Computers and Mathematics with Applications 138 (2023) 65–87

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Lattice Boltzmann method–based efficient GPU simulator for vapor transport in the boundary layer over a moist soil: Development and experimental validation

Jakub Klinkovský [a,*,1], Andrew C. Trautz [b,c], Radek Fučík [a], Tissa H. Illangasekare [c]

[a] Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Trojanova 13, 120 00 Prague, Czech Republic
[b] Research Civil Engineer – Geotechnical Engineering and Geosciences Branch, US Army Engineer Research and Development Center, United States of America
[c] Center for Experimental Study of Subsurface Environmental Processes (CESEP), Department of Civil and Environmental Engineering, Colorado School of Mines, United States of America

ARTICLE INFO

Keywords:
Lattice Boltzmann method
Mixed-hybrid finite element method
Multi-GPU implementation
Evaporation
Wind tunnel

ABSTRACT

We present an efficient computational approach for simulating component transport within single-phase free flow in the boundary layer over porous media. A numerical model based on this approach is validated using experimental data generated in a climate-controlled wind tunnel coupled with a soil test bed. The developed modeling approach is based on a combination of the lattice Boltzmann method (LBM) for simulating the fluid flow and the mixed-hybrid finite element method (MHFEM) for solving constituent transport. Both those methods individually, as well as when coupled, are implemented entirely on a GPU accelerator in order to utilize its computational power and avoid the hardware limitations caused by slow communication between the GPU and CPU over the PCI-E bus. In order to utilize vast computational resources available on modern supercomputers, the implementation is extended for distributed multi-GPU computations based on domain decomposition and the Message Passing Interface (MPI). We describe the mathematical details behind the computational method, focusing primarily on the coupling mechanisms. The performance of the solver is demonstrated on a modern high-performance computing system. Flow and transport simulation results are validated and compared herein with experimental velocity and relative humidity measurements made above a flat partially saturated soil layer exposed to steady air flow. Model robustness and flexibility is demonstrated by introducing cuboidal bluff-bodies to the flow in several different experimental scenarios. The experimentally measured values are available in a publicly available dataset that can serve as a benchmark for future studies. Finally, we discuss potential improvements for the model as well as future experimental efforts.

1. Introduction

Numerous proprietary or open-source computational tools are available for solving partial differential equations originating from mathematical modeling of various biological, environmental, or industrial problems. In particular, computational software such as deal.II [1], DUNE [2], OpenFOAM [3], TOUGH2 [4], MFiX [5], ANSYS Fluent [6] or COMSOL Multiphysics [7] is suitable for complex multi-physics simulations involving multiphase or compositional flows. However, each software has limitations: the underlying numerical methods may restrict the applicability of the software; the approach for code execution may cause limited or no advantage of using high-performance architectures, in particular graphical processing units (GPUs), for the acceleration of computations; and the software design in general might make it difficult to combine different tools for solving coupled problems. Furthermore, extending large software packages such as the aforementioned ones with novel mathematical approaches and numerical methods is challenging and unfeasible for most external users.

We have developed a novel computational approach based on a combination of the lattice Boltzmann method and the mixed-hybrid finite element method for simulating component transport within single-phase free flow. The lattice Boltzmann method (LBM) [8] is a modern and efficient numerical method capable of simulating numerous problems in computational fluid dynamics (CFD), including turbulent fluid

* Corresponding author.
E-mail address: jakub.klinkovsky@fjfi.cvut.cz (J. Klinkovský).
[1] *Address*: Katedra matematiky, FJFI CVUT v Praze, Trojanova 13, 120 00 Praha 2, Czech Republic.

https://doi.org/10.1016/j.camwa.2023.02.021
Received 22 June 2022; Received in revised form 7 December 2022; Accepted 22 February 2023

flows. The mixed-hybrid finite element method (MHFEM) [9] is a well established general numerical method for solving partial differential equations. In this work, we use the MHFEM formulation presented in [10] and extend it with the coupling to LBM. There are multiple reasons why we pursue this approach:

- Both LBM and MHFEM individually have advantages compared to traditional numerical methods. LBM is based on the mesoscopic description of the fluid and the computational algorithm avoids the solution of Poisson equation for pressure [8], which is the most time consuming part of algorithms based on finite difference methods (FDM) or finite volume methods (FVM). On the other hand, MHFEM is a general method that combines different finite element approximations of scalar as well as vector functions. For some problems, it may provide higher accuracy and robustness compared to standard FEM or FVM approaches [11,12].
- Both LBM and MHFEM individually can be efficiently parallelized and implemented for modern high-performance architectures, including GPU accelerators. All computations in the coupled LBM-MHFEM solver can be executed entirely on a GPU accelerator in order to utilize its computational power and avoid the hardware limitations caused by slow communication between the GPU and CPU over the PCI-E bus.
- The coupled solver can utilize vast computational resources available on typical supercomputers by decomposing the domain and dividing the computation between multiple workers (GPUs) which communicate over the Message Passing Interface (MPI) [13].

Verification of the developed approach against exact solutions is not possible, since there are no known exact analytical solutions for a coupled system of Navier–Stokes and transport equations. We therefore turn to controlled experimental data for model validation purposes, specifically posing this effort in terms of the problem of bare-soil evaporation, a key component of Earth's hydrologic, carbon, and energy balances [14]. In this context, airflow has been shown to be a strong forcing parameter responsible for driving the process evaporation from bare soils or any porous medium [15,16]. Given the complexity of near-surface airflow, evaporation estimates in practice rely on simplified flow parameterizations or sparse wind measurements. We pose that the development of the combined LBM-MHFEM flow and transport model herein could support these issues by providing a way to simulate near-surface flow fields and subsequent above-ground vapor transport with a high level of fidelity, in turn improving evaporation estimates [17].

The data used for validation in this paper include velocity and relative humidity profiles measured in a low speed, climate controlled wind tunnel where 3D turbulent flow above a soil surface with cuboidal bluff bodies were investigated in [18]. The wind tunnel used for these measurements was designed specifically for the investigation of coupled soil–plant–atmosphere processes as soil-test beds containing any desired soil(s) and vegetation can be interfaced at a sufficiently large scale along a 7.4 m long test-section.

The focus of this paper is the presentation of this coupled numerical scheme as well as its validation using experimental data obtained in controlled environments and simplified flow geometry. While we present this investigation in terms of a bare-soil evaporation problem, it is important to note that with relatively minor modifications this numerical solver could be expanded to support a wide range of applications currently being investigated by the authors, including: heat transfer in the context of disturbed environments, training AI/ML super-resolution algorithms, dust forecasting and mitigation efforts, and greenhouse gas loading. More importantly, the work presented herein should be viewed as the initial steps toward the development of a flexible numerical solver that can be used to support fundamental process understanding and exploration, guide sensitivity analysis for measurements in regions with high uncertainty, inform sampling in future experimental effort or help fill gaps in datasets where it was not possible to make measurements.

The paper is organized as follows: first a general description of the experiments, mathematical model and boundary conditions is given, followed by details of the coupled LBM-MHFEM computational approach, and finally, a validation of our model is presented. The model is compared both qualitatively and quantitatively to experimental data measured in three configurations resulting in different flow regimes.

2. Problem formulation

In this section, the background for numerical simulations is introduced, namely the experimental facility and procedures are described, and the mathematical model is formulated.

2.1. Experimental setup and methodology

All experimental data used in this paper were generated in the Center for Experimental Study of Subsurface Environmental Processes (CESEP) wind tunnel–porous media test facility now located at the US Army Engineer Research and Development Center (ERDC) Synthetic Environment for Near-Surface Sensing and Experimentation (SENSE) Research Facility. The facility is centered around a closed-circuit, climate-controlled, low-speed wind tunnel that can be interfaced with soil test-beds of varying size. The test facility was designed specifically for the investigation of coupled soil–plant–atmosphere processes, including air flow and heat and mass transport at a 1:1 scale; the wind tunnel meets similarity criteria and is therefore suitable for momentum scaling studies as well. A brief description of the components relevant to this paper is presented below for the convenience of the reader. Details concerning the test facility can be found in [19,16].

The wind tunnel was interfaced with a 7.15 m long and 0.3 m wide soil test-bed along the centerline of its 7.4 m long, 1 m wide, and 1 m tall test-section. The experimental datasets of [18] were chosen for model validation due to their use of synthetic plants (i.e., porous limestone blocks) instead of living vegetation. This approach significantly simplified the airflow component of the problem (i.e., the plant can be treated as a traditional bluff body) while still retaining key hydrodynamic characteristics. The dimensions of each block were nominally $(3.15 \times 3.15 \times 29.5) \pm 0.1$ cm and they were planted in a vertical position (10.0 ± 0.3) cm deep, leaving the 19.5 cm high part above ground.

A total of three experimental configurations featuring two synthetic plants were explored by [18]. Each configuration was characterized in terms of the spacing between the two synthetic plants: 15 cm spacing (EX-1), 45 cm spacing (EX-2), and 105 cm spacing (EX-3). The positions of the synthetic plants in these configurations are given by:

- EX-1 (15 cm spacing): $x_{\mathrm{I}} = -5.197$ m, $x_{\mathrm{II}} = -5.047$ m,
- EX-2 (45 cm spacing): $x_{\mathrm{I}} = -5.647$ m, $x_{\mathrm{II}} = -5.197$ m,
- EX-3 (105 cm spacing): $x_{\mathrm{I}} = -6.247$ m, $x_{\mathrm{II}} = -5.197$ m.

Note that in the coordinate system used hereafter, $x = 0$ m corresponds to the upstream entrance of the test-section of the wind tunnel (right hand side), $x = -7.4$ m corresponds to the downstream exit of the test section (left hand side), $z = 0$ m corresponds to the ground surface and $y = 0$ m corresponds to the centerline of the test-section. At the time that the datasets associated with [18] were generated, the authors did not have access to a LiDAR system that could scan the exact position of the bluff bodies. Uncertainty of the placement of the bluff bodies is thus given as $\delta x_{\mathrm{I}} = \delta x_{\mathrm{II}} = \delta y = \pm 5$ mm. The maximum deviatory angle of attack of the bluff bodies relative to the direction of the flow is similarly given as 5°.

During each experiment, the mean wind speed, air temperature, and relative humidity of the air entering the test-section was controlled and continuously monitored. Given the coupling of the test-section with the partially saturated soil test-bed, a variable temperature and vapor

Table 1
Experimental climate conditions.

		EX-1	EX-2	EX-3
Mean Air Temperature	[°C]	25.83 ± 1.52	25.70 ± 1.06	23.98 ± 0.21
Mean Surface Temperature	[°C]	22.10 ± 0.23	23.50 ± 0.80	22.07 ± 0.03
Mean Soil Temperature	[°C]	20.72 ± 0.31	22.40 ± 0.18	20.92 ± 0.11

gradient was observed above the soil tank. The soil temperature was controlled by varying the exterior temperature of the soil test-bed and the soil moisture was hydrostatically distributed and allowed to freely evaporate. Table 1 provides a summary of the average soil, surface and air temperatures in the experiments. Variability around the mean value was caused by the cycling of the individual climate control systems (i.e., heater, chiller, humidifier, dehumidifier) throughout the duration of the experiments. The exterior temperature of the soil test-bed and barometric pressure were also measured through out the experiments; the reader is referred to [18,19] for more details.

The measurements of [19,18] are available as a public dataset [20]. Note that only a subset of these measurements is relevant for the purpose of this paper, namely the airflow properties (velocity, RMS, Reynolds stress) and relative humidity above the soil surface. Given the size of the domain and the experimental setup, the number of airflow and relative humidity measurements were necessarily constrained. The locations of airflow and relative humidity were varied between the three configurations based on the spacing distance between the synthetic plants and the resulting flow regime created. The measurement locations are highlighted in the figures in the last section of this paper. The laser used to make the flow measurements was mounted on an automated traverse located outside the wind tunnel test-section. Uncertainty in the exact location where the measurements were made can be given as $\delta x = \delta y = \pm 5$ mm and $\delta z = \pm 1$ mm. The sensor used to measure relative humidity was similarly mounted on an automated traverse located within the test-section; uncertainty associated with this system is given as $\delta x = \pm 10$ mm and $\delta y = \delta z = \pm 5$ mm.

Air flow statistics (i.e., velocity, turbulence intensity, Reynolds stress) above the soil surface were measured using two-dimensional laser Doppler velocimetry [19], providing high frequency data with an accuracy of 5%. Relative humidity was measured with an accuracy of ± 0.03 using a relative humidity–temperature (RHT) sensor constructed by the University Corporation of Atmospheric Research. Compared with the laser, these RHT sensors have a significantly lower sampling rate ($\approx$1 Hz); data collected over a 30 second window were averaged at each measurement location [19].

Furthermore, water loss from the wetted surface of the synthetic plants was measured in a separate small scale experiment [19]. These data were used to calculate an average mass flux of $\Phi_\star = 0.128$ g cm^{-2} d^{-1}; this value was assumed to be applicable to all three configurations given similarity in applied climate conditions.

2.2. Mathematical model

The air flow in the free space above the soil surface is governed by the Navier–Stokes equations. As the model targets low Mach number situations (Ma ≈ 0.003 in the wind tunnel), the fluid is considered to be incompressible [21,22]. The momentum and mass conservation equations for the air are written as

$$\nabla \cdot \boldsymbol{v} = 0, \tag{1a}$$

$$\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla \boldsymbol{v} = -\frac{1}{\rho} \nabla p + \nu \Delta \boldsymbol{v}, \tag{1b}$$

where $\boldsymbol{v} = [v_x, v_y, v_z]$ [m s^{-1}] is the air velocity, p [Pa] is the pressure, ρ [kg m^{-3}] is the air density, and ν [m^2 s^{-1}] is the kinematic viscosity of the air. In general, these quantities are given as functions of spatial coordinates $\boldsymbol{x} = [x, y, z] \in \Omega_1 \subset \mathbb{R}^3$ and time $t \in (0, t_{\max})$.

Table 2
Model parameters for air under standard atmospheric conditions (25 °C and pressure of 1 bar).

Density ρ [24]	1.184 kg m^{-3}
Kinematic viscosity ν [24]	15.52×10^{-6} m^2 s^{-1}
Molecular diffusivity D of water vapor in air [25]	25.52×10^{-6} m^2 s^{-1}

The mass conservation law for a component α within a gas mixture in $\Omega_2 \times (0, t_{\max})$ can be derived [23] and written as

$$\frac{\partial \rho_\alpha}{\partial t} + \nabla \cdot \left(\rho_\alpha \boldsymbol{v} + \boldsymbol{J}_\alpha \right) = 0, \tag{2}$$

where ρ_α [kg m^{-3}] is the density of the component α and $\boldsymbol{J}_\alpha$ [kg m^{-2} s^{-1}] is the diffusive flux. As in Eq. (1a), no sources/sinks are considered in Eq. (2). The diffusive flux is given by the Fick's law [23] as

$$\boldsymbol{J}_\alpha = -\rho D \nabla \omega_\alpha, \tag{3}$$

where D [m^2 s^{-1}] is the molecular diffusivity coefficient and $\omega_\alpha = \rho_\alpha / \rho$ [-] is the mass fraction of the component α in the mixture.

In this work, a single component $\alpha \equiv H_2O$ representing water vapor dispersed in air is considered. The component density ρ_α corresponds to the absolute humidity (i.e., the mass of the water vapor per unit volume) and the mass fraction ω_α corresponds to the specific humidity. The relative humidity ϕ [-] is defined as

$$\phi = \frac{p_\alpha}{p_\alpha^*}, \tag{4}$$

where p_α [Pa] is the partial pressure of water vapor in the mixture and p_α^* [Pa] is the equilibrium vapor pressure of water over a flat surface of pure water at a given temperature. Assuming that the mixture behaves as an ideal gas at constant temperature T [K], the ideal gas law $p_\alpha = \rho_\alpha R_\alpha T$ with the specific gas constant R_α [J kg^{-1} K^{-1}] means that the partial pressure p_α is proportional to the absolute humidity ρ_α and thus the relative humidity equals

$$\phi = \frac{\rho_\alpha}{\rho_\alpha^*}, \tag{5}$$

where ρ_α^* [kg m^{-3}] is the saturated absolute humidity corresponding to p_α^*. Using $\rho_\alpha = \phi \rho_\alpha^*$, Eq. (3) and the assumption of constant density ρ, Eq. (2) transforms to the conservative transport equation

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \boldsymbol{v} - D \nabla \phi) = 0. \tag{6a}$$

Combining Eqs. (1a) and (6a) leads to the non-conservative form

$$\frac{\partial \phi}{\partial t} + \boldsymbol{v} \cdot \nabla \phi - \nabla \cdot (D \nabla \phi) = 0. \tag{6b}$$

Although a slightly variable temperature distribution above the soil tank was observed during the experiments [18], we assume its impact on the density, kinematic viscosity, molecular diffusivity, and relative humidity to be negligible compared to the sensor accuracies. The isothermal model given by Eqs. (1) and (6) is used with constant parameters ρ, ν, and D. Furthermore, the fluid density ρ is assumed to be independent of the relative humidity ϕ and the effect of gravity is neglected due to the dimensions of the experimental facility. Model parameters for air under standard atmospheric conditions are given in Table 2.

For the purpose of this paper, we are interested in simulating only the free flow region, where the soil-atmosphere and synthetic plant

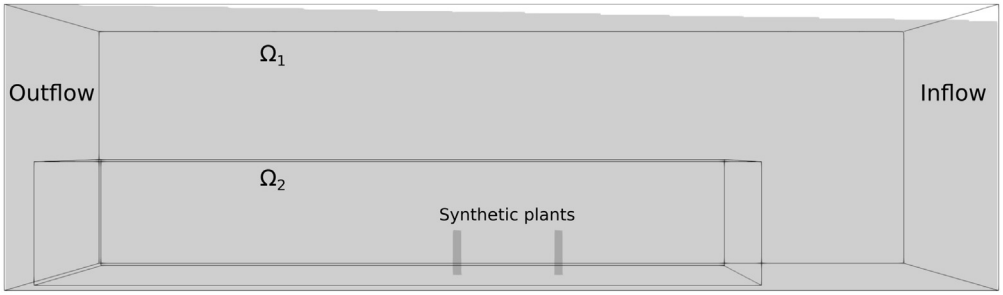


Fig. 1. Illustration of the computational domains Ω_1 and Ω_2 for Eqs. (1) and (6). Note that Ω_1 conforms to the actual dimensions (width and height) along a 3.89 m length of the wind tunnel test-section where experimental measurements were made.

