Wake modeling in complex terrain using a hybrid Eulerian-Lagrangian Split Solver

Franz G. Fuchs1, Adil Rasheed1, Mandar Tabib1 and Eivind Fonn1

1Applied Mathematics, SINTEF ICT, Strindveien 4, 7035, Trondheim, Norway
E-mail: {franz.fuchs, adil.rasheed, mandar.tabib, eivind.fonn}@sintef.no

Abstract. Wake vortices (WVs) generated by aircraft are a source of risk to the following aircraft. The probability of WV related accidents increases in the vicinity of airport runways due to the shorter time of recovery after a WV encounter. Hence, solutions that can reduce the risk of WV encounters are needed to ensure increased flight safety. In this work we propose an interesting approach to model such wake vortices in real time using a hybrid Eulerian-Lagrangian approach. We derive an appropriate mathematical model, and show a comparison of the different types of solvers. We will conclude with a real life application of the methodology by simulating how wake vortices left behind by an aircraft at the Værnes airport in Norway get transported and decay under the influence of a background wind and turbulence field. Although the work demonstrates the application in an aviation context the same approach can be used in a wind energy context.

1. Introduction and Related Work

Wake vortices (WVs) are a flow feature generated by all aircraft as they fly, land and take-off. They are created as a result of lift forces acting on the aircraft wings. WVs generated by an aircraft are very strong and represent a source of considerable danger to following aircraft. The probability of WV related accidents increases closer to the airport runway due to higher air traffic density and due to the interaction with the ground: WVs can rebound and linger on in the flight path vicinity. Hence, to ensure safety, a minimum distance must be maintained between two aircraft during landing/taking-off. This serves as a bottleneck to higher airport capacity and sometimes results in flight delays. In order to mitigate WV effects, it is important to study their evolution and decay rate and the factors influencing it. The most advanced systems employ the use of expensive LIDARS and RADARS [9]. These are emerging as reliable technologies but they are still very expensive and have limited range of applicability [11].

Numerical modelling of turbulent flows based on the Navier–Stokes equations (NSE), if accurate, can be a cost effective alternative and easily employable. Using the material derivative operator $\frac{D}{Dt} = (\partial_t + (\mathbf{u} \cdot \nabla))$ the incompressible NSE are given in terms of velocity $\mathbf{u}$ and pressure $p$ by

$$\frac{D\mathbf{u}}{Dt} = -\nabla p + \nu \nabla^2 \mathbf{u}, \quad \text{div}(\mathbf{u}) = 0,$$

(1)

or in terms of vorticity $\omega$ and velocity $\mathbf{u}$ by

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

(2)
where $\nu \geq 0$ is the kinematic viscosity. Note that the vorticity form does not involve a pressure term.

Generally, the types of solvers for the NSE can be categorized into Eulerian and Lagrangian. Eulerian solvers divide the computational domain into a mesh that is fixed in space, whereas Lagrangian solvers use nodes (often called “particles”) that move with the local velocity. Vortex Particle Methods (VPMs)\(^4\) are Lagrangian type solvers based on the vorticity form \(^2\) of the NSE. When comparing VPMs with Eulerian solvers based on the velocity-pressure form \(^1\) we have the following.

(i) **Pressure/Velocity Solver:** A dominating cost of Eulerian solvers is related to the pressure term, whereas VPMs need to calculate the induced velocity.

(ii) **Boundary Conditions:** Eulerian solvers are very well suited for accurately resolving flow structures near solid boundaries, but satisfying boundary conditions at infinity is a difficult task. The situation is reversed for VPMs.

(iii) **Numerical Diffusion:** Eulerian solvers tend to be diffusive and dampen high-intensity vortical structures very quickly. This can be alleviated by the use of high order methods and fine grids. However, this gives rise to large systems of equations that are expensive to solve. Conversely, Lagrangian solvers are intrinsically adaptive, and numerical diffusion is negligible in practice.

