Wake Preservation Using a Coupled Eulerian-Lagrangian Solver

Christopher Wales * Dorian J. Jones † and Ann L. Gaitonde‡
University of Bristol, Bristol, BS8 1TR, United Kingdom

A new vortex particle approach seeding approach for a coupled Vortex Particle Method-Computational Fluid Dynamics solver. The vortex particles are seeded based on the vortcity flux through a 2D slice in the mesh. Results are presented for a wingtip vortex for the coupled approach on a coarse mesh and standard CFD on a fine mesh. The affect of particle seeding resulting on the level of vorticity captured are investigated.

I. Introduction

Accurate predicting the flows with strong shed vorticity are important in several cases; such as helicopter blade-vortex interaction, propeller wakes and aircraft wakes in take-off and landing. Blade-vortex interaction occurs when the trailing wake from a helicopter rotor interacts with the oncoming blade causing unsteady loading on the blades and aerodynamic noise [1]. The wake from a wing mounted propeller has a strong influence on the flow over wing and tailplane effecting the aircraft loads, especially the aircraft moments. Wingtip vortices that are generated on aircraft take-off and landing can persist for a long time. This determines the required separation distance to ensure the safety of following aircraft. The ability to accurately predict these types of flows represents a challenge in fluid simulation methods. The literature shows that Lagrangian, Eulerian and hybrid Eulerian/Lagrangian Computational Fluid Dynamics (CFD) techniques have been applied to these kinds of flows.

Grid based Eulerian CFD methods are widely used for modelling viscous flows in aerospace applications, typically using an unsteady Reynolds Averaged Navier-Stokes (URANS) framework with a turbulence model. Eulerian methods suffer from two sources of numerical dissipation which typically causes the tip vortex to diffuse prematurely, making long term simulations of the wingtip vortex difficult. The first source of numerical dissipation is the intrinsic numerical dissipation related to the local grid spacing. The second source is the artificial dissipation used to stabilise the commonlly used central difference schemes. Refining the mesh in regions of high flow gradients, such as regions of high vorticity, increases the accuracy of the solution. This results in a requirement for a fine mesh and thus very large computational cost. An other approach, which avoids the mesh refinement is to use a higher order method. Eulerian CFD methods have been successful applied to modelling the long term behaviour of the wake behind a full aircraft [2]. A hybrid scheme was used in [3] where a 2^{nd} order finite volume scheme was used near the aircraft and 6^{th} order finite difference scheme to resolve the vortex away from the aircraft.

Lagrangian vortex particle approaches are mesh free method where the vorticity is carried by particles that are transported by the flow field. The particles are convected without the non-linear terms encountered in Eulerian methods making them suitable for long term simulations of compact vortical structures. A range of different vortex particle methods have been used to model aircraft wakes and tip vorticies; a vortex filament method was used in [4] to model a simplified wake structure, a vortex panel method in [5] for the same purpose and a vortex particle method with a billion particles to model the aircraft wake [6].

Applying the viscous boundary conditions is harder to implement in VPM then in the Eulerain frame work. This has lead to the idea of using a hybrid approach, where an Eulerain frame work is used close to the surface where viscous effects dominate. A Lagraigain approach is then used away from the surface to convect the vorticity. Most methods also employ an 'overlap' region to couple the two methods where both solvers are used and the solutions are interpolated between each other. The first example of this is by Sitaraman [7] who coupled a RANS solver to a Particle Vorticity Transport Method (PVTM) to simulate rotor wakes. Anusonti-Inthra and Floros [8] extend this to a viscous PVTM to model the flow in isolated wing wakes. Zhao *et al.* [9] employ a viscous vortex particle method together with two different RANS solvers to investigate rotor wake flow. Pahla *et al.* [10] take a slightly different approach. Whilst the domain is still decomposed into regions, the Lagrangian method is applied to the entire domain whilst the Eulerian solver is only applied to the region close to the solid boundary. Essentially, the CFD is used to correct the particle method in the near-field.

*Research Associate, Department of Aerospace Engineering
†Professor of Aerodynamics, Department of Aerospace Engineering
‡Professor of Aerodynamics, Department of Aerospace Engineering

In this work, we apply a hybrid method that couples a RANS solver with a Vortex Particle Method. The domain is not decomposed into regions; instead the VPM is used to correct the CFD. This is done by including the influence of the particles in the CFD solution through the Split Velocity Method, which has previously been used in the simulation of gusts [11, 12]. This approach has been previously implemented in [13, 14]. The seeding approach was found to be numerically ill-conditioned. In this paper a new approach for seeding the vortex particles will be presented and applied at capturing a wing tip vortex.

II. Methodology

The Navier-Stokes equations govern the motion of fluid flow. These equations can be solved either in an Eulerian or a Lagrangian formulation. The Eulerian approach is typically used in CFD where the flow quantities are considered a function of the spatial locations as they change in time. Eulerian solvers require a fine mesh throughout the regions of interest to accurately capture the flow gradients. This leads to large numbers of mesh cells and thus high computational costs. Lagrangian vortex particle methods are an alternative method for transporting vorticity through the flow domain without having to refine the mesh. In this work vorticity passing through a 2D region in the flow is identified and turned into equivalent vortex particles. The velocity influence of these particles is then subtracted from the CFD solution. The vortex particles are convected using a Lagrangian solver modified by the surrounding CFD solution. This allows a coarser CFD mesh to be used without the vorticity being dissipated. The VPM has been coupled with the CFD solver using the SVM method using the implementation similar to that defined in [13]. Details of the VPM and SVM implementation are repeated in this paper along with the new seeding approach.

The coupling process is shown in Fig. 1. There are four main stages per timestep. The first is to identify and vorticity in the seeding region of the domain and add the vorticity to new particles, this step is described in section II.A.3. Next, the particles are updated for one timestep of the vortex particle method, see section II.A. Then the velocities induced by the vortex particles on the CFD mesh are calculated. Finally these velocities are injected into the CFD solution using the SVM method, as described in section II.A.1, and the CFD flow field is solved for one timestep.

A. Vortex Particle Method

Vortex Particle Methods discretises the flow into particles, with strengths related to the vorticity over the particle volume, and are convected using a transport equation in a Lagragian framework. The vorticity is related to the velocity field by

$$\omega = \nabla \times u. \tag{1}$$

Taking the curl of the Euler equations leads to the resulting equations for the evolution of the vorticity

$$\frac{\partial \omega}{\partial t} + \mathbf{U}.\nabla\omega = \omega.\nabla\mathbf{U} \tag{2}$$

Using a point vortex can lead to numerical problems due to sigularities at the point voticies. Instead regularized vortex particles are used, which have a finet core size, given by

$$\omega(\mathbf{x}, t) = \sum_{i=1}^{N} \Gamma_i(t)\zeta_{\sigma_i}(\mathbf{x} - \mathbf{x}_i(t)) \tag{3}$$

where Γ_i is the strength of particle i, ζ_σ is the regularized smoothing kernel [15] and σ_i is the radius of particle i. A constant particle radius has been used, so $\sigma_i = \sigma$. For a three-dimensional case, the regularized smoothing kernel is given by

$$\zeta_\sigma = \frac{1}{\sigma^3}\zeta\left(\frac{|\mathbf{x}|}{\sigma}\right) \tag{4}$$

where $\zeta(\rho)$ is the smoothing or cutoff function. The cutoff function should be smooth and accurate [16] and several choices are listed in the paper by Wincklemans and Leonard [17]. In this work the second-order 3D gaussian smoothing, given by Eq. (5), is used.

$$\zeta\left(\rho\right) = \left(\frac{2}{\pi}\right)^{1/2} \exp\left(-\frac{\rho^2}{2}\right) \tag{5}$$

