VI International Conference on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2015
B. Schrefler, E. Oñate and M. Papadrakakis(Eds)

A DYNAMICALLY ADAPTIVE LATTICE BOLTZMANN METHOD FOR PREDICTING WAKE PHENOMENA IN FULLY COUPLED WIND ENGINEERING PROBLEMS

RALF DEITERDING* AND STEPHEN L. WOOD†

*German Aerospace Center (DLR)
Institute for Aerodynamics and Flow Technology
Bunsenstr. 10, 37073 Göttingen, Germany
e-mail: ralf.deiterding@dlr.de

†University of Tennessee - Knoxville
The Bredesen Center
Knoxville, TN 37996, USA
e-mail: swood@utk.edu

Key words: Lattice Boltzmann Method, Fluid-Structure Coupling, Wind Turbines, Hinged Wing, Adaptive Mesh Refinement

Abstract. The essential components of a dynamically adaptive, parallel lattice Boltzmann method particularly tailored for coupled wind engineering are described. By utilizing a level set approach for geometry embedding the method can handle rotating and moving structures effectively and is thereby genuinely suited for fluid-structure coupling problems involving low-Mach number flows. The approach is validated for the canonical six degrees of freedom test case of a driven two-segment hinged wing. Subsequently, the wake field in an array of three Vestas V27 wind turbines at prescribed rotation rate and under constant inflow condition is simulated for two different scenarios. These results demonstrate that the time-explicit nature and the low dissipation properties of the lattice Boltzmann scheme in combination with dynamic mesh adaptation are able to predict well-resolved vortex structures created by realistic rotor speeds far downstream of the turbines at moderate computational costs.

1 Introduction

Many aerodynamic wind engineering problems are characterized by a strongly coupled interaction between moving structures and fluid flow. An example of particular technical relevance are horizontal axis wind turbines. Here, the incoming flow drives the motion and elastic deformation of the rotor, which in itself generates large-scale wake structures that can affect downstream turbines considerably. When multiple wind turbines are placed in

an array, the question of optimal placement arises. If a turbine is exposed to a major vortex field, its energy output will generally be reduced and additionally induced structural vibrations will cause disproportionately accelerated material fatigue. In order to simulate the flow field in a turbine array layout, e.g., for a dominant wind direction, vortex or disc actuator models are presently most frequently adopted. Computational fluid dynamics (CFD) with accurate consideration of the moving structures is still rarely applied, which is due to the complexities involved in solving the weakly or incompressible Navier-Stokes equations on moving three-dimensional meshes effectively, cf. [10].

As an alternative to the implicit, typically pressure-correction based CFD solution algorithms generally applied in wind engineering [10], we adopt in here the lattice Boltzmann method (LBM). The LBM is based on solving the Boltzmann equation in a specially chosen, discrete phase space and is fully explicit in time [7]. The scheme is normally constructed on uniform Cartesian grids and geometrically complex boundaries are considered with an immersed boundary approach, making the method well suited for considering moving structures. Here, we utilize a level set distance function to represent embedded objects. Dynamic mesh adaptation is applied in addition in order to increase the local resolution based on the level set function and features detected in the flow field [4]. Distributed memory parallelization is adopted to allow for large-scale simulations.

The paper is organized as follows: In Section 2, we recall the construction principles of the LBM and our embedded boundary treatment method. In Section 3, the block-based mesh adaptation procedure and in particular the incorporation of the LBM are presented. Section 4 explains our approach in dealing with embedded geometries. Section 5 discusses a coupled validation configuration of a two-segment hinged wing with torsion damper and Section 6 presents first simulations of entire wind turbines with prescribed motion of the rotor that demonstrate the benefit of the proposed overall approach. The conclusions are given in Section 7.

2 Lattice Boltzmann method

The lattice Boltzmann method is based on computing approximations of the Boltzmann equation with a simplified collision operator

$$\partial_t f + \mathbf{u} \cdot \nabla f = \omega(f^{eq} - f) \tag{1}$$

on a rectangular grid of characteristic domain length L with isotropic mesh spacing Δx under the assumption of a small Knudsen number $\mathrm{Kn} = l_f/L \ll 1$, where the mean free path length l_f is replaced with Δx. A crucial idea of the LBM is to approximate Eq. (1) in a specially chosen discrete phase space, in which a partial density distribution function $f_\alpha(\mathbf{x}, t)$ is associated to every discrete lattice velocity $\mathbf{e}_\alpha$. The total density distribution is given as $\rho(\mathbf{x}, t) = \sum_\alpha f_\alpha(\mathbf{x}, t)$ and the macroscopic moments as $\rho(\mathbf{x}, t)u_i(\mathbf{x}, t) = \sum_\alpha \mathbf{e}_{\alpha i} f_\alpha(\mathbf{x}, t)$. A splitting approach is then adopted that first solves the homogeneous transport equation with the time-explicit update step

$$\mathcal{T}: \qquad \tilde{f}_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = f_\alpha(\mathbf{x}, t). \tag{2}$$