(iv) **Time Step Size for Explicit Discretizations of the Advection Terms:** In contrast to Eulerian type solvers, where the step size is limited by a CFL condition of the form $\Delta t \leq C \Delta x / |u|_{\infty}$, the stability constraint for VPMs is given by $\Delta t \leq D / |\nabla u|_{\infty}$. However, another stability constraint of VPMs requires particles to be remeshed in regular intervals. In \(^3\) it is shown that that a remeshed particle method is stable for $D < 1$.

The simulation of wake vortices generated by a wing/aircraft is a widely studied subject due to its importance in aviation. There is a huge variety of solvers for the NSE that can be applied in this case. An overview of solvers in European aircraft design from 2002 can be found in Vos et al.\(^16\). Eulerian type solvers for direct numerical simulation (DNS), Reynolds-averaged Navier-Stokes equations (RANS), and large eddy simulation (LES) can readily be applied to idealized and realistic wake vortex simulations (see e.g. \(^10\)). Concerning Lagrangian type solvers, Winckelmans et al.\(^17\) present VPMs for DNS and LES for the viscous and inviscid NSE in two and three dimensions, with efficient implementation on parallel computers and the interaction with the ground as the main aim. In \(^2\) Chatelain et al. introduce a high-resolution VPM for DNS in three dimensions. The focus of the paper is on efficient/parallel methods for the use of billions of particles. The computation of the operators for diffusion and vortex stretching, as well as the calculation of the induced velocity field, are performed on a regular grid. There are many cases where a hybrid Eulerian-Lagrangian method is computationally advantageous in terms of efficiency. A general introduction to hybrid solvers can be found in \(^4\), and for a recent paper we refer to Palha et al.\(^12\).

For real time simulation, high fidelity simulations of the whole domain of interest around an airport are prohibitive and hence an alternative approach is required. In order to predict the transport and decay of aircraft wake vortices with a model that allows for fast simulation, our approach is inspired by the “WAKE4D prediction platform”\(^5\). This approach consists of placing computational gates, i.e. 2D-planes, along the flight path of an aircraft landing/taking-off. As the aircraft passes such a computational gate, a vortex pair is initialized and its transport and decay is studied depending on the local background velocity field. However, our approach differs from WAKE4D in the following points. Firstly, we use a fully resolved vortex field, whereas WAKE4D aims at representing each primary and secondary vortex with one particle only. Secondly, we use a background velocity that stems from a high-fidelity RANS simulation and we allow our computational gates to move, by tracking the movement of the
vortices, resulting in a computationally efficient algorithm. Finally, we model the influence of
the background velocity field directly in the model equations.

The main objectives of the paper are as follows:

(i) Comparing mesh based and mesh-less methods for simulating the transport and diffusion
of wakes in an idealized environment.

(ii) Demonstrate the utility of a hybrid Eulerian-Lagrangian approach to real time simulation
of wake-vortex transport and diffusion in a complex terrain.

2. Model and Approach

In order to predict the transport and decay of aircraft wake vortices with a model that allows for
fast simulation, our approach tries to combine the advantages of Eulerian type solvers and VPMs
by decomposing the velocity field into the sum of two components: one that is time-dependent,
and one that is independent of time. This is motivated by the fact that the ambient velocity is
approximately constant in the timescales given by the generation of wake vortices by an aircraft
until those vortices have dissipated. The approach in this work is schematically represented by
Figure 1. Terrain data and boundary conditions (for wind, temperature and turbulence) from
a global model is preprocessed and provided to a mesoscale model which serves as the input for
a micro-scale model, see section 2.2. The microscale model is a finite element method (FEM)
based code. The computed background flow, temperature and turbulence fields are stored in
a data cube. The data is then processed for the use in the wake model, which is based on a
hybrid Eulerian-Lagrangian split solver. Additionally, flight paths and aircraft specifications
are utilized to initialize the wake vortices along the flight path at pre-decided 2D planes called
“gates”. We will start by describing the mathematical model.