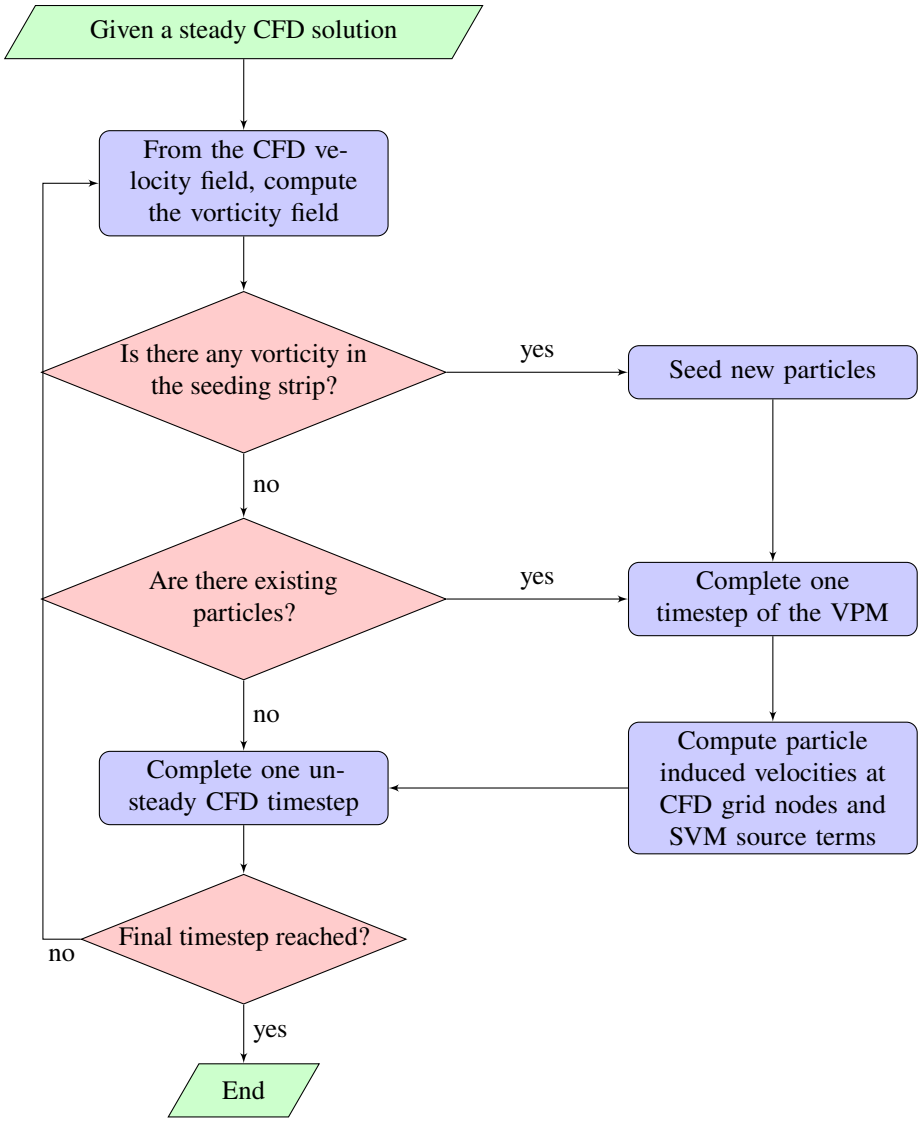


Fig. 1 Flowchart of the coupled CFD-VPM solver

The equations of motion for a regularized vortex particle method are given by

$$\frac{d}{dt}\mathbf{x}\left(t\right) = \mathbf{u}_\sigma\left(\mathbf{x}\left(t\right), t\right) \tag{6}$$

$$\frac{d}{dt}\Gamma\left(t\right) = \omega \cdot \nabla \mathbf{u}_\sigma V \tag{7}$$

where V is the volume associated with a single particle. The right hand side of Eq. (7) corresponds to stretching. The velocity $\mathbf{u}_\sigma\left(\mathbf{x}, t\right)$ is computed using the generalized Biot-Savart equation, which is

$$\mathbf{u}_\sigma\left(\mathbf{x}, t\right) = -\sum_{i=1}^{N} \mathbf{K}_\sigma\left(\mathbf{x} - \mathbf{x}_i(t)\right)\Gamma_i(t). \tag{8}$$

When written in terms of Green's function for the Laplace equation this velocity is given by

$$\mathbf{u}_\sigma\left(\mathbf{x}, t\right) = -\sum_{i=1}^{N} \nabla G(\mathbf{x} - \mathbf{x}_i(t)) \times g_\sigma\left(\frac{\mathbf{x}}{\sigma}\right)\boldsymbol{\Gamma}_i(t) \tag{9}$$

where $g_\sigma(\rho)$ is defined as

$$g_\sigma(\rho) = erf\left(\frac{\rho}{2^{1/2}}\right) - \left(\frac{2}{\pi}\right)^{1/2} \rho \exp(-\rho^2/2) \tag{10}$$

The Biot-Savart equation, Eq. (8), is solved twice. First it is used to calculate the velocities and gradients the particles induce on each other. This velocity, along with velocity and gradients are interpolated from the CFD is used to convect and stretch the vortex particles. Formally the velocity field in the VPM is formed as the summation of the CFD and VPM and the local vorticity ω is assumed entirely captured by the VPM.

$$\mathbf{U} = \mathbf{U}_{CFD} + \mathbf{U}_{VLM} \tag{11}$$

Secondly, Biot-Savart is used to calculate the velocities induced by the particles at the CFD mesh nodes, which are then used to couple the VPM method to the CFD solution using the Split Velocity Method. A fourth order Runge-Kutta method is used to solve, Eq. (6), and update the particle positions.

Note that in this work the vortex particle method is assumed inviscid. Modifications to include the viscosity in the VPM are possible but not explored here. The effects of viscosity are therefore assumed to be captured entirely by the CFD.

1. Split Velocity Method

The derivation of the Split Velocity Method begins with the unsteady Navier-Stokes equations on a fixed mesh. Then the velocity and energy are decomposed into prescribed components and an unknown component given by

$$u = \tilde{u} + \hat{\hat{u}} \qquad v = \tilde{v} + \hat{\hat{v}} \qquad w = \tilde{w} + \hat{\hat{w}} \qquad E = \tilde{E} + \hat{\tilde{E}} + \hat{\hat{E}} \tag{12}$$

where $\hat{\hat{u}}$, $\hat{\hat{v}}$ and $\hat{\hat{w}}$ are the prescribed velocity components which in this case are the induced vortex particle velocity components. The pressure remains unchanged, which means that the energy components are given by

$$E = \underbrace{\frac{p}{\rho(\gamma-1)} + \frac{1}{2}(\tilde{u}^2 + \tilde{v}^2) + \tilde{w}^2)}_{\tilde{E}} + \underbrace{(\tilde{u}\hat{\hat{u}} + \tilde{v}\hat{\hat{v}} + \tilde{w}\hat{\hat{w}})}_{\hat{\tilde{E}}} + \underbrace{\frac{1}{2}(\hat{\hat{u}}^2 + \hat{\hat{v}}^2) + \hat{\hat{w}}^2)}_{\hat{\hat{E}}} \tag{13}$$

The Navier-Stokes equations for the Split Velocity Method are then obtained by substituting the decompositions into the unsteady Navier-stokes equations giving