Here, we apply the D3Q19 model for which the lattice velocities are defined as

$$\mathbf{e}_\alpha = \begin{cases} 0, & \alpha = 0, \\ (\pm 1, 0, 0)c, (0, \pm 1, 0)c, (0, 0, \pm 1)c, & \alpha = 1, \ldots, 6, \\ (\pm 1, \pm 1, 0)c, (\pm 1, 0, \pm 1)c, (0, \pm 1, \pm 1)c, & \alpha = 7, \ldots, 18, \end{cases}$$

with $c = \Delta x/\Delta t$. The physical speed of sound c_s is related to c by $c_s = c/\sqrt{3}$. The right-hand of Eq. (1) is integrated subsequently by the collision operator

$$\mathcal{C} : f_\alpha(\cdot, t + \Delta t) = \tilde{f}_\alpha(\cdot, t + \Delta t) + \omega \Delta t \left(\tilde{f}^{eq}_\alpha(\cdot, t + \Delta t) - \tilde{f}_\alpha(\cdot, t + \Delta t) \right) \tag{3}$$

with equilibrium function

$$f^{eq}_\alpha(\rho, \mathbf{u}) = \rho t_\alpha \left[1 + \frac{3\mathbf{e}_\alpha \mathbf{u}}{c^2} + \frac{9(\mathbf{e}_\alpha \mathbf{u})^2}{2c^4} - \frac{3\mathbf{u}^2}{2c^2} \right] \tag{4}$$

with $t_0 = 1/3$, $t_\alpha = 1/18$ for $\alpha = 1, \ldots, 6$ and $t = 1/36$ for $\alpha = 7, \ldots, 18$. The variation in hydrodynamic pressure for the equilibrium function (4) reads $\delta p = \sum_\alpha f^{eq}_\alpha c_s^2 = (\rho - \rho_0) c_s^2$.

Applying a Chapman-Enskog expansion procedure, it can be shown [8] that the sketched LBM converges to a solution of the weakly compressible Navier-Stokes equations

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{5a}$$

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u}. \tag{5b}$$

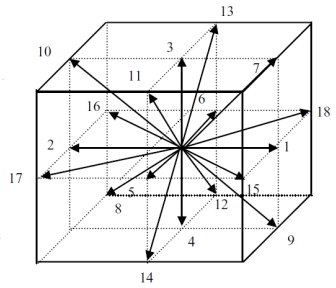

Figure 1: The velocities $\mathbf{e}_\alpha$ of the D3Q19 lattice.

It can be shown further, cf. [7], that the kinematic viscosity ν and collision frequency ω are connected by the relation

$$\omega = \tau^{-1} = \frac{c_s^2 \Delta t}{\nu + \Delta t c_s^2/2}. \tag{6}$$

While the sketched model can be used directly to simulate laminar flows, it is mandatory to apply a turbulence model in addition in high Reynolds number situations. In the context of LBM, it is common to adopt a large eddy simulation approach and assume that the partial density distribution functions used in the scheme represent the resolved scales. The subgrid scale turbulence is then considered by adding a turbulent viscosity ν_t to the physical one and by utilizing the effective viscosity $\nu^\star = \nu + \nu_t = \frac{1}{3}\left(\tau^\star - \frac{1}{2}\right) c\Delta x$ with $\tau^\star = \tau + \tau_t =: 1/\omega^\star$ in (3) throughout the scheme. Like Hou *et al.* [8], we apply the Smagorinsky model to evaluate ν_t, for which $\nu_t = (C_{sm}\Delta x)^2 \bar{S}$ is used, with

$$\bar{S} = \sqrt{2 \sum_{i,j} \bar{\mathbf{S}}_{ij} \bar{\mathbf{S}}_{ij}}, \qquad \bar{\mathbf{S}}_{ij} = -\frac{1}{2\rho_0 c_s^2 \tau} \sum_a \mathbf{e}_{\alpha i} \mathbf{e}_{\alpha j} (\bar{f}_\alpha - \bar{f}^{eq}_\alpha) \tag{7}$$

yielding

$$\tau_t = \frac{1}{2} \left(\sqrt{\tau^2 + 18\sqrt{2}(\rho_0 c^2)^{-1} C_{sm}^2 \Delta x \bar{S}} - \tau \right). \tag{8}$$

