Sliding mesh simulations of a wind turbine rotor with actuator line lattice-Boltzmann method

Ribeiro, André F.P.; Muscari, Claudia

DOI
10.1002/we.2821

Publication date
2023

Document Version
Final published version

Published in
Wind Energy

Received: 8 September 2022 | Revised: 20 March 2023 | Accepted: 2 April 2023

DOI: 10.1002/we.2821

RESEARCH ARTICLE

WILEY

Sliding mesh simulations of a wind turbine rotor with actuator line lattice-Boltzmann method

André F. P. Ribeiro[1,2] | Claudia Muscari[3]

[1]SIMULIA Fluids, Dassault Systemes, Stuttgart, Germany

[2]Aerospace Engineering, Delft University of Technology, Delft, The Netherlands

[3]3ME, Delft University of Technology, Delft, The Netherlands

Correspondence
André F. P. Ribeiro, SIMULIA Fluids, Dassault Systemes, Meitnerstr. 4, 70563 Stuttgart, Germany.
Email: andre.ribeiro@3ds.com

Abstract

Simulating entire wind farms with an actuator line model requires significant computational effort, especially if one is interested in wake dynamics and wants to resolve the tip vortices. A need to explore unconventional approaches for this kind of simulation emerges. In this work, the actuator line method is implemented within a lattice-Boltzmann flow solver, combined with a sliding mesh approach. Lattice-Boltzmann solvers have advantages in terms of performance and low dissipation, while the sliding mesh allows for local refinement of the blade and tip vortices. This methodology is validated on a well-documented case, the NREL Phase VI rotor, and the local refinement is demonstrated on the NREL 5 MW rotor. Results show good agreement with reference Navier–Stokes simulations. Advantages and limitations of the sliding mesh approach are identified.

KEYWORDS

NREL 5 MW, NREL Phase VI, tip vortices, wake aerodynamics

1 | INTRODUCTION

The climate crisis caused the European Union and the member states to consider net zero greenhouse gas emissions as a legally binding target for 2050.[1] While wind energy is mature enough to help substantially with such ambitious plans, we still have margins of improvements from all perspectives. We can make better use of the favorable sites available for wind plant installation with effective layout and control of the plant itself. Turbines can be made more durable and to have a longer operational life through the use of better materials, which would also reduce the environmental impact of discarding the turbine components. A detailed and clear report on these points can be found in the work of Veers et al.[2] Any technical progress requires the support of data which can be obtained via experiments and/or numerical simulations. For wind energy applications, given the large Reynolds numbers and the multiple length scales involved, not only is direct numerical simulation infeasible but also large eddy simulation (LES) on fully resolved geometries is too computationally demanding to be used extensively.

In a seminal work published in 2002,[3] the actuator line (AL) model is first presented. Computational fluid dynamics (CFD) is used with LES as the turbulence closure, and the presence of the turbine is accounted for by solving the blade element theory (BET) equations for discrete blade radial sections, where the computed lift and drag is projected onto the flow field as a momentum source along rotating blade lines. In the following two decades, this hybrid approach has become the go-to high fidelity method when dealing with wind turbine wakes and wind farms simulations, mostly using Navier–Stokes (NS) solvers.

The BET step needs only airfoil polars and a locally sampled velocity as input. This velocity should simultaneously account for the deceleration caused by the rotor induction and ignore the blade-local flow effects. One possibility is to sample locally at the AL point (local sampling). Physically this would be justified by potential flow theory: the AL point is the center of the bound vortex cross-section where the effect of the bound vortex itself is null. Potential flow theory, however, does not hold for the conditions in which wind turbines operate and it surely does not hold when using kernel functions different from the classic isotropic Gaussian one. Alternatives have been discussed such as integral sampling,[4] Lagrangian sampling,[5] and the effective velocity model.[6]

The BET forces are computed along lines aligned with each blade. They are applied to the momentum equation as source terms by projecting the line forces into volumetric body force surrounding each line. The most common choice is then to use a three-dimensional Gaussian projection function that is isotropic and fixed in width along the blade span. This makes the resultant body-force field around an AL appear as a cylindrical cloud surrounding the line. More advanced functions lead to a distribution that more closely resembles an actual blade.[4] Several studies have demonstrated the model's reliability in a number of different conditions[4,7,8]; however, result quality is still strongly dependent on the method used to sample the incoming velocity, the characteristics of the chosen projection function, and the quality of the tabulated data.

Recently, particular effort has gone into reducing the number of tuning parameters on AL and making choices based on a better understanding of the physics involved. An important contribution was made by Shives and Crawford,[9] relating the width of the Gaussian projection function to the chord length. Respecting such requirements leads to much finer meshes than traditionally used by AL, hence increasing the computational cost of the simulations. This opens opportunities for alternative CFD approaches and meshing strategies.

The lattice-Boltzmann method (LBM)[10] can reproduce the same physics as the NS equations,[11] while presenting substantial advantages in performance and numerical dissipation. LBM-based solvers have been shown to perform unsteady simulations orders of magnitude faster than NS-based ones,[12] while displaying low-dissipation characteristics equivalent to high order methods.[13] Due to these advantages, some effort has been made in developing LBM-based AL models.[14,15] However, as traditionally LBM is used for isothermal simulations, this limits the use of these methods, as thermal effects are critical for an accurate description of the atmospheric boundary layer.[16]

In this work, we perform the first AL implementation into a state-of-the-art LBM solver that is able to handle thermal effects,[17] local mesh refinement, and complex terrain, which we henceforth refer to as AL-LBM, in an attempt to address the limitations outlined in the previous paragraph. The commercial LBM solver PowerFLOW® is used.

Standard AL simulations are performed with constant refinement on the entire rotor disk, as the forces rotate with the rotor, while the mesh remains static. While a static mesh is advantageous for its simplicity, a sliding mesh approach would allow for local refinement around the rotor blades.[4] Hence, another novel contribution of this work, is the use of a sliding mesh for the AL simulations,[18] which has potential to improve the efficiency of advanced AL models.[4] The objectives of this work are to validate the AL-LBM implementation we have developed and to demonstrate the advantages and disadvantages of the sliding mesh approach for AL simulations in general.

2 | METHODS

2.1 | NS solver

For almost all CFD problems, the reference model is some convenient expression of the NS equations. Each model comes with a level of empiricism and simplifications that have to be taken into account when making the other choices and analyzing the results. In particular, for wind energy, we consider an incompressible NS formulation:

$$\rho\frac{\partial\vec{u}}{\partial t}+\rho(\vec{u}\cdot\nabla)\vec{u}=-\nabla p+\mu\nabla^2\vec{u}, \tag{1}$$

$$\nabla\cdot\vec{u}=0, \tag{2}$$

where ρ is the fluid density, t is the time, $\vec{u}$ is the velocity vector, p is the pressure, and μ is the dynamic viscosity. Force terms can be added to the right side of the equations, in order to model gravity, Coriolis effects, and other body forces. These equations are usually solved using the finite volume method.[19] Most of the CFD codes simulating horizontal axis wind turbines wakes rely on LES, meaning that they simulate the large scales and model the small ones. The effect of the large scale momentum flux caused by the action of the small or unresolved scales is represented by the subgrid-scale (SGS) Reynolds stress. The models used to approximate it are called SGS or sub filter-scale models[20] and most of these models work by adding an eddy viscosity to μ in Equation (1).

We use NS results in this work as validation to our LBM simulations through comparison with SOWFA,[16] a set of libraries, solvers, and boundary conditions specific to wind energy applications, based on the OpenFOAM® finite volume CFD software. SOWFA is open source and inherits the capabilities of OpenFOAM, allowing for relatively inexpensive wind turbine simulations by providing AL and actuator disk solvers, thus making simple, Cartesian grids that are accurate enough to run the simulations on. SOWFA's AL implementation is widely used in the community and is considered state of the art. The AL solver is based on a pressure implicit with splitting of operators (PISO) algorithm.[21] PISO is well suited for simulations where the time step is controlled by external factors and temporal accuracy is important, as in LES. In such cases, the assumption of slow variation over non-linearity holds and the cost of momentum assembly and solution can be safely avoided. SOWFA can also be coupled with FAST, NREL's aero-servo-elastic tool.

