Predictive wind turbine simulation with an adaptive lattice Boltzmann method for moving boundaries

Ralf Deiterding[1], **Stephen L. Wood**[2]

[1] Aerodynamics and Flight Mechanics Research Group, University of Southampton, Highfield Campus, Southampton SO17 1BJ, UK

[2] University of Tennessee - Knoxville The Bredesen Center, Knoxville, TN 37996, USA

E-mail: r.deiterding@soton.ac.uk

Abstract. Operating horizontal axis wind turbines create large-scale turbulent wake structures that affect the power output of downwind turbines considerably. The computational prediction of this phenomenon is challenging as efficient low dissipation schemes are necessary that represent the vorticity production by the moving structures accurately and that are able to transport wakes without significant artificial decay over distances of several rotor diameters. We have developed a parallel adaptive lattice Boltzmann method for large eddy simulation of turbulent weakly compressible flows with embedded moving structures that considers these requirements rather naturally and enables first principle simulations of wake-turbine interaction phenomena at reasonable computational costs. The paper describes the employed computational techniques and presents validation simulations for the Mexnext benchmark experiments as well as simulations of the wake propagation in the Scaled Wind Farm Technology (SWIFT) array consisting of three Vestas V27 turbines in triangular arrangement.

1. Introduction

The majority of available computational fluid dynamics (CFD) methods for wind engineering approximate the incompressible or weakly compressible Navier-Stokes equations, which leads to a globally coupled problem that in practice can only be solved by iteration. In the case of a wind turbine, the motion of the rotor needs to be captured while still representing the static tower and nacelle geometry. A typical approach is to enclose the rotor in a moving cylinder mesh that is either perfectly embedded [1] or overlaps [2] with a static background mesh. Complicate and computing intensive special grid intersection and interpolation algorithms are mandatory that can also be difficult to parallelize.

So far, simulations of turbines that resolve structural and topographic details accurately to the cm scale have concentrated on simulations of the NREL Phase VI [3] and the Mexnext benchmark experiments [4], in which large-scale laboratory rotors are operated at prescribed rate of rotation in quasi-uniform inflow. The objective is the numerical prediction of thrust and torque on the rotor and the determination of the near-turbine velocity deficit. Using unstructured grids throughout, Lynch and Smith [5] reported a computational effort of $\sim 12,000\,$h CPU for a grid of $\sim 67\,$M cells to compute one revolution of the NREL Phase VI rotor ($\sim 180\,$h CPU/1M cells/revolution). This is in good agreement with a documented result from the structured Wind Multi-Block code by Liverpool University that required 7128 h CPU with a grid of $\sim 34\,$M cells to compute one revolution of the Mexnext three-bladed rotor, which corresponds to an effort of 209 h CPU/1M cells/revolution [4]. An even better computational performance of only 1152 h CPU per revolution of the Mexnext rotor on a 28.3 M cell grid is documented for the structured multi-block code EllipSys3D ($\sim$ 40.7 h CPU/1M cells/revolution) [4]. However, nacelle and tower could not be represented leading to large velocity errors when the turbine is at a yawed position. Already for 0° inflow, major deviations to the experiment with a 14.9% difference in thrust and a 15.0% difference in torque had to be noticed [6].

As an alternative, we adopt in here the lattice Boltzmann method (LBM). Instead of approximating the Navier-Stokes equations, the approach is based directly on the Boltzmann equation, the fundamental stochastic particle equation of gas dynamics [7]. By discretizing the particle velocity space on an equidistant Cartesian grid (the lattice) and employing a time-explicit streaming and collision algorithm [8], in which transport of quantities is always exactly by one grid point, the LBM exhibits the properties of a low dissipation scheme for weakly compressible flows, while computing times are considerably reduced compared to typical Navier-Stokes-based CFD solvers [9]. On uniform grids, unsteady flow computations can easily show performance gains by up to two orders. A large number of lattice Boltzmann schemes is nowadays available, cf. [10]. Being a type of Cartesian immersed boundary method, the LBM is also principally well suited for modeling fluid-rotor interaction.

In the following, we sketch the basics of our numerical approach in Section 2 and present validation and benchmark simulations for the Mexnext configuration in Section 3. In Section 4, a simulation of the SWIFT array at uniform inflow conditions is discussed, demonstrating the capabilities of our LBM-based software system. The conclusions follow in Section 5.

2. Adaptive lattice Boltzmann method

The LBM is based on solving the simplified Boltzmann equation $\partial_t f + \mathbf{u} \cdot \nabla f \ = \omega(f^{eq} - f)$. Partial density distribution functions $f_\alpha(x,t)$ are associated to transport in a discrete velocity direction $\mathbf{e}_\alpha$. The total density distribution is given as $\rho = \sum_\alpha f_\alpha(\mathbf{x},t)$ and the macroscopic moments are $\rho u_i = \sum_\alpha \mathbf{e}_{\alpha i} f_\alpha(\mathbf{x},t)$. The discrete lattice Boltzmann equation reads

$$f_\alpha(\mathbf{x} + e_\alpha \Delta t, t + \Delta t) = f_\alpha(\mathbf{x},t) + \omega_L \Delta t \left(f_\alpha^{eq}(\mathbf{x},t) - f_\alpha(\mathbf{x},t) \right), \tag{1}$$

