Advanced Medium-Order Modelling of a Wind Turbine Wake with a Vortex Particle Method Integrated within a Multilevel Code

Joseph Saverin∗, David Marten∗, George Pechlivanoglou∗, Christian Oliver Paschereit∗

∗ Chair of Fluid Dynamics, Hermann-Föttinger-Institut, Technische Universität Berlin, Müller-Breslau-Str. 8, 10623 Berlin, Germany
E-mail: j.saverin@tu-berlin.de

Abstract. The current paper describes an aerodynamic model for treatment of wind turbine wakes. For accurate treatment of the wake evolution for the near wake, along with interaction with local flow features, a model with low numerical diffusion has been chosen, a vortex particle method, which inherently allows treatment also of shearing effects and viscous diffusion. Treatment of blade loading is facilitated with the use of a lifting-line model. Details of correct specification of distributed and shed vortical elements in the blade wake are provided. Reduction of the computation cost has been achieved by implementing the model within a multilevel framework. In addition the model has been highly parallelised, so that relatively quick simulations at high fidelity can be achieved on the order of seconds. The ability of the model to produce results of comparable accuracy to CFD is demonstrated by comparison to the MEXICO test rotor.

1. Introduction
As the wind energy industry tends towards larger turbine diameters and more flexible blades, the ability of the designing engineer to be able to make quick, reliable predictions of the impact of a wake shadow becomes increasingly important. A method is desired which allows for quick, qualitatively accurate velocity deficit profiles which include dominant wake structures. The simulation of a wind turbine wake has a rich and diverse history, beginning with the application of simplified momentum methods in the 1980’s [1, 2]. These methods however do not treat unsteadiness or terrain interaction. The problem was then investigated by using linearised Reynolds-Averaged Navier Stokes finite volume solvers [3, 4]. Although undeniably more accurate, these methods, and in general finite volume simulation methods are still very computationally expensive. Higher order aerodynamic treatments have been introduced, such as the use of Large-Eddy Simulation (LES) turbulence modelling, however the computational expense is still exorbitant and in no way amenable to a design environment. Notable efforts have been made in the calculation of the general motion or meandering of the wake, including passive tracer methods such as the dynamic wake meandering model (DWM) [5]. An acceptable medium-order solution to the problem appears to be vorticity based methods, which allow a Lagrangian treatment of the flow field [6]. These methods are also amenable to acceleration not only by their inherent formulation by separating near and far-field influence, but also as they
allow simple formulation within a parallel computation framework. The application of a vortex particle (VP) method has been chosen for the analysis here as it allows for all important wake features to be inherently treated [7]. This includes unsteadiness, trailing and bound vorticity, wake self-induction, vorticity evolution, and interaction with boundaries [8]. Furthermore, the ability of the method to include viscous interaction and therewith shear layers is allowed by making use of the particle strength exchange (PSE) scheme [9]. Although this is not treated in this work, future work shall allow for both laminar and turbulent shear stresses to be included in the wake modelling [10, 11]. In addition, opportunity exists for the coupling of VP methods to those outlines earlier.

High fidelity resolution of a turbulent velocity field requires a very large particle count $N$, however the purpose of this paper is to demonstrate that relatively accurate results can be achieved with an accelerated method and a reasonable particle count ($N < 20K$) for the near wake. The extension to the far wake is intended for future work. Paramount to the validity of the results here is to stress that the method has been implemented within a multilevel method, which has been shown in previous works to reduce the computational expense of the vortex dynamic evaluation from quadratic evaluation cost order $O(N^2)$ to linear $O(N)$ [12, 13]. The calculations carried out here were accomplished on a desktop computer, and required calculation times on the order of seconds. The application of these methods should hence act to strengthen the arguments for current design methodology to advance beyond simplified methods such as the blade element momentum (BEM) model. The paper proceeds by initially outlining the physical model used, followed by the application of the multilevel method. The results are then presented which provide a comparison to experimental wake measurements.

