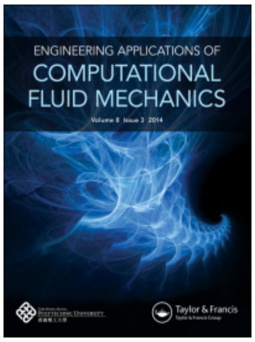


Engineering Applications of Computational Fluid Mechanics

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/tcfm20

Near-wall modeling of forests for atmosphere boundary layers using lattice Boltzmann method on GPU

Xinyuan Shao, Marta Camps Santasmasas, Xiao Xue, Jiqiang Niu, Lars Davidson, Alistair J. Revell & Hua-Dong Yao

To cite this article: Xinyuan Shao, Marta Camps Santasmasas, Xiao Xue, Jiqiang Niu, Lars Davidson, Alistair J. Revell & Hua-Dong Yao (2022) Near-wall modeling of forests for atmosphere boundary layers using lattice Boltzmann method on GPU, Engineering Applications of Computational Fluid Mechanics, 16:1, 2143-2156, DOI: 10.1080/19942060.2022.2132420

To link to this article: https://doi.org/10.1080/19942060.2022.2132420

ENGINEERING APPLICATIONS OF COMPUTATIONAL FLUID MECHANICS
2022, VOL. 16, NO. 1, 2143–2156
https://doi.org/10.1080/19942060.2022.2132420

Taylor & Francis
Taylor & Francis Group

Near-wall modeling of forests for atmosphere boundary layers using lattice Boltzmann method on GPU

Xinyuan Shao[a], Marta Camps Santasmasas[b], Xiao Xue[a], Jiqiang Niu[c,d], Lars Davidson[a], Alistair J. Revell[b] and Hua-Dong Yao[a]

[a]Department of Mechanics and Maritime Sciences, Division of Fluid Dynamics, Chalmers University of Technology, Göteborg, Sweden; [b]School of Mechanical, Aerospace and Civil Engineering, The University of Manchester, Manchester, UK; [c]School of Mechanical Engineering, Southwest Jiaotong University, Chengdu, Sichuan, People's Republic of China; [d]State Key Laboratory of Automotive Simulation and Control, Jilin University, Changchun, Jilin, People's Republic of China

ABSTRACT

In this paper, the simulation and modeling of the turbulent atmospheric boundary layers (ABLs) in the presence of forests are studied using a lattice Boltzmann method with large eddy simulation, which was implemented in the open-source program GASCANS with the use of Graphic Processing Units (GPU). A method of modeling forests in the form of body forces injected near the wall is revisited, while the effects of leaf area density (LAD) on the model accuracy is further addressed. Since a uniform cell size is applied throughout the computational domain, the wall-normal height of the near-wall cells is very large, theoretically requiring a wall function to model the boundary layer. However, the wall function is disregarded here when the forest is modeled. This approximation is validated based on the comparison with previous experimental and numerical data. It concludes that for the ABL conditions specified in this study as well as a large body of literature, the forest forces overwhelm the wall friction so that the modeling of the latter effect is trivial. Constant and varying LAD profiles across the forest zone are defined with the same total leaf area despite the varying one being studied previously. It is found that the two LAD profiles provide consistent predictions. The present forest modeling can therefore be simplified with the use of the constant LAD without degrading the model accuracy remarkably.

ARTICLE HISTORY
Received 9 July 2022
Accepted 26 September 2022

KEYWORDS
Wind energy; atmospheric boundary layers; forest modeling; lattice Boltzmann method; large-eddy simulation; GPU

1. Introduction

When wind farms are located within forest terrain, the presence of trees are known to generate strong atmospheric turbulence, which can in turn lead to an adverse impact on the performance and fatigue life of wind turbines (Nebenführ & Davidson, 2017). In recent years, there has been increased focus on the siting of wind turbines within forests. This is in part motivated by the opportunity to reduce the visual and environmental noise footprint of turbines, since the forest canopy and background noise provide a natural screen. It is also driven by need, since around 30% of the Earth's land surface is covered by forest. It is thus increasingly relevant to find an effective method to simulate wind field under the influence of forests to help inform the siting at the early design phase of a wind farm.

The effect of forests may be considered as a kind of surface roughness which is modeled by introducing a roughness length (Lo, 1990). However, this method is only applicable to short vegetation. The permeability of the foliage should also be taken into account for relatively high forests. Shaw and Schumann (1992) presented the effect of forests in their simulations as a drag force applied by the trees on the flow and the drag force varies according to the varying leaf area density (LAD). Their method has been accepted by many researchers and has been used to investigate many aspects of canopy flows, for example, coherent structures in canopy flow (Dupont & Brunet, 2009; Finnigan et al., 2009; Nezu & Sanjou, 2008), the influence of LAD on canopy flow (Dupont & Brunet, 2008b) and edge flow due to heterogeneity in forest structure (Dupont & Brunet, 2008a).

Large-eddy simulation (LES) has proven to be an useful tool to unravel the engineering problems such as wind farms flow prediction (Stevens et al., 2014, 2015; Stevens & Meneveau, 2017). The atmospheric boundary layer (ABL) can be simulated using conventional finite difference methods (FDM) or finite volume methods (FVM), for instance, PALM (Maronga et al., 2015), ICON (Dipankar et al., 2015), and MicroHH (Van

CONTACT Xiao Xue xiaox@chalmers.se

Heerwaarden et al., 2017). There are many studies which have endeavored to combine LES with forest models. Shaw and Schumann (1992) successfully represented a forest as a drag layer and heat sources in their LES. Shaw and Patton (2003) further refined the forest model by including skin friction effect under the LES framework.

LES is commonly adopted in the Navier–Stokes (NS) equation framework. Despited LES has been largely reduce the computational cost comparing to the direct numerical simulations(DNS), however, it is still computationally challenging due to the data-dependency nature raised by the conventional CFD methods.LBM can improve the parallel efficiency since each lattice cells information is stored and updated locally (Krüger et al., 2017). Studies have been conducted to involve LES into the framework of lattice Boltzmann models (LBM) (Hou et al., 1994). LBM is thus seen as a good alternative to more common methods based on NS solvers. In the past decades, LBM has become more widespread on account of its broad applicability across a range of complex fluid dynamics problems ranging from micro-nano scales (Belardinelli et al., 2015; Xue et al., 2020, 2018) to macroscopic scales (Filippova et al., 2001; Toschi & Bodenschatz, 2009; Xue et al., 2022) under low Mach numbers. A recent study combined LBM and LES to simulate ABL flows (Feng et al., 2021). They simulated neutral, stable and convective ABLs over a forest in reference to the cases in Nebenführ and Davidson (2015), where LES implemented with a FVM was used. Their simulations included the forest effects using a forest model. Furthermore, since the mesh resolution near the ground wall was not sufficient to resolve small-scale turbulent fluctuations, the effect was recovered using the Monin–Obukhov wall model. Their simulations results showed good agreement with the reference data in Nebenführ and Davidson (2015). However, the detailed comparison between the varying LAD and constant LAD is not presented. Also, most LES LBM computation are based on multi CPU architectures (Harwood, O'Connor et al., 2018; Krause et al., 2021; Latt et al., 2021), the intrinsic nature of parallel-friendly nature makes LBM well-suited for GPU architectures.

A difficulty in the application of LBM on GPUs is that although various mesh refinement techniques have been proposed, they suffer from reduced computational efficiency. Another concern is that as lattice points remain in a cubic structure, the mesh refinement is generally limited to octree based methods with a scale of 1-2 in each direction. It means that high aspect ratio cells, which are often used in wall-bounded turbulent flows, are generally not possible. To avoid loss of the computational efficiency, meshes with uniform cells can be adopted for ABL simulations on GPUs. However, a wall function in principle should be used because the near-wall mesh resolution is not sufficient to resolve small scale flow fluctuations. When the forest effects near ground walls are modeled, questions arise about whether the forest drag force is dominant and whether the wall function can be neglected. Moreover, it is unclear whether the accuracy of the forest modeling is sensitive to the LAD profile. These problems motivate the present study. We will investigate near-wall modeling to implement a uniform cell size in the whole computational domain, replacing the local mesh refinement. The effects of the wall function and forest model will be addressed. The influence of the LAD profile will be studied.

2. Methodology

2.1. The lattice Boltzmann method

The flow is described based on the mesoscale lattice Boltzmann equations. The classical Bhatnagar–Gross–Kroog (BGK) LBM collision operator and D3Q19 model are adopted for the LBM. This model is a three-dimensional lattice model with 19 discretized velocity directions $\mathbf{c}_i$ ($i = 0 \cdots Q-1$). We consider $f_i(\mathbf{x}, t)$ as the discretized particle's probability distribution function on the ith direction of a lattice cell. The lattice cell is located at position $\mathbf{x}$ at time t. The LBM governing equation for the fluid, considering collision and forcing, can be written as:

$$f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) - \Omega \left[f_i(\mathbf{x}, t) - f_i^{\mathrm{eq}}(\mathbf{x}, t) \right] + \Delta t F_i(\mathbf{x}, t), \tag{1}$$

where Ω is a collision kernel (Krüger et al., 2017; Succi, 2001), Δt is the marching time step, F_i is the volume force acting on the fluid following the approach from Guo et al. (2002).