2.1. The Model

In order to derive a valid model, we decompose the total velocity into \( \mathbf{u} = \mathbf{v} + \mathbf{\bar{u}} \), where \( \mathbf{\bar{u}} \) is
a time-averaged mean value of the atmospheric turbulence with \( \text{div}(\mathbf{\bar{u}}) = 0 \) and \( \mathbf{u}_t = 0 \), and
\( \mathbf{v} \) is the time dependent velocity induced by the aircraft with \( \text{div}(\mathbf{v}) = 0 \). A straightforward
derivation shows that this ansatz results in a PDE for the vorticity \( \mathbf{\omega} = \text{curl}(\mathbf{v}) \) as follows.

\[
\omega_t + ((\mathbf{v} + \mathbf{\bar{u}}) \cdot \nabla) \mathbf{\omega} = (\mathbf{\omega} \cdot \nabla) (\mathbf{v} + \mathbf{\bar{u}}) + \nu \nabla^2 \mathbf{\omega} + S(\mathbf{v}, \mathbf{\bar{u}}),
\]

with

\[
S(\mathbf{v}, \mathbf{\bar{u}}) = \text{curl}(\mathbf{\bar{u}}) \cdot \nabla (\mathbf{v} + \mathbf{\bar{u}}) - ((\mathbf{v} + \mathbf{\bar{u}}) \cdot \nabla) \text{curl}(\mathbf{\bar{u}}) + \nu \nabla^2 \text{curl}(\mathbf{\bar{u}}),
\]

and \( \text{div}(\mathbf{\bar{u}}) = \text{div}(\mathbf{v}) = 0, \mathbf{u}_t = 0 \),

(3)

together with appropriate initial and boundary conditions. Note that the source term \( S(\mathbf{v}, \mathbf{\bar{u}}) \)
disappears in case of a curl-free background velocity – in particular if \( \mathbf{\bar{u}} \) is identically zero. In 2
dimensions the equations simplify to
\[ \omega_t + ((v + \bar{u}) \cdot \nabla) \omega = \nu \nabla^2 \omega + S(v, \bar{u}) \]
\[ S(v, \bar{u}) = -((v + \bar{u}) \cdot \nabla) \text{curl}(\bar{u}) + \nu \nabla^2 \text{curl}(\bar{u}). \] (4)

Using this model, the current work can be broken down into two major computational steps – one for the background field and one for the wake vortices – described in the following.

2.2. Computation of Background Fields
The background fields are computed using a microscale Semi IMplicit Reynolds Averaged (SIMRA) code which is based on a FEM discretization of the mass, momentum and energy conservation equations. Turbulence is modelled using two equations: one for turbulent kinetic energy and another for turbulent dissipation. A projection method is used for the solution of the Reynolds equations, and a mixed FEM formulation is used for space discretization. A Taylor-Galerkin method is used for time discretization. A special feature of this model is the use of logarithmic element interpolation at the near-ground location in order to satisfy logarithmic boundary conditions accurately. The code has been parallelized using Message Passing Interface (MPI). A variant of this model is also being used for forecasting turbulence at 20 Norwegian airports at a horizontal resolution of 75m x 75m (13). This kind of spatial resolution is a prerequisite to the vortex-particle based method described next. In the current work the microscale model was forced using results from the mesoscale code HARMONIE based on AROME physics (15).

2.3. Computation of Transport and Decay of Wake-Vortices
For the time-dependent part, we employ a Eulerian-Lagrangian split solver in order to efficiently simulate the transport and decay of wake vortices. The VPM simulation only covers the relevant part of the domain, e.g., 2D or possibly 3D-slices along the flight path, similar to De Visscher et al. [5].

We split equation (4) into three parts – advection, source terms, and diffusion – as follows

\[ \omega_t + ((v + \bar{u}) \cdot \nabla) \omega = 0, \] (5a)
\[ \omega_t = S(v, \bar{u}), \] (5b)
\[ \omega_t = \nu \nabla^2 \omega. \] (5c)

The reason for this splitting is that advection can be solved very efficiently with a Lagrangian method – see Section 3 – and the remaining operators with Eulerian methods on regular meshes/grids. Furthermore, explicit discretizations of the diffusion term will potentially (depending on \( \nu \)) lead to much smaller time steps compared to the non-stiff source term.