We use the typical settings for SOWFA taken from the provided examples, which include a second-order backwards Euler time discretization and an interpolation scheme equivalent to second-order central differences for pressure and velocity, while fluxes use a Rhie–Chow-like formulation.[22] The SGS model is a one equation turbulence kinetic energy-based model.

2.2 | Lattice-Boltzmann solver

In the last two decades, the LBM[23] has been growing in popularity as an alternative to the NS equations. The objective is the same: to have a description of the velocity and pressure fields over time, based on a certain mesh and time steps. However, instead of using continuum mechanics as the foundation for describing fluid flow, statistical mechanics are used, considering the fluid as particles that interact with each other and the boundary conditions.

2.2.1 | The LBM

We first need to consider that a particle can be in a certain spatial coordinate $\vec{x}$. Then, we consider that the particle has a certain velocity $\vec{c}$, which is not the fluid velocity. These quantities change as a function of time t. As dealing with individual particles quickly becomes too expensive, we instead work with the distribution function $f(\vec{x},\vec{c},t)$, which represents the probability that particles at position $\vec{x}$ and time t have the velocity $\vec{c}$. This can also be interpreted as the number of particles in a certain location and time that are traveling with a certain velocity. Fluid properties can be extracted from the moments of the distribution function. The Boltzmann equation describes the behavior of f:

$$\frac{\partial}{\partial t}f(\vec{x},\vec{c},t)+\vec{c}\nabla f(\vec{x},\vec{c},t)=C, \tag{3}$$

where C is the collision operator, which defines how particles interact and is related to the fluid viscosity. This equation has similarities with the NS equations, with a time derivative of the main quantity as the first term, an advection component as the second term, and a viscosity term on the right side.

In LBM, besides space-time, which is discretized as in NS solvers, the particle velocities must also be discretized to obtain a method that can be computationally efficient. Particles can only travel in a finite number of directions and velocities $\vec{c}_i$. The choice of the discretization of the velocity space is a critical part of the LBM. In general, the fewer directions particles can travel to, the less physics are captured by the method and the more directions, the higher the computational cost. For isothermal, quasi-incompressible cases, particles only need to travel to adjacent cells

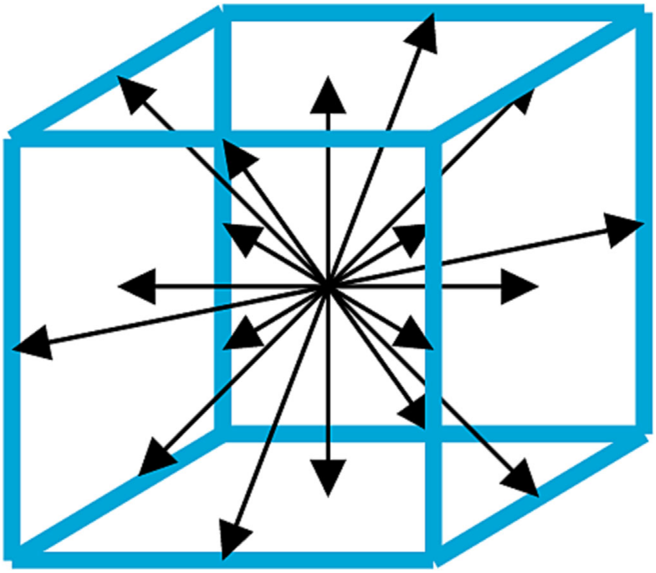

FIGURE 1 Discretization of the velocity space into 19 vectors.

for mass and momentum conservation to be achieved. A cost-effective option is to use the D3Q19 formulation, that is, a 3D 19-state model. This means that particles can travel to 18 adjacent cells or stay in their current cell. This is shown in Figure 1, where the arrows denote the directions particles can travel in, for a lattice of cubic cells.

These assumptions lead to the lattice-Boltzmann equation:

$$f_i(\vec{x}+\vec{c}_i\Delta t, t+\Delta t) = f_i(\vec{x},t) + C_i(\vec{x},t), \tag{4}$$

where the subscript i denotes the velocities that particles are assumed to be able to travel to and Δt is the time step. As in the NS equations, a force term can be added to the lattice-Boltzmann equation, in order to simulate the effect of gravity, Coriolis forces, or other fields that affect the flow, and the eddy viscosity can be added to the fluid viscosity in C. Equation (4) can be interpreted as such: the particle density function for discrete $\vec{x}, \vec{c}, t$ is computed based on two processes, the advection and the collision steps. In the advection step, a particle is streamed from one cell to another by $\vec{c}\Delta t$. One key factor of this step is that, with appropriate choices for $\vec{c}$ and Δt, no interpolation is needed to compute fluxes, in contrast to the advection of the NS equations. This makes this step very inexpensive, computationally, while also making it have little numerical dissipation. The collision step accounts for the interaction between particles with different velocities in the same cell. This is usually computed with the Bhatnagar–Gross–Krook (BGK)[24] approximation, which only depends on simple mathematics based on local quantities, again making it computationally inexpensive and suited for parallelization, unlike the Laplacian in the NS equations.

Our LBM computations are conducted with PowerFLOW, a commercial CFD code. It employs a $k-\epsilon$ RNG turbulence model[25] with a swirl model,[26] which reduces the eddy viscosity in the presence of resolved flow structures, an approach called very large eddy simulation (VLES). This LBM-VLES approach has been validated on numerous simulations of vortical flows.[27,28] A Cartesian mesh is used, which is convenient for comparisons to SOWFA, as the same mesh topology can be employed in both solvers. A unique capability of PowerFLOW is in the handling of very complex geometry,[29] which will be useful for future onshore wind turbine simulations with complex terrain. This is done with the concept of surface elements, or surfels.[30] Other features relevant to wind farm simulations are the ability to simulate complex buoyancy effects, by using a finite difference thermal solver,[31] and the ability to use a sliding mesh.

2.2.2 | Sliding mesh

A sliding mesh[18] is used to rotate the AL in LBM. The sliding mesh allows the blades to always be in the same location relative to the surrounding cells, making the local velocity sampling more stable, as long as aeroelastic effects are not present. With the sliding mesh, local refinement can also be put on the blades and the tip vortex if needed, without having to refine the entire rotor disk. LBM is solved using explicit time schemes, leading to very small time steps. However, implicit solvers like SOWFA are often limited in their time step when using AL in order to avoid that the blade tip moves by more than one element in a single time step. This could potentially be mitigated with the sliding mesh, which could allow for larger time steps in implicit solvers.

Sliding meshes also present some disadvantages. Depending on the methods used, some degree of interpolation in the interface between the static and rotating cells is needed, leading to some dissipation of the flow structures in that region. Another issue is that the the rotor must fall in a cylindrical rotating region, separate from the tower, which can be challenging when simulating cases with very small tower clearance. Finally, a sliding mesh adds some computational cost to any simulation, which may or may not be compensated by coarsening the elements between the blades.

2.3 | AL method

The AL method combines the lifting line method with CFD, by using BET based on the local flow quantities provided by the CFD solver, and applying the airfoil forces on the CFD simulation as body forces. The main motivation for the AL is that much coarser meshes can be used, as solid walls are removed from the simulation.

2.3.1 | BET

BET is based on a 2D representation of rotor blades. Instead of considering the flow around the entire blade, cylindrical cuts centered around the rotor axis are made along the radius, and airfoil sectional quantities are calculated. For a given section with incoming axial flow $\vec{U}_a$, the rotation of the blade introduces a tangential flow velocity $\vec{U}_t$ in the airfoil reference frame. The sum of these two velocities is the relative velocity vector $\vec{U}_{rel}$, whose angle relative to the rotor plane is ϕ. The airfoil, which is rotated with respect to the rotor plane with a twist angle θ, has an angle of attack

$\alpha = \phi - \theta$. The airfoil generates a drag force $\vec{D}$, aligned with $\vec{U}_{rel}$ and a lift force $\vec{L}$, perpendicular to $\vec{U}_{rel}$. These forces can be transformed from the airfoil coordinate system into the rotor plane coordinate system, leading to an axial force $\vec{F}_a$, or thrust, and a tangential force $\vec{F}_t$, which is associated with the rotor torque. These vectors and angles are shown in Figure 2. Note that $\vec{U}_a$ does not need to equal the free-stream velocity and $\vec{U}_t$ does not need to equal the local kinematic velocity. The blades and wakes create a disturbance in the flow and this can be accounted for when using BET by sampling the velocities from the CFD simulation, in the case of AL.