$$\begin{gathered}
\frac{\partial}{\partial t}\begin{bmatrix} \rho \\ \rho(\tilde{u}+\hat{\hat{u}}) \\ \rho(\tilde{v}+\hat{\hat{v}}) \\ \rho(\tilde{w}+\hat{\hat{w}}) \\ \rho(\tilde{E}+\hat{\tilde{E}}+\hat{\hat{E}}) \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} \rho(\tilde{u}+\hat{\hat{u}}) \\ \rho(\tilde{u}+\hat{\hat{u}})^2 + p \\ \rho(\tilde{u}+\hat{\hat{u}})(\tilde{v}+\hat{\hat{v}}) \\ \rho(\tilde{u}+\hat{\hat{u}})(\tilde{w}+\hat{\hat{w}}) \\ (\rho(\tilde{E}+\hat{\tilde{E}}+\hat{\hat{E}}))(\tilde{u}+\hat{\hat{u}}) \end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix} \rho(\tilde{v}+\hat{\hat{v}}) \\ \rho(\tilde{v}+\hat{\hat{v}})(\tilde{u}+\hat{\hat{u}}) \\ \rho(\tilde{v}+\hat{\hat{v}})^2 + p \\ \rho(\tilde{v}+\hat{\hat{v}})(\tilde{w}+\hat{\hat{w}}) \\ (\rho(\tilde{E}+\hat{\tilde{E}}+\hat{\hat{E}}))(\tilde{v}+\hat{\hat{v}}) \end{bmatrix} + \frac{\partial}{\partial z}\begin{bmatrix} \rho(\tilde{v}+\hat{\hat{v}}) \\ \rho(\tilde{w}+\hat{\hat{w}})(\tilde{u}+\hat{\hat{u}}) \\ \rho(\tilde{w}+\hat{\hat{w}})(\tilde{v}+\hat{\hat{v}}) \\ \rho(\tilde{w}+\hat{\hat{w}})^2 + p \\ (\rho(\tilde{E}+\hat{\tilde{E}}+\hat{\hat{E}}))(\tilde{w}+\hat{\hat{w}}) \end{bmatrix} \\
+\frac{\partial}{\partial x}\begin{bmatrix} 0 \\ \sigma_{xx} \\ \sigma_{xy} \\ \sigma_{xz} \\ (\tilde{u}+\hat{\hat{u}})\sigma_{xx} + (\tilde{v}+\hat{\hat{v}})\sigma_{xy} + (\tilde{w}+\hat{\hat{w}})\sigma_{xz} + q_x \end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix} 0 \\ \sigma_{xy} \\ \sigma_{yy} \\ \sigma_{yz} \\ (\tilde{u}+\hat{\hat{u}})\sigma_{xy} + (\tilde{v}+\hat{\hat{v}})\sigma_{yy} + (\tilde{w}+\hat{\hat{w}})\sigma_{yz} + q_y \end{bmatrix} \\
+\frac{\partial}{\partial y}\begin{bmatrix} 0 \\ \sigma_{xz} \\ \sigma_{yz} \\ \sigma_{zz} \\ (\tilde{u}+\hat{\hat{u}})\sigma_{xz} + (\tilde{v}+\hat{\hat{v}})\sigma_{yz} + (\tilde{w}+\hat{\hat{w}})\sigma_{zz} + q_z \end{bmatrix} \\
= 0
\end{gathered} \tag{14}$$

where,

$$\sigma_{xx} = \frac{2}{3}\frac{\mu}{Re}\left(2\frac{\partial\tilde{u}}{\partial x} - \frac{\partial\tilde{v}}{\partial y} - \frac{\partial\tilde{w}}{\partial z}\right) + \frac{2}{3}\frac{\mu}{Re}\left(2\frac{\partial\hat{\hat{u}}}{\partial x} - \frac{\partial\hat{\hat{v}}}{\partial y} - \frac{\partial\hat{\hat{w}}}{\partial z}\right) \quad \sigma_{yy} = \frac{2}{3}\frac{\mu}{Re}\left(2\frac{\partial\tilde{v}}{\partial y} - \frac{\partial\tilde{u}}{\partial x} - \frac{\partial\tilde{w}}{\partial z}\right) + \frac{2}{3}\frac{\mu}{Re}\left(2\frac{\partial\hat{\hat{v}}}{\partial y} - \frac{\partial\hat{\hat{u}}}{\partial x} - \frac{\partial\hat{\hat{w}}}{\partial z}\right)$$
$$\sigma_{zz} = \frac{2}{3}\frac{\mu}{Re}\left(2\frac{\partial\tilde{w}}{\partial z} - \frac{\partial\tilde{u}}{\partial x} - \frac{\partial\tilde{v}}{\partial y}\right) + \frac{2}{3}\frac{\mu}{Re}\left(2\frac{\partial\hat{\hat{w}}}{\partial z} - \frac{\partial\hat{\hat{u}}}{\partial x} - \frac{\partial\hat{\hat{v}}}{\partial y}\right)$$

$$\sigma_{xy} = \frac{\mu}{Re}\left(\frac{\partial\tilde{u}}{\partial y} + \frac{\partial\tilde{v}}{\partial x}\right) + \frac{\mu}{Re}\left(\frac{\partial\hat{\hat{u}}}{\partial y} + \frac{\partial\hat{\hat{v}}}{\partial x}\right)$$
$$\sigma_{xz} = \frac{\mu}{Re}\left(\frac{\partial\tilde{u}}{\partial z} + \frac{\partial\tilde{w}}{\partial x}\right) + \frac{\mu}{Re}\left(\frac{\partial\hat{\hat{u}}}{\partial z} + \frac{\partial\hat{\hat{w}}}{\partial x}\right)$$
$$\sigma_{yz} = \frac{\mu}{Re}\left(\frac{\partial\tilde{w}}{\partial y} + \frac{\partial\tilde{v}}{\partial z}\right) + \frac{\mu}{Re}\left(\frac{\partial\hat{\hat{w}}}{\partial y} + \frac{\partial\hat{\hat{v}}}{\partial z}\right)$$

$$q_x = -\frac{\mu}{Pr}\frac{1}{Re}\frac{1}{(\gamma-1)M_\infty^2}\frac{\partial T}{\partial x} \quad q_y = -\frac{\mu}{Pr}\frac{1}{Re}\frac{1}{(\gamma-1)M_\infty^2}\frac{\partial T}{\partial y} \quad q_y = -\frac{\mu}{Pr}\frac{1}{Re}\frac{1}{(\gamma-1)M_\infty^2}\frac{\partial T}{\partial z}$$

$$T = \frac{\gamma M_\infty^2 p}{\rho}$$

Pr, μ and Re are the Prandtl number, dynamic viscosity and Reynolds number respectively. Separating the induced velocities from the rest of the solution and after some manipulation of the terms, the Navier-Stokes equations can be rewritten as

$$\begin{aligned}
&\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho\tilde{u}\\ \rho\tilde{v}\\ \rho\tilde{E}\end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix}\rho(\tilde{u}+\hat{\hat{u}})\\ \rho\tilde{u}(\tilde{u}+\hat{\hat{u}})+p\\ \rho\tilde{v}(\tilde{u}+\hat{\hat{u}})\\ \rho\tilde{w}(\tilde{u}+\hat{\hat{u}})\\ \rho\tilde{E}(\tilde{u}+\hat{\hat{u}})+p\tilde{u}\end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix}\rho(\tilde{v}+\hat{\hat{v}})\\ \rho\tilde{u}(\tilde{v}+\hat{\hat{v}})\\ \rho\tilde{v}(\tilde{v}+\hat{\hat{v}})+p\\ \rho\tilde{w}(\tilde{v}+\hat{\hat{v}})\\ \rho\tilde{E}(\tilde{v}+\hat{\hat{v}})+p\tilde{v}\end{bmatrix} + \frac{\partial}{\partial z}\begin{bmatrix}\rho(\tilde{v}+\hat{\hat{v}})\\ \rho\tilde{u}(\tilde{w}+\hat{\hat{w}})\\ \rho\tilde{v}(\tilde{w}+\hat{\hat{w}})\\ \rho\tilde{w}(\tilde{w}+\hat{\hat{w}})+p\\ \rho\tilde{E}(\tilde{w}+\hat{\hat{w}})+p\tilde{w}\end{bmatrix}\\
&+\frac{\partial}{\partial x}\begin{bmatrix}0\\ \sigma_{xx}\\ \sigma_{xy}\\ \sigma_{xz}\\ \tilde{u}\sigma_{xx}+\tilde{v}\sigma_{xy}+\tilde{w}\sigma_{xz}+q_x\end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix}0\\ \sigma_{xy}\\ \sigma_{yy}\\ \sigma_{yz}\\ \tilde{u}\sigma_{xy}+\tilde{v}\sigma_{yy}\tilde{w}\sigma_{yz}+q_y\end{bmatrix} + \frac{\partial}{\partial z}\begin{bmatrix}0\\ \sigma_{xz}\\ \sigma_{yz}\\ \sigma_{zz}\\ \tilde{u}\sigma_{xz}+\tilde{v}\sigma_{yz}\tilde{w}\sigma_{zz}+q_z\end{bmatrix}\\
&+\begin{bmatrix}0\\ s_m(\hat{\hat{u}})\\ s_m(\hat{\hat{v}})\ s_m(\hat{\hat{w}})\\ s_e(\hat{\hat{u}},\hat{\hat{v}},\hat{\hat{w}})\end{bmatrix} = 0.
\end{aligned} \tag{15}$$