2. Vortex Particle Dynamics

The equation set to be solved is arrived at by assuming an incompressible fluid. The velocity $\vec{u}$ (and by definition) the vorticity $\vec{\omega}$ fields are hence divergence free:

$$\nabla \cdot \vec{u} = 0, \quad \nabla \cdot \vec{\omega} = 0$$

(1)

By taking the curl of the Navier-Stokes equation, one arrives at the vorticity transport equation:

$$\frac{d\vec{\omega}}{dt} = \frac{\partial \vec{\omega}}{\partial t} + (\vec{u} \cdot \nabla)\vec{\omega} = (\vec{\omega} \cdot \nabla)\vec{u} + \nu \nabla^2 \vec{\omega}$$

(2)

These constitute a solvable equation set to trace the evolution of vorticity in the fluid. We proceed by using quadrature to treat the vorticity in the fluid as a set of vortex particles, whose position and strength are integrated in a Lagrangian sense [7]. In this method, the vorticity field is expressed as the sum of a set of discrete vortex particles:

$$\vec{\omega}(\vec{x}) = \sum_p \vec{\omega}_p \text{vol}_p \delta(\vec{x} - \vec{x}_p) = \sum_p \vec{\Gamma}_p \delta(\vec{x} - \vec{x}_p)$$

(3)

Where $\text{vol}_p$ is the representative volume of region occupied by the particle with vorticity vector $\vec{\omega}_p$. We introduce here the normalized distance from the particle $\rho = (\vec{x} - \vec{x}_p)/\sigma = \vec{r}/\sigma$ where $\sigma$ is the coresize of the particle and focus upon symmetric regularization functions, for which $\zeta = \zeta(||\vec{r}||/\sigma) = \zeta(\rho)$. The choice of regularization function is dependent upon the type of analysis desired, along with the given application. A collection of regularization functions for 2D and 3D applications is provided in [10]. Making use of the regularized Biot-Savart kernel $K_\sigma$ gives for the velocity field $\vec{u}_\sigma(\vec{x})$:

$$\vec{u}_\sigma(\vec{x}) = -\sum_p \frac{q_\sigma(\rho)}{||\vec{x} - \vec{x}_p||^3}(\vec{x} - \vec{x}_p) \times \vec{\Gamma}_p = \sum_p K_\sigma \times \vec{\Gamma}_p$$

(4)
Where \( q_\alpha \) depends upon the regularization function \( \zeta : q_\alpha(\rho) = \int_0^\rho \zeta(t) t^2 \, dt. \)

**Particle Evolution**

The substantial derivative of particle position is given by Eq. 4. Furthermore, neglecting viscous effects in the fluid, Eq. 2 allows the evolution of particle strength (referred to as stretching) to be modeled. The evolution equations are given by:

\[
\frac{d}{dt} \vec{x}_p = \vec{u}(\vec{x}) ; \quad \frac{d}{dt} \vec{r}_p = \text{vol}_p \frac{d}{dt} \vec{\omega}_p = \text{vol}_p (\vec{\omega}_p \cdot \nabla) \vec{u}(\vec{x}_p) \tag{5}
\]

This above equation for stretching is referred to as the classical scheme of particle strength evolution. By manipulating the velocity gradient tensor \((\vec{u} \cdot \nabla)\) [14], the evolution of the vorticity field can be equivalently recast into the transpose or mixed schemes:

\[
\frac{d}{dt} \vec{\omega} = (\vec{\omega} \cdot \nabla^T) \vec{u}(\vec{x}_p) \quad \text{(Transpose)} \quad \frac{d}{dt} \vec{\omega} = \frac{1}{2} (\vec{\omega} \cdot (\nabla + \nabla^T)) \vec{u}(\vec{x}_p) \quad \text{(Mixed)} \tag{6}
\]

It should be noted that although the above three formulations are equivalent for a continuous vorticity distribution \( \vec{\omega} \), they are not necessarily equivalent for a discrete velocity field \( \vec{\omega}_s \).

