Engineering with Computers
https://doi.org/10.1007/s00366-023-01854-3

ORIGINAL ARTICLE

Lattice-Boltzmann LES modelling of a full-scale, biogas-mixed anaerobic digester

Davide Dapelo[1] · Adrian Kummerländer[2] · Mathias J. Krause[2] · John Bridgeman[1]

Received: 15 September 2022 / Accepted: 16 May 2023

Abstract

An Euler–Lagrange multicomponent, non-Newtonian Lattice-Boltzmann method is applied for the first time to model a full-scale gas-mixed anaerobic digester for wastewater treatment. Rheology is modelled through a power-law model and, for the first time in gas-mixed anaerobic digestion modelling, turbulence is modelled through a Smagorinsky Large Eddy Simulation model. The hydrodynamics of the digester is studied by analysing flow and viscosity patterns, and assessing the degree of mixing through the Uniformity Index method. Results show independence from the grid size and the number of Lagrangian substeps employed for the Lagrangian sub-grid simulation model. Flow patterns are shown to depend mildly on the choice of bubble size, but not the asymptotic degree of mixing. Numerical runs of the model are compared to previous results in the literature, from a second-ordered Finite-Volume Method approach, and demonstrate an improvement, compared to literature data, of 1000-fold computational efficiency, massive parallelizability and much finer attainable spatial resolution. Whilst previous research concluded that the application of LES to full-scale anaerobic digestion mixing is unfeasible because of high computational expense, the increase in computational efficiency demonstrated here, now makes LES a feasible option to industries and consultancies.

Keywords Lattice-Boltzmann · OpenLB · Anaerobic digestion, full-scale · Non-Newtonian · Euler–Lagrange

List of symbols

Δt_{inj}	Time interval between the injection of two bubbles, s
Φ	Phase space
$\boldsymbol{\Phi}$	Sourcing term, kg m^3
Ξ	Collision step, kg m^3
$\lvert\dot{\gamma}\rvert$	Shear rate magnitude, s^{-1}
$\lvert\dot{\gamma}\rvert_{\mathrm{ref}}$	Reference shear rate magnitude, s^{-1}
$\delta^3_{\cdot,\cdot}$	3D Kronecker delta
δt	Lattice timestep, s
δx	Lattice cell size, m
ϕ	Finite-difference tracer field, m^{-3}
μ	Apparent dynamics viscosity, Pa s
μ_{max}	Apparent dynamics viscosity, maximum range value, Pa s
μ_{min}	Apparent dynamics viscosity, minimum range value, Pa s
μ_{PL}	Apparent dynamic viscosity before turbulence correction, Pa s
ν	Kinematic viscosity, m^2 s^{-1}
ν_{PL}	Apparent kinematic viscosity before turbulence correction, m^2 s^{-1}
ν_{ref}	Reference kinematic viscosity, m^2 s^{-1}
ν_{turb}	Turbulent kinematic viscosity, m^2 s^{-1}
ρ	Liquid phase density, kg m^{-3}
σ	Liquid phase shear stress, Pa
τ	Lattice relaxation time, s
$\boldsymbol{A}_K$	Acceleration of the Kth Lagrangian sub-grid particle, m s^{-2}
$\mathscr{C}$	Collision operator, kg m^3
C_{d}	Drag coefficient
C_{Smago}	Smagorinsky constant
$\boldsymbol{F}_K$	Total force acting on the Kth Lagrangian sub-grid particle, N
$\boldsymbol{F}^{\mathrm{a}}_K$	Added-mass force acting on the Kth Lagrangian sub-grid particle, N

✉ Davide Dapelo
d.dapelo@liverpool.ac.uk

[1] School of Engineering, University of Liverpool, Liverpool, UK

[2] Lattice Boltzmann Research Group, Karlsruhe Institute of Technology, Karlsruhe, Germany

Published online: 17 June 2023

$\boldsymbol{F}_K^{\mathrm{b}}$	Buoyancy force acting on the *K*th Lagrangian sub-grid particle, N
$\boldsymbol{F}_K^{\mathrm{d}}$	Drag force acting on the *K*th Lagrangian sub-grid particle, N
K	Power-law consistency coefficient, Pa s^n
K	(As a subscript) Generic label to a Lagrangian sub-grid particle
M_K	Mass of the *K*th Lagrangian sub-grid particle, kg
Ma	Mach number
$\mathscr{P}_K$	Tuple representing the *K*th Lagrangian sub-grid particle
Re	Reynolds number
Re_{p}	Particle Reynolds number
R_K	Nominal radius of the *K*th Lagrangian sub-grid particle, m
$\boldsymbol{S}$	Rate of shear tensor, s^{-1}
$\mathscr{S}$	Source term operator, kg m^3
$\boldsymbol{U}$	Nominal velocity scale, m s^{-1}
UI	Uniformity index
$\boldsymbol{U}_K$	Spatial coordinate of the *K*th Lagrangian sub-grid particle, m s^{-1}
U_{LB}	Lattice velocity
U_{LB}^0	Reference lattice velocity
$\boldsymbol{X}_K$	Spatial coordinate of the *K*th Lagrangian sub-grid particle, m
$\boldsymbol{X}_K^{\mathrm{next}}$	Spatial coordinate of the *K*th Lagrangian sub-grid particle, approximated at the nearest lattice node, m
$\boldsymbol{c}$	Mesoscopic velocity, m s^{-1}
$\boldsymbol{c}_i$	*i*th discretised lattice (mesoscopic) velocity, m s^{-1}
c_{s}	Lattice speed velocity, m s^{-1}
d	Bubble diameter, m
f	One-particle density function, kg m^{-3}
$f^{(1)}$	First-order multiscale term of the one-particle distribution function, kg m^{-3}
$f^{(\mathrm{eq})}$	Equilibrium one-particle density function, kg m^{-3}
$\boldsymbol{g}$	Acceleration of gravity, m s^{-2}
n	Power-law index
n_x	Number of lattice sites across the tank's diameter
n_x^0	Reference number of lattice sites across the tank's diameter
p	Pressure, Pa
s	Number of Lagrangian subcycles
t	Time, s
$\boldsymbol{u}$	Liquid phase velocity, m s^{-1}
u_{surr}	Liquid phase velocity magnitude in the surroundings of a rising biogas bubble, m s^{-1}
w_i	Weight of the *i*th component of the equilibrium particle distribution
$\boldsymbol{x}$	Discretised lattice spatial coordinate, m
$\cdot^*$	Dimensionless version of the argument represented by the $\cdot$
CFD	Computational fluid dynamics
CPUs	CPU-second (i.e., number of seconds a given numerical simulation takes to be run, times number of CPU cores employed)
EU	European Union
FV	Finite-volume
LB	Lattice-Boltzmann
LES	Large Eddy simulations
TS	Total solid content
WFD	EU Water Framework Directive
WwTW	Wastewater treatment work

1 Introduction

Over the next decades, the wastewater industry will continue to be subjected to unprecedented challenges, as worldwide demands for food and clean water are expected to rise by 50% and 30% respectively [1]. Furthermore, implementation of the EU Water Framework Directive (WFD) is driving an increase of energy consumption by up to 60% in wastewater treatment works (WwTWs) over the next 10–15 years [2], due to tighter discharge requirements. The wastewater-energy link must be clearly addressed in order to mitigate, and adapt towards, climate change.

Since 2009, wastewater treatment works across each major European country have produced over 1 M tonnes sludge per country per year [3]. The preferred method to treat sludge is mesophilic (22–41 °C) anaerobic digestion with mixing occurring through biogas injection. Through this process, sludge is degraded by anaerobic bacteria into stable digestate and biogas (a mixture of mainly methane and carbon dioxide). Biogas is usually directed to a combined heat and power unit for energy recovery. Mixing is necessary for correct digestion and can be responsible for anywhere between 17% and 73% of digester energy consumption [4]. This level of consumption is largely suboptimal, as experimental evidence [5] shows that input mixing power can be reduced by to 50% without affecting the digestion process. To address mitigation and adaptation to climate change, it is therefore necessary to rethink mixing design practices and operation protocols, with the goal of balancing input mixing energy against output biogas production, rather than merely considering digestate quality.

Over the years, (segregated) Finite-Volume (FV) Computational Fluid Dynamics (CFD) has been successfully employed to model gas-mixed anaerobic digesters [6–19].

A CFD approach to design and system analysis offers multiple benefits, including a saving of time and money arising from avoiding lengthy and time-consuming experiments, and providing an insight to flow patterns which are unattainable from optical visualisation techniques (sludge is opaque) or tracer-response methods (which provide no more than a black-box description of the system). This progress has enabled the development of structured modelling protocols to significantly improve energy performance of both new and existing full-scale digester [20]. However, limitations in the industrial applicability of this approach persist, as long simulation runtimes ($\geq$ 2 days) render the deployment of the above-mentioned strategies excessively time-consuming. Furthermore, the multi-core parallel run of most common Finite-Volume schemes (viz., up to second-order) is hampered by poor parallel performance [21], mainly due to the high proportion of non-scalable inter-core communication operations involved in solving the Poisson pressure equation [22]. Indeed, previous Finite-Volume models of full-scale anaerobic digesters [17, 20, 23] could not scale up beyond 36 cores. As such, traditional Finite-Volume CFD cannot benefit from the on-going evolution of high-performance computing. In turn, this makes it impractical, or very time-consuming, to employ accurate but resource-intensive methodologies, e.g., the Large Eddy Simulations (LES). Indeed, no LES model for full-scale gas-mixed anaerobic digestion has been developed so far: only [24] has developed a LES Finite-Volume model for full-scale anaerobic digestion, but for mechanical, not gas, mixing, and concluded that LES is impractical due to the excessive computational expense required.

A potential solution to both the problems listed above is offered by the Lattice-Boltzmann (LB) method, a relatively recent CFD alternative to the Finite-Volume approach with recent industrial applications comprising, among other things, Ball-Grid-Array encapsulation process, heat flux inside refrigerated vehicles, internal-combustion engine and 3D-printed wet-scrubber nozzle [25–28]. Lattice-Boltzmann is essentially a Finite-Difference method equipped with tunable diffusivity [29]. Lattice-Boltzmann presents tangible advantages over the traditional Finite-Volume approach, such as: (i) full explicitness free from internal loops, resulting in a well-defined, limited number of floating-point operations per timestep; and (ii) strong parallelizability due to reduced non-scalable inter-core communication thanks to a formal and implementational distinction between non-local and non-linear parts of the algorithm and non-local access usually limited to first-neighbour cells only. Furthermore, the Lattice-Boltzmann method has the advantage over other models suitable to parallel computing (viz., high-order Finite-Volume, [21]), of (iii) implementational simplicity, as its structured grid approach and first-neighbour-only non-local access allow it to avoid complex stencils and to implement boundary conditions in a straightforward manner. Multiphase models, both Euler–Euler [30] and Euler–Lagrange [31–35], are available. A comparison between a Lattice-Boltzmann LES model and its Finite-Volume counterpart applied to a internal-combustion engine [27] showed that the former ran 32 times faster, thus making the usage of LES much more practical: “The faster calculation speed for NWM-LES using LBM is advantageous to address industrial applications and to enable ‘overnight’ calculations that previously took weeks. Therefore, faster design cycles and operating condition tests are feasible”. Coming to anaerobic digestion, a Lattice-Boltzmann model for a laboratory-scale gas-mixed digester [34] has been shown to perform around 180 times faster than its Finite-Volume analogue [14] whilst being able to run on ten times more processors without appreciable efficiency decrease. Thus, Lattice-Boltzmann’s superior numerical efficiency and parallelizability allow much finer grids than Finite-Volume at comparable numerical expense and runtimes, amply compensating for the errors arising from the traditional lack of local mesh refinement, unstructured and body-fitted grids in the traditional Lattice-Boltzmann implementations. It is therefore clear that Lattice-Boltzmann models can deliver a significant benefit to the operation of numerical modelling anaerobic digestion with gas mixing.

The topic of mixing improvement in full-scale anaerobic digestion has been widely investigated through CFD despite severe limitations in validation procedure: the intrinsically opaque and hazardous nature of sludge, as well as the impracticability of taking digesters out of production for experimental purposes, means that no experimental data concerning full-scale anaerobic digesters for wastewater treatment are available in the literature. As a result, there are examples in the literature where researchers provide unvalidated results in full-scale anaerobic digestion [8–10, 13]; rely on full-scale validation conducted on black-box measurements such as impeller power number [36]; or validate a model against flow patterns from a lab-scale setup, and then apply it to the full-scale [14, 17, 20, 23, 37].

Despite the advantages of CFD and, to a greater extent, Lattice-Boltzmann method, for anaerobic digestion modelling, only a limited amount of work has been dedicated to modelling the gas mixing processes [6, 10, 14, 17, 20, 23, 34, 35, 38]; among this, only [34, 35] employed the Lattice-Boltzmann; and finally, none reports full-scale Lattice-Boltzmann models with gas mixing. Dapelo et al. [34] introduced the first-ever multiphase Lattice-Boltzmann model for gas-mixed anaerobic digestion in a laboratory-scale set-up, and [35] demonstrated that the sub-grid Euler–Lagrange Lattice-Boltzmann method can be successfully employed to model laboratory and full-scale anaerobic digesters. In both [34, 35], the models were validated against laboratory-scale experiments conducted by [14]. However, a knowledge gap persists as no Lattice-Boltzmann model for gas-mixed industrial-scale digesters has been reported in the literature.

Within the work reported here, the sub-grid method developed and validated lab-scale in [35] is used for the first time to model a full-scale setup reproducing a real wastewater treatment digester, applying the same approach towards validation of [14, 17, 20, 23, 37] and thereby filling the above-mentioned knowledge gap. Flow and viscosity patterns are analysed, and the degree of mixing is evaluated through the Uniformity Index (UI) method proposed by [9]. The effect of different modelling parameters on the simulation outcome is assessed. The results are discussed and compared to previous second-order Finite-Volume work on a similar design [17, 20, 23]. It is shown how the Lattice-Boltzmann method offers advantages over the method used therein, and has a clear potential to overcome the issues concerning industrial applicability of CFD-based mixing-improvement strategies described above. Likewise, it is shown that the introduction of a Lattice-Boltzmann-based model makes the application of LES to full-scale anaerobic digestion practically feasible to industries and consultancies.

This paper is structured as follows. Sludge is modelled in Sect. 2: the assumptions underlying the multiphase model are laid down in Sect. 2.1; then, the model is described within the Lattice-Boltzmann framework in Sects. 2.2 and 2.3; finally, the pseudocode algorithm is reported in Sect. 2.4. The model's implementation in OpenLB is reported in Sect. 3, and a short description of OpenLB and the innovation it has brought in the field of parallel computing is offered in Sect. 3.1. then, the results are reported in Sect. 4, and specifically: flow patterns (Sect. 4.1); grid independence (Sect. 4.2); mixing efficiency (Sect. 4.3); dependence of the results from the choice of Lagrangian subcycles (Sect. 4.4) and bubble size (Sect. 4.5); and scaling-up (Sect. 4.6). Then, a discussion is performed (Sect. 5), and conclusions are drawn (Sect. 6).

2 Modelling of sludge

Within this work, one of the models described in [35], with the geometry of [17, 23] is used. It is summarised here for the sake of clarity.

2.1 Assumptions

Sludge is a complex mixture of organic and inorganic solids arranged in fragments of various dimensions (from colloid molecules to sand or gravel), water and biogas bubbles where gas mixing is employed. The range of inter-phase phenomena include bubble–liquid (two-way) and bubble–bubble (four-way) momentum transfer, solid–liquid interactions such as grit sedimentation and scum flotation, and complex liquid rheology characterised by shear thinning, shear banding, yield stress and thixotropy.

