Journal of Computational Science 51 (2021) 101363

Contents lists available at ScienceDirect

Journal of Computational Science

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

Lattice-Boltzmann coupled models for advection–diffusion flow on a wide range of Péclet numbers

Davide Dapelo [a,*], Stephan Simonis [b,c], Mathias J. Krause [b,c], John Bridgeman [a]

[a] Faculty of Engineering and Informatics, University of Bradford, Bradford BD7 1DP, United Kingdom
[b] Lattice Boltzmann Research Group, Karlsruhe Institute of Technology, Karlsruhe, Germany
[c] Institute for Applied and Numerical Mathematics, Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany

ARTICLE INFO

Keywords:
Lattice-Boltzmann
OpenLB
Advection–diffusion
Finite-difference

ABSTRACT

Traditional Lattice-Boltzmann modelling of advection–diffusion flow is affected by numerical instability if the advective term becomes dominant over the diffusive (*i.e.*, high-Péclet flow). To overcome the problem, two 3D one-way coupled models are proposed. In a traditional model, a Lattice-Boltzmann Navier–Stokes solver is coupled to a Lattice-Boltzmann advection–diffusion model. In a novel model, the Lattice-Boltzmann Navier–Stokes solver is coupled to an explicit finite-difference algorithm for advection–diffusion. The finite-difference algorithm also includes a novel approach to mitigate the numerical diffusivity connected with the upwind differentiation scheme.

The models are validated using two non-trivial benchmarks, which includes discontinuous initial conditions and the case $\mathrm{Pe}_g \to \infty$ for the first time, where Pe_g is the grid Péclet number. The evaluation of Pe_g alongside Pe is discussed. Accuracy, stability and the order of convergence are assessed for a wide range of Péclet numbers. Recommendations are then given as to which model to select depending on the value Pe_g—in particular, it is shown that the coupled finite-difference/Lattice-Boltzmann provide stable solutions in the case $\mathrm{Pe} \to \infty$, $\mathrm{Pe}_g \to \infty$.

1. Introduction

Advection–diffusion occurs in a wide range of fluid-flow problems, including transport of chemicals, natural convection heat transfer when thermal expansion can be ignored, assessment of mixing through a passive scalar tracer, and many more.

Over the last decades, numerical modelling – in particular, computational fluid dynamics (CFD) – has been extensively used as a cheap but robust alternative to often lengthy and expensive experiments [1]. The lattice-Boltzmann is a particular type of explicit finite-difference method, with a tuneable diffusivity parameter [2]; this characteristics provides an advantage over traditional explicit finite-difference methods, as it allows the use of much larger timesteps than what would otherwise required to maintain stability. More significantly, it represents a valid alternative to implicit/segregated finite-volume CFD methods traditionally used to solve the Navier–Stokes equations because of a number of advantages in terms of numerical efficiency and parallelizability, *viz.*: (*i*) full explicitness, with no internal loop being required—therefore, every timestep is updated through a limited and well-defined number of floating-point operations; (*ii*) separation between non-linear and non-local part of the algorithm with the latter being usually limited to first neighbour-access, thereby allowing large parallel runs with no significant efficiency loss due to non-scalable inter-processor communication; and (*iii*) structure simplicity, allowing relatively simple extensions towards a wide variety of phenomena [2–4]. [5] showed that a Lattice-Boltzmann model for laboratory-scale gas mixing on a non-Newtonian fluid performed around 180 times faster than a finite-volume analogue [6], and ran on ten times more processors with no apparent loss of efficiency.

Whilst the Lattice-Boltzmann method has been applied to 1D and 2D advection–diffusion problems [7–9], the models demonstrate numerical instability in advection-dominant flow (high Péclet number) as the relaxation time approaches the critical value of 1/2. Proposed solutions valid for the 1D, 2D and 3D [10–12] mitigate the problem, but still exhibit instability for $\mathrm{Pe} \gtrapprox 10^4$. A more recent fractional-step method [13] requires doubling the number of timesteps and the number of lattice points across the coordinate directions; furthermore, it performs calculations requiring access to second-neighbour lattice sites, thereby compromising parallelizability. Therefore, there is clear a need for an advection–diffusion Lattice-Boltzmann method with the

[*] Correspondence to: B2.20, Chesham Building, Bradford BD7 1DP, United Kingdom.
E-mail address: ddapelo@bradford.ac.uk (D. Dapelo).

https://doi.org/10.1016/j.jocs.2021.101363
Received 23 December 2020; Received in revised form 28 March 2021; Accepted 30 March 2021
Available online 7 April 2021
1877-7503/© 2021 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Nomenclature

Φ	Phase space
Ξ	Navier–Stokes collision step, kg m^3
Ω	Computational domain
$\delta(\cdot)$	Dirac's delta
δx	Lattice cell size, m
δt	Lattice timestep, s
ϕ	Concentration of the advection–diffusion species, m^{-3}
ϕ^{new}	Concentration of the advection–diffusion species evaluated at the new timestep, m^{-3}
κ	Advection–diffusion lattice relaxation time, s
μ	Dynamics viscosity, Pa s
ν	Kinematic viscosity, m^2 s^{-1}
ρ	Density, Kg m^{-3}
σ	Shear stress, Pa
τ	Navier–Stokes lattice relaxation time, s
A^0	Finite-difference coefficient related to $\phi(\boldsymbol{x})$
$A_n^{\pm}$	Finite-difference coefficient related to $\phi\left(\boldsymbol{x} \pm \hat{\mathbf{e}}_n \delta x\right)$
$\mathscr{C}$	Collision operator, kg m^3
Co	Courant number
L	Nominal length scale, m
L_x	Benchmark's length, m
L_y	Benchmark's width & benchmark's nominal length scale, m
L_z	Benchmark's depth, m
Ma	Mach number
Nd	Finite-difference stability dimensionless number for diffusivity
NU	Finite-difference stability dimensionless number for nominal velocity
N_y	Voxels per unit length across the benchmark's cross section
Pe	Péclet number
Pe$_\text{g}$	Grid Péclet number
$\mathfrak{R}$	Reynolds number
Re$_\text{g}$	Grid Reynolds number
$\boldsymbol{U}$	Nominal advection velocity scale & benchmark's advection velocity, m s^{-1}
$\boldsymbol{c}_i$	ith discretized lattice velocity, m s^{-1}
c_s	Lattice speed velocity, m s^{-1}
d	Diffusivity, m^2 s^{-1}
d_{num}	Numerical diffusion term connected to the upwind differentiation scheme for advection, m^{-3}
$\hat{\mathbf{e}}_n$	Unit vector along the positive direction of the n coordinate
f	Navier–Stokes one-particle density function, kg m^{-3}
f_i	Navier–Stokes discretized one-particle density function relative to the ith lattice direction, kg m^{-3}
$f^{(\text{eq})}$	Navier–Stokes equilibrium one-particle density function, kg m^{-3}
$f_i^{(\text{eq})}$	Navier–Stokes discretized equilibrium one-particle density function relative to the ith lattice direction, kg m^{-3}
g_i	Advection–diffusion discretized one-particle density function relative to the ith lattice direction, kg m^{-3}
$g_i^{(\text{eq})}$	Advection–diffusion discretized equilibrium one-particle density function relative to the ith lattice direction, kg m^{-3}
p	Pressure, Pa
t	Time, s
$\boldsymbol{u}$	Advection velocity, m s^{-1}
$\boldsymbol{x}$	Spatial coordinate, m
x_0	x coordinate of the benchmark's initial tracer distribution, m
$\cdot^*$	Dimensionless version of the argument represented by the $\cdot$
$[\cdot]^{\text{sch}}$	Differentiation scheme "sch" being applied to the $\cdot$ inside the brackets
CPUs	CPU-second (*viz.*, number of seconds a given numerical simulation takes to be run, times number of CPU cores employed)
MLUPps	Millions of lattice updates per processor per second

characteristics of high Péclet number stability, and requiring access to first-neighbouring cells only.

Within this article, two Lattice-Boltzmann models are proposed to reproduce a diffusive scalar field concentration advected by a hydrodynamic velocity field. In both the models, hydrodynamics and advection–diffusion are simulated in a coupled way. The two models are composed of a Lattice-Boltzmann submodel to solve hydrodynamics, and a second submodel for advection–diffusion. The coupling occurs one-way, *viz.*: the solution to the hydrodynamics subproblem influences the advection–diffusion solution, but the converse does not occur. In the "traditional" model, the advection–diffusion submodel is defined as a Lattice-Boltzmann model, whilst in the "novel" model, an explicit finite-difference submodel is used. The rationale for introducing an explicit finite-difference submodel for advection–diffusion is based on the fact that finite-difference is the most obvious choice of model when the computational domain is composed of a cubic lattice (as is the case of the numerical work presented here), and, for the very specific case of an explicit method with access to first neighbours only, it maintains the same level of parallelizability and efficiency as the standard Lattice-Boltzmann method. Scaling tests are performed in order to verify this claim. In both models, at each timestep, the Lattice-Boltzmann submodel solves the Navier–Stokes equations to update the hydrodynamics; the coupling is performed by passing the obtained velocity field to the advection–diffusion submodel; and finally, the latter solves the advection–diffusion equation and updates the scalar field concentration.

In the literature cited above, the assessments are performed in terms of the Péclet number, or other numbers tracing the balance between advective and a given form of diffusive (chemical, or thermal) transport. This approach provides a good assessment of the physics underlying the phenomena being analysed, but does not necessarily provide a reliable assessment of numerical stability. [9] defines the grid Reynolds number Re$_\text{g}$ from lattice quantities, and identifies that a simulation is stable if Re$_\text{g}$ $\gtrsim 10$, heuristically meaning that "The lattice should always be sufficiently fine to resolve local vortices". [14,15] introduce the grid Péclet number Pe$_\text{g}$ as the analogue to Re$_\text{g}$ in advection–diffusion models. This approach implies a separation between physical interpretation of a process (*i.e.*, the balance between advective and diffusive transport), which is expressed through $\mathfrak{R}$, Pe, …; and stability assessment of a numerical simulation (*i.e.*, whether or not the lattice is fine enough to resolve the local vortices), which is expressed through Re$_\text{g}$, Pe$_\text{g}$, …. In parallel with the assessment of Pe$_\text{g}$, the evaluation of the numerical stability is also performed in [9,14,15] through methodologies involving the mutual balance between Re$_\text{g}$ or Pe$_\text{g}$, the lattice relaxation time

τ (or κ in the advection–diffusion) and the characteristic velocity U or the courant number $\mathrm{Co} \equiv U\delta t/\delta x$ (where δx is the lattice spacing and δt the timestep).