The collision kernel is the classical BGK collision kernel which has been widely adopted to various applications (Succi, 2001). The BGK collision operator fixes the single relaxation time τ for the colliding process. Thus, the collision operator can be represented as:

$$\Omega \left[f_i(\mathbf{x}, t) - f_i^{\mathrm{eq}}(\mathbf{x}, t) \right] = \frac{\Delta t}{\tau} \left[f_i(\mathbf{x}, t) - f_i^{\mathrm{eq}}(\mathbf{x}, t) \right] \tag{2}$$

The collision kernel relaxes the distribution function towards the local equilibrium f_i^{eq}:

$$f_i^{\mathrm{eq}}(\mathbf{x}, t) = \omega_i \rho(\mathbf{x}, t) \left[1 + \frac{\mathbf{c}_i \cdot \mathbf{u}(\mathbf{x}, t)}{c_s^2} + \frac{[\mathbf{c}_i \cdot \mathbf{u}(\mathbf{x}, t)]^2}{2c_s^4} - \frac{[\mathbf{u}(\mathbf{x}, t) \cdot \mathbf{u}(\mathbf{x}, t)]}{2c_s^2} \right], \tag{3}$$

where ω_i is a suitable weight needed to impose the isotropy in the interaction, $\rho(\mathbf{x}, t)$ and $\mathbf{u}(\mathbf{x}, t)$ are the macroscopic hydrodynamic quantities for density and velocity, respectively. The volume force $F_i(\mathbf{x}, t)$ in Equation (1) can be obtained by:

$$F_i(\mathbf{x}, t) = \left(1 - \frac{1}{2\tau}\right)\omega_i \times \left[\frac{\mathbf{c}_i - \mathbf{u}(\mathbf{x}, t)}{c_s^2} + \frac{\mathbf{c}_i \cdot \mathbf{u}(\mathbf{x}, t)}{c_s^4}\mathbf{c}_i\right] \cdot \mathbf{F}, \tag{4}$$

where $\mathbf{F}$ is the volume acceleration.

The macro-scale quantities for the density, momentum and momentum flux can be calculated from the distribution function, the discrete velocity, and the volume force:

$$\rho(\mathbf{x}, t) = \sum_{i=0}^{Q-1} f_i(\mathbf{x}, t), \tag{5}$$

$$\rho(\mathbf{x}, t)\mathbf{u}(\mathbf{x}, t) = \sum_{i=0}^{Q-1} f_i(\mathbf{x}, t)\mathbf{c}_i + \frac{1}{2}\mathbf{F}\Delta t, \tag{6}$$

$$\Pi(\mathbf{x}, t) = \sum_{i=0}^{Q-1} f_i(\mathbf{x}, t)\mathbf{c}_i\mathbf{c}_i, \tag{7}$$

where the momentum flux can be defined as $\Pi(\mathbf{x}, t) = \Pi_{\text{eq}}(\mathbf{x}, t) + \Pi_{\text{neq}}(\mathbf{x}, t)$, where non-equilibrium parts are defined as:

$$\Pi_{\text{neq}}(\mathbf{x}, t) = \sum_{i=0}^{Q-1} f_i^{\text{neq}}(\mathbf{x}, t)\mathbf{c}_i\mathbf{c}_i, \tag{8}$$

f_i^{neq} is the non-equilibrium distribution function which can be obtained via $f_i^{\text{neq}}(\mathbf{x}, t) = f_i(\mathbf{x}, t) - f_i^{\text{eq}}(\mathbf{x}, t)$.

2.2. Smagorinsky subgrid-scale model

The basic idea of LES is not only to resolve relatively large scales, but also to use a subgrid-scale (SGS) model to estimate unresolved small scales. The detailed implementation of the Smagorinsky SGS model can be found in Hou et al. (1994) and Koda and Lien (2014). Here, we briefly summarize the essential part of this model. The key of the LES modeling of Smagorinsky is to model the effective viscosity ν_{total}, which can be seen as the sum of the molecular viscosity ν_0 and the turbulent viscosity ν_t:

$$\nu_{\text{total}} = \nu_0 + \nu_t, \tag{9}$$

where ν_t is represented by:

$$\nu_t = C_{\text{smag}}\Delta^2 \left|\bar{\mathbf{S}}\right|, \tag{10}$$

where C_{smag} is the Smagorinsky constant, Δ represents the filter size, and $|\bar{\mathbf{S}}|$ is the filtered strain rate tensor, which can be obtained by:

$$|\bar{\mathbf{S}}| = \frac{-\tau_0\rho c\Delta x + \sqrt{(\tau_0\rho c\Delta x)^2 + 18\sqrt{2}\rho C_{\text{smag}}\delta^2 Q^{1/2}}}{6\rho C\delta^2}, \tag{11}$$

where τ_0 is the original relaxation time from the input, $c = \Delta x/\Delta t$, and $Q^{1/2}$ can be written as:

$$Q^{1/2} = \sqrt{\Pi_{\text{neq}} : \Pi_{\text{neq}}}, \tag{12}$$

where Π_{neq} is the non-equilibrium part of the momentum flux tensor Π shown in Equation (8). Following (Koda & Lien, 2014), we can obtain the total relaxation time τ_{total}, which is written as:

$$\tau_{\text{total}} = \frac{\tau_0}{2} + \frac{\sqrt{(\tau_0\rho c)^2 + 18\sqrt{2}C_{\text{smag}}Q^{1/2}}}{2\rho c}. \tag{13}$$

Finally, we replace τ in Equation (2) with τ_{total} and obtain the Smagorinsky SGS collision kernel.

2.3. Forest model near the wall

We model the effects of the forest by introducing an additional drag force near the wall. The drag force is modeled with the help of the forest drag coefficient C_D, the LAD a_f and the local wind velocity $\boldsymbol{u} = (u, v, w)$ (Shaw & Schumann, 1992). Thus, the drag force of the forest, which is added in the streamwise direction, can be written as:

$$F_f(x, y, z, t) = -C_D a_f(y) \left|\boldsymbol{u}(x, y, z, t)\right| u(x, y, z, t), \tag{14}$$

where y refers to the vertical distance to the wall and $u(x, y, z)$ is the streamwise velocity. In this study, for the purpose of comparison with the previous study (Bergström et al., 2013), C_D is set to 0.15, which has been adopted in their work.

According to Lalic and Mihailovic (2004), the LAD distribution $a_f(y)$ can be described by the function:

$$a_f(y) = a_{f_m}\left(\frac{h - y_m}{h - y}\right)^n \exp\left[n\left(1 - \frac{h - y_m}{h - y}\right)\right] \tag{15}$$

where

$$n = \begin{cases} 6 & 0 \le y < y_m \\ \frac{1}{2} & y_m \le y \le h \end{cases} \tag{16}$$

The parameter h is the total height of the forest, and y_m is the location where $a_f(y)$ gets its maximum value a_{f_m}.

3. Numerical implementation

The simulations presented in this paper are performed using the open source CFD software GASCANS (Santasmasas, 2021), which implements LBM on multiple graphic processing units (GPUs) using CUDA and C++. The main aim of GASCANS is to provide fast and accurate simulation of turbulent flows over complex geometries. To further increase its usability, it incorporates a synthetic eddy method (SEM) to generate turbulent velocity fluctuations from mean flow data and GASCANS can be coupled at run time with another CFD software (Santasmasas, 2021). It also implements immersed boundary method (IBM) and a structural solver for the simulation of filaments submerged in fluid.

The flow solver implements the BGK collision scheme, the LES Smagorinsky turbulence model and forcing scheme described in Section 2. The algorithm follows a merged stream-collision time loop and stores the particle distribution functions using a double-population scheme (Latt et al., 2021). The simulation data is stored in a mesh formed by equally sized cubic cells using a structure of arrays (SOA) pattern.

Figure 1 shows the GASCANS architecture and interactions between its components; the features not used for this paper are grayed out. The code is divided into two sections: the Application contains the user interface and inputs, while the Library contains the core software. This structure allows for the Application to be reworked or substituted to suit the user's needs without modifying the core of GASCANS. The user inputs are through the Parameters class, and the terrain and forest geometry is read from point cloud files. See (Santasmasas, 2021) for a detailed explanation of the GASCANS functionality and structure. The point cloud file contains a list of coordinates covering the forest's volume. GASCANS reads the file and assigns a code to each cell that contains one or more forest coordinates. The code for each forest cell contains the following information:

- h: height of the forest, calculated as the bounding box of the forest object in the vertical direction.
- y: distance in the vertical direction between the lower bounding box of the forest object and the current cell.
- y_m: distance in the vertical direction between the lower bounding box of the forest and the maximum value of the LAD distribution.

This information is stored into the forest cell's code using a bit mask to reduce the GPU memory required to store the information. Algorithm 1 shows the forest implementation in GASCANS.

4. Validation of simulation platform

Channel flow is a typical case of wall-bounded turbulence. The accuracy of the simulation platform, which uses the LES with the Smagorinsky SGS model, is validated based on the database of direct numerical simulation (DNS) at $Re_\tau = 180$ in Moser et al. (1999).