The resulting equations have the same form as the Navier-Stokes equations for a moving mesh, with some additional sources terms. This means that the SVM method can be implemented in a moving mesh CFD solver by setting the mesh velocities to the vortex particle induced velocities and modifying the code to include the source terms. In previous work [11] on gust simulations it was found that the source terms only became important for gust lengths much shorter than required for aircraft certification. For this work the source terms have been included and are given by

$$s_m(\cdot) = \rho\left\{\frac{\partial\cdot}{\partial t} + (\tilde{u}+\hat{\hat{u}})\frac{\partial\cdot}{\partial x} + (\tilde{v}+\hat{\hat{v}})\frac{\partial\cdot}{\partial y} + (\tilde{w}+\hat{\hat{w}})\frac{\partial\cdot}{\partial z}\right\} \tag{16}$$

$$
\begin{aligned}
s_e(\hat{\hat{u}}, \hat{\hat{v}}, \hat{\hat{w}}) &= \tilde{u} s_m(\hat{\hat{u}}) + \tilde{v} s_m(\hat{\hat{v}}) + \tilde{w} s_m(\hat{\hat{w}}) + p \left[\frac{\partial \hat{\hat{u}}}{\partial x} + \frac{\partial \hat{\hat{v}}}{\partial y} + \frac{\partial \hat{\hat{w}}}{\partial yz} \right] \\
&+ \sigma_{xx} \frac{\partial \hat{\hat{u}}}{\partial x} + \sigma_{yy} \frac{\partial \hat{\hat{v}}}{\partial y} + \sigma_{zz} \frac{\partial \hat{\hat{w}}}{\partial z} + \sigma_{xy} \left[\frac{\partial \hat{\hat{v}}}{\partial x} + \frac{\partial \hat{\hat{u}}}{\partial y} \right] + \sigma_{xz} \left[\frac{\partial \hat{\hat{w}}}{\partial x} + \frac{\partial \hat{\hat{u}}}{\partial z} \right] + \sigma_{yz} \left[\frac{\partial \hat{\hat{w}}}{\partial y} + \frac{\partial \hat{\hat{v}}}{\partial z} \right]
\end{aligned} \tag{17}
$$

The terms are deemed as sources as they only involve derivatives of the induced velocities. It is noted that the stress tensors, σ, in SVM are calculated based on velocity derivatives for total velocities u and v meaning that they include the induced vortex particle velocities for the calculation of viscous fluxes. This is to eliminate the introduction of dissipative source terms arising from separating velocity derivatives.

Note also that the above is merely a rewriting of the Navier-Stokes equations. Any additional assumptions over those usually made for RANS CFD solvers come from the VPM.

2. Fast Multipole Method

Calculating the induced velocity at each mesh point requires the valuation of, Eq (8), for each vortex particles. This means that the cost of this step is $O(NM)$, where N is the number of particles and M is the number CFD mesh points. Similarly calculating the source terms, Eqs. (16) and (17), directly requires $O(NM)$. The Fast Multipole Method (FMM) [18] allows the cost of the velocity computation to be reduced to $O(N)$, details of the implementation of the FMM for both the induced velocity and source terms can be found in [13].

The cost of the velocity influence on the CFD grid is also reduced using a localisation scheme where CFD points beyond a set distance from any particle are assumed to have a negligible induced velocity. In this work it was found that assuming a distance of 10 times the core radius made no measurable difference to the solutions.

3. Particle seeding

Vortex particles are initialised or "seeded" according to the circulation in the CFD solution. A 2D region is identified and particles are seeded based on the circulation flux through the 2D plane. The 2D region is split up into a uniform grid. A slice through the 3D mesh is taken where it intersects the 2D seeding plane, creating a 2D intermediate mesh. The vorticity and velocity are then interpolated from the CFD solution onto the 2D mesh that results from the slicing. Integrating the vorticity flux through each cell in the grid, as shown in the following equation, gives the particle strength

$$
\Gamma_i = \sum_j^N A_j \omega_j \mathbf{u}_j \Delta t, \tag{18}
$$

where i is the cell index in the 2D grid, j are the cells in the intermediate mesh that overlap with cell i and A_j are the associated cell areas. The uniform 2D grid spacing, h, is set to

$$
h = u_\infty \Delta t \tag{19}
$$

The particles are placed at the circulation moment centre given by

$$
\mathbf{x}_i = \frac{\sum_j^N A_j \omega_j (\mathbf{x}_j + \mathbf{u}_j \Delta t)}{\sum_j^N A_j \omega_j}, \tag{20}
$$

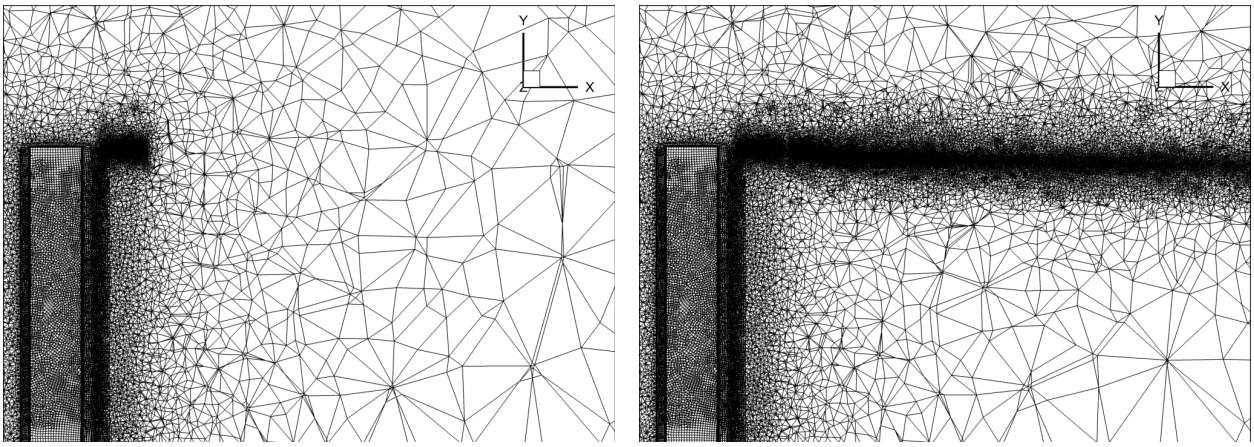
where $\mathbf{x}_i$ is the particle seeding location and $\mathbf{x}_j$ is the centre of the intermediate cell j. The particle radius, σ, is then calculated by

$$
\sigma = \frac{h}{\beta} \tag{21}
$$

where β is the particle overlap ratio, which should be less than 1 to ensure that particles overlap [19].

III. Results

Reuslts are presented for modelling a wingtip vortex. The wing configuration studied was rectangular and had a constant and untwisted NACA 0015 profile along the entire span. The wing had an aspect ratio of 6.6. Only half the wing was simulated, with a symmetry plane. An angle of attack of 12° was used. The coarse mesh used for the VPM is shown in Figure 2a. This mesh has some mesh refinement close to the wing tip to capture the initial wing tip vortex formulation, then rapidly coarsen $0.4c$ behind the wing. The fine mesh CFD simulations are performed on a mesh



(a) VPM coarse mesh **(b) CFD fine mesh**

Fig. 2 VPM and CFD meshes

which has been refined in the region of the wing tip vortex and the resulting mesh is shown in Figure 2b. For both the CFD and SVM the DLR TAU code [20] was used, which is and unstructured solver with a cell-centred finite volume scheme. The convective flux of the mean flow equations is approximated using a central difference scheme with matrix dissipation. For the time integration a dual-timestepping method was used, where the time-derivative is discretised using a second-order Backward Difference Formula (BDF). Each time step is converged using the Backward Euler method in which the linear system is solved using the Lower-Upper Symmetric Gauss-Seidel (LUSGS) scheme. For the turbulence model the Spalart Allmaras has been used.