BET needs to be corrected near the tip and root of the blade, due to the 3D flow features created by the finite number of blades. This is usually done with the Prandtl tip corrections. While AL simulation will produce tip and root vortices which will reproduce some of the desired effects of tip corrections automatically, the radius of said vortices will be limited by the cell size, meaning they can be much larger than in reality. For this reason, AL simulations still often employ root and tip corrections with some degree of empiricism.[32] A popular method[33] is based on the Glauert formulation, which multiplies the forces of each blade section by a function F_{tip}, based on the local section radius r.

$$F_{tip} = \frac{2}{\pi}\cos^{-1}\left(e^{-g\frac{B(R_{tip}-r)}{2r\sin(|\phi|)}}\right), \tag{5}$$

where B is the number of blades, R_{tip} is the tip radius, and g is an empirical correction function. In order to apply the correction to the root, a very similar equation is used:

$$F_{root} = \frac{2}{\pi}\cos^{-1}\left(e^{-g\frac{B(r-R_{root})}{2r\sin(|\phi|)}}\right), \tag{6}$$

where R_{root} is the root radius. In practical terms, F_{tip} and F_{root} reduce the forces on blade sections to zero at the tip and root, while having small effects near the blade center. According to the original blade element momentum theory these equations are derived from, ϕ is always positive. However, negative values could occur in case of strong turbulence or an oscillating rotor. Hence, as done in SOWFA, we force ϕ to be positive, in order to keep the exponentials below one and, consequently, the input to the arccosines in a valid range. The correction function g can be used to tune results to better match experimental values of a specific rotor.

The final expressions for the axial and tangential forces magnitudes from BET for a radial strip with a length Δr and a chord c are

$$F_a = F_{tip}F_{root}\frac{1}{2}\rho U_{rel}^2 \Delta r c[C_l(\alpha)\cos(\phi) + C_d(\alpha)\sin(\phi)], \tag{7}$$

$$F_t = F_{tip}F_{root}\frac{1}{2}\rho U_{rel}^2 \Delta r c[C_l(\alpha)\sin(\phi) - C_d(\alpha)\cos(\phi)], \tag{8}$$

where C_l and C_d are the lift and drag coefficients respectively, which are typically measured or calculated for a given airfoil shape and Reynolds number and provided in a table as a function of α.

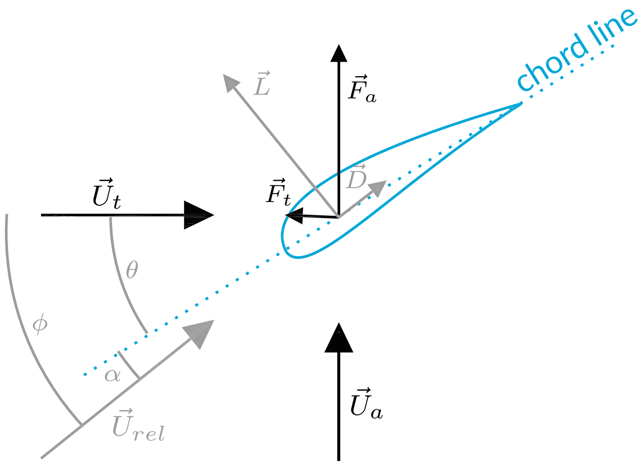

FIGURE 2 Vectors and angles involved in 2D flow around the section of a turbine blade.

2.3.2 | AL implementation

The AL model used in this study was implemented with the user-defined equations and scripting capabilities of PowerFLOW, which usually models blades as solid walls.[34] Henceforth, we refer to this as AL-LBM. Velocity sampling on the actuator points is achieved by averaging values of the cells within a user-defined radius (which we set to ϵ) around $\vec{x}_0$. Projection is done with the classical 3D-Gaussian distribution:

$$\vec{F}_{AL}(\vec{x}) = -\vec{F}_{BET}\frac{1}{\epsilon^3 \pi^{2/3}} e^{-\frac{\left\| \vec{x}-\vec{x}_0 \right\|^2}{\epsilon^2}}, \tag{9}$$

where $\vec{F}_{AL}$ is the body force to be applied in the CFD simulation at the coordinate $\vec{x}$; $\vec{F}_{BET}$ is the force obtained in the BET step or the vector sum of F_a and F_t; $\vec{x}_0$ is the position where $\vec{F}_{BET}$ is applied; and ϵ is the Gaussian width, a projection parameter that determines the size of the spherical region where the body forces are applied. The explicit time marching scheme uses much smaller time steps than OpenFOAM, about 30 times for the same grid size. Hence, we choose not to update the body forces at every time step. As the body forces are applied in the rotating reference frame, the conversion from the tangential direction to the vertical and horizontal directions is done in the first time step and does not need to be updated, hence if the flow is steady, the body forces remain constant over time. This is in contrast with a static mesh approach, where even steady forces need to be updated frequently, to reflect on the changes in azimuthal position.

Another difference between LBM and NS is that LBM is almost always compressible, even if weakly for most codes. AL codes such as SOWFA are typically based on the incompressible NS equations, meaning that they are limited to low Mach numbers. The methodology implemented in this work could be used for cases with stronger compressibility effects, although the literature on using AL for compressible flows is scarce (the only applications in aerospace to the authors' knowledge are for low Mach numbers[35,36]) and such applications would need to be validated.

3 | VALIDATION: NREL PHASE VI WIND TURBINE

For our validation, we consider the SOWFA example case based on the NREL Phase VI rotor,[37] a 10 m diameter turbine with two blades. The rotor is simulated here without the nacelle and tower for simplicity. Tip corrections are also omitted. Simulations are conducted with uniform incoming free-stream velocity $U_\infty = 7\,\text{m/s}$. The fluid density ρ is set to 1.23 kg/m^3, and the free-stream pressure p_∞ is set to one atmosphere in PowerFLOW and zero gauge pressure in SOWFA. The sides of the domain use free slip walls, and a pressure outlet boundary condition is employed at the end of the domain. Simulations use the same domain size. The turbine rotates at 71.9 RPM and 60 points per blade are used to describe the AL. Simulations are run for 20 s of physical time, and results are averaged over the last 5 s.

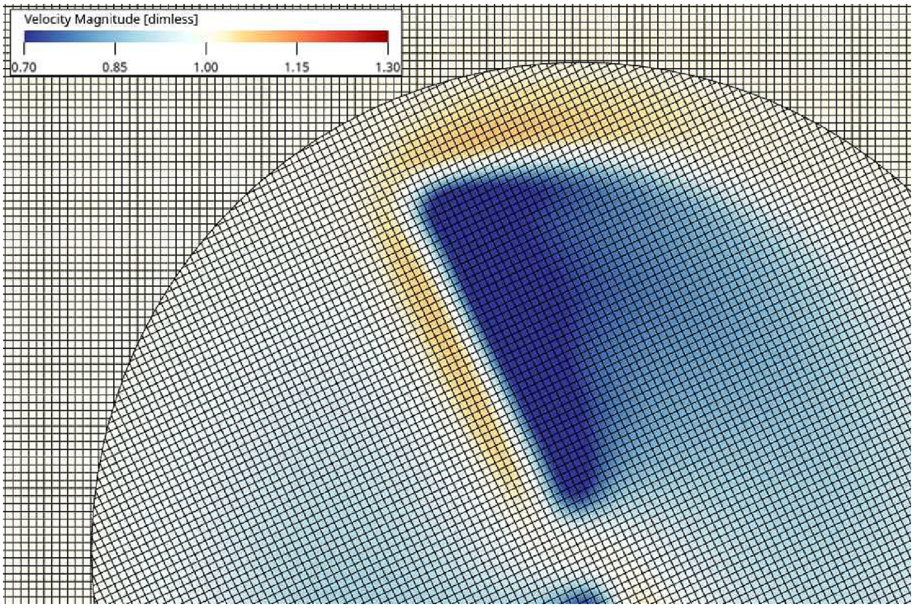

FIGURE 3 Slice on rotor plane showing an example of the computational grid and sliding mesh interface on the AL-LBM. Colors show velocity magnitude divided by U_∞

The sliding mesh interface in PowerFLOW is placed at a distance of 3.04ϵ away from the rotor disk, which is where the exponential in Equation (9) reduces to 1/10,000. The value of ϵ was set to $0.035D$,[4] where D is the rotor diameter. As these simulations are using coarse meshes, the tip vortices are resolved with approximately five elements across their diameter. This caused some dissipation of the vortices as they crossed the sliding mesh interface. Hence, we refined the LBM grid around the downstream face of the sliding mesh to avoid dissipation. This extra refinement is likely not needed for finer meshes, but is the main drawback in the sliding mesh approach. Everywhere else in the domain, the two meshes are identical, with a cell size of $D/100$ in most of the domain including the rotor wake, growing by factors of 2 up to a final cell size of $D/6.25$ in the outer edges of the simulation volume.