The computational domain is shown in Figure 2. The domain dimensions are non-dimensionalized based on the height of $H = 1$ meter. That is, the dimensions are $L_x \times L_y \times L_z = 12 \times 2 \times 4\, m^3$ in the stream-

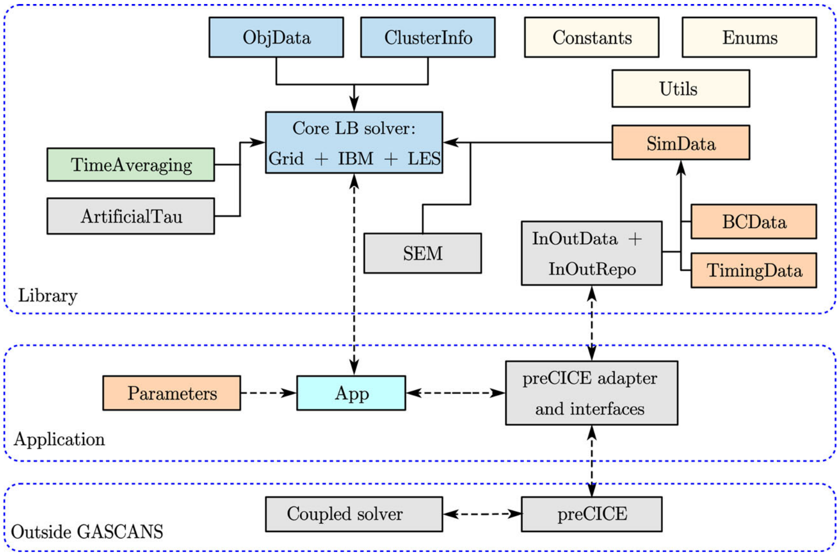


Figure 1. Illustration of GASCANS architecture and relationship between its components. The solid arrows indicate aggregation (i e. the classes at the beginning of the arrow are part of the class that the arrow points to) and the dashed arrows indicate primary communication. The isolated boxes represent definitions and utilities accessible by the core LB solver.

Algorithm 1 Implementation of forest forcing with varying leaf area density in GASCANS.

```
Read the user input data: $C_D$, $a_{f_m}$, $y_m$ in the vertical direction.
Read the forest geometry and generate the forest code for each forest cell.
Set the body force for all cells to 0.
for time = 0 to time = last time do
    for all cells do
        Stream
        if cell x is forest then
            Calculate local cell velocity $\mathbf{u}(\mathbf{x}, t)$
            Calculate forest force Equation (14)
            Correct the local cell velocity using the forest force
        end if
        Collide using the calculated body force Equation (1)
    end for
end for
```

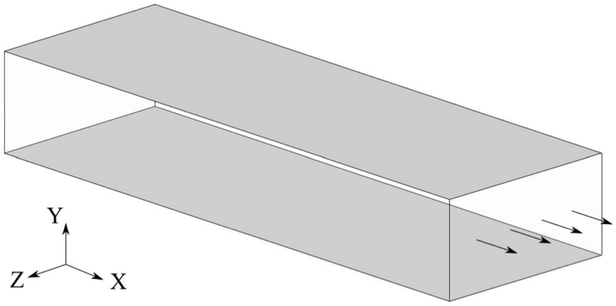


Figure 2. The computational domain of the turbulent channel flow. The domain dimensions, normalized with the length scale $H = 1$ meter, are $L_x \times L_y \times L_z = 12 \times 2 \times 4$ along the x, y and z axes. The walls are marked in gray, and the arrows indicate the flow direction.

wise, vertical and spanwise directions. The mesh consists of cubic lattice cells. There are 31 cells per meters. Correspondingly, the mesh contains 372 × 62 × 124 lattice cells.

The upper and lower walls are set with the no-slip boundary condition, and the remaining boundaries of the computational domain with the periodic boundary condition. The flow is driven by a volume body force from Equation (4), which is defined as:

$$\mathbf{F} = \left[\frac{\rho u_\tau^2}{H}, 0, 0\right]. \tag{17}$$

The body force acts equivalently as the pressure gradient that balances the wall shear stress in the fully developed channel flow.

Table 1. The simulation parameters for the turbulent channel flow.

Parameter	Value	Parameter	Value
Δx [m]	1/31	L_x	12[m] (372 LBU)
Δt [s]	0.001	L_y	2[m] (62 LBU)
ν [m^2/s]	1/3250	L_z	4[m] (124 LBU)
F [N]	0.002986	C_{smag}	0.01 LBU

The parameters set for the simulation are listed in Table 1. Here C_{smag} is the Smagorinsky model constant, as shown in Equation (10). The dimensionless density ρ is set to unity in the lattice Boltzmann unit (LBU). The fluid is incompressible. The channel flow simulation is initialized with the zero velocity field driven by the body force F given in Table 1. A small block of 15 × 15 × 64 m^3 is placed at $x = 1\ m$ in the middle of the spanwise width. The simulation with the block is first performed for 16.5 periods to develop turbulence. Here one period denotes the time that the flow takes to pass through the domain. Then, the block is removed, and the simulation continues for another 82 periods before computing the spatial-averaged statistics.

Figure 3 shows the instantaneous streamwise velocity distribution and the Q-criterion of $Q = 0.0002/s^2$. A plane is placed at the vertical position of $\boldsymbol{y^+ = 9}$ to visualize streamwise velocity contours. Streaks featuring the boundary layer are observed. This is consistent with the previous DNS (Moser et al., 1999). Plentiful turbulent structures are identified within the flow field based on the Q-criterion. Hairpin vortices exist near the walls, leading to streak contours of the streamwise velocity. As shown in Figure 4, the statistics of the flow quantities from the present LBM LES method are compared to Koda's LBM LES data (Koda & Lien, 2014), and the FVM DNS by Moser et al. (1999). The non-dimensional time-averaged streamwise velocity is defined as $\langle u^+ \rangle = \langle u \rangle / u_\tau$. Here $\langle \cdot \rangle$ is the averaging operator, and u_τ denotes the friction velocity. The wall normal distance is defined as $y^+ = y u_\tau / \nu$. The averaging is done in time and then in x-z planes that are parallel to the top and bottom walls. The non-dimensional root mean square (RMS) velocity components (u^+_{rms}, v^+_{rms} and w^+_{rms}) are also normalized with the friction velocity u_τ. The present results are consistent with Koda's LBM LES data, especially in the near wall region. But these results exhibit discrepancies in comparison with Moser's DNS data. It is observed that the first layer cell heights near the wall are $\Delta y_1^+ = 3$ in the LES cases, indicating that the resolution of the LES mesh is not fine enough to resolve the boundary layer. This is the reason why both LBM LES methods underestimate $\langle u^+ \rangle$ as compared to DNS. Due to the same reason, the RMS values of the velocity components predicted in the LES cases are less potent than the DNS. It suggests that a

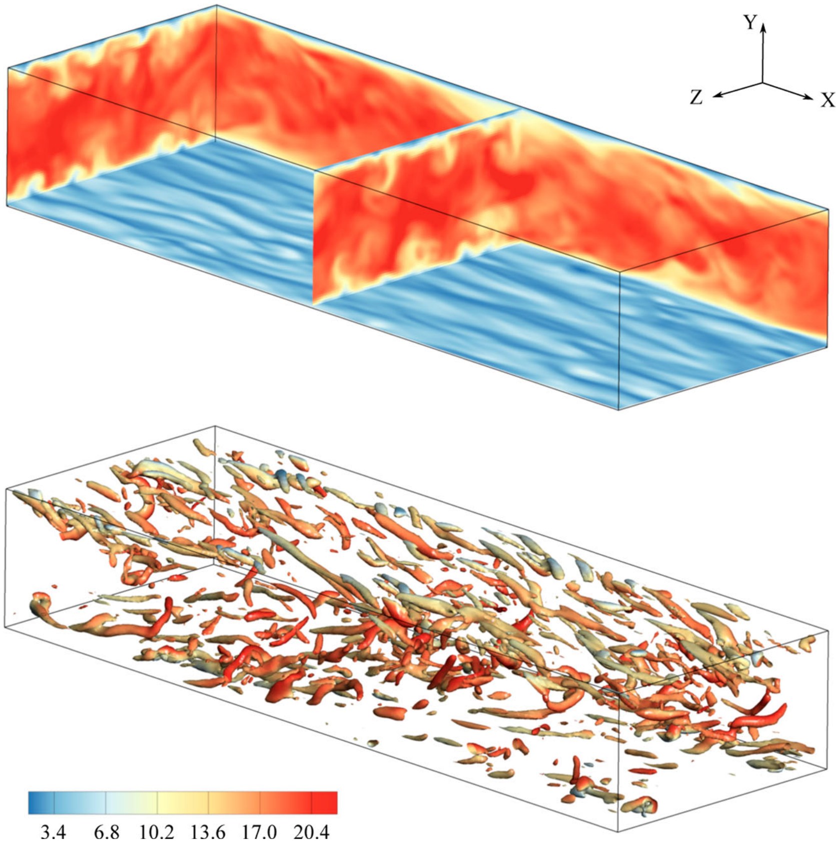


Figure 3. (a) The instantaneous non-dimensional streamwise velocity, u/u_τ, of the turbulent channel flow at $Re_\tau = 180$. The bottom plane shown here is located at $y^+ = 9$. (b) Isosurfaces of the Q-criteron of $Q = 0.0002/s^2$, which are colored with the streamwise velocity.