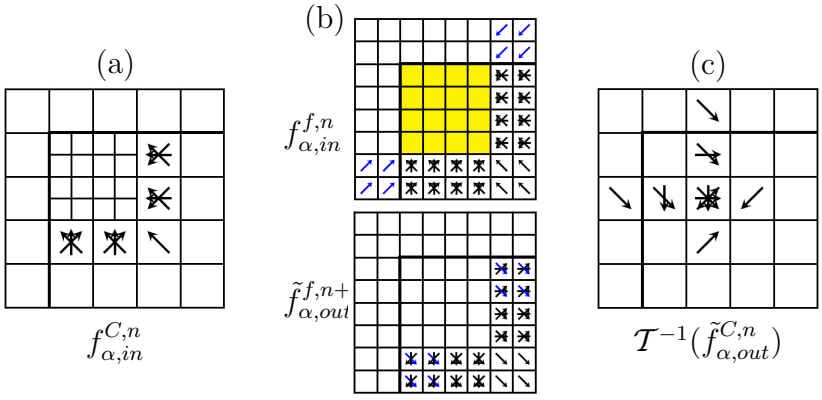

Figure 2: Visualization of partial density distribution functions involved in necessary data exchange at a coarse-fine boundary. The thick black lines indicate a physical boundary. (a) Coarse distributions going into fine grid; (b) ingoing interpolated fine distributions in halos (top), outgoing distributions in halos after two fine-level transport steps (bottom); (c) averaged distributions replacing coarse values before update is repeated in cells next to boundary.

3 Dynamic mesh adaptation

For local dynamic mesh adaptation we have adopted the block-structured adaptive mesh refinement (SAMR) method after Berger & Collela [1]. In order to fit smoothly into our existing, fully parallelized finite volume SAMR software system AMROC [4], 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. The spatial mesh width Δx_l and the time step Δt_l are refined by the same factor r_l, where we assume $r_l \geq 2$ for $l > 0$ and $r_0 = 1$. Note that in an adaptive LBM the collision frequency ω_l is not a constant but needs to be adjusted according to Eq. (3) for the update on each level. In addition to this, the interface region requires a specialized treatment. Distinguishing between the transport and collision operators, $\mathcal{T}$ and $\mathcal{C}$, cf. Eqs. (2) and (3), the steps of our method for a refinement factor of 2 are:

1. Complete update on coarse grid: $f^{C,n+1}_\alpha := \mathcal{C}\mathcal{T}(f^{C,n}_\alpha)$
2. Use coarse grid distributions $f^{C,n}_{\alpha,in}$ that propagate into the fine grid, cf. Fig. 2(a), to construct initial fine grid halo values $f^{f,n}_{\alpha,in}$, cf. Fig. 2(b).
3. Complete transport $\tilde{f}^{f,n}_\alpha := \mathcal{T}(f^{f,n}_\alpha)$ on whole fine mesh. Collision $f^{f,n+1/2}_\alpha := \mathcal{C}(\tilde{f}^{f,n}_\alpha)$ is applied only in the interior cells (yellow in Fig. 2(b)).
4. Repeat 3. to obtain $\tilde{f}^{f,n+1/2}_\alpha := \mathcal{T}(f^{f,n+1/2}_\alpha)$ and $f^{f,n+1}_\alpha := \mathcal{C}(\tilde{f}^{f,n+1/2}_\alpha)$.

5. Average outgoing distributions from fine grid halos (Fig. 2(c)), that is $\tilde{f}^{f,n+1/2}_{\alpha,out}$ in the inner halo layer and $\tilde{f}^{f,n}_{\alpha,out}$ (outer halo layer) to obtain $\tilde{f}^{C,n}_{\alpha,out}$.

6. Revert transport for averaged outgoing distributions, $\bar{f}^{C,n}_{\alpha,out} := \mathcal{T}^{-1}(\tilde{f}^{C,n}_{\alpha,out})$, and overwrite those in the previous coarse grid time step, cf. Fig. 2(d).

7. Parallel synchronization of $f^{C,n}_{\alpha}, \bar{f}^{C,n}_{\alpha,out}$ on entire level.

8. Repeat complete update on coarse grid cells next to coarse-fine boundary only: $f^{C,n+1}_{\alpha} := \mathcal{C}\mathcal{T}(f^{C,n}_{\alpha}, \bar{f}^{C,n}_{\alpha,out})$

This algorithm is computationally equivalent to the method by Chen *et al.* [2] but tailored to the SAMR recursion that updates coarse grids in their entirety before fine grids are computed. Because of the nonlinearity of the collision operator $\mathcal{C}$ it becomes necessary under this paradigm to repeat the LBM update for those coarse grid cells that share a face or corner with a fine grid.

4 Embedded structure handling

We represent non-Cartesian boundaries implicitly on the adaptive Cartesian grid by utilizing a scalar level set function φ that stores the distance to the boundary surface. The boundary surface it located exactly at $\varphi = 0$ and the boundary outer normal in every mesh point can be evaluated as $\mathbf{n} = -\nabla\varphi/|\nabla\varphi|$ [3]. We treat a fluid cell as an embedded ghost cell if its *midpoint* satisfies $\varphi < 0$.