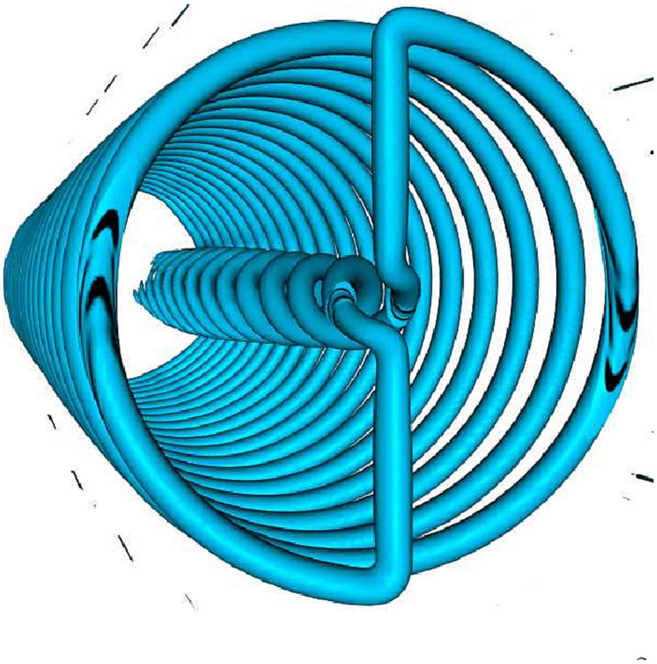

FIGURE 4 AL-LBM isosurfaces of $Q = 5s^{-2}$

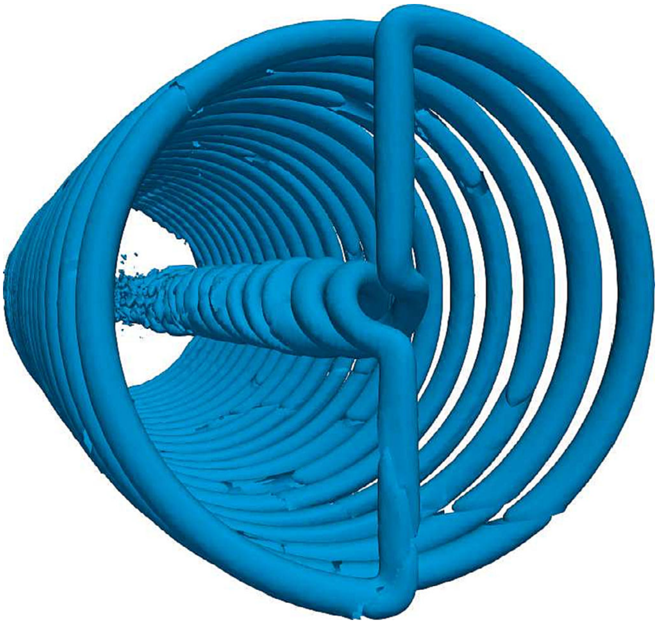

FIGURE 5 SOWFA isosurfaces of $Q = 5s^{-2}$

Figure 3 shows instantaneous AL-LBM results on a slice that is on the rotor plane. Velocity magnitude and the computational grid are shown, where the sliding mesh interface can be seen as the circle around the rotor. This image was captured with the rotor at an azimuth of 25° and is zoomed in to allow for the grid lines to be distinguished.

Isosurfaces of the Q criterion[38] from the AL-LBM and SOWFA simulations are shown in Figures 4 and 5, respectively. Results are very similar, with some differences in the breakup of the blade root vortices and small artifacts appearing around the sliding mesh. In both cases, the simulations are able to preserve the tip vortices as long as the fine mesh resolution of $D/100$ is maintained.

The thrust obtained with AL-LBM is 2% higher than in NS, while the torque is 5% higher. Given the differences in the velocity sampling between the two methods, we consider this good agreement. A breakdown of the forces is shown in Figures 6 and 7, where the axial and tangential forces along the blades are shown for the SOWFA and AL-LBM cases. Here, R is the rotor radius. While minor differences are present, we consider the codes to be in good agreement.

A quantitative comparison of the wake is shown in Figure 8, where the mean pressure (p) coefficient $Cp=(p-p_\infty)/(0.5\rho U_\infty^2)$, and the mean and variance of the streamwise (u), horizontal (v), and vertical (w) components of velocity are shown. Results are computed on a vertical line 10 m (about one diameter) downstream of the rotor. Results are in excellent agreement, with minor variations in some of the quantities, mostly near the root vortices. The AL-LBM simulations were conducted with merely six BET steps to update the body forces in the first few seconds of physical time. After that, velocities on the sampling points converged to a steady state and more couplings between BET and CFD became unnecessary. This would not be possible if the rotor were tilted, if strong tower effects were present, or if the incoming flow were turbulent.

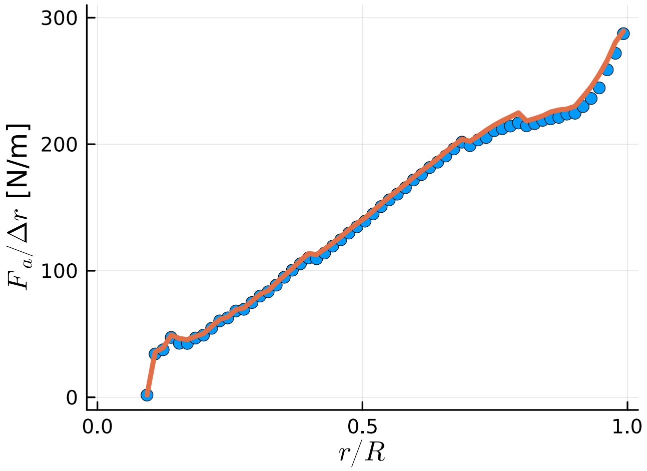

FIGURE 6 Axial force per unit span over blade radius. SOWFA results shown as blue circles, AL-LBM results as orange lines.

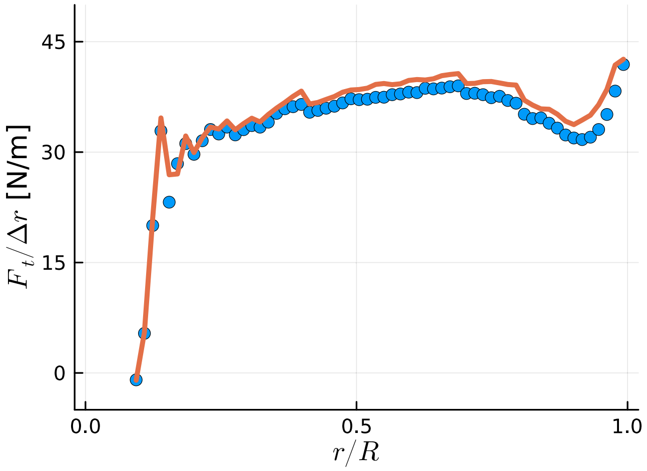

FIGURE 7 Tangential force per unit span over blade radius. SOWFA results shown as blue circles, AL-LBM results as orange lines.