(bluff body)–atmosphere interfaces are treated using boundary conditions (described in Section 2.3). Hence, the computational domains Ω_1 and Ω_2 are considered as shown in Fig. 1. Based on the experimental setup described in Section 2.1, the computational domain Ω_1 is defined as an inset of the whole test-section starting at a downstream distance of -3.507 m from the test-section inlet. The total dimensions of Ω_1 are approximately $3.89\,\text{m} \times 1\,\text{m} \times 1.13\,\text{m}$. The upper side of the domain Ω_1 coincides with the inclined ceiling of the wind tunnel; the back and front sides coincide with the test-section walls. The bottom boundaries of both domains Ω_1 and Ω_2 coincide with the soil surface in which the two synthetic plants were planted as illustrated in Fig. 1. The dimensions of the subdomain Ω_2 are $2.94\,\text{m} \times 0.7\,\text{m} \times 0.5\,\text{m}$.

Note that Eq. (1) is solved in domain Ω_1 and Eq. (6) is solved in domain Ω_2. Since Eqs. (1) and (6) are coupled only via the velocity field $\boldsymbol{v}$, Eq. (1) can be solved without Eq. (6) and the latter can be solved only in a subdomain of Ω_1, i.e., $\Omega_2 \subset \Omega_1$.

2.3. Initial and boundary conditions

Both Eqs. (1) and (6) must be supplemented by initial and boundary conditions suitable for this problem. In all simulations presented in this paper, the velocity field was initialized by zero ($\boldsymbol{v}(\boldsymbol{x}, 0) = \boldsymbol{0}$) and the initial relative humidity profile varied with height as $\phi(\boldsymbol{x}, 0) = \phi_{\text{in}}(z)$, where $\phi_{\text{in}}(z)$ determines the inflow boundary condition for relative humidity (specified in Section 3.2.1). Various boundary conditions are used throughout the simulations. The sides, ceiling, and floor (including soil) of the test-section are impermeable and thus modeled using the standard no-slip boundary condition. A simple outflow condition based on a fixed pressure value and a zero velocity gradient in the normal direction is used on the left hand side of the domain in Fig. 1 (i.e., test-section exit). On the right hand side of the domains Ω_1 and Ω_2 in Fig. 1, the inflow boundary condition for velocity $\boldsymbol{v}$ and relative humidity ϕ, respectively, are prescribed. Numerical details related to these conditions are given in Sections 3.1.1, 3.1.2 and 3.2.1.

Furthermore, additional boundary conditions are applied on the sides of synthetic plants where "transpiration" is modeled by prescribing velocity and relative humidity profiles in order to match the experimentally measured mass flux of water lost through the surface of the plants. Water loss does not occur homogeneously on the surface of the synthetic plants since a capillary fringe of fully wetted surface was observed at the bottom of the plants. Therefore, water vapor is released with higher concentration near the soil surface and the top part of the plants is in equilibrium with the ambient environment. This is modeled in the simulations by prescribing a height-variable relative humidity profile on the sides of the synthetic plants as depicted in Fig. 2. Based on the experiments, the capillary fringe height is approximately $z_{\min} = 3.5$ cm. The prescribed relative humidity is $\phi_{\max} = 1$ below $z_{\min}$ and $\phi_{\min} = \phi_{\text{in}}(0.195)$ above $2z_{\min}$, where the ambient relative humidity $\phi_{\min}$ is set to the value prescribed on the inflow in the height of the synthetic plants. We also assume a transition part between $z_{\min}$ and $2z_{\min}$ where the surface is not fully wetted and the prescribed relative humidity decays linearly from $\phi_{\max}$ to $\phi_{\min}$. The profile shown in Fig. 2 corresponds to the configuration EX-2 with 45 cm spacing; the value of $\phi_{\min}$ was different in the other configurations.

To match the experimentally measured mass flux $\Phi_\star$ (see Section 2.1), we also need to prescribe a non-zero velocity in the normal direction on the sides of the plants. For example, given the mean relative humidity $\phi_{\text{mean}} = 0.437$ from Fig. 2 and assuming that saturated air has absolute humidity of $23\ \text{g}\,\text{m}^{-3}$ at 25 °C, the mass flux $\Phi_\star$ corresponds to the velocity $V_\star = 1.5\ \text{mm}\,\text{s}^{-1}$. The velocity $V_\star$ is prescribed only on the downstream side of each synthetic plant as a constant profile $v_x = -V_\star$, which coincides with the direction of the mean flow. On the remaining sides of the plants, the standard no-slip condition is used as we do not model advective transpiration on these sides for simplicity. On the upstream side, the velocity would have to be prescribed in the direction opposite to the mean flow, and on the sides parallel to the mean flow it would have to be prescribed in the tangential direction. However, water vapor is still released diffusively from the sides where the no-slip condition is used.

3. Coupled LBM-MHFEM computational approach

In this section, the computational approach for solving Eqs. (1) and (6) based on the combination of the lattice Boltzmann and mixed-hybrid finite element methods is described. Both methods are introduced and their coupling in a time-adaptive manner is explained. Finally, various implementation details are described.

3.1. Lattice Boltzmann method

The lattice Boltzmann method [8] (LBM) is an alternative to traditional computational methods such as finite difference, finite volume, and finite element methods. LBM can be formulated as a time-explicit scheme that can be easily parallelized [8] and the advent of general-purpose computing on graphics processing units (GPGPU) made large-scale numerical simulations of turbulent flows feasible [26–31]. For the simulation of air flow in the wind tunnel, we could take advantage of the previously developed parallel LBM algorithms validated in [32–34].

Instead of directly solving the Navier–Stokes equation (1), LBM approximates the temporal evolution of macroscopic quantities such as density ρ, velocity $\boldsymbol{v}$, and other variables (e.g., pressure, stress tensor, etc.) using probability moments of discrete density distribution functions $f_q = f_q(\boldsymbol{x}, t)$, $q = 1, \dots, Q$, where Q denotes the number of discrete velocities per lattice site. In this paper, we consider a regular 3D lattice $\hat{\Omega}$ covering the domain Ω_1 with the D3Q27 model consisting of $Q = 27$ discrete velocities per lattice site denoted as $\boldsymbol{\xi}_q$, $q = 1, \dots, Q$. The evolution of each f_q is described by the discrete Boltzmann transport equation

$$f_q(\hat{\boldsymbol{x}} + \Delta\hat{t}\boldsymbol{\xi}_q, \hat{t} + \Delta\hat{t}) - f_q(\hat{\boldsymbol{x}}, \hat{t}) = C_q(\hat{\boldsymbol{x}}, \hat{t}), \tag{7}$$

for all discrete lattice sites $\hat{\boldsymbol{x}} \in \hat{\Omega}$ and time levels $\hat{t} \in \hat{I} = \{0, \dots, N_t\}$. The dimensionless time step $\Delta\hat{t}$ is set to unity, i.e., $\Delta\hat{t} = 1$. The term C_q in Eq. (7) denotes the discrete collision operator; in this paper we

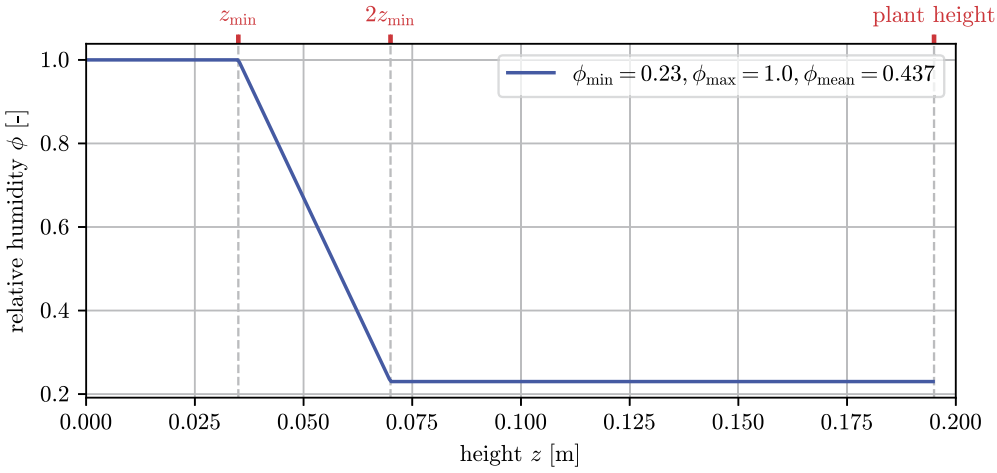


Fig. 2. Relative humidity profile prescribed as the boundary condition on the synthetic plants in the configuration with 45 cm spacing (EX-2).

use the cumulant operator proposed in [27] with the relaxation rates set as suggested therein and including the approximations of the spatial velocity derivatives to reduce the artifacts due to the absence of higher-order cumulants [35,36]. Macroscopic quantities can be recovered in $\hat{\Omega} \times \hat{I}$ by taking moments of the discrete density distribution functions f_q. The macroscopic density ρ (in lattice units) for example, is given by

$$\rho(\hat{\boldsymbol{x}}, \hat{t}) = \sum_{q=1}^{Q} f_q(\hat{\boldsymbol{x}}, \hat{t}) \tag{8}$$

and the macroscopic velocity $\boldsymbol{v}$ computed from the macroscopic momentum density $\rho\boldsymbol{v}$ (in lattice units) given by

$$\rho(\hat{\boldsymbol{x}}, \hat{t})\boldsymbol{v}(\hat{\boldsymbol{x}}, \hat{t}) = \sum_{q=1}^{Q} f_q(\hat{\boldsymbol{x}}, \hat{t})\boldsymbol{\xi}_q. \tag{9}$$

The detailed derivation of LBM is not within the scope of this paper; the reader is referred to [8] for details.

We use the full-way bounce-back boundary condition [8] to prescribe the no-slip condition on impermeable walls. The boundary condition for velocity on the downstream faces of synthetic plants (see Section 2.3) is realized via the modified bounce-back condition [8] by specifying zero tangential and small non-zero normal velocity of the moving wall. On the inflow, the discrete distribution functions are approximated by the discrete equilibrium functions evaluated from the known macroscopic variables [37–39]. The macroscopic velocity profiles that were considered for the inflow boundary condition in this work are described in the following subsections. On the outflow, the extrapolation outflow boundary condition is used to approximate the discrete distribution functions.

3.1.1. Inflow boundary condition: mean velocity profile

Using this boundary condition, a time-constant inflow velocity profile approximating the mean free flow velocity measured in the wind tunnel is prescribed. The inflow velocity is set to $\boldsymbol{v}_{\text{in}} = [-v_{\text{in},x}(z), 0, 0]$, where the component $v_{\text{in},x}$ is specified as a function of height z [m] based on the 1/7-th power law [40]:

$$v_{\text{in},x}(z) = \begin{cases} v_{\max}\left(\dfrac{z}{z_\delta}\right)^{\frac{1}{7}} & \text{if } z \le z_\delta, \\ v_{\max}\left(\dfrac{z_{\max} - z}{z_\delta}\right)^{\frac{1}{7}} & \text{if } z_{\max} - z \le z_\delta, \\ v_{\max} & \text{otherwise,} \end{cases} \tag{10}$$

where $z_{\max} = 1.0624$ m is the height of the domain Ω_1 in Fig. 1 at the inflow boundary, $z_\delta = 0.1$ m is the estimated boundary layer height and $v_{\max} = 0.8\,\text{m}\,\text{s}^{-1}$ corresponds to the mean free stream velocity in the wind tunnel. The parameters were chosen based on the experimentally measured velocity profiles and the same values are used in all three spacing configurations.

3.1.2. Inflow boundary condition: velocity fluctuations

As will be seen in Section 4, prescribing a time-constant velocity profile at the inflow boundary may lead to non-physical results, because the simulated flow field may remain laminar until it reaches the first obstacle placed in the domain. An alternative is to induce turbulent flow by adding synthetic fluctuations to the prescribed velocity profile. The inflow velocity $\boldsymbol{v}_{\text{in}} = \boldsymbol{v}_{\text{in}}(\boldsymbol{x}, t)$ is decomposed as

$$\boldsymbol{v}_{\text{in}}(\boldsymbol{x}, t) = \overline{\boldsymbol{v}}_{\text{in}}(\boldsymbol{x}) + \boldsymbol{v}'_{\text{in}}(\boldsymbol{x}, t), \tag{11}$$

where $\overline{\boldsymbol{v}}_{\text{in}}(\boldsymbol{x})$ is the mean (time-averaged) value given in Section 3.1.1, and $\boldsymbol{v}'_{\text{in}}(\boldsymbol{x}, t)$ is the velocity fluctuation.

The procedure for generating synthetic turbulent fluctuating velocity field $\boldsymbol{v}'_{\text{in}}$ is based on [41–44]. In the simulation, inflow velocity fluctuations for a uniform grid with the spacing Δx are computed at discrete time levels $t_n = n\Delta t$, where n is an integer denoting the time level and Δt is the time step. Firstly, independent realizations of random fluctuations $\hat{\boldsymbol{v}}'_{\text{in}}$ are generated for each time level for the specified length scale $\mathcal{L}_{\text{int}}$ and energy spectrum of synthetic turbulence. In this work, we set the turbulent length scale to one half of the inflow boundary layer z_δ used in Eq. (10), i.e. $\mathcal{L}_{\text{int}} = 0.05$ m, and use the modified von Kármán spectrum [42,44] with the highest wave number $\kappa_{\max} = 2\pi/\Delta x$, smallest wave number $\kappa_{\min} = \kappa_e/5$, 3000 discrete modes, turbulent kinetic energy $k_{\text{in}} = 10^{-2}\,\text{m}^2\,\text{s}^{-2}$, and kinematic viscosity ν given in Table 2. Then, time correlation between the realizations is introduced using an asymmetric time filter

$$(\boldsymbol{v}'_{\text{in}})^n = a(\boldsymbol{v}'_{\text{in}})^{n-1} + b(\hat{\boldsymbol{v}}'_{\text{in}})^n, \tag{12}$$

where $\boldsymbol{v}'_{\text{in}}$ denotes the time-correlated field, $\hat{\boldsymbol{v}}'_{\text{in}}$ denotes the time-independent field, subscripts denote the time levels and the coefficients are chosen as $a = \exp(-\Delta t/\mathcal{T}_{\text{in}})$ and $b = \sqrt{1 - a^2}$, where $\mathcal{T}_{\text{int}} = \mathcal{L}_{\text{int}}/|v_{x,\text{in},\max}|$. The time filter ensures that $\mathcal{T}_{\text{int}}$ corresponds to the turbulent integral time scale and that the variance of the generated fluctuations is preserved [44].

Note that the fluctuations generated with the aforementioned procedure are isotropic which is an acknowledged simplification of anisotropic real-world turbulence. The procedure could be improved based on a specified anisotropic Reynolds stress tensor [44], however, even using the inflow condition based on isotropic synthetic turbulence lead to improved results in this work. Experimentally, the problem was treated as 2D, i.e., the transverse flow statistics, such as the components $\overline{v'_y v'_y}$, $\overline{v'_x v'_y}$, and $\overline{v'_y v'_z}$ of the Reynolds stress tensor, were not quantified.

3.2. Mixed-hybrid finite element method

The transport equation (6) has been incorporated into the mathematical framework of our previous work, the *NumDwarf* solver [10]. *NumDwarf* was originally developed for simulating multicomponent transport phenomena in porous media, but the numerical scheme is implemented for a PDE system written in the form

$$\sum_{j=1}^{n} N_{i,j} \frac{\partial Z_j}{\partial t} + \sum_{j=1}^{n} \boldsymbol{u}_{i,j} \cdot \nabla Z_j + \nabla \cdot \left[m_i \left(-\sum_{j=1}^{n} \mathbf{D}_{i,j} \nabla Z_j + \boldsymbol{w}_i \right) + \sum_{j=1}^{n} Z_j \boldsymbol{a}_{i,j} \right] + \sum_{j=1}^{n} r_{i,j} Z_j = f_i \tag{13}$$

for $i = 1, \dots, n$, where $\boldsymbol{Z} = [Z_1, \dots, Z_n]^T$ is the vector of unknown functions depending on spatial and temporal coordinates and the symbols $\mathbf{N} = [N_{i,j}]_{i,j=1}^n$, $\boldsymbol{u} = [\boldsymbol{u}_{i,j}]_{i,j=1}^n$, $\boldsymbol{m} = [m_i]_{i=1}^n$, $\mathbf{D} = [\mathbf{D}_{i,j}]_{i,j=1}^n$, $\boldsymbol{w} = [\boldsymbol{w}_i]_{i=1}^n$, $\boldsymbol{a} = [\boldsymbol{a}_{i,j}]_{i,j=1}^n$, $\boldsymbol{r} = [r_{i,j}]_{i,j=1}^n$, $\boldsymbol{f} = [f_i]_{i=1}^n$ are given coefficients. Note that the coefficients may, in general, depend on $\boldsymbol{Z}$. After spatial and temporal discretizations, the derivation of the numerical scheme applies the semi-implicit approach of the frozen coefficients method to non-linear terms in Eq. (13). Hence, only the terms in Eq. (13) containing Z_j are treated implicitly and the coefficients are either independent of $\boldsymbol{Z}$ or use the values of $\boldsymbol{Z}$ from the current time level.

NumDwarf can be used for any problem whose governing equations can be written in a compatible form, including multiphase and multicomponent flow in porous media [45,46]. Both Eq. (6a) and Eq. (6b) can be converted to this form by taking $n = 1$, $\boldsymbol{Z} = [\phi]$, $\mathbf{N} = [1]$, $\boldsymbol{m} = [1]$, $\boldsymbol{D} = [D]$, $\boldsymbol{w} = \mathbf{0}$, $\boldsymbol{r} = \mathbf{0}$, $\boldsymbol{f} = [0]$, and depending on the equation:

- $\boldsymbol{u} = \mathbf{0}$, $\boldsymbol{a} = [\boldsymbol{v}]$ for Eq. (6a),
- $\boldsymbol{u} = [\boldsymbol{v}]$, $\boldsymbol{a} = \mathbf{0}$ for Eq. (6b).

The effects of coupling LBM with MHFEM for a conservative and a non-conservative formulation of the problem are studied on a benchmark problem in Appendix A.

The system of Eqs. (13) must be supplemented by a suitable initial condition $\boldsymbol{Z}(\boldsymbol{x}, 0) = \boldsymbol{Z}_{\text{ini}}(\boldsymbol{x})$ and boundary conditions. A Dirichlet-type boundary condition is used to prescribe fixed values of the primary variable on the inflow and bottom parts of the domain boundary $\partial\Omega_2$. The exact values of relative humidity used in this work are specified later in Section 3.2.1. On the remaining parts of $\partial\Omega_2$, which are either impermeable walls where the no-slip condition on velocity is imposed or free-stream boundaries of Ω_2 inside Ω_1, a Neumann-type condition is used to prescribe zero gradient of the unknown function in the normal direction.