In order to implement non-Cartesian boundary conditions with the LBM, we have chosen to pursue a 1st order accurate ghost fluid approach that was already available in AMROC [3]. In our technique, the density distributions in embedded ghost cells are adjusted to model the boundary conditions of a non-Cartesian reflective wall moving with velocity $\mathbf{w}$ before applying the unaltered LBM. The last step 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 $\mathbf{u}' = 2\mathbf{w} - \bar{\mathbf{u}}$, cf. [4]. From the newly constructed macroscopic values the density distributions in the embedded ghost cells are simply set to $f^{eq}_{\alpha}(\rho', \mathbf{u}')$.

Real-world geometries are considered in AMROC as triangular surface meshes. The computation of the level set distance information in every Cartesian mesh point could principally be accomplished by simply iterating over the entire surface mesh; yet, this would lead to detrimental performance for increasing mesh size. The problem is equivalent to determining for every Cartesian cell the closest facet on the surface mesh. For this purpose, we employ a specially developed algorithm based on characteristic reconstruction and scan conversion developed by Mauch [9] that is used to compute the distance exactly only in a small band around the embedded structure.

The dynamics of multi-body systems undergoing interaction with the fluid are modeled as sets of triangulated surface meshes configured in kinetic chains. The dynamics of

these mechanisms are solved by a recursive Newton-Euler method at each time step [12]. Considering an arbitrary link with a coordinate frame located at point $\mathbf{P}$ that is not coincident with its associated body's center of mass, the force and torque applied by the preceding link are

$$\begin{pmatrix} \mathbf{F} \\ \boldsymbol{\tau}_{\mathrm{P}} \end{pmatrix} = \begin{pmatrix} m\mathbf{1} & -m[\mathbf{c}]^{\times} \\ m[\mathbf{c}]^{\times}\mathbf{I}_{\mathrm{cm}} & -m[\mathbf{c}]^{\times}[\mathbf{c}]^{\times} \end{pmatrix} \begin{pmatrix} \mathbf{a}_{\mathrm{P}} \\ \boldsymbol{\alpha} \end{pmatrix} + \begin{pmatrix} m[\boldsymbol{\omega}]^{\times}[\boldsymbol{\omega}]^{\times}\mathbf{c} \\ [\boldsymbol{\omega}]^{\times}(\mathbf{I}_{\mathrm{cm}} - m[\mathbf{c}]^{\times}[\mathbf{c}]^{\times})\,\boldsymbol{\omega} \end{pmatrix}. \tag{9}$$

Here, we additionally define the total force and torque acting on a body, $\mathbf{F} = (\mathbf{F}_{FSI} + \mathbf{F}_{prescribed}) \cdot \boldsymbol{\mathcal{C}}_{xyz}$ and $\tau = (\tau_{FSI} + \tau_{prescribed}) \cdot \boldsymbol{\mathcal{C}}_{\alpha\beta\gamma}$ respectively. Where $\boldsymbol{\mathcal{C}}_{xyz}$ and $\boldsymbol{\mathcal{C}}_{\alpha\beta\gamma}$ are the translational and rotational constraints, respectively. $\mathbf{F}_{FSI}$ and τ_{FSI} are determined for each body by integrating the fluid pressure on the triangular facets of the respective body's surface mesh. Each surface mesh is associated with a kinetic link in a chain that begins with a base link in the global coordinate frame. Links are connected by joints that may be independently constrained in six degrees of freedom relative to the preceding link. The evolution of the triangular surface mesh as well as the velocity $\mathbf{w}$ in each node are communicated to the LBM fluid solver in dedicated coupling time steps. The data exchange corresponds to the time step of an SAMR level but this does not have to be the finest refinement level available, cf. [5].

Triangulated surface meshes can be generated by lofting closed 3D curves or read from the standard STL format. Each surface mesh is associated with a kinetic link in a chain that begins with a base link in the global coordinate frame. Links are connected by joints that may be independently constrained in six degrees of freedom relative to the preceding link. Constraints, $\boldsymbol{\mathcal{C}}$, which may be prescribed motions or reaction forces are enforced during the backward calculation step of the Newton-Euler method as joint forces are calculated by proceeding from the distill link of a kinetic chain to the base link. The angular and linear position, velocity and acceleration of each link in terms of its preceding link are calculated during the forward calculation step. This formulation readily facilitates the analysis of motions, forces, and moments on each link and triangulated surface in the global coordinate frame or in any of the link coordinate frames.