The above-mentioned work does not explicitly consider the case of zero diffusivity, *i.e.*$\mathrm{Pe_g} \to \infty$; in fact, it can be implied from [9] and [14] that the simulations are intrinsically unstable in this specific case. However, an explicit assessment of the case $\mathrm{Pe_g} \to \infty$ is justified by its importance in applications, *e.g.* in assessing mixing quality through scalar tracer concentration, or wherever diffusive processes are negligible altogether. For this reason, the work described within this article is focussed on the assessment of model stability in terms of $\mathrm{Pe_g}$ alone, and the case Pe, $\mathrm{Pe_g} \to \infty$ is explicitly taken into consideration for the first time within the research concerning the Lattice-Boltzmann method; the balance between $\mathrm{Pe_g}$, Co and relaxation time is considered only for the purpose of verifying the coherence of the simulations. In order to provide the most possible robust assessment, two benchmarks are implemented in a non-trivial way as a Gaussian concentration packet or a sinusoidal wave spreading and being advected by a constant velocity field, with a tracer concentration presenting a sharp discontinuity as initial condition; within both periodical and non-periodical computational domains. Previous work [16] shows that Lattice-Boltzmann advection–diffusion observes second-order convergence on simulations starting from continuous initial conditions, and first-order if the initial condition presents a discontinuity. This reduction of the order of convergence is explained by the fact that, in the discontinuous case, the eigenvalues of the Lattice-Boltzmann evolution matrix are close to −1, thereby causing the onset of oscillations. However, [16] is limited in analysis to 1D simulations: within this work, the analysis is extended to the 3D case, thereby providing a complete assessment of the convergence for the model and filling the knowledge gap in the literature.

All the simulations are performed using OpenLB (www.openlb.net) version 1.3 [4,17].

This article is structured as follows. In Section 2, the two models are outlined. In Section 3, the Navier–Stokes Lattice-Boltzmann model and the advection–diffusion Lattice-Boltzmann are described. Then, the explicit finite-difference model is described in Section 4. Two benchmarks to assess performance and stability are described in Section 5. The results are reported in Section 6, and recommendations on which model to use depending on the value of $\mathrm{Pe_g}$ are given in Section 7. Finally, conclusions are drawn in Section 8.

2. General outline of the model

Both the proposed coupled and the standard models follow the general algorithm described below.

Algorithm 1: General algorithm

```
Initialize hydrodynamics (density field $\rho$ and velocity field $\boldsymbol{u}$) and
advection–diffusion characteristics (scalar concentration field $\phi$);
for timestep: $t = 0$ to $t_{\max}$ by $\delta t$ do
    Hydrodynamics update: a standard Lattice-Boltzmann
    submodel solves the weakly-compressible Navier–Stokes
    equations:

    $\partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0\ ;$    (1)

    $\partial_t (\rho \boldsymbol{u}) + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla p + \mu \nabla^2 \boldsymbol{u}$    (2)

    (with $p$ being the pressure and $\nu$ the dynamic viscosity) and
    updates the velocity field $\boldsymbol{u}$;
    Pass the updated velocity field $\boldsymbol{u}$ to the submodel for
    advection–diffusion;
    Advection–diffusion update: an explicit finite-difference
    submodel (or an advection–diffusion Lattice-Boltzmann
    submodel for the comparison model) solves the
    advection–diffusion equation:

    $\partial_t \phi + \nabla \cdot (\phi \boldsymbol{u}) = d \nabla^2 \phi$    (3)

    (with $d$ being the diffusivity) and updates the concentration
    field $\phi$;
```

3. Lattice-Boltzmann modelling

The Lattice-Boltzmann model is a mesoscopic model insofar as it aims at finding a trajectory $f(\boldsymbol{x}, \boldsymbol{c}, t)$ in the phase space $\boldsymbol{\Phi} \equiv \left\{(\boldsymbol{x}, \boldsymbol{c}) \in \mathbb{R}^3 \times \mathbb{R}^3\right\}$. The "one-particle density function" $f(\boldsymbol{x}, \boldsymbol{c}, t)$ describes the probability of finding one particle of fluid within the elemental cube $(\boldsymbol{x}, \boldsymbol{x} + d\boldsymbol{x})$ with a velocity comprised within the cubic interval $(\boldsymbol{c}, \boldsymbol{c} + d\boldsymbol{c})$ at the time t. The observable fields (*viz.*, density, velocity and shear stress) correspond to the particle density function's zeroth, first and second moments respectively [9]:

$$\rho = \int f \, d\boldsymbol{c}\ ; \tag{4}$$

$$\rho \boldsymbol{u} = \int f \, \boldsymbol{c} \, d\boldsymbol{c}\ ; \tag{5}$$

$$\rho \boldsymbol{u} \otimes \boldsymbol{u} - \sigma = \int f \, \boldsymbol{c} \otimes \boldsymbol{c} \, d\boldsymbol{c}\ . \tag{6}$$

The model solves the Boltzmann equation:

$$\left(\partial_t + \boldsymbol{c} \cdot \nabla\right) f = \mathscr{C}[f]\ . \tag{7}$$

Eq. (7) describes continuity for f in the phase space, plus a source–sink term arising from inter-particle collisions. In the diluted gas hypothesis, only binary particle collisions contribute to the collision operator $\mathscr{C}$. Under the isotropy assumption, it is possible to model the potentially extremely complicated operator $\mathscr{C}$ [18] as an average, relaxing towards the equilibrium particle density function $f^{(\mathrm{eq})}$:

$$\mathscr{C}[f] = -\frac{1}{\tau}\left(f - f^{(\mathrm{eq})}\right)\ , \tag{8}$$

where τ is the relaxation time, and $f^{(\mathrm{eq})}$ is the Maxwell equilibrium distribution [9]:

$$f^{(\mathrm{eq})}(\boldsymbol{x}, t) := \rho(\boldsymbol{x}, t) \left(\frac{1}{2\pi c_s^2}\right)^{3/2} \exp\left\{-\frac{[\boldsymbol{u}(\boldsymbol{x}, t)]^2}{2c_s^2}\right\}\ . \tag{9}$$

c_s is the speed of sound, and density and velocity are evaluated through Eqs. (4) and (5).

Space is discretized as a cubic lattice of lattice size δx, time is subdivided into timesteps of interval δt, and the three-dimensional velocity space is discretized as a finite set of vectors $\left\{\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{q-1}\right\}$ pointing to the zeroth, first, second and third neighbours of a generic lattice site, with moduli respectively $\sqrt{0}$, $\sqrt{1}$, $\sqrt{2}$ and $\sqrt{3}$ times $\delta x/\delta t$. A given discretization is conventionally identified by a tag DdQq, with d the dimensionality of the problem and q the dimension of the discrete velocity space. $f(\boldsymbol{x}, \boldsymbol{c}, t)$ is redefined over the discretized lattice and timesteps, and rewritten as a set $f_i(\boldsymbol{x}, t)$, each f_i indicating the probability of finding a particle with the corresponding discretized velocity $\boldsymbol{c}_i$. The integrals in Eqs. (4), (5) and (6) are redefined as summations over the velocity set:

$$\rho = \sum_i f_i\ ; \tag{10}$$

$$\rho \boldsymbol{u} = \sum_i f_i \, \boldsymbol{c}_i\ ; \tag{11}$$

$$\rho \boldsymbol{u} \otimes \boldsymbol{u} - \sigma = \sum_i f_i \, \boldsymbol{c}_i \otimes \boldsymbol{c}_i\ . \tag{12}$$

The error arising from the discretization of the velocity set is cancelled by writing the Maxwell equilibrium function (Eq. (9)) in terms of orthonormal Hermite polynomials up to the second order:

$$f_i^{(\mathrm{eq})} = w_i\,\rho\left[1+\frac{\boldsymbol{u}\cdot\boldsymbol{c}_i}{c_\mathrm{s}^2}+\frac{\left(\boldsymbol{u}\cdot\boldsymbol{c}_i\right)^2-c_\mathrm{s}^2u^2}{2c_\mathrm{s}^4}\right]\,, \tag{13}$$

where the weights w_i and the speed of sound are defined in a standard way for the specific DdQp lattice, with the latter usually being defined as:

$$c_\mathrm{s} := \frac{1}{\sqrt{3}}\frac{\delta x}{\delta t}\,. \tag{14}$$

The Boltzmann Eq. (7), considering the BGK assumption (Eq. (8)), becomes the Lattice-Boltzmann Equation:

$$f_i\left(\boldsymbol{x}+\boldsymbol{c}_i\,\delta t,\,t+\delta t\right) = f_i\left(\boldsymbol{x},\,t\right)-\frac{1}{\tau}\left[f_i\left(\boldsymbol{x},\,t\right)-f_i^{(\mathrm{eq})}\left(\boldsymbol{x},\,t\right)\right]\,. \tag{15}$$

The updating process for each lattice site at a given timestep comprises two steps. First: a local, non-linear *collision*:

$$\Xi_i\left(\boldsymbol{x},\,t\right) = f_i\left(\boldsymbol{x},\,t\right)-\frac{1}{\tau}\left[f_i\left(\boldsymbol{x},\,t\right)-f_i^{(\mathrm{eq})}\left(\boldsymbol{x},\,t\right)\right]\,. \tag{16}$$

Then: a linear, non-local *streaming*:

$$f_i\left(\boldsymbol{x}+\boldsymbol{c}_i,\,t+1\right) = \Xi_i\left(\boldsymbol{x},\,t\right)\,. \tag{17}$$

In this way, the adiabatic dynamics with a compressibility error of Ma^2 with Ma being the Mach number, are recovered. In the limit $\mathrm{Ma} \ll 1$, the Lattice-Boltzmann Eq. (15) reproduces the Navier–Stokes (Eqs. (1)) and (2)) in the incompressible limit [9], with pressure and kinematic viscosity defined as:

$$p := \rho c_\mathrm{s}^2\,, \qquad \nu := c_\mathrm{s}^2\left(\tau-\frac{\delta t}{2}\right)\,. \tag{18}$$