### 3. Multilevel Method

The development of numerical techniques to speed up the calculation of long-range influence has progressed rapidly in the last 20 years [15, 16, 17]. The basic concepts of the multilevel method have been outlined in Brandt et al. [17, 18]. For a full description of the methodology applied here, the reader is referred to van Garrel [12]. The method relies upon representation of the domain of interest with increasing coarser spatial meshes. The space is organized into an octree type discretisation where each parent branch contains the children elements of the next lower branch level. The source values at the lowest grid level are interpolated onto a mesh of representative grid nodes \( \vec{x}_{g,i} \), distributed throughout the containing domain region, or *box*. Considering first the influence of the nodes in a single source box on the nodes in a single receiver box, the influence of the kernel \( K \) at receiver points is approximated through the kernel values calculated between the grid nodes contained in both boxes \( \tilde{K} \) and a resulting interpolation error \( \epsilon \):

\[
K(\vec{x}_i, \vec{x}_j) = \tilde{K}(\vec{x}_i, \vec{x}_j) + \epsilon \approx \sum_i \sum_j K(\vec{x}_{g,i}, \vec{x}_{g,j}) L_j(\vec{x}_{g,j}) L_i(\vec{x}_{g,k}) + \epsilon \tag{7}
\]

Here \( L(\vec{x}) = L(x_{i,1}, x_{i,2}, \cdots) \) is a mapping function which interpolates source strength with \( L_i(\vec{x}) \) to the source box grid nodes (see Fig. 2, and influence \( \phi(x_i) \) is interpolated with \( L_j(\vec{x}) \) from the surrounding receiver grid nodes. The influence on a single point can be given as:

\[
\phi(\vec{x}_j) = \sum_i \sum_j \sum_k K(\vec{x}_{g,i}, \vec{x}_{g,j}) L_j(\vec{x}_{g,j}) L_i(\vec{x}_{g,k}) \phi(\vec{x}_i) \tag{8}
\]

Which can be arbitrarily reordered to:

\[
\phi(\vec{x}_j) = \sum_i L_j(\vec{x}_{g,j}) \left\{ \sum_j K(\vec{x}_{g,i}, \vec{x}_{g,j}) \left( \sum_i L_i(\vec{x}_{g,k}) \phi(\vec{x}_i) \right) \right\} \tag{9}
\]

The outer summation is referred to as the interpolation step. The innermost summation is the adjoint interpolation [19] where source strengths are used with the adjoint interpolation...
matrix to determine interpolation node strengths and the process in curly brackets represents the evaluation of the influence of each source grid node on each receiver grid node. The process of anterpolation can be carried out recursively into higher branches, so that progressively courser representations are considered, as illustrated in Fig. 1. These are then anterpolated up to the highest branch, evaluations are carried out, and interpolated down to the lower branch order. As described in van Garrel [12], this is sufficient for smooth kernels, however the evaluation of the near field for asymptotically smooth kernels (as are most kernels of interest in flow problems), the interpolation error is prohibitively large. This is overcome by evaluating the near-field of the base branch level directly.

Figure 1: Box heirarchy for a given evaluation point (receiver node) located on the blade midspan. The increasingly courser grid treatment in the far field is visible.

3.1. Algorithm Overview
The implementation follows the progressive application of Eq. 9, and are carried out as follows:

(i) **Base anterpolation:** Source strengths $\Gamma(\vec{x}_i)$ are anterpolated onto grid nodes $\Gamma_g(\vec{x}_{i,g})$ which occupy containing box.

(ii) **Branch anterpolation:** Grid node strengths for each child box are anterpolated onto parent box grid nodes.

(iii) **Course-grid summation:** The interaction between boxes, or more specifically, between the grid nodes of these boxes are calculated.

(iv) **Branch interpolation:** Influence at parent grid node points are interpolated to children grid nodes.

(v) **Base interpolation:** Influence at base boxes $\phi(\vec{x}_{g,j})$ are interpolated to evaluation nodes $\phi(\vec{x}_j)$.

(vi) **Near-field evaluation:** Near field sources are evaluated directly.