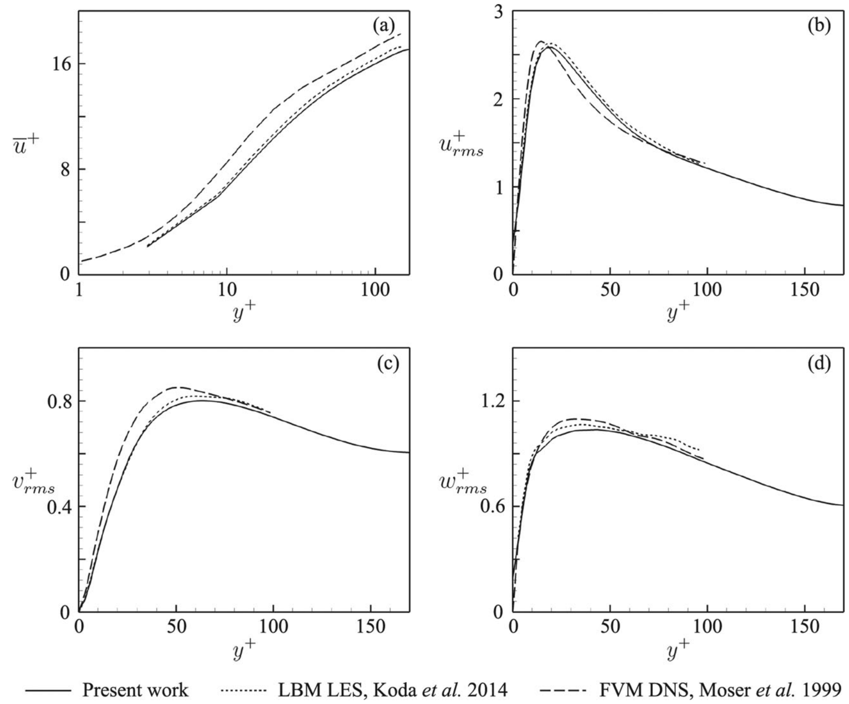


Figure 4. The comparison of the present LBM LES, Koda's LBM LES (Koda & Lien, 2014) and the FVM DNS (Moser et al., 1999). (a) The non-dimensional averaged streamwise velocity $\langle u^+ \rangle$ as a function of y^+. (b–d) The non-dimensional RMS velocity components in the streamwise, wall normal and spanwise directions.

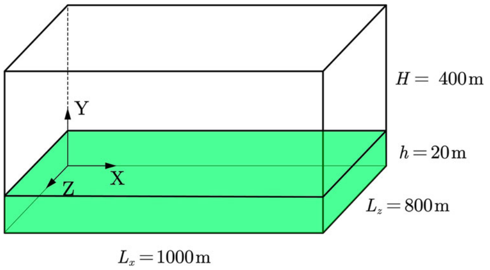


Figure 5. The computational domain of the ABL. The green color marks out the zone where a homogeneous forest with a height of $h = 20\,m$ is modeled above the ground. The freestream flow moves along the x-axis.

wall function or similar wall treatment should be applied to the near wall region to remedy the underestimations. We therefore use a forest model for the wall treatment in the ABL simulations in the next section. Nonetheless, the largest discrepancy of $\langle u^+\rangle$ is less than 5%, and the largest RMS velocity difference less than 10%.

5. Atmospheric boundary layers over forest

5.1. Case description

A classical ABL case reported in Nebenführ and Davidson (2014, 2015) is studied using the present simulation platform. The computational domain is shown in Figure 5. The physical dimensions are $L_x \times L_y \times L_z = 1000 \times 400 \times 800\,m^3$, which was also used and validated in Nebenführ and Davidson (2014, 2015). To produce the representative forest effects, a forest with a homogeneous distribution over the ground is assumed (Nebenführ & Davidson, 2014). The height of the forest is $h = 20\,m$. The freestream flow direction is set along the x axis. The forest friction velocity is defined as $u_* = [\langle u'v'\rangle^2 + \langle v'w'\rangle^2]^{1/4}$ at the location of $y/h = 2$ (Nebenführ & Davidson, 2014, 2015).

The flow is driven by a constant non-dimensional body force of 2.38×10^3 per unit volume on the basis of u_* and L_x. The force is imposed in the whole domain. At the bottom wall below the forest, a no-slip boundary condition is applied. The inlet and outlet are set with the periodic boundary condition, and the spanwise side boundaries are also set with the same boundary condition. The upper boundary is specified with a forced equilibrium boundary condition (Latt et al., 2008). The validity to apply this boundary condition will be addressed in the following discussion. Also, the upper wall is sufficiently far away from the bottom, and only the boundary layer region near the bottom wall where wind turbines are embedded are of interest for the wind energy harvesting, this forced equilibrium boundary condition is suitable for the present ABL case.

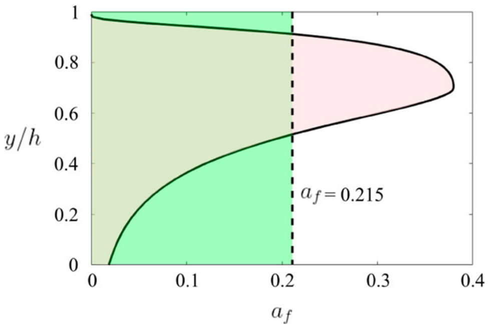


Figure 6. The distributions of LAD in two situations such as constant LAD of $a_f = 0.215$, indicated by the dash line, and varying LAD with a_f following Equation (15) with $z_m = 0.7h$ and $a_{fm} = 0.38$, indicated by the solid curve. The green colors the area of $a_f = 0.215$, which is equal to the area of the varying LAD case colored in red.

Table 2. The simulated ABL cases in terms of the LAD.

	Case 1 (Constant LAD)	Case 2 (Varying LAD Equation (15))	Case 3 (No forest)
$a_f(y)$	0.215	$z_m = 0.7h, a_{fm} = 0.38$	0.0

In reality, the LAD a_f varies with the vertical height of a tree, as the tree crown has dense leaves and the trunk is less obstructing. In the current work, we investigate three distributions of LAD, as listed in Table 2. Case 1 has constant LAD in the vertical direction, indicating that the leaf and obstructions are uniform along a tree. Case 2 has varying LAD, which is more similar to the real situation (Nebenführ & Davidson, 2014, 2015). Case 3 has no forest force imposed (i.e. constant LAD of $a_f = 0$). However, the LAD distributions of cases 1 and 2 have the same total obstruction, which is defined as $A_f = \int_0^h a_f(z)\,dz$, whereas case 3 has no obstruction. The LAD distributions of cases 1 and 2 are illustrated in Figure 6. In case 1, $a_f = 0.215$. In the varying LAD case, a_f is set on the basis of Equation (15) with the coefficients $z_m = 0.7h$ and $a_{f_m} = 0.38$.

Moreover, a sensitivity study on the forest drag coefficient C_D is performed for the varying LAD with $C_D = 0.13, 0.15$, and 0.2. The normalized time-averaged streamwise velocity as function of y/h is shown in Figure 7. All results almost overlap with each other, indicating that the model is not sensitive to C_D. Therefore, choosing $C_D = 0.15$ in the present study is robust for the following analysis.

To develop the turbulence, three small boxes are set near the inlet in the initial stage of the simulations. The non-dimensional box size is $20 \times 20 \times 20\,m^3$. Three boxes are uniformly distributed along the spanwise direction. After 15 periods of the flow passing through the computational domain when turbulence is developed,

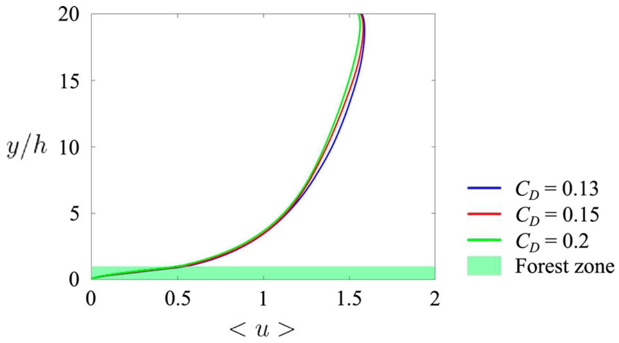


Figure 7. The normalized time-averaged streamwise velocity computed using the varying LAD with the different forest drag coefficient C_D.