NumDwarf is based on the mixed-hybrid finite element method (MHFEM). In this work, we use conforming unstructured cuboidal meshes denoted as $\mathcal{K}_h$ for the discretization of the domain Ω_2. The set of all faces of the mesh will be denoted as $\mathcal{E}_h$, the subsets of interior and exterior faces of the mesh will be denoted as $\mathcal{E}_h^{\text{int}}$ and $\mathcal{E}_h^{\text{ext}}$, respectively, and the set of faces of an individual element $K \in \mathcal{K}_h$ will be denoted as $\mathcal{E}_K$. Using the discontinuous Galerkin method, scalar functions are approximated in the space of piecewise constant functions $D_0(\mathcal{K}_h)$, where the basis functions ϕ_K are indicator functions for mesh elements $K \in \mathcal{K}_h$. Vector functions are approximated in the lowest-order Raviart–Thomas–Nédélec function space $\mathbf{RTN}_0(\mathcal{K}_h)$.

We denote the diffusive flux in Eq. (13) by $\boldsymbol{q}_i = m_i \boldsymbol{v}_i$, where

$$\boldsymbol{v}_i = -\sum_{j=1}^{n} \mathbf{D}_{i,j} \nabla Z_j + \boldsymbol{w}_i. \tag{14}$$

The approximated vector functions $\boldsymbol{v}_i$, $\boldsymbol{u}_{i,j}$, $\boldsymbol{a}_{i,j}$ are given in the basis of the space $\mathbf{RTN}_0(K)$ as

$$\boldsymbol{v}_i|_K = \sum_{E \in \mathcal{E}_K} v_{i,K,E} \boldsymbol{\omega}_{K,E}, \quad \boldsymbol{u}_{i,j}|_K = \sum_{E \in \mathcal{E}_K} u_{i,j,K,E} \boldsymbol{\omega}_{K,E},$$
$$\boldsymbol{a}_{i,j}|_K = \sum_{E \in \mathcal{E}_K} a_{i,j,K,E} \boldsymbol{\omega}_{K,E}, \tag{15}$$

where $\boldsymbol{\omega}_{K,E} \in \mathbf{RTN}_0(K)$ are the finite element basis functions. The discretization of $\boldsymbol{q}_i$ and $\boldsymbol{v}_i$ subsequently follows the usual procedure described in [10]. Note that the mass-lumping technique is used to stabilize the scheme. The discretization of the terms with velocities $\boldsymbol{u}_{i,j}$ and $\boldsymbol{a}_{i,j}$ is also briefly outlined in [10], but here we provide more details for convenience. By multiplying the terms $\boldsymbol{u}_{i,j} \cdot \nabla Z_j$ and $\nabla \cdot (Z_j \boldsymbol{a}_{i,j})$ by a basis function $\phi_K \in D_0(\mathcal{K}_h)$, integrating over the domain Ω_2 and using the Green's formula together with the properties of the $\mathbf{RTN}_0(K)$ basis functions, we obtain

$$\int_{\Omega_2} \boldsymbol{u}_{i,j} \cdot \nabla Z_j \varphi_K = \int_{\partial K} Z_j \varphi_K \boldsymbol{u}_{i,j} \cdot \boldsymbol{n}_{\partial K} - \int_K Z_j \varphi_K \nabla \cdot \boldsymbol{u}_{i,j}$$
$$\approx \sum_{E \in \mathcal{E}_K} Z^{\text{upw}}_{i,j,K,E} u_{i,j,K,E} - Z_{j,K} \sum_{E \in \mathcal{E}_K} u_{i,j,K,E}, \tag{16a}$$

$$\int_{\Omega_2} \nabla \cdot \left(Z_j \boldsymbol{a}_{i,j} \right) \varphi_K = \int_{\partial K} Z_j \varphi_K \boldsymbol{a}_{i,j} \cdot \boldsymbol{n}_{\partial K} \approx \sum_{E \in \mathcal{E}_K} Z^{\text{upw}}_{i,j,K,E} a_{i,j,K,E}, \tag{16b}$$

where $\boldsymbol{n}_{\partial K}$ denotes the outward unit normal vector on ∂K, $Z_{j,K}$ represents the mean value of Z_j on the element K and $Z^{\text{upw}}_{i,j,K,E}$ is a yet undefined value of Z_j on the face $E \in \mathcal{E}_K$ as viewed from the element $K \in \mathcal{K}_h$. Note that the discretization of the advection terms differs in the term containing $\sum_{E \in \mathcal{E}_K} u_{i,j,K,E}$, which corresponds to the discretization of $\nabla \cdot \boldsymbol{u}_{i,j}$ on the element $K \in \mathcal{K}_h$. This is the key distinction between the conservative and non-conservative formulations, as will be clear from the numerical results.

The discrete scheme for Eq. (13) is obtained by applying the Euler method for temporal discretization and the semi-implicit approach of the frozen coefficients method for the linearization in time. Consistently with [10, Eq. (15)], it can be written as

$$\frac{|K|}{\Delta t_k} \sum_{j=1}^{n} N^k_{i,j,K} \left(Z^{k+1}_{j,K} - Z^k_{j,K} \right) + \sum_{j=1}^{n} \sum_{E \in \mathcal{E}_K} Z^{\text{upw}}_{i,j,K,E} \left(a^k_{i,j,K,E} + u^k_{i,j,K,E} \right)$$
$$+ \sum_{E \in \mathcal{E}_K} m^{k,\text{upw}}_{i,E} v^{k+1}_{i,K,E}$$
$$+ \sum_{j=1}^{n} \left(|K| r^k_{i,j,K} - \sum_{E \in \mathcal{E}_K} u^k_{i,j,K,E} \right) Z^{k+1}_{j,K} = |K| f^k_{i,K} \tag{17}$$

for all $K \in \mathcal{K}_h$, $i \in \{1, \dots, n\}$ and time levels $k \in \mathbb{N}$. Here $|K|$ denotes the volume of the element K and $\Delta t_k = t_{k+1} - t_k$ is the discrete time step. The superscripts k and $k+1$ indicate the time level where the corresponding term is evaluated. The symbols $N^k_{i,j,K}$, $m^{k,\text{upw}}_{i,E}$, $r^k_{i,j,K}$, and $f^k_{i,K}$ represent coefficients obtained from the discretization of the corresponding quantities in Eq. (13), see [10] for details.

Two upwind schemes for the term $Z^{\text{upw}}_{i,j,K,E}$ are tested in this paper. The first one, hereafter denoted as *explicit upwind*, evaluates the term at time level $t = t_k$ as

$$Z^{\text{upw}}_{i,j,K_1,E} \equiv Z^{k,\text{upw}}_{i,j,K_1,E} = \begin{cases} Z^k_{j,K_1} & \text{if } a^k_{i,j,K_1,E} + u^k_{i,j,K_1,E} \geq 0, \\ Z^k_{j,K_2} & \text{otherwise,} \end{cases} \tag{18}$$

for all interior faces $E \in \mathcal{E}_h^{\text{int}}$ such that $E \in \mathcal{E}_{K_1} \cap \mathcal{E}_{K_2}$. Note that this scheme is symmetric, i.e., $Z^{k,\text{upw}}_{i,j,K_1,E} = Z^{k,\text{upw}}_{i,j,K_2,E}$. This scheme was used also in [10]. The second scheme, hereafter denoted as *implicit upwind*, was introduced in [11,47] and evaluates the term at time level $t = t_{k+1}$ as

$$Z^{\text{upw}}_{i,j,K_1,E} \equiv Z^{k+1,\text{upw}}_{i,j,K_1,E} = \begin{cases} Z^{k+1}_{j,K_1} & \text{if } a^k_{i,j,K_1,E} + u^k_{i,j,K_1,E} \geq 0, \\ 2Z^{k+1}_{j,E} - Z^{k+1}_{j,K_1} & \text{otherwise,} \end{cases} \tag{19}$$

where $Z_{j,E}^{k+1}$ is the unknown trace of Z_j on the face $E \in \mathcal{E}_h^{\text{int}}$ (also called a Lagrange multiplier in the context of mixed finite element methods). The motivation for this choice is that since $Z_{j,E}$ provides an approximation to the mean value $\frac{1}{2}(Z_{j,K_1} + Z_{j,K_2})$, then $2Z_{j,E} - Z_{j,K_1}$ approximates Z_{j,K_2}. This allows to proceed with the hybridization procedure in MHFEM even when upwinded terms are treated implicitly, but the scheme is not symmetric, i.e., $Z_{i,j,K_1,E}^{k+1,\text{upw}} \neq Z_{i,j,K_2,E}^{k+1,\text{upw}}$. However, it is not a conceptual problem since the matrix of the resulting system of linear algebraic equations is non-symmetric even when the explicit upwind scheme is applied. Both upwind schemes are compared on a benchmark problem in Appendix A.

The linear system of equations for the unknown values $Z_{j,K}^{k+1}$, $K \in \mathcal{K}_h$ and $Z_{j,E}^{k+1}$, $E \in \mathcal{E}_h$ is closed by adding equations that represent the balance of the normal components of the flux $\boldsymbol{Q}_i = \boldsymbol{q}_i + \sum_{j=1}^{n} Z_j(\boldsymbol{a}_{i,j} + \boldsymbol{u}_{i,j})$ across interior faces of the mesh. The balance conditions can be written in the discrete form as

$$\sum_{\ell=1}^{2} \left[m_{i,E}^k v_{i,K_\ell,E}^{k+1} + \sum_{j=1}^{n} Z_{i,j,K_\ell,E}^{\text{upw}} \left(a_{i,j,K_\ell,E}^k + u_{i,j,K_\ell,E}^k \right) \right] = 0 \tag{20}$$

for all $i \in \{1, \dots, n\}$ and all interior faces $E \in \mathcal{E}_h^{\text{int}}$ such that $E \in \mathcal{E}_{K_1} \cap \mathcal{E}_{K_2}$. Note that $v_{i,K_\ell,E}^{k+1}$ is given in terms of Z_{j,K_ℓ}^{k+1} and $Z_{j,E}^{k+1}$, see [10]. For exterior faces $E \in \mathcal{E}_h^{\text{ext}}$, constraints given by the discretized boundary conditions are used instead of Eq. (20). The following steps depend on which upwind technique is used. In case of the explicit upwind scheme Eq. (18), the advection terms in Eq. (20) cancel out, because the upwind scheme is symmetric and both velocities $\boldsymbol{a}_{i,j}$ and $\boldsymbol{u}_{i,j}$ are assumed to belong to the space $\mathbf{RTN}_0(\mathcal{K}_h)$. Hence, the resulting conditions enforce only the balance of the diffusive flux $\boldsymbol{q}_i$, which leads to the scheme described in [10]. In case of the non-symmetric implicit upwind scheme Eq. (19), however, the balance Eq. (20) must be considered as a whole and the MHFEM modification from [10] that allows the scheme with explicit upwind to deal with vanishing diffusion in non-linear PDEs no longer applies. Nonetheless, the implicit upwind scheme is applicable to the linear transport equation (6) that is considered in this work. Detailed analysis of this upwind scheme can be found in [47].

The final discrete scheme is obtained by following the hybridization procedure, see [9–11]. The per-element averages $Z_{j,K}^{k+1}$, $K \in \mathcal{K}_h$ are eliminated from the large, indefinite system of linear algebraic equations (Eqs. (17) and (20) with boundary conditions) and a smaller, definite system of linear algebraic equations for the traces $Z_{j,E}^{k+1}$, $E \in \mathcal{E}_h$ is obtained. The MHFEM computational algorithm leads to the solution of one non-symmetric linear system per time step, it allows to use unstructured meshes and can be efficiently parallelized.

3.2.1. Inflow boundary condition for relative humidity

In the problem described in Section 2, the inflow boundary condition for relative humidity ϕ is specified as a function $\phi_{\text{in}}(z)$ of height z [m]:

$$\phi_{\text{in}}(z) = \begin{cases} \phi_{\max} + \phi_{\text{lin}} \dfrac{z_{\text{pow}} - z}{z_{\text{pow}}} & \text{if } z \le z_{\text{pow}}, \\ \phi_{\max} - (\phi_{\max} - \phi_{\min}) \left(\dfrac{z - z_{\text{pow}}}{z_{\text{const}} - z_{\text{pow}}} \right)^{\gamma} & \text{if } z > z_{\text{pow}} \text{ and } z \le z_{\text{const}}, \\ \phi_{\min} & \text{if } z > z_{\text{const}}, \end{cases} \tag{21}$$

where the common parameters $\phi_{\text{lin}} = 0.01$ and $z_{\text{pow}} = 0.005$ m describe the profile below the minimal height for measurements, $z_{\text{const}} = 0.195$ m corresponds to the height of the synthetic plants, and the remaining parameters $\phi_{\min}$, $\phi_{\max}$, γ are fitted values to the experimental data (see Table 3 and the supplementary materials). Furthermore, the value $\phi_{\text{in}}(0)$ is used for the fixed-value boundary condition at the bottom side of the domain Ω_2 that coincides with the interface between the soil tank and free space.

Table 3
Values of fitted parameters $\phi_{\min}$, $\phi_{\max}$, γ for the relative humidity inflow boundary condition (21).

	EX-1	EX-2	EX-3
$\phi_{\min}$	0.236	0.230	0.217
$\phi_{\max}$	0.270	0.275	0.265
γ	0.184443	0.262402	0.402284

3.3. Computational algorithm and time adaptivity

The previous two sections described the numerical methods used for the discretizations of Eqs. (1) and (6), respectively. In this section, we describe the main computational algorithm of the coupled scheme focusing on its adaptive time control.

The initialization of the solver includes setting all physical parameters, discretization of the domain Ω_1 by the lattice $\hat{\Omega}$, discretization of the domain Ω_2 by the unstructured mesh $\mathcal{K}_h$, and decomposition of the lattice and mesh into subdomains for distributed computing. Then the solver allocates all data structures, applies the initial conditions and starts the main time-loop. To optimize the efficiency of the solver, we developed an adaptive time-stepping algorithm based on a "CFL-like" condition. The time-stepping part of the computational algorithm can be summarized as follows:

1. Set the time tracking variables $t_L := 0$ and $t_M := 0$, physical time step Δt and final time $t_{\max} = N_t \Delta t$.
2. While $t_L < N_t \Delta t$, repeat these steps:
   (a) After every 1000 iterations, recompute inflow velocity fluctuations that will be used in the next 1000 iterations.
   (b) Perform one iteration of LBM.
   (c) Set $t_L := t_L + \Delta t$.
   (d) If $t_M < t_L$, perform these steps:
      i. Interpolate the velocity field from the regular lattice to the unstructured mesh. See Section 3.4 for details regarding this procedure.
      ii. Compute $C = \max_E \{ |\boldsymbol{v}_E| \Delta t / |E| \}$, where $E \in \mathcal{E}_h$ goes over all faces of the unstructured mesh, $|\boldsymbol{v}_E|$ is the magnitude of the interpolated velocity on the face E and $|E|$ is the characteristic length of the face E.
      iii. Set the time step for MHFEM: $\Delta t_M := \Delta t \lfloor C_{\max}/C \rfloor$ if $C \le C_{\max}$, else $\Delta t_M := \Delta t / \lceil C/C_{\max} \rceil$.
      iv. Set the number of MHFEM iterations: $n_M := 1$ if $C \le C_{\max}$, else $n_M := \lceil C/C_{\max} \rceil$.
      v. Perform n_M iterations of MHFEM with the time step Δt_M. The MHFEM algorithm is described in [10].
      vi. Set $t_M := t_M + n_M \Delta t_M$.
   (e) If t_L is a multiple of a pre-defined time period, make a snapshot of the current data for visualization/post-processing.

Initially, two separate time tracking variables t_L and t_M are created, one for computations on the lattice and the other for computations on the mesh. The lattice variable t_L is the main one which is checked in the time loop condition (step 2 above). The variable t_M is incremented separately in the step 2d that is responsible for the coupling between computations on the lattice and the mesh. Here the velocity field is interpolated from the lattice to the mesh and the time-step control factor C is computed. The time step Δt_M for the mesh computations can be either longer or shorter than Δt depending on the value of C. If $C \le C_{\max}$, where $C_{\max}$ is a pre-defined constant, the time step Δt_M is set as an integral multiple of Δt and the time variable t_M is pushed forward to the new time level by just one iteration of MHFEM. Hence, several following LBM iterations can be performed successively until the lattice time t_L overruns t_M and the condition 2d is satisfied. On the other hand, if $C > C_{\max}$, the MHFEM solver needs to perform $n_M > 1$ iterations to move to the new time level $t_M + \Delta t$ with a shorter time step $\Delta t_M = \Delta t / n_M$.

In practice, we found empirically that limiting the time step Δt_M with $C_{\max} = \frac{1}{2}$ is necessary to ensure the stability of the coupled solver. The overall performance of the solver depends on the concrete values of Δt_M selected by the adaptive algorithm, which are influenced by the quantities needed to compute the time-step control factor C, i.e., local velocity magnitude, lattice time step (which is related to the lattice space step), and the local space step of the unstructured mesh (expressed by $|E|$ in the algorithm).

3.4. Interpolation of the velocity field

Since Eqs. (1) and (6) are coupled by the velocity field $\boldsymbol{v}(\boldsymbol{x}, t)$, the numerical approach relies on the interpolation of the approximate velocity field computed by LBM and its projection into the finite element space used by MHFEM. Note that the spatial discretization of the domain $\Omega_2 \subset \Omega_1$ is generally different than the equidistant lattice on Ω_1; it may be a regular grid with different space steps or even an unstructured mesh.

The interpolation of the velocity field can be requested at any point $\boldsymbol{x} \in \Omega_2$. The surrounding lattice points $\hat{\boldsymbol{x}} \in \hat{\Omega}$ can be easily found and the linear or cubic interpolation in $\mathbb{R}^3$ can be used to obtain the velocity at $\boldsymbol{x}$ from the velocities at $\hat{\boldsymbol{x}}$. Note that linear interpolation can be implemented more efficiently than cubic interpolation as it uses fewer input data points. Our implementation of the cubic interpolation causes the whole solver to run about three times slower than when using the linear interpolation. Both linear and cubic interpolations are compared on a benchmark problem in Appendix A and based on those results, linear interpolation is used for the main simulations presented in Section 4.