where the time step $\Delta t = \Delta x / \left(\sqrt{3} c_s\right)$ is based on the speed of sound c_s, which corresponds to a trivial advection step (aka *streaming*) of f_α in direction of $\mathbf{e}_\alpha$ by exactly one lattice point. In this paper, a three-dimensional stencil with 19 distribution functions, i.e., 19 update equations (1) is used (aka D3Q19). For the right hand side of (1) we adopt the equilibrium function

$$f_\alpha^{eq}(\rho, \mathbf{u}) = \rho t_\alpha \left[1 + \frac{\mathbf{e}_\alpha \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_\alpha \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u}^2}{2c_s^2} + \frac{(\mathbf{e}_\alpha \mathbf{u})^3}{2c_s^6} - \frac{(\mathbf{e}_\alpha \mathbf{u})\mathbf{u}^2}{2c_s^4} \right] \tag{2}$$

with coefficients t_α determined from the specific discrete velocity set. It can be shown via a Chapman-Enskog expansion that for vanishing Knudsen number the method converges to a solution of the weakly compressible Navier-Stokes equations [11]. Note that considering the last two terms in (2) corresponds to an extended 3rd order scheme that is stable for higher flow velocities than the normally used 2nd-order accurate LBM scheme [12], which was of vital importance for the stability of the simulations of Section 3.

The hydrodynamic pressure for the equilibrium function (2) reads $p = \sum_\alpha f_\alpha^{eq} c_s^2 = \rho c_s^2$. Kinematic viscosity ν and discrete collision frequency ω_L are connected by the relation

$$\omega_L = \tau_L^{-1} = \frac{c_s^2}{(\nu + \Delta t c_s^2/2)}. \tag{3}$$

To consider high Reynolds number flows, a large eddy simulation (LES) approach with eddy viscosity model is adopted [13]. The effective viscosity $\nu^\star = \nu + \nu_t = \frac{1}{3}\left(\tau^\star - \frac{1}{2}\right) \Delta x^2 / \Delta t$ is used

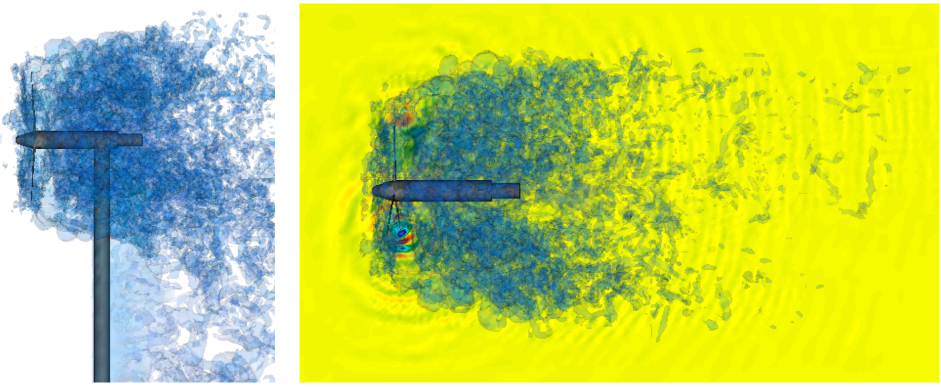

Figure 1. Mexnext rotor simulation for 0° yaw. Flow field at $t = 10\,\mathrm{s}$ visualized by displaying iso-surfaces of vorticity, side (left) and top view (right)

with $\tau_L^\star = \tau_L + \tau_t =: 1/\omega_L^\star$ in (1). Like Hou *et al.* [11], we apply the Smagorinsky model to evaluate the turbulent eddy viscosity as $\nu_t = (C_s f \Delta x)^2 |\bar{S}|$. The filtered strain tensor can be directly expressed via a second moment as $\bar{S}_{ij} = \left(2\rho c_s^2 \tau_L^\star\right)^{-1} \sum_a \mathbf{e}_{\alpha i} \mathbf{e}_{\alpha j} (\bar{f}_\alpha - \bar{f}_\alpha^{eq})$, from which eventually $\tau_t = \frac{1}{2}\left[\left(\tau_L^2 + 18\sqrt{2}(\rho\Delta x^2/\Delta t^2)^{-1} C_s^2 \Delta x \bar{S}\right)^{1/2} - \tau_L\right]$ can be derived.

Since the utilization of uniform lattices is not effective for realistic geometries, the LBM has been incorporated into the AMROC software [14], which supports block-structured adaptive mesh refinement (SAMR) algorithms generically. In order to fit smoothly into the existing, fully parallelized software, we have implemented the LBM cell-based, which makes the scheme also conservative in ρ and ρu_i. In the SAMR approach, finite volume cells are clustered with a special algorithm into non-overlapping rectangular grids. The grids have a suitable layer of halo cells for synchronization and applying inter-level and physical boundary conditions. Refinement levels are integrated recursively and with successively smaller time steps on higher levels. Distributions streaming across refinement boundaries during the recursive temporal update are considered and employed in the collision step of the LBM by a correction-type algorithm that was especially designed for the SAMR methodology, cf. [15]. Note that the resolution alteration on each level requires the utilization of a different collision frequency, cf. Eq. (3).