For 3D-slices equation (3) is split in a similar way to equations (5a)-(5c). The additional term “\( \text{curl}(\bar{u}) \cdot \nabla (v + \bar{u}) \)” in \( S(v, \bar{u}) \) can be discretized in a straightforward manner with central differences. The extra term “\((\omega \cdot \nabla) (v + \bar{u})\)” related to vortex stretching can be discretized in a Eulerian or Lagrangian manner.

The main idea behind VPM is to approximate the continuous vorticity field by a linear superposition of discrete particles often called “blobs”

\[ \omega(x) \approx \omega_c(x) := \sum_p \Gamma_p \delta_c(x - x_p), \] (6)

where \( \Gamma_p \) are the particle strengths, \( x_p \) are the particle locations, and \( \delta_c \) is the reconstruction kernel. The reconstruction kernel is the convolution of the Dirac delta function with a mollifier with scaling parameter \( \varepsilon \).
After initializing the vorticity on a regular grid, the proposed method consists of the following sub-steps – depicted in Figure 2 – which are performed at each time level.

(i) **Populate:** In order to increase the efficiency of our algorithm, only vortex particles with a vorticity larger than a certain threshold are instantiated. A check is performed if the change to the total circulation/vorticity field is still acceptably small. This results in $N$ particles with position $x_p(t)$ and corresponding vorticity $\omega_p = \omega(x_p)$ for $1 \leq p \leq N$.

(ii) **Advection:** Using a Lagrangian formulation of equation (5a), the advection of the $p$th particle is simply given by the following ordinary differential equation

$$\frac{dx_p}{dt} = v(x_p) + \bar{u}(x_p).$$

(7)

In order to evaluate the right hand side, the pre-computed background velocity $\bar{u}$ has to be evaluated at the particle position in the global, three-dimensional domain and the velocity components with respect to the local coordinate system of the computational gate (2D plane or 3D rectangle) have to be extracted. Furthermore, the induced velocity $v$ at the point $x_p$ has to be computed from the vorticity field given by $\{x_q, \omega_q\}_{1 \leq q \leq N}$. To recover $v$ we use the Biot-Savart integral, which relates velocity to vorticity through

$$v = \int K(x - y) \times \omega(y) dy, \quad \text{with} \quad K(x) = \nabla g(x) = \begin{cases} -\frac{x^2}{2|\mathbf{x}|^2}, & d = 2, \\ -\frac{x}{4|\mathbf{x}|^3}, & d = 3. \end{cases}$$

(8)

After discretization we have that our vorticity field is given by

$$v(x_p) = \sum_{q=1}^{N} K_\varepsilon(x_p - x_q) \Gamma_q,$$

(9)

where the kernel $K_\varepsilon$ is the convolution of $K$ with a mollifier with regularization parameter $\varepsilon$. For more details we refer to [4]. Clearly a naive algorithm to recover the velocity at all points has the complexity $O(N^2)$, which becomes prohibitive for large $N$. However, the Fast Multipole Method (FMM) [8] can be directly employed here. FMM reduces the complexity to $O(\log(1/\delta)N)$, where $\delta$ is a given tolerance. There are many software packages providing implementations of the FMM on the CPU and the GPU. In this paper we decided to use a black box solver developed by Fong and Darve [7].

To advance in time, a standard Runge-Kutta (RK) method can be employed with time step $\Delta t_i = D/\max_p(\nabla(v_p + \bar{u}_p))$. The gradient of the velocity fields is easy to compute, since...
the particle locations are on a regular grid in the first RK step. In addition to the gradient, we can easily compute (and store) the minimum and maximum values of the background velocity field $\bar{u}$, needed in the next substep. In the numerical examples presented in this paper we have used a second order RK method with Courant number $D = 0.75$.