The finite element space used by MHFEM imposes requirements on the interpolation of the velocity field. According to [9], the Raviart–Thomas–Nédélec space of the lowest order $\mathbf{RTN}_0(\mathcal{K}_h)$ that is used for the finite element–approximation of the velocity field in this work is formed by functions $\boldsymbol{\omega} \in [L^2(\Omega_2)]^3$ such that:

1. for any element $K \in \mathcal{K}_h$, the restriction of $\boldsymbol{\omega}$ to K, $\boldsymbol{\omega}|_K$, must belong to the finite element space $\mathbf{RTN}_0(K)$ on the element K,
2. $\boldsymbol{\omega}$ satisfies the balancing condition for normal traces on all interior faces $E \in \mathcal{E}_h^{\text{int}}$ of the mesh, i.e., $\int_E \boldsymbol{\omega}|_{K_1} \cdot \boldsymbol{n}_{K_1,E} + \int_E \boldsymbol{\omega}|_{K_2} \cdot \boldsymbol{n}_{K_2,E} = 0$ for all $E \in \mathcal{E}_h^{\text{int}}$, $E \in \mathcal{E}_{K_1} \cap \mathcal{E}_{K_2}$, where $\boldsymbol{n}_{K_\ell,E}$ is the unit normal vector on the face E oriented outward from the element K_ℓ, $\ell = 1, 2$.

An interpolation strategy compatible with these requirements is as follows. First, velocity is evaluated at the element face centers $\boldsymbol{x}_E$ for all $E \in \mathcal{E}_h$. This can be done at any time level t yielding the approximate velocity values $\hat{\boldsymbol{v}}(\boldsymbol{x}_E, t)$ which are then used for the projection into the $\mathbf{RTN}_0(\mathcal{K}_h)$ space. The discrete velocity field is assumed to be piecewise constant on the element sides $\mathcal{E}_h$ and the values $\hat{\boldsymbol{v}}(\boldsymbol{x}_E, t)$ define the components corresponding to the face E in the finite element spaces $\mathbf{RTN}_0(K_j)$ of the elements K_j adjacent to the face E.

Finally, it is important to note that numerical schemes for Eqs. (6a) and (6b) do not behave equivalently with a general discrete velocity field interpolated to the mesh. This is because the discrete velocity field computed by LBM may not satisfy Eq. (1a) exactly and even if it did, the interpolation scheme combines values from different locations in the flow field on a single element. Hence, the field interpolated to the unstructured mesh may be locally non-conservative, i.e., the discrete approximation of the velocity divergence $\sum_{E \in \mathcal{E}_K} \hat{\boldsymbol{v}}(\boldsymbol{x}_E, t) \cdot \boldsymbol{n}_{K,E}$ on element $K \in \mathcal{K}_h$ may be non-zero. The accuracy of the numerical scheme applied to the conservative form of Eq. (6a) and non-conservative form of Eq. (6b) are studied in Appendix A on a benchmark problem. The presented results show that solving the conservative transport equation with a highly turbulent velocity field may lead to large deviations in the numerical solution, whereas solving the non-conservative transport equation with the same velocity field results in significantly more accurate solution. Therefore, the non-conservative formulation of the transport equation (6b) is used for the main simulations in Section 4.

An alternative approach to address the problem of non-conservative velocity field might be to use a post-processing algorithm to recover the discrete divergence-free condition on the given mesh. The problem of compatibility between flow schemes producing a discrete velocity field and transport schemes using the interpolated velocity was extensively researched and several velocity post-processing algorithms were developed [48–50]. However, such post-processing would incur additional cost to the computational algorithm and the approach is not investigated further in this paper.

3.5. Implementation remarks

The LBM and MHFEM parts of the coupled solver were developed and tested separately in [32–34] and [10], respectively. Both parts individually as well as the algorithms specific to the coupling between them utilize the open-source TNL library [51] for parallelization and distributed computing on GPU clusters. TNL natively supports and provides a unified high-level interface for modern parallel architectures such as CPUs, GPU accelerators (via CUDA [52]) and distributed systems (via MPI [13]). Furthermore, TNL provides common building blocks for numerical solvers, including data structures and parallel algorithms for linear algebra, structured grids and unstructured meshes. Using the data structures and algorithms from TNL is beneficial for performance, because they allow to avoid running expensive computations on the CPU and having to transfer large datasets between the system memory and accelerators over the PCI-E bus. Instead, all expensive parts of the computational algorithm are executed on the GPU accelerators and the CPU is responsible only for the orchestration of the work and occasional sequential steps such as handling input and output.

The coupled solver uses a regular lattice for LBM and an unstructured mesh for MHFEM. The use of the lattice is the main limiting factor for the flexibility of the solver, because extra care must be taken when setting up a simulation to ensure proper alignment of immersed boundaries such as the synthetic plants used in this paper. This could be improved by using interpolated boundary conditions for LBM [8], they are however not currently implemented in our solver. On the other hand, the MHFEM part of the solver can be used on complex domain geometries with unstructured mesh discretizations. The details related to the decomposition of a regular lattice overlapped with an unstructured mesh are described in the following section.

An important choice related to the solver performance is the selection of the algorithm for the solution of large systems of linear equations. We have found that the best performance is obtained using the BiCGstab method combined with the Jacobi preconditioner, both implemented in TNL [51]. In the simulations presented in this paper, the BiCGstab method took at most 4 iterations to converge in most of the time steps, so improved performance cannot be expected from stronger preconditioners.

3.6. Domain decomposition for overlapped lattice and mesh

The combination of a lattice overlapped with an unstructured mesh requires special attention when the solver is run in a distributed fashion, e.g. utilizing multiple GPU accelerators. Both the lattice and the mesh have to be decomposed into subdomains and each assigned to a GPU. Sufficiently wide overlapping regions on the lattice subdomains have to be generated to ensure that each GPU can interpolate the velocity field from its lattice subdomains to its mesh subdomain. Furthermore, since computations on the lattice and the mesh are never executed concurrently, it is desirable to balance the sizes of the subdomains in order to achieve good computational efficiency.

Fig. 3 illustrates the problems with decomposition on an example involving a non-uniform cuboidal mesh that is refined around the two synthetic plants in configuration EX-1. Due to limitations of our LBM implementation, only 1D decompositions (i.e., such that all interfaces between two lattice subdomains are planes perpendicular to the x-axis)

can be considered. Fig. 3a shows a naive approach with uniformly sized lattice subdomains (highlighted with rainbow-colored rectangles), which leads to highly non-uniform distribution of mesh cell counts in each subdomain (indicated by percentages below the figure). In order to solve this balancing problem, we implemented a decomposition strategy which optimizes the lattice as well as mesh subdomains such that each GPU is assigned approximately the same number of lattice sites as well as mesh cells. The essential idea is to first determine the part of the domain where the lattice and mesh overlap, perform its decomposition such that an optimal mesh decomposition is achieved, and then decompose the remaining parts of the lattice (which do not overlap with the mesh) to add up to the optimal number of lattice sites in each subdomain.

For a given regular lattice and an unstructured mesh covering the domain Ω_1 and its subdomain Ω_2, respectively, the decomposition procedure (with N_{ranks} denoting the number of MPI ranks used in the computation and N_{cells} denoting the total number of mesh cells) can be summarized as follows:

1. For all x-coordinates of the lattice sites, count the number of mesh cells whose centroid is located left of this x-coordinate. Use linear interpolation to obtain a continuous interpolant function $F(x)$ that is increasing from 0 to N_{cells}.
2. Find the smallest interval $[x_0, x_{N_{\text{ranks}}}]$ such that $F(x_{N_{\text{ranks}}}) - F(x_0) = N_{\text{cells}}$. This interval identifies the part of the lattice that is overlapped by the mesh, i.e., the dark transparent rectangle in Fig. 3.
3. Find a partition $\{x_0, x_1, \dots, x_{N_{\text{ranks}}-1}, x_{N_{\text{ranks}}}\}$ of the interval $[x_0, x_{N_{\text{ranks}}}]$ such that each subinterval contains approximately $N_{\text{cells}}/N_{\text{ranks}}$ mesh cells:
   (a) Define the objective function $f(x_1, \dots, x_{N_{\text{ranks}}-1})$ which measures the imbalance of mesh cells included in each subinterval based on the function F.
   (b) Minimize the objective function using the gradient descent method and the uniform interval partition as initial condition.
   (c) Round the solution from $\mathbb{R}$ to the lattice coordinates (i.e., from `double` to `int`). As the rounding does not ensure the optimal result in integer precision, we additionally minimize the objective function in integer precision. We try to iteratively increment/decrement each component of the solution as long as it improves the partition.
4. Decompose the remaining parts of the lattice which do not overlap with the mesh. Note that these parts of the lattice are decomposed separately in reversed order (i.e., from right to left) in order to allow merging the non-overlapping subdomains with the adjacent mesh-overlapping subdomains (see the red and gray subdomains in Fig. 3b).

The result of this decomposition procedure is illustrated in Fig. 3b. Overall, the decomposition algorithm optimizes the computational cost and memory requirements of each MPI rank at the cost of increased communication due to increased number of lattice subdomains.

4. Validation results

4.1. Computational methodology

All three experimental configurations were simulated up to the final time $t_{\text{max}} = 100$ s in three different resolutions, hereafter denoted as RES-1, RES-2, and RES-3. A reference lattice and mesh were generated for the initial resolution RES-1 and the space step is halved with each subsequent resolution. The simulations were computed on a system with two AMD EPYC 7763 processors and eight Nvidia Tesla A-100 GPU accelerators with NVLink interconnection. See Table 4 for the characteristics of each resolution and computational resources needed for the simulations.

To match the experimental methodology described in Section 2.1, time-averaging was employed to produce statistical quantities such as the mean and variance of the flow velocity and relative humidity. Time-averaging is implemented as part of the simulation code where statistical quantities are updated in every time step using the Welford's online algorithm [53–55]. Note that no space-averaging is applied to the simulation results.

4.2. On qualitative and quantitative approaches for comparing experiments and simulations

This section describes the approaches adopted in this study for comparing simulation results with the physical experimental data. Consider Dooge [56] who wrote that, "the term model is used to describe a system which is simpler than the prototype system and which can reproduce some but not all of the characteristics thereof. Accordingly, a model is related to those particular aspects of the behavior of the prototype for which understanding or prediction is required". Following this reasoning, the authors would like to emphasize that our objective is not to exactly reproduce the experimental results of [18] – nor is this even a feasible task. Our goal is to instead ensure that the model is able to capture the general patterns and behavior observed in the experimental data.

The sparsity of these datasets prevents a reliable reconstruction of the flow field that can be compared quantitatively with the three-dimensional simulation results. To address this issue, simulation data were extracted at the same locations as the experimentally measured vertical profiles for direct quantitative comparison of the results. Therefore, while not as informative as a point-by-point comparison, the qualitative comparison of the simulated flow field with interpolated experimental data in 2D is still important for ensuring that the model is able to capture general patterns and behavior.

As previously discussed, relative humidity and flow statistics were measured with accuracies of ± 0.03 and 5%, respectively. It should be noted that there are a large number of variables at play with respect to the experimental setup and methodologies employed by [18]. Collectively, these are not accounted for in the model and are the likely explanation for any of the observed differences between the experimental and simulation results. A summary of possible factors contributing to experimental uncertainty include:

- The low sampling frequency of relative humidity measurements leads to a smearing effect due to spatio-temporal averaging of the vapor transport in the streamwise direction.
- The reliance on a sensor immersed in the flow to measure relative humidity is furthermore invasive; its presence disturbs the flow field locally and therefore the relative humidity distribution; the impact of the sensor was not quantified by [18].
- The climate controls (i.e., heater, chiller, humidifier, dehumidifier) continuously fluctuated during the experiments, typically by no more than temperature ± 1 °C, relative humidity ± 3 %, and velocity ± 0.05 m s^{-1}. This can lead to momentary increases or decreases in any of the aforementioned atmospheric variables; this variability was accounted for in the model.
- Uncertainty in the physical placement of the bluff bodies within the test-section, see the section on the experimental setup. The model and physical experiments may therefore differ slightly. The measurements did not explore variability in the transverse direction (i.e., y-axis).
- Uncertainty in the physical locations where the flow and relative humidity measurements were made due to the accuracy of the automated traversing systems. If the laser was not perfectly centered behind a synthetic plant for example, it would measure slightly different flow behavior than it would otherwise. See the experimental setup section for more details.

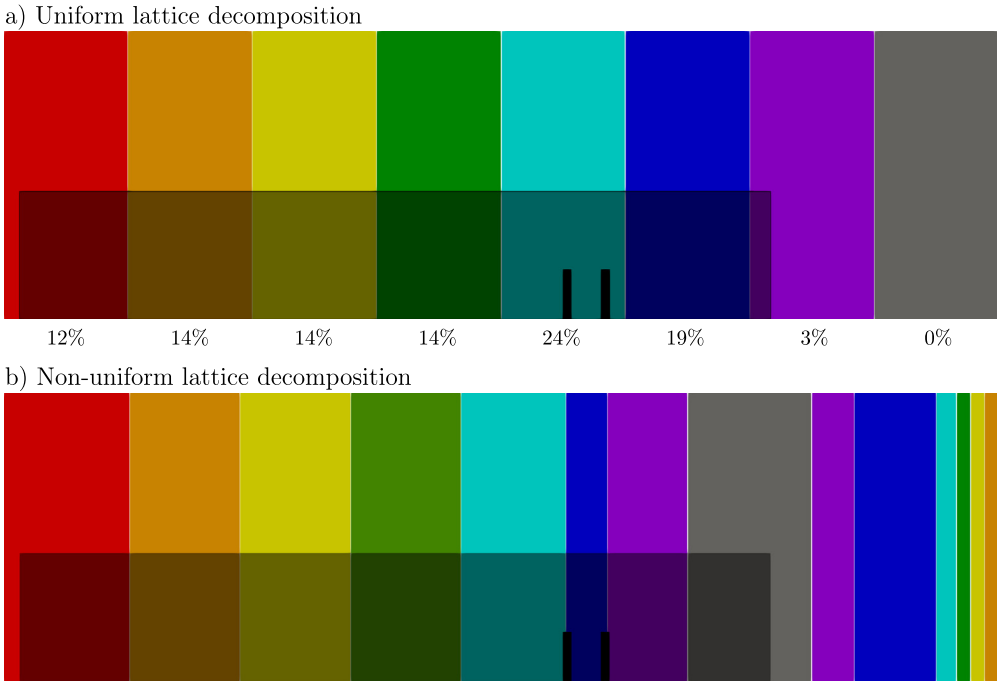


Fig. 3. Domain decompositions of a regular lattice (rainbow-colored subdomains) overlapped with an unstructured mesh (dark transparent rectangle) that is refined around the synthetic plants (two small black rectangles). The percentages below the case a) indicate the portion of the total number of mesh cells included in the corresponding lattice subdomain. All lattice subdomains in the case b) include 1/8 of the total number of mesh cells.

Table 4
Characteristics of each resolution used for presented simulations. The computational times were achieved using 8 Nvidia Tesla A-100 cards with NVLink interconnection. The total computational time is broken down to cumulative contributions from LBM computation, velocity interpolation, and MHFEM computation; the remaining time includes initialization and output of the data.

	RES-1	RES-2	RES-3
Lattice space step	7.88 mm	3.94 mm	1.97 mm
Lattice dimensions	$496 \times 96 \times 144$	$991 \times 224 \times 288$	$1981 \times 480 \times 575$
No. of lattice sites	approx. $7 \cdot 10^6$	approx. $64 \cdot 10^6$	approx. $547 \cdot 10^6$
No. of mesh cells	approx. $1.5 \cdot 10^6$	approx. $12 \cdot 10^6$	approx. $96 \cdot 10^6$
Total memory	2.5 GiB	24 GiB	200 GiB
Base time step Δt	1.33×10^{-3} s	3.33×10^{-4} s	8.32×10^{-5} s
Average no. of LBM iters per MHFEM step ($\lfloor C_{\max}/C \rfloor$)	1	2	4
Computational time	10 min	65 min	15 h 12 min
– LBM computation	1 min	11 min	6 h 8 min
– velocity interpolation	46 s	2 min	30 min
– MHFEM computation	4 min	30 min	7 h 54 min

- Uncertainty in the exact angle of the laser beams. If the beams were not exactly perpendicular to the flow, there could be some resulting skewness in the measured flow properties.

4.3. Qualitative comparison via 2D flow fields

Upon investigation of the velocity profiles from simulations using the time-constant (i.e., steady) inflow velocity profile given by Eq. (10), we noticed that vortical structures induced by the flow around obstacles were different immediately behind the first and second synthetic plants in the configurations EX-2 and EX-3 with 45 cm and 105 cm spacing, respectively. The difference is caused by the fact that when steady flow is prescribed at the inflow boundary, the flow immediately before the first plant is still steady, whereas the flow reaching the second plant is already turbulent. This scenario is non-physical, since natural air flow measured in the experiments was turbulent ($\mathrm{Re} > 10^4$). In order to introduce additional realism to the simulations and to address the above issue, the inflow velocity profile was modified through the inclusion of synthetic fluctuations based on Section 3.1.2. The effect of small perturbations on the inflow boundary is that they enhance the development of the turbulent boundary layer in simulations with limited spatial resolution and domain size. A qualitative comparison of the high-resolution simulated mean horizontal velocity (v_x) fields using both steady (time-constant) and unsteady (time-varying) boundary conditions are presented for the configurations EX-2 and EX-3 in Fig. 4. Vortical structures caused by recirculating flow are observed downstream of both synthetic plants regardless of the applied boundary condition, but the overall size of this recirculating region is affected by the boundary condition. The steady inflow velocity boundary condition leads to the development of a region with low velocity downstream of the first plant in the flow direction that is larger than that observed when the time-varying boundary condition is applied. When these results are compared with the experimental data (i.e., the vertical columns in Fig. 4), it is clear that the time-varying boundary condition captures the observed behavior better. Hence, only the time-varying inflow boundary condition is considered for the results discussed hereafter.