To simplify the problem of modelling sludge, a sub-grid, two-way coupled Euler–Lagrange model with non-Newtonian rheology and a large-eddy-simulation turbulence model is introduced. The assumptions and simplifications underlying the choice of this model, as well as the justifications underpinning them, are listed as follows.

(i) Sedimentation and flotation are ignored because they respectively take place in years/months and days/weeks, whilst the timescale of the mixing is up to 2 h.

(ii) Solid phase is considered as a suspension of liquid phase, and its effect on the latter is modelled as the liquid phase's non-Newtonian pseudoplastic rheology [39], with more complex rheological phenomena being ignored. In a pseudoplastic power-law model, the apparent viscosity μ is a function of the magnitude of the sear rate $|\dot{\gamma}|$, as follows:

$$\mu = K|\dot{\gamma}|^{n-1}, \tag{1}$$

with K and n ($0 < n < 1$) being respectively the consistency and power-law coefficients. Although both K and n depend on temperature and total solids content

Table 1 Power-law coefficients, cutoffs and density of sludge at T=35 °C. From [40]

TS (%)	K (Pa s^n)	n (–)	$\lvert\dot{\gamma}\rvert$ range (s^{-1})	$\mu_{\min}$ (Pa s)	$\mu_{\max}$ (Pa s)	Density (kg m^{-3})
2.5	0.042	0.710	226–702	0.006	0.008	1000.36
5.4	0.192	0.562	50–702	0.01	0.03	1000.78
7.5	0.525	0.533	11–399	0.03	0.17	1001.00
9.1	1.052	0.467	11–156	0.07	0.29	1001.31
12.1	5.885	0.367	3–149	0.25	2.93	1001.73

[40], temperature dependence is ignored as K and n are considered constant at the fixed temperature of 35 °C, this being the ideal temperature for mesophilic conditions. Table 1 reports typical values of n and K at 35 °C. For the sake of simplicity, sludge density is set to 1000 kg m^{-3}. Equation (1) returns unphysically high or low apparent viscosity values for low or high values of $|\dot{\gamma}|$ respectively; this is avoided in a standard way by introducing a minimum and maximum cutoff value for the apparent viscosity, $\mu_{\min}$ and $\mu_{\max}$.

(iii) (iii) As reported in the following Results sections, the Reynolds number is found to be comprised between 3600 and 6100, and therefore, turbulence is modelled. A large eddy simulation (LES) model is chosen, and the Smagorinsky constant C_{Smago} is set to 0.14.

(iv) Bubble–bubble interaction, bubble coalescence and breakup are ignored, as they were found not to occur in experimental work [14]. Conversely, mixing occurs because of the momentum of the buoyant bubbles being transferred to the surrounding liquid phase. Therefore, the bubble–liquid interaction must be modelled; i.e. bubble–liquid two-way coupling is considered whereby momentum is transferred from the liquid phase to the bubbles ("*forward-coupling*"); and from the bubbles to the liquid phase ("*back-coupling*").

(v) The smallest grid cells used in this work are cubes of 9 cm size, which are larger than the largest bubble diameter (5 cm). Previous work [35] showed that liquid phase flow patterns can be effectively reproduced through a sub-grid Euler–Lagrange model, and consequently, bubbles are considered as pointwise within this work. Bubbles are also assume to have the same density of air, i.e. 1 kg m^{-3}.

2.2 Modelling and simulation: sub-grid Euler–Lagrange bubbly phase

The dispersed bubbly phase is modelled as a collection of sub-grid elements $\mathscr{P}_K$, or "particles" [34, 41]—one (spherical) bubble per particle. As rotational effects and deviations from sphericity were found to be negligible in previous work [14, 34], it is possible to represent each $\mathscr{P}_K$ as a tuple of numbers consisting of: coordinate $\boldsymbol{X}_K$, velocity $\boldsymbol{U}_K$, acceleration $\boldsymbol{A}_K$, nominal radius R_K and mass M_K:

$$\mathscr{P}_K \equiv \left(\boldsymbol{X}_K, \boldsymbol{U}_K, \boldsymbol{A}_K, R_K, M_K\right). \tag{2}$$

At any Lattice-Boltzmann update, each $\mathscr{P}_K$ within the domain is updated separately via verlet integration of Newton's second law [42] over a number s of "Lagrangian subcycles" with Lagrangian timestep $\delta t/s$:

$$\boldsymbol{F}_K = M_K \boldsymbol{A}_K. \tag{3}$$

The resultant $\boldsymbol{F}_K$ of the forces acting on $\mathscr{P}_K$ is modelled as a sum of buoyancy $\boldsymbol{F}_K^{\mathrm{b}}$, added mass $\boldsymbol{F}_K^{\mathrm{a}}$ and gravity $\boldsymbol{F}_K^{\mathrm{d}}$:

$$\boldsymbol{F}_K = \boldsymbol{F}_K^{\mathrm{b}} + \boldsymbol{F}_K^{\mathrm{a}} + \boldsymbol{F}_K^{\mathrm{d}}. \tag{4}$$

Forward-coupling is achieved by modelling $\boldsymbol{F}_K^{\mathrm{b}}$, $\boldsymbol{F}_K^{\mathrm{a}}$ and $\boldsymbol{F}_K^{\mathrm{d}}$ in terms of the liquid phase's local values of the macroscopic fields: in [35], different models were tested, and the best results in terms of both convergence and numerical expense were achieved when the values of the liquid phase density and velocity fields ρ and $\boldsymbol{u}$ at the Kth particle's position $\boldsymbol{X}_K$ were determined through linear interpolation across the cells surrounding $\mathscr{P}_K$; conversely, the value of the kinematic viscosity ν was approximated to the nearest cell $\boldsymbol{X}_K^{\mathrm{next}}$. The same approach is then adopted here. For buoyancy we have:

$$\boldsymbol{F}_K^{\mathrm{b}} = -\left[\frac{4}{3}\pi R_K^3 \rho\left(\boldsymbol{X}_K\right) - M_K\right]\boldsymbol{g} \tag{5}$$

with $\boldsymbol{g}$ being the acceleration of gravity. Added mass is given by:

$$\begin{aligned}\boldsymbol{F}_K^{\mathrm{a}} &= -\frac{1}{2}\rho\left(\boldsymbol{X}_K\right)\frac{4}{3}\pi R_K^3 \frac{\mathrm{d}}{\mathrm{d}t}\left[\boldsymbol{U}_K - \boldsymbol{u}\left(\boldsymbol{X}_K\right)\right] \\ &\simeq -\frac{1}{2}\rho\left(\boldsymbol{X}_K\right)\frac{\pi}{6}d_K^3 A_K.\end{aligned} \tag{6}$$

The drag force is defined as:

$$\begin{aligned}\boldsymbol{F}_K^{\mathrm{d}} &= -\frac{1}{2}\rho\left(\boldsymbol{X}_K\right)C_{\mathrm{d}}\left(\mathrm{Re}_{\mathrm{P}}\right)\pi R_K^2 \left|\boldsymbol{U}_K - \boldsymbol{u}\left(\boldsymbol{X}_K\right)\right| \\ &\left[\boldsymbol{U}_K - \boldsymbol{u}\left(\boldsymbol{X}_K\right)\right].\end{aligned} \tag{7}$$

As in [35], Morsi's drag coefficient [43] C_{d} is used. The particle Reynolds number Re_{P} is evaluated as:

$$\mathrm{Re}_{\mathrm{P}} = \frac{2R_K\left|\boldsymbol{U}_k - \boldsymbol{u}\left(\boldsymbol{X}_K\right)\right|}{\nu\left(\boldsymbol{X}_K^{\mathrm{next}}\right)}. \tag{8}$$

2.3 Modelling and simulation: Lattice-Boltzmann method for the fluid phase

The Lattice-Boltzmann model solves the one-particle density function $f(\boldsymbol{x}, \boldsymbol{c}, t)$, which is defined as the probability of finding one ideally pointwise and indivisible portion of fluid with velocity in $[\boldsymbol{c}, \boldsymbol{c} + d\boldsymbol{c}]$ and position in $[\boldsymbol{x}, \boldsymbol{x} + d\boldsymbol{x}]$ at the time t. The method is mesoscopic insofar as the observable macroscopic fields of density $\rho(\boldsymbol{x}, t)$, velocity $\boldsymbol{u}(\boldsymbol{x}, t)$ and shear stress $\boldsymbol{\sigma}(\boldsymbol{x}, t)$ are not directly resolved—rather, they are evaluated from f's first three moments [44]:

$$\rho = \int f \,\mathrm{d}\boldsymbol{c}; \tag{9}$$

$$\rho \boldsymbol{u} = \int f \boldsymbol{c} \,\mathrm{d}\boldsymbol{c}; \tag{10}$$

$$\rho \boldsymbol{u} \otimes \boldsymbol{u} = \sigma + \int f \boldsymbol{c} \otimes \boldsymbol{c} \,\mathrm{d}\boldsymbol{c}. \tag{11}$$

f's continuity equation in the phase space $\Phi(\boldsymbol{x}, \boldsymbol{c})$ takes the name of the Boltzmann equation:

$$\left(\partial_t + \boldsymbol{c} \cdot \nabla\right) f = \mathscr{C}[f]. \tag{12}$$

The "collision operator" $\mathscr{C}$ is a source-sink term, modelling the effect of *f* of inter-particle collisions taking place within the cube $[\boldsymbol{x}, \boldsymbol{x} + \mathrm{d}\boldsymbol{x}]$ between *t* and $\mathrm{d} + \mathrm{d}t$. Under the diluted gas assumption (which is considered to hold in the work presented here), only binary collisions are accounted for in $\mathscr{C}$. Under the widely-adopted Bhatnagar–Gross–Krook (BGK) hypothesis [45], collisions occur isotropically and induce a relaxation of *f* towards an equilibrium distribution $f^{(\mathrm{eq})}$ with relaxation time τ:

$$\mathscr{C}[f] = -\frac{f - f^{(\mathrm{eq})}}{\tau}. \tag{13}$$

The equilibrium distribution is the Maxwell equilibrium distribution [44]:

$$f^{(\mathrm{eq})}(\boldsymbol{x}, t) := \rho(\boldsymbol{x}, t)\left(\frac{1}{2\pi c_\mathrm{s}^2}\right)^{3/2} \exp\left\{-\frac{[\boldsymbol{u}(\boldsymbol{x}, t)]^2}{2c_\mathrm{s}^2}\right\}, \tag{14}$$

where velocity and density are evaluated through Eqs. (9) and (10), and c_s is the speed of sound.

Simulations consist of trajectories on the discretized phase space, with constant timestep δt. The phase space $\Phi(\boldsymbol{x}, \boldsymbol{c})$ is discretized as follows. The (spatial) discretized computational domain is defined as a 3D cubic lattice, with δx being the distance between two first-neighbouring sites. The discretized velocity space is generated by a set of vectors $\{\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{q-1}\}$ not mutually independent. $\boldsymbol{c}_0$ is the zero vector; the others point from one lattice site to its first neighbour and have module $\delta x/\delta t$; or to its second neighbour neighbour and have module $\sqrt{2}\,\delta x/\delta t$; or to its third (module $\sqrt{3}\,\delta x/\delta t$). The different choices of discretization are conventionally labelled through a tag D*d*Q*q*: *d* represents the spatial dimension (in this work, 3); *q* the number of vectors spanning the velocity space. In place of $f(\boldsymbol{x}, \boldsymbol{c}, t)$ we now have the discretized set $f_i(\boldsymbol{x}, t)$, where the latter is defined as the probability of finding one portion of fluid at the lattice site $\boldsymbol{x}$ with velocity $\boldsymbol{c}_i$ at the time *t*. The zeroth, first and second moments of *f* (Eqs. 9, 10 and 11) are evaluated as summations, in place of integrals, over the velocity set:

$$\rho = \sum_i f_i; \tag{15}$$

$$\rho \boldsymbol{u} = \sum_i f_i \boldsymbol{c}_i; \tag{16}$$

$$\rho \boldsymbol{u} \otimes \boldsymbol{u} = \sigma + \sum_i f_i \boldsymbol{c}_i \otimes \boldsymbol{c}_i. \tag{17}$$

Using a discretized velocity space induces a discretization error—however, this error source can be removed if the Maxwell equilibrium function (Eq. 14) is written as a linear combination of Hermite polynomials. To ensure density and momentum conservation, only the Hermite polynomials up to the second order are needed [44]. As such, the discretized equilibrium function reads as:

$$f_i^{(\mathrm{eq})} = w_i\,\rho\left[1 + \frac{\boldsymbol{u} \cdot \boldsymbol{c}_i}{c_\mathrm{s}^2} + \frac{\left(\boldsymbol{u} \cdot \boldsymbol{c}_i\right)^2 - c_\mathrm{s}^2 u^2}{2c_\mathrm{s}^4}\right]. \tag{18}$$

The values of the weights are set in a standard way depending of the specific D*d*Q*p* lattice. Similarly, the speed of sound is defined as:

$$c_\mathrm{s} := \frac{1}{\sqrt{3}}\frac{\delta x}{\delta t}. \tag{19}$$

The application of Eqs. (9) and (10) [44] allows evaluation of the macroscopic fields and recovery of the adiabatic dynamics with a Mach-number-dependent compressibility error of Ma^2. If the BGK assumption (Eq. 13) is adopted the Boltzmann Eq. (12) is discretized into the Lattice-Boltzmann equation:

$$f_i\left(\boldsymbol{x} + \boldsymbol{c}_i\,\delta t, t + \delta t\right) = f_i(\boldsymbol{x}, t) - \frac{f_i(\boldsymbol{x}, t) - f_i^{(\mathrm{eq})}(\boldsymbol{x}, t)}{\tau}. \tag{20}$$

Implementation of Eq. (20) is split into two steps: a local, non-linear *collision*:

$$\Xi_i(\boldsymbol{x}, t) = f_i(\boldsymbol{x}, t) - \frac{f_i(\boldsymbol{x}, t) - f_i^{(\mathrm{eq})}(\boldsymbol{x}, t)}{\tau}; \tag{21}$$

and a linear, non-local *streaming*:

$$f_i\big(\boldsymbol{x}+\boldsymbol{c}_i,\, t+\delta t\big) = \Xi_i(\boldsymbol{x},\, t). \tag{22}$$

A multiscale ("Chapman-Enskog") expansion shows that the Lattice-Boltzmann Eq. (20) reproduces the incompressible Navier–Stokes equations under the limit Ma $\ll 1$ [44]. Pressure and kinematic viscosity take the values:

$$p := \rho c_{\mathrm{s}}^2, \qquad \nu := c_{\mathrm{s}}^2\left(\tau - \frac{\delta t}{2}\right). \tag{23}$$

Non-Newtonian rheology and turbulence are accounted for following [34, 35]. The relaxation time is treated as a field $\tau(\boldsymbol{x},\, t)$ rather than a parameter; its value is stored alongside $f_i(\boldsymbol{x},\, t)$ and initialised at the first timestep by inverting the second of Eqs. (23) using a bespoke reference value ν_{ref} for the kinematic viscosity (see Sect. 3). τ is then updated locally at every timestep before the collision phase (Eq. 21), as follows:

1. Power-law rheology is modelled as in [46].The rate of shear tensor $S_{\alpha\beta} \equiv \frac{1}{2}\big(\partial_\alpha u_\beta + \partial_\beta u_\alpha\big)$ is evaluated locally from the second momentum of the first-order multiscale term of f, defined as $f_i^{(1)}$ [44]:

$$\begin{aligned}\boldsymbol{S}(\boldsymbol{x},\, t) &= -\frac{1}{2\rho\, c_{\mathrm{s}}^2\, \tau(\boldsymbol{x},\, t)} \sum_i f_i^{(1)}(\boldsymbol{x},\, t)\, \boldsymbol{c}_i \otimes \boldsymbol{c}_i \\ &\simeq -\frac{1}{2\rho\, c_{\mathrm{s}}^2\, \tau(\boldsymbol{x},\, t)} \sum_i \Big[f_i(\boldsymbol{x},\, t) - f_i^{(\mathrm{eq})}(\boldsymbol{x},\, t)\Big] \boldsymbol{c}_i \otimes \boldsymbol{c}_i.\end{aligned} \tag{24}$$

   Dynamic viscosity $\mu_{\mathrm{PL}}(\boldsymbol{x},\, t)$ and, consequently, kinematic viscosity $\nu_{\mathrm{PL}} \equiv \mu_{\mathrm{PL}}/\rho$ are obtained from the power-law Eq. (1) through the substitution:

$$|\dot\gamma| \equiv \sqrt{2\boldsymbol{S} : \boldsymbol{S}}, \tag{25}$$

   and then τ is recalculated from the second of Eq. (23), with ν_{PL} being used instead of ν.
2. Smagorinsky turbulence is modelled as in [47]. The shear rate magnitude $|\dot\gamma|$ is calculated after power-law correction by reapplying Eqs. (24) and (25). The Smagorinsky closure with $C_{\mathrm{Smago}} = 0.14$ is then applied in order to compute the turbulent linematic viscosity:

$$\nu_{\mathrm{turb}} = \nu_{\mathrm{PL}} + C_{\mathrm{Smago}}|\dot\gamma|. \tag{26}$$

   The final value of τ is then calculated by inverting once more the second of Eq. (23), with ν_{turb} in place of ν.