(iii) Re-Grid: Due to clustering and dispersion the Lagrangian grid becomes distorted over time. A necessary condition for convergence of VPMs is that the particle blobs overlap $[1]$, i.e. that $\varepsilon > h$, where $h$ is the maximum distance between particles and $\varepsilon$ is the blob size. In order to satisfy this condition, it is necessary to remesh/re-grid the distorted particle field at regular intervals. This can be done by interpolating the field values from the distorted grid onto a uniform one. This uniform grid has to take the effect of the background flow into account, something we call gate control. This gate control consists of moving and extending the regular grid of the previous time step according to the minimum and maximum values of the background velocity field, which has already been calculated in the previous step. Let $\tilde{\omega}_q$ be the values at the distorted grid points at $\tilde{x}_q$, then the new values $\omega_p$ at $x_p$ can be determined by

$$
\omega_p = \sum_q \tilde{\omega}_q W \left( \frac{x_p - \tilde{x}_q}{h} \right).
$$

(10)

In this paper we use a 6th order interpolator given by

$$W(\xi) = \frac{1}{\pi} \theta(\xi_1) \theta(\xi_2) e^{-|\xi|^2}, \quad \text{where} \quad \theta(s) = \frac{1}{2} \left( \frac{15}{4} - 5s^2 + s^4 \right).$$

(11)

Eldredge et al. [6] found $h = 1.7 \Delta x$, to be a good value, where $\Delta x$ is the particle distance. Re-gridding at every time step allows to perform the diffusion and source operators very efficiently on a regular grid.

(iv) Source: The right hand side of equation (5b) is evaluated with central differences. Initial conditions are used from the re-gridded values. A first order explicit Euler method is used to integrate from $t_i$ to $t_{i+1} = t_i + \Delta t_i$. This serves as initial condition for the diffusion sub-step.

(v) Diffusion: We use a standard second order discretization (in space and time) in order to approximate the solution of equation (5c). An explicit second order RK method limits the time step to be $\Delta t_{\text{diff}} = \frac{\Delta x^2}{\nu}$, where $\nu > 0$ and $\Delta x$ is the grid distance (equal in all directions). Hence, we take the appropriate number of substeps in order to integrate from time $t_i$ to $t_{i+1} = t_i + \Delta t_i$. For large $\nu$ or small $\Delta x$ this number becomes prohibitive, and an implicit scheme should be used.

This algorithm is implemented using C++. At the moment it is not exploiting parallelism. We will continue by comparing this algorithm in an idealized setting before we show its capabilities on a realistic case.

3. Numerical Results

To demonstrate the efficiency and applicability of the proposed approach we start by conducting a comparison with a Eulerian FEM solver (see section 2.2) in the case of vanishing background velocity. This is followed by an example illustrating the effect of the background velocity. The example indicates furthermore how the proposed approach can be applied in a wind turbine wake application. Finally, we showcase a realistic scenario where an aircraft lands in the vicinity of the Værnes airport in Norway.
3.1. Comparison of Hybrid VPM with Eulerian Solver

In order to compare different numerical methods, we use the so-called Lamb-Oseen vortex as initial condition, since an analytical solutions of the NSE exists for this case. In terms of the vorticity $\omega$ and radius $r$ the initial condition is given by

$$\omega(r,0) = \frac{\Gamma_0}{\pi \lambda_0^2} e^{-r^2/\lambda_0^2},$$

and the analytical solution of the NSE is given by

$$\omega(r,t) = \frac{\Gamma_0}{\pi (\lambda_0^2 + 4\nu t)} e^{-r^2/(\lambda_0^2 + 4\nu t)}.$$  \hfill (13)

In the absence of viscosity the Lamb-Oseen vortex presents a steady state solution. We would like to point out that this examples involves advection, but the advection is along iso-lines circular around the center of the vortex. In the following we use $\Gamma_0 = 1$, $\lambda_0 = 0.15$, and viscosity $\nu = 0.01$. Figure 3 shows how the initial vortex dissipates with increasing time. It can be seen that the hybrid VPM approximates the analytical solution very well using only $100 \times 100$ particles.