3.1. Lattice-Boltzmann model for the advection–diffusion equations

A comparison between the advection–diffusion (Eq. (3)) and the Navier–Stokes equation for the momentum (Eq. (2)) shows that the latter can be regarded as an advection–diffusion equation for the momentum density vector $\rho\boldsymbol{u}$ in place of the scalar concentration ϕ. This analogy allows the export of the same model developed in Section 3 for the Navier–Stokes case, to the advection–diffusion [9]. Alternatively, it is possible to derive the Lattice-Boltzmann advection–diffusion model directly from the macroscopic equation (3) following the approach of [19]. Either way, a discretized one-particle density function is conventionally defined as $g_i(\boldsymbol{x},\,t)$, and the concentration ϕ is obtained from its zeroth moment:

$$\phi = \sum_i g_i\,. \tag{19}$$

The only difference from the Navier–Stokes case is that higher-order moments of g_i do not possess physical meaning because only one single scalar conservation law (*viz.*, mass) is approximated in the advection diffusion equation (Eq. (3)), and therefore are not evaluated. In fact, the physical advection velocity $\boldsymbol{u}$ appearing in Eq. (3) is imposed externally (*viz.*, copied from the hydrodynamics model). The rest of the model is left unchanged: *i.e.*the definition of speed of sound as a numerical constant (Eq. (14)) is maintained unaltered, the Lattice-Boltzmann equation is defined as in Eq. (15):

$$g_i\left(\boldsymbol{x}+\boldsymbol{c}_i\,\delta t,\,t+\delta t\right) = g_i\left(\boldsymbol{x},\,t\right)-\frac{1}{\kappa}\left[g_i\left(\boldsymbol{x},\,t\right)-g_i^{(\mathrm{eq})}\left(\boldsymbol{x},\,t\right)\right] \tag{20}$$

(with an advection–diffusion relaxation time κ), the equilibrium function follows Eq. (13) with ϕ in place of ρ:

$$g_i^{(\mathrm{eq})} = w_i\,\phi\left[1+\frac{\boldsymbol{u}\cdot\boldsymbol{c}_i}{c_\mathrm{s}^2}+\frac{\left(\boldsymbol{u}\cdot\boldsymbol{c}_i\right)^2-c_\mathrm{s}^2u^2}{2c_\mathrm{s}^4}\right]\,, \tag{21}$$

and the diffusivity d takes the place of the kinematic viscosity ν in Eq. (18):

$$d := c_\mathrm{s}^2\left(\kappa-\frac{\delta t}{2}\right)\,. \tag{22}$$

The same lattice of the Lattice-Boltzmann Navier–Stokes model is used in order to avoid interpolation, and with the same timestep δt to simplify the time iteration. Consequently, the values of the speed of found for the advection–diffusion and the Navier–Stokes models are the same.

3.2. Stability constraint of the lattice-Boltzmann model for advection–diffusion

The Péclet number is defined as:

$$\mathrm{Pe} := \frac{LU}{d}\,, \tag{23}$$

where L and U are respectively the length and velocity scales of a given problem. As the Reynolds number is the ratio of inertial to viscous forces in the Navier–Stokes case, Pe can be interpreted as the balance between advective and diffusive transport. The advection–diffusion relaxation time κ approaches the stability-critical value of $1/2$ when Pe grows, exactly as τ in the Navier–Stokes case if $\mathfrak{R}$ grows [13].

Following the analogy between Navier–Stokes and advection-diffusion, a first criterion to assess advection–diffusion Lattice-Boltzmann can be outlined through defining the grid Péclet number following [9]:

$$\mathrm{Pe}_\mathrm{g} := \frac{L^*\delta x^*}{d^*}\,. \tag{24}$$

The stability criterion reads:

$$\mathrm{Pe}_\mathrm{g} \lesssim 10\,. \tag{25}$$

Within this work, the dimensionless version ζ^* of the quantity ζ is computed from its units of measure $\{\zeta\}$ as follows:

$$\zeta^* := \zeta\,\delta x^{-a}\,\delta t^{-b} \qquad \text{if}\quad \{\zeta\} = \mathrm{m}^a\,\mathrm{s}^b\,. \tag{26}$$

Considering that, within this work, we have $L = N_y\delta x$ with N_y being the number of cells comprised within L, it follows that Pe_g can be equivalently defined as:

$$\mathrm{Pe}_\mathrm{g} = \frac{U\,\delta x}{d} = \mathrm{Pe}\frac{\delta x}{L} = \frac{\mathrm{Pe}}{N_y}\,. \tag{27}$$

A second criterion for advection–diffusion Lattice-Boltzmann stability comes from the same analogy. [20] studied the neutral stability Co–τ^* curves, and [9] provided a simplified mathematical expression to defined the stability domain of the Lattice-Boltzmann method. This expression can be extended to the advection–diffusion as follows:

$$\begin{cases} \mathrm{Co} \leq 0.4 & \text{if}\quad \kappa^* \geq 0.55\,; \\ \kappa^* > \frac{1}{2}+\frac{1}{8}\mathrm{Co} & \text{if}\quad 0.5 < \kappa^* < 0.55\,. \end{cases} \tag{28}$$

As the focus of this work is to compare the stability of the Lattice-Boltzmann and the explicit finite-difference advection–diffusion models in terms of Pe_g and in the specific case $\mathrm{Pe}_\mathrm{g} \to \infty$, the criterion of Eq. (25) is mainly employed. The robustness of the subsequent conclusions is then assessed through the criterion of Eq. (28).

4. Finite-difference model for the advection–diffusion equations

An explicit finite-difference discretization [21] of the macroscopic advection–diffusion equation (3 is performed on the same lattice of the Lattice-Boltzmann Navier–Stokes model in order to avoid interpolation, and with the same timestep δt to simplify the time iteration:

$$\frac{\phi^\mathrm{new}-\phi}{\delta t}+\left[\nabla\cdot\left(\phi\boldsymbol{u}\right)\right]^\mathrm{adv} = \left[d\nabla^2\phi\right]^\mathrm{diff}\,. \tag{29}$$

The “new” superscript indicates that the value of the concentration field is referred to the timestep being currently resolved; conversely, its absence indicates that the value of the concentration field refers to the previous timestep. This formulation allows an explicit evaluation of ϕ^new without timestep-internal loops, as in the Lattice-Boltzmann case. The differentiation schemes $[\,\cdot\,]^\mathrm{sch}$ are chosen as: the standard second-order central scheme for the diffusion term; and either standard first-order upwind, or standard second-order central schemes for the advection term.

In three dimensions, the central diffusion scheme reads:

$$\left[d\nabla^2\phi(\boldsymbol{x})\right]^{\mathrm{diff}} = d\,\frac{\sum_{n\in\{x,y,z\}}\left[\phi\left(\boldsymbol{x}+\hat{\mathbf{e}}_n\delta x\right)+\phi\left(\boldsymbol{x}-\hat{\mathbf{e}}_n\delta x\right)\right]-6\phi(\boldsymbol{x})}{\delta x^2}\,. \tag{30}$$

The central advection scheme reads:

$$[\nabla\cdot(\phi\boldsymbol{u})]^{\mathrm{adv\_central}} = \frac{\sum_{n\in\{x,y,z\}}u_n\left[\phi\left(\boldsymbol{x}+\hat{\mathbf{e}}_n\delta x\right)-\phi\left(\boldsymbol{x}-\hat{\mathbf{e}}_n\delta x\right)\right]}{2\delta x}\,. \tag{31}$$

The upwind:

$$[\nabla\cdot(\phi\boldsymbol{u})]^{\mathrm{adv\_upwind}} = \frac{\sum_{n\in\{x,y,z\}}\boldsymbol{\Phi}\left[\phi(\boldsymbol{x}),\boldsymbol{u},\hat{\mathbf{e}}_n\right]}{\delta x}\,, \tag{32}$$

with:

$$\boldsymbol{\Phi}\left[\phi(\boldsymbol{x}),\boldsymbol{u},\hat{\mathbf{e}}_n\right] = \begin{cases} u_n\left[\phi(\boldsymbol{x})-\phi\left(\boldsymbol{x}-\hat{\mathbf{e}}_n\delta x\right)\right]\,, & u_n \geq 0\,; \\ u_n\left[\phi\left(\boldsymbol{x}+\hat{\mathbf{e}}_n\delta x\right)-\phi(\boldsymbol{x})\right]\,, & u_n < 0\,. \end{cases} \tag{33}$$

The orders of convergence of the schemes above (Eqs. (30)–(32)) can be verified by substituting Taylor expansions of $\phi\left(\boldsymbol{x}\pm\hat{\mathbf{e}}_n\delta x\right)$ around $\phi(\boldsymbol{x})$ in the respective equations.

4.1. Stability constraints of the finite-difference model for advection-diffusion

Substituting the appropriate differentiation schemes (Eqs. (30)–(32)), it is possible to rewrite Eq. (29) in the following general form:

$$\phi^{\mathrm{new}} = A^0\phi(\boldsymbol{x}) + \sum_{n\in\{x,y,z\}}A_n^+\phi\left(\boldsymbol{x}+\hat{\mathbf{e}}_n\delta x\right) + \sum_{n\in\{x,y,z\}}A_n^-\phi\left(\boldsymbol{x}-\hat{\mathbf{e}}_n\delta x\right)\,. \tag{34}$$

As a necessary condition for the algorithm to be stable, all the coefficients A in the equation above should be non-negative. This requirement, in turn, poses constraints on the values of velocity, diffusivity, lattice spacing and timestep, as follows.

A first constraint comes from the requirement that A^0 should be non-negative. This is related to the explicit nature of the model, and consists of a superior limit to the timestep value. Within this work, it is formulated as follows:

$$\mathrm{Nd} := \frac{\delta x^2}{6d\ \delta t} > 1\,. \tag{35}$$

Whilst this constraint usually poses serious limits to explicit methods, it does not prevent a successful application of the coupled model because in Lattice-Boltzmann modelling, the timestep value is usually kept low by the assumption $U\ \delta t/\delta x \sim \mathrm{Ma} \ll 1$. Conversely, once δx and δt are chosen following the balance between accuracy, stability and numerical expense usually performed in Lattice-Boltzmann modelling [9], Eq. (35) poses a superior limit to the diffusivity.

A second constraint comes from the requirement that A_n^- should be non-negative in the case of the central advection scheme:

$$\mathrm{NU} := \frac{2d}{U\ \delta x} > 1\,. \tag{36}$$

This constraint poses an inferior limit to the ratio of diffusivity over reference velocity. A comparison between Eqs. (27) and (36) shows that:

$$\mathrm{NU} = \frac{2}{\mathrm{Pe_g}}\,. \tag{37}$$

Therefore, the constraint of Eq. (36) is equivalent to a constraint on $\mathrm{Pe_g}$:

$$\mathrm{Pe_g} \lesssim 5\,. \tag{38}$$

As in Section 3.2, the interest is directed towards the stability criterion of Eq. (38) in order to compare the stability of the Lattice-Boltzmann and the explicit finite-difference advection–diffusion models in terms of $\mathrm{Pe_g}$ and in the specific case $\mathrm{Pe_g} \to \infty$. The criterion of Eq. (35) is used as a background verification of the applicability of the explicit finite-difference method.