This process is diagrammatically illustrated in Fig. 2.

3.2. Vortex Particle Multilevel Method (VPML)
In the work here the multilevel method has been applied to treat the influence due to a set of vortex particles. This is accomplished by applying the steps of the above algorithm to the particle distribution. A vortex particle is characterised by the given circulation vector $\vec{\Gamma}_i$, which is anterpolated onto higher grid levels. Significant reduction in calculation time is achieved by observing that each box-grid node hierarchy has identical geometry. This allows the entire course grid influence calculation to be expressed as a matrix product, as detailed in the work by...
van Garrel [12]. The influence calculation however is naturally a function of the source particle coresize \( \sigma \), as seen in Eq. 4. At sufficient distance from the source point however, the influence can be treated as for a singular \( (q_s = 1) \) particle as validated in Saverin et al. [13]. The influence of the vortex stretching in the coarse grid step is accomplished with the use of the properties of the interpolation function. Barycentric Langrangian interpolation has been used here [20], which allows the spatial derivatives on the grid nodes to be determined. This quantity is interpolated to the position of the receiver nodes in order to quantify the velocity gradient tensor there and evaluate the stretching term as given in Eqs. 5, 6.

4. Comparative Study
The turbine chosen for the experimental comparison here is the MEXICO (Model EXperiments In COntrolled conditions) research turbine [21], due to the large amount of wake velocity measurements available and the relatively large range of comparative codes provided in the analysis.

4.1. Experimental Setup
Reference is made to the velocity calculations and predictions made in the MexNext measurement campaign [22, 23]. Experimental parameters are summarised in Table 1. Blade parameters have been taken from reference [24].

4.2. Numerical Model
A lifting line formulation has been applied to determine shed circulation over the blade as described substantially in Marten et al [25]. The Kutta-Joukowski rule is used to determine bound circulation along the blade span. The Kutta condition at the trailing edge of the blade is used to determine shed and trailing filament strength. Crucial to the accuracy of this method is the quality of the lift \( c_l(\alpha) \) and drag \( c_d(\alpha) \) polars, here taken from [26] and interpolated with the Montgomery method to cover the range of angles of attack \(-180^\circ \leq \alpha \leq 180^\circ\). It should be noted here that these airfoil polars are derived from experiments where \( Re > 10^6 \), and rotational corrections have not been applied. These \( Re \) values have generally not been reached in this experiment, and hence it is possible that discrepancies arise. For the vortex particles a second order Guassian regularisation function has been used [7]. Particle position and strength are integrated with an explicit second order Runge-Kutta scheme. The polynomial order of the VPML model is taken to be \( P = 3 \). A sensitivity analysis has been carried out and the adequate (converged) simulation parameters are given in Table 2.

5. Results
The performance of the method is first demonstrated. The validation of the lifting line model is then demonstrated by comparing predicted blade loading. This shows that the induced velocities...
in the rotor plane are correctly calculated, and also that the polars used for the analysis are sufficient for the given $Re$ number range. Finally the validation of the model for the near wake is demonstrated.

5.1. Model performance

Comparison is made here to direct evaluation. The evaluation of the VP kernel by direct evaluation has been implemented in a fully parallelised architecture to be evaluated on a graphical processing unit (GPU). Furthermore, in the VPML method the course grid summation (iii) and near-field evaluation (vi) constitute in general over 95% of total computation time. These have been also highly parallelised as described below:

- Course grid summation: As described earlier, the course-grid summation can be treated as a matrix product. The implementation of the open source CLBlast library [27] achieved a speed-up of factor 10 as compared to CPU evaluation.
- Near-field evaluation: By tiling boxes on the GPU, as was carried out for the direct evaluation, a speed-up of the near-field evaluation time by a factor of roughly 10 was achieved.