5 Fluid-structure interaction validation

A canonical problem of fluid-structure interaction and wake prediction proposed by Toomey & Eldredge [11] is selected for validating the fluid-structure interaction simulation method. This model, depicted in Fig. 3, utilizes a system of two articulated rigid bodies connected by a torsion spring and damper. The kinematics of the centroid of the driven wing are prescribed, while the trailing body responds passively to the aerodynamic and inertial/elastic forces. The principle unknown in this rigid body dynamics problem is the hinge angle θ. The parametric kinematic equations

$$X_t(t) = \frac{A_0}{2}\frac{G_t(ft)}{\max Gt}C(ft), \quad \alpha_1(t) = -\beta\frac{G_r(ft)}{\max Gr}, \quad G_t(t) = \int_t \tanh[\sigma_t cos(2\pi t')]dt', \tag{10}$$

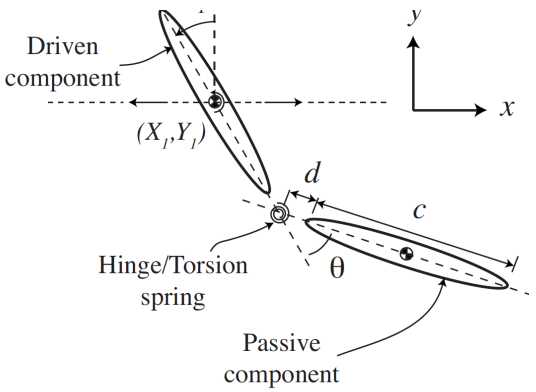

Figure 3: Model system consisting of two rigid elliptical sections connected by a hinge with torsion spring and damper.

Table 1: Kinematic parameters

$A_0\,(cm)$	7.1
$c\,(cm)$	5.1
$d\,(cm)$	0.25
β	$\pi/4$
σ_t	0.628
σ_r	0.628
Φ	0
Re_t	73, 370
Re_r	100, 500
$\rho_b\,(kg/m^3)$	5080
$f\,(Hz)$	0.15

Table 2: Nondimensional mean and peak force and moments

	$Re_r = 100$		$Re_r = 500$	
	Mean	Peak	Mean	Peak
F_x	2.78	430.2	3.78	429.8
$F_x\,(\%)$	-3.3	-9.8	-3.4	-9.9
F_y	39.0	206.2	52.3	210.2
$F_y\,(\%)$	-3.4	-15	-5	-7
M	-3.3	248	0.9	228.5
$M\,(\%)$	26	30	17	19

$$G_r(t) = \tanh[\sigma_r cos(2\pi t + \Phi)], \quad C(t) = \frac{\tanh(8t-2)+\tanh(2)}{1+\tanh(2)}. \tag{11}$$

describe the motion of the driven body. The parameters utilized in this work and in Case 1 of [11] to specify the kinematics through the translational, $G_t(t)$, and rotational, $G_r(t)$, shape functions are given in Table 1. The start-up conditioner, $C_t(t)$, is applied to the translational kinematics to avoid an impulsive start. The translational and rotational Reynolds numbers are based on the peak translational, V, and rotational, $2\pi\beta\sigma_r fc/\tanh(\sigma_r)$, velocities as shown in

$$Re_t = Vc/\nu, \quad Re_r = 2\pi\beta\sigma_r fc^2/(\tanh(\sigma_r)\nu). \tag{12}$$

The mean and peak values of the dimensionless fluid dynamic force, $F_{x,y} = 2F^*_{x,y}/(\rho_f^2 c^3)$, and moment, $M = 2M^*/(\rho_f f^2 c^4)$, generated by the wing motion are presented in Table 2.

The relative error between the results of this work and those in [11] is also presented in Table 2. The wing deflection and vorticity production at Re = 500 are depicted in Fig. 4. Figure 5 displays the hinge deflection angle for experiments and simulations in

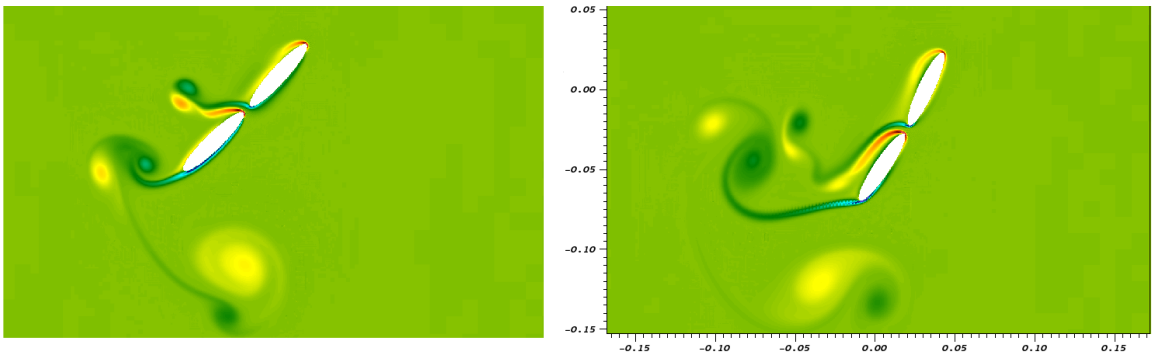

Figure 4: Computed vorticity field at t/T= 0.6 (left), 0.8 (right).