Next, we compare the efficiency of proposed VPM to a Eulerian FEM method (see section 2.2). Figure 4 shows how computational time increases with decreasing $L^1$-error (linked to increasing refinement level) for both the VPM and the FEM method. Firstly, we can observe that the error constant is smaller for the VPM. Secondly, the slope of VPM is steeper than the one of FEM. A possible explanation for this is that the step size for VPM is independent of the refinement level, but the number of time steps increases with the refinement level for explicit FEMs. Note, that the hybrid VPM uses an explicit second order discretization of the diffusion term. Furthermore, the plot shows that the VPM uses an $O(N)$ algorithm to compute the velocity field.
Figure 4. Efficiency plot for the Lamb-Oseen vortex with $\nu = 0.01$ at time $t = 1$ comparing an Eulerian FEM method (see section 2.2) to the proposed hybrid VPM.

Figure 5. Example illustrating the effect of “side wind” on the vorticity with $\nu = 0$.

3.2. The Effect of Background Velocity

In this subsection, we show how the proposed model can be used in the setting of wind turbine wakes, illustrating the effect of a non-zero background velocity field in equation (4). To this end, we use the following (divergence-free) background field

$$\mathbf{u}(x, y) = \begin{cases} 
(u, 0)^T, & \text{if } y \geq L, \\
(u(2y/L - y^2/L^2), 0)^T, & \text{if } 0 < y < L.
\end{cases} \quad (14)$$

Note that this in fact represents an analytic steady-state solution of the NSE, for the case of laminar flow between two flat “plates” or surfaces, one moving with zero velocity, the other with velocity $L$. This function mimics a laminar boundary layer, see black arrows in Figure 5(b).

Figure 5 shows the result of a simulation with the proposed hybrid VPM. The model parameters are $L = 1$, $U = 1/4$, zero viscosity $\nu = 0$, and equation (14) is shifted by $-1$ in the $y$-direction in order to fit the computational domain $(x, y) \in [-1.5, 1.5] \times [-1, 1]$. The initial vorticity is again given by equation (12) with $\Gamma_0 = 1$, $\lambda_0 = 0.15$. The simulation shown in the figure uses $300 \times 150$ particles/blobs. As expected, the figure reveals that the initial Lamb-Oseen vortex is advected with the local flow velocity.

3.3. Demonstration for Wake Modeling Close to the Værnes Airport

Finally, we showcase the hybrid VPM described above in a real-life scenario. Figure 6 shows the background field computed by the approach described in section 2.2 (see 13 for details), as well as a flight path and computational gates with initial vortices. The FEM uses an adaptive grid with $300 \times 300 \times 40$ grid points. On each gate the computation using the VPM with $200 \times 200$ particles was performed in near real time (computational and physical time are $O(\text{min})$) on one core on a standard desktop. The figure shows how the gates move with the background velocity, how the vortices interact and move downwards, and how the vortex strength decays over time.
3.4. Extending the Concept to Modelling Turbine Wakes

Figure 6 shows the flow structure on the leeward side the NREL 5MW turbine. The simulation was conducted using a Sliding Mesh Interface approach using a $k - \epsilon$ turbulence model. The simulation was run on 256 cores of a 2.6GHz Intel(R) Xeon(R) CPU machine. For one full rotation it took almost seven days. More detail about the simulation and conclusions can be found in [14]. Because of the significant computational cost associated with such simulations, a complete study of the evolution of such wakes at a wind farm scale is still conducted using simplified models like actuator line or actuator disk. The current work presents a computationally efficient alternative. Although, its applicability has been demonstrated in an aviation context, work is on to apply it to modelling wind turbine wakes. The idea is to choose a certain number of 2D planes (two sample planes are shown in Figure 7(b)) and initialize them with the vorticity data either through real time LIDAR measurement or from the results of high fidelity simulations. A VPM is then used to simulate the evolution of wakes using the background wind field, which is computed using a meso-micro coupled approach. The 3D picture as a function of time will be possible to generate by interpolating the 2D results.