The wall clock calculation time for the method is displayed in Fig. 3. It can be seen that for both methods of direct evaluation of the particle set, the calculation costs scale with $O(N^2)$. The calculation time however is drastically reduced by using the parallelised form of direct evaluation. In comparison to this, the calculation costs for the VPML method scale with $O(N)$. The large overhead costs of the grid tree setup and initial calculation of the course grid interaction matrices

| Rotor Radius | 2.25 $m$ |
|---------------|----------|
| Hub Diameter  | 0.5 $m$  |
| Rotational Speed | 424.5 $RPM$ |
| Wind speed    | 10, 15, 24 $ms^{-1}$ |
| Tip Speed Ratio | 9.99, 6.66, 4.16 |

| Blade Discretization | 25 elements (cosine) |
| Azimuthal Discretization | $5^\circ$ |
| Timestep               | $3.93 \cdot 10^{-3} s$ |
| Coresize $\sigma$     | 0.3 $m$ |
| Stretching Scheme      | Transpose |
contribute to the much larger initial simulation time costs of the VPML method. Above a certain value of $N$ however, generally found to be around ten thousand, the VPML method becomes computationally cheaper. For the simulations shown here, minimum box size $H = 1.5 \text{m}$ and polynomial order $P = 2$. Greater accuracy is attainable by increasing $P$, however induced velocities were found to be essentially identical to direct evaluation for these parameters.

![Figure 4: Normal loading as a function of radial position. From left to right: $U_\infty = 10.15$ and $24 \text{ m}\text{s}^{-1}$. Legend: MEXICO experiment (dots), VPML method (solid red), Technion CFD (dashed green), ECN-CFX (solid blue).](image)

5.2. Comparison of Loading

The turbine loads are reproduced here for three TSR values. In all cases the results have been compared to full finite-volume CFD simulations in order to provide a comparative basis to higher-order methods. These results are taken from reference [22]. In general, this document contains numerous CFD codes for comparison, for the work here the two codes representing the best agreement with experiment have been chosen. The calculated normal loads are shown in Fig. 4. Here comparison is made to the Ansys CFX Fluent CFD solver used by ECN [21], which makes use of the $k-\omega$ SST Menter turbulence model [28]. The second set of CFD simulations were carried out at the Technion University with the commercial software STAR-CCM [22] using the standard $k-\omega$ SST turbulence model [29]. In these simulations the effect of the tower, nacelle and hub are included. It is seen that for the high and optimal TSR range the agreement with the CFD codes is excellent. For the operating case $U_\infty = 24 \text{ m}\text{s}^{-1} (\text{TSR} 4.16)$ it is seen that the blade loading is generally overpredicted by all models. The particular sensitivity seen here is possibly explained by the fact that the blade is exposed to significantly higher angles of attack along its span, and the NACA airfoil section is post-stall in this domain. It should hence be expected that the stall behaviour of the blade leads to wake blockage which is not accounted for in the VPML method, and which is also not reflected in the interpolated polars. It can generally be stated that the airfoil polar behaviour appears satisfactory for the medium and high TSR ranges and hence the analysis can proceed assuming that the particle strength definition is occurring satisfactorily at these operating points.

5.3. Wake Velocity Comparison

Wake velocities are compared here again to full finite volume CFD simulations. Although the loading was found to deviate for the windspeed $U_\infty = 24 \text{ m}\text{s}^{-1}$, in this case, due to the high inflow velocity, the pitch of the wake helices is quite large and the wake is therewith swept away quicker. This reduces interaction between blade wakes and facilitates a simplified case of the wake self-influence. For this reason results have been compared here also for this inflow velocity. In addition to the two codes used in the blade loading analysis, comparison is also made here to the CENER Wind Multi-Block (WMB) finite volume CFD code [30]. For the following analysis
the standard reference frame for a wind turbine analysis is used where the $x$ axis is aligned with the rotor shaft, and traversing downstream implies positive $x$ direction.

\begin{figure}[h]
\centering
\includegraphics[width=0.4\textwidth]{axial_velocity.png}
\caption{Axial velocity along the axial traverse for the radial position $R = 1.8$ m for $U_\infty = 24$ ms$^{-1}$.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=0.4\textwidth]{radial_velocity.png}
\caption{Axial velocity along the radial traverse at $x = 0.3$ m for $U_\infty = 24$ ms$^{-1}$.}
\end{figure}