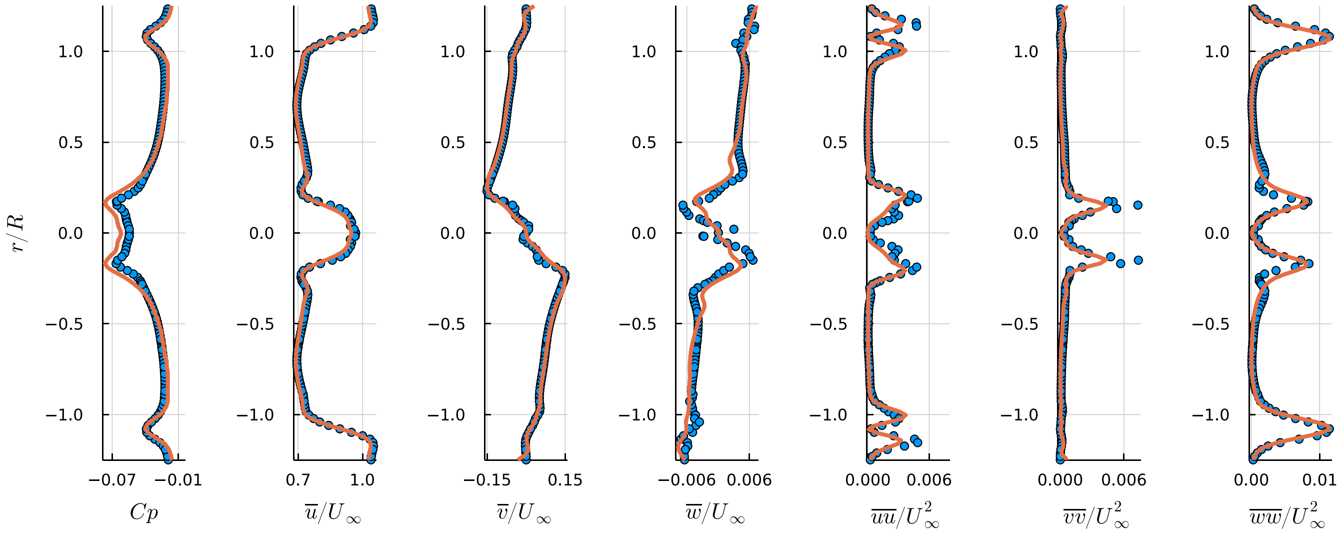

FIGURE 8 Pressure and velocity profiles along a vertical line 10 m downstream of the rotor. SOWFA results shown as blue circles, AL-LBM results as orange lines.

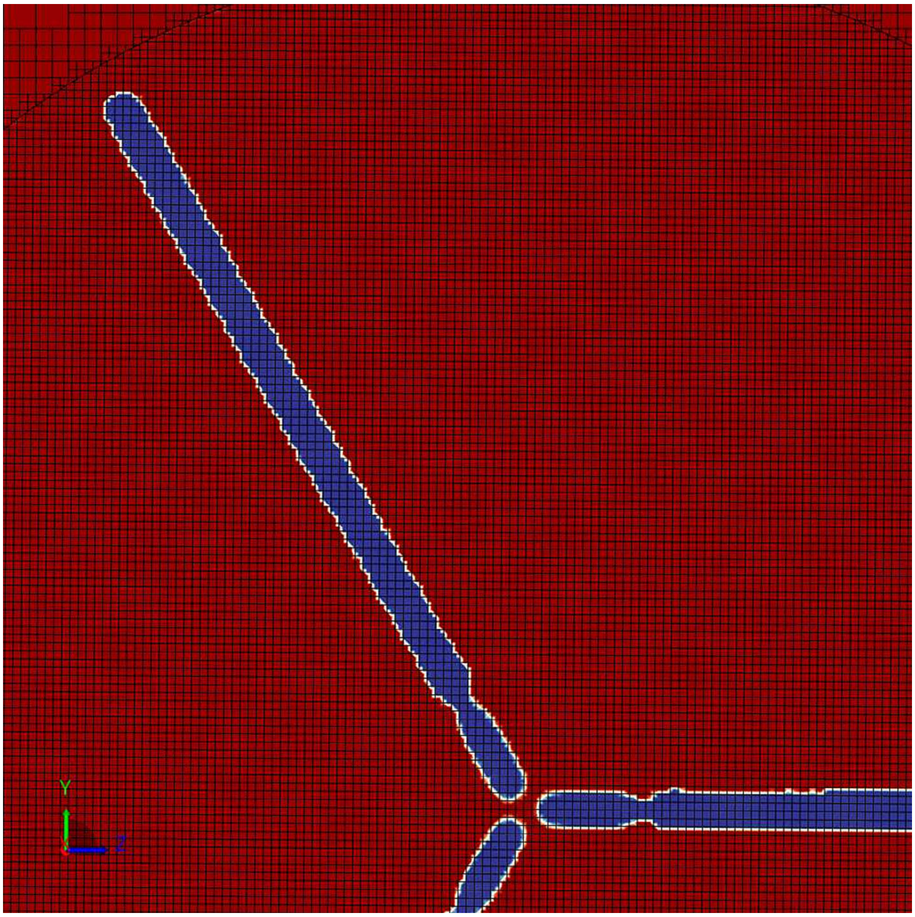

FIGURE 9 Rotor refined mesh. Actuator line shown in blue. Every other grid line drawn in black.

4 | REFINING THE BLADES: NREL 5 MW WIND TURBINE

We now move on to a finer mesh around the blades, to demonstrate the advantage of the sliding mesh. In AL simulations, it is often preferred to minimize the size of ϵ in order not to artificially extend the radius of the blades. This requires either mesh refinement or more advanced tip corrections.[39] Nonisotropic projection functions have also been proposed,[4] where ϵ is a fraction of the local airfoil chord and thickness. As ϵ tends to be at least twice the local cell size,[40] these nonisotropic models require much finer meshes than originally envisioned with AL. With the sliding mesh, we can refine the blades without refining the entire rotor, which we demonstrate in this section.

We use the SOWFA example case based on the NREL 5 MW rotor,[41] a 126 m diameter turbine with three blades. Again, we make a few simplifications by removing the tower, the nacelle, the tilt and precone angles, and the controller. These can be included in future simulations. Tip and root corrections are added with $g = 1$. Simulations are conducted with uniform incoming free-stream velocity $U_\infty = 8$ m/s. The turbine rotates at 9.16 RPM and 40 points per blade are used to describe the AL. We set ϵ to have the same size as the previous simulation relative to the finest cell size, that is, 3.5 times the finest cell. Other settings are the same as used in Section 3.

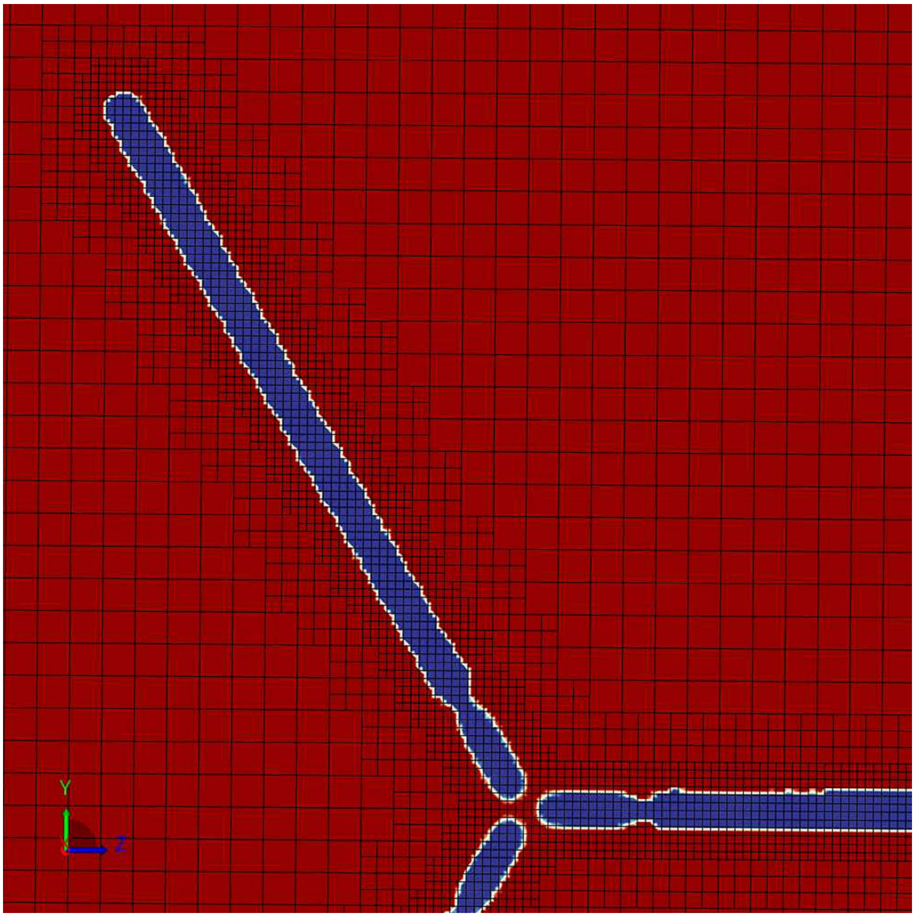

FIGURE 10 Blade refined mesh. Actuator line shown in blue. Every other grid line drawn in black.