In order to implement non-Cartesian boundary conditions with the LBM, we have chosen to adopt for now a generic 1st order accurate ghost fluid approach that was already available in AMROC [14]. Applied to LBM, this technique adjusts the distributions in embedded ghost cells to consider the boundary conditions of a non-Cartesian reflective wall moving with prescribed velocity $\mathbf{w}$, which involves interpolation and mirroring of ρ, $\mathbf{u}$ across the boundary to ρ' and $\bar{\mathbf{u}}$ and modification of the macro-velocity in the immersed boundary cells to $\mathbf{u}' = 2\mathbf{w} - \bar{\mathbf{u}}$. From the newly constructed macroscopic values the density distributions in the embedded ghost cells are simply set to $f_\alpha^{eq}(\rho', \mathbf{u}')$ before computing a normal LBM update.

Geometrically complex structures, such as a wind turbine, are constructed as multi-body systems and are represented as sets of triangulated surface meshes configured in kinetic chains represented by quaternions. The dynamics of these mechanisms are solved by a recursive Newton-Euler method at each time step [16]. This triangulation is transformed into a scalar level set function φ on the Cartesian mesh that stores the distance to the boundary surface. The benefit of using a smooth distance function is that the normal to the nearest boundary location can always be evaluated as $\mathbf{n} = \nabla\varphi / |\nabla\varphi|$. The level set is computed exactly only in a small band around the embedded structure using a specially developed algorithm based on characteristic reconstruction and scan conversion [17].

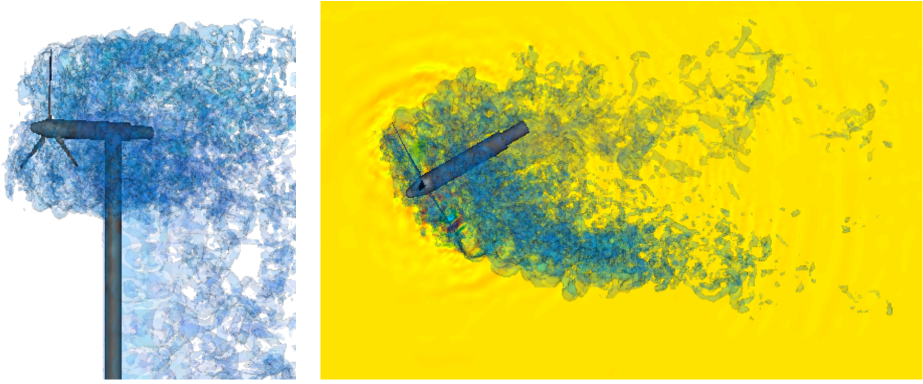

Figure 2. Mexnext rotor simulation for 30° yaw. Flow field at $t = 10\,\mathrm{s}$ visualized by displaying iso-surfaces of vorticity, side (left) and top view (right).

Table 1. Rotor thrust and torque predictions versus experimental results.

0° **yaw**

	Exp.	Sim.
F_x [N]	1517	1633
T_x [Nm]	284.6	307.9

30° **yaw**

	Exp.	Sim.
F_x [N]	13.66	14.80
T_x [Nm]	7.72	8.36

Table 2. Normalized deviation (in %) versus experimental results in predicted flow velocity along transects.

	yaw	0°		30°	
	transect	in	out	in	out
Axial	u_x	6.416	7.663	5.742	6.410
	u_y	3.400	4.061	3.043	3.373
	u_z	3.073	3.678	2.752	3.068
		up	down	up	down
Radial	u_x	6.556	7.325	7.093	6.655
	u_y	3.409	3.809	3.684	3.466
	u_z	3.242	3.659	3.511	3.294

3. Validation for Mexnext laboratory rotor

We have validated the level-set-based dynamically adaptive lattice Boltzmann code for the case of the Mexnext laboratory rotor with diameter $D = 4.5\,\mathrm{m}$ at prescribed rate of revolution and constant inflow velocity of 14.93 m/s [4]. The simulations replicate from [4] Case 1.1 examining aligned operation ($u_0 = 14.93\,\mathrm{m/s}$, $\rho = 1.246\,\mathrm{kg/m^3}$, $T_\infty = 284.03\,\mathrm{K}$, $p_\infty = 101922\,\mathrm{Pa}$) and Case 2.1 examining yawed operation at 30° ($u_0 = 14.99\,\mathrm{m/s}, \rho = 1.237\,\mathrm{kg/m^3}$, $T_\infty = 285.96\,\mathrm{K}$, $p_\infty = 101847\,\mathrm{Pa}$). In both cases the turbine operates a prescribed rotation rate 424.5 rpm and -2.3° blade pitch.

A computational domain of $30\,\mathrm{m} \times 20\,\mathrm{m} \times 20\,\mathrm{m}$ is discretized with a base grid of $120 \times 80 \times 80$ and three additional levels refined by the isotropic factors 2, 2, and 4 are used, which gives a finest resolution around structures of $1.5625\,\mathrm{cm} = 0.0035D$. In quasi-steady state, our dynamically adaptive code uses $\sim 12\,\mathrm{M}$ cells in total. For instance, for the 30° yaw case, the computational effort is 267 h CPU per rotation, where overall 10 s physical time (70.75 revolutions) have been computed. The computation was run on 120 cores Intel Ivybridge and required 157.6 hours wall time in total. Normalized per 1 M cells and per revolution, our current implementation required only $\sim 22.25\,\mathrm{h}$ CPU/1M cells/revolution, which obviously compares quite favorably to previous computations, cf. Section 1.