4. Conclusion

In this paper we have presented a new methodology allowing real time prediction of trailing aircraft wake vortices in the vicinity of airports. This new approach combines high-fidelity background fields with high-fidelity real time wake simulations. Work is in progress to extend the approach to three dimensions and to incorporating solid (no-slip) boundary conditions. In addition, we plan to apply the developed approach to wind energy applications; in particular for analyzing terrain and wake effects on wind farm performance.
(a) Vertical side plane.
(b) Two sample planes on which the vorticity can be computed with a VPM.

Figure 7. Wind turbine wakes computed using a sliding mesh interface.

References
[1] J. T. Beale, *A convergent 3-d vortex method with grid-free stretching*, Mathematics of computation, 46 (1986), pp. 401–424.
[2] P. Chatelain, A. Curioni, M. Bergdorff, D. Rossinelli, W. Andreoni, and P. Koumoutsakos, *Billion vortex particle direct numerical simulations of aircraft wakes*, Computer Methods in Applied Mechanics and Engineering, 197 (2008), pp. 1296–1304.
[3] G.-H. Cottet, J.-M. Etancelin, F. Péringuey, and C. Picard, *High order semi-lagrangian particle methods for transport equations: numerical analysis and implementation issues*, ESAIM: Mathematical Modelling and Numerical Analysis, 48 (2014), pp. 1029–1060.
[4] G.-H. Cottet and P. D. Koumoutsakos, *Vortex methods: theory and practice*, Cambridge university press, 2000.
[5] I. De Visscher, G. Winckelmans, T. Lonfils, L. Bricteux, M. Duponcheel, N. Bourgeois, et al., *The wake4d simulation platform for predicting aircraft wake vortex transport and decay: Description and examples of application*, AIAA Paper, 7994 (2010), p. 2010.
[6] J. D. Eldredge, T. Colonius, and A. Leonard, *A vortex particle method for two-dimensional compressible flow*, Journal of Computational Physics, 179 (2002), pp. 371–399.
[7] W. Fong and E. Darve, *The black-box fast multipole method*, Journal of Computational Physics, 228 (2009), pp. 8712–8725.
[8] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, Journal of computational physics, 73 (1987), pp. 325–348.
[9] F. Holzäpfel, T. Gerz, F. Köpp, E. Stumpf, M. Harris, R. I. Young, and A. Dolfi-Bouteubre, *Strategies for circulation evaluation of aircraft wake vortices measured by lidar*, Journal of Atmospheric and Oceanic Technology, 20 (2003), pp. 1183–1195.
[10] F. Holzäpfel, T. Hofbauer, T. Gerz, and U. Schumann, *Aircraft wake vortex evolution and decay in idealized and real environments: methodologies, benefits and limitations*, Springer, 2002.
[11] F. Holzäpfel and M. Steen, *Aircraft wake-vortex evolution in ground proximity: analysis and parameterization*, AIAA journal, 45 (2007), pp. 218–227.
[12] A. Palha, L. Manickathan, C. S. Ferreira, and G. van Bussel, *A hybrid eulerian-lagrangian flow solver*, arXiv preprint arXiv:1505.03368, (2015).
[13] A. Rasheed and K. Srl, *A multiscale turbulence prediction and alert system for airports in hilly regions*, Aerospace Conference, ISBN: 978-1-4799-558-4 (2016), pp. 1–10.
[14] M. S. Siddiqui, A. Rasheed, M. Tabib, and K. Trond, *Numerical analysis of nrel 5mwe wind turbine: A study towards a better understanding of wake characteristic and torque generation mechanism*, Journal of Physics Conf. Series, Accepted (2016).
[15] P. Unden and L. E. a. Rontu, *HIRLAM-5 Scientific Documentation*, Scientific Documentation, 2002.
[16] J. Vos, A. Rizzi, D. Darraqq, and E. Hirschel, *Navier–stokes solvers in european aircraft design*, Progress in Aerospace Sciences, 38 (2002), pp. 601–697.
[17] G. Winckelmans, R. Cocle, L. Dufresne, and R. Capart, *Vortex methods and their application to trailing wake vortex simulations*, Comptes Rendus Physique, 6 (2005), pp. 467–486.