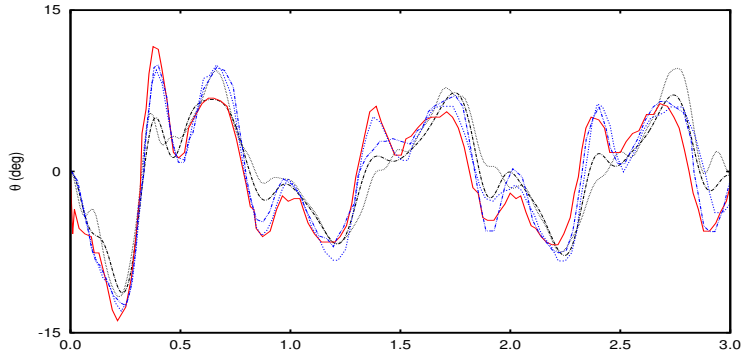

Figure 5: Hinge deflection angle over time. Experimental results (–); Toomey-Eldredge Re=100 (- -), Re=500 (–··); Current Re=100 (- · -), Re=500 (· · ·).

[11] and our simulations through three periods of motion. The dimensional torsion spring and damper coefficients utilized are $K^* = 5.5 \times 10^{-2}\,\mathrm{kg\,m^2/s^2}$ and $R^* = 6.0 \times 10^{-4}\,\mathrm{kg\,m/s^2}$ respectively. These values differ from those used in the viscous vortex particle method (VVPM) simulations by Toomey & Eldredge, however, the wing behavior and predicted loads are comparable and in good agreement with the experiments in [6, 11]. Note that although a no-slip boundary condition is applied at the wing surface, no interfacial shear forces are considered yet in this work. It follows that the fluid loads, particularly the moment, differ between this work and the VVPM used in [11], as shown in Table 2, while the hinge deflection presented in Fig. 5 is in good agreement. Without considering structural loading from shear forces yet, this work predicts the peak forces within 15% of those simulated by the VVPM and reveals the rotation of the driven body influencing hinge deflection with the same periodic trend as the experiments conducted by Toomey & Eldredge [6, 11].

6 Simulation of wakes behind wind turbines

Utilizing the developed LBM solver for moving geometries, we have carried out a simulation campaign to test the suitability of the overall approach to simulate the flow

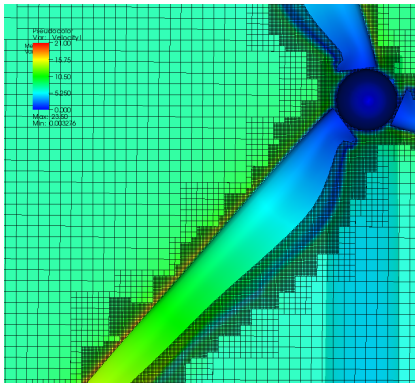

Figure 6: Snapshot of SAMR Mesh in rotor midplane.

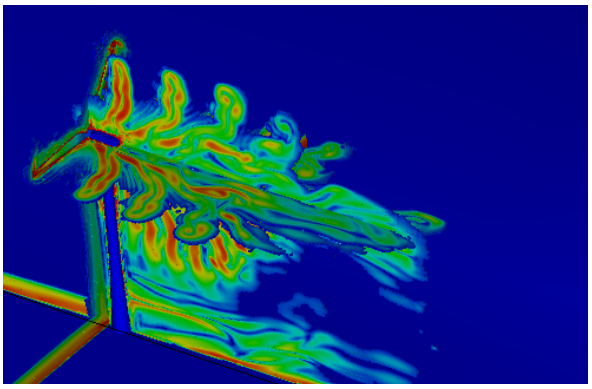

Figure 7: Emerging vortex field at $t \approx 9.3\,\mathrm{s}$. Rotation was started at $t = 4\,\mathrm{s}$.

fields created by wind turbines. We have built a tailored flexible surface mesh model of a Vestas V27 turbine. The V27 has a rotor diameter of $D = 27\,\mathrm{m}$, a tower height of $\sim$ 35 m and achieves its maximal energy output of 225 kW at wind velocities from 14 to 25 m/s. A prototypical ground topology is also included into the surface mesh model that represents the time-dependent geometry with $\sim 23,300$ facets per turbine. It is assumed that the inflow wind direction is always in direction of the turbine middle axis and the pitch blade angle is at 0 degree.

In the first computation, the wind velocity is $u_1 = 7\,\mathrm{m/s}$ with a boundary layer profile of 5 m height assumed near the ground. A simulation domain of extensions $200\,\mathrm{m} \times 100\,\mathrm{m} \times 100\,\mathrm{m}$ is used and discretized with a base grid of $400 \times 200 \times 200$ cells. Dynamic refinement with three additional levels refined by the factors $r_{1,2} = 2$ and $r_3 = 4$ is applied. The two highest levels are reserved to refine the moving surface mesh of rotor and tower at a geometry resolution of $\Delta x = 3.125\,\mathrm{cm}$; level 1 is used to dynamically adapt to the wake region using an empiric error estimation criterion on $|\mathbf{u}|$. Figure 6 shows a snapshot of the Cartesian cells in the rotor midplane and the moving structure colored by the length of the prescribed velocity vector. The computation is run for 141,344 highest level iterations to a final time of $t_e = 30\,\mathrm{s}$, where a constant rotation with 15 rpm (corresponding to a power generation of $\sim 52.5\,\mathrm{kW}$) is started at $t = 4\,\mathrm{s}$.