Vorticity snapshots of the simulated flow fields at $t = 10\,\mathrm{s}$ at 0° and 30° yaw angle are shown in Figs. 1 and 2, respectively, giving a good qualitative representation of the quasi-steady wake structures. The graphics in the top row of Fig. 3 display normal and tangential loads on an individual blade averaged over the interval $[5\,\mathrm{s}, 10\,\mathrm{s}]$, i.e., over ~ 35 rotations, as it is passing

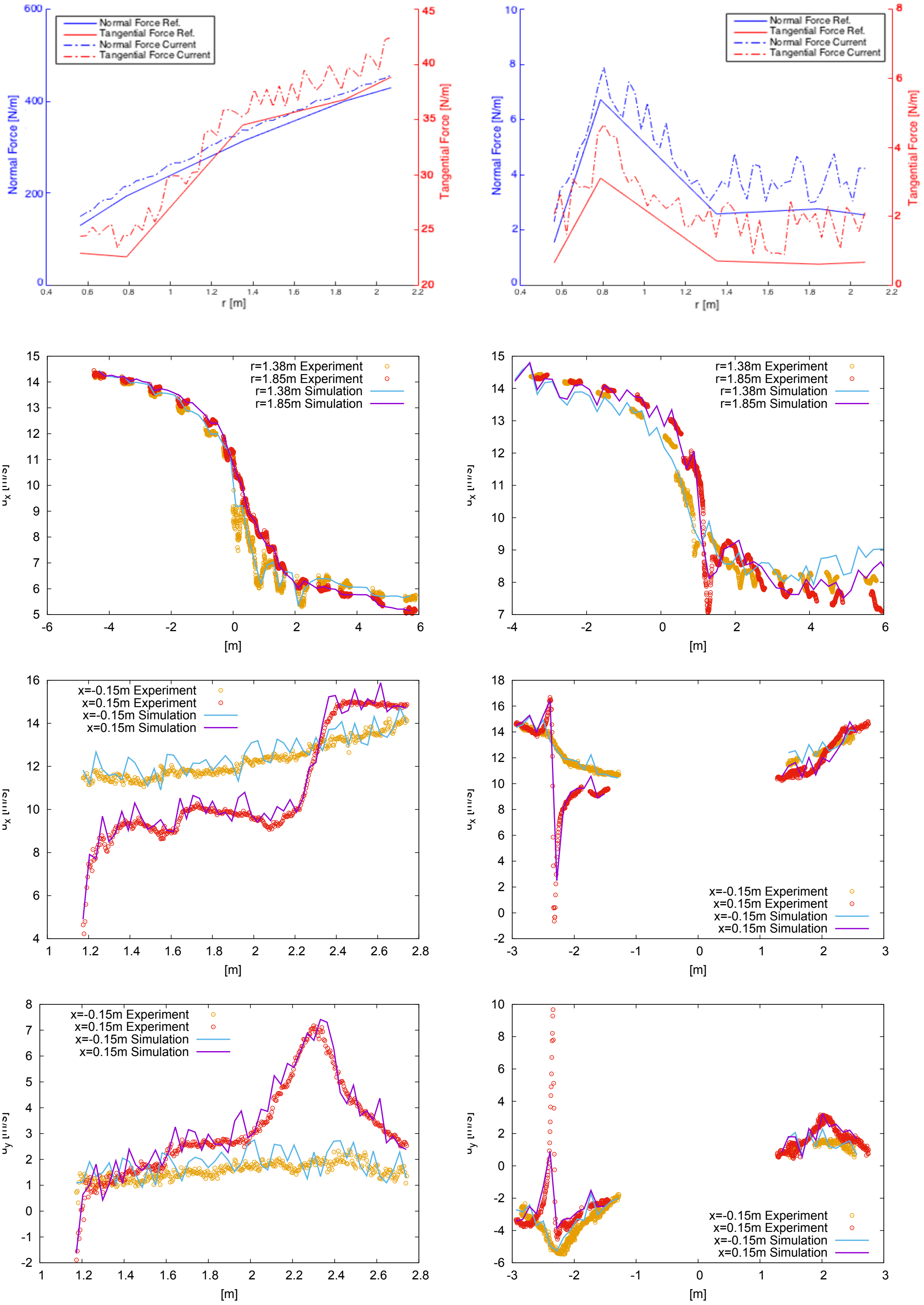

Figure 3. Top row: averaged single-blade loads in simulation (current) and experiment (force reference). Lower three rows: comparison of time-averaged velocity in simulation and experiment along transects. The velocity in the axial direction is compared along axial (2nd row) and radial (3rd row) transects, the radial velocity only along radial transects (bottom row). The 0° yaw case is shown in the left column, the 30° yaw case in the right.