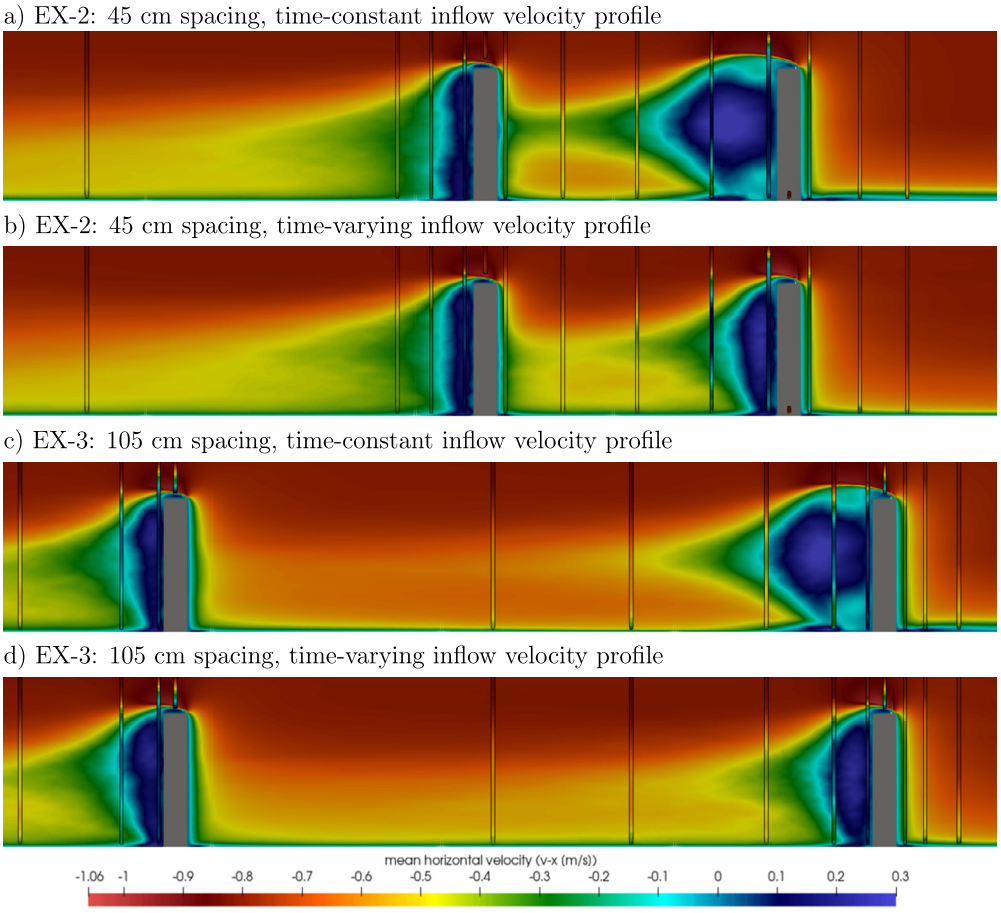


Fig. 4. Simulated mean horizontal velocity (v_x) fields along the plane $y = 0$ with overlain 1D velocity profiles for the configurations EX-2 (45 cm spacing) and EX-3 (105 cm spacing). The velocity profile prescribed at the inflow boundary was either time-constant (subfigures a and c) or time-varying (subfigures b and d). Only a small region of interest around the synthetic plants is shown to improve visibility. The background color corresponds to the high-resolution simulation (RES-3); vertical columns show experimental data.

The profiles of mean horizontal and vertical velocity (v_x and v_z), root-mean-square of the turbulent horizontal and vertical velocity (RMS_x and RMS_z), and mean relative humidity (ϕ) along the plane $y = 0$ are shown in Fig. 5 for EX-1, Fig. 6 for EX-2, and Fig. 7 for EX-3. In all three cases, the background color corresponds to the high-resolution simulation (RES-3) and the narrow vertical columns highlight superimposed experimental data measured at the corresponding locations.

The flow field observed around the synthetic plants depends on the spacing between the roughness elements [57]:

- In the closest spacing configuration (EX-1, Fig. 5), the downstream synthetic plant is located within the turbulent wake of the upstream plant. In this scenario, stable vortices are formed between the plants and the regime is termed skimming flow.
- In the intermediate spacing configuration (EX-2, Fig. 6), the wake created by the upstream plant does not fully develop before the flow reaches the downstream plant and affects the flow around it. Thus, this scenario is termed wake interference flow.
- In the widest spacing configuration (EX-3, Fig. 7), the synthetic plants are spaced sufficiently far apart so as to allow for full development of individual wake zones. This flow regime is termed isolated roughness flow.

These flow regimes can be clearly discerned in the simulated flow fields shown in Figs. 5 to 7. The skimming flow and the isolated roughness flow regimes seem to be captured well by the high-resolution simulation (RES-3). As discussed at the beginning of this section, the horizontal position of the wakes in the wake interference and isolated roughness flow regimes is affected by the inflow boundary condition used in the simulation. While the width of the simulated upstream wake in the isolated roughness flow regime (Fig. 7) is in good agreement with the experiment, it is underestimated in the wake interference flow regime (Fig. 6). Aside from this disagreement, the wakes formed behind the downstream plant as well as flow profiles far away from the plants are qualitatively in good agreement with the experiments in all three scenarios.

In the case of relative humidity, the greatest differences between experiments and simulations occur in the vertical profiles measured immediately downstream of the synthetic plants (i.e., on the left side of the gray rectangles in the figures). In the following section, we will quantify the differences in the profiles immediately downstream of the first plant and compare them with the measurements accuracy.

4.4. Quantitative comparison via 1D graphs

In this section, the simulation results and experimental data are compared quantitatively in the vertical columns immediately downstream of the first synthetic plant; the comparisons in the other measurement locations are included in the supplementary materials. The graphs of the mean velocity (v_x and v_z) profiles are shown in Figs. 8 to 10 and the graphs of the relative humidity (ϕ) profiles are shown in Figs. 11 to 13. For convenience, the horizontal position of each profile is indicated schematically by the red bar above each graph on the right hand side. In each graph, the experimental data (red line) and simulation data in three resolutions (dark blue line for RES-1, light blue line for RES-2, orange line for RES-3) are compared. The dashed lines indicate the sta-

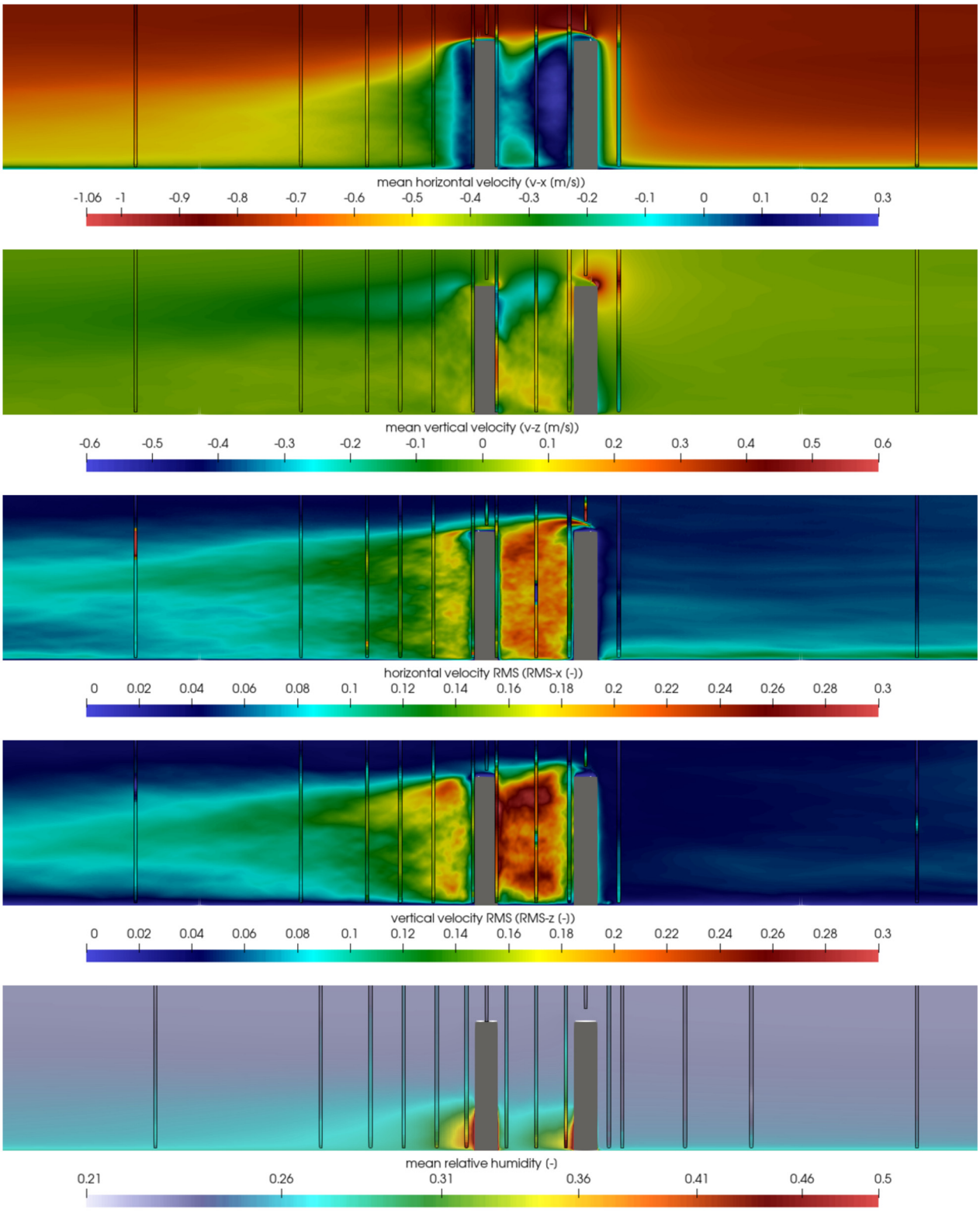


Fig. 5. Simulated flow fields along the plane $y = 0$ with overlain 1D profiles: mean horizontal and vertical velocity (v_x and v_z), root-mean-square of horizontal and vertical velocity (RMS_x and RMS_z), and mean relative humidity (ϕ) for the configuration with 15 cm spacing between synthetic plants (EX-1). Only a small region of interest around the synthetic plants is shown to improve visibility. The background color corresponds to the high-resolution simulation (RES-3); vertical columns show experimental data.

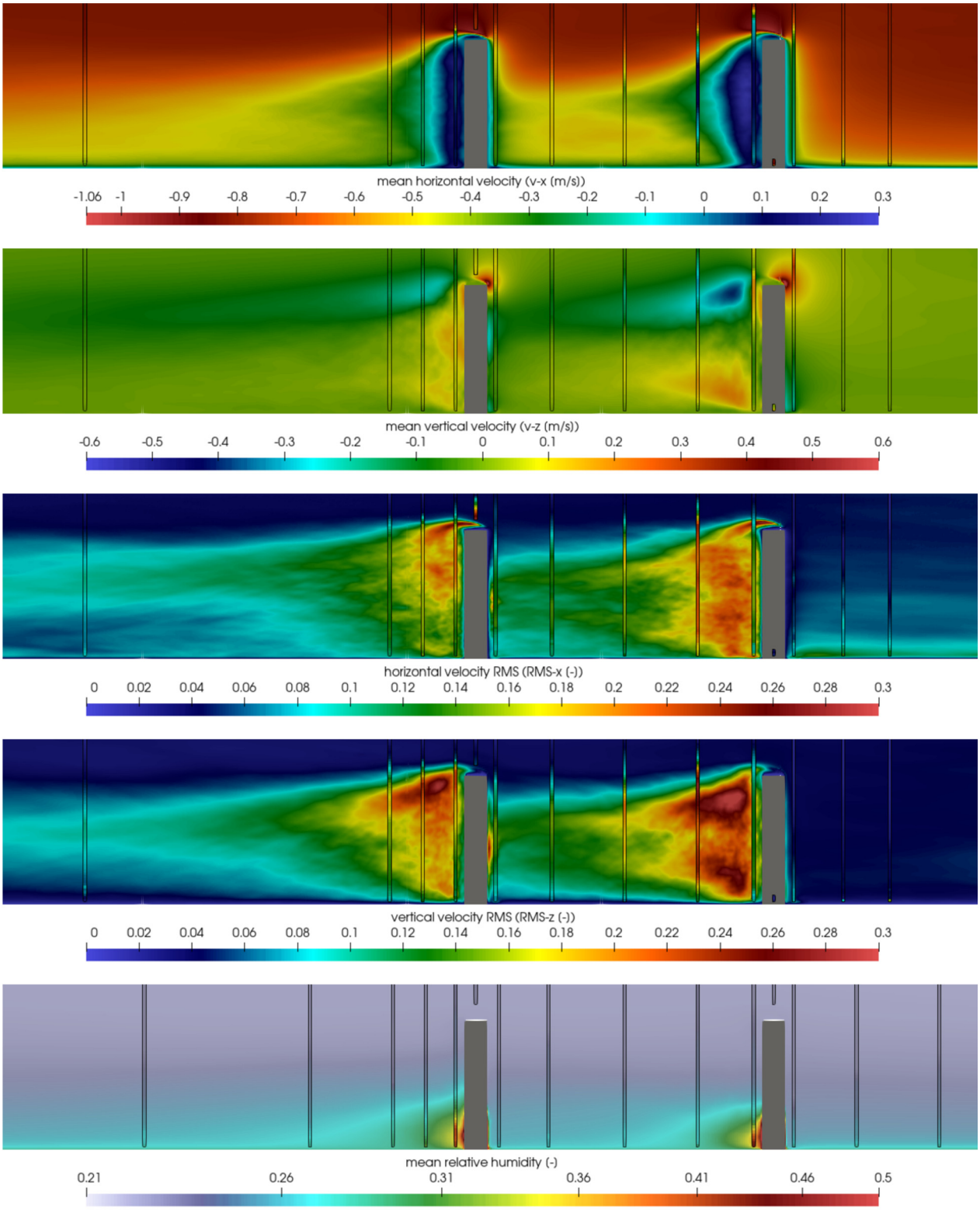


Fig. 6. Simulated flow fields along the plane $y = 0$ with overlain 1D profiles: mean horizontal and vertical velocity (v_x and v_z), root-mean-square of horizontal and vertical velocity (RMS_x and RMS_z), and mean relative humidity (ϕ) for the configuration with 45 cm spacing between synthetic plants (EX-2). Only a small region of interest around the synthetic plants is shown to improve visibility. The background color corresponds to the high-resolution simulation (RES-3); vertical columns show experimental data.

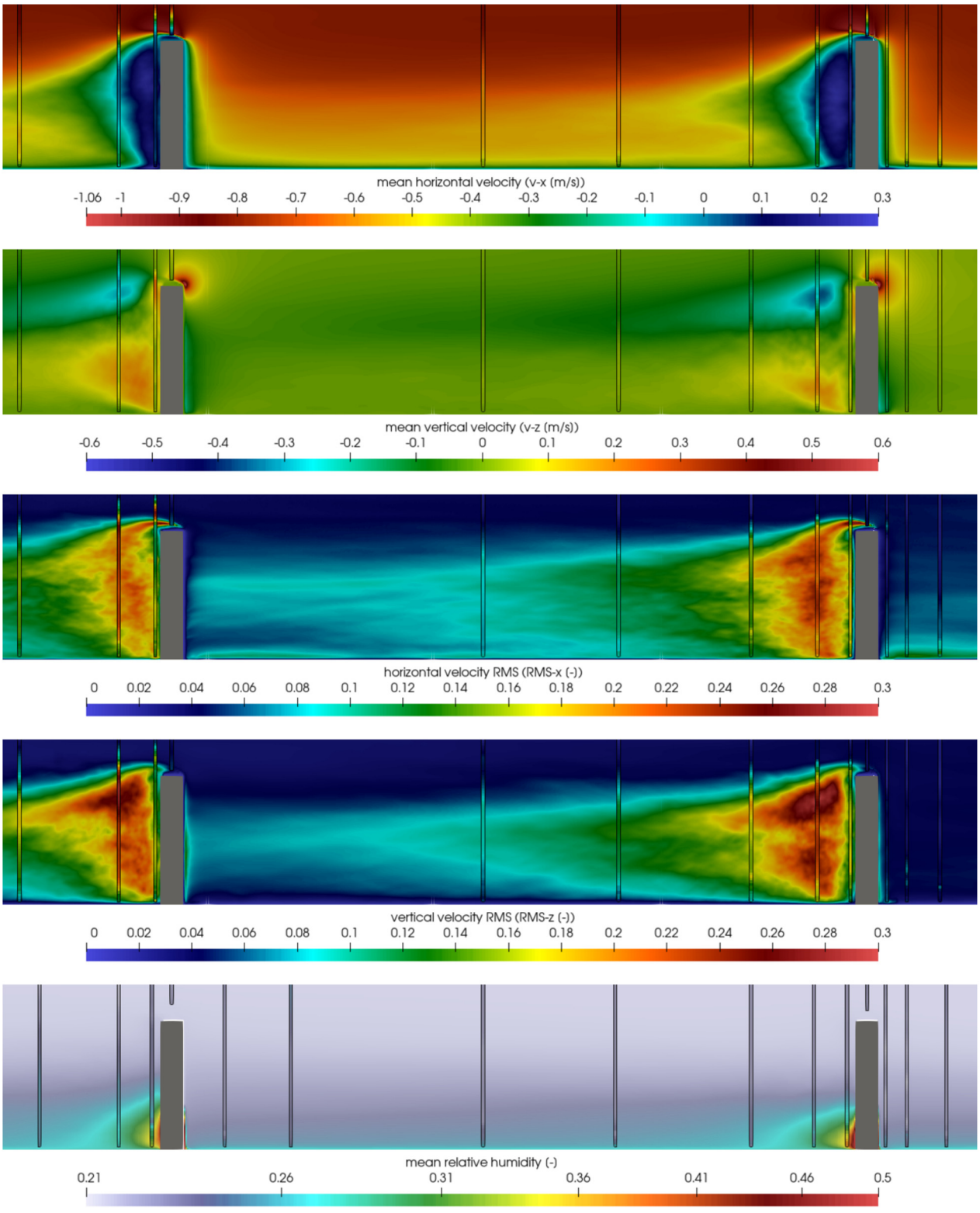


Fig. 7. Simulated flow fields along the plane $y = 0$ with overlain 1D profiles: mean horizontal and vertical velocity (v_x and v_z), root-mean-square of horizontal and vertical velocity (RMS_x and RMS_z), and mean relative humidity (ϕ) for the configuration with 105 cm spacing between synthetic plants (EX-3). Only a small region of interest around the synthetic plants is shown to improve visibility. The background color corresponds to the high-resolution simulation (RES-3); vertical columns show experimental data.