The fine CFD mesh preserves the wingtip vortex well as shown in 4a, which shows vorticity contour plot through the centre line of the wingtip vortex. However, running the CFD on the coarse mesh results in the vorticity dissipation as soon as it reaches the unrefined region of the mesh as shown Figure 4b. The vortex particles are seeded in a region $0.3c$ behind the aerofoil as shown by the red line in Figures 4 to 5. As result of the vortex particles being isotropic the grid spacing of the seeding region is set to $\Delta T U_\infty$, this means that the number of particles inserted per timestep is a function of the timestep size. Figure 4c show the vorticity contours for a VPM simulation for same timestep size as the CFD. This shows that the lower vorticity levels are maintained into the the coarse mesh region, however the higher levels of vorticity aren't captured as well. Part of the reason for this is that the particles are seeded based on circulation strength which has the effect of averaging the vorticity in the seeding grid, shown in Figure 3. This also results in the vorticity being spreed over a larger area which can be seen from the vorticity iso surface show in Figure 5. Reducing the time step to seed approximately double and quadruple the number of particle improves the results for the higher levels of vorticity as show in Figures 4. Increasing the number of particles from 121 to 225 shows a large improvement in capturing the high levels of vorticity with a smaller improvement going from 225 to 484 particles. To increase the resulting solution resolution while keeping the timestep the same, one simulation was performed where the seeding box size has halved in each direction and the vorticity flux though each box split between two particles in the local flow direction. Compared to quartering the timestep to get a similar 2D seeding grid this approach matches the lower levels of vorticity well as shown in 5e and 5f. At the higher levels of vorticity this approach of splitting the seeded vortex particles initial captures the vorticity better then not splitting the vortex particles at the same timestep size as can be seen by comparing Figures 4c and 4f. However, the higher levels of vorticity aren't preserved as well as the smaller timesteps simulations, Figures 4d and 4e. This indicates that a smaller time step is required for the VPM method to preserve the higher levels of vorticity. Figures 6 and 7 show slices the tip vortex at 2 and 3 chords behind the seeding region. The show that resolution of the seeding region better captures the core of the vortex region. Splitting the vortex particles and seeding two particles per timestep for each seeding grid cell results in a more fragmented vortex core.

IV. Conclusion

The vorticity flux through a 2D seeding grid has been shown to be a viable approach for seeding particles into the flow for a coupled CFD-VPM code. This approach works well for lower levels of vorticity. Higher resolution seeding

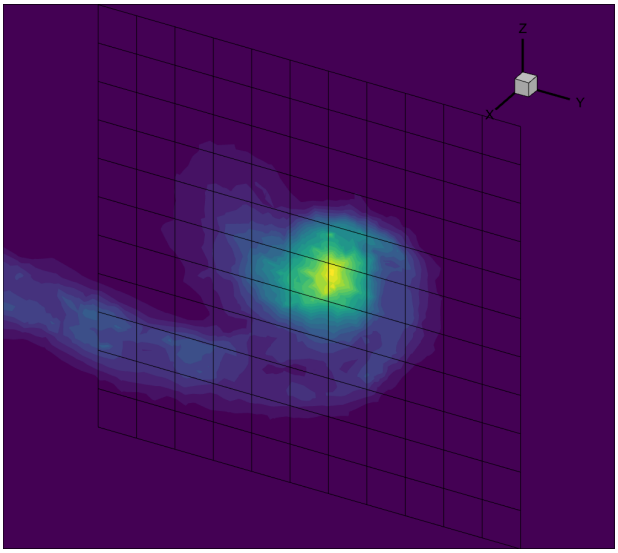


Fig. 3 Vorticity at intersection between seeding grid and CFD mesh

grids are required to capture higher levels of vorticity. In the base line approach this requires reducing the timestep increasing the simulation cost. Seeding multiple particles in the streamwise direction per timestep allows a finer seeding grid to be used without changing the timestep and improves the initial capturing of high levels of vorticity. However the higher levels of vorticity are not maintend as well as the small timestep cases. The next step would be to decouple the timestep size used in the VPM from the CFD, to allow smaller timesteps in VPM without decreasing the CFD timestep.

Acknowledgments

This work was carried out within the Clean Sky 2 research programme PERTURB, funded under grant number 831927. The authors gratefully acknowledge the support of Airbus Defence and Space. This work was carried out using the computational facilities of the Advanced Computing Research Centre, University of Bristol - http://www.bristol.ac.uk/acrc/.

References

[1] Srinivasan, G. R., "Computations of Two-Dimensional Airfoil-Vortex Interactions," Tech. Rep. 3885, NASA, 1985.

[2] Stumpf, E., "Study of Four-Vortex Aircraft Wakes and Layout of Corresponding Aircraft Configurations," *Journal of Aircraft*, Vol. 42, No. 3, 2005, pp. 722–730. https://doi.org/10.2514/1.7806, URL https://doi.org/10.2514/1.7806.

[3] Changfoot, D. M., Malan, A. G., and Nordström, J., "Hybrid Computational-Fluid-Dynamics Platformto Investigate Aircraft Trailing Vortices," *Journal of Aircraft*, Vol. 56, No. 1, 2019, pp. 344–355. https://doi.org/10.2514/1.C035022, URL https://doi.org/10.2514/1.C035022.

[4] Rossow, V., "Inviscid modeling of aircraft trailing vortices," Tech. Rep. 409, NASA, 1977.

[5] Smith, S., and Kroo, I., "Induced Drag Computations on Wings with Accurately MoModel Wakes," *Journal of Aircraft*, Vol. 34, No. 2, 2006, pp. 253–255.

[6] Chatelain, P., Curioni, A., Bergdorf, M., Rossinelli, D., Andreoni, W., and Koumoutsakos, P., "Billion vortex particle direct numerical simulation of aircraft wakes," *Comput. Methods Appl. Mech. Engrg*, Vol. 197, No. 13-16, 2008, pp. 1296–1304.

[7] Sitaraman, J., "CFD based unsteady aerodynamic modeling for rotor aeroelastic analysis," Ph.D. thesis, University of Maryland, 2003.

[8] Anusonti-Inthra, P., and Floros, M., "Coupled CFD and Particle Vortex Transport Method: Wing Performance and Wake Validations," *38th Fluid Dynamics Conference and Exhibit*, 2008.

[9] Zhao, j., He, C., Zhang, L., Zhao, H., and Hu, P., "Coupled Viscous Vortex Particle Method and Unstructured Computational Fluid Dynamics Solver for Rotorcraft Aerodynamic Interaction Analysis," *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2011.

[10] Pahla, A., Manickathan, L., Ferreira, C., and Van Bussel, G., "A hybrid eulerian-lagrangian flow solver," , 2015. ArXiv preprint arXiv:1505.03368.

[11] Wales, C., Gaitonde, A., and Jones, D., "Simulation of airfoil gust response using prescribed velocities," *IFASD*, 2011.

[12] Huntley, S., Jones, D., and Gaitonde, A., "2D and 3D gust repsonse using a prescribed velocity method," *46th AIAA Fluid Dynamics Conference*, 2016.

[13] Huntley, S. J., Jones, D., and Gaitonde, A., "Wingtip Vortex Preservation Using a Coupled Vortex Particle Method and Computational Fluid Dynamics Solver," *23rd AIAA Computational Fluid Dynamics Conference*, 2017. https://doi.org/10.2514/6.2017-4405, URL https://arc.aiaa.org/doi/abs/10.2514/6.2017-4405.

[14] Huntley, S. J., Jones, D. P., and Gaitonde, A. L., "Vortex Preservation Using Coupled Eulerian–Lagrangian Solver," *Journal of Aircraft*, Vol. 56, No. 2, 2019, pp. 457–468. https://doi.org/10.2514/1.C034875.

[15] Speck, R., "Generalized Algebraic Kernels and Multipole Expansions for Massively Parallel Vortex Particle Methods," Ph.D. thesis, Universität Wuppertal, 2011.

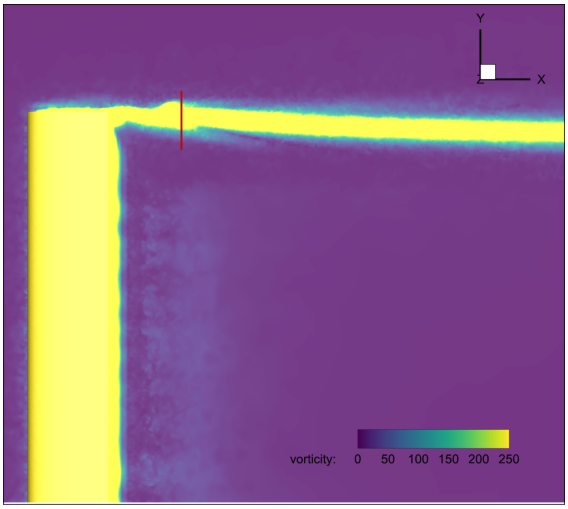
[16] Cottet, G., and Koumoutsakos, P. D., *Vortex Methods: Theory and Practice*, Cambridge Univ Pr, 2000.

[17] Wincklemans, G., and Leonard, A., "Contributions to Vortex Particle Methods for the computation of three-dimensional incompressible unsteady flows," *Journal of Computational Physics*, Vol. 109, 1993, pp. 247–273.