5.3.1. Axial Velocity ($U_{ax}$)  
Results are shown for both axial and radial traverses.

**Axial Traverse**  
The results for $U_\infty = 24$ ms$^{-1}$ are shown in Fig. 5. Experimental velocity components were collected for a single snapshot in time at radial position $R = 1.4$ m, blade 1 azimuthal position $\phi = 0^\circ$ (12 o'clock) position and axial positions $-4.5$ m $\leq y \leq 4.5$ m. Upstream deceleration in the slipstream is seen which implies induction effects are correctly modeled. The reduction in velocity through the rotor is also clearly seen. The oscillations caused as the measurement point traverses past the discrete wake helices is also well captured along with the general reduction in wake velocity.

**Radial Traverse**  
The axial velocity measured along the downstream radial traverse for $U_\infty = 24$ ms$^{-1}$ are shown in Fig. 6. Here the $x$ position is held constant at $x = 0.3$ m and the velocity sensor traverses the radial coordinate at the azimuthal position $\phi = 90^\circ$. It should be noted here that the results for the upstream radial traverse agreed excellently with experiment, however are skipped here as these are of less interest. The variation of $U_{ax}$ is predicted excellently and is in fact better than the predictions made by the CFD solvers. Results were also collected for the higher operational speeds, however particularly due to vortex stretching at higher induction, the method generally underpredicted the induced velocities. This is seen to generally be a feature of all simulation methods, and is a point for further investigation. In particular the ability of the method to include vortex particle remeshing has been shown to be an important feature in previous work [10, 31, 14]. This is necessary to ensure that for longer time simulations, the so-called overlap criteria [7] is satisfied. This shall be implemented in future work.

5.3.2. Radial Velocity ($U_{rad}$)  
Considering the radial system to be defined along the $x$ axis, for steady, ideal conditions the wake should essentially be symmetrical hence justifying the introduction of a radial system here. The radial velocity represents the expansion or contraction of the wake. Shown in Fig. 7 and 8 are $U_{rad}$ as a function of axial position. Two values of radial coordinate are shown here to show that the method also predicts the expansion at different radial positions. It can be seen that the method predicts very well the wake expansion, again
Figure 7: Radial velocity along the axial traverse for the radial position \( R = 1.4 \) m for \( U_\infty = 15 \) ms\(^{-1}\).

Figure 8: Radial velocity along the axial traverse for the radial position \( R = 1.8 \) m for \( U_\infty = 15 \) ms\(^{-1}\).

in some cases better than CFD. Wake expansion is seen to decay for increasing axial location, implying wake convergence.

5.3.3. Tangential Velocity (\( U_{tang} \)) This represents the swirl velocity of the wake. The results of the inspection of \( U_{tang} \) over the downstream radial traverse are given in Fig. 9 and 10. It is seen that the method here predicts excellently the distribution of \( U_{tang} \) for the \( U_\infty = 24 \) ms\(^{-1}\) case. This agreement is seen to be less satisfactory for the higher induction case \( U_\infty = 15 \) ms\(^{-1}\). The fluctuations in the wake, caused by variations in trailing vorticity over the blade are satisfactorily captured. Properly identifying this requires finer spanwise resolution and a generally higher particle count. Furthermore, the interaction of such internal shear layers within the wake necessitates the correct modeling of viscous interaction.

Figure 9: Tangential velocity along the radial traverse at \( x = 0.3 \) m for \( U_\infty = 24 \) ms\(^{-1}\).

Figure 10: Tangential velocity along the radial traverse at \( x = 0.3 \) m for \( U_\infty = 15 \) ms\(^{-1}\).
Conclusion
This paper presents the implementation the VPML method, an accelerated medium-order aerodynamic calculation tool for the calculation of wind turbine wake velocities. The ability to predict wake velocities at medium to high TSR values is presented. In some cases the method is seen to perform with greater accuracy than much more computationally expensive CFD codes. The combination of these results, along with the significant acceleration of the method demonstrate that the arguments for industry to progress to aerodynamic methods of higher order than the traditional BEM become increasingly attractive.