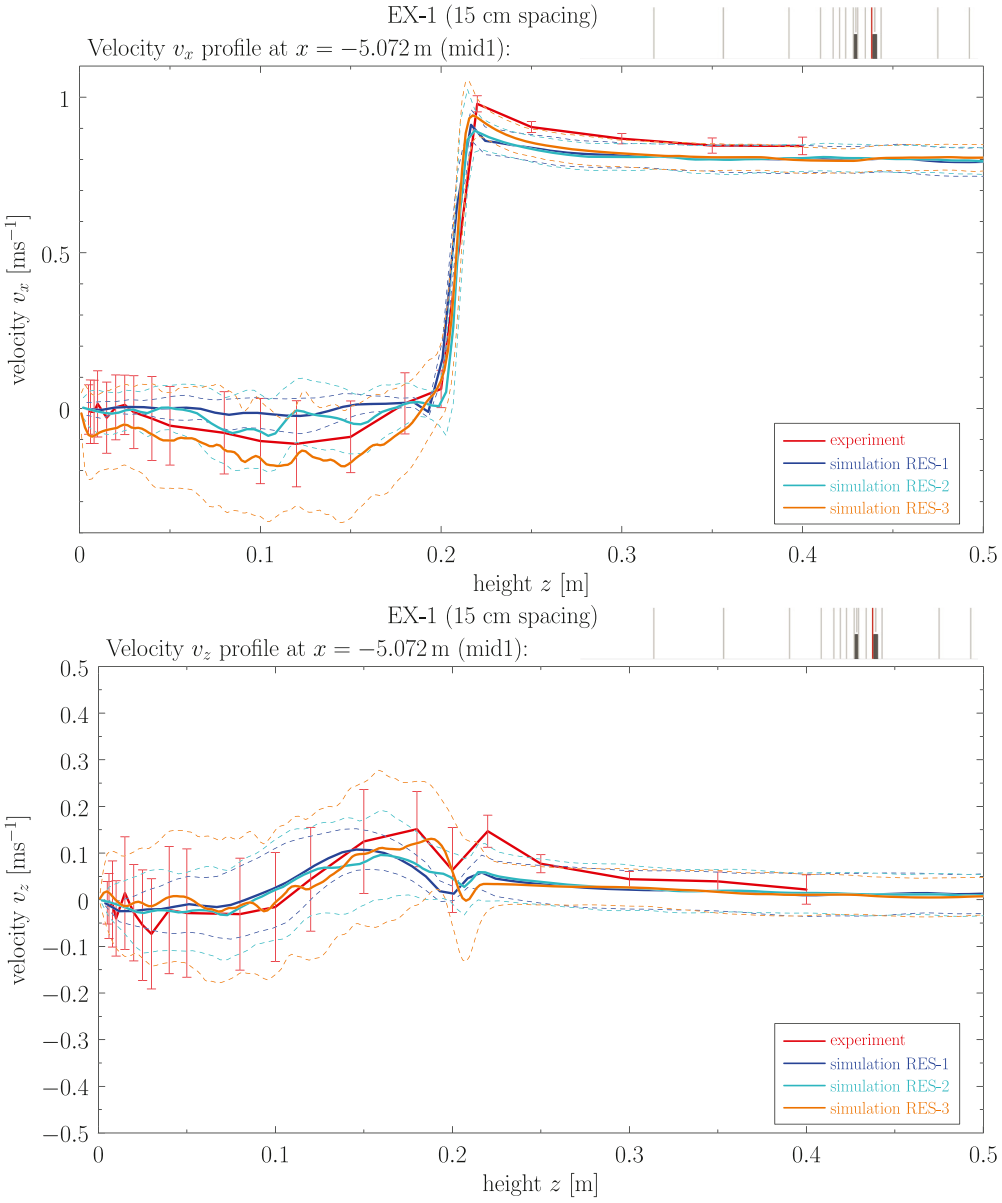


Fig. 8. Quantitative comparison of horizontal and vertical velocity profiles (v_x and v_z) at the first position downstream of the first synthetic plant in the 15 cm spacing (EX-1). The red bar above each graph highlights the position of the profile relative to the synthetic plants (dark rectangles) and other measurement locations (thin gray bars).

tistical deviation of the respective simulated quantity from its mean value (full lines). In case of the measured velocity profiles in Figs. 8 to 10, the statistical deviation is indicated by errorbars. On the other hand, the low sampling frequency of the RHT sensors prevents any information on the turbulent transport of the water vapor from being inferred from the data. In this case, the errorbars in Figs. 11 to 13 represent the sensor accuracy rather than statistical deviation of the samples.

The graphs in Figs. 8 and 10 for the cases EX-1 and EX-3, respectively, show the best match between the simulated velocity profiles and experimental data. It can be observed that of the three resolutions compared in the graphs, the low-resolution simulation RES-1 differs the most from the experimental data and the high-resolution simulation RES-3 is closest to the experimental data. For the remaining case EX-2, the graphs in Fig. 9 show larger differences that correspond to the qualitative disagreement described in the previous section. The primary cause is presumably the uncertainty related to the exact conditions in the experiment. Upon closer examination of the documentation from this experiment, we found that the upstream plant may have been planted not perfectly perpendicular to the ground surface, which would result in different flow behavior around and above the plant. However, there is no data that would allow us to reproduce this scenario in the simulation.

In case of the relative humidity profiles, the largest difference between the experimental data and simulations is observed in Fig. 11 for EX-1 where the simulated profiles are underestimated. These results suggest that improvements could be made by modifying the boundary condition applied on the plant. The boundary condition described in Section 2.3 is based on a mass flux that was measured in a separate experiment under different atmospheric forcing conditions [19] than those applied herein; the applied mass fluxes may therefore underestimate the conditions actually present in the simulated scenarios. Furthermore, it should be noted that when the plants are located close together, the plants may compete for the limited resource [18] which might result in increased collective evaporation rate compared to the cases where the plants are located far away from each other. In the cases EX-2 and EX-3 featuring different flow regimes, the graphs in Figs. 12

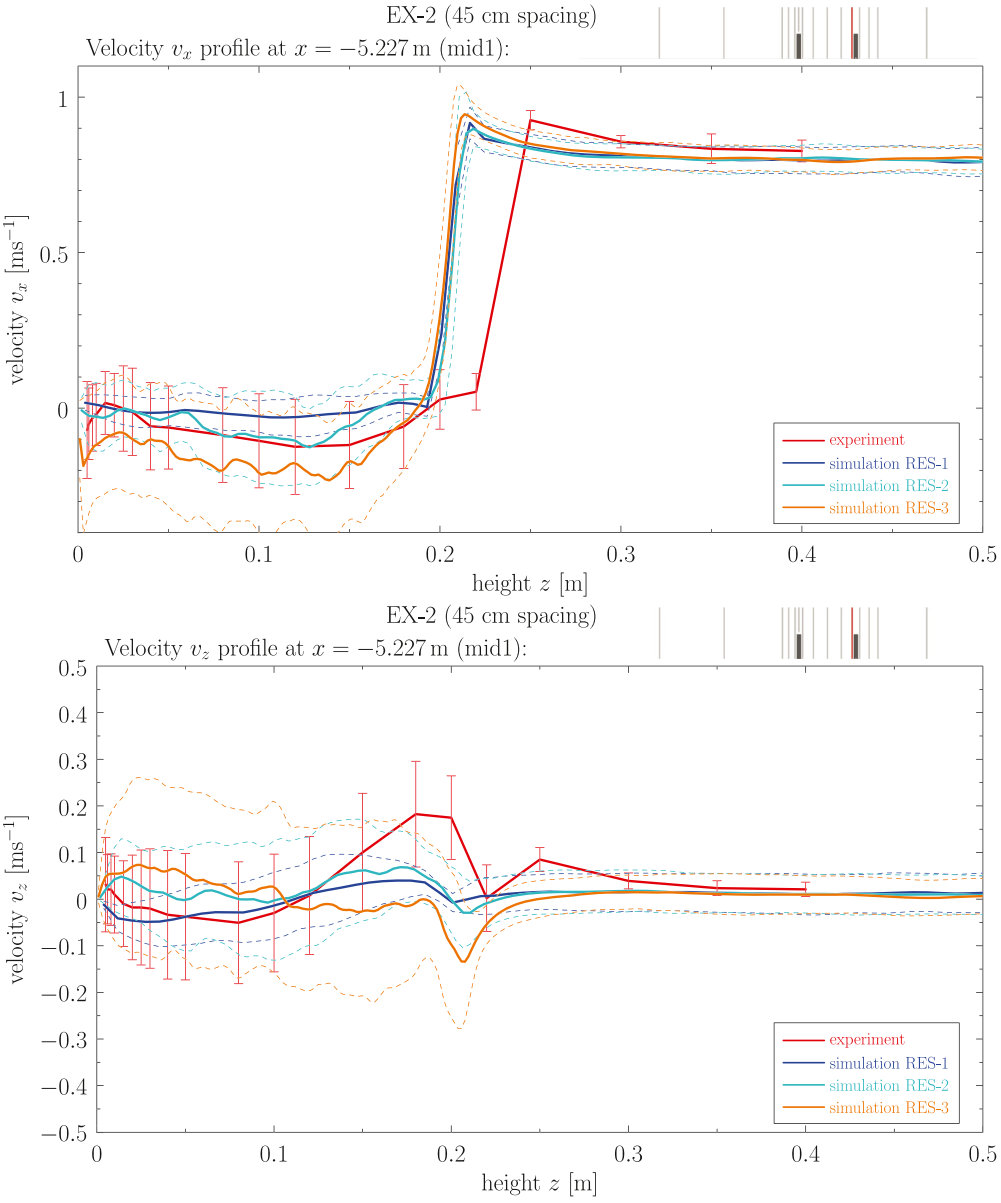


Fig. 9. Quantitative comparison of horizontal and vertical velocity profiles (v_x and v_z) at the first position downstream of the first synthetic plant in the 45 cm spacing (EX-2). The red bar above each graph highlights the position of the profile relative to the synthetic plants (dark rectangles) and other measurement locations (thin gray bars).

and 13 indicate that the simulation results are well within the accuracy of the RHT sensors.

4.5. Computational performance analysis

To demonstrate the computational efficiency of the implemented LBM-MHFEM solver, we performed a strong scaling study whose results are shown in Table 5 for the experimental configuration EX-1 in the resolution RES-2. Due to limitations of the computational system, the performance scaling analysis could be performed only with 8 GPUs. Because of the coupling with MHFEM, the overall performance of the solver in GLUPS (giga-LUPS, *billions of lattice updates per second*) is about 4-5× lower compared to a standalone LBM solver, depending on the adaptively selected time steps. The efficiency decreases with increasing the number of GPUs used in the computation, which is a typical behavior in strong scaling analyses caused by reduced work per GPU and increased communication-to-work ratio. Considering that our implementation is limited by the one-dimensional domain decomposition of the lattice, the 80% efficiency achieved on 8 GPUs is a satisfactory result. Higher efficiency can be expected for weak scaling studies where the amount of work is kept proportional to the number of GPUs. However, due to the adaptive time stepping strategy used in the coupled solver, it is not straightforward to analyze the weak scaling, because the performance of the solver depends on the number of time steps where MHFEM is executed, which would be different for each problem size.

5. Conclusion

In this paper, we presented an efficient computational method for vapor transport in the boundary layer above a partially saturated soil. The solver is based on the combination of lattice Boltzmann and mixed-hybrid finite element methods and can utilize modern GPU-based high-performance computing systems. The paper deals with mutual collaboration between experimental and computational methodologies.

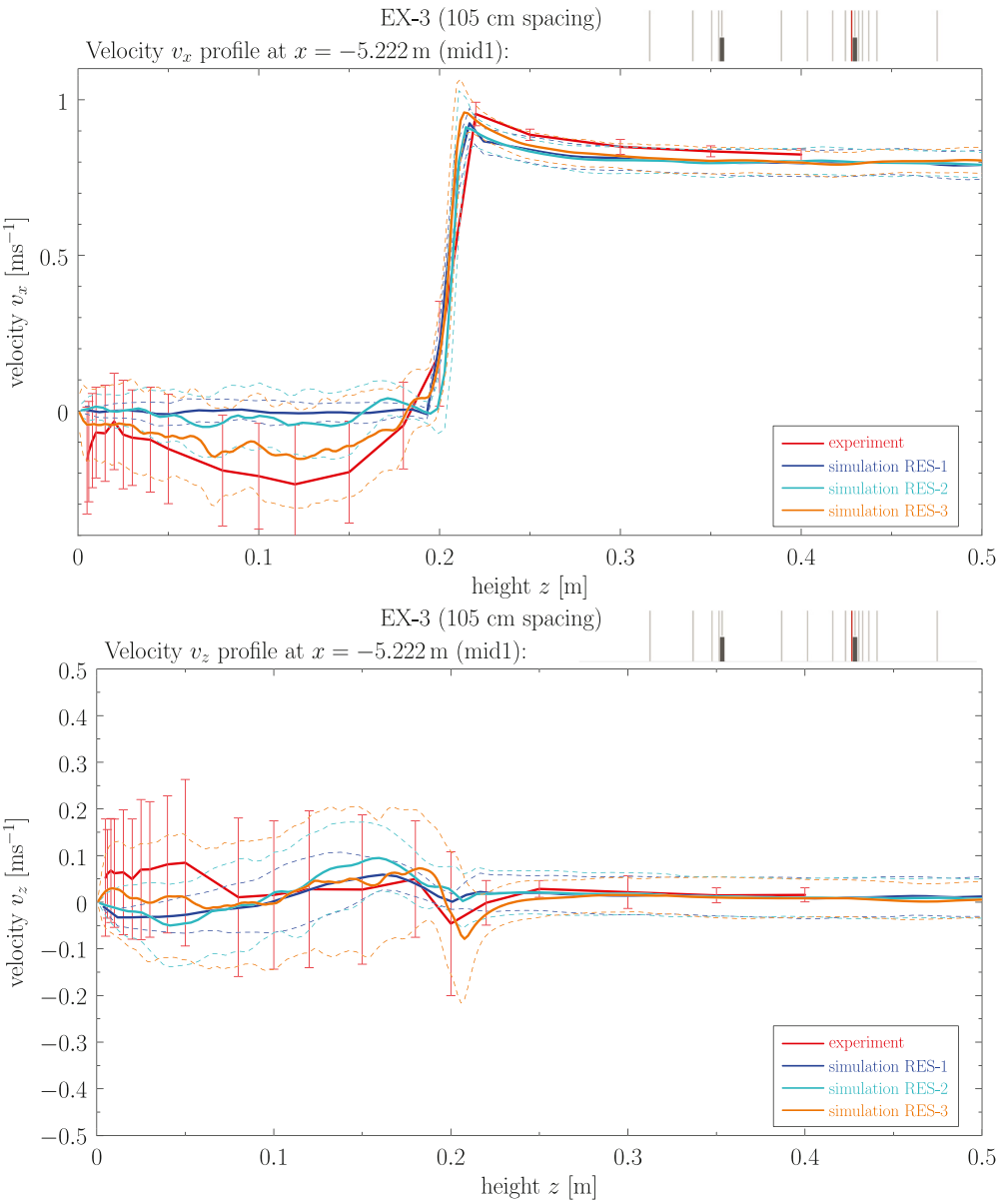


Fig. 10. Quantitative comparison of horizontal and vertical velocity profiles (v_x and v_z) at the first position downstream of the first synthetic plant in the 105 cm spacing (EX-3). The red bar above each graph highlights the position of the profile relative to the synthetic plants (dark rectangles) and other measurement locations (thin gray bars).

Table 5
Strong scaling of the coupled LBM-MHFEM solver for the scenario EX-1 in the resolution RES-2. N_{GPUs} denotes the number of Nvidia Tesla A-100 GPUs used in the computation, the Time column includes the computational time without initialization, the performance metric GLUPS stands for *billions of lattice updates per second* and Eff denotes the parallel efficiency.

N_{GPUs}	Time [min]	GLUPS	Eff
1	392	1.0	1.00
2	202	1.9	0.96
4	110	3.7	0.92
8	62	6.4	0.80

The model was validated with experimental data measured above a flat partially saturated soil layer featuring synthetic plants arranged in several configurations. The model relies on experimental data for the input for boundary conditions: the inflow velocity and humidity profiles and the average mass flux of water loss from the plants; experimental data used in this study were generated by [19,18] and are publicly available in [20].

Based on the validation study presented in this paper, we can draw reasonable predictions about the flow and transport behavior inside the computational domain. The performance of the coupled solver depends on the selected lattice and mesh sizes (i.e., spatial resolution) and the adaptively selected time steps. The highest-resolution simulations presented in this paper, which compare the best to the experimental data, require about 200 GiB memory and 15.25 h computational time on 8 Nvidia Tesla A-100 cards to simulate 100 s of physical time. The simulations in lower resolutions are not as accurate, but require less memory and shorter computational time compared to the highest resolution. A strong scaling analysis was performed for a lower resolution giving a parallel efficiency of 80% on 8 Nvidia Tesla A-100 cards. Scalability problems that are likely to occur on large-scale supercomputers (e.g., due to one-dimensional decomposition of the domain) were not investigated here due to the availability of computational resources. The generalization of the domain decomposition procedure from Section 3.6

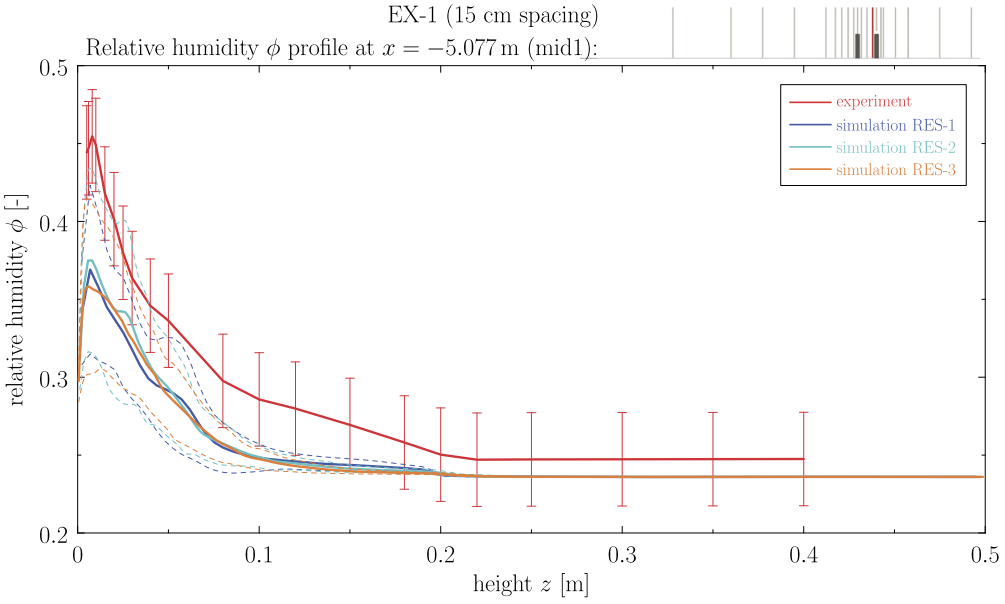


Fig. 11. Quantitative comparison of relative humidity profiles (ϕ) at the first position downstream of the first synthetic plant in the 15 cm spacing (EX-1). The red bar above the graph highlights the position of the profile relative to the synthetic plants (dark rectangles) and other measurement locations (thin gray bars).