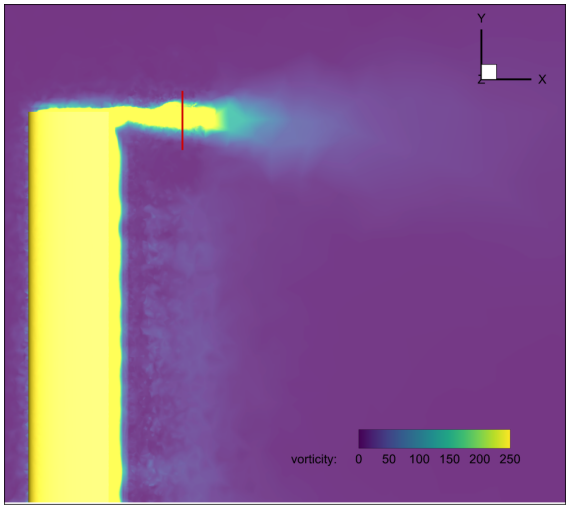
[18] Cheng, H., Greengard, L., and Rokhlin, V., "A fast adaptive multipole algorithm in three dimensions," *Journal of Computational Physics*, Vol. 155, 1999, pp. 466–498.

[19] Barba, L., Leonard, A., and Allen, C., "Advances in viscous vortex methods-meshless spatial adaptation based on radial basis function interpolation," *Int. J. Numer. Meth. Fluids*, Vol. 47, 2005, pp. 387–421.

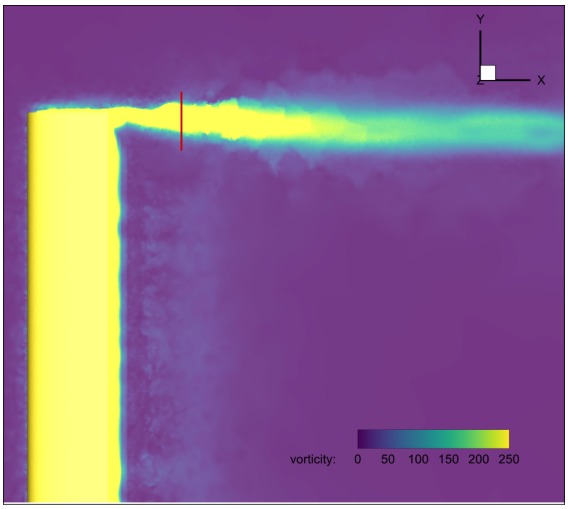
[20] Schwamborn, D., Gerhold, T., and Heinrich, R., "The DLR TAU-code: Recent applications in research and industry," 2006.



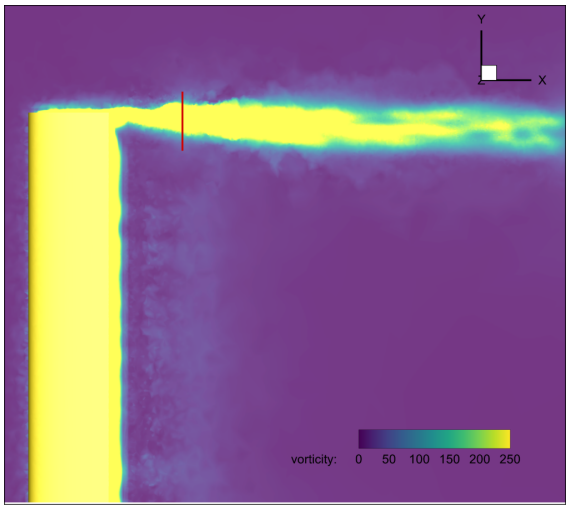
(a) CFD on fine mesh



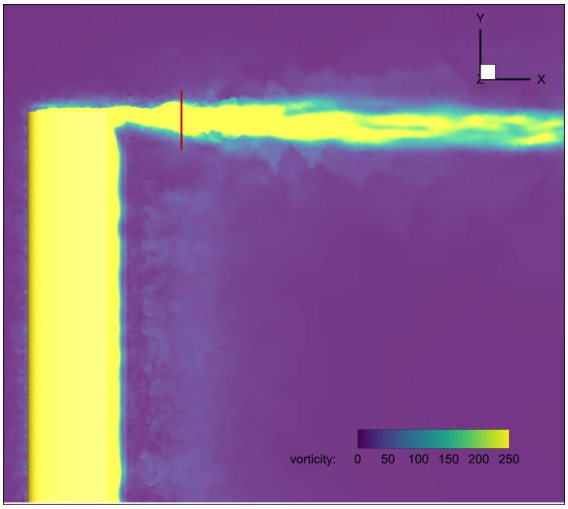
(b) CFD on coarse mesh



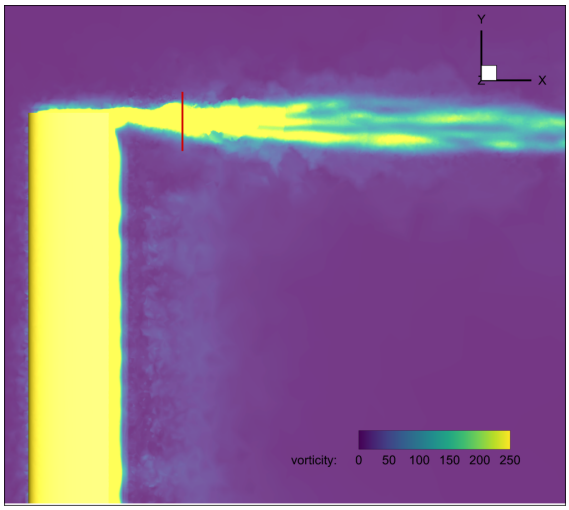
(c) CFD+VPM on coarse mesh, 121 particles per time step



(d) CFD+VPM on coarse mesh, 225 particles per time step

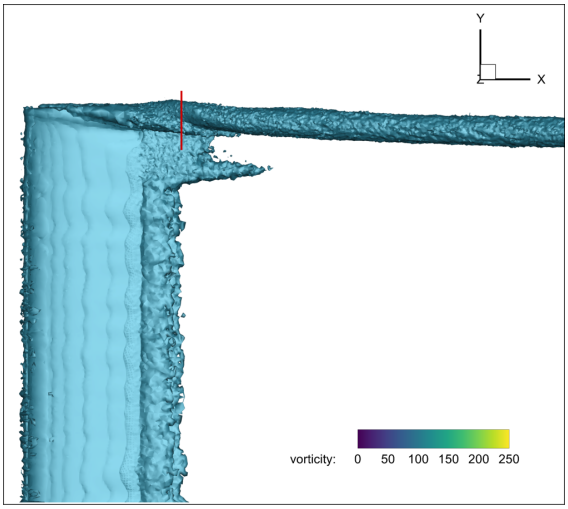


(e) CFD+VPM on coarse mesh, 484 particles per time step

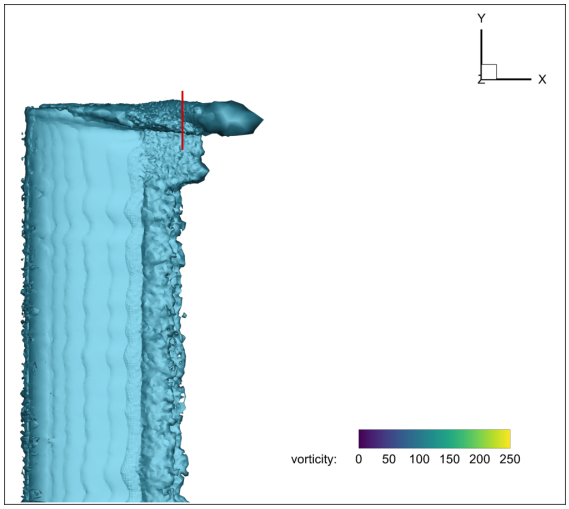


(f) CFD+VPM on coarse mesh, split seeding

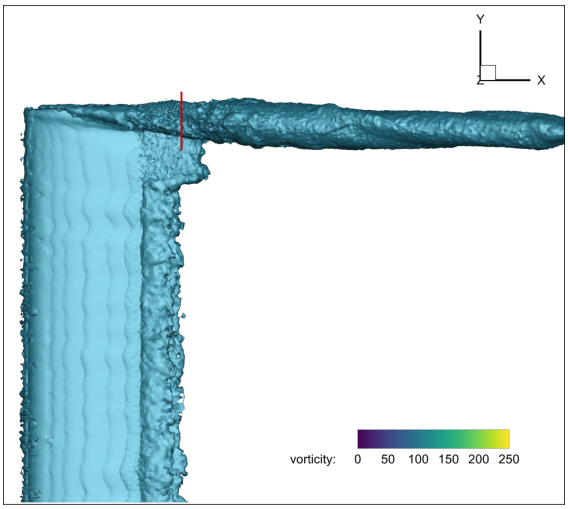
Fig. 4 Vorticity slice through the tip vortex calculated using CFD and VPM.