4.2. Negative-diffusivity correction

Although Eq. (36) poses a strict stability constraint only if the central advection scheme is adopted, bespoke numerical results in the upwind advection case show the presence of numerical diffusivity along the direction of the flow, proportional to U. This is due to the fact that the first truncated term in Eq. (32), corresponding to second-ordered Taylor expansions of $\phi\left(\boldsymbol{x}\pm\hat{\mathbf{e}}_n\delta x\right)$ around $\phi(\boldsymbol{x})$, takes the form of:

$$d_{\mathrm{num}} = \frac{\left|u_n\right|\delta x}{2}\left(\frac{\partial^2\phi}{\partial x^2}+\frac{\partial^2\phi}{\partial y^2}\right)\,. \tag{39}$$

A comparison of this term to the diffusion differentiation scheme (Eq. (30)) shows that this term can be interpreted as an artificial anisotropic diffusivity contribution.

In order to mitigate this effect, a novel approach, consisting of multiplying the artificial diffusivity (Eq. (39)) by a tuning factor α and subtracting it from the diffusivity scheme, is proposed. The second-order central differentiation scheme for diffusivity is thus redefined as follows when employed in combination with the upwind scheme for advection:

$$\left[d\nabla^2\phi(\boldsymbol{x})\right]_\alpha^{\mathrm{diff\_corr}} = \frac{\sum_{n\in\{x,y,z\}}\left(d-\alpha\frac{\left|u_d\right|\delta x}{2}\right)\left[\phi\left(\boldsymbol{x}+\hat{\mathbf{e}}_n\delta x\right)+\phi\left(\boldsymbol{x}-\hat{\mathbf{e}}_n\delta x\right)\right]-6\phi(\boldsymbol{x})}{\delta x^2}\,. \tag{40}$$

5. Numerical work

A periodical and a non-periodical benchmark are defined. Over the periodical benchmark, simulations are performed through only the Lattice-Boltzmann advection–diffusion model, with both smooth and sharp initial conditions for the concentration field ϕ. Over the non-periodical benchmark, simulations are performed through both the Lattice-Boltzmann and the finite-difference advection–diffusion models, but only with a sharp initial condition. As detailed below, the smooth-start simulations reproduce a sinusoid, and the sharp-start one a series of Gaussian packets.

5.1. Periodical benchmark

A simplified version of the general algorithm (Section 2) is applied—specifically, the hydrodynamics is not solved, and a constant velocity field $\boldsymbol{U} = (2.5,\ 2.5,\ 2.5)$ m/s is passed to the advection–diffusion model. A computational domain Ω is defined as a periodical cube size $L_y = 2$ m. The simulations are performed over a time of 1.52 s.

Five different values of N_y, corresponding to 25, 50, 100, 200 and 400, are chosen. The timestep is determined under diffusive scaling through the prescription $\delta t = \delta x^2 \cdot 1\ \mathrm{s/m^2}$. Four different values of Pe, corresponding to 100, 1 000, 10 000 and 100 000, are chosen. The corresponding values of the relaxation time in dimensionless units κ^* are 0.65, 0.515, 0.5015 and 0.50015, and the corresponding values of the Courant number Co are 0.2, 0.1, 0.05, 0.025 and 0.0125 respectively. The corresponding values of d and $\mathrm{Pe_g}$ are determined through Eqs. (23) and (27) respectively.

A time-averaged discrete relative L^2-error norm is calculated via:

$$\overline{\mathrm{err}} = \frac{1}{M}\sum_{i=1}^{M}\mathrm{err}_{L^2}\left(t_i\right)\,, \tag{41}$$

where:

$$\mathrm{err}_{L^2}\left(t_i\right) = \sqrt{\frac{\sum_{\boldsymbol{x}\in\Omega}\left[\phi\left(\boldsymbol{x},t_i\right)-\phi^{\mathrm{an}}\left(\boldsymbol{x},t_i\right)\right]^2}{\sum_{\boldsymbol{x}\in\Omega}\left[\phi^{\mathrm{an}}\left(\boldsymbol{x},t_i\right)\right]^2}}\,. \tag{42}$$

ϕ is the numerical outcome of the simulations, while ϕ^{an} is the bespoke analytical solution.

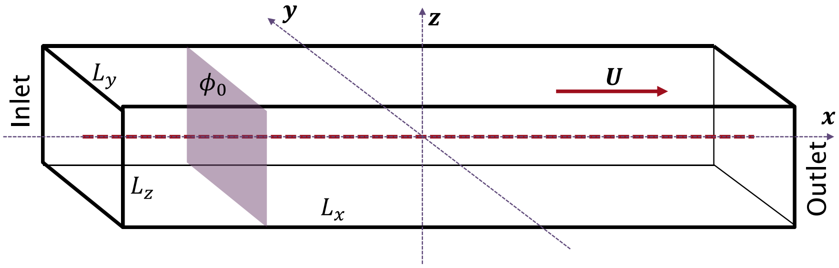
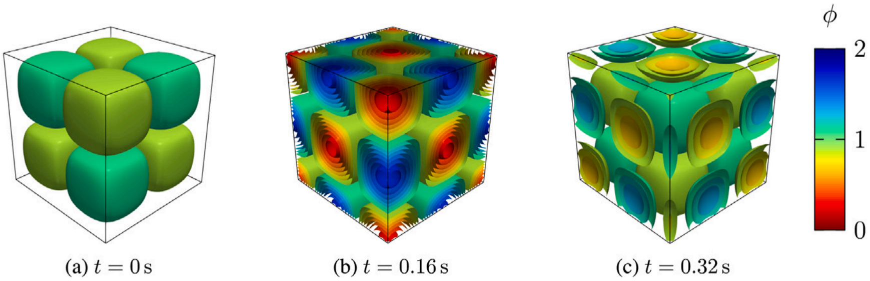


Fig. 1. Non-periodical benchmark for sharp initial condition.



(a) $t = 0\,\mathrm{s}$ (b) $t = 0.16\,\mathrm{s}$ (c) $t = 0.32\,\mathrm{s}$

Fig. 2. Periodical benchmark for smooth initial condition. Iso-contour solution for $N_y = 200$ and $\mathrm{Pe} = 100$.

5.2. Non-periodical benchmark

The full general algorithm is applied as described in Section 2. The computational domain (Fig. 1) consists of a square-section pipe with $L_y = L_z = 1$ m and $L_x \gg L_y$ being set depending on the specific run. The flow velocity is a constant $\boldsymbol{U}$ with magnitude of 0, 0.1 or 2 m/s, directed towards the x direction and is once again set depending on the specific run. The boundary conditions for the Lattice-Boltzmann Navier–Stokes solver are: inlet and outlet are defined as in the figure; free-slip on the walls. For the advection–diffusion solvers, the boundary conditions on the wall are defined as: bounce-back for the Lattice-Boltzmann, no-penetration for the finite-difference. The potential error arising from setting bounce-back and no-penetration conditions to inlet and outlet is avoided by choosing $L_x \gg L_y,\ L_z$ and by stopping the simulations before the Gaussian packet's 3σ tail reached the outlet.

The Navier–Stokes lattice relaxation time is set to $\tau = 0.8$. The timestep is determined under diffusive scaling through the prescription $\delta t = \delta x^2 \cdot 1\ \mathrm{s/m}^2$. Unless otherwise specified, it is set $N_y = 11$. It follows that Co amounts to 0, 0.009 and 0.182 for U corresponding respectively to 0, 0.1 and 2.0 m/s.

Simulation results and errors are evaluated at each timestep separately, in terms of the value of ϕ along the dashed line in Fig. 1 (indicated as Λ below). Only the last timestep is considered for further analysis. The error is computed as the L_2 distance between numerical and analytical values of ϕ over the L_2 norm of the analytical solution, evaluated along the same dashed line:

$$\mathrm{err}_{L^2}\left(t_i\right) = \sqrt{\frac{\sum_{\boldsymbol{x}\in\Lambda}\left[\phi\left(\boldsymbol{x}, t_i\right) - \phi^{\mathrm{an}}\left(\boldsymbol{x}, t_i\right)\right]^2}{\sum_{\boldsymbol{x}\in\Lambda}\left[\phi^{\mathrm{an}}\left(\boldsymbol{x}, t_i\right)\right]^2}}\ . \tag{43}$$

5.3. Smooth initial condition

The initial condition is set as:

$$\phi_0(\boldsymbol{x}) \equiv \phi(\boldsymbol{x}, 0) = \sin(\pi x)\sin(\pi y)\sin(\pi z) + 1\ . \tag{44}$$

It can be demonstrated [22] that the benchmark possesses the following analytical solution:

$$\phi^{\mathrm{an}}(\boldsymbol{x}, t) = \sin\left[\pi\left(x - u_x t\right)\right]\sin\left[\pi\left(y - u_y t\right)\right]\sin\left[\pi\left(z - u_z t\right)\right]e^{-3\pi^2 td} + 1\ . \tag{45}$$

An example of solution is provided in Fig. 2.

5.3.1. Non-periodical

A spreading Gaussian density packet advecting along $\boldsymbol{U}$ is modelled. The initial condition for ϕ is defined as:

$$\phi_0(\boldsymbol{x}) \equiv \phi(\boldsymbol{x}, 0) = \begin{cases} \frac{1}{L_y L_z \delta x}\ , & x \in \left(x_0 - \frac{\delta x}{2}\ , x_0 + \frac{\delta x}{2}\right)\ ; \\ 0\ , & \text{elsewhere.} \end{cases} \tag{46}$$

The initial position x_0 is either 0 or $L_x/4$ depending on the particular run. It can be demonstrated that the benchmark possesses the following analytical solution:

$$\phi^{\mathrm{an}}(\boldsymbol{x}, t) = \frac{1}{L_y L_z\sqrt{4\pi td}}\exp\frac{-\left(x - Ut - x_0\right)^2}{4td}\ , \tag{47}$$

and the initial condition for the analytical solution is the Dirac's delta:

$$\phi^{\mathrm{an}}(\boldsymbol{x}, 0) = \frac{1}{L_y L_z}\delta\left(x - x_0\right)\ . \tag{48}$$

A proof of the mutual coherence of Eqs. (46), (47) and (48) is given in the Supplementary Material.