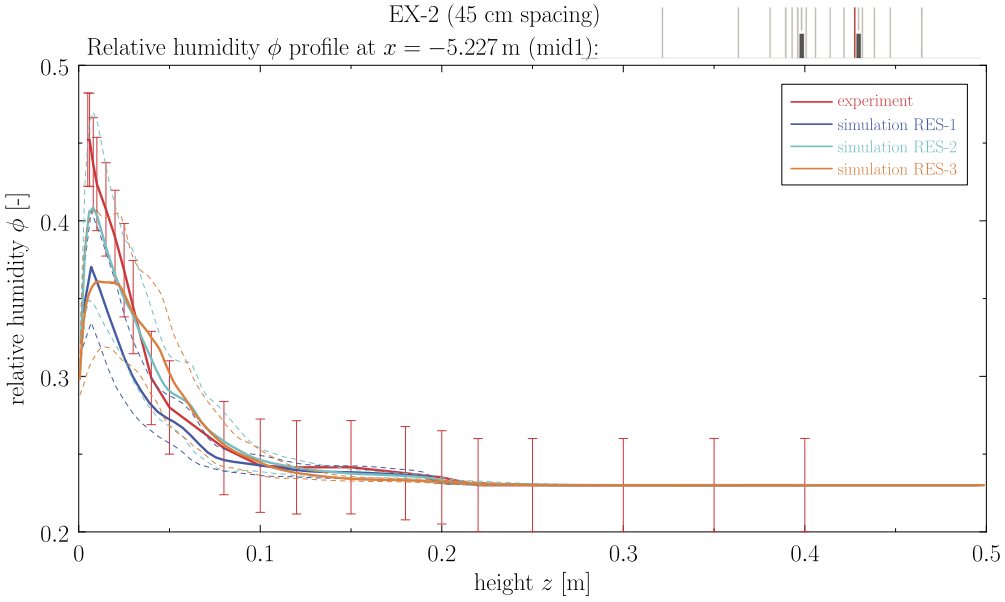


Fig. 12. Quantitative comparison of relative humidity profiles (ϕ) at the first position downstream of the first synthetic plant in the 45 cm spacing (EX-2). The red bar above the graph highlights the position of the profile relative to the synthetic plants (dark rectangles) and other measurement locations (thin gray bars).

to improve the scalability on large supercomputers may be subject of future research.

The results presented herein suggest several key areas where future experimental efforts could be improved, allowing the analysis of this model's performance to be extended and further explored. For example, extending measurements with flow characteristics in the transverse direction (e.g., v_y, RMS_y, $\overline{v'_x v'_y}$, $\overline{v'_y v'_z}$) would allow us to compare the turbulent kinetic energy and improve the fluctuating inflow velocity condition for the simulations. Another possible improvement is to arrange measurements in horizontal profiles in regions behind the plants, which would allow us to study the convergence of the numerical method (i.e., the effect of mesh resolution) by comparing the horizontal location of the vortical structures. Last but not least, the applicability of the measured evaporative mass flux to the close spacing scenario EX-1 should be investigated. Improving the methodology for measuring the evaporation from the plants would allow for prescribing more appropriate boundary conditions.

The work presented in this paper explores new possibilities in the efficient solution of various multiphysics problems using modern hardware architectures. The developed model is based on the combination of LBM for fluid flow and MHFEM for a general system of advection-diffusion-reaction PDEs. The simulator for vapor transport in air is just a first application that could be extended into a more general software tool capable of solving other physical phenomena such as non-isothermal flow, multicomponent flow, land-atmospheric interaction, etc. There are many potential applications in combination with the experimental research, such as developing an efficient tool for a sensitivity analysis of measurements, supplementing sparse experimental datasets in regions where measurements would be too expensive or unfeasible, or predicting the behavior of the studied system in virtual scenarios.

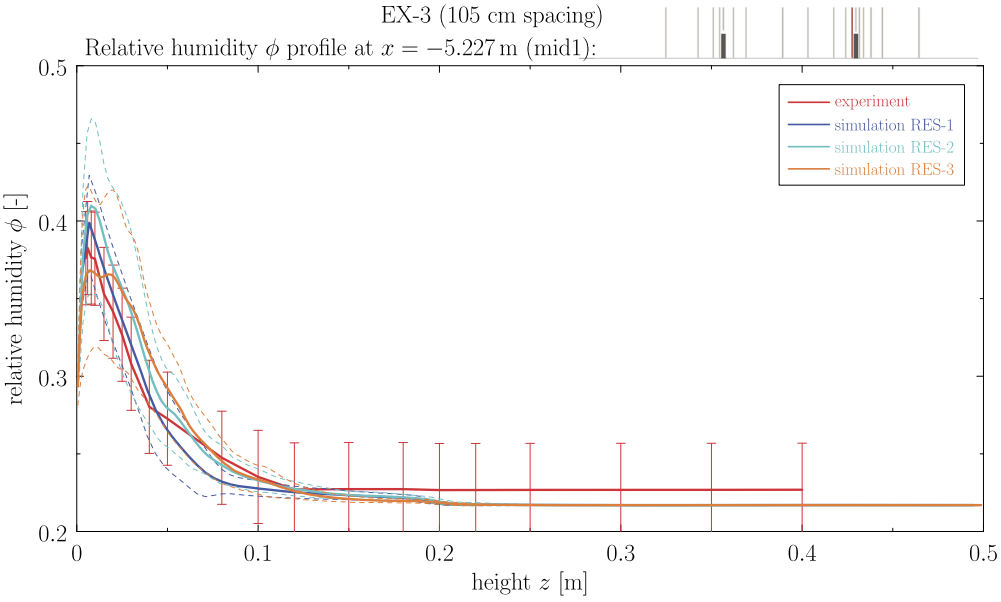


Fig. 13. Quantitative comparison of relative humidity profiles (ϕ) at the first position downstream of the first synthetic plant in the 105 cm spacing (EX-3). The red bar above the graph highlights the position of the profile relative to the synthetic plants (dark rectangles) and other measurement locations (thin gray bars).

Table A.6
Characteristics of lattice and grid resolutions used for the numerical analysis.

	RES-A1	RES-A2	RES-A3
Lattice space step	8.06 mm	3.97 mm	1.97 mm
Lattice dimensions	$217 \times 128 \times 128$	$441 \times 256 \times 256$	$889 \times 512 \times 512$
MHFEM grid dimensions	$128 \times 64 \times 64$	$256 \times 128 \times 128$	$512 \times 256 \times 256$
No. of lattice sites	approx. 3.5×10^6	approx. 29×10^6	approx. 233×10^6
No. of grid cells	approx. 0.5×10^6	approx. 4×10^6	approx. 33×10^6
Base time step Δt	1.39×10^{-3} s	3.38×10^{-4} s	8.32×10^{-5} s
Average no. of LBM iters per MHFEM step ($\lfloor C_{\max}/C \rfloor$)	2	4	9

Data availability

The experimental data used for validation are available as a public dataset https://doi.org/10.17632/6fryw4xzgh.1.

Acknowledgement

The work was supported by the Ministry of Education, Youth and Sports of the Czech Republic under the Inter-Excellence grant no. LTAUSA19021 and the OP RDE grant no. CZ.02.1.01/0.0/0.0/16_019/0000765: *Research Center for Informatics*, the Czech Science Foundation project no. 21-09093S, and the Student Grant Agency of the Czech Technical University in Prague project no. SGS20/184/OHK4/3T/14.

The experimental portion of this study was conducted at the Center for Experimental Study of Subsurface Processes (CESEP) at the Colorado School of Mines in Golden, Colorado, USA. The wind tunnel in which the experiments were performed is now located at the Synthetic Environment for Near-Surface Sensing and Experimentation (SENSE) Research Facility located at the US Army Corps of Engineers (USACE) Engineer Research and Development Center (ERDC) Waterways Experimental Station (WES) in Vicksburg, Mississippi, USA. Researchers with questions regarding the SENSE Research Facility and its potential use should contact the second author.

Appendix A. Numerical analysis of the conservative vs. non-conservative formulation

In this section, we study numerically the convergence of the coupled LBM-MHFEM scheme using an artificial benchmark problem. The aim of this section is to study the differences between the conservative and non-conservative formulations of the transport equation.

Equation (1) governing the fluid flow is solved in a cuboidal channel $\Omega_1 = [0, 1.75] \times [0, 1] \times [0, 1]$ (dimensions are in meters) with parameters similar to the main problem discussed in the paper (kinematic viscosity $\nu = 15.52 \times 10^{-6}\,\mathrm{m^2\,s^{-1}}$, mean inflow velocity magnitude $v_{\max} = 1\,\mathrm{m\,s^{-1}}$). Note that the channel is free of all obstacles, but we induce turbulent flow using the unsteady (time-varying) inflow boundary condition described in Section 3.1.2.

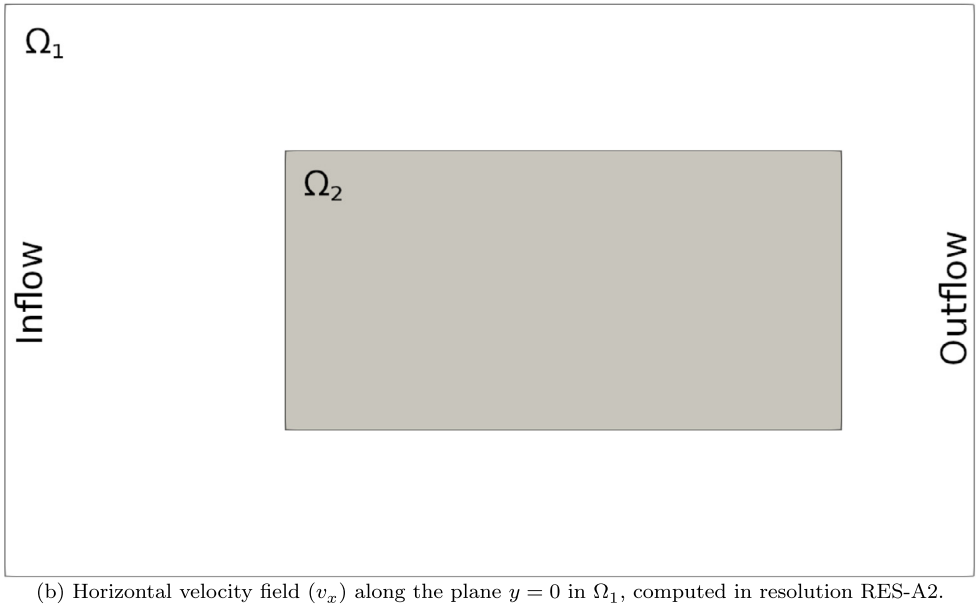
The fluid flow is coupled with a transport equation either in the conservative form Eq. (6a) or non-conservative form Eq. (6b), where ϕ [-] is exempt from its physical meaning and for the purpose of this section, it is interpreted as the concentration of a generic constituent transported by the fluid. The diffusion coefficient $D = 25.52 \times 10^{-6}\,\mathrm{m^2\,s^{-1}}$ is set the same as in the main problem discussed in the paper (see Table 2). The transport equation is solved in domain $\Omega_2 = [0.5, 1.5] \times [0.25, 0.75] \times [0.25, 0.75]$ (in meters) that is completely immersed in the domain Ω_1 (i.e., none of the domain boundaries coincide: $\partial\Omega_1 \cap \partial\Omega_2 = \emptyset$). See Fig. A.14a for schematic configuration of the domains.

In order to study the differences between the conservative and non-conservative formulations, the initial and boundary conditions for the transport equation are posed as follows. Initially, we set $\phi = 1$ uniformly

Table A.7
Results of the numerical analysis for different formulations and variants of the MHFEM scheme.

Interp.	Upwind	Resolution	Conservative		Non-conservative	
			$\|\phi - \phi_h\|_1$	$\|\phi - \phi_h\|_2$	$\|\phi - \phi_h\|_1$	$\|\phi - \phi_h\|_2$
linear	explicit	RES-A1	4.01×10^{-3}	1.11×10^{-2}	1.21×10^{-14}	3.70×10^{-13}
		RES-A2	2.01×10^{-3}	5.71×10^{-3}	5.05×10^{-15}	2.85×10^{-13}
		RES-A3	7.82×10^{-4}	2.28×10^{-3}	8.00×10^{-15}	1.34×10^{-12}
	implicit	RES-A1	3.24×10^{-3}	8.95×10^{-3}	3.62×10^{-13}	2.40×10^{-12}
		RES-A2	1.62×10^{-3}	4.64×10^{-3}	5.80×10^{-14}	2.62×10^{-13}
		RES-A3	6.23×10^{-4}	1.82×10^{-3}	3.97×10^{-14}	2.19×10^{-13}
cubic	explicit	RES-A1	3.25×10^{-3}	8.75×10^{-3}	1.19×10^{-14}	3.56×10^{-13}
		RES-A2	1.31×10^{-3}	3.63×10^{-3}	7.44×10^{-15}	5.01×10^{-13}
		RES-A3	3.98×10^{-4}	1.09×10^{-3}	8.68×10^{-15}	1.45×10^{-12}
	implicit	RES-A1	2.63×10^{-3}	7.08×10^{-3}	5.28×10^{-13}	2.46×10^{-12}
		RES-A2	1.07×10^{-3}	2.96×10^{-3}	5.73×10^{-14}	2.50×10^{-13}
		RES-A3	3.24×10^{-4}	8.83×10^{-4}	3.37×10^{-14}	1.91×10^{-13}

(a) Illustration of the computational domains Ω_1 and Ω_2 (2D cross-section along the plane $y = 0$).



(b) Horizontal velocity field (v_x) along the plane $y = 0$ in Ω_1, computed in resolution RES-A2.

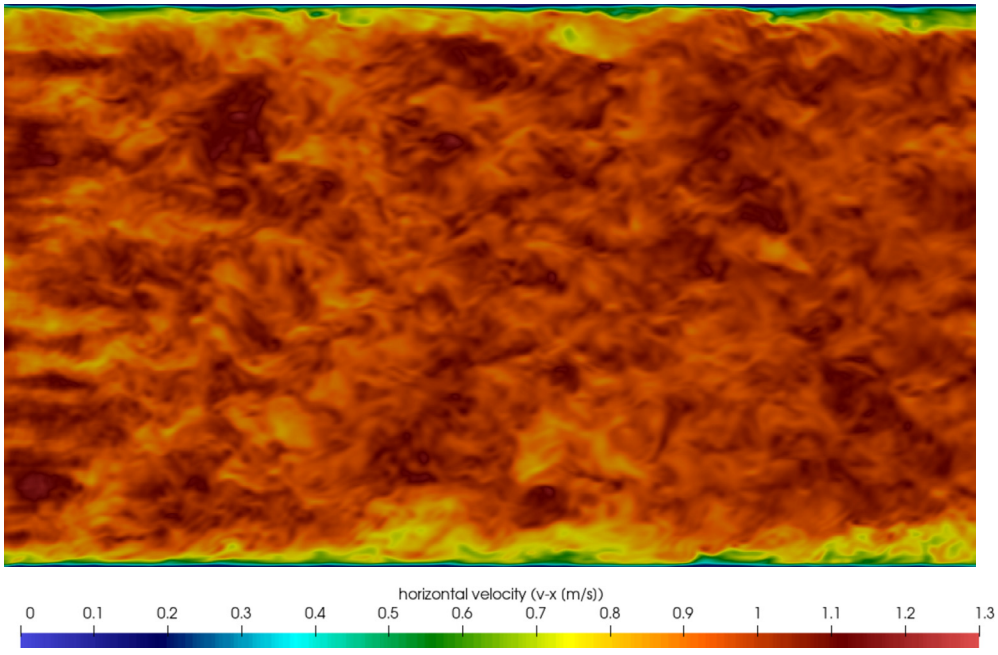


Fig. A.14. Schematic configuration of the computational domains in the benchmark problem and the resulting flow field computed in Ω_1.

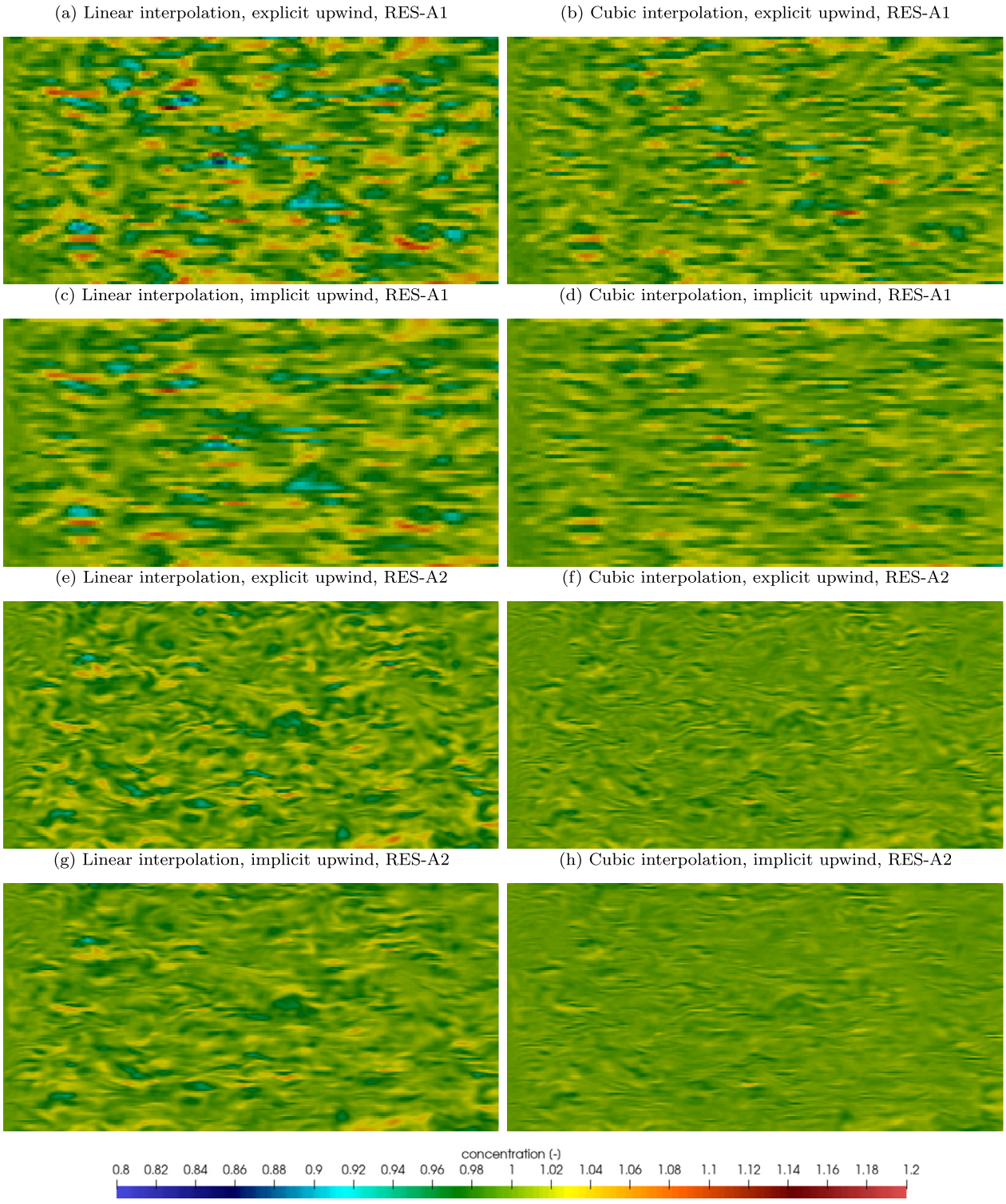


Fig. A.15. Simulated concentration field (ϕ) along the plane $y = 0$ in Ω_2 in the benchmark problem using the *conservative formulation* of the transport equation (6a). Several configurations of the numerical scheme are compared: linear and cubic interpolation of the velocity from LBM to MHFEM, and discretization of the advection term in the MHFEM scheme based on explicit and implicit upwind. Only the first two resolutions RES-A1 and RES-A2 are shown here.

in the whole domain Ω_2. On the inflow boundary ($x = 0.5$), we prescribe a fixed value $\phi = 1$. On all remaining parts of $\partial\Omega_2$, we prescribe a zero gradient in the normal direction ($\frac{\partial\phi}{\partial \boldsymbol{x}} \cdot \boldsymbol{n} = 0$). Given a divergence-free velocity field due to Eq. (1a), this initial-boundary-value problem has a trivial analytical solution $\phi(\boldsymbol{x}, t) = 1$ for all $\boldsymbol{x} \in \Omega_2$ and $t > 0$.