Figure 7 displays by color the length of the vorticity vector in two 2D planes shortly after starting the rotation. The creation of an emerging helical main vortex structure in the downwind direction emanating from the blade tips can be inferred. This computation was run on 96 cores of a cluster with Intel Xeon-Westmere CPUs and required $\sim 10,400\,\mathrm{h}$ CPU in total, i.e., 108 h wall clock time.

The second test setup considers three V27 turbines and corresponds to the U.S. Department of Energy's Scaled Wind Farm Technology (SWIFT) facility. Two turbines are positioned $3D$ apart in the wind direction; the third turbine is placed $5D$ downstream exactly in the rotor middle axis of the first one. This allows direct comparison of the wake field between two interacting turbines with an undisturbed one. The computational

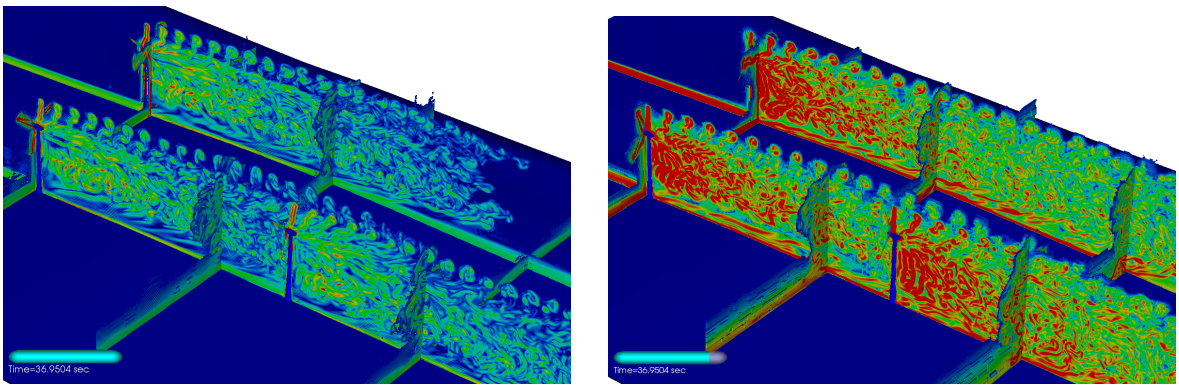

Figure 8: 2D Color planes depict the length of the vorticity vector for 7 m/s (left) and 25 m/s (right) inflow velocity at $t \approx 37$ s.

setup is similar to the previous simulation, where a domain of 488 m × 240 m × 100 m is discretized with a base resolution of 448 × 240 × 100 cells and again refined isotropically by the factors 2,2,4 (resolution near the rotors is $\Delta x = 6.25$ cm). Dynamic refinement of the wake field is permitted now up to level 2, yielding again a resolution in the wake of $\Delta x = 25$ cm.

Two configurations are compared: the case with $u_1 = 7$ m/s inflow velocity and all turbines operating at 15 rpm and a simulation with $u_1 = 25$ m/s and 43 rpm, which corresponds to the maximally allowed rotation rate under normal operations. 94,224 highest level iterations to a final time of $t_e = 40$ s are computed. Figure 8 depicts the wake fields for both cases after $t \approx 37$ s simulated time. The color coding in both graphics uses the same scale and it is obvious that vorticity production is considerably increased in the second configuration. In both simulations, the radius of the main vortices are only sightly increasing and vorticity seems overall exceptionally well preserved. A strong influence of the tower on the wake field is apparent and also the difference between the wake fields behind the isolated turbine versus the two turbines aligned in the wind direction is striking. A three-dimensional visualization of the wake field for the second configuration in its quasi-steady state in Fig. 9 reveals this difference especially well. Because of the higher wind velocity the number of vortex rotations reduces in the second configuration from ~ 15 to ~ 12. In both simulations, incident pressure and velocity on the downstream turbine are reduced (not specially shown). Its rotation apparently induces a less pronounced helical vortex than the upstream turbines, as can be seen in Fig. 9.

Level	Grids	Cells
0	3,234	10,752,000
1	11,921	21,020,256
2	66,974	102,918,568
3	896	5,116,992

Table 3: Grids and cells at t_e.