5.3.2. Periodical

A superposition of spreading Gaussian density packets advecting along $\boldsymbol{U}$ is modelled. The initial condition for ϕ is defined as:

$$\phi_0(\boldsymbol{x}) \equiv \phi(\boldsymbol{x}, 0) = \begin{cases} \frac{1}{\sqrt{4\pi d\,\delta t}} + 1, & x \in \left(x_0 - \frac{\delta x}{2}\ , x_0 + \frac{\delta x}{2}\right)\ ; \\ 1\ , & \text{elsewhere.} \end{cases} \tag{49}$$

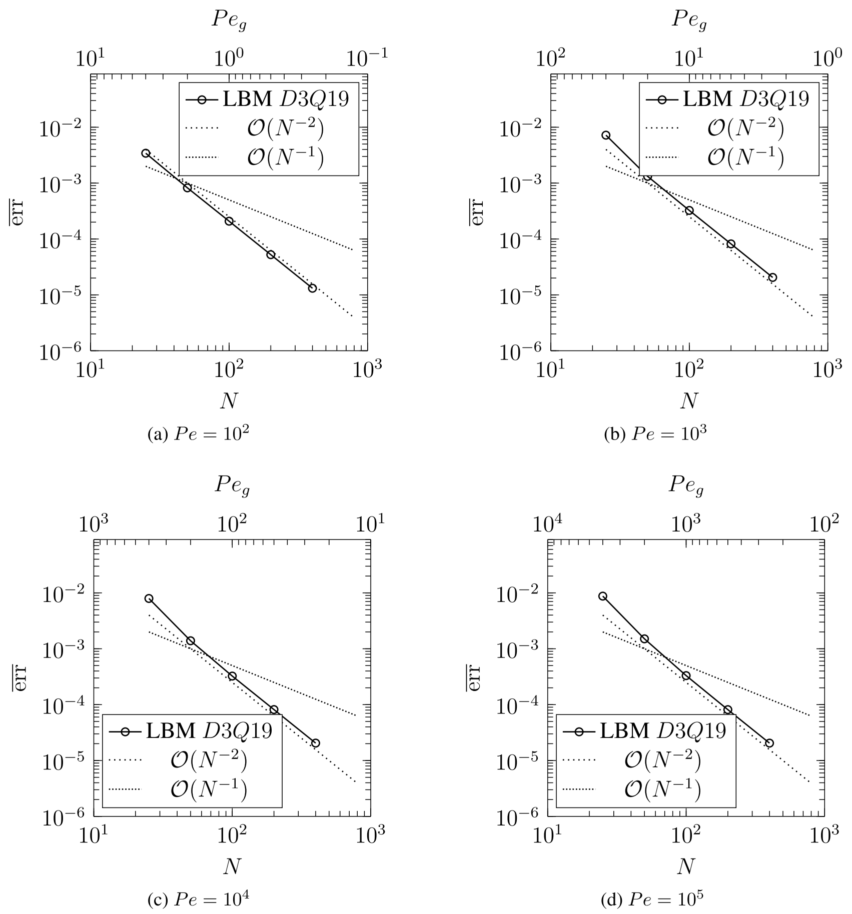
It can be demonstrated [23] that the benchmark possesses the following analytical solution:

$$\phi^{\mathrm{an}}(\boldsymbol{x}, 0) = \frac{1}{\sqrt{4\pi dt}}\sum_{k\in\mathbb{Z}}\exp\frac{-\left(x - x_0 - Ut + kL\right)^2}{4dt} + 1\ , \tag{50}$$

and the initial condition for the analytical solution is the Dirac's comb:

$$\phi^{\mathrm{an}}(\boldsymbol{x}, 0) = \sum_{k\in\mathbb{Z}}\delta\left(x - x_0 - kL\right)\ . \tag{51}$$

The proof of the mutual coherence of Eqs. (49), (50) and (51) follows from Section 5.3.1 and in the Supplementary Material because of the linearity of the advection–diffusion equation.



(a) $Pe = 10^2$

(b) $Pe = 10^3$

(c) $Pe = 10^4$

(d) $Pe = 10^5$

Fig. 3. Convergence of the Lattice-Boltzmann model for several values of Pe, smooth start.

6. Results

A series of numerical simulations reproducing the smooth-start benchmark was performed on a Lenovo X380 laptop equipped with a 4-core, 8-thread Intel Core i5-8250U CPU. Another series, reproducing the sharp-start, was performed on a Dell XPS 13 9600 laptop equipped with a 4-core, 8-thread Intel Core i7-1065G7. A third series of simulations for testing purposes was computed on a maximum of 10 nodes with two deca-core Intel Xeon E5-2660 v3 each. Finally, the scaling-up tests were performed on three 40-core Lenovo ThinkSystem SR65 units.

6.1. Smooth start: Lattice-Boltzmann

The convergence behaviour of the Lattice-Boltzmann model for advection–diffusion on the smooth start case is reported in Fig. 3. The second-order convergence is evident.

6.2. Sharp start on the periodical benchmark: Lattice-Boltzmann

The convergence behaviour of the Lattice-Boltzmann model for advection–diffusion on the smooth start case is reported in Fig. 4. A deterioration of the convergence from second to first order can be observed, as noted by [16]. However, such deterioration occurs only for $\mathrm{Pe_g} \lesssim 100$—that is, a relaxed version of the stability criterion of Eq. (25). In other words, the violation of the above-mentioned criterion leads to results that still retain stability, but at the price of a lower order of convergence.

Concerning the second stability criterion (Eq. (28)), it can be verified that the simulations with Pe = 10 000 and Pe = 100 000 fail the second line of Eq. (28), despite the fact that the simulations with Pe = 10 000 and $\mathrm{Pe_g} \lesssim$ still display second-order convergence.

6.3. Sharp start on the non-periodical lattice: Lattice-Boltzmann

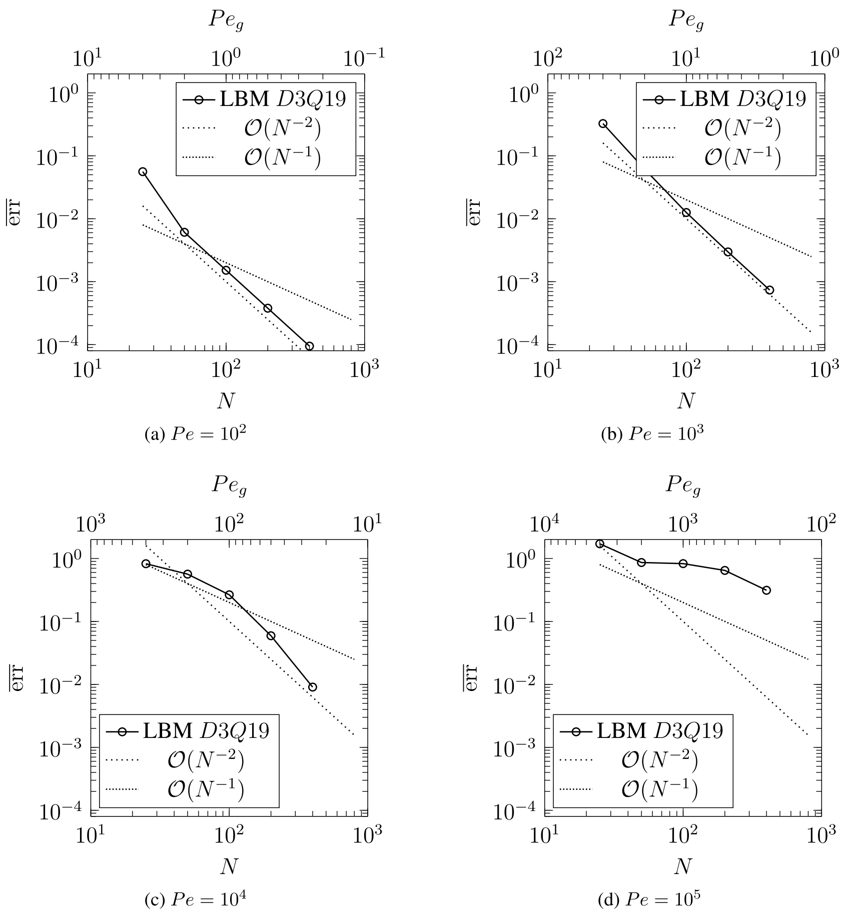
The convergence behaviour of the Lattice-Boltzmann model for advection–diffusion with $\mathrm{Pe_g} = 0$ on the sharp start case is reported in Fig. 5. The value of diffusivity is set to $d = 0.01\,\mathrm{m^2/s}$. In this case, the first-order convergence is evident, as in [16].

In Fig. 6(a), the value of ϕ along the x coordinate is plotted for $U = 0$ and for values of d comprised between 0.001 and 1 $\mathrm{m^2/s}$. The error is shown to be below 4%. However, Fig. 6(b) evidences the presence of oscillations for $d = 0\,\mathrm{m^2/s}$.

The results of tests performed on $U > 0$, corresponding to Co = 0.00909, are shown in Fig. 7. The corresponding values of d, Pe, $\mathrm{Pe_g}$ and κ^* are reported in Table 1. In all the cases, the computational time was between 20 and 25 CPUs. Fig. 7(a) shows the onset of oscillations for $\mathrm{Pe} \gtrapprox 400$, which results in an increase in the error (Fig. 7(b)). From this value of Pe onwards, the simulations are also found to fail the stability criterion of Eq. (28). Oscillations dominate the figure for Pe → ∞ and eliminate the model's predictive power.

6.4. Sharp start: finite-difference

The convergence behaviour of the explicit finite-difference model is reported in Figs. 8 (for diffusive-only systems) and 9 (for $U > 0$). In the case $U > 0$, the same values of d, Pe and $\mathrm{Pe_g}$ of Section 6.3 and Table 1 are used. Nd = 50 in all the runs, while NU is infinite in the $U = 0$ runs (Fig. 8) and spans from 2 to 11.6 in the $U > 0$ case (Fig. 9). Therefore, the stability constraints reported in Eqs. (35) and (36) are met. Second-order convergence is evident for both the central and the upwind advection schemes in the zero advective velocity case (Fig. 8). This is to be expected, as the only scheme in place is the second-order diffusion scheme when advection is not present. In the case of



(a) $Pe = 10^2$

(b) $Pe = 10^3$

(c) $Pe = 10^4$

(d) $Pe = 10^5$

Fig. 4. Convergence of the Lattice-Boltzmann model for several values of Pe, sharp start, periodical benchmark.