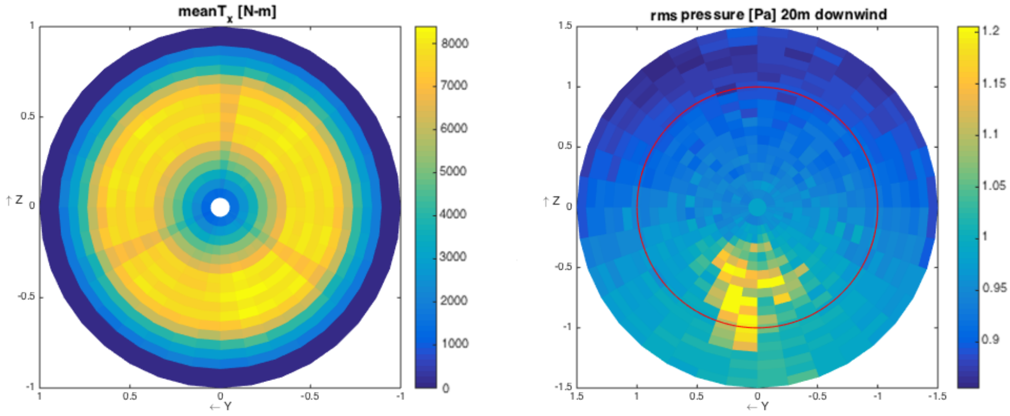

Figure 4. Flow field of Vestas V27 turbine operating at 33 rpm in uniform 8 m/s inflow averaged over 10 s. Left: Torque along blades, right: pressure fluctuations 20 m downwind.

through the vertically upward position. Note that the comparative experimental results (solid lines in these two graphics) use data from five measurement locations only. The normal and tangential loads are all overestimated by approximately 8% with the largest errors occurring near the blade root and tips. Mean rotor loads during $[5, 10]$ s are summarized in Table 1. Fluctuations in loads over the radius in the range of a few N/m can likely be attributed to small-scale vortex structures in the highly turbulent flow fields, cf. Figs. 1 and 2. While for 0° our normal load prediction is comparable to other 3D CFD codes, the tangential load estimate is closer to the experimental results than any prediction in [4], especially for larger values of r.

To compare with PIV data from the experiments, samples were taken along axial and radial transects at specified locations (cf. [4]) at a sampling rate of 7.075 Hz and again time-averaged over the interval $[5\,\text{s}, 10\,\text{s}]$. The most important of these comparisons are shown in Fig. 3. The agreement is apparently quite good and the velocity predictions generally exhibit maximal deviations to the experimental data of similar relative magnitude as the mean rotor loads, see Table 2. In the graphics in the second row of Fig. 3 the velocity deficit in axial direction due to the rotor can clearly be seen. In the third row graphics, but particularly in the bottom row, depicting the radial velocity, the vortex production at the blade tip can be inferred.

4. Wake propagation in SWIFT array

In order to assess the wake prediction capabilities of the adaptive LBM, we have carried out a simulation of the U.S. Department of Energy Scaled Wind Farm Technology (SWIFT) facility at uniform inflow conditions. The SWIFT array uses Vestas V27 turbines to obtain full-scale wake interaction data. The V27 is a small-scale turbine of rotor diameter of $D = 27$ m and has a tower height of ~ 35 m. It achieves a maximal energy output of 225 kW at wind velocities from 14 to 25 m/s. The time-dependent surface geometry model for each turbine consists of $\sim 23,300$ triangles.

In all subsequent computations it is assumed that the inflow is in direction of the turbine middle axis and that the pitch blade angle is 0°. The turbines are always operating at prescribed rate of rotation of 33 rpm. Since this is a prototypical investigation, simplified inflow conditions are used in behalf of a more realistic turbulent atmospheric boundary layer profile: The inflow velocity is constant across the rotor diameter of velocity 8 m/s, with no turbulent fluctuations, and initially a laminar boundary layer profile of 5 m height is assumed near the ground.

A single turbine is considered first to provide some ad-hoc model validation. Dynamic isotropic refinement with three additional levels refined by the factors 2, 2 and 4 is applied,

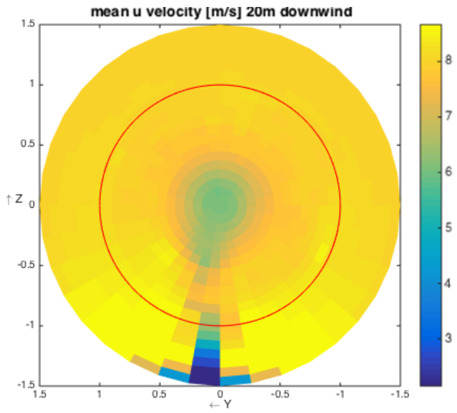

Figure 5. Vestas V27 turbine operating at the conditions of Fig. 4. Axial velocity 20 m downwind.

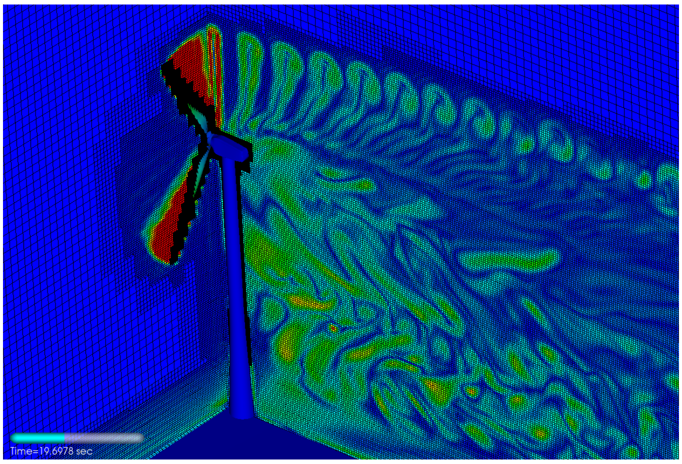

Figure 6. Adaptive mesh near a single turbine at $t = 19.7\,\mathrm{s}$. Color planes visualize the magnitude of the vorticity vector.