TABLE 1 Ratio of computational costs associated with simulations using a static mesh with rotor refinement (Static Rotor) and a sliding mesh with blade refinement (Sliding Blade).

	$\frac{\text{Static Rotor}}{\text{Sliding Blade}}$ (ratio)
Number of cells	1.05
Number of time steps	4
CPU time (implicit NS)	4.2
CPU time (explicit LBM)	1.23

For the computational mesh, we start with the rotor and wake resolved with cubic cells of side $D/100$, as in Section 3. We use two approaches to refine this case: rotor refinement and blade refinement. For rotor refinement, we refine every cell within a distance of 3.04ϵ from the rotor disk. This is illustrated in Figure 9. For blade refinement, we refine cells at a distance of 3.04ϵ from the lines that represent the blades, as illustrated in Figure 10. In both cases, we quadruple the refinement near the body forces, leading to a local cell size of $D/400$. Compared to the starting point, the two approaches correspond to increasing the cell count in the simulation by 5% for rotor and 0.4% for blade refinement.

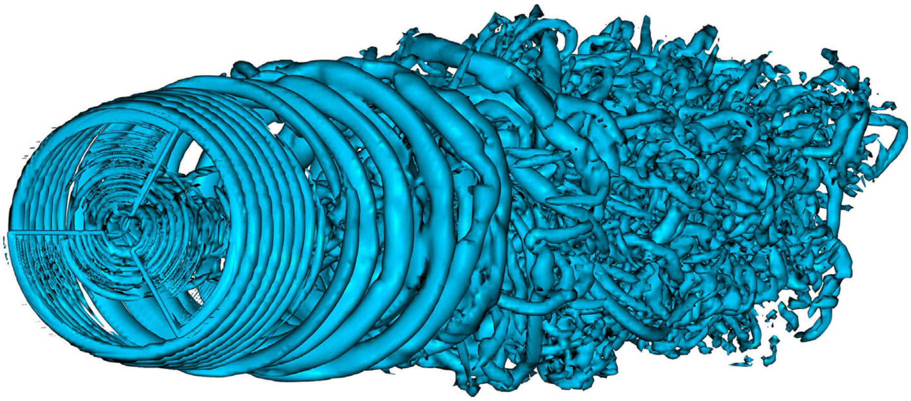

FIGURE 11 Isosurfaces of $\lambda_2 = 0.5[1.0] - 5D^2/U_\infty^2$. Rotor refined case in AL-LBM.

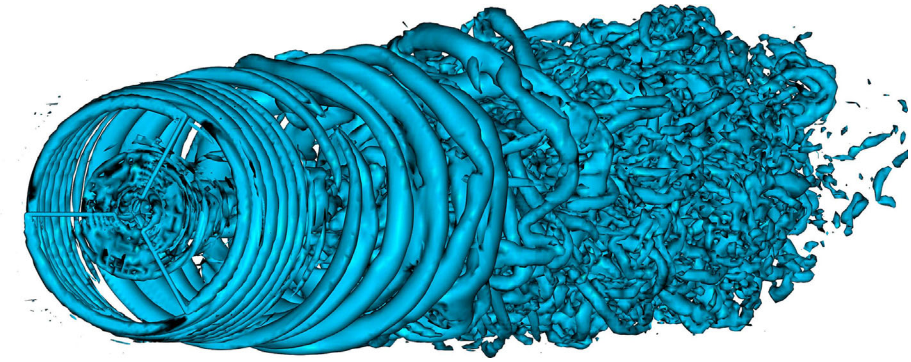

FIGURE 12 Isosurfaces of $\lambda_2 =$ -$5D^2/U_\infty^2$. Blade refined case in AL-LBM.

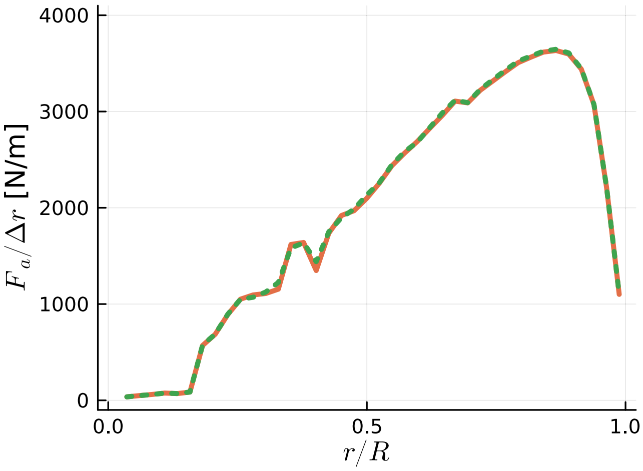

FIGURE 13 Axial force per unit span over blade radius. AL-LBM rotor refined results as orange lines, AL-LBM blade refined results as green dotted lines.

While the cell count increases do not seem excessive at a first glance, one must recall that usually AL simulations have their time step limited to avoid having the blade skip a cell in the simulation. A sliding mesh approach does not necessarily suffer from this limitation and even if certain implementations have a limit on the rotation allowed in a single time step on the sliding mesh interface, such region can be at a much coarser resolution than the blades. While rotor refinement increases the cell count by 5%, the associated time step reduction in a typical NS AL solver would mean quadrupling the number of time steps. Hence, for this particular example, an AL simulation using an implicit NS solver with a sliding mesh would be 4.2 times cheaper than the stationary mesh approach, the first using blade refinement and the latter rotor refinement.

In the case of LBM, simulations use explicit time marching schemes, with very small time steps that would avoid the blade skipping a cell. However, local time stepping is typically employed. The finest cells are updated every time step, with the next level being updated every 2 time steps, the next level every 4, then 8, and so on. In this simulation, the cells with size $D/400$ are hence updated four times more often than the cells with size $D/100$. This means the finest cells are expensive relative to the other cells, and in this particular case, the rotor refinement is about 23% more expensive than the blade refinement, even though the difference in the two approaches in terms of total cell count is only 6%.

These costs are summarized in Table 1. The conclusion is that, given a sliding mesh implementation that does not change the simulation cost substantially, the time savings of using such an approach over the typical static meshes can be significant, if finer resolution on the blades is desired. This is more true for NS than LBM, due to the small time steps inherent to most explicit LBM solvers.

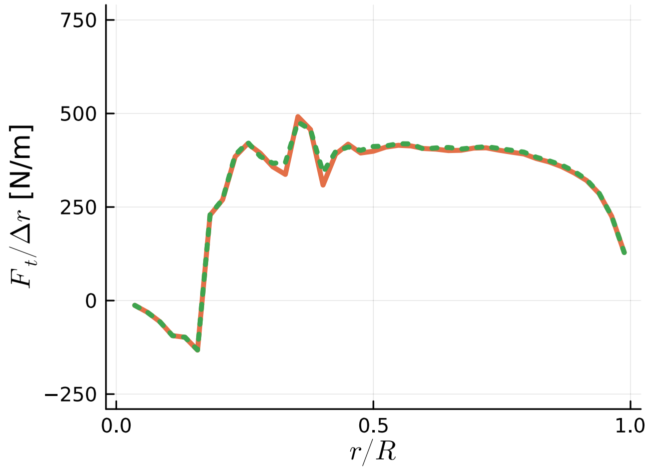

FIGURE 14 Tangential force per unit span over blade radius. AL-LBM rotor refined results as orange lines, AL-LBM blade refined results as green dotted lines.