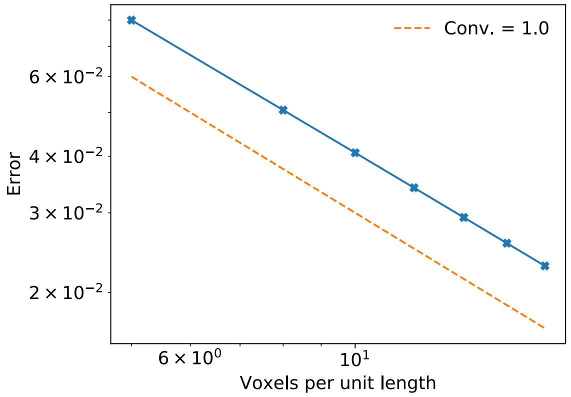


Fig. 5. Convergence of the Lattice-Boltzmann model for $U = 0$, sharp start.

Table 1
Parameters of the Lattice-Boltzmann simulations for $U > 0$.

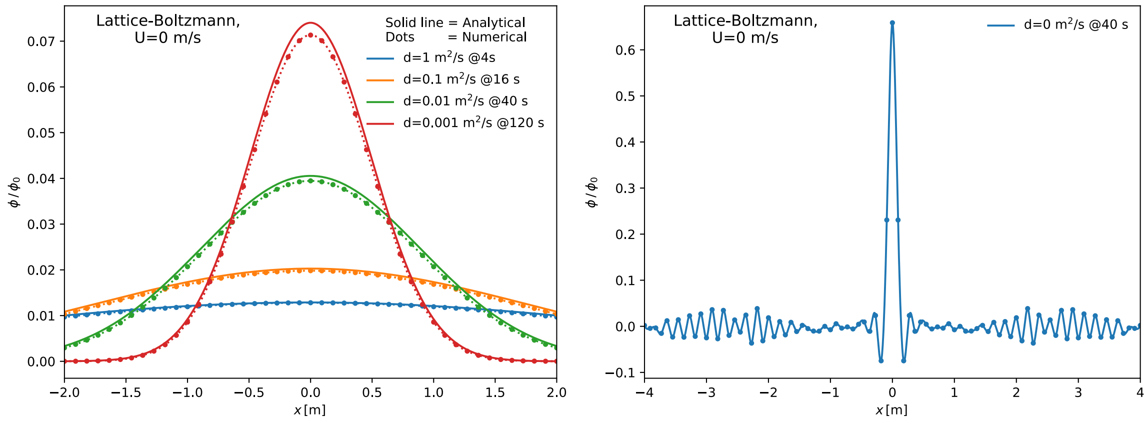
d (m^2/s)	Pe	Pe$_g$	κ^*
0.01	10	0.909	0.54
0.005	20	1.818	0.52
0.002	50	4.545	0.508
0.001	100	9.09	0.504
0.0005	200	18.18	0.502
0.0003	333	30.3	0.5012
0.00025	400	36.4	0.501
0.0002	500	45.5	0.5008
0.0001	1 000	90.9	0.5004
0	∞	∞	0.5

non-zero advection (Fig. 9), the central scheme displays second-order convergence, while the upwind converges to the first order for $N_y \geq 20$.

Considering numerical simulations were run for $U > 0$, as expected, divergent behaviour is observed almost immediately for Nd $\lesssim$ 0.5 regardless of the advection scheme. Consequently, only simulations with Nd > 50 are reported in the rest of this article.

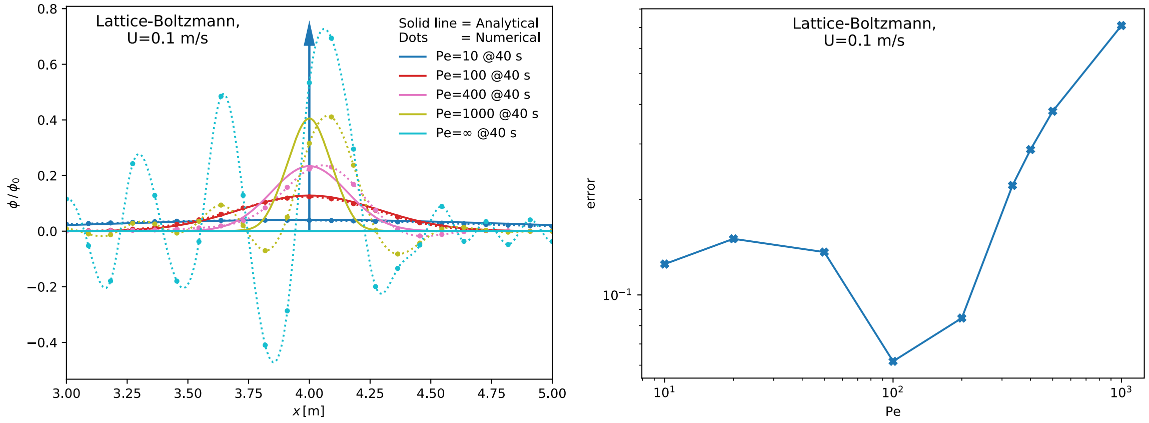
Fig. 10 shows the outcome of simulations where the central advection scheme is used. In the cases with $U = 0.1\,\mathrm{m/s}$, the computational time was between 20 and 25 CPUs. Oscillations appear at Pe $\simeq$ 100 (Fig. 10(a)), and eliminate the model's predictive power for higher values of Pe (Fig. 10(b)). A comparison between Figs. 7(b) and 10(c) shows that the error associated to the simulations is larger than the one experienced in the Lattice-Boltzmann case.

The simulations performed with the upwind scheme are reported in Fig. 11. In the cases with $U = 0.1\,\mathrm{m/s}$, the computational time was between 20 and 25 CPUs. Contrary to the upwind case (Fig. 10) and the Lattice-Boltzmann model (Fig. 7), the oscillations are suppressed and the instabilities removed—crucially, the model's predictive power is maintained also in the case Pe → ∞. As predicted, the numerical diffusivity is relevant: a comparison between Fig. 11(c), Figs. 7(b) and 10(c) shows that the error associated to the upwind finite-difference simulations up to Pe = 1 000 is comparable (albeit slightly smaller) to the central finite-difference, and larger than the one experienced in the Lattice-Boltzmann case.



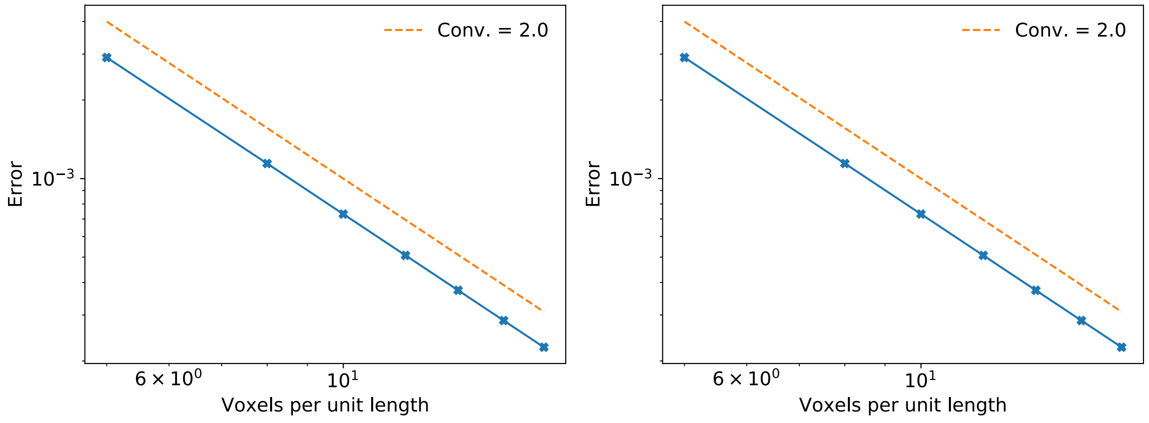
(a) (b)

Fig. 6. Lattice-Boltzmann model for $U = 0$ and $N_y = 11$, sharp start. 6(a): Non-zero diffusivity. The error amounted to 3.95% in the $d = 1$ m^2/s run, 3.69% in the $d = 0.1$ m^2/s, 3.68% in the $d = 0.01$ m^2/s and 3.86% in the $d = 0.001$ m^2/s at the time indicated in the legend. 6(b): Zero diffusivity.



(a) (b)

Fig. 7. Lattice-Boltzmann model for $U > 0$ and $N_y = 11$, sharp start. 7(a): $U = 0.1$ m/s at 40 s. 7(b): Error over Pe for $U = 0.1$ m/s at 40 s.



(a) (b)

Fig. 8. Convergence of the finite-difference model for $U = 0$ m/s and $d = 0.01$ m^2/s, sharp start. 8(a): central advection scheme. 8(b): upwind advection scheme.

6.5. Sharp start: finite-difference. negative-diffusivity correction

The convergence behaviour of the explicit finite-difference method with upwind advection scheme and negative-diffusivity correction is reported in Fig. 12. The model with the negative-diffusivity correction (Fig. 12(b)) is shown to display the same order of convergence of the pure upwind (Fig. 12(a)), but is able to display convergent behaviour at coarser meshes.