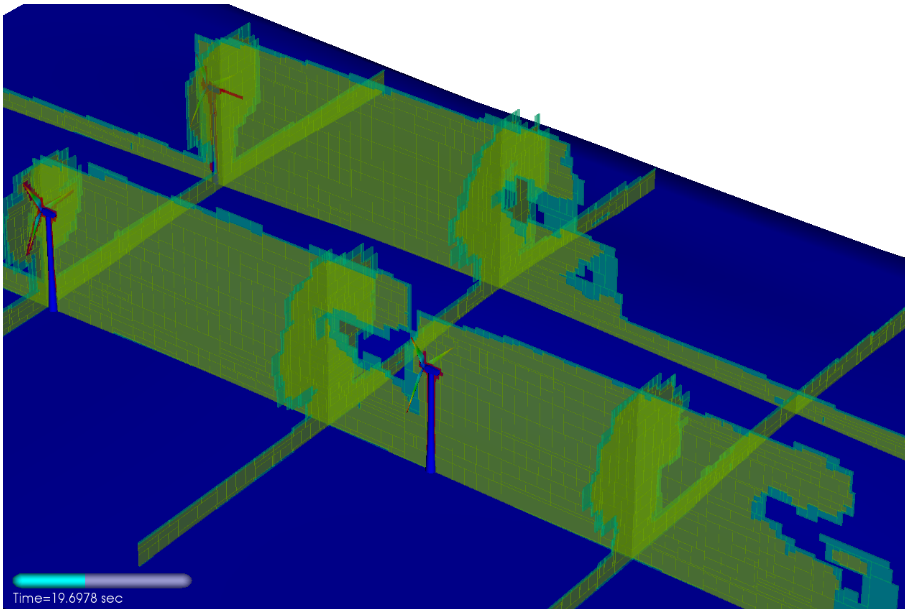

Figure 7. Domains of refinement and structured blocks visualized by color at $t = 19.7\,\mathrm{s}$.

ensuring a resolution of $\Delta 3.125\,\mathrm{cm}$ around solid structures. A physical time of 18 s is simulated and from $t = 8\,\mathrm{s}$ onward, pressure, axial velocity, and torque have been sampled every 0.034 s on 18 radial sections along the blades and 20 m downwind of the rotor plane. Figures 4 and 5 display this data averaged over the last 10 s of the computation. Note that the sections have an extension of $0.75D$. Visualized are the torque on the blades, the root-mean-square (rms) variation of the pressure 20 m downwind as well as the axial velocity at the same downwind location. From the left graphic of Fig. 4, an average power coefficient of $C_p = 0.44$ and an average torque coefficient of $C_t = 0.78$ can be deduced. This would correspond to 81 kW production, which is within 5 % of the manufacturer's rated values [18]. In the right graphic of Fig. 4 and in Fig. 5 the influence of tower and nacelle is imminent. Pressure fluctuations stemming from a low velocity recirculating region behind the tower and nacelle are clearly captured. The low streamwise velocity at the