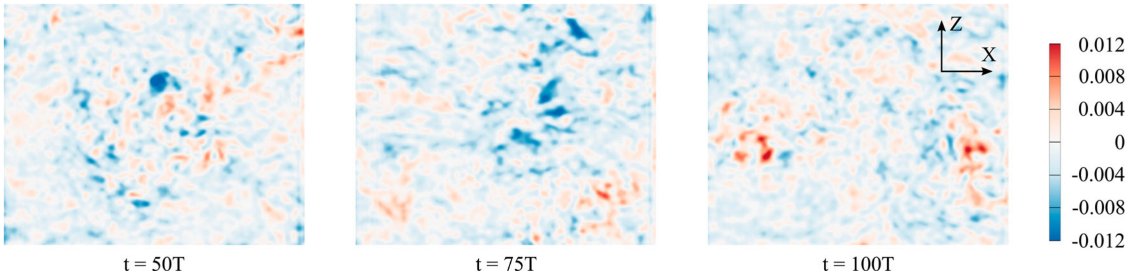
the boxes are removed from the computational domain. The simulation is then run for additional 80 periods to develop turbulence. The time-averaged statistics are obtained by averaging the data for 100 periods.

As shown in Figure 8, the normalized instantaneous velocity gradient near the upper boundary is overall less than 0.012. Here the normalization is based on the velocity of the equilibrium boundary condition and the unit meter. It indicates that the change in velocity per meter is below 1.2% relative to the given velocity, and the flow close to the upper boundary is nearly in equilibrium in both time and space.

5.2. Mesh independence study

The mesh quality for the simulation accuracy is examined with the meshes with coarse, medium and fine cell size. The constant LAD of $a_f = 0.215$ is adopted for the forest modeling. The information of the meshes are presented in Table 3. The coarse mesh has the number of cells in the three dimensions as $N_x \times N_y \times N_z = 200 \times 80 \times 160$, resulting in 2.6×10^6 cells in total. We define the refinement ratio as the ratio of a refined cell size to the reference cell size. In reference to the coarse mesh, the refinement ratios of the medium and fine meshes are 0.8 and $1/1.75 \approx 0.57$.

The computational time and memory costs relevant to the meshes are also reported in Table 3. The GPU, GeForce RTX 2070 SUPER, with a total memory of 8 GB is used for the computations. The coarse mesh simulation consumes the computing time of approximately 8.5 hours per 1×10^6 time steps and the memory of nearly 1.44 GB. For the medium mesh simulation, the computing time is 1.65 times larger than the coarse mesh, and the memory cost is 1.5 times larger. The fine mesh simulation is 4.59 times faster and 4.5 times more memory consumption. The increasing ratios of the computing time are comparable with those of the memory costs. On the other hand, the total number of cells in the medium and fine meshes increase with ratios of 1.95 and 5.36, which are larger than the ratios of the computing time and memory costs. It suggests that the increased mesh size improves the computational efficiency. The reason is that there are modules responsible for processing and storing solution data in the present simulation platform. The additional computational and memory costs from these modules are independent on the mesh size, so they are more weighted in the coarse mesh case, as compared with the other



t = 50T

t = 75T

t = 100T

Figure 8. Snapshots of the normalized instantaneous velocity gradient near the upper boundary at different time. Here T represents the time period of the flow passing through the computational domain.

Table 3. The meshes and respective computational costs in the mesh independence study.

Meshes	$N_x \times N_y \times N_z$[§]	N_{tot}[¶]	Δt[†]	T_c (hours)[‡]	Memory (GB)
Coarse	$200 \times 80 \times 160$	2.6×10^6	0.3	8.5	1.44
Medium	$250 \times 100 \times 200$	$2.6 \times 10^6 \cdot 1.95$	0.2	$8.5 \cdot 1.65$	$1.44 \cdot 1.5$
Fine	$350 \times 140 \times 280$	$2.6 \times 10^6 \cdot 5.36$	0.1	$8.5 \cdot 4.59$	$1.44 \cdot 4.5$

[§] The number of cells in the three dimensions.
[¶] The total number of cells.
[†] The non-dimensional time interval of each time step.
[‡] The computing time per 1×10^6 time steps.

refined mesh cases. Moreover, the GPU solves the LBM equations on all cells at the same time. An increased number of cells has less impact on the computing time if the former coarse mesh does not fill all capacity of the GPU. Thus, GPU is well suited for the LBM-based large scale applications, although most LBM applications carried out on GPU is memory-bounded (Harwood, Wenisch et al., 2018, June 11–15, Glasgow, UK; Santasmasas, 2021). However, new techniques have been developed to improve the memory access on GPU (Obrecht et al., 2014, 2016). Despite relatively smaller CFL numbers needed to maintain the numerical stability of LBM as compared to FVM, solving LBM has proven to be faster than solving the Navier–Stokes equations due to its intrinsic low data dependency, that is, each lattice cell can be updated locally without a need to wait for the neighbor cells' update (Krüger et al., 2017; Marié et al., 2009).

The normalized averaged streamwise velocity along the wall normal direction for the simulations of the three meshes is shown in Figure 9. The averaging is done in time and then in x–z planes that are parallel to the bottom wall. Hereafter, the averaged quantities reported in this study are obtained using the same averaging method. As compared to the experimental data from Bergström et al. (2013), the fine mesh simulation shows the best agreement. In the range of the experimental data, the coarse mesh simulation gives better prediction than the medium mesh simulation. But these two simulations become similar above $y/h = 14$, which is far from the bottom wall and forest. The profile of the coarse mesh simulation rises more slowly for $y/h > 2$. It implies that there are less turbulent vortices resolved by the coarse mesh.

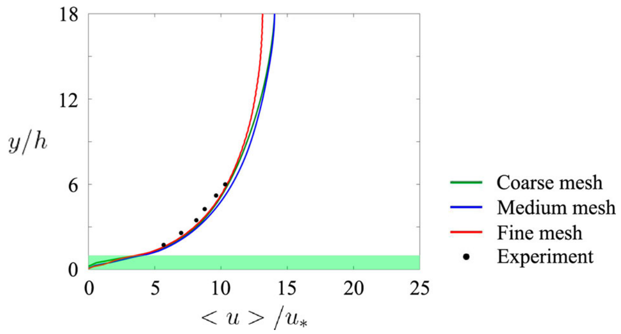


Figure 9. The averaged streamwise velocity obtained from the present simulations with the constant LAD and from the experiments by Bergström et al. (2013).

The Reynolds stress tensor components, $\langle u'u'\rangle$, $\langle v'v'\rangle$, $\langle w'w'\rangle$ and $\langle u'v'\rangle$, normalized with the averaged wall friction velocity, u_*, are shown in Figure 10. It is observed that the coarse mesh leads to the worst prediction of all components in reference to the experimental data from Bergström et al. (2013), whereas the prediction of the fine mesh is most consistent. Therefore, a general trend is that the mesh refinement improves the simulation accuracy.

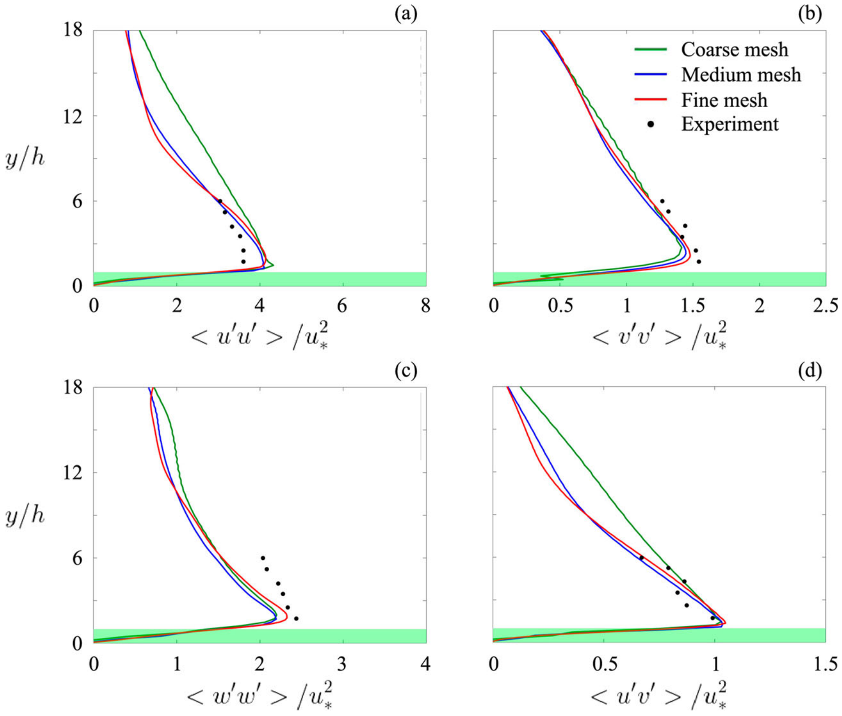


Figure 10. The normalized Reynolds stress tensor components computed with the three meshes and reported in the experiments by Bergström et al. (2013) Here the constant LAD is set in the forest modeling.

For the streamwise component $\langle u'u' \rangle / u_*^2$ in Figure 10, the slope of the profile from the coarse mesh simulation is more straight than the other meshes. This is a typical phenomenon when the turbulent flow is less developed. As a consequence, velocity fluctuations owing to turbulence vortices are insufficient to form a steeper profile. This effect is also found for the wall normal component $\langle v'v' \rangle / v_*^2$ and the spanwise component $\langle w'w' \rangle / w_*^2$, but it is not so significant as $\langle u'u' \rangle / u_*^2$. Since streamwise velocity fluctuations have larger magnitudes than spanwise and wall normal velocity fluctuations, the streamwise velocity ones are more influential in the formation of the Reynolds stress component $\langle u'v' \rangle$. As seen in subfigure 10d, the profiles of $\langle u'v' \rangle / u_*^2$ show similar trends to $\langle u'u' \rangle / u_*^2$.