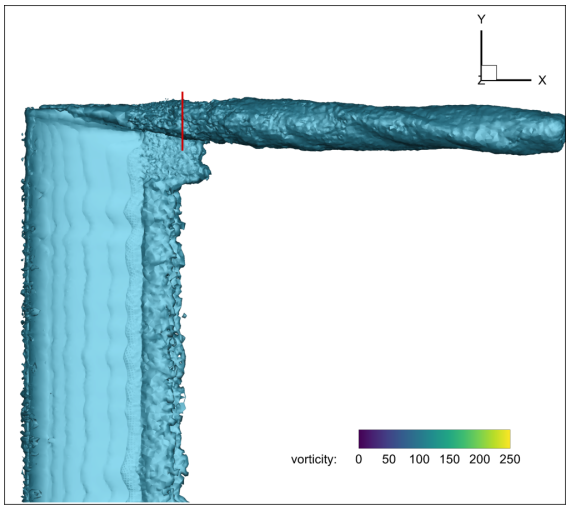
(a) CFD on fine mesh



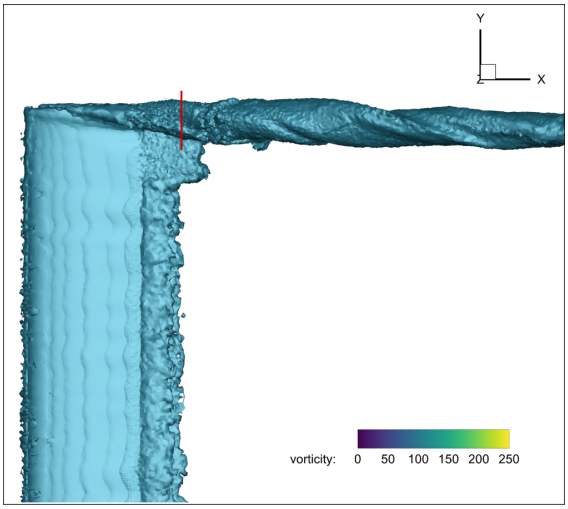
(b) CFD on coarse mesh



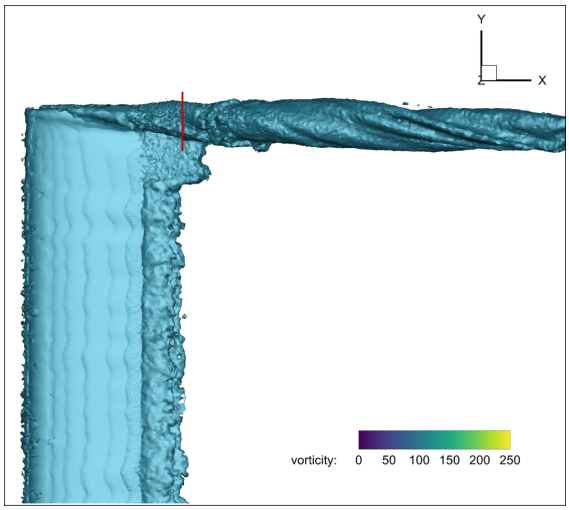
(c) CFD+VPM on coarse mesh, 121 particles per time step



(d) CFD+VPM on coarse mesh, 225 particles per time step

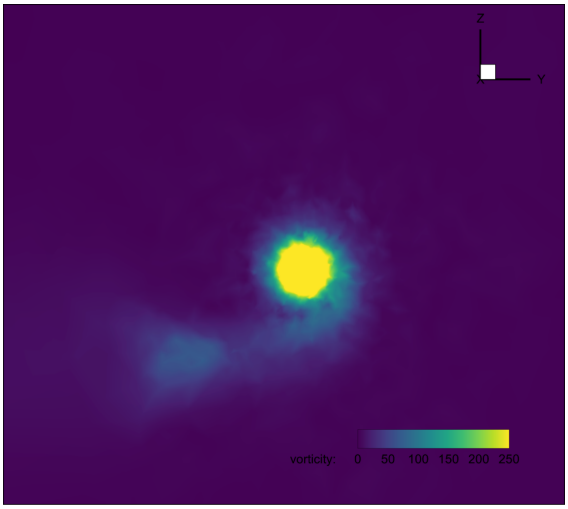


(e) CFD+VPM on coarse mesh, 484 particles per time step

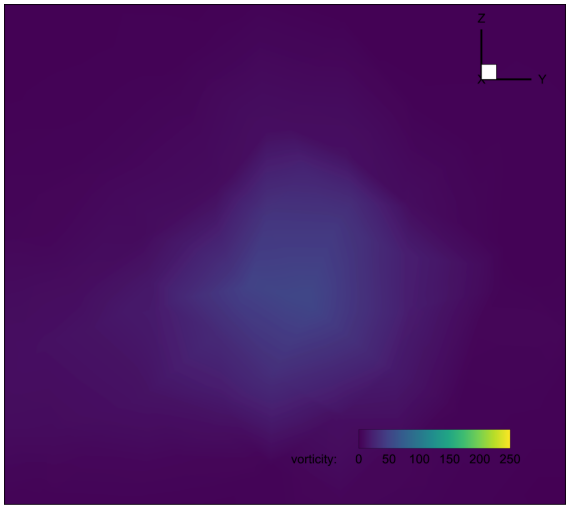


(f) CFD+VPM on coarse mesh, split

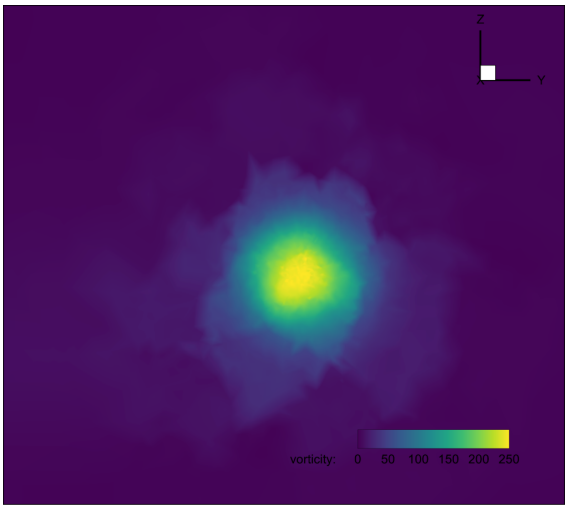
Fig. 5 Vorticity iso-surface calculated using CFD and VPM. Red line shows particle seeding location