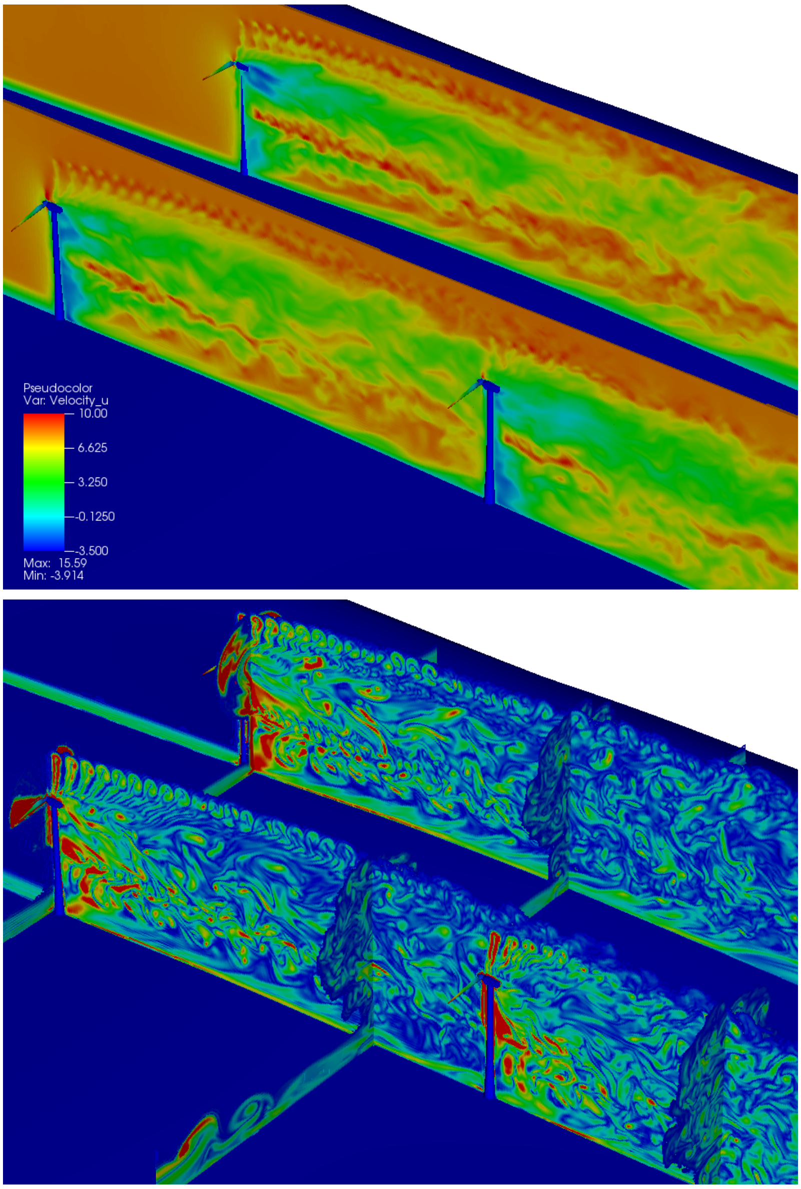

Figure 8. Fully established wakes in quasi-steady conditions after $t = 50.9\,\mathrm{s}$ simulated time. Top: color plot of velocity component in wind direction; bottom: color plot of vorticity magnitude.

bottom of the sampling circle is due to the vortical structures emanating from the base of the tower and ground.

In the SWIFT array, three Vestas V27 are placed in triangular arrangement with respect to the prevalent wind direction allowing a direct comparison between an undisturbed and a two-turbine wake. In our case, the turbines are located at $x = 0$, $z = 0$; $x = 135\,\mathrm{m}$, $z = 0$; and $x = 0$, $z = 80\,\mathrm{m}$ and the inflow is in the x-direction. Using a domain of extensions $14D \times 3D \times 3D$ on a dynamic 4-level mesh refined isotropically by the factors 2, 2, 4, the computation ensures a resolution of $6.25\,\mathrm{cm} = 0.0023D$ near the geometry and $0.0092D$ in the wakes. When the wakes are fully established, the dynamic mesh uses $\sim 147\,\mathrm{M}$ cells instead of 44.04 billion in the uniform case (factor ~ 300), where $\sim 102\,\mathrm{M}$ cells are concentrated on the second finest level. Using 192

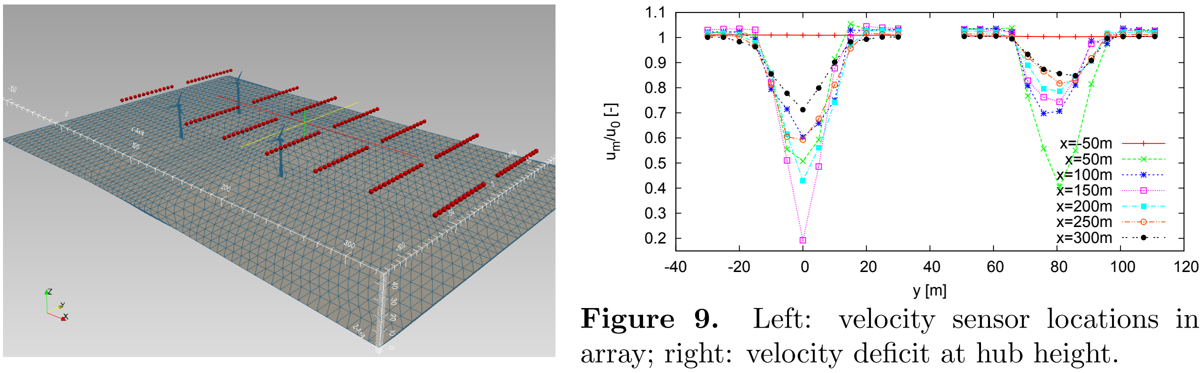

Figure 9. Left: velocity sensor locations in array; right: velocity deficit at hub height.

cores Intel-Xeon-E5, the computational effort was just 784.4 h CPU or 4.09 h wall clock time per revolution, i.e., ∼ 5.34 h CPU/1M cells/revolution. While in the simulations in Section 3 the turbine geometry and the wake were refined identically and at the highest level, the majority of cells is now on the second finest level, which is four times coarser, and used for downstream wake capturing. The utilization of hierarchical time steps inherent to the SAMR method (cf. Section 2) explains the reduction in computational effort per revolution by roughly a factor of 4 compared to Section 3.

Figures 6 and 7 visualize the dynamically constructed mesh at an intermediate time. Figure 6 shows the individual computational cells as well as the magnitude of the vorticity vector in color. It can clearly be seen that the highest resolution follows the moving structure, while the second highest level is refined based on fluid vorticity. Figure 7 shows by color the different levels of mesh refinement and it can be inferred that the downwind wakes are still in a development stage and close to reaching the x-location of the downwind turbine. In this graphic, also the individual refinement blocks characteristic for the SAMR approach can be discerned. Note that, as the refinement is evolving, parallel load-balancing and dynamic repartitioning based on a space-filling curve algorithm happen automatically within the AMROC framework, cf. [14].