References
[1] Jensen N A note on wind generator interaction Tech. rep. Risø National Laboratory
[2] Katic I, Højstrup J and Jensen N O A simple model for cluster efficiency Tech. rep. Risø National Laboratory
[3] Ott S, Berg J and Nielsen M 2011 Linearized cfd models for wakes Tech. rep. Risø National Laboratory report R-1772(EN)
[4] Larsen G C, Madsen H A, Bingöl F, Mann J, Ott S, Sørensen J N, Okulov V, Troldborg N, Nielsen M, Thomsen K, Larsen T J and Mikkelsen R 2007 Dynamic wake meandering modeling Tech. rep. Risø National Laboratory report R-1607(EN)
[5] Branlard E 2017 Wind Turbine Aerodynamics and Vorticity-Based Methods (Springer International Publishing)
[6] Winckelmans G 1989 Topics in Vortex methods for the Computation of two- and three-dimensional incompressible unsteady flows Ph.D. thesis California Institute of Technology
[7] Ploumhans P, Winckelmans G, Salmon J K, Leonard A and Warren M S J. Comp. Phys. 178
[8] Degond S and Mas-Gallic P 1989 Mathematics of computation 53 485–507
[9] Winckelmans G and Leonard A 1993 J. Comp. Phys 109 247–273
[10] Cottet G and Koumoutsakos P 2000 Vortex Methods: Theory and Practice (Cambridge University Press)
[11] van Garrel A 2016 Multilevel Panel Method for Wind Turbine Rotor Flow Simulations Ph.D. thesis University of Twente - Enschede
[12] Saverin J, Marten D, Pechilvanoglou G, Nayeri C and Paschereit C 2018 AIAA SciTech Forum (Kissimee, Florida, USA)
[13] Brandt A 1982 Multigrid Methods p 220
[14] Brandt A and Lubrecht A 1990 Journal of Computational Physics 348–370
[15] Venner C and Lubrecht A 2000 Multilevel Methods in Lubrication vol 37 (Elsevier Science)
[16] Schoeppers J, Boorsma K, Cho T, Gomez-Iradi S, Schaffarczyk P, Shen W, Lutz T, Meister K, Stoevesandt B, Schreck S, Micallef D, Pereira R, Sant T, Madsen H and Sorensen N 2012 Final report of ica task 29: Mextnext (phase 1) Tech. rep. ECN
[17] Schoeppers J, Boorsma K, Gomez-Iradi S, Schaffarczyk P, Madsen H, Sorensen N, Shen W, Lutz T, Schulz C, Herraez I and Schreck S 2014 Final report of ica wind task 29: Mextnext (phase 2) Tech. Rep. ECN-E–14–060 ECN
[18] Schoeppers J and Snel H 2007 Model experiments in controlled conditions- final report Tech. rep. Energy Research Centre of the Netherlands
[19] Marten D 2015 Proceedings of the ASME Turbo Expo 2015: Turbine Technical Conference and Exposition
[20] Bertagnolio F, Sorensen N, Johansen J and Fuglsang P 2001 Wind turbine airfoil catalogue Tech. Rep. 1280 Technical University of Denmark, Risoe-R.
[21] Nugteren C 2018 IWOCL’18: International Workshop on OpenCL 53 485–507
[22] Menter F R 1994 AIAA 32 1598–1605
[23] Ferziger J and Milovan A 2002 Computational Methods for Fluid Dynamics Aeronautical and Aerospace Engineering (Springer, Berlin)
[24] Gomez-Iradi S, Stejl R and Barakos G 2009 J. Solar Energy Engineering 131
[25] Koumoutsakos P 1993 Direct Numerical Simulations of Unsteady Separated Flows Using Vortex Methods Ph.D. thesis California Institute of Technology