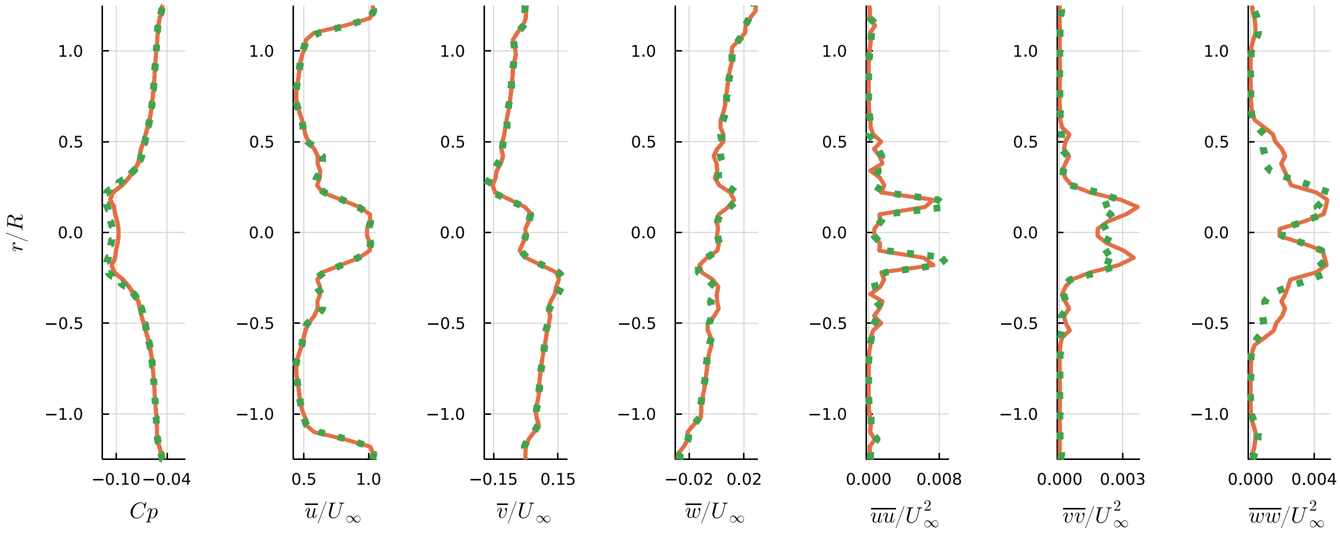

FIGURE 15 Pressure and velocity profiles along a vertical line 126 m downstream of the rotor. AL-LBM rotor refined results as orange lines, AL-LBM blade refined results as green dotted lines.

Cases with rotor and blade refinement were simulated in AL-LBM and are shown in Figures 11 and 12, respectively. Instantaneous isosurfaces of λ_2[42] are shown, with remarkably similar results, given the chaotic nature of the flow. The higher tip speed ratio and number of blades on the NREL 5 MW rotor compared to the NREL Phase VI rotor make the tip vortices interact earlier, creating complex wake structures.[43] Differences can be seen between the two simulations, with small secondary structures in the rotor plane, in the inner half of the rotor radius, being better resolved in the rotor refined case. However, these differences are small and the tip and hub vortices are very similar in the two simulations. For both cases, the cell size increases by a factor of 4 as soon as the vortices leave the blade or disk refined regions. Therefore, the tip vortices in Figures 11 and 12 have a substantially larger radius than the bound vortices that represent the blades.

The thrust and torque obtained with the two methods exhibit excellent agreement. The thrust of the blade refined case is 0.4% higher than the rotor refined case, while the torque is 1% higher. A breakdown of the axial and tangential forces is shown in Figures 13 and 14. The forces are almost indistinguishable, indicating the methods provide results that are consistent with each other.

Finally, we again look at a vertical profile one diameter downstream of the rotor and compare pressure and velocity components in Figure 15. We observe excellent agreement between the rotor and blade refined simulations, with only minor differences in some of the peak variance values near the hub. This demonstrates the capability of the sliding mesh to perform local refinement in AL simulations without appreciable accuracy loss, with the aforementioned substantial reductions in run time.

5 | CONCLUSIONS

In this work, we demonstrated the AL-LBM implementation and compared the results with a reference NS code, with good agreement between the solutions being observed. The LBM approach often presents performance advantages over unsteady NS solvers, as demonstrated by other studies.[12,15] Here, we implemented the AL method in a state of the art solver which can include thermal effects, hence being a potential replacement for NS simulations of wind farms, as the atmospheric boundary layer can be properly modeled. Our main focus was on the impact the sliding mesh in AL simulations, which we believe is a first in the literature. The advantages of the sliding mesh are:

- The mesh can be refined around the blades and tip vortices, while keeping most of the rotor disk at a coarse resolution. We demonstrated that refining the blades can be dramatically more efficient than refining the full rotor in a simulation with a sliding mesh, while providing the same results.
- The body forces do not have to be updated at every time step, which can lead to some computational cost being saved by skipping the BET steps. For rotors in non-homogeneous flows, the update frequency has to be high, but for homogeneous conditions, few (fewer than 10) updates to the body forces seems sufficient to obtain converged results.
- The sampling stage of the AL algorithm is always done in the same cells of the CFD simulation (in rigid cases), which can improve fluctuations induced from interpolation of the velocity field and can simplify the implementation of the algorithm.
- The sampling always occurring in the same cells also means that the CFD solver only needs to search for the cells that correspond to the ALs once, at the start of the simulation. Performing this search at every time step and within a parallel computing framework can be a bottleneck in AL codes.
- For implicit solvers, the time step can potentially be increased. In traditional AL simulations, the time step is limited so the blade does not travel more than one cell in one time step, which can be mitigated with a sliding mesh, in particular when cells are refined around the blades and coarsened before reaching the interface between rotating and fixed reference frames.

The sliding mesh also has some drawbacks, such as:

- The sliding mesh interface introduces challenges in terms of parallel computing, as the neighbors of the cells change over time. This and the calculation of the flow variables across the interface can increase the computational time.
- The interface between rotating and fixed reference frames can introduce some dissipation or numerical artifacts, depending on the implementation and grid resolution.
- With traditional AL implementations, the Gaussian body forces are allowed to come very close to each other, as with small tower clearance for wind turbines. A sliding mesh approach would require that every object represented by a body force is at a distance of twice the influence length of the body forces (here taken as 3.04ϵ). Hence, small tower clearance or simulations of multiple adjacent rotors can be a challenge. Being able to refine the blades means that the value of ϵ can also be reduced, potentially reducing the impact of this limitation.

Assuming the computational savings and losses when using a sliding mesh on a traditional AL setup cancel out, for typical cases with coarse blades one would likely not benefit tremendously from using a sliding mesh approach. On the other hand, when resolving the blades more finely, the advantages of blade refinement over rotor refinement are substantial. One could potentially achieve the same gains using adaptive mesh refinement, but the cost and complexity of adding it to a simulation are a significant drawback when compared to a sliding mesh approach.

The implementation used in this work can easily be extended to include improvements to the AL method, such as a non-spherical projection of the body forces. Future work can include the atmospheric boundary layer in stable, neutral, and unstable conditions, including complex terrain. Then, simulation of wind farms can be achieved taking full advantage of the LBM solver used in this work.

AUTHOR CONTRIBUTIONS

André F. P. Ribeiro developed the AL model in the LBM code, ran the AL-LBM simulations. Claudia Muscari ran the SOWFA simulations, contributed to the AL-LBM development with advice and information. Both authors contributed to the conception of the work, writing the paper, and results analysis.

CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflict of interests.

DATA AVAILABILITY STATEMENT

Data are available on request from the authors.

ORCID

André F. P. Ribeiro https://orcid.org/0000-0001-7519-9059

PEER REVIEW

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1002/we.2821.