The used SAMR grids and total number of cells on each level for the second configuration at t_e are given in Table 3. Note that a corresponding uniform mesh would require $44 \cdot 10^9$ cells and take four times more time steps than level 2, which contains the majority of cells. A benchmark run on 288 cores on a cluster of

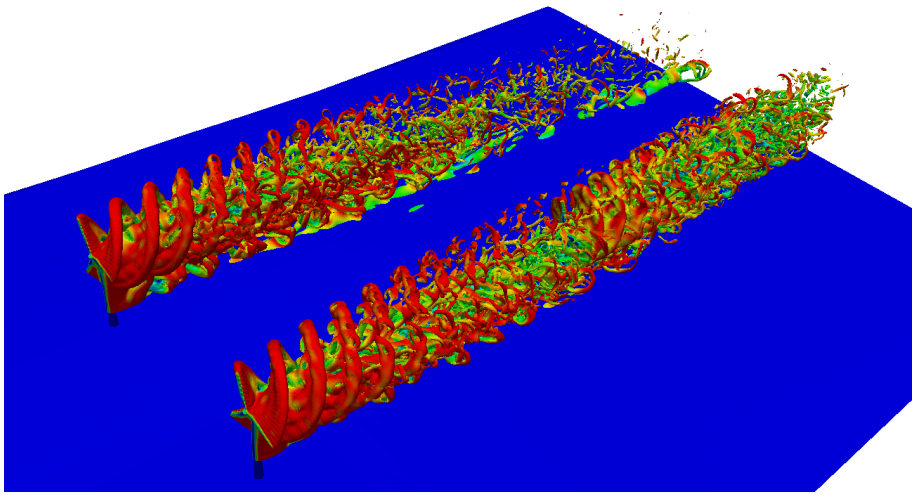

Figure 9: Quasi-steady three-dimensional wake field for the configuration with $u_1 = 25\,\mathrm{m/s}$ and 43 rpm visualized by coloring an iso-surface based on the length of the vorticity vector by the local total velocity.

Intel-Ivybridge CPUs required just 38.5 h wall clock time (11,090 h CPU) to advance the coupled, adaptive computation further from $t_e = 40$ to 50 s, which gives evidence for the practical feasibility of carrying out these computations on compute clusters of moderate size.

7 Conclusions

The first prototype of a dynamically adaptive, three-dimensional lattice-Boltzmann method for simulating coupled wind engineering problems has been developed. The results show that the overall approach can easily handle large structural motion and high coupling frequencies, while the low dissipation properties of the lattice Boltzmann scheme in combination with dynamic mesh adaptation are able to resolve even complex vortex structures at moderate computational costs. First coupled validation has been achieved for a canonical FSI problem from [11]. As a first verification for wind turbine simulation, we have demonstrated that our approach is able to simulate the propagation of wake fields created by the prescribed rotation of accurate Vestas V27 wind turbine rotor geometry, including the interaction with the tower, with apparent good quality and reasonable compute time. Immediate future work will concentrate on fully validating the approach, considering also the exchange of viscous shear forces beside hydrodynamic pressures, for available laboratory benchmarks, e.g., the Mexico wind tunnel experiments [10] and then to incorporate the dynamic elastic response of the blades into the turbine model.

Acknowledgments

Stephen L. Wood was supported by the TN-SCORE Energy Scholar program funded by NSF EPS-1004083 during this work.

REFERENCES

[1] M. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:64–84, 1988.

[2] H. Chen, O. Filippova, J. Hoch, K. Molvig, R. Shock, C. Teixeira, and R. Zhang. Grid refinement in lattice Boltzmann methods based on volumetric formulation. *Physica A*, 362:158–167, 2006.

[3] R. Deiterding. A parallel adaptive method for simulating shock-induced combustion with detailed chemical kinetics in complex domains. *Computers & Structures*, 87:769–783, 2009.

[4] R. Deiterding. Block-structured adaptive mesh refinement - theory, implementation and application. *European Series in Applied and Industrial Mathematics: Proceedings*, 34:97–150, 2011.

[5] R. Deiterding and S. L. Wood. Parallel adaptive fluid-structure interaction simulations of explosions impacting building structures. *Computers & Fluids*, 88:719–729, 2013.

[6] J. D. Eldredge, J. Toomey, and A. Medina. On the roles of chord-wise flexibility in a flapping wing with hovering kinematics. *Journal of Fluid Mechanics*, 659:94–115, 9 2010.

[7] D. Hähnel. *Molekulare Gasdynamik*. Springer, 2004.

[8] S. Hou, J. Sterling, S. Chen, and G. D. Doolen. A lattice Boltzmann subgrid model for high Reynolds number flows. In A. T. Lawniczak and R. Kapral, editors, *Pattern formation and lattice gas automata*, volume 6, pages 151–166. Fields Inst Comm, 1996.

[9] S. P. Mauch. *Efficient Algorithms for Solving Static Hamilton-Jacobi Equations*. PhD thesis, California Institute of Technology, 2003.

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

[11] J. Toomey and J. D. Eldredge. Numerical and experimental study of the fluid dynamics of a flapping wing with low order flexibility. *Physics of Fluids (1994-present)*, 20(7):–, 2008.

[12] L. Tsai. *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*. Wiley, 1999.