The momentum transfer from bubbles to liquid phase (viz., back-coupling) can be included by modifying the Lattice-Boltzmann Eq. (20) through a general procedure, due to [48]: a momentum source term due to a body force is added to the Lattice-Boltzmann Eq. (20):

$$\begin{aligned} f_i\big(\boldsymbol{x}+\boldsymbol{c}_i,\, t+\delta t\big) =& f_i(\boldsymbol{x},\, t) - \frac{f_i(\boldsymbol{x},\, t) - f_i^{(\mathrm{eq})}(\boldsymbol{x},\, t)}{\tau} \\ &+ \left(1 - \frac{1}{2\tau}\right)\mathscr{S}_i(\boldsymbol{x},\, t)\end{aligned} \tag{27}$$

and consequently, the collision Eq. (21):

$$\begin{aligned} \Xi_i(\boldsymbol{x},\, t) =& f_i(\boldsymbol{x},\, t) - \frac{f_i(\boldsymbol{x},\, t) - f_i^{(\mathrm{eq})}(\boldsymbol{x},\, t)}{\tau} \\ &+ \left(1 - \frac{1}{2\tau}\right)\mathscr{S}_i(\boldsymbol{x},\, t),\end{aligned} \tag{28}$$

where the source term $\mathscr{S}_i$ is a function of a particle-dependent forcing term $\boldsymbol{\Phi}$:

$$\mathscr{S}_i := w_i\left[\frac{\boldsymbol{c}_i - \boldsymbol{u}}{c_{\mathrm{s}}^2} + \frac{(\boldsymbol{c}_i \cdot \boldsymbol{u})\boldsymbol{c}_i}{c_{\mathrm{s}}^4}\right] \cdot \boldsymbol{\Phi}. \tag{29}$$

For the specific case of implementing the back-coupling, [35] proposed different models. The best performing in terms of convergence and numerical expense consisted of equating the forcing $\boldsymbol{\Phi}$ (in Eq. 29) to $-\boldsymbol{F}_K$ (in Eq. 4) to the most near cell from $\mathscr{P}_K$, and summing over the particles and the number of Lagrangian subcycles occurring within a Lattice-Boltzmann update:

$$\boldsymbol{\Phi} = -\sum_s \sum_K \boldsymbol{F}_K\ \delta^3_{\boldsymbol{x},\boldsymbol{X}^{\mathrm{next}}}. \tag{30}$$

Finally, Eq. (16) is modified as follows:

$$\rho\boldsymbol{u} = \sum_i f_i \boldsymbol{c}_i + \frac{1}{2}\sum_i \mathscr{S}_i \boldsymbol{c}_i. \tag{31}$$

2.4 Simulation algorithm

The model follows Algorithm 1 below.

Algorithm 1 General algorithm

```
Initialize Lattice-Boltzmann ($\boldsymbol{\Phi} = \mathbf{0}$; $\rho$, $\boldsymbol{u}$, $\tau$)
for timestep: $t = 0$ to $t_{\max}$ by $\delta t$ do
    Update $\tau$: Equations 23, 24, 25, 1 and 26
    Collide: Equation 28
    Streaming: Equation 22
    Apply boundary conditions
    Create particles depending on value of $t$  Reset $\boldsymbol{\Phi}$
    for Particle $\mathscr{P}_K$: $K = 1$ to $K_{\max}$ by 1 do
        Remove $\mathscr{P}_K$ if exits the computational domain
        for Lagrangian subcycle: $s = 1$ to $s_{\max}$ by 1 do
            Verlet integration: Equation 3
            Increase $\boldsymbol{\Phi}$ as follows:
                $$\boldsymbol{\Phi} \mathrel{+}= -\boldsymbol{F}_K\, \delta^3_{\boldsymbol{x},\, \boldsymbol{X}^{\text{next}}}$$
        end for
    end for
end for
```

3 Numerical experiments: setup

A cylindrical digester with an inclined base as [17, 23] is simulated (Fig. 1). A series of 12 nozzles, placed at equal distances along a circular manifold above the sloped bottom of the tank, is considered. Table 2 reports the geometric details. Mixing occurs through a circular manifold of 12 rectangular leaf-sparger nozzles with an equivalent diameter d_{noz}. A set of material numbers is defined to facilitate boundary condition treatment—in other words, a n integer is assigned to a given portion of computational domain. Following Fig. 1b, c, material number 0 is assigned to out-of-domain cells which do not undertake any lattice operation; 1 to the bulk and is subjected to lattice update but not to boundary conditions; 2 to the wall and floor boundaries and is subjected to both lattice update and bespoke boundary conditions; 3 to the liquid phase's free surface and is subjected only to bespoke boundary conditions.

The Reynolds number Re is evaluated as:

$$\text{Re} = \frac{U\, d_{\text{noz}}}{\nu_{\text{ref}}}. \tag{33}$$

The reference velocity U is the theoretical asymptotic rising bubble velocity, obtained by imposing a static balance between buoyancy and drag force using Morsi's model [43], multiplied by a heuristic correction set to 0.25 [35]. The reference kinematic viscosity ν_{ref} is evaluated through substitution of the reference shear rate $|\dot{\gamma}|_{\text{ref}}$ onto the rheology characteristic Eq. (1), with the former being defined as the shear rate occurring around an asymptotically rising bubble:

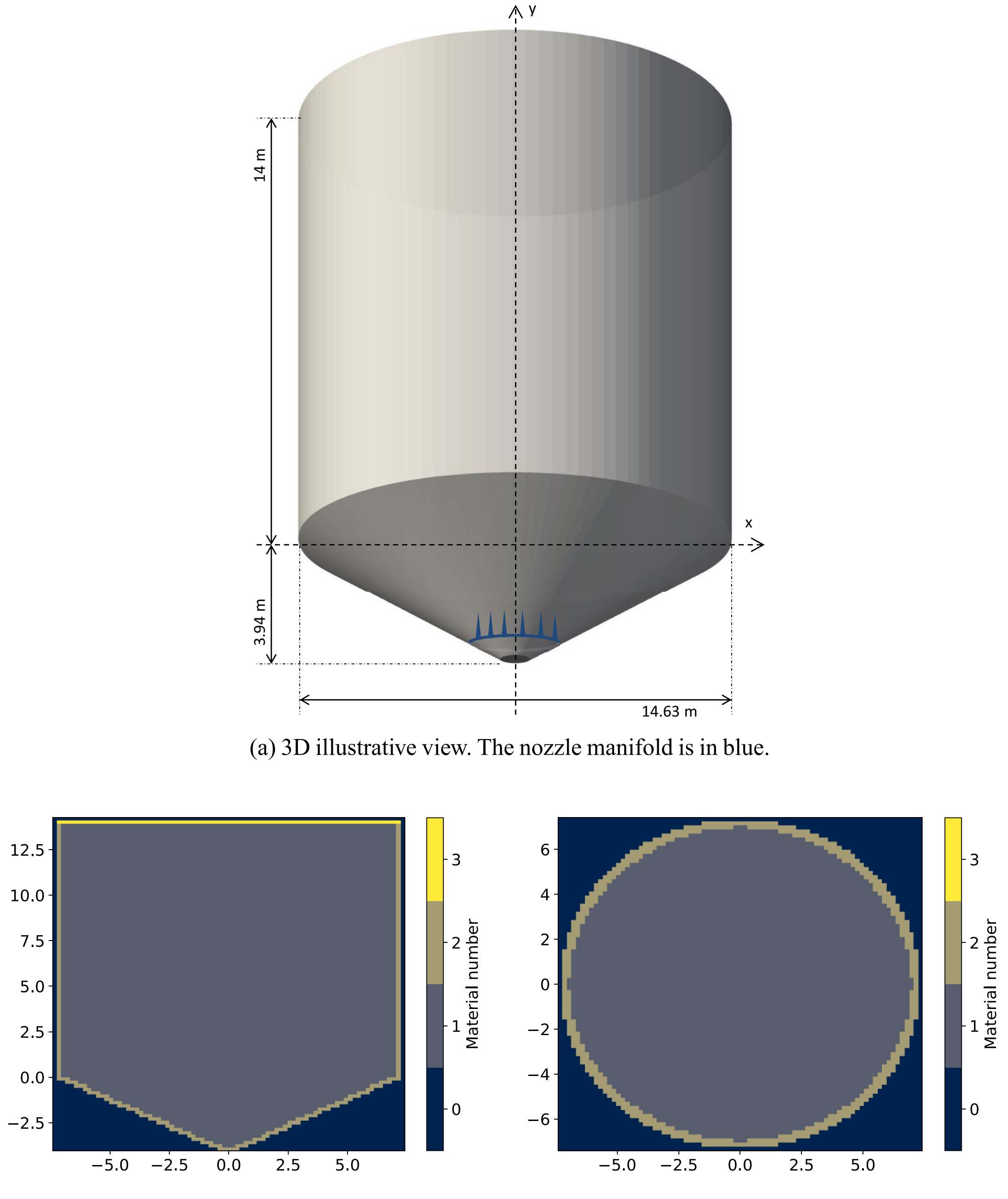
$$|\dot{\gamma}|_{\text{ref}} = \frac{U - u_{\text{surr}}}{2d} \simeq \frac{U}{2d} \tag{34}$$

because the velocity of the portion of liquid phase surrounding a rising bubble u_{surr} is negligible if compared to the

Table 2 Details of the digester geometry, from [17, 23]

External diameter	D_{ext}	14.63 m
Diameter at the bottom of the frustum	D_{int}	1.09 m
Cylinder height	h	14 m
Frustum height	h_0	3.94 m
Distance of original nozzle series from axis	R_1	1.83 m
Distance of new nozzle series from axis	R_2	5.49 m
Distance of nozzles from bottom	h_{noz}	0.3 m
Leaf sparger's equivalent diameter	d_{noz}	9.35 cm
Gas flow rate per nozzle	q_{noz}	2.3585 10^{-3} m^{3}s^{-1}

Sludge feed inlet is located on a side wall, at a height $h/4$ below the top sludge level. Courtesy of Severn Trent Water Ltd



(a) 3D illustrative view. The nozzle manifold is in blue.

(b) Material numbers, front wiew

(c) Material numbers, top wiew

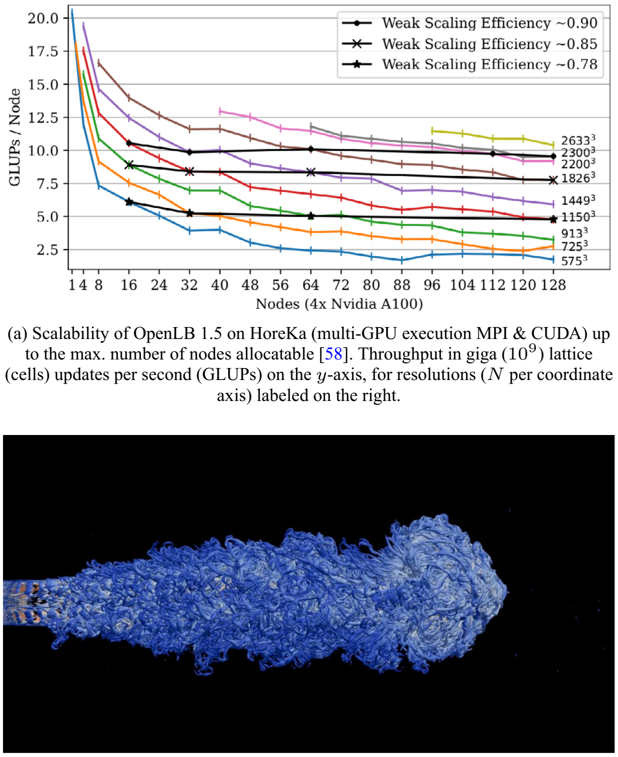
Fig. 1 Computational domain

asymptotic rising bubble velocity U. Re was found to span between 3600 (15 mm bubble diameter) and 6100 (50 mm bubble diameter).

The simulations are performed on D3Q27 cubic lattices with linear dimension n_x spanning from 30 to 160 lattice sites across the diameter D_{ext}, respectively corresponding to total numbers of cells spanning from 27,344 to 4,271,774. The dimensionless velocity $U_{\mathrm{LB}} \equiv \delta x/\delta t \cdot 1\,\mathrm{s/m}$ is set according to diffusive scaling:

$$U_{\mathrm{LB}} = U^0_{\mathrm{LB}} \frac{n^0_x}{n_x}, \tag{35}$$

with $U^0_{\mathrm{LB}} = 0.15$ and $n^0_x = 60$. Simulated time spans between 600 and 3600 s. Free-slip boundary condition is defined for the top free surface (material number 2), and Bouzidi no-slip for walls and bottom (material number 1). The maximum values of y^+ around the walls are found to be 30, 13 and 5 for 2.5%, 5.4% and 7.5% TS respectively, and the average value



(a) Scalability of OpenLB 1.5 on HoreKa (multi-GPU execution MPI & CUDA) up to the max. number of nodes allocatable [58]. Throughput in giga (10^9) lattice (cells) updates per second (GLUPs) on the y-axis, for resolutions (N per coordinate axis) labeled on the right.

(b) Q-Criterion iso-contours of GPU-based 3D turbulent nozzle flow simulation using 2.5 billion cells [58]. Colours indicate velocity magnitude from low (blue) to high (orange).

Fig. 2 OpenLB scaling-up on HoreKa

were respectively 4, 0.16 and 0.08; as such, no wall function is implemented. At the initial timestep, the fluid phase is quiescent and no bubbles are present in the system. Bubbles of diameter $15 < d < 50$ mm are introduced in the computational domain (viz., a new tuple $\mathscr{P}_K$ is defined) one at a time, at a position h_{noz} above the sloping base of the tank. The time interval between two bubble injection is defined as follows:

$$\Delta t_{\mathrm{inj}} = \frac{\pi d^3}{6\, q_{\mathrm{noz}}}. \tag{36}$$

As Δt_{inj} is in general not a multiple of δt, bubbles are in practice injected at the first subsequent timestep. The bubbles crossing the liquid surface are deleted.