The fine mesh results in overall the best prediction, as indicated in Figures 9 and 10. Based on Table 3, the averaged first-layer cell height over the wall is $\langle \Delta y+ \rangle = 165$. Moreover, the results of the medium and fine meshes are nearly the same. This converged prediction means that both medium and fine mesh resolutions are sufficient to simulate the current ABL case. According to the scalability of the computational time and memory costs with respect to the mesh size found in Table 3, the fine mesh has the best performance in computation. Since the fine mesh resolves more small turbulence scales than the medium mesh, it is chosen in the following simulations and analysis although more computational costs are required.

Based on Table 3, the averaged first-layer cell height over the wall is $\langle \Delta y+ \rangle = 165$. This infers that the wall modeling should be used in principle. However, the practice in the current simulations show that good agreement is achieved without using a wall function. The reason is that the forest forces are much larger than the wall friction. This effect will be discussed in the subsequent section on the forest modeling.

5.3. Effects of forest modeling

To investigate the effects of the forest modeling, the simulation cases with different LAD are defined in Table 2. Case 1 (the modeled forest with the constant LAD) has been discussed in the preceding section, and it is found producing consistent prediction in comparison to the experiments.

For case 2 (the modeled forest with the varying LAD) and case 3 (zero LAD, i.e. no modeled forest), the instantaneous normalized streamwise velocity u/u_* are shown in Figure 11. Case 2 exhibits generally much smaller velocity fluctuations than case 3 in the whole computational domain. Fluctuations are visualized at a x–z plane of $y/h = 0.5$, which is the center position of the modeled forest zone. Velocity magnitudes in this plane in both cases are smaller than those in the region far from the bottom wall. As case 2 includes the drag forces from the forest modeling and from the wall friction, the small fluctuations within the forest zone are explainable for this case. There are no forest drag forces imposed in case 3. In other words, case 3 does not have a forest zone. Its near-wall flow is only dragged by the wall friction. Therefore, the forest dragging effect is dominant in reducing the velocity of the ABL flow, as compared with the wall friction. It is observed in case 2 that the velocity quickly increases to larger values outside the forest zone. This also qualitatively indicates that the significant dragging effect from the modeled forest. The same finding was reported in Nebenführ and Davidson (2014, 2015).

The vertical profiles of the averaged streamwise velocity $\langle u \rangle / u_*$ in all simulated cases listed in 2, as well as the FVM LES with the same varying LAD setting as case 2 in Nebenführ and Davidson (2014, 2015) and the experiment data in Bergström et al. (2013), are plotted in Figure 12. In the near ground zone of $y/h < 6$ where large part of the boundary layer is located, cases 1 and

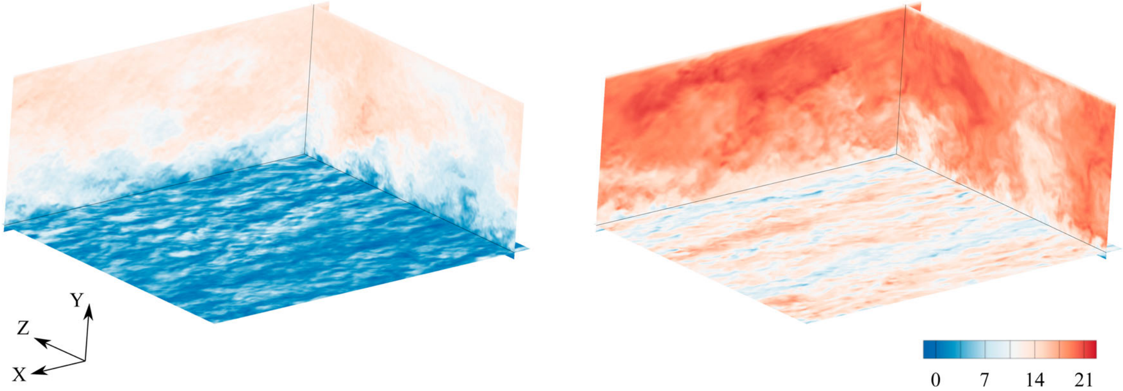


Figure 11. A snapshot of the instantaneous normalized streamwise velocity u/u_*. (a) The simulation with the forest modeling of the varying LAD; (b) the simulation with no modeled forest. The bottom surface visualized here is located at $y/h = 0.5$, which is the middle of the forest zone.

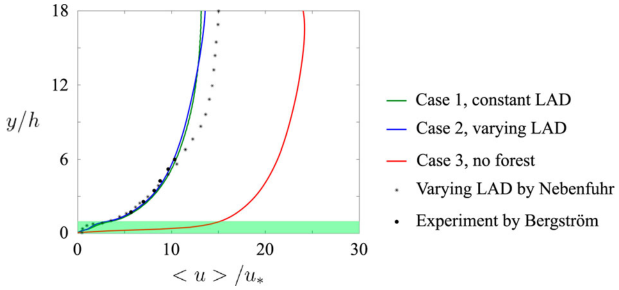


Figure 12. The averaged streamwise velocity in the present simulations with different forest modeling settings, as well as from the simulations using the FVM LES by Nebenführ and Davidson (2014, 2015) and the experiments by Bergström et al. (2013).

2 exhibit good agreement with the experimental data. Both FVM LES and LBM LES agree with the experimental data for $y/h < 5$, but the discrepancies between them are observed for larger heights. The reason is that the flow simulated with the FVM LES has a larger bulk velocity than the other one with the LBM LES. It is worth noting that the bulk velocity in the atmosphere is technically difficult to measure and, therefore, was not obtained in the experiments. The flow parameters in the simulations have been adjusted to achieve the agreement with the experimental data. As turbulent fluctuations in flows at high Reynolds numbers show universal statistics, the current flow simulation is also suitable to provide enough information about the turbulence statistics. This will be analyzed in the following discussion. For example, in Figure 13 where the two methods produce similar RMS values of velocity fluctuations. The mesh for the FVM LES contains 10 cells in the forest zone. It has been demonstrated that the mesh resolution is sufficient to resolve the profile of the varying LAD illustrated in Figure 6 (Nebenführ & Davidson, 2014, 2015). There are 7 cells in the fine mesh for the current simulations, which also resolves the same varying LAD profile. The comparison of cases 1 and 2 indicates that the constant and varying LADs lead to slight differences in the prediction of the mean velocity in the forest zone and most of the zone above the forest.

In case 3 that has no forest, the boundary layer development is completely different from cases 1 and 2 considering the forest effects. The velocity gradient in the forest zone in case 3 is much larger than the other cases. A similar phenomenon was also reported in Nebenführ and Davidson (2014, 2015). The reason is that the forest model introduces significant drag forces into the flow. Based on Equation (14), the forest drag force is independent from the wall friction. Its vertical distribution is not governed by the classical boundary layer statistics, which can be resumed using a wall function. Although the wall function is ignored, only considering the forest model is still able to predict the mean velocity profile matching the experiments. This is explainable because the forest force overwhelms wall friction.

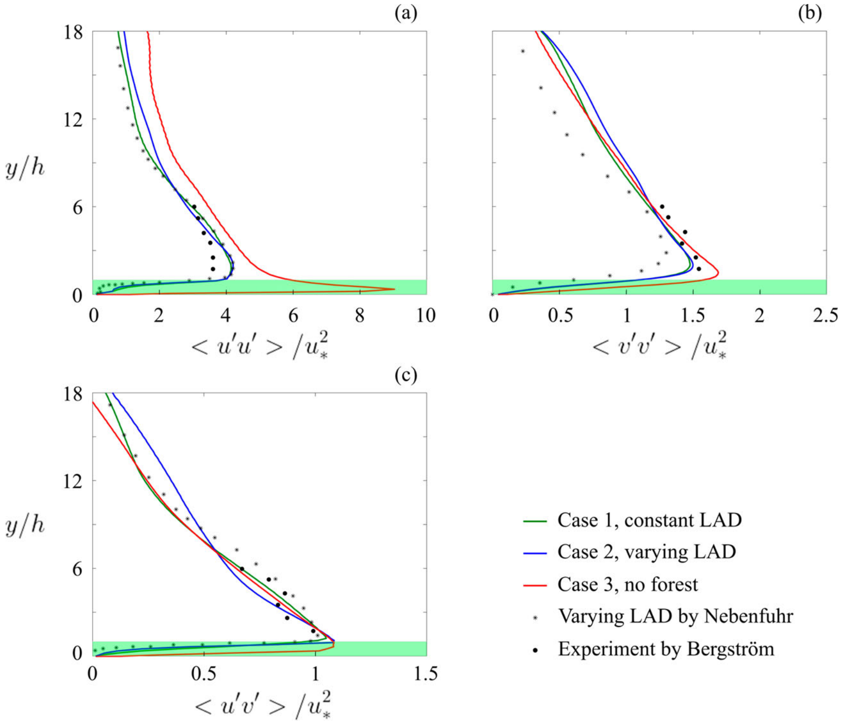


Figure 13. The stress tensor components in the present simulations with different forest modeling settings, as well as from the simulations using the FVM LES by Nebenführ and Davidson (2014, 2015) and the experiments by Bergström et al. (2013).

