{r}_2||}\right) \vec{\nabla} \otimes \left(\vec{r}_1 \times \vec{r}_2\right)$$
$$+ \Bigg[\left(\frac{\vec{r}_1 \cdot \vec{r}_2}{r_1^3} - \frac{1}{r_2}\right)\vec{r}_1 + \left(\frac{\vec{r}_1 \cdot \vec{r}_2}{r_2^3} - \frac{1}{r_1}\right)\vec{r}_2$$
$$+ \frac{2||\vec{r}_1 \times \vec{r}_2||^{2(n-1)} \vec{r_0} \cdot \left(\frac{\vec{r}_1}{||\vec{r}_1||} - \frac{\vec{r}_2}{||\vec{r}_2||}\right)}{||\vec{r}_1 \times \vec{r}_2||^{2n} + ||\sigma \vec{r_0}||^{2n}} \vec{r_0} \times \left(\vec{r}_1 \times \vec{r}_2\right)$$
$$\Bigg] \otimes \left(\vec{r}_1 \times \vec{r}_2\right) \Bigg]$$

Where:

$$\vec{\nabla} \otimes \left(\vec{r}_1 \times \vec{r}_2\right) = \begin{pmatrix} 0 & r_{0_z} & -r_{0_y} \\ -r_{0_z} & 0 & r_{0_x} \\ r_{0_y} & -r_{0_x} & 0 \end{pmatrix}$$

References

[1] Rosenhead, L., “The formation of vortices from a surface of discontinuity,” *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, Vol. 134, No. 823, 1931, pp. 170–192.

[2] Alvarez, E. J., and Ning, A., “Reviving the Vortex Particle Method: A Stable Formulation for Meshless Large Eddy Simulation,” 2022. https://doi.org/10.48550/arXiv.2206.03658.

[3] Winckelmans, G., Cocle, R., Dufresne, L., and Capart, R., “Vortex methods and their application to trailing wake vortex simulations,” *Comptes Rendus Physique*, Vol. 6, No. 4, 2005, pp. 467–486. https://doi.org/10.1016/j.crhy.2005.05.001, aircraft trailing vortices.

[4] 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, 1999, pp. 305–345. https://doi.org/10.1006/jcph.1999.6258.

[5] Mas-Gallic, S., “Contribution à l’Analyse Numérique des Méthodes Particulaires,” Ph.D. thesis, Université Paris VI, 1987. https://doi.org/10.15368/theses.2010.104.

[6] 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.

[7] Berdowski, T., Ferreira, C., and Walther, J., “3D Lagrangian VPM: simulations of the near-wake of an actuator disc and horizontal axis wind turbine,” *Journal of Physics: Conference Series*, Vol. 753, 2016. https://doi.org/10.1006/jcph.2000.6656.

[8] Pinon, G., Mycek, P., Germain, G., and Rivoalen, E., “Numerical simulation of the wake of marine current turbines with a particle method,” *Renewable Energy*, Vol. 46, 2012, pp. 111–126. https://doi.org/10.1016/j.renene.2012.03.037.

[9] Calabretta, J., "A Three Dimensional Vortex Particle-Panel Code for Modeling Propeller Airframe Interaction," Ph.D. thesis, Faculty of California Polytechnic State University, 06 2010.

[10] 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*, Vol. 34, 2020. https://doi.org/10.1007/s00162-019-00510-1.

[11] Caprace, D.-G., Chatelain, P., and Winckelmans, G., "Lifting Line with Various Mollifications: Theory and Application to an Elliptical Wing," *AIAA Journal*, Vol. 57, No. 1, 2019, pp. 17–28. https://doi.org/10.2514/1.J057487.

[12] Tan, J. F., Sun, Y. M., Zhou, T. Y., Barakos, G. N., and Green, R. B., "Simulation of the aerodynamic interaction between rotor and ground obstacle using vortex method," *CEAS Aeronautical Journal*, Vol. 10, 2019, pp. 733–753. https://doi.org/10.1007/s13272-018-0346-8.

[13] Manickathan, L., "Hybrid Eulerian-Lagrangian Vortex Particle Method. A Fast and Accurate Numerical Method for 2D Vertical-Axis Wind Turbine," Ph.D. thesis, Delft University of Technology, 12 2014.

[14] Shi, Y., Xu, G., and Wei, P., "Rotor wake and flow analysis using a coupled Eulerian–Lagrangian method," *Engineering Applications of Computational Fluid Mechanics*, Vol. 10, No. 1, 2016, pp. 384–402. https://doi.org/10.1080/19942060.2016.1174887.

[15] Mansfield, J. R. J., "A Dynamic Lagrangian Large Eddy Simulation Scheme for the Vorticity Transport Equation," Ph.D. thesis, John Hopkins University, 1997.

[16] He, C., and Zhao, J., "Modelling Rotor Wake Dynamics With Viscous Vortex Particle Method," *AIAA Journal*, Vol. 47, No. 4, 2009, pp. 902–915. https://doi.org/10.2514/1.36466.

[17] Singh, P., "Aeromechanics of Coaxial Rotor Helicopters Using the Viscous Vortex Particles Method," Ph.D. thesis, University of Michigan, 2020.