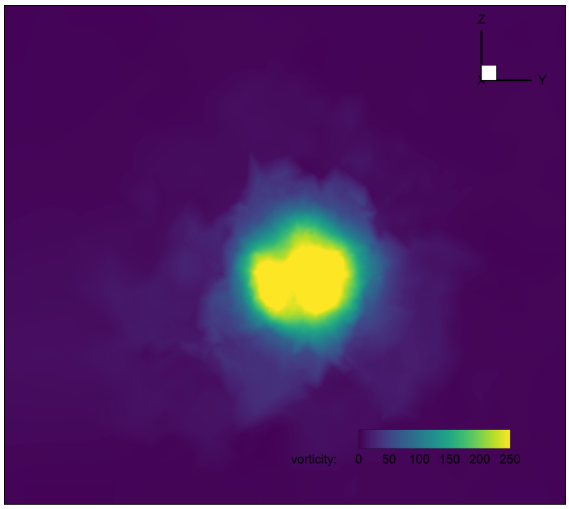
(a) CFD on fine mesh



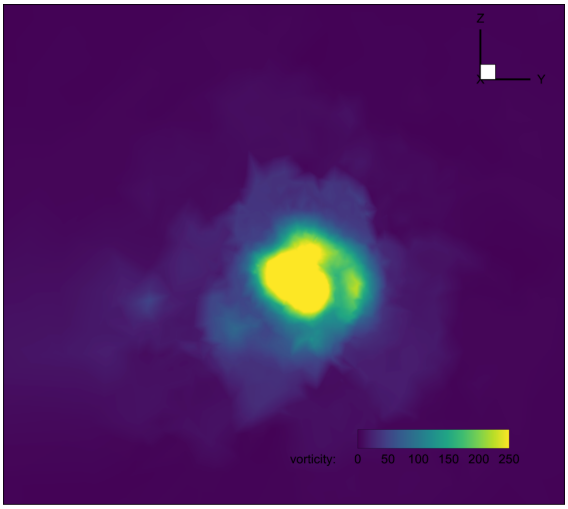
(b) CFD on coarse mesh



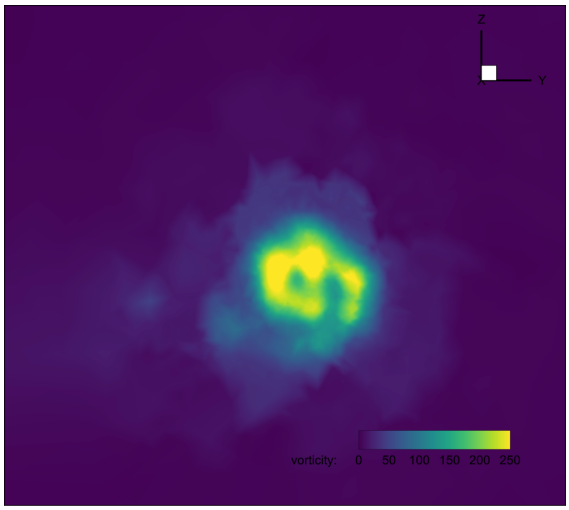
(c) CFD+VPM on coarse mesh, 121 particles per time step



(d) CFD+VPM on coarse mesh, 225 particles per time step

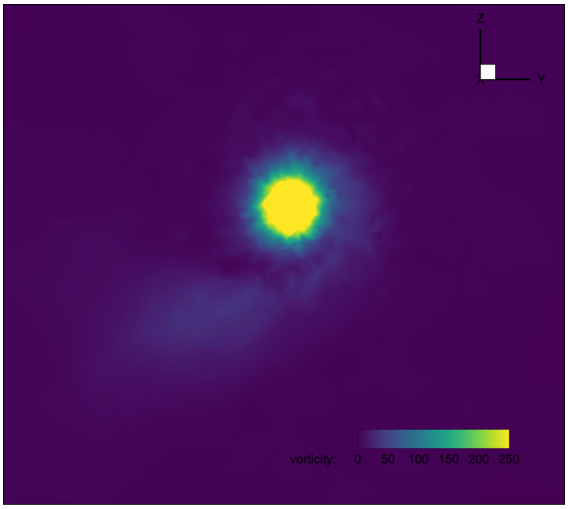


(e) CFD+VPM on coarse mesh, 484 particles per time step

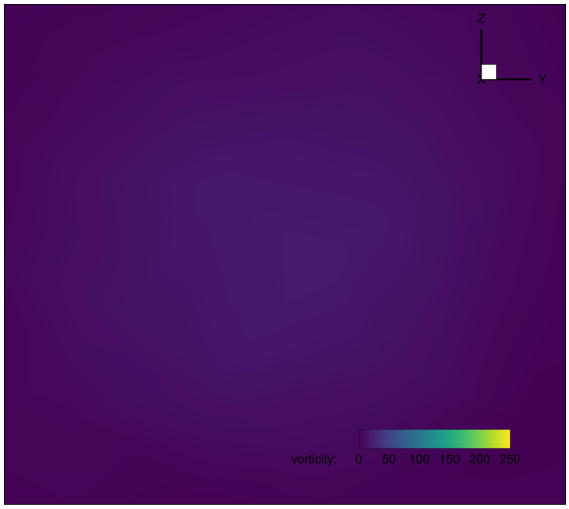


(f) CFD+VPM on coarse mesh, split

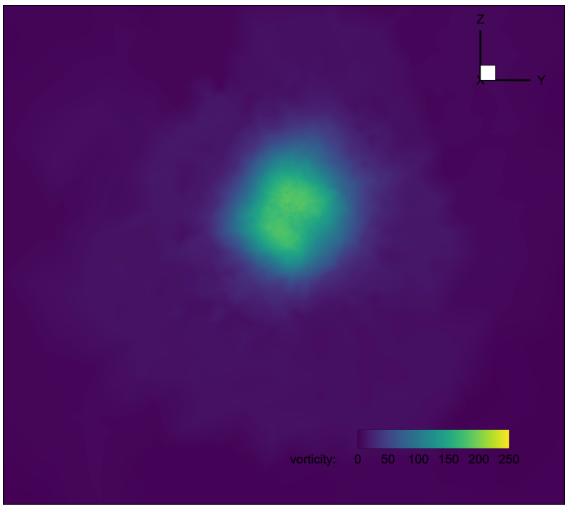
Fig. 6 Vorticity slice through wake 1.3c behind wing tip



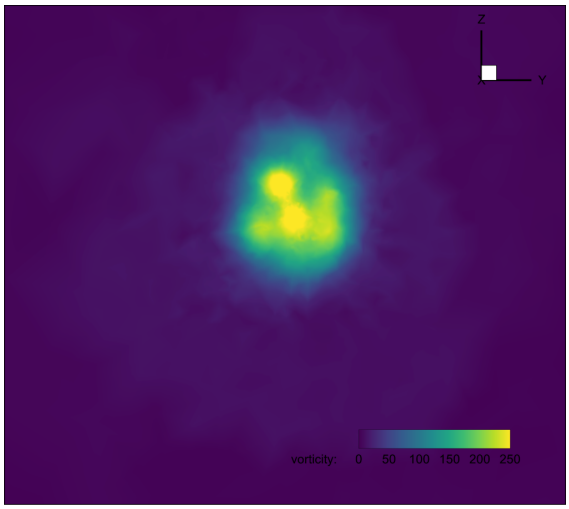
(a) CFD on fine mesh



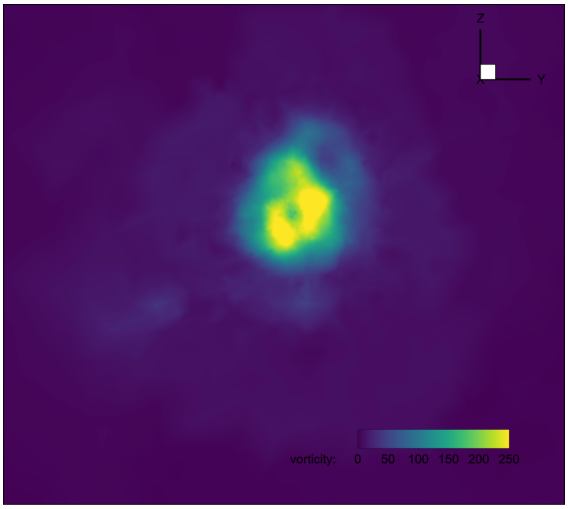
(b) CFD on coarse mesh



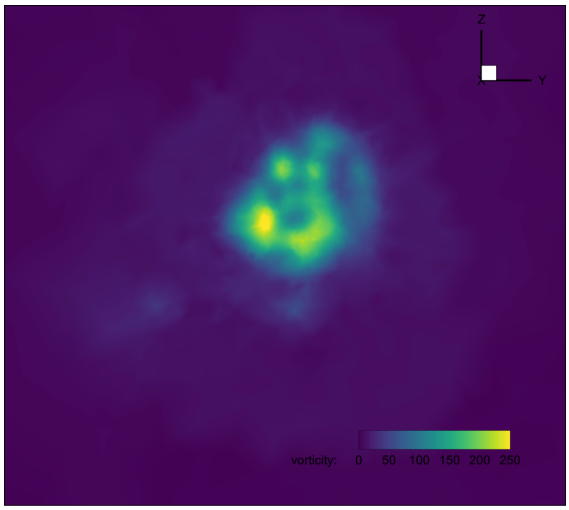
(c) CFD+VPM on coarse mesh, 121 particles per time step



(d) CFD+VPM on coarse mesh, 225 particles per time step



(e) CFD+VPM on coarse mesh, 484 particles per time step



(f) CFD+VPM on coarse mesh, split

Fig. 7 Vorticity slice through wake 2.3c behind wing tip