The vertical profiles of the normalized stress tensor components are displayed in Figure 13. In the simulations set with the forest model (cases 1 and 2, as well as the FVM LES in Nebenführ & Davidson, 2014, 2015), the predicted trends of $\langle u'u' \rangle$ are consistent. But a mismatch between these methods and the experimental data is observed in the near forest zone of $2 < y/h < 4$. Cases 1 and 2 also show $\langle v'v' \rangle$ and $\langle u'v' \rangle$ comparable with the experimental data. The different LAD settings of cases 1 and 2 lead to discrepancies between their results. But the results are approximately identical at the first cell height. Recalling Equation (14), the forest drag force modeled is proportional to the local kinetic energy. Since the velocity at the wall is zero based on the no-slip boundary condition, the local forest drag force diminishes there. The small velocity near the wall leads to the negligible force, although the constant LAD is much larger than the varying LAD near the wall (see Figure 6). As the majority difference of the two LAD settings are defined near the wall for y/h less than 0.4, their influence on the drag force production is limited owing to the small local velocity. As a consequence, the profiles of the tensor statistics simulated with the two LAD settings are similar. Nevertheless, the present study only demonstrates the forest effect modeled with the LADs shown Figure 6. If the LAD definition is changed, the forest effects could change and, thus, should be thoroughly examined.

As the forest drag force is not modeled, case 3 gives different prediction in comparison with cases 1 and 2, in particular, for $\langle u'u' \rangle / u_*^2$ displayed in Figure 13. Obvious differences are identified for all stress tensor components within the forest zone, and $\langle u'u' \rangle / u_*^2$ also shows inconsistency at vertical locations far from the forest zone. Therefore, the mismatch conveys that the modeled forest drag force play a critical role in predicting the statistics of flow fluctuations in the boundary layer, in addition to its importance for the mean flow statistics that is shown in Figure 12. As a summary, from Figures 12 and 13, we can see that both constant and vary LAD agrees well with the experiment data at the height of wind turbine. This implies the possibility to extend the current model with wind turbines models. One typical model for wind turbines is the actuator disk model(ADM) (Abkar & Porté-Agel, 2016; Strickland et al., 2022; Yang & Sotiropoulos, 2013); it simplifies the wind turbine as a disk to reduce the computational cost. The other existing model is named as actuator line model(ALM) (Ravensbergen et al., 2020). This model has more detailed information for the wind turbines and can perform results with higher accuracy. Interest readers can refer to Stevens et al. (2018), who compared the ALM and ADM to show the difference between these two models and the wind tunnel experiment. With the models mentioned above added to our present study, it would be interesting to study the possible placement or optimization of wind turbine distribution in a wind farm in the presence of forest terrain.

6. Conclusions

A GPU-based simulation platform using an LBM method for LES has been proven to be suitable for computing ABL flows with the forest presence, which are of interest in the site selection of wind turbine farms. The forest effects are modeled by defining a volume drag force in relation to the local velocity and LAD. The way of implementing the force for LBM is presented in detail. However, to refrain from influencing the GPU computational efficiency, the meshes are not locally refined in the method developed in this study. Uniform-sized lattices are generated in the whole computational domain. This leads to large $\Delta y+$ near the wall, which in principle requires a wall function. We introduce the approximations near the ground wall that the wall function is not taken into account, and that the forest effects are considered.

The near-wall approximations are made because the forest drag force overwhelms the wall friction. This is validated based on the qualitative and quantitative analysis of the results. The statistics of the mean flow velocity and stress tensor components are well consistent with the previous FVM-LES simulations and experiments. The mesh independence study illustrates that the modeled forest zone is resolved with 7 cells.

One advantage of the near-wall approximations is that the LBM program is compatible with the GPU architecture, so the GPU speedup is preserved. Moreover, the wall treatment is ignored to reduce the computational cost. The platform extends the capabilities of the open-source simulation tool GASCANS developed at the University of Manchester.

The forest effects are examined for a classic vertically varying LAD distribution, which was simulated in previous studies using the FVM-LES, and a constant distribution possessing the same total area as the varying one. The results between these two LADs differ only slightly. Nevertheless, it would be interesting to investigate the effects of the LAD vertical profile on the characteristics of ABL flows such as time-space correlations (Yao et al., 2008). In the present study, the LAD model is investigated with specific parameters such as a constant height, while the height can be changed in the reality. An interesting future work is thus to explore the range of the parameters that are suitable for the modeling.

Acknowledgments

The authors would like to acknowledge the funding from Chalmers Area of Advance Transport. The computations and data handling were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC).

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was partially funded by the Swedish Research Council through [grant agreement number 2018-05973].

ORCID

Xiao Xue http://orcid.org/0000-0002-1193-463X
Jiqiang Niu http://orcid.org/0000-0002-9180-9571
Lars Davidson http://orcid.org/0000-0003-2903-4004
Hua-Dong Yao http://orcid.org/0000-0002-9547-7877

References

Abkar, M., & Porté-Agel, F. (2016). Influence of the Coriolis force on the structure and evolution of wind turbine wakes. *Physical Review Fluids*, *1*(6), 063701. https://doi.org/10.1103/PhysRevFluids.1.063701.

Belardinelli, D., Sbragaglia, M., Biferale, L., Gross, M., & Varnik, F. (2015). Fluctuating multicomponent lattice Boltzmann model. *Physical Review E*, *91(2)*, 023313. https://doi.org/10.1103/PhysRevE.91.023313

Bergström, H., Alfredsson, H., Arnqvist, J., Carlén, I., Dellwik, E., Fransson, J., & Söderberg, S (2013). *Wind power in forests: Wind and effects on loads*. Elforsk AB.

Dipankar, A., Stevens, B., Heinze, R., Moseley, C., Zängl, G., Giorgetta, M., & Brdar, S. (2015). Large eddy simulation using the general circulation model ICON. *Journal of Advances in Modeling Earth Systems*, *7*(3), 963–986. https://doi.org/10.1002/jame.v7.3

Dupont, S., & Brunet, Y. (2008a). Edge flow and canopy structure: A large-eddy simulation study. *Boundary-Layer Meteorology*, *126*(1), 51–71. https://doi.org/10.1007/s10546-007-9216-3

Dupont, S., & Brunet, Y. (2008b, May). Influence of foliar density profile on canopy flow: A large-eddy simulation study. *Agricultural and Forest Meteorology*, *148*(6–7), 976–990. https://doi.org/10.1016/j.agrformet.2008.01.014

Dupont, S., & Brunet, Y. (2009). Coherent structures in canopy edge flow: A large-eddy simulation study. *Journal of Fluid Mechanics*, *630*, 93–128. https://doi.org/10.1017/S0022112009006739

Feng, Y., Miranda-Fuentes, J., Guo, S., Jacob, J., & Sagaut, P. (2021). ProLB: A lattice Boltzmann solver of large-eddy simulation for atmospheric boundary layer flows. *Journal of Advances in Modeling Earth Systems*, *13*(3), e2020MS002107. https://doi.org/10.1029/2020MS002107

Filippova, O., Succi, S., Mazzocco, F., Arrighetti, C., Bella, G., & Hänel, D. (2001 Jul). Multiscale lattice Boltzmann schemes with turbulence modeling. *Journal of Computational Physics*, *170*(2), 812–829. https://linkinghub.elsevier.com/retrieve/pii/S0021999101967646.doi:10.1006/jcph.2001.6764.

Finnigan, J. J., Shaw, R. H., & Patton, E. G. (2009). Turbulence structure above a vegetation canopy. *Journal of Fluid Mechanics*, *637*, 387–424. https://doi.org/10.1017/S0022112009990589

Guo, Z., Zheng, C., & Shi, B. (2002). Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Physical Review E*, *65*(4), 046308. https://doi.org/10.1103/PhysRevE.65.046308

Harwood, A. R., O'Connor, J., Munoz, J. S., Santasmasas, M. C., & Revell, A. J. (2018). LUMA: A many-core, fluid–structure interaction solver based on the lattice-Boltzmann method. *SoftwareX*, *7*, 88–94. https://doi.org/10.1016/j.softx.2018.02.004

Harwood, A. R., Wenisch, P., & Revell, A (2018, June 11–15, Glasgow, UK). A real-time modelling and simulation platform for virtual engineering design and analysis. Proceedings of 6th European Conference on Computational Mechanics (ECCM 6) and 7th European Conference on Computational Fluid Dynamics (ECFD 7), p. 504.

Hou, S., Sterling, J., Chen, S., & Doolen, G (1994). A lattice Boltzmann subgrid model for high Reynolds number flows. ArXiv preprint comp-gas/9401004.