In Fig. 13, the action of the variation of the negative-diffusivity parameter α over the results of the simulations is displayed. In all cases, the computational time was between 25 and 35 CPUs with peaks up to 67 CPUs. The results show that the numerical diffusivity cannot be

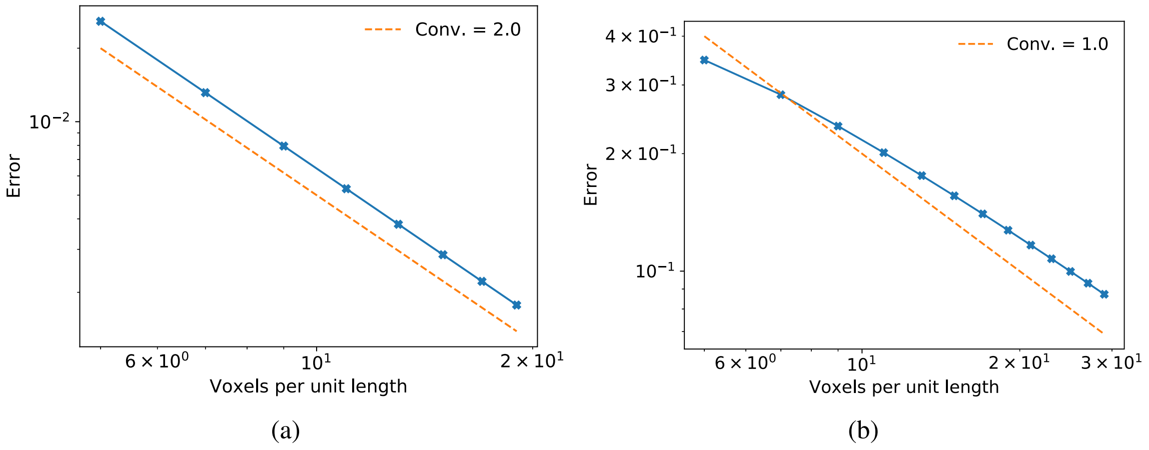


Fig. 9. Convergence of the finite-difference model for $U = 0.05\,\mathrm{m/s}$ and $d = 0.01\,\mathrm{m^2/s}$, sharp start. 9(a): central advection scheme. 9(b): upwind advection scheme.

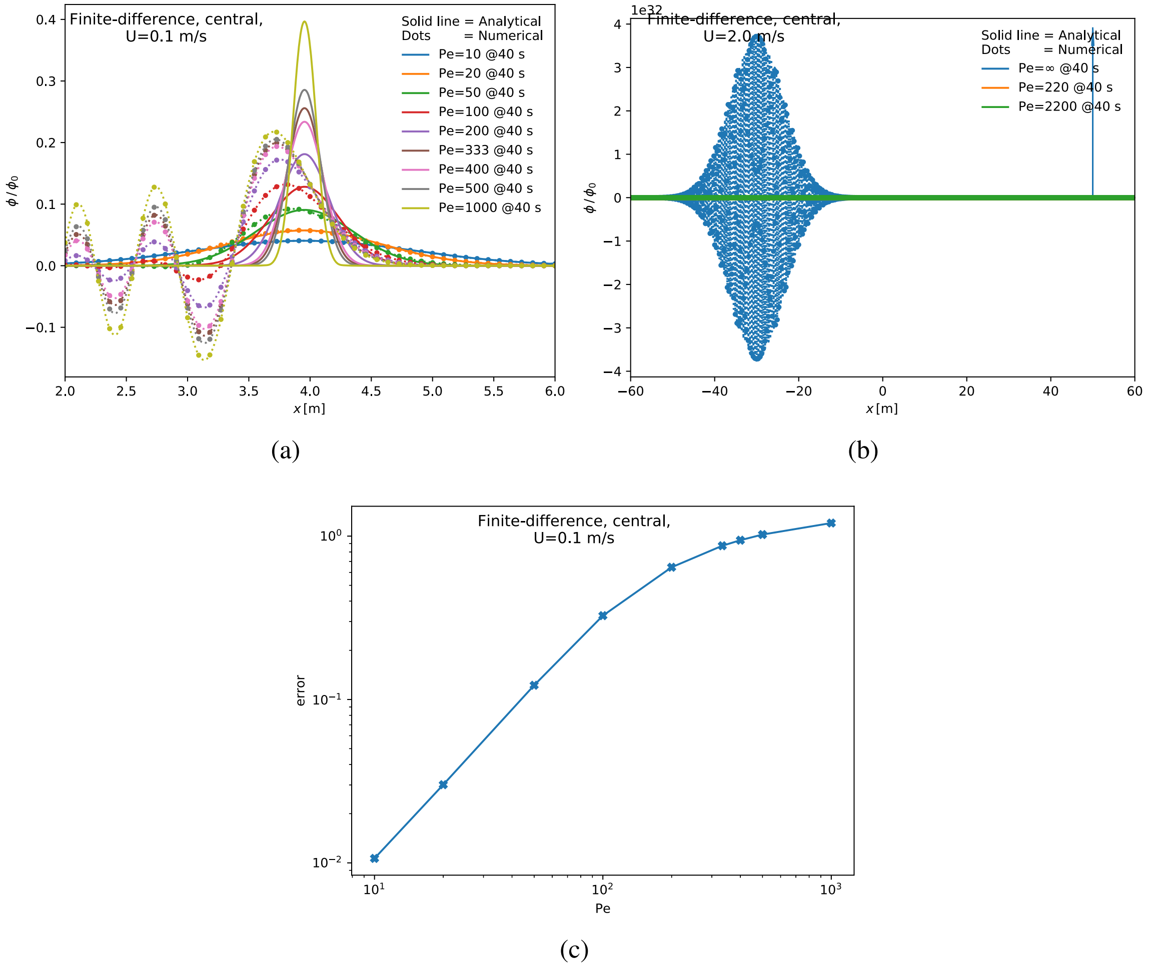
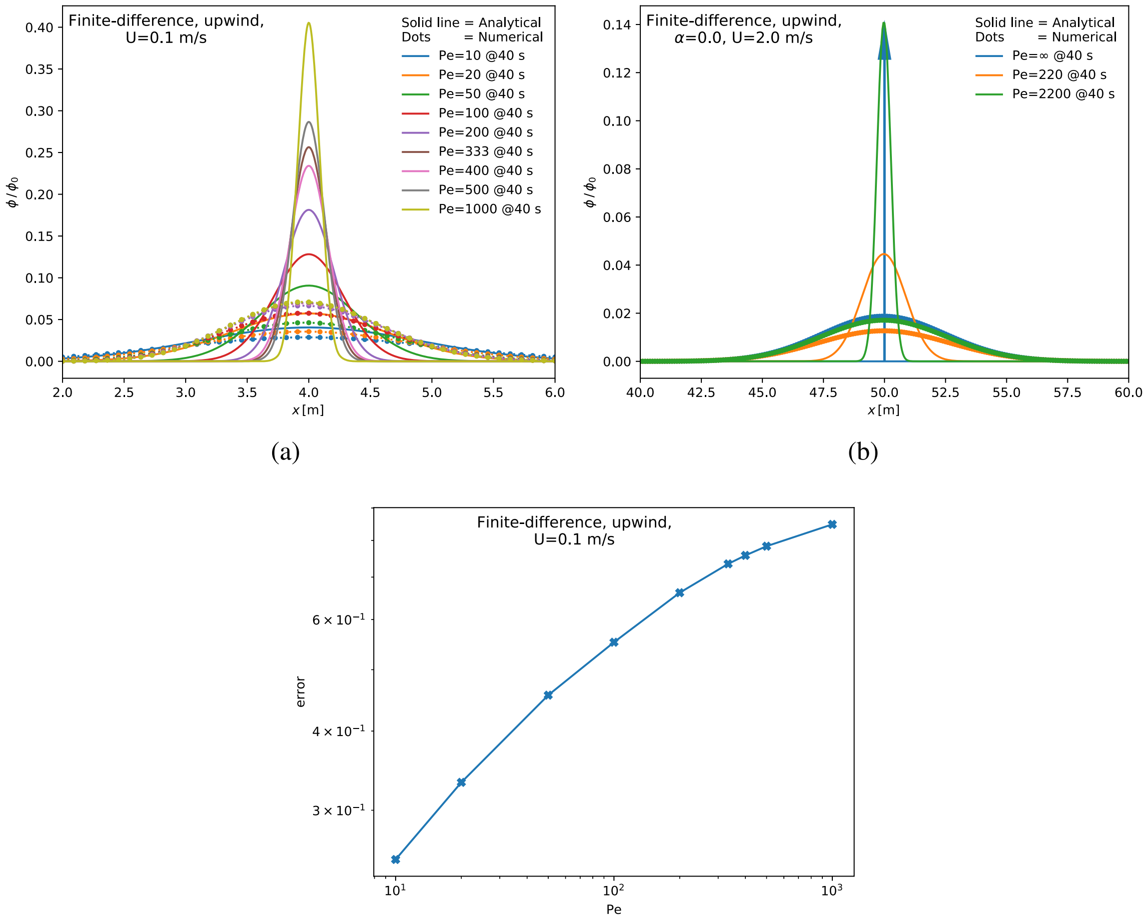


Fig. 10. Explicit finite-difference model for $U > 0$ and $N_y = 11$, sharp start, central differentiation scheme for advection. 10(a): $U = 0.1\,\mathrm{m/s}$ at 40 s. 10(b): $U = 2.0\,\mathrm{m/s}$ at 40 s. 10(c): Error over Pe for $U = 0.1\,\mathrm{m/s}$ at 40 s.

removed; however, it is shown that the introduction of the negative-diffusivity term allows a significant reduction of the error of up to 50%. Fig. 13(f) shows that, for Pe $\gtrapprox 1\,000$, values of α larger than 0.8 result in the appearance of oscillations; it is therefore necessary to strike a balance between precision (*viz.*, suppression of the numerical diffusion) and stability (that is, preventing oscillations). This balance is achieved by minimizing the value of the error as a function of α and is achieved for $\alpha = 0.8145 \pm 0.0025$ for Pe = 10, $\alpha = 0.890 \pm 0.001$ for Pe = 100 and $\alpha = 0.769 \pm 0.001$ for Pe = 1000.

6.6. *Upscaling tests for the non-periodical case*

In Fig. 14, a comparison between the Lattice-Boltzmann and the finite-difference submodels in terms of upscaling is reported. The results are given in terms of MLUPps (millions of lattice operations per

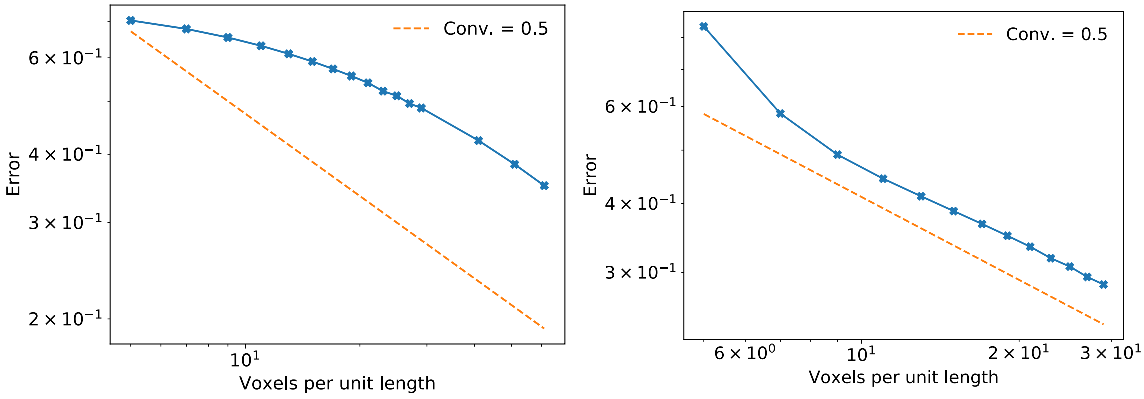


(a)

(b)

(c)

Fig. 11. Explicit finite-difference model for $U > 0$ and $N_y = 11$, sharp start, upwind differentiation scheme for advection. 11(a): $U = 0.1\,\mathrm{m/s}$ at 40 s. 11(b): $U = 2.0\,\mathrm{m/s}$ at 40 s. 11(c): Error over Pe for $U = 0.1\,\mathrm{m/s}$ at 40 s.