[18] 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, 2005. https://doi.org/10.1002/fld.811.

[19] Rossi, L., "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.

[20] Alvarez, E. J., and Ning, A., "High-Fidelity Modeling of Multirotor AerodynamicInteractions for Aircraft Design," *AIAA Journal*, Vol. 58, No. 10, 2020. https://doi.org/10.2514/1.J059178.

[21] 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.

[22] Mycek, P., Pinon, G., Germain, G., and Rivoalen, E., "A self-regularising DVM–PSE method for the modelling of diffusion in particle methods," *Comptes Rendus Mécanique*, Vol. 341, No. 9, 2013, pp. 709–714. https://doi.org/10.1016/j.crme.2013.08.002.

[23] Mycek, P., Pinon, G., Germain, G., and Rivoalen, E., "Formulation and analysis of a diffusion-velocity particle model for transport-dispersion equations," *Computational and Applied Mathematics*, Vol. 35, 2016, pp. 447–473. https://doi.org/10.1007/s40314-014-0200-5.

[24] Lacombe, G., and Mas-Gallic, S., "Presentation and analysis of a diffusion-velocity method," *Esaim: Proceedings*, Vol. 7, 1999, pp. 225–233. https://doi.org/10.1051/proc:1999021.

[25] Rivoalen, E., Huberson, S., and Hauville, F., "Simulation numérique des équations de Navier-Stokes 3D par une méthode particulaire," *Comptes Rendus de l'Académie des Sciences - Series IIB - Mechanics-Physics-Chemistry-Astronomy*, Vol. 324, No. 9, 1997, pp. 543–549. https://doi.org/10.1016/S1251-8069(97)83187-9.

[26] 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.

[27] Kwak, D., "Three-dimensional time dependent computation of turbulent flow," Ph.D. thesis, Stanford University, California, Jan. 1975.

[28] Vreman, A. W., "An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications," *Physics of Fluids*, Vol. 16, No. 10, 2004, pp. 3670–3681. https://doi.org/10.1063/1.1785131.

[29] Germano, M., "Turbulence: the filtering approach," *Journal of Fluid Mechanics*, Vol. 238, 1992, p. 325–336. https://doi.org/10.1017/S0022112092001733.

[30] Wang, T., Yokota, R., and Barba, L. A., "ExaFMM: a high-performance fast multipole method library with C++ and Python interfaces," *Journal of Open Source Software*, Vol. 6, No. 61, 2021. https://doi.org/10.21105/joss.03145.

[31] 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.

[32] Du, Z., and Selig, M., *A 3-D stall-delay model for horizontal axis wind turbine performance prediction*, AIAA/ASME Wind Energy Symposium, 1998, pp. 9–19. https://doi.org/10.2514/6.1998-21.

[33] Sugar-Gabor, O., "A general numerical unsteady non-linear lifting line model for engineering aerodynamics studies," *The Aeronautical Journal*, Vol. 122, No. 1254, 2018, pp. 1199–1228. https://doi.org/10.1017/aer.2018.57.

[34] 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.

[35] Hauville, F., "Optimization of calculation methods for unsteady vortical flows," Ph.D. thesis, Université du Havre, 1996.

[36] Vatistas, G. H., Kozel, V., and Mih, W. C., "A simpler model for concentrated vortices," *Experiments in Fluids*, Vol. 11, 1991, pp. 73–76. https://doi.org/10.1007/BF00198434.

[37] Proulx-Cabana, V., Tuan, N., Prothin, S., Michon, G., and Laurendeau, E., "A Hybrid Non-Linear Unsteady Vortex Lattice-Vortex Particle Method for Rotor Blades Aerodynamic Simulations," *Fluids*, Vol. 7, 2022, p. 81. https://doi.org/10.3390/fluids7020081.

[38] Lamb, H., *Hydrodynamics*, Cambridge University Press, 1932.

[39] Saffman, P. G., "The velocity of viscous vortex rings," *Stud. Appl. Math.*, Vol. 49, 1970, pp. 371–380. https://doi.org/10.1002/sapm1970494371.

[40] L. Cambier, S. H., and Plot, S., "The Onera elsA CFD software: Input from research and feedback from industry," *Mechanics and Industry*, Vol. 14, 2013, pp. 159–174. https://doi.org/10.1051/meca/2013056.

[41] Wilcox, D., *Turbulence Modeling for CFD (Third Edition) (Hardcover)*, Dcw Industries, 2006.

[42] Anderson, J., *Fundamentals of Aerodynamics*, 6th ed., McGraw-Hill Education, 2017.

[43] Wickersheim, R., Kostek, A., Lößle, F., Keßler, M., Boisard, R., Reboul, G., Barbarino, M., and Gardner, A., "Experimental Investigation of UAV Rotor Aeroacoustics and Aerodynamics with Computational Cross-Validation," 48th European Rotorcraft Forum, 2022, pp. 1–14.

[44] Kostek, A., Braukmann, J., Lößle, F., Gardner, A., Riziotis, V., Miesner, S., Keßler, M., Visingardi, A., and Boisard, R., "Experimental Investigation of Quadrotor Aerodynamics with Computational Cross-Validation," 2023, pp. 1–16. https://doi.org/10.4050/F-0079-2023-17991.