3.1 OpenLB and Lattice-Boltzmann for platform-transparent saturation of modern HPC machines

The Lattice-Boltzmann is particularly suitable for extreme- and exa-scale simulations of fluid flows [49, 50] as, contrarily to conventional numerical techniques, close-to-optimal speedups are reachable. For example, LES of fluid flows in an injector with Lattice-Boltzmann-implemented in OpenLB (www.openlb.net) allow a speedup of 32 (simulation) and 424 (meshing) compared to Finite-Volume implementations in OpenFOAM (www.openfoam.org) on a similar setup with fixed accuracy [27].

The open-source C++ Lattice-Boltzmann framework OpenLB (https://www.openlb.net/) has been continuously developed since 2006. Krause (IANM/MVM/KIT). OpenLB contains a broad range of LBM implementations for several classes of partial differential equations (PDEs) for transport multi-physics including initial, boundary, and coupling methods [51]. Besides highly efficient simulations of turbulent, reactive, particulate and thermal fluid flow models, even coupled radiative transport or melting and conjugate heat transfer are realizable [27, 52–55].

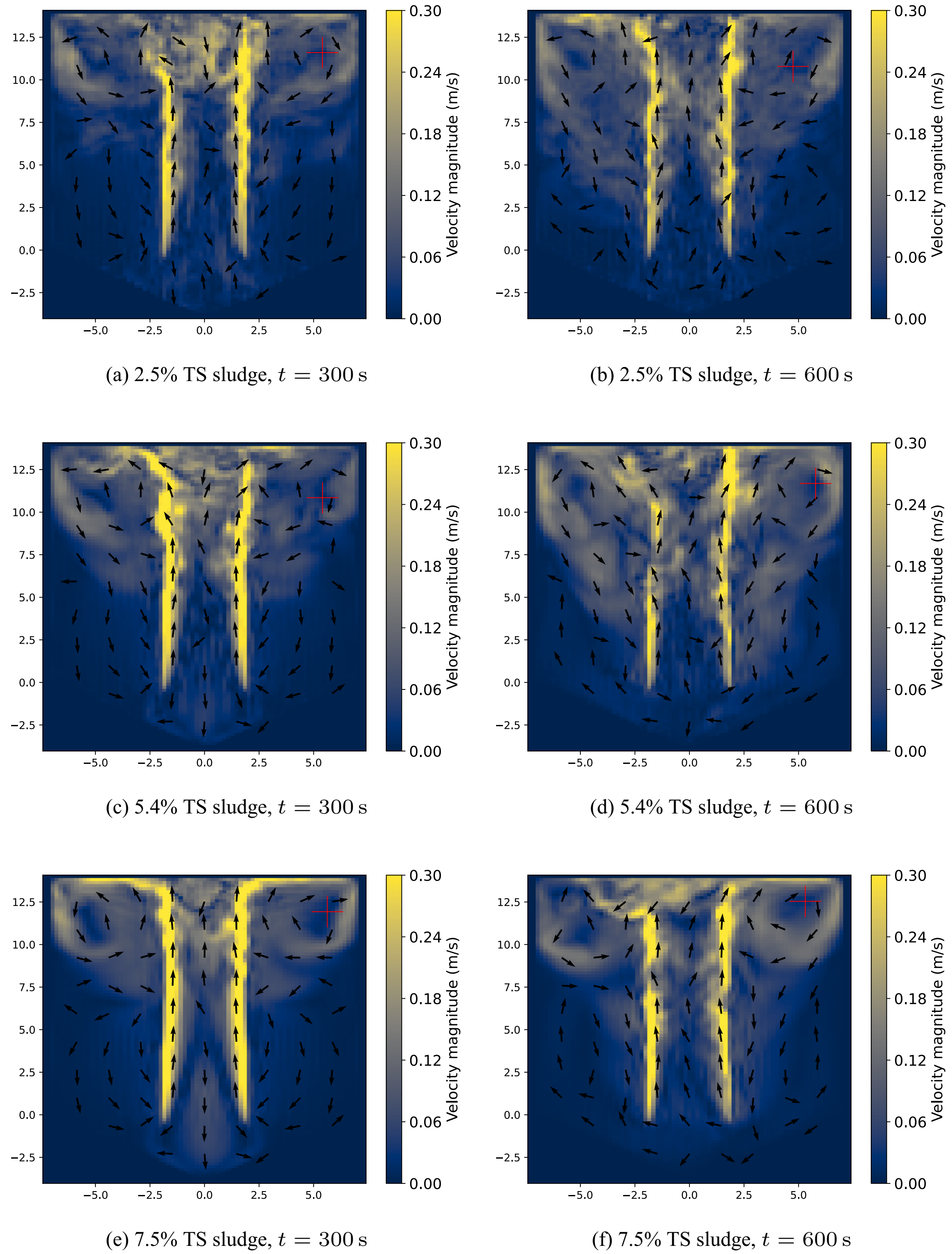
Specifically designed for large scale data generation, OpenLB supports efficient and platform-transparent executions, both on single-instruction-multiple-data (SIMD, vectorization) central processing units (CPUs), and general-purpose graphical processing units (GPGPUs) [56]. This is augmented by virtual memory manipulation and automatic code generation in order to reduce the arithmetic load per kernel [57], saturating the available memory bandwidth on current CPU and GPU targets [58]. The parallel efficiency of OpenLB was recently evaluated at up to perfect 1.0 (weak scaling) and very good 0.94 (strong scaling) on the HoreKa supercomputer (https://www.scc.kit.edu/dienste/horeka.php, 66/Top500 June 2022) at SCC (KIT), using both CPU-only and accelerated GPU partitions (Fig. 2). At the moment, a peak amount of 1.3×10^{12} grid cell updates per second is realizable on 128 accelerator nodes with 4x NVIDIA A100 GPUs each.

4 Numerical experiments: results

The simulations are run on one 40-core Lenovo ThinkSystem SR65 CPU. The computational expense spans between 500 and 45,000 CPUs (cumulative figure summed across the cores), depending on the run. OpenLB (www.openlb.net) version 1.3 [51, 59], a generalistic open-source library for parallel Lattice-Boltzmann modelling, equipped with optimized load-balancing strategies [60] and a vectorised A–A streaming algorithm [61], is used.

4.1 Flow patterns

Figure 3 shows snapshots of flow patterns at different times (t = 300 and 600 s) for different values of TS. The numbers around the plots and in all the following refer to the spatial coordinates as Fig. 1a. Qualitatively, high-velocity narrow areas, with velocity directed vertically upwards, are observed



(a) 2.5% TS sludge, $t = 300$ s

(b) 2.5% TS sludge, $t = 600$ s

(c) 5.4% TS sludge, $t = 300$ s

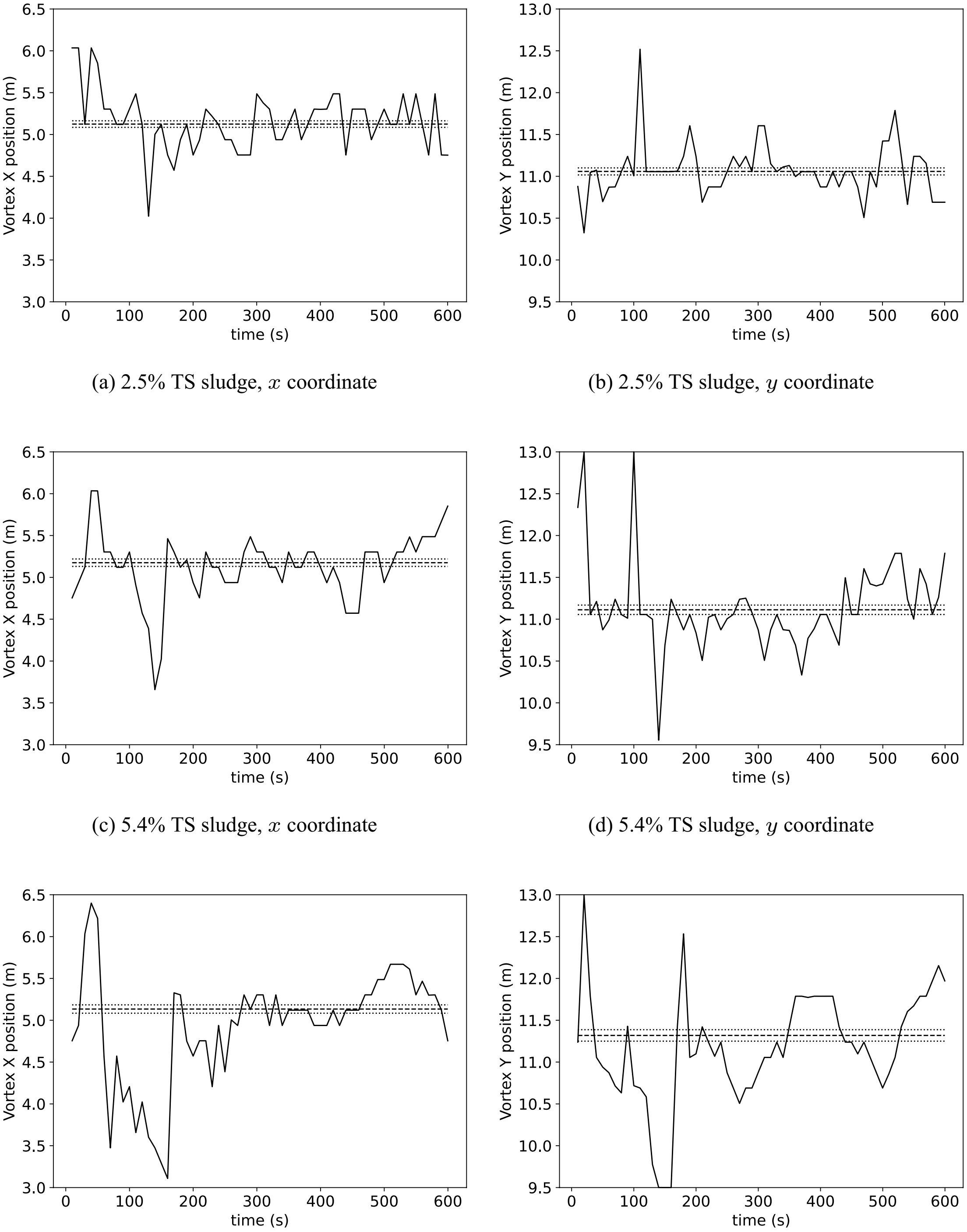
(d) 5.4% TS sludge, $t = 600$ s

(e) 7.5% TS sludge, $t = 300$ s

(f) 7.5% TS sludge, $t = 600$ s

Fig. 3 Flow patterns and vortex position, $n_x = 80$, 50 mm bubble size

above the nozzle locations; this corresponds to the drag effect exerted by rapidly-rising bubbles to the liquid phase. As the rapidly-rising flow approaches the liquid surface, it is deviated in a radial direction and then, as it approaches the walls, downwards. The flow is finally directed towards the rising columns, forming a toroidal vortex. Smaller-scale structures are also present, indicating the turbulent nature of the flow. The snapshots display fluctuations over time



(a) 2.5% TS sludge, x coordinate

(b) 2.5% TS sludge, y coordinate

(c) 5.4% TS sludge, x coordinate

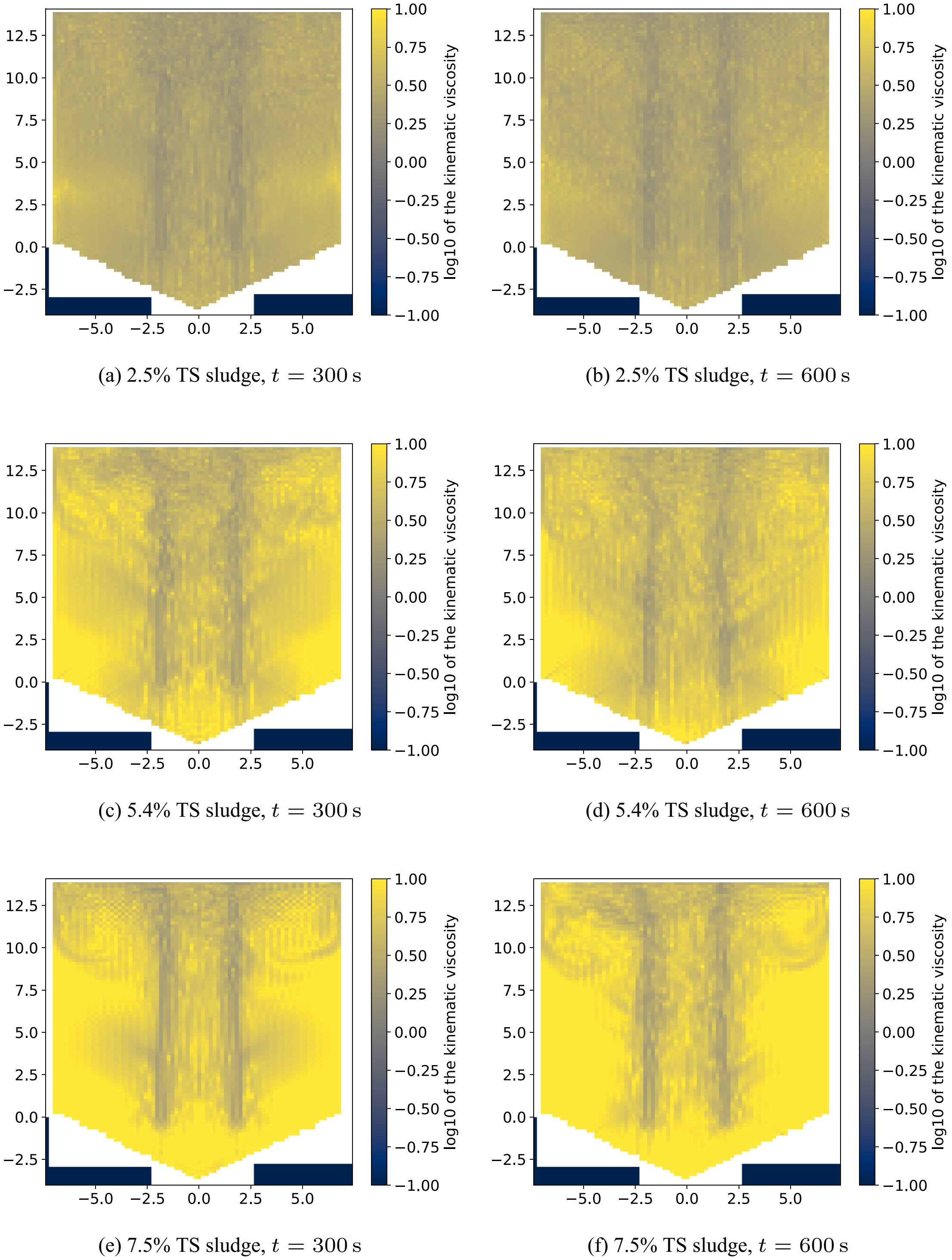
(d) 5.4% TS sludge, y coordinate

(e) 7.5% TS sludge, x coordinate

(f) 7.5% TS sludge, y coordinate

Fig. 4 Vortex coordinates over time, $n_x = 80$, 50 mm bubble size. Solid lines: instantaneous values. Dashed lines: averages, computed from 200 s onwards. Dotted lines: statistical standard errors computed from adapted standard deviations, computed from 200 s onwards

around this general description; this is in agreement with the fact that turbulence is modelled through LES. The large-scale flow patterns remain practically unchanged irrespective of the value of TS: the only qualitative aspect that varies depending of the value of TS is the relative prominence of the small-scale structures connected to turbulence; such structures tend to smooth out as the TS increases. This is in