REFERENCES

1. Bouckaert S, Pales AF, McGlade C, et al. Net zero by 2050: a roadmap for the global energy sector; 2021.
2. Veers P, Dykes K, Lantz E, et al. Grand challenges in the science of wind energy. *Science*. 2019;366(6464):eaau2027.
3. Sørensen JN, Shen WZ. Numerical modeling of wind turbine wakes. *J Fluids Eng*. 2002;124(2):393-399.
4. Churchfield MJ, Schreck SJ, Martinez LA, Meneveau C, Spalart PR. An advanced actuator line method for wind energy applications and beyond. In: 35th Wind Energy Symposium; 2017. American Institute of Aeronautics and Astronautics.
5. Xie S. An actuator-line model with Lagrangian-averaged velocity sampling and piecewise projection for wind turbine simulations. *Wind Energy*. 2021;24(10):1095-1106.
6. Schito P, Zasso A. Actuator forces in CFD: RANS and LES modeling in OpenFOAM. *J Phys: Confer Ser*. 2014;524(1):012160.
7. Shen WZ, Zhu WJ, Sørensen JN. Actuator line/Navier–Stokes computations for the MEXICO rotor: comparison with detailed measurements. *Wind Energy*. 2012;15(5):811-825.
8. Stevens RJAM, Martínez-Tossas LA, Meneveau C. Comparison of wind farm large eddy simulations using actuator disk and actuator line models with wind tunnel experiments. *Renew Energy*. 2018;116:470-478.
9. Shives M, Crawford C. Mesh and load distribution requirements for actuator line CFD simulations. *Wind Energy*. 2013;16(8):1183-1196.
10. Chen H, Teixeira C, Molvig K. Digital physics approach to computational fluid dynamics: some basic theoretical features. *Int J Modern Phys C*. 1997;08(4):675-684.
11. Chen H, Chen S, Matthaeus WH. Recovery of the Navier-Stokes equations using a lattice-gas Boltzmann method. *Phys Rev A*. 1992;45:R5339-R5342.
12. Manoha E, Caruelle B. Summary of the LAGOON solutions from the benchmark problems for airframe noise computations-III workshop. In: 21st AIAA/CEAS Aeroacoustics Conference; 2015. American Institute of Aeronautics and Astronautics.
13. Brès G, Prot F, Freed D. Properties of the lattice Boltzmann method for acoustics. In: 15th AIAA/CEAS Aeroacoustics Conference (30th AIAA Aeroacoustics Conference); 2012. American Institute of Aeronautics and Astronautics.
14. Rullaud S, Blondel F, Cathelain M. Actuator-line model in a lattice Boltzmann framework for wind turbine simulations. *J Phys: Confer Ser*. 2018;1037:022023.
15. Asmuth H, Olivares-Espinosa H, Ivanell S. Actuator line simulations of wind turbine wakes using the lattice Boltzmann method. *Wind Energy Sci*. 2020;5(2):623-645.
16. Churchfield MJ, Lee S, Michalakes J, Moriarty PJ. A numerical study of the effects of atmospheric and wake turbulence on wind turbine dynamics. *J Turbul*. 2012;13:N14.
17. Chen S, Liu H, Zheng C. Numerical study of turbulent double-diffusive natural convection in a square cavity by LES-based lattice Boltzmann model. *Int J Heat Mass Transfer*. 2012;55(17):4862-4870.
18. Zhang R, Sun C, Li Y, Satti R, Shock R, Hoch J, Chen H. Lattice Boltzmann approach for local reference frames. *Commun Comput Phys*. 2011;9(5):1193-1205.
19. Ferziger JH, Peric M. *Computational Methods for Fluid Dynamics*. 3rd ed.: Springer; 2002.
20. Smagorinsky J. General circulation experiments with the primitive equations: I. The basic experiment. *Mon Weather Rev*. 1963;91(3):99-164.
21. Issa RI. Solution of the implicitly discretised fluid flow equations by operator-splitting. *J Comput Phys*. 1986;62(1):40-65.
22. Rhie CM, Chow WL. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA J*. 1983;21(11):1525-1532.
23. Succi S. *The Lattice Boltzmann Equation: For Fluid Dynamics and Beyond*. Clarendon Press; 2001. https://books.google.com/books?id=OC0Sj_xgnhAC

24. Bhatnagar PL, Gross EP, Krook M. A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Phys Rev*. 1954;94:511-525.
25. Yakhot V, Orszag SA, Thangam S, Gatski TB, Speziale CG. Development of turbulence models for shear flows by a double expansion technique. *Phys Fluids A: Fluid Dyn*. 1992;4(7):1510-1520.
26. Fares E, Duda B, Ribeiro AFP, Knig B. Scale-resolving simulations using a lattice Boltzmann-based approach. *CEAS Aeronaut J*. 2018;9(4):721-733.
27. Ferris R, Sacks M, Cerizza D, Ribeiro AFP, Khorrami MR. Aeroacoustic computations of a generic low boom concept in landing configuration: Part 1—aerodynamic simulations. In: AIAA AVIATION FORUM; 2021. American Institute of Aeronautics and Astronautics.
28. Ribeiro AFP. Unsteady analysis of ground vortex ingestion with LBM-VLES. In: AIAA SCITECH Forum; 2021. American Institute of Aeronautics and Astronautics.
29. Khorrami MR, Konig B, Fares E, Ribeiro A, Czech M, Ravetta PA. Airframe noise simulations of a full-scale large civil transport in landing configuration. In: AIAA AVIATION FORUM; 2021. American Institute of Aeronautics and Astronautics.
30. Chen H, Teixeira C, Molvig K. Realization of fluid boundary conditions via discrete Boltzmann dynamics. *Int J Modern Phys C*. 1998;09(8):1281-1292.
31. Bonnal B, Vanoli E, Placko A, Combbias S. Methodology for simulation of soak-back in a helicopter engine bay using lattice Boltzmann method. In: Turbo Expo: Power for Land, Sea, and Air. American Society of Mechanical Engineers; 2020.
32. Martnez-Tossas LA, Meneveau C. Filtered lifting line theory and application to the actuator line model. *J Fluid Mech*. 2019;863:269-292.
33. Shen WZ, Mikkelsen R, Sørensen JN, Bak C. Tip loss corrections for wind turbine computations. *Wind Energy*. 2005;8(4):457-475.
34. Casalino D, van der Velden WC, Romani G. A framework for multi-fidelity wind-turbine aeroacoustic simulations. In: 28th AIAA/CEAS Aeroacoustics 2022 Conference; 2022. American Institute of Aeronautics and Astronautics.
35. Stokkermans TCA, van Arnhem N, Sinnige T, Veldhuis LLM. Validation and comparison of RANS propeller modeling methods for tip-mounted applications. *AIAA J*. 2019;57(2):566-580.
36. Merabet R, Laurendeau E. Actuator line method for helicopter rotors computations in various flight conditions. In: CASI AERO 2021; 2021. https://www.casi.ca/resources/Documents/AERO/2021/Full%20Papers/Actuator%20Line%20Method%20for%20Helicopter%20Rotors%20Computations%20in.pdf
37. Hand M, Simms D, Fingersh L, et al. Unsteady aerodynamics experiment phase VI: wind tunnel test configurations and available data campaigns. *Technical Report*. TP-500-29955; 2001.
38. Dubief Y, Delcayre F. On coherent-vortex identification in turbulence. *J Turbul*. 2000;1:N11.
39. Dag KO, Sørensen JN. A new tip correction for actuator line computations. *Wind Energy*. 2020;23(2):148-160.
40. Troldborg N. Actuator line modelling of wind turbine wakes. *PhD Thesis*, Technical University of Denmark; 2009. https://backend.orbit.dtu.dk/ws/portalfiles/portal/5289074/Thesis.pdf
41. Jonkman J, Butterfield S, Musial W, Scott G. Definition of a 5-MW reference wind turbine for offshore system development. *Technical Report*. TP-500-38060; 2009. https://www.nrel.gov/docs/fy09osti/38060.pdf
42. Jeong J, Hussain F. On the identification of a vortex. *J Fluid Mech*. 1995;285:69-94.
43. Felli M, Guj G, Camussi R. Effect of the number of blades on propeller wake evolution. *Exper Fluids*. 2008;44:409-418.

How to cite this article: Ribeiro AFP, Muscari C. Sliding mesh simulations of a wind turbine rotor with actuator line lattice-Boltzmann method. *Wind Energy*. 2023;1-15. doi:10.1002/we.2821