Koda, Y., & Lien, F. S. (2014). The lattice Boltzmann method implemented on the GPU to simulate the turbulent flow over a square cylinder confined in a channel. *Flow, Turbulence and Combustion*, *94*(3), 495–512. doi:10.1007/s10494-014-9584-y.

Krause, M. J., Kummerländer, A., Avis, S. J., Kusumaatmaja, H., Dapelo, D., Klemens, F., Gaedtke, M., Hafen, N., Mink, A., Trunk, R., Marquardt, J. E., Maier, M.-L., Haussmann, M., & Simonis, S. (2021). OpenLB – Open source lattice Boltzmann code. *Computers & Mathematics with Applications*, *81*, 258–288. https://doi.org/10.1016/j.camwa.2020.04.033

Krüger, T., Kusumaatmaja, H., Kuzmin, A., Shardt, O., Silva, G., & E. M. Viggen (2017). *The lattice Boltzmann method*. Springer International Publishing. https://doi.org/10.1007/978-3-319-44649-3.

Lalic, B., & Mihailovic, D. T. (2004). An empirical relation describing leaf-area density inside the forest for environmental modeling. *Journal of Applied Meteorology*, *43*(4), 641–645. https://doi.org/10.1175/1520-0450(2004)043¡0641:AERDLD¿2.0.CO;2

Latt, J., Chopard, B., Malaspinas, O., Deville, M., & Michler, A. (2008). Straight velocity boundaries in the lattice Boltzmann method. *Physical Review E*, *77*(5), 056703. https://doi.org/10.1103/PhysRevE.77.056703

Latt, J., Coreixas, C., & Beny, J. (2021 Oct). Cross-platform programming model for many-core lattice Boltzmann simulations. *PLoS One*, *16*, 4e0250306.http://arxiv.org/abs/2010.11751doi:10.1371/journal.pone.0250306.

Latt, J., Malaspinas, O., Kontaxakis, D., Parmigiani, A., Lagrava, D., Brogi, F., Belgacem, M. B., Thorimbert, Y., Leclaire, S., Li, S., Marson, F., Lemus, J., Kotsalos, C., Conradin, R., Coreixas, C., Petkantchin, R., Raynaud, F., Beny, J., & Chopard, B. (2021). Palabos: Parallel lattice Boltzmann solver. *Computers & Mathematics with Applications*, *81*, 334–350. https://doi.org/10.1016/j.camwa.2020.03.022

Lo, A. K. (1990). On the determination of zero-plane displacement and roughness length for flow over forest canopies. *Boundary-layer Meteorology*, *51*(3), 255–268. https://doi.org/10.1007/BF00122140

Marié, S., Ricot, D., & Sagaut, P. (2009). Comparison between lattice Boltzmann method and Navier–Stokes high order schemes for computational aeroacoustics. *Journal of Computational Physics*, *228*(4), 1056–1070. https://doi.org/10.1016/j.jcp.2008.10.021

Maronga, B., Gryschka, M., Heinze, R., Hoffmann, F., Kanani-Sühring, F., Keck, M., Ketelsen, K., Letzel, M. O., Sühring, M., & Raasch, S. (2015). The parallelized large-eddy simulation model (PALM) version 4.0 for atmospheric and oceanic flows: Model formulation, recent developments, and future perspectives. *Geoscientific Model Development*, *8*(8), 2515–2551. https://doi.org/10.5194/gmd-8-2515-2015

Moser, R. D., Kim, J., & Mansour, N. N. (1999). Direct numerical simulation of turbulent channel flow up to $Re_\tau = 590$. *Physics of Fluids*, *11*(4), 943–945. doi:10.1063/1.869966

Nebenführ, B., & Davidson, L (2014). *Influence of a forest canopy on the neutral atmospheric boudary layer-A LES study*. TMM10: 10th International Ercoftac Symposium on Turbulence Modelling and Measurements.

Nebenführ, B., & Davidson, L. (2015). Large-eddy simulation study of thermally stratified canopy flow. *Boundary-layer Meteorology*, *156*(2), 253–276. https://doi.org/10.1007/s10546-015-0025-9

Nebenführ, B., & Davidson, L. (2017). Prediction of wind-turbine fatigue loads in forest regions based on turbulent LES inflow fields. *Wind Energy*, *20*(6), 1003–1015. https://doi.org/10.1002/we.v20.6

Nezu, I., & Sanjou, M. (2008). Turburence structure and coherent motion in vegetated canopy open-channel flows. *Journal of Hydro-Environment Research*, *2*(2), 62–90. https://doi.org/10.1016/j.jher.2008.05.003

Obrecht, C., Asinari, P., Kuznik, F., & Roux, J. J. (2014). High-performance implementations and large-scale validation of the link-wise artificial compressibility method. *Journal of Computational Physics*, *275*, 143–153. https://doi.org/10.1016/j.jcp.2014.06.062

Obrecht, C., Asinari, P., Kuznik, F., & Roux, J. J. (2016). Thermal link-wise artificial compressibility method: GPU implementation and validation of a double-population model. *Computers & Mathematics with Applications*, *72*(2), 375–385. https://doi.org/10.1016/j.camwa.2015.05.022

Ravensbergen, M., Mohamed, A. B., & Korobenko, A. (2020). The actuator line method for wind turbine modelling applied in a variational multiscale framework. *Computers & Fluids*, *201*, 104465. https://doi.org/10.1016/j.compfluid.2020.104465

Santasmasas, M. C. (2021). *Hybrid gpu/cpu navier-stokes lattice boltzmann method for urban wind flow*. The University of Manchester (United Kingdom).

Shaw, R. H., & Patton, E. G. (2003). Canopy element influences on resolved-and subgrid-scale energy within a large-eddy simulation. *Agricultural and Forest Meteorology*, *115*(1–2), 5–17. https://doi.org/10.1016/S0168-1923(02)00165-X

Shaw, R. H., & Schumann, U. (1992). Large-eddy simulation of turbulent flow above and within a forest. *Boundary-Layer Meteorology*, *61*(1), 47–64. https://doi.org/10.1007/BF02033994

Stevens, R. J., Gayme, D. F., & Meneveau, C. (2014). Large eddy simulation studies of the effects of alignment and wind farm length. *Journal of Renewable and Sustainable Energy*, *6*(2), 023105. https://doi.org/10.1063/1.4869568

Stevens, R. J., Gayme, D. F., & Meneveau, C. (2015). Coupled wake boundary layer model of wind-farms. *Journal of Renewable and Sustainable Energy*, *7*(2), 023115. https://doi.org/10.1063/1.4915287

Stevens, R. J., Martínez-Tossas, L. A., & Meneveau, C. (2018). Comparison of wind farm large eddy simulations using actuator disk and actuator line models with wind tunnel experiments. *Renewable Energy*, *116*, 470–478. https://doi.org/10.1016/j.renene.2017.08.072

Stevens, R. J., & Meneveau, C. (2017). Flow structure and turbulence in wind farms. *Annual Review of Fluid Mechanics*, *49*(1), 311–339. https://doi.org/10.1146/fluid.2017.49.issue-1

Strickland, J. M., Gadde, S. N., & Stevens, R. J. (2022). Wind farm blockage in a stable atmospheric boundary layer. *Renewable Energy*, *197*, 50–58. https://doi.org/10.1016/j.renene.2022.07.108

Succi, S. (2001). *The lattice boltzmann equation: For fluid dynamics and beyond*. Oxford University Press.

Toschi, F., & Bodenschatz, E. (2009). Lagrangian properties of particles in turbulence. *Annual Review of Fluid Mechanics*, *41*, 375–404. https://doi.org/10.1146/annurev.fluid.010908.165210

Van Heerwaarden, C. C., Van Stratum, B. J., Heus, T., Gibbs, J. A., Fedorovich, E., & Mellado, J. P. (2017). MicroHH 1.0: A computational fluid dynamics code for direct numerical simulation and large-eddy simulation of atmospheric boundary layer flows. *Geoscientific Model Development*, *10*(8), 3145–3165. https://doi.org/10.5194/gmd-10-3145-2017

Xue, X., Biferale, L., Sbragaglia, M., & Toschi, F. (2020). A lattice Boltzmann study on Brownian diffusion and friction of a particle in a confined multicomponent fluid. *Journal of Computational Science*, *47*, 101113. https://doi.org/10.1016/j.jocs.2020.101113

Xue, X., Sbragaglia, M., Biferale, L., & Toschi, F. (2018). Effects of thermal fluctuations in the fragmentation of a nanoligament. *Physical Review E*, *98*(1), 012802. https://doi.org/10.1103/PhysRevE.98.012802

Xue, X., Yao, H. D., & Davidson, L. (2022). Synthetic turbulence generator for lattice Boltzmann method at the interface between RANS and LES. *Physics of Fluids*, *34*(5), 055118. https://doi.org/10.1063/5.0090641

Yang, X., & Sotiropoulos, F (2013). On the predictive capabilities of LES-actuator disk model in simulating turbulence past wind turbines and farms. In *2013 American Control Conference* (pp. 2878–2883).

Yao, H. D., He, G. W., Wang, M., & Zhang, X. (2008). Time correlations of pressure in isotropic turbulence. *Physics of Fluids*, *20(2)*, 025105. https://doi.org/10.1063/1.2870111.