(a) 2.5% TS sludge, $t = 300$ s

(b) 2.5% TS sludge, $t = 600$ s

(c) 5.4% TS sludge, $t = 300$ s

(d) 5.4% TS sludge, $t = 600$ s

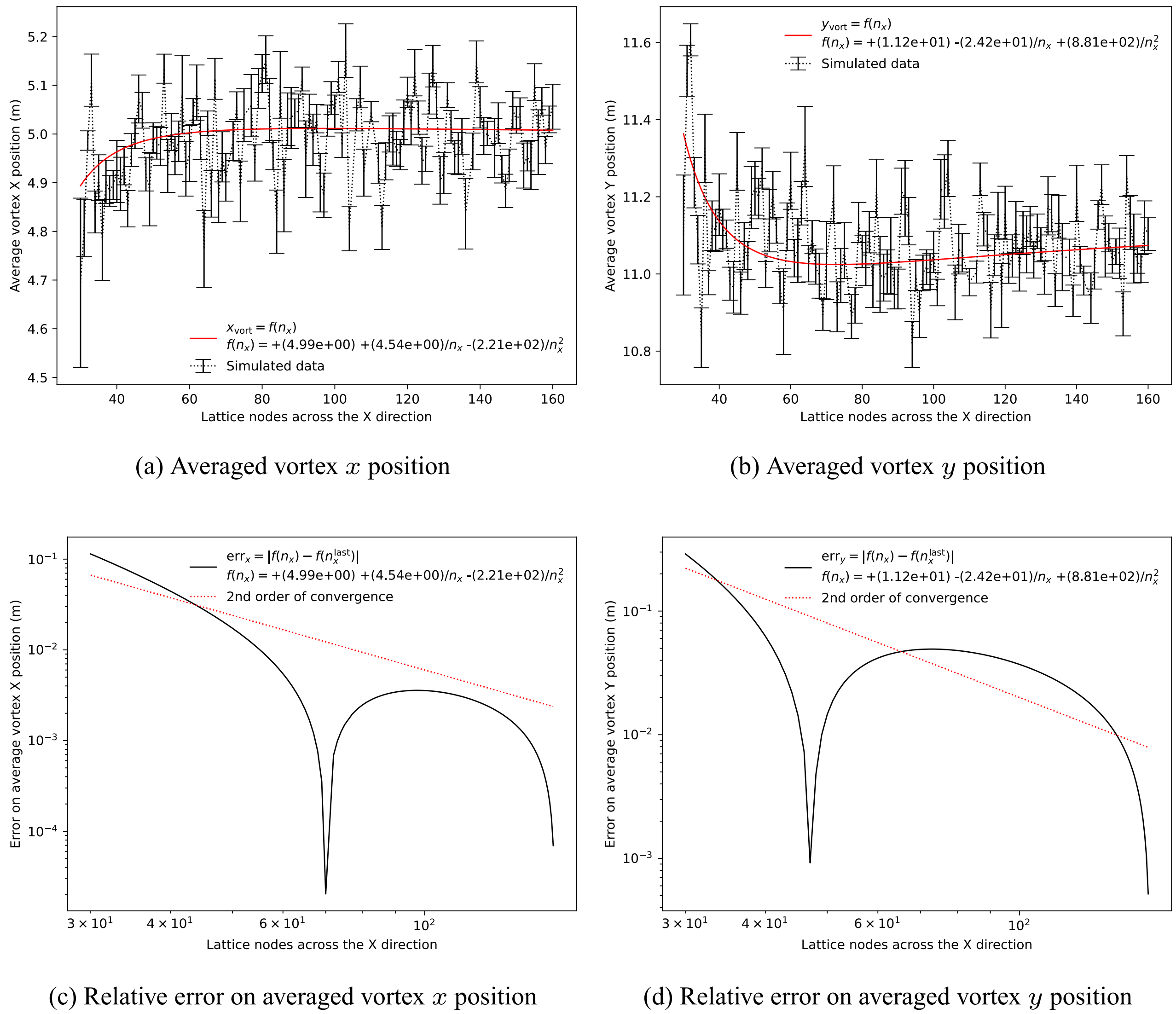
(e) 7.5% TS sludge, $t = 300$ s

(f) 7.5% TS sludge, $t = 600$ s

Fig. 5 Apparent viscosity over characteristic viscosity, $n_x = 80$, 50 mm bubble size

agreement with the fact that Re decreases (and therefore, the flow is less turbulent) as TS increases.

A quantitative description is offered by the coordinates of the vortex. For every timestep, the vortex is found by searching the *x*–*y* position minimizing the velocity magnitude within the square window $[3.0,\ 6.5\,\mathrm{m}] \times [9.5,\ 13.0\,\mathrm{m}]$ starting from an initial guess of $(5.0,\ 11.0\,\mathrm{m})$. The resulting vortex position is marked with a red cross in the snapshot of Fig. 3, and tracked over time in Fig. 4. As in the case of the flow patterns, the vortex position oscillates around an average value, as expected



(a) Averaged vortex x position

(b) Averaged vortex y position

(c) Relative error on averaged vortex x position

(d) Relative error on averaged vortex y position

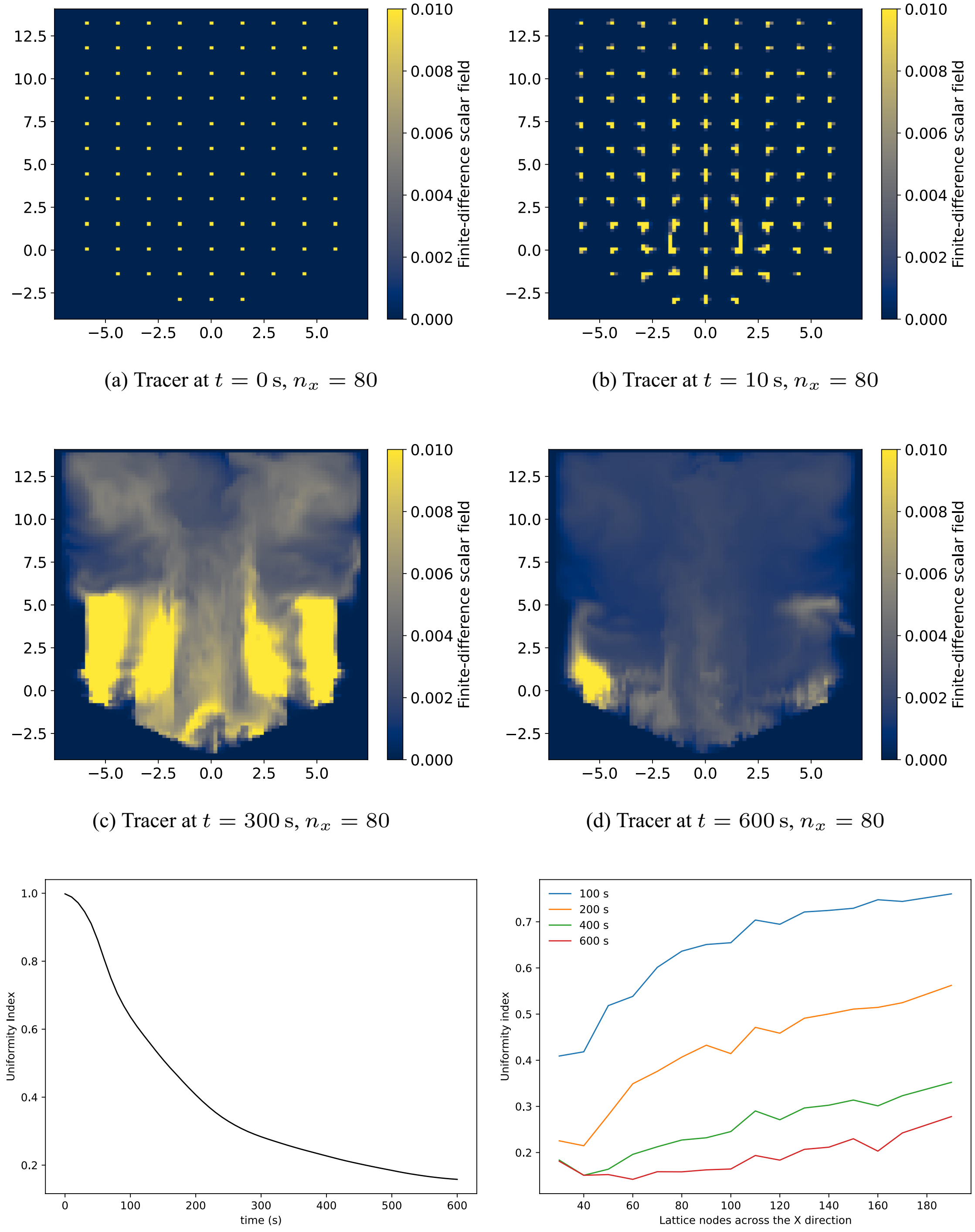
Fig. 6 Averaged vortex position and relative error, 2.5% TS sludge, 50 mm bubble size. Average performed for $t \geq 200$ s

from a LES model. For all the TS values, the stationary-oscillating regime is reached after an initial transient period of around ∼ 200 s. Both the average and the statistical standard error (the latter being computed on the same number of samples for all the runs) have similar values for all the values of TS, thereby confirming the qualitative observation of unchanging large-scale flow patterns irrespective of the value of TS.

In Fig. 5, snapshots of the apparent viscosity are reported. The viscosity patterns become more evident and less uniform as TS rises, indicating a more prominent power-law behaviour for higher values of TS.

4.2 Grid independence

The grid independence test is reported in Fig. 6. The test is performed over the vortex coordinate, with the average being taken for each run by averaging between the instantaneous values for 200 s $\leq t \leq$ 600 s and errorbars as in Fig. 4. The curves display rapid oscillations. Despite this, a best fit against a function of the type $f\left(n_x\right) = a + b/n_x + c/n_x^2$ shows that the values oscillate around an approximatively horizontal asymptote for $n_x \gtrsim 80$, with a relative error (defined as the absolute value of the relative difference between the datum and the series' last value) comprised between 0.2 and 2%. It is therefore possible to consider the results as grid independent for $n_x > 80$, and the value of $n_x = 80$ is chosen for all the other runs reported in this work as the best compromise between accuracy and numerical expense.



(a) Tracer at $t = 0$ s, $n_x = 80$

(b) Tracer at $t = 10$ s, $n_x = 80$

(c) Tracer at $t = 300$ s, $n_x = 80$

(d) Tracer at $t = 600$ s, $n_x = 80$

(e) Uniformity index over time, $n_x = 80$

(f) Uniformity index for different n_x

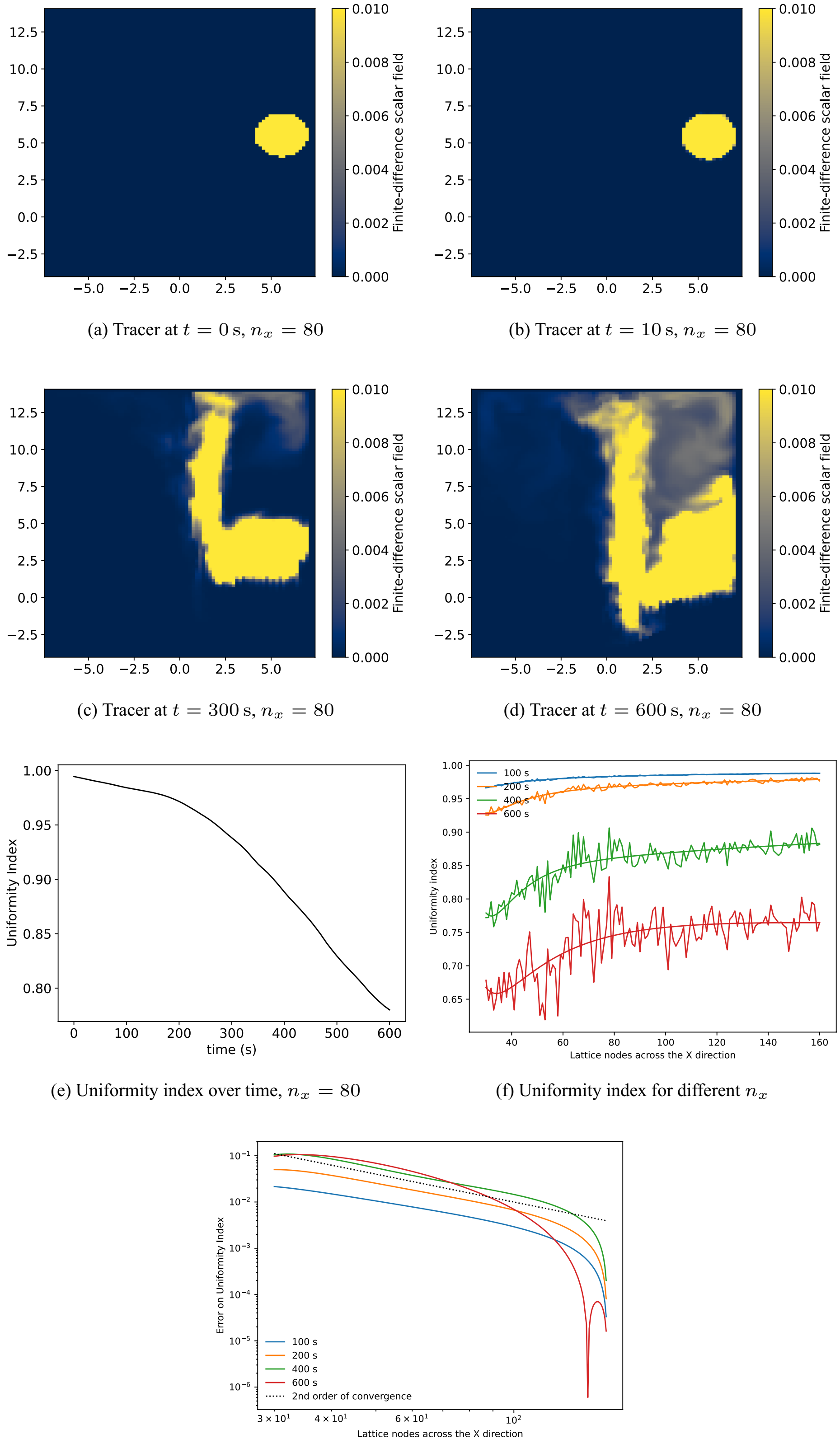
Fig. 7 "Sparse" results, 2.5% TS sludge, 50 mm bubble size

4.3 Uniformity Index

In [20, 23], the uniformity index (UI), as introduced in [9], was found to be the best single-number quantitative criterion to assess mixing. Given a numerical macroscopic scalar field ("tracer") ϕ evolving according to an advection–diffusion equation with zero diffusivity:

$$\partial_t \phi + (\boldsymbol{u} \cdot \boldsymbol{\nabla})\phi = 0, \tag{37}$$