Figure 8 displays the fully established wake fields. Note the transitioning of the tip vortices into large central wake structures behind the individual turbine and how this process is clearly altered in the lower row. It is apparent from Fig. 8 that – thanks to the low dissipation properties of the LBM – the turbine wakes are well preserved throughout the entire domain. A very different vortex shedding pattern is evident in the tower wakes compared to wakes created by the rotor motion. Figure 9 displays the wind velocity deficit at hub height temporally averaged over the time interval [40 s, 50 s]. Sensor locations are placed in lines 50 m apart at $z = 37$ m. The sensor positions are indicated by the red spheres in the left graphic of Fig. 9. The right graphics depicts the deficit in the normalized mean velocity component in the x-direction. The introduction of an additional velocity deficit by the second turbine, which is decreasing with increasing downstream distance, can be seen. An asymmetric shift towards the two-turbine column in the velocity deficit behind the isolated turbine indicates a slight interaction even in the lateral direction. Interesting to note is the larger velocity deficit at $x = 50$ m and its faster recovery in the single-turbine wake. This is a clear indication of the resistance and resulting pressure increase from the downwind turbine that happens in the two-turbine column. Note that this effect could not accurately be represented other than by a detailed numerical simulation.

5. Conclusions

The prototype of a dynamically adaptive, three-dimensional lattice Boltzmann method for simulating the wake fields behind fully resolved wind turbines has been developed. The implementation has been validated using data from the Mexnext 4.5 m rotor experiments at both 0° and 30° yaw angle. Preliminary wake investigations for the SWIFT array, consisting of three Vestas V27 turbines, have been carried out. Our method can deal well with complex solid structures under motion. For the Mexnext configurations it is found that the deviations to the experimental results are of similar magnitude for both yaw angles.

It is further found that the approach is computationally efficient and well suited for propagating wake structures over long distances without adding excessive amounts of numerical dissipation. This result was recently confirmed in a meticulous investigation of a simplified train model at scale 1:25 [19]. In this study, LES and detached eddy simulations (DES) with incompressible fluid solvers from the popular OpenFOAM software package were compared in a similar way as above to LES computations of our new LBM solver. Again, while delivering time-averaged force and roll moment predictions in best agreement with available experimental results, our implementation was approximately 16 times faster than OpenFOAMs PISO solver using a static unstructured mesh.

As a next step, it is planned to incorporate enhanced wall-near treatment approaches into the LBM for turbulent flows. For cases that fully resolve turbulent boundary layers, the damping of the Smagorinsky constant by the van Driest approach can be easily implemented, cf. [20]; for high Reynolds number situations, in which accurate resolution of the boundary layer is prohibitively expensive with a Cartesian method, a wall function model will be implemented. A successful and consistent integration of a wall function model into the LBM with Smagorinsky large-eddy simulation approach has been demonstrated by Malaspinas & Sagaut [21]. Their approach uses an implicit iteration locally in each wall-near cell to construct a set of density distribution functions to specify the skin friction provided to the outer flow according to boundary layer theory. Incorporating improved wall-near treatments into our LBM-based LES code can be expected to improve its computational accuracy further.

References

[1] Hsu M C, Akkerman I and Bazilevs Y 2014 *Wind Energy* **17** 461–481

[2] Gundling C, Sitaraman J, Roget B and Masarati P 2015 *Wind Energy* **18** 1893–1916

[3] Simms D A 2001 Nrel unsteady aerodynamics experiment in the nasa-ames wind tunnel: A comparison of predictions to measurements Tech. Rep. NREL/TP-500-29494 National Renewable Energy Laboratory

[4] Schepers J G and Boorsma K 2012 Final report of iea task 29: Mexnext (phase 1) – Analysis of Mexico wind tunnel measurements Tech. Rep. ECN-E-12-004 European research Centre of the Netherlands

[5] Lynch C E and Smith M J 2013 *Wind Energy* **16** 1033–1048

[6] Soslashrensen N N, Bechmann A, Rethore P E and Zahle F 2014 *Wind Energy* **17** 75–86

[7] Succi S 2001 *The Lattice Boltzmann Equation: For Fluid Dynamics and Beyond* (Oxford Science Publications)

[8] Hähnel D 2004 *Molekulare Gasdynamik* (Springer)

[9] Chen S and Doolen G 1998 *Annu. Rev. Fluid Mech.* **30** 329–364

[10] Guo Z and Shu C 2013 *Lattice Boltzmann Method and Its Applications in Engineering* (World Scientific Publications Co.)

[11] Hou S, Sterling J, Chen S and Doolen G D 1996 *Pattern formation and lattice gas automata* vol 6 ed Lawniczak A T and Kapral R (Fields Inst Comm) pp 151–166

[12] X Shan X-F Yuan H C 2006 *J. Fluid Mech.* **550** 413–441

[13] Aidun C K and Clausen J A 2010 *Annu. Rev. Fluid Mech.* **42** 439–472

[14] Deiterding R 2011 *European Series in Applied and Industrial Mathematics: Proceedings* **34** 97–150

[15] Deiterding R and Wood S L 2015 *IV Int. Conf. on Coupled Problems in Science and Engineering* ed Schrefler B, Onate E and Papadrakakis M pp 489–500

[16] Tsai L 1999 *Robot Analysis: The Mechanics of Serial and Parallel Manipulators* (Wiley)

[17] Mauch S P 2003 *Efficient Algorithms for Solving Static Hamilton-Jacobi Equations* Ph.D. thesis California Institute of Technology

[18] 1994 V27-225 kW, 50 Hz wind turbine with tubular/lattice tower Research Report 941129 1.2.0.24 Vestas

[19] Fragner M M and Deiterding R 2015 *Proc. Parallel CFD 2015*

[20] Weickert M, Teike G, Schmidt O and Sommerfeld M 2010 *Comput. Math. Appl.* **59** 2200–2214

[21] Malaspinas O and Sagaut P 2014 *J. Comput. Phys.* **275** 25–40