The coupled problem is solved numerically using the LBM-MHFEM scheme as described in Section 3. Several variants of the MHFEM scheme were used, namely explicit or implicit upwind, and linear or cubic interpolation of the velocity field. Each variant was computed in three resolutions denoted as RES-A1, RES-A2, and RES-A3, see Table A.6. To illustrate the turbulent flow field in Ω_1, Fig. A.14b shows the horizontal velocity (v_x) field in the final time $t_{\max} = 10$ s. Fig. A.15 shows qualitative differences between the concentration (ϕ) fields that were computed using different variants of the MHFEM scheme. Since the fields obtained using any variant with the non-conservative formulation were visually indistinguishable from the constant analytical solution on the scale used in Fig. A.15, only the conservative formulation variants are shown in the figure. Note that for given resolution, the velocity field is the same in all variants of the MHFEM scheme. Quantitative comparison is presented in Table A.7 in terms of L^p norms of the differences between the analytical solution $\phi = 1$ and each numerical solution ϕ_h.

Both qualitative and quantitative results in Fig. A.15 and Table A.7 indicate that for the conservative formulation, changing linear interpolation to cubic, as well as changing the explicit upwind discretization to implicit upwind, leads to smoother and more accurate results. Furthermore, all these variants converge to the analytical solution as the lattice and grid are refined. However, even the most accurate numerical solution obtained using the conservative formulation exhibits an error that is larger by orders of magnitude compared to the non-conservative formulation, even in the coarsest resolution. The only difference between the discretizations of the non-conservative and conservative formulations is in Eq. (16) where the former contains a term corresponding to the discrete divergence of velocity. The results indicate that this extra term can be understood as a compensation for the non-zero divergence of the discrete velocity field interpolated on the mesh. Furthermore, it can be noticed in Table A.7 that changing the interpolation or upwind scheme does not have a significant effect on the error when the non-conservative formulation is used. In the finest resolution RES-A3, using the linear interpolation and explicit upwind is not only advantageous for the performance of the solver, but also leads to a smaller error.

Appendix B. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.camwa.2023.02.021.

The supplementary materials for this paper include additional graphical results, including 2D flow fields simulated in all resolutions (RES-1, RES-2, RES-3) showing the effect of mesh resolution qualitatively, and 1D graphs comparing the velocity and relative humidity profiles quantitatively at all measurement locations.

References

[1] W. Bangerth, R. Hartmann, G. Kanschat, deal.II – a general purpose object oriented finite element library, ACM Trans. Math. Softw. 33 (4) (2007) 24/1–24/27, https://doi.org/10.1145/1268776.1268779.

[2] P. Bastian, M. Blatt, C. Engwer, A. Dedner, R. Klöfkorn, S.P. Kuttanikkad, M. Ohlberger, O. Sander, The distributed and unified numerics environment (DUNE), in: Proc. of the 19th Symposium on Simulation Technique, 2006.

[3] H. Jasak, A. Jemcov, Z. Tukovic, et al., OpenFOAM: a C++ library for complex physics simulations, in: International Workshop on Coupled Methods in Numerical Dynamics, vol. 1000, IUC, Dubrovnik, Croatia, 2007, pp. 1–20.

[4] K. Pruess, C.M. Oldenburg, G. Moridis, TOUGH2 user's guide version 2, Tech. Rep., Lawrence Berkeley National Lab. (LBNL), Berkeley, CA (United States), 1999.

[5] M. Syamlal, W. Rogers, T.J. O'Brien, MFIX documentation theory guide, Tech. Rep., USDOE Morgantown Energy Technology Center, WV (United States), 1993.

[6] ANSYS Inc., ANSYS Fluent 12.0 Documentation, ANSYS Inc., 2009, https://www.afs.enea.it/project/neptunius/docs/fluent/.

[7] COMSOL Multiphysics, Introduction to COMSOL Multiphysics®, COMSOL Multiphysics, Burlington, MA, 1998.

[8] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, E.M. Viggen, The Lattice Boltzmann Method, Springer, 2017.

[9] F. Brezzi, M. Fortin, Mixed and Hybrid Finite Elements Methods, Springer Series in Computational Mathematics, Springer-Verlag, 1991.

[10] R. Fučík, J. Klinkovský, J. Solovský, T. Oberhuber, J. Mikyška, Multidimensional mixed–hybrid finite element method for compositional two-phase flow in heterogeneous porous media and its parallel implementation on GPU, Comput. Phys. Commun. 238 (2019) 165–180, https://doi.org/10.1016/j.cpc.2018.12.004.

[11] F.A. Radu, N. Suciu, J. Hoffmann, A. Vogel, O. Kolditz, C.H. Park, S. Attinger, Accuracy of numerical simulations of contaminant transport in heterogeneous aquifers: a comparative study, Adv. Water Resour. 34 (1) (2011) 47–61, https://doi.org/10.1016/j.advwatres.2010.09.012.

[12] C. Yu, K. Malakpoor, J.M. Huyghe, Comparing mixed hybrid finite element method with standard FEM in swelling simulations involving extremely large deformations, Comput. Mech. 66 (2) (2020) 287–309, https://doi.org/10.1007/s00466-020-01851-z.

[13] Message Passing Interface Forum, MPI: A Message-Passing Interface Standard Version 3.1, https://www.mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf, Jun. 2015.

[14] W. Brutsaert, Evaporation into the Atmosphere: Theory, History, and Applications, Springer, 1982.

[15] Y. Ishihara, E. Shimojuma, H. Harada, Water vapor transfer beneath bare soil where evaporation is influenced by a turbulent surface wind, J. Hydrol. 131 (1992) 63–104, https://doi.org/10.1016/0022-1694(92)90213-F.

[16] A.C. Trautz, T.H. Illangasekare, S. Howington, Experimental testing scale considerations for the investigation of bare-soil evaporation dynamics in the presence of sustained above-ground airflow, Water Resour. Res. 54 (11) (2018) 8963–8982, https://doi.org/10.1029/2018WR023102.

[17] A.C. Trautz, T.H. Illangasekare, S. Howington, A. Cihan, Sensitivity of a continuum-scale porous media heat and mass transfer model to the spatial-discretization length-scale of applied atmospheric forcing data, Water Resour. Res. 55 (4) (2019) 3520–3540, https://doi.org/10.1029/2018WR023923.

[18] A.C. Trautz, T.H. Illangasekare, I. Rodriguez-Iturbe, Role of co-occurring competition and facilitation in plant spacing hydrodynamics in water-limited environments, Proc. Natl. Acad. Sci. 114 (35) (2017) 9379–9384, https://doi.org/10.1073/pnas.1706046114.

[19] A.C. Trautz, T.H. Illangasekare, I. Rodriguez-Iturbe, K. Heck, R. Helmig, Development of an experimental approach to study coupled soil-plant-atmosphere processes using plant analogs, Water Resour. Res. 53 (4) (2017) 3319–3340, https://doi.org/10.1002/2016WR019884.

[20] A.C. Trautz, T.H. Illangasekare, Evapotranspiration from Two Limestone Blocks (Plant Analog) Experiments, Mendeley Data, 2022.

[21] P.-L. Lions, N. Masmoudi, Incompressible limit for a viscous compressible fluid, J. Math. Pures Appl. 77 (6) (1998) 585–627, https://doi.org/10.1016/S0021-7824(98)80139-6.

[22] B. Desjardins, E. Grenier, Low Mach number limit of viscous compressible flows in the whole space, R. Soc. Lond. Proc., Ser. A, Math. Phys. Eng. Sci. 455 (1986) (1999) 2271–2279, https://doi.org/10.1098/rspa.1999.0403.

[23] R.B. Bird, W.E. Stewart, E.N. Lightfoot, Transport Phenomena, 2nd edition, John Wiley & Sons, Inc., 2002.

[24] Engineering ToolBox, Air – dynamic and kinematic viscosity, [Online; Accessed 27-March-2022] (2003), https://www.engineeringtoolbox.com/air-absolute-kinematic-viscosity-d_601.html.

[25] W.J. Massman, A review of the molecular diffusivities of H2O, CO2, CH4, CO, O3, SO2, NH3, N2O, NO, and NO2 in air, O2 and N2 near STP, Atmos. Environ. 32 (6) (1998) 1111–1127, https://doi.org/10.1016/S1352-2310(97)00391-9.

[26] S.K. Kang, Y.A. Hassan, The effect of lattice models within the lattice Boltzmann method in the simulation of wall-bounded turbulent flows, J. Comput. Phys. 232 (1) (2013) 100–117, https://doi.org/10.1016/j.jcp.2012.07.023.

[27] M. Geier, M. Schönherr, A. Pasquali, M. Krafczyk, The cumulant lattice Boltzmann equation in three dimensions: theory and validation, Comput. Math. Appl. 70 (4) (2015) 507–547, https://doi.org/10.1016/j.camwa.2015.05.001.

[28] P. Kumar, K. Kutscher, M. Mößner, R. Radespiel, M. Krafczyk, M. Geier, Validation of a VRANS-model for turbulent flow over a porous flat plate by cumulant lattice Boltzmann DNS/LES and experiments, J. Porous Media 21 (5) (2018), https://doi.org/10.1615/JPorMedia.v21.i5.60.

[29] C. Peng, N. Geneva, Z. Guo, L.-P. Wang, Direct numerical simulation of turbulent pipe flow using the lattice Boltzmann method, J. Comput. Phys. 357 (2018) 16–42, https://doi.org/10.1016/j.jcp.2017.11.040.

[30] M. Wittmann, T. Zeiser, G. Hager, G. Wellein, Domain decomposition and locality optimization for large-scale lattice Boltzmann simulations, Comput. Fluids 80 (2013) 283–289, https://doi.org/10.1016/j.compfluid.2012.02.007.

[31] A. Zakirov, A. Perepelkina, V. Levchenko, S. Khilkov, Streaming techniques: revealing the natural concurrency of the lattice Boltzmann method, J. Supercomput. 77 (10) (2021) 11911–11929, https://doi.org/10.1007/s11227-021-03762-z.

[32] R. Fučík, P. Eichler, R. Straka, P. Pauš, J. Klinkovský, T. Oberhuber, On optimal node spacing for immersed boundary–lattice Boltzmann method in 2D and 3D, Comput. Math. Appl. 77 (4) (2019) 1144–1162, https://doi.org/10.1016/j.camwa.2018.10.045.

[33] P. Eichler, R. Fučík, R. Straka, Computational study of immersed boundary-lattice Boltzmann method for fluid-structure interaction, Discrete Contin. Dyn. Syst., Ser. S 14 (3) (2021) 819, https://doi.org/10.3934/dcdss.2020349.
[34] P. Eichler, V. Fuka, R. Fučík, Cumulant lattice Boltzmann simulations of turbulent flow above rough surfaces, Comput. Math. Appl. 92 (2021) 37–47, https://doi.org/10.1016/j.camwa.2021.03.016.
[35] M. Geier, A. Pasquali, M. Schönherr, Parametrization of the cumulant lattice Boltzmann method for fourth order accurate diffusion part I: derivation and validation, J. Comput. Phys. 348 (2017) 862–888, https://doi.org/10.1016/j.jcp.2017.05.040.
[36] M. Geier, A. Pasquali, M. Schönherr, Parametrization of the cumulant lattice Boltzmann method for fourth order accurate diffusion part II: application to flow around a sphere at drag crisis, J. Comput. Phys. 348 (2017) 889–898, https://doi.org/10.1016/j.jcp.2017.07.004.
[37] J. Latt, B. Chopard, O. Malaspinas, M. Deville, A. Michler, Straight velocity boundaries in the lattice Boltzmann method, Phys. Rev. E 77 (2008) 056703, https://doi.org/10.1103/PhysRevE.77.056703.
[38] A.A. Mohamad, S. Succi, A note on equilibrium boundary conditions in lattice Boltzmann fluid dynamic simulations, Eur. Phys. J. Spec. Top. 171 (1) (2009) 213–221, https://doi.org/10.1140/epjst/e2009-01031-9.
[39] M. Haussmann, A.C. Barreto, G.L. Kouyi, N. Rivière, H. Nirschl, M.J. Krause, Large-eddy simulation coupled with wall models for turbulent channel flows at high Reynolds numbers with a lattice Boltzmann method—application to Coriolis mass flowmeter, Comput. Math. Appl. 78 (10) (2019) 3285–3302, https://doi.org/10.1016/j.camwa.2019.04.033.
[40] F.M. White, Viscous Fluid Flow, McGraw-Hill, New York, 2005.
[41] L. Davidson, M. Billson, Hybrid LES-RANS using synthesized turbulent fluctuations for forcing in the interface region, Int. J. Heat Fluid Flow 27 (6) (2006) 1028–1042, https://doi.org/10.1016/j.ijheatfluidflow.2006.02.025.
[42] L. Davidson, Using isotropic synthetic fluctuations as inlet boundary conditions for unsteady simulations, Adv. Appl. Fluid Mech. 1 (2007) 1–35.
[43] L. Davidson, Hybrid LES-RANS: inlet boundary conditions for flows including recirculation, in: Fifth International Symposium on Turbulence and Shear Flow Phenomena, Begel House Inc., 2007, pp. 689–694.
[44] L. Davidson, Fluid Mechanics, Turbulent Flow and Turbulence Modeling, 2018.
[45] J. Solovský, R. Fučík, M.R. Plampin, T.H. Illangasekare, J. Mikyška, Dimensional effects of inter-phase mass transfer on attenuation of structurally trapped gaseous carbon dioxide in shallow aquifers, J. Comput. Phys. 405 (2020) 109178, https://doi.org/10.1016/j.jcp.2019.109178.
[46] A.H. Askar, T.H. Illangasekare, A. Trautz, J. Solovský, Y. Zhang, R. Fučík, Exploring the impacts of source condition uncertainties on far-field brine leakage plume predictions in geologic storage of CO2: integrating intermediate-scale laboratory testing with numerical modeling, Water Resour. Res. 57 (9) (2021) e2021WR029679, https://doi.org/10.1029/2021WR029679.
[47] F. Brunner, F.A. Radu, P. Knabner, Analysis of an upwind-mixed hybrid finite element method for transport problems, SIAM J. Numer. Anal. 52 (1) (2014) 83–102, https://doi.org/10.1137/130908191.
[48] M.G. Larson, A.J. Niklasson, A conservative flux for the continuous Galerkin method based on discontinuous enrichment, Calcolo 41 (2) (2004) 65–76, https://doi.org/10.1007/BF02637255.
[49] S. Sun, M.F. Wheeler, Projections of velocity data for the compatibility with transport, Comput. Methods Appl. Mech. Eng. 195 (7–8) (2006) 653–673, https://doi.org/10.1016/j.cma.2005.02.011.
[50] C.E. Kees, M.W. Farthing, C.N. Dawson, Locally conservative, stabilized finite element methods for variably saturated flow, Comput. Methods Appl. Mech. Eng. 197 (51–52) (2008) 4610–4625, https://doi.org/10.1016/j.cma.2008.06.005.
[51] T. Oberhuber, J. Klinkovský, R. Fučík, TNL: numerical library for modern parallel architectures, Acta Polytech. 61 (SI) (2021) 122–134, https://doi.org/10.14311/AP.2021.61.0122.
[52] NVIDIA, CUDA Toolkit Documentation, Version 11.6, Nvidia, 2022, https://docs.nvidia.com/cuda/archive/11.6.0/.
[53] B. Welford, Note on a method for calculating corrected sums of squares and products, Technometrics 4 (3) (1962) 419–420, https://doi.org/10.1080/00401706.1962.10490022.
[54] R.F. Ling, Comparison of several algorithms for computing sample means and variances, J. Am. Stat. Assoc. 69 (348) (1974) 859–866, https://doi.org/10.1080/01621459.1974.10480219.
[55] T.F. Chan, G.H. Golub, R.J. LeVeque, Algorithms for computing the sample variance: analysis and recommendations, Am. Stat. 37 (3) (1983) 242–247, https://doi.org/10.1080/00031305.1983.10483115.
[56] J.C.I. Dooge, Looking for hydrologic laws, Water Resour. Res. 22 (1986) 46S–58S, https://doi.org/10.1029/WR022i09Sp0046S.
[57] S.A. Wolfe, W.G. Nickling, The protective role of sparse vegetation in wind erosion, Prog. Phys. Geogr. 17 (1) (1993) 50–68, https://doi.org/10.1177/030913339301700104.