Fig. 8 “Ball” results, 2.5% TS sludge, 50 mm bubble size



(a) Tracer at $t = 0$ s, $n_x = 80$

(b) Tracer at $t = 10$ s, $n_x = 80$

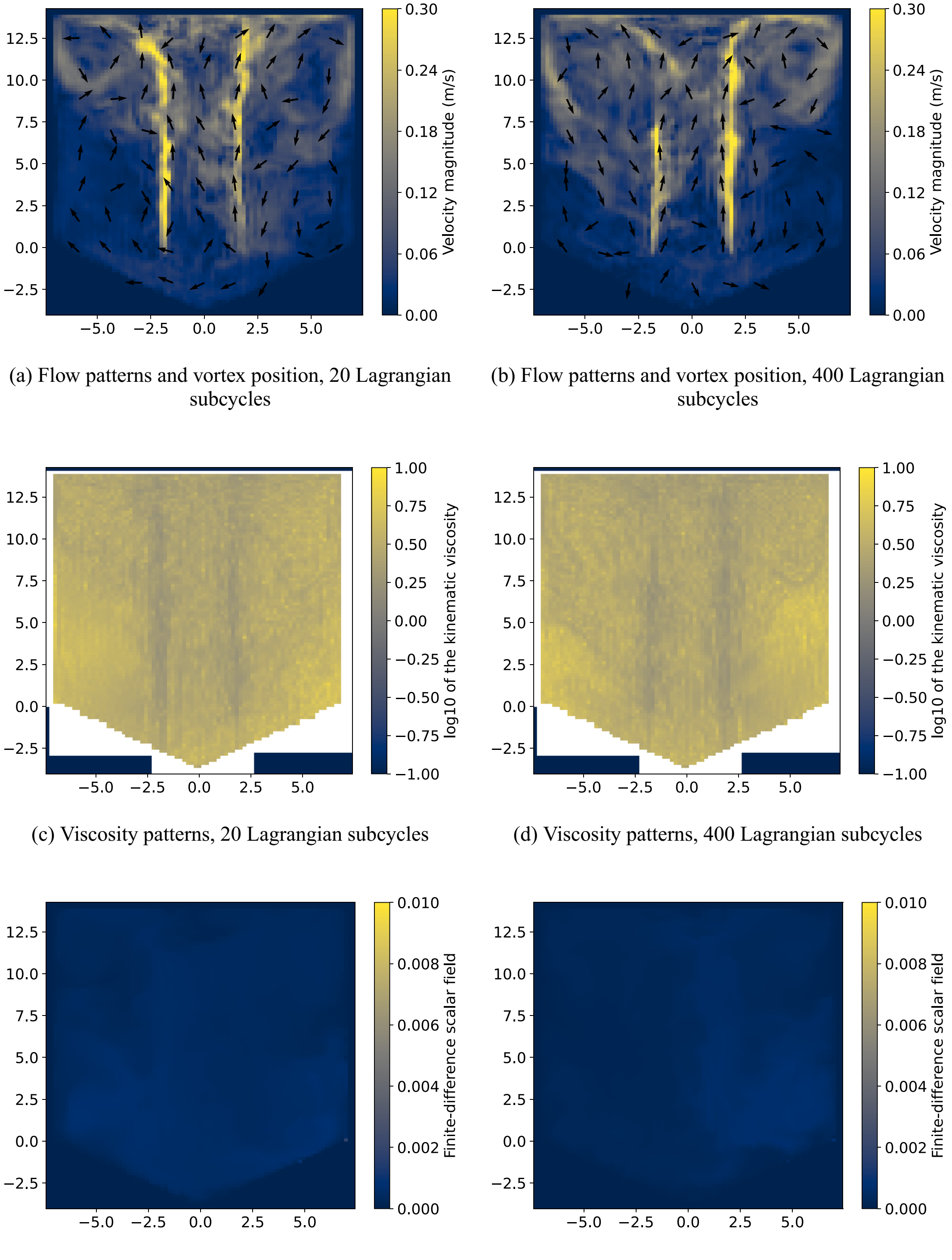
(c) Tracer at $t = 300$ s, $n_x = 80$

(d) Tracer at $t = 600$ s, $n_x = 80$

(e) Uniformity index over time, $n_x = 80$

(f) Uniformity index for different n_x

(g) Relative error on the uniformity index for different n_x



(a) Flow patterns and vortex position, 20 Lagrangian subcycles

(b) Flow patterns and vortex position, 400 Lagrangian subcycles

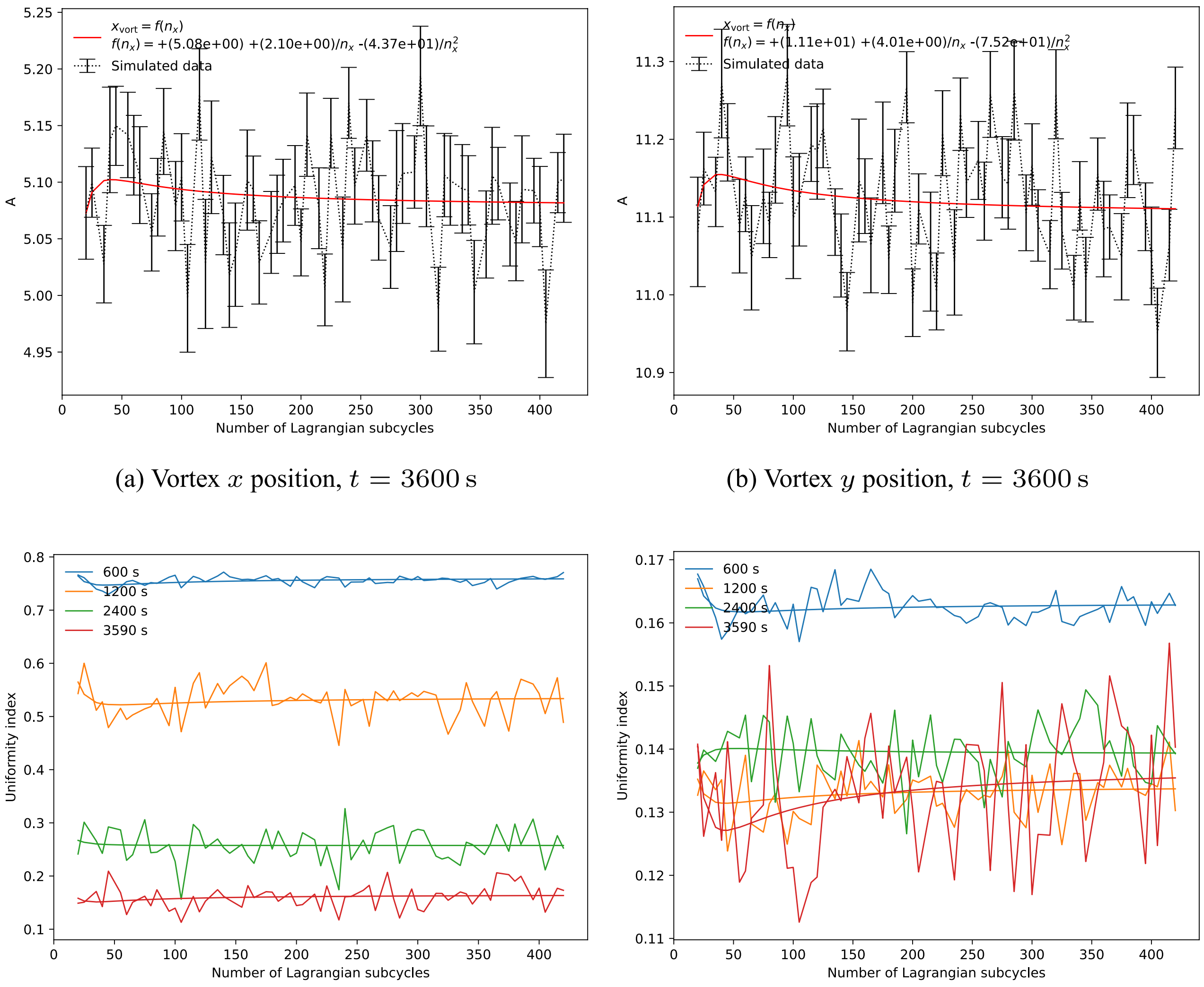
(c) Viscosity patterns, 20 Lagrangian subcycles

(d) Viscosity patterns, 400 Lagrangian subcycles

(e) Tracer, “Ball” results, 20 Lagrangian subcycles

(f) Tracer, “Ball” results, 400 Lagrangian subcycles

Fig. 9 Flow patterns, vortex position, viscosity patterns and tracer for different numbers of Lagrangian subcycles at $t = 3600$ s. 2.5% TS sludge, $n_x = 80$, 50 mm bubble diameter



(a) Vortex x position, $t = 3600$ s

(b) Vortex y position, $t = 3600$ s

(c) Uniformity index, “Ball” results

(d) Uniformity index, “Sparse” results

Fig. 10 Dependence on number of Lagrangian subcycles, $n_x = 80$, 2.5% TS sludge, 50 mm bubble size

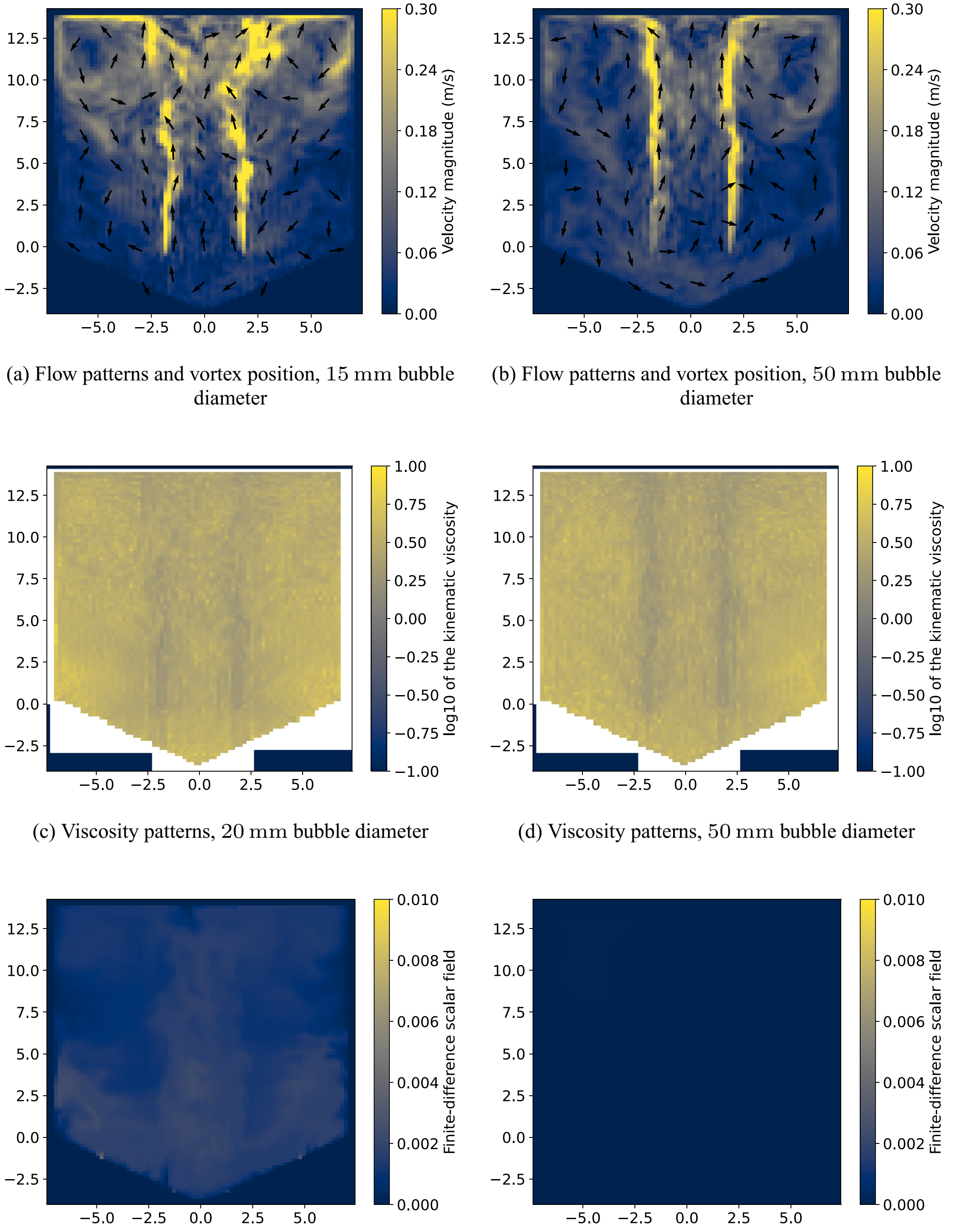
the uniformity index is defined in a cubic lattice as:

$$\mathrm{UI} := \frac{1}{2\langle\phi\rangle}\langle|\phi - \langle\phi\rangle|\rangle, \tag{38}$$

with $\langle\cdot\rangle$ being the average over the lattice sites. As a consequence of how it is defined, the value of UI varies between 0 for perfect mixing to 1 for complete inhomogeneity. The tracer ϕ is solved through an explicit Finite-Volume method as in [62]. The first-order upwind scheme for the advection term is preferred over the central second-order for the sake of numerical stability. No negative-diffusivity correction is set [62].

Figure 7 shows the evolution of the tracer ϕ over time up to $t = 600$ s, when ϕ being initialised to 0 almost everywhere, and to 1 in single cells evenly distributed throughout the computational domain (Fig. 7a). This initial condition is labelled as “Sparse”. After 10 s from the start of the simulation (Fig. 7b), the positions of the pockets with non-zero ϕ remain unaltered, showing that the flow patterns have not yet developed enough to displace them away from their original positions. However, numerical diffusion is evident, and no improvement is observed when negative anti-diffusion is set—this is the reason why anti-diffusion is not set in the work reported here. As a result, despite the evolution towards homogenisation (Fig. 7c, d) and decrease of the uniformity index (Fig. 7e), the latter displays an evident grid dependence with its value at given timesteps depends on grid size (Fig. 7f), as numerical diffusion depends on δx [62].

In Fig. 8, another initial condition for the scalar tracer ϕ is proposed, under the name of “Ball”. ϕ is initialised



(a) Flow patterns and vortex position, 15 mm bubble diameter

(b) Flow patterns and vortex position, 50 mm bubble diameter

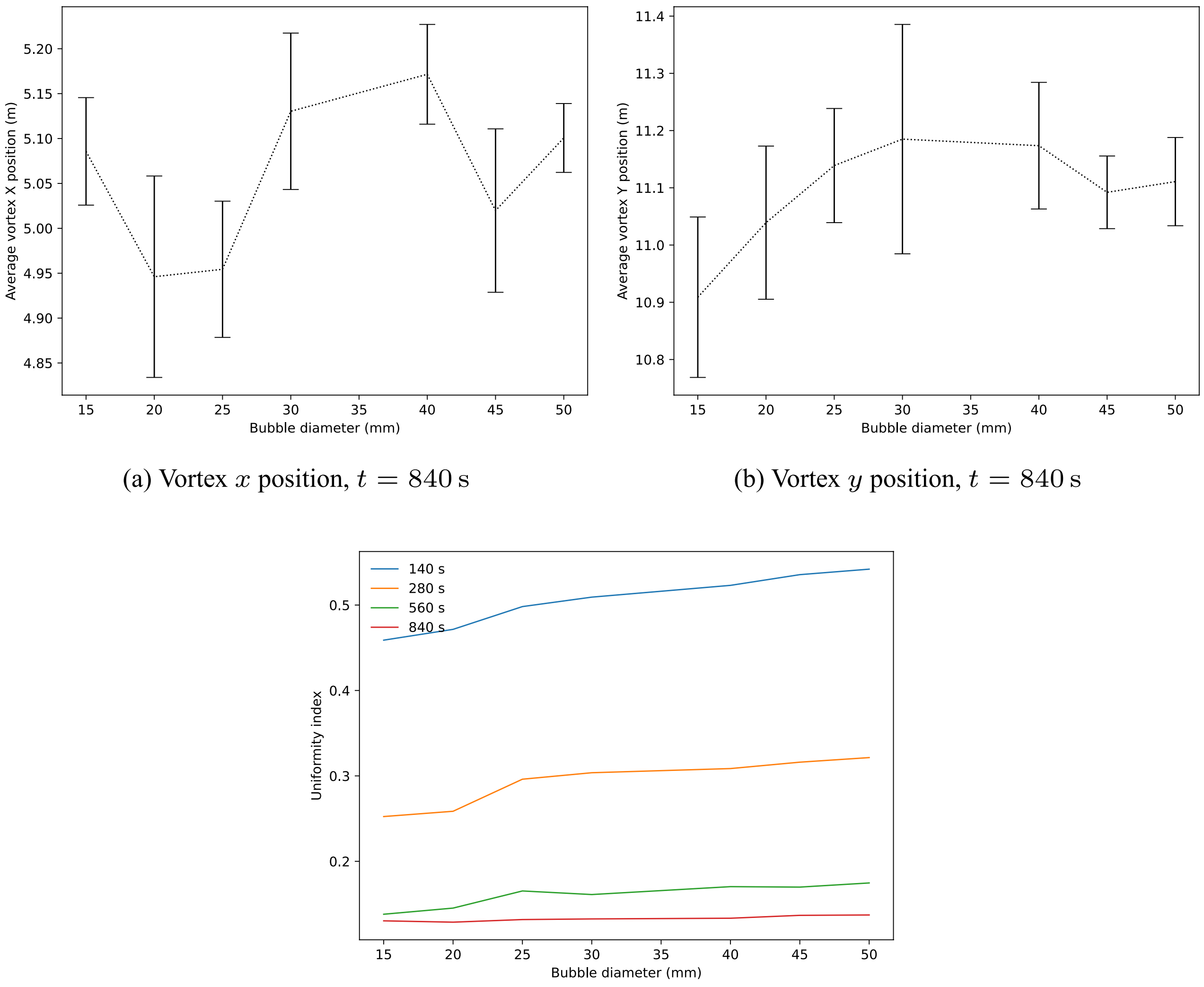
(c) Viscosity patterns, 20 mm bubble diameter

(d) Viscosity patterns, 50 mm bubble diameter

(e) Tracer at $t = 840$ s, “Sparse” results, 20 mm bubble diameter

(f) Tracer at $t = 840$ s, “Sparse” results, 50 mm bubble diameter

Fig. 11 Flow patterns, vortex position, viscosity patterns and tracer for different bubble sizes at $t = 840$ s. 2.5% TS sludge, $n_x = 80$, 100 Lagrangian subcycles



(a) Vortex x position, $t = 840$ s

(b) Vortex y position, $t = 840$ s

(c) Uniformity index

Fig. 12 Dependence on bubble size. “Sparse” results, $n_x = 80$, 2.5% TS sludge

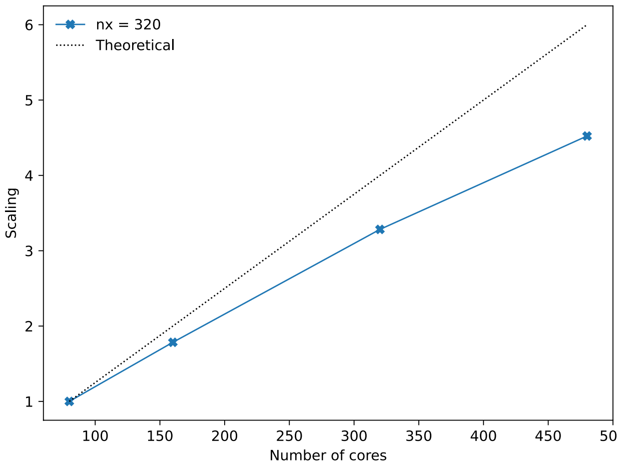


Fig. 13 Strong scaling, 2–12 nodes, each with 40 CPUs

to 0 everywhere and to 1 in a sphere sideways (Fig. 8a), around the location of the inlet [17, 23]. At the start of the run, the bulk of non-zero concentration field appears to be advected downwards by the flow patterns with marginal diffusion phenomena (Fig. 8b), until the tracer finally starts to be spread across the computational domain, after being brought in contact with the rising bubble column (Fig. 8c, d). Only at that point does the uniformity index start to fall significantly (Fig. 8e). This description is in agreement with [23], where new sludge just injected into the system finds itself in a position analogous to the scalar tracer described here and in the cited article, and initially undertakes only a poor level of mixing.

The observation of negligible diffusion phenomena (Fig. 8b) is corroborated by the analysis of the behaviour of the uniformity index for different values of n_x (Fig. 8f) fitted against a function of the type $\sum_n n_x^{-n}$, $0 \leq n \leq 4$, and the corresponding relative error (Fig. 8g). Despite oscillations, a clear horizontal trend is observed at all the time snapshots for $n_x \geq 80$, in agreement with the conclusions of Sect. 4.2. Above that threshold, the uniformity index appears to converge to approximately second order, and its relative error falls below 5%. This observation: (i) indicates that the "Ball" configuration is less affected by numerical diffusivity than the "Sparse", and is therefore the most suitable to investigate the model's behaviour under variation of its parameters; and (ii) further confirms mesh independence for $n_x \geq 80$ wherever the initial conditions allow one to ignore numerical diffusivity.

4.4 Lagrangian subcycles

Figure 9 reports qualitative snapshots, taken at 3600 s, of flow patterns, viscosity and scalar tracer field ϕ, for two different numbers of Lagrangian subcycles (viz., 20 and 400). No qualitative difference between the results of the different numbers of Lagrangian timesteps can be identified. Further, in Fig. 10, vortex position and the value of the uniformity index at different times are reported as a function of the number of Lagrangian subcycles, for both the "Sparse" and "Ball" initial conditions. A best fit of the form $\mathrm{UI} = \sum_i a_i s^{-n}$, $0 \leq i \leq 2$, is reported. Apart from local oscillations attributable to noise, the relevant parameters are observed to be independent from the number of Lagrangian subcycles, within a relative error of $\sim 10^{-2}$.

4.5 Bubble size

Figure 11 reports qualitative snapshots, taken at 840 s, of flow patterns, viscosity and scalar tracer field ϕ, for two different bubble sizes (viz., 15 and 50 mm). Flow patterns snapshots (Fig. 11a, b) display qualitatively more intense flow patterns in simulations with smaller bubble size. The latter also present a more prominent presence of small-scale structures, indicating a higher level of turbulence. Conversely, viscosity flow patterns (Fig. 11c, d) and final tracer ϕ distribution (Fig. 11e, f) are not found to be affected by the choice of bubble size.

From these observations, it can be argued that smaller bubbles manage to mix the system faster thanks to more intense flow patterns and turbulence intensity, whilst the final level of mixing remains unaffected. This picture is confirmed by the quantitative picture (Fig. 12). The vortex positions (Fig. 12a, b) do not provide relevant information due to the large uncertainty bars. However, the uniformity index (Fig. 12c), for early timesteps (140 s) clearly shows that smaller bubble sizes produce lower UI values, with a difference of UI value between 50 and 15 mm bubble sizes of 0.083, corresponding to around 18% of 15 mm bubble size's UI value. This difference decreases for further timesteps, with the relative error at the final timestep being below 10^{-2}.

4.6 Scaling-up

A strong scaling-up test was performed (Fig. 13) on up to 12 Intel Xeon Gold Skylake cores, each containing 40 CPU cores. The simulations were run selecting $n_x = 320$ (corresponding to over 32,113,000 cells), 50 mm bubble size, for 60 s simulated time. The scale-up clearly shows an increasing trend throughout the whole range of number of cores, with a loss of efficiency of less than 25% at the highest number of cores. Such loss of efficiency is likely due to the non-scalable inter-core communication taking place at the streaming phase; and to the Lagrangian particles. Indeed, within OpenLB, a Lagrangian particle is simulated by the CPU core responsible for the subdomain where the given particle is located; this means that non-scalable operations occur when a particle crosses a subdomain division as its data are communicated from the target to the destination CPU; and asymmetric load balancing occurs when the particles are not distributed uniformly throughout the computational domain (which is the case for the simulation work described within this article).

5 Discussion

The results reported in Sect. 4.2 show that the flow patterns are grid independent for $n_x \gtrsim 80$. However, the noise manifesting itself as oscillations in Fig. 6 makes the determination of the order of convergence challenging. Similar considerations hold for the uniformity index: once the grid-dependent effect of numerical diffusion is singled out, UI displays grid independence for a similar value of $n_x \gtrsim 80$, with analogue considerations about noise (Fig. 8f).

The results reported in Sect. 4.4 indicate that the simulations are independent of the number of Lagrangian subcycles. As such, the number of Lagrangian subcycles remains a non-physical tuning parameter, to be tuned depending on the particular application and the specific bubble size, to strike the best balance between numerical expense (i.e., the numerical expense is proportional to the number of Lagrangian subcycles), and stability (i.e., the Lagrangian solver becomes unstable if the number of Lagrangian timesteps falls below a threshold dependent on specific application and bubble size).

In Sect. 4.5, bubble size is shown to affect flow the patterns and the transient evolution of the uniformity index, but not the final value of UI—or in other words, the prediction of steady-state mixing quality. This is in agreement with the observations on the same geometry, applying a second-order Finite-Volume Reynolds-Averaged-Navier–Stokes model with Reynolds-stress turbulence model and the same power-law rheology model, on OpenFOAM 2.3.0 (www.openfoam.org/version/2-3-0) [17, 20, 23].

Overall, the Lattice-Boltzmann predictions of flow patterns, uniformity index and degree of mixing produced by the numerical work presented here are in agreement with observations from the above-mentioned Finite-Volume CFD work [17, 20, 23]. However, the Lattice-Boltzmann model presented here offers two critical advantages over the previous Finite-Volume, specifically:

(i) *Numerical efficiency*. Lattice-Boltzmann runs with $n_x = 80$ need around 10,500 CPUs to run over grids of 536,171 cells for 600–3600 s simulated time, for a specific resource usage of 5.4–33·10^{-6} CPUs per s per cell. In contrast, the Finite-Volume analogue took around 13,500,000 CPUs for running over a grid of 394,400 cells for 300 s simulated time on 36 Intel Xeon E5-2690 v3 Haswell (2.6 GHz) cores, for a specific resource usage of 0.11 CPUs per s per cell. This makes the Lattice-Boltzmann model over 1000 times faster than the Finite-Volume model previously used in the literature to solve this problem. Similarly, the LES simulations performed in [24] for a full-scale mechanically-mixed digester were conducted on 188,289 cells for around 60 s simulated time and took 1,205,200 CPUs, for a specific resource usage of 0.11 CPUs per s per cell—the same value as [17, 20, 23]. Although the present Lattice-Boltzmann work and the previous Finite-Volume are conducted on different machines, the large performance difference between this and the previous models, as well as the similarity in performance between [17, 20, 23] and [24], allow us to confidently rule out any detrimental effect attributable to differences in hardware.

(ii) *Resolution*. Lattice-Boltzmann's best balance between numerical efficiency and precision was found to be $n_x = 80$, for 536,171 cells (Sect. 4.2). By contrast, the same balance returned 98,420 cells for a $\pi/6$ wedge of the computational domain in the Finite-Volume model. This shows that the Lattice-Boltzmann model comfortably allows much finer grids than the Finite-Volume, thereby producing much more detailed predictions. In fact, the Lattice-Boltzmann model allowed a level of detail of the flow patterns, especially concerning smaller-scale turbulent patterns (Fig. 11a, b), which would be unachievable in the Finite-Volume results reported in [17, 23] because of the above-mentioned difference in numerical efficiency and Finite-Volume's scaling-up problems [22].

(iii) *Scaleup*. Scaling-up performance is discussed in Sect. 4.6. Despite the limiting factors discussed therein, the strong-scaling plot (Fig. 13) clearly shows an increasing trend, without reaching a plateau at 480 cores (it was not possible to perform simulations with more cores due to hardware limitations). This constitutes a notable improvement over previous Finite-Volume models, as the plateau was previously reached at 36 cores [17, 23].

Considering the usage of cache memory. Lattice-Boltzmann methods are memory-intensive: the model presented here uses 32 floats per lattice cell, much more than the previous Finite-Volume (which uses 5 floats per cell). This high memory usage does not usually pose a limitation to numerical performance as access to cache memory is fast—and indeed, it did not pose a limitation to the performance of the runs descripted here. However, care should be taken in checking that the available cache memory can meet a run's memory requirement, before performing the run itself. OpenLB maps the computational domain into a lattice, or `SuperLattice` object, which is in turn divided into sub-lattices, or `BlockLattice` objects. Each `BlockLattice` is loaded onto the cache memory of a single CPU [51]. As such, increasing the number of cores usually resolves possible problems of shortage of memory. Considering the effectiveness of this strategy in preventing memory over-usage, and considering that memory usage does not constitute a bottleneck to numerical performance, we did not perform a comparison between memory usage of the model presented here, and its Finite-Volume predecessors [17, 23].

6 Conclusions

A Lattice-Boltzmann LES model of a full-scale, biogas-mixed anaerobic digester has been presented for the first time. Scaleup, convergence and the effects of bubble size and number of Lagrangian subcycles on the model predictions concerning the digester's hydrodynamics have been assessed.

A comparison between Lattice-Boltzmann and Finite-Volume on an analogue applications shows that the former is over 1,000 times more computationally efficient, allows resolution of flow patterns in much more detail, and allow a feasible, resource-effective usage of LES in anaerobic digestion modelling for the first time. Thus, the work presented here is a comparison between two specific models being used to solve a problem of significant interest and relevance to the wastewater industry. It should not be considered as a benchmark in a strict sense—such benchmark work would require tests on a wide range of models being conducted on the same hardware running under the same conditions, and is out of the scope of this research. Notwithstanding this limitation, it can be concluded that the Lattice-Boltzmann is a more convenient modelling choice for full-scale gas mixing in anaerobic digestion, than the most common second-order Finite-Volume approaches.

Industries and consultancies will be able to use the results described here as guidance to improve full-scale digesters' mixing efficiency via UI maximization. In this respect, the code used here will be available in a future official release of the OpenLB package.

Acknowledgements The authors gratefully acknowledge Peter Vale and Severn Trent Ltd. for kindly providing the details of the digester geometry. This research was funded by the UKRI Engineering and Physical Sciences Research Council via EPSRC Grant (EP/R01485X/1, Computational Methods for Anaerobic Digestion Optimization, "CoMAnDO"). The authors would like to thank the University of Bradford and University of Liverpool's High Performance Computing (HPC) Services, where part of the computational work was carried out (i.e., the scaling-up was performed at the University Liverpool, and the remainder at the University of Bradford). Krause's contribution was made possible by funding from the German Research Foundation (DFG 436212129). No contribution other than financial support came from the above-mentioned funding bodies.

Data Availability Not applicable.

Declarations

Conflict of interest No interests to declare.

References

1. WWAP (World Water Assessment Programme) (2012) The United Nations world water development report 4: managing water under uncertainty and risk. Technical report. UNESCO, Paris. ISBN: 978-92-3-104235-5
2. European Environment Agency (2015) Waterbase—UWWTD: urban waste water treatment directive—reported data. Technical report
3. Eurostat (2021) Sewage sludge production and disposal. http://appsso.eurostat.ec.europa.eu/nui/show.do?lang=en &dataset=env_ww_spd. Accessed 8 February 2021
4. Owen WF (1982) Anaerobic treatment processes. In: Energy in wastewater treatment. Prentice-Hall, Inc., Englewood Cliffs, NJ
5. Kress P, Nägele HJ, Oechsner H, Ruile S (2018) Effect of agitation time on nutrient distribution in full-scale CSTR biogas digesters. Bioresour Technol 247:1–6. https://doi.org/10.1016/j.biortech.2017.09.054
6. Vesvikar MS, Al-Dahhan MH (2005) Flow pattern visualization in a mimic anaerobic digester using CFD. Biotechnol Bioeng 89(6):719–732. https://doi.org/10.1002/bit.20388
7. Karim K, Thoma GJ, Al-Dahhan MH (2007) Gas-lift digester configuration effects on mixing effectiveness. Water Res 41(14):3051–3060. https://doi.org/10.1016/j.watres.2007.03.042
8. Meroney RN, Colorado PE (2009) CFD simulation of mechanical draft tube mixing in anaerobic digester tanks. Water Res 43(4):1040–1050. https://doi.org/10.1016/j.watres.2008.11.035
9. Terashima M, Goel R, Komatsu K, Yasui H, Takahashi H, Li YY, Noike T (2009) CFD simulation of mixing in anaerobic digesters. Biores Technol 100(7):2228–2233. https://doi.org/10.1016/j.biortech.2008.07.069. **(ISBN: 0960-8524)**
10. Wu B (2010) CFD simulation of gas and non-Newtonian fluid two-phase flow in anaerobic digesters. Water Res 44(13):3861–3874. https://doi.org/10.1016/j.watres.2010.04.043
11. Bridgeman J (2012) Computational fluid dynamics modelling of sewage sludge mixing in an anaerobic digester. Adv Eng Softw 44(1):54–62. https://doi.org/10.1016/j.advengsoft.2011.05.037
12. Sindall RC, Bridgeman J, Carliell-Marquet C (2013) Velocity gradient as a tool to characterise the link between mixing and biogas production in anaerobic waste digesters. Water Sci Technol 67(12):2800–2806. https://doi.org/10.2166/wst.2013.206 ISBN: www.iwaponline.com/wst/06712/wst067122800.htm
13. Craig KJ, Nieuwoudt MN, Niemand LJ (2013) CFD simulation of anaerobic digester with variable sewage sludge rheology. Water Res 47(13):4485–4497. https://doi.org/10.1016/j.watres.2013.05.011. **(ISBN: 1879-2448 (Electronic) 0043-1354 (Linking))**
14. Dapelo D, Alberini F, Bridgeman J (2015) Euler-Lagrange CFD modelling of unconfined gas mixing in anaerobic digestion. Water Res 85:497–511. https://doi.org/10.1016/j.watres.2015.08.042
15. Hurtado FJ, Kaiser AS, Zamora B (2015) Fluid dynamic analysis of a continuous stirred tank reactor for technical optimization of wastewater digestion. Water Res 71:282–293. https://doi.org/10.1016/j.watres.2014.11.053
16. Zhang Y, Yu G, Yu L, Siddhu MAH, Gao M, Abdeltawab AA, Al-Deyab SS, Chen X (2016) Computational fluid dynamics study on mixing mode and power consumption in anaerobic mono- and co-digestion. Biores Technol 203:166–172. https://doi.org/10.1016/j.biortech.2015.12.023
17. Dapelo D, Bridgeman J (2018) Euler–Lagrange computational fluid dynamics simulation of a full-scale unconfined anaerobic digester for wastewater sludge treatment. Adv Eng Softw 117:153–169. https://doi.org/10.1016/j.advengsoft.2017.08.009
18. Lebranchu A, Delaunay S, Marchal P, Blanchard F, Pacaud S, Fick M, Olmos E (2017) Impact of shear stress and impeller design on the production of biogas in anaerobic digesters. Biores Technol 245(June):1139–1147. https://doi.org/10.1016/j.biortech.2017.07.113
19. Meister M, Rezavand M, Ebner C, Pümpel T, Rauch W (2018) Mixing non-Newtonian flows in anaerobic digesters by impellers and pumped recirculation. Adv Eng Softw 115:194–203. https://doi.org/10.1016/j.advengsoft.2017.09.015
20. Dapelo D, Bridgeman J (2020) A CFD strategy to retrofit an anaerobic digester to improve mixing performance in wastewater treatment. Water Sci Technol 81(8):1646–1657. https://doi.org/10.2166/wst.2020.086
21. Tsoutsanis P, Antoniadis AF, Jenkins KW (2018) Improvement of the computational performance of a parallel unstructured WENO finite volume CFD code for Implicit Large Eddy Simulation. Comput Fluids 173:157–170. https://doi.org/10.1016/j.compfluid.2018.03.012

22. Hawkes J, Vaz G, Phillips AB, Cox SJ, Turnock SR (2018) On the strong scalability of maritime CFD. J Mar Sci Technol 23(1):81–93. https://doi.org/10.1007/s00773-017-0457-7
23. Dapelo D, Bridgeman J (2018) Assessment of mixing quality in full-scale, biogas-mixed anaerobic digestion using CFD. Biores Technol 265:480–489. https://doi.org/10.1016/j.biortech.2018.06.036
24. Wu B (2012) Large eddy simulation of mechanical mixing in anaerobic digesters. Biotechnol Bioeng 109(3):804–812. https://doi.org/10.1002/bit.24345
25. Abas A, Abdullah M, Ishak M, As N, Khor SF (2015) Lattice Boltzmann and Finite Volume simulations of multiphase flow in BGA encapsulation process. J Eng Appl Sci 10(17):7354–7360
26. Gaedtke M, Wachter S, Rädle M, Nirschl H, Krause MJ (2018) Application of a lattice Boltzmann method combined with a Smagorinsky turbulence model to spatially resolved heat flux inside a refrigerated vehicle. Comput Math Appl 76(10):2315–2329. https://doi.org/10.1016/j.camwa.2018.08.018
27. Haussmann M, Ries F, Jeppener-Haltenhoff JB, Li Y, Schmidt M, Welch C, Illmann L, Böhm B, Nirschl H, Krause MJ, Sadiki A (2020) Evaluation of a near-wall-modeled large Eddy Lattice Boltzmann method for the analysis of complex flows relevant to IC engines. Computation 8(2):43. https://doi.org/10.3390/computation8020043
28. Reinke F, Hafen N, Haussmann M, Novosel M, Krause MJ, Dittler A (2022) Applied geometry optimization of an innovative 3D-printed wet-scrubber nozzle with a lattice Boltzmann method. Chem Ing Tec 94(3):348–355. https://doi.org/10.1002/cite.202100151
29. Succi S (2015) Lattice Boltzmann 2038. Europhys Lett 109(5):50001. https://doi.org/10.1209/0295-5075/109/50001. (**ISBN: 0295-5075**)
30. Trunk R, Henn T, Dörfler W, Nirschl H, Krause MJ (2016) Inertial dilute particulate fluid flow simulations with an Euler–Euler lattice Boltzmann method. J Comput Sci 17:438–445. https://doi.org/10.1016/j.jocs.2016.03.013
31. Maier ML, Henn T, Thaeter G, Nirschl H, Krause MJ (2017) Multiscale simulation with a two-way coupled lattice Boltzmann method and discrete element method. Chem Eng Technol 40(9):1591–1598. https://doi.org/10.1002/ceat.201600547
32. Krause MJ, Klemens F, Henn T, Trunk R, Nirschl H (2017) Particle flow simulations with homogenised lattice Boltzmann methods. Particuology 34:1–13. https://doi.org/10.1016/j.partic.2016.11.001
33. Rettinger C, Rude U (2017) A comparative study of fluid-particle coupling methods for fully resolved lattice Boltzmann simulations. Comput Fluids 154:74–89. https://doi.org/10.1016/j.compfluid.2017.05.033
34. Dapelo D, Trunk R, Krause MJ, Bridgeman J (2019) Towards Lattice-Boltzmann modelling of unconfined gas mixing in anaerobic digestion. Comput Fluids 180:11–21. https://doi.org/10.1016/j.compfluid.2018.12.008
35. Dapelo D, Trunk R, Krause MJ, Cassidy N, Bridgeman J (2020) The application of Buckingham pi theorem to Lattice-Boltzmann modelling of sewage sludge digestion. Comput Fluids 209:104632. https://doi.org/10.1016/j.compfluid.2020.104632
36. Wu B (2010) CFD simulation of mixing in egg-shaped anaerobic digesters. Water Res 44(5):1507–1519. https://doi.org/10.1016/j.watres.2009.10.040
37. Sindall RC, Dapelo D, Leadbeater T, Bridgeman J (2017) Positron emission particle tracking (PEPT): a novel approach to flow visualisation in lab-scale anaerobic digesters. Flow Meas Instrum 54:250–264. https://doi.org/10.1016/j.flowmeasinst.2017.02.009
38. Zhang M, Zhang L, Jiang B, Yin Y, Li X (2008) Calculation of Metzner constant for double helical ribbon impeller by computational fluid dynamic method. Chin J Chem Eng 16(5):686–692. https://doi.org/10.1016/S1004-9541(08)60141-X
39. Wu B, Chen S (2008) CFD simulation of non-Newtonian fluid flow in anaerobic digesters. Biotechnol Bioeng 99(3):700–711. https://doi.org/10.1002/bit.21613
40. Landry H, Laguë C, Roberge M (2004) Physical and rheological properties of manure products. Appl Eng Agric 20(3):277–288. https://doi.org/10.13031/2013.16061. (**ISBN: 0883-8542**)
41. van Wachem BGM, Almstedt AE (2003) Methods for multiphase computational fluid dynamics. Chem Eng J 96(1–3):81–98. https://doi.org/10.1016/j.cej.2003.08.025. (**ISBN: 1385-8947**)
42. Verlet L (1967) Computer experiments on classical fluids. I. Thermodynamical properties of Lennard–Jones molecules. Phys Rev 159(1):98–103. https://doi.org/10.1103/PhysRev.159.98
43. Morsi SA, Alexander AJ (1972) An investigation of particle trajectories in two-phase flow systems. J Fluid Mech 55(02):193–208. https://doi.org/10.1017/S0022112072001806
44. Kruger T, Kasumaatmaja H, Kuzmim A, Shardt O, Silva G, Viggen EM (2017) The Lattice Boltzmann method. Springer, Berlin
45. Bhatnagar PL, Gross EP, Krook M (1954) A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. Phys Rev 94(3):511–525. https://doi.org/10.1103/PhysRev.94.511. (**ISBN: 0031-899X**)
46. Boyd J, Buick J, Green S (2006) A second-order accurate lattice Boltzmann non-Newtonian flow model. J Phys A Math Gen 39(46):14241–14247. https://doi.org/10.1088/0305-4470/39/46/001
47. Hou S, Sterling J, Chen S, Doolen GD (1996) A Lattice Boltzmann subgrid model for high Reynolds number flows. In: Lawniczak AT, Kapral R (eds) Pattern formation and lattice gas automata, vol 6. AMS-Fields Institute Communications, Toronto
48. Guo Z, Zheng C, Shi B (2002) Discrete lattice effects on the forcing term in the lattice Boltzmann method. Phys Rev E Stat Phys Plasmas Fluids Relat Interdiscip Topics. https://doi.org/10.1103/PhysRevE.65.046308
49. Falcucci G, Amati G, Fanelli P, Krastev VK, Polverino G, Porfiri M, Succi S (2021) Extreme flow simulations reveal skeletal adaptations of deep-sea sponges. Nature 595(7868):537–541. https://doi.org/10.1038/s41586-021-03658-1. (**Accessed 2023-05-09**)
50. Succi S, Amati G, Bernaschi M, Falcucci G, Lauricella M, Montessori A (2019) Towards exascale Lattice Boltzmann computing. Comput Fluids 181:107–115. https://doi.org/10.1016/j.compfluid.2019.01.005
51. Krause MJ, Kummerlaender A, Avis SJ, Kusumaatmaja H, Dapelo D, Klemens F, Gaedtke M, Hafen N, Mink A, Trunk R, Marquardt JE, Maier M-L, Haussmann M, Simonis S (2021) OpenLB-open source lattice Boltzmann code. Comput Math Appl 81:258–288. https://doi.org/10.1016/j.camwa.2020.04.033
52. Trunk R, Bretl C, Thäter G, Nirschl H, Dorn M, Krause MJ (2021) A study on shape-dependent settling of single particles with equal volume using surface resolved simulations. Computation 9(4):40. https://doi.org/10.3390/computation9040040
53. Siodlaczek M, Gaedtke M, Simonis S, Schweiker M, Homma N, Krause MJ (2021) Numerical evaluation of thermal comfort using a large eddy lattice Boltzmann method. Build Environ 192:107618. https://doi.org/10.1016/j.buildenv.2021.107618
54. Mink A, Schediwy K, Posten C, Nirschl H, Simonis S, Krause MJ (2022) Comprehensive computational model for coupled fluid flow, mass transfer, and light supply in tubular photobioreactors equipped with glass sponges. Energies 15(20):7671. https://doi.org/10.3390/en15207671
55. Gaedtke M, Abishek S, Mead-Hunter R, King A, Mullins BJ, Nirschl H, Krause MJ (2020) Total enthalpy-based lattice Boltzmann simulations of melting in paraffin/metal foam composite

phase change materials. Int J Heat Mass Transf 155:119870. https://doi.org/10.1016/j.ijheatmasstransfer.2020.119870

56. Kummerländer A, Avis S, Kusumaatmaja H, Bukreev F, Dapelo D, Großmann S, Hafen N, Holeksa C, Husfeldt A, Jeßberger J, Kronberg L, Marquardt JE, Mödl J, Nguyen J, Pertzel T, Simonis S, Springmann L, Suntoyo N, Teutscher D, Zhong M, Krause MJ (2022) OpenLB Release 1.5: open source lattice Boltzmann code. Zenodo. https://doi.org/10.5281/ZENODO.6469606. https://zenodo.org/record/6469606 Accessed 10 May 2023
57. Kummerländer A, Dorn M, Frank M, Krause MJ (2023) Implicit propagation of directly addressed grids in lattice Boltzmann methods. Concurr Comput Practice Exper. https://doi.org/10.1002/cpe.7509
58. Kummerlaender A, Bukreev F, Berg S, Dorn M, Krause MJ (2022) Advances in computational process engineering using Lattice boltzmann methods on high performance computers for solving fluid flow problems. In: High performance computing in science and engineering '22. Springer, Solan
59. Krause MJ, Avis S, Dapelo D, Hafen N, Haußmann M, Gaedtke M, Klemens F, Kummerländer A, Maier M-L, Mink A, Ross-Jones J, Simonis S, Trunk R (2019) OpenLB Release 1.3: open source lattice Boltzmann code. Zenodo. https://doi.org/10.5281/zenodo.3625967
60. Heuveline V, Krause MJ, Latt J (2009) Towards a hybrid parallelization of lattice Boltzmann methods. Comput Math Appl 58(5):1071–1080. https://doi.org/10.1016/j.camwa.2009.04.001
61. Mohrhard M, Thäter G, Bludau J, Horvat B, Krause MJ (2019) An auto-vecotorization friendly parallel lattice Boltzmann streaming scheme for direct addressing. Comput Fluids 181:1–7. https://doi.org/10.1016/j.compfluid.2019.01.001
62. Dapelo D, Simonis S, Krause MJ, Bridgeman J (2021) Lattice-Boltzmann coupled models for advection–diffusion flow on a wide range of Péclet numbers. J Comput Sci 51(April):101363. https://doi.org/10.1016/j.jocs.2021.101363