(a)

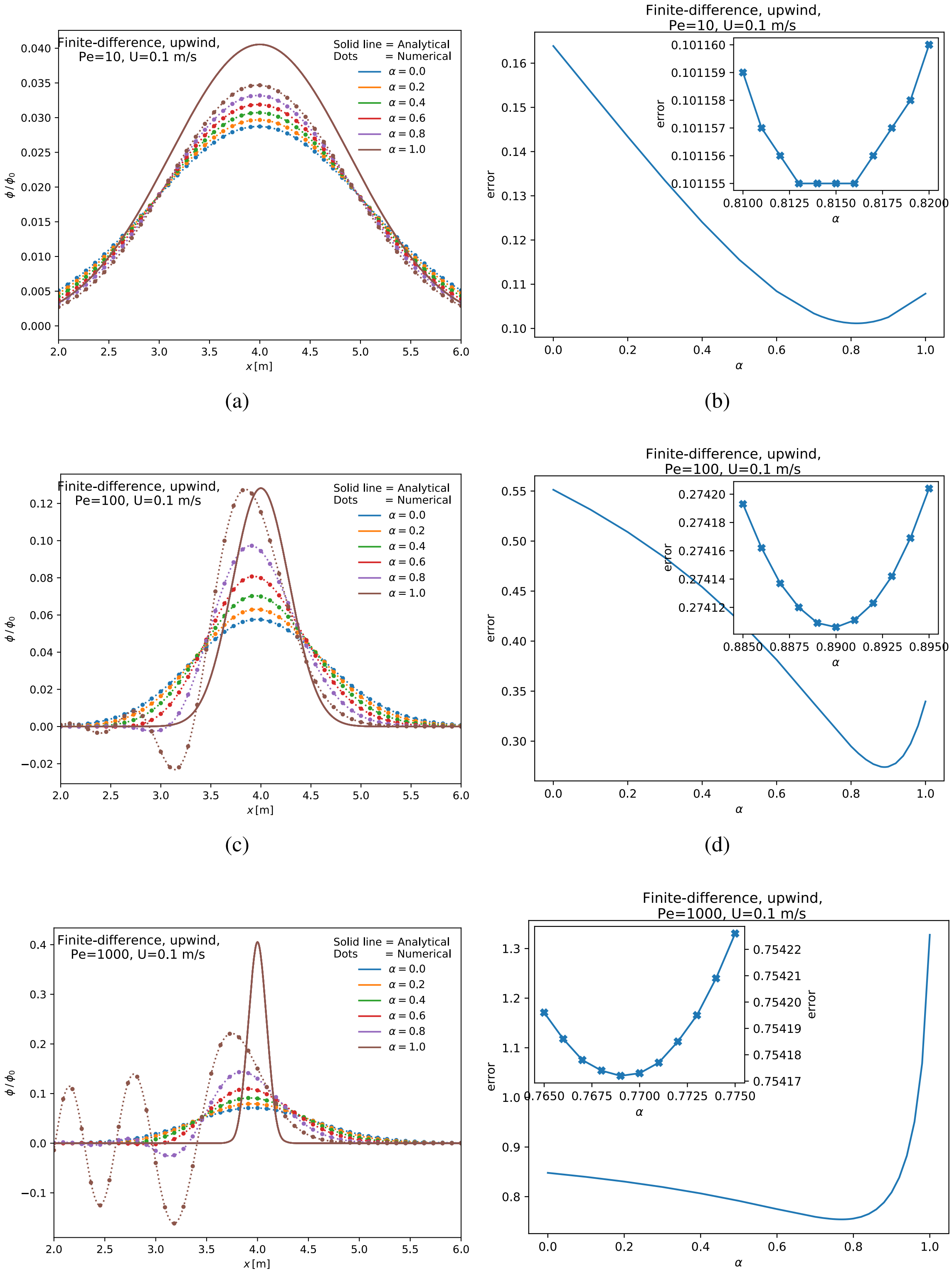
(b)

Fig. 12. Convergence of the finite-difference model for $U = 2\,\mathrm{m/s}$ and $N_y = 11$, sharp start, corrected upwind scheme. 12(a): $\alpha = 0$ correction (*viz.*, pure upwind scheme). 12(b): $\alpha = 0.6$ correction.

processor per second). The speed-up of a given lattice at a number of processors n_1 with respect to the initial number of processors n_0 is given as:

$$S = \frac{n_1 \mathrm{MLUPps}\left(n_1\right)}{n_0 \mathrm{MLUPps}\left(n_0\right)} . \tag{52}$$

That gives, for instance, the value $S = 4.012$ for 120 processors against a theoretical speed-up of $S^{\mathrm{th}} \equiv n_1/n_0 = 6$ in the finite-difference model for the $N = 317^3$ lattice. It can be seen that both the model display similar upscaling characteristics. This proves that the models behave similarly in terms of parallelizability.



(a)

(b)

(c)

(d)

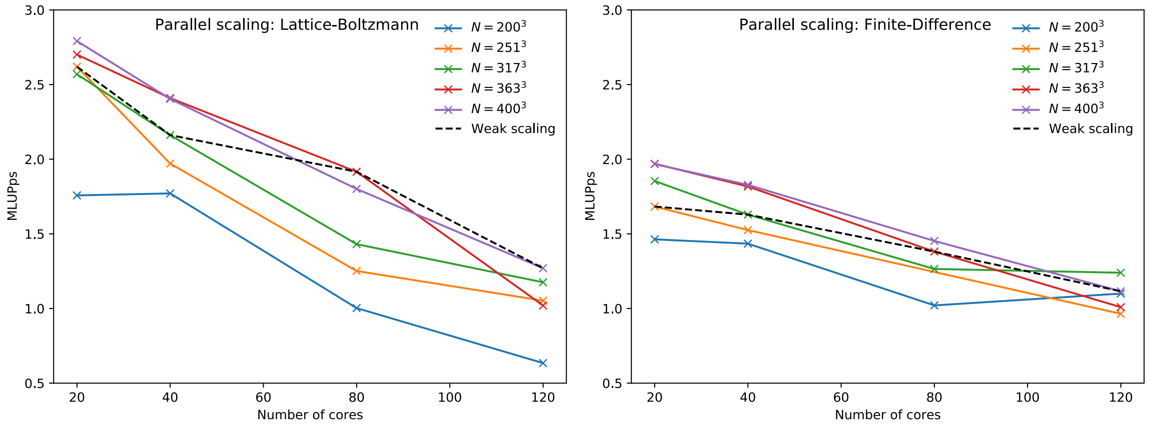
(e)

(f)

Fig. 13. Convergence of the finite-difference model for $U = 0.1\,\mathrm{m/s}$ and $N_y = 11$, sharp start, corrected upwind scheme. 13(a): Pe = 10 at 40 s. 13(b): Pe = 10: Error over α at 40 s. 13(c): Pe = 100 at 40 s. 13(d): Pe = 100: Error over α at 40 s. 13(e): Pe = 1000 at 40 s. 13(f): Pe = 1000: Error over α at 40 s.

7. Discussion

The Lattice-Boltzmann advection–diffusion simulations with smooth initial conditions (Section 6.1) display the expected second-order convergence. The Lattice-Boltzmann advection–diffusion simulations with non-smooth initial conditions (Section 6.3), are found to retain second-order convergence if they respect relaxed versions of the stability conditions proposed in the literature; otherwise, they demonstrate a deterioration of convergence from second to first order. Specifically, the $\mathrm{Pe_g}$ criterion (Eq. (27)) is found to be determinant, as transition



(a)

(b)

Fig. 14. Scaling tests for the non-periodical case. 14(a): Lattice-Boltzmann Navier–Stokes solver coupled with the Lattice-Boltzmann advection–diffusion solver. 14(b): Lattice-Boltzmann Navier–Stokes solver coupled with the finite-difference advection–diffusion solver.

from second to first order of convergence is observed to occur for $\mathrm{Pe_g} \gtrsim 100$ and in the case $\mathrm{Pe_g} = 0$ (which makes it reasonable to suppose that the same happens in a sufficiently small neighbour of $\mathrm{Pe_g} = 0$). Conversely, the stability criterion based on the simplified version of [20]'s observation (Eq. (28)) is found to be either irrelevant or a duplication of the $\mathrm{Pe_g}$ criterion for the numerical setup taken in consideration within this work.

Lattice-Boltzmann and explicit finite-difference models are shown to be equivalent in terms of computational expense. The upwind finite-difference with the negative-diffusivity correction constitutes an exception, but the increase in computational expense is within 50%. Peaks of computational expense should be considered carefully, as they are likely to be generated by sudden increases of resource requirements in the laptop's operating system. Therefore, the comparison between Lattice-Boltzmann and finite-difference models is limited to numerical precision and predictive power.

A comparison of the error over Pe and a conversion of the latter into $\mathrm{Pe_g}$ for the Lattice-Boltzmann (Fig. 7(b)), the central (Fig. 10(c)) and the upwind finite-difference models (Fig. 11(c)) in the context of sharp initial conditions shows that: (*i*) for $\mathrm{Pe_g} < 10$, the finite-difference model with the central scheme for advection produces the smallest amount of error; (*ii*) for $10 < \mathrm{Pe_g} < 100$, the Lattice-Boltzmann is superior; and (*iii*) the finite-difference model with the upwind scheme for advection is never competitive. These statements are justified by the following considerations: (*i*) the central finite-difference performs better than the other methods for $\mathrm{Pe_g} < 10$ because it displays second-order convergence whereas the upwind is only first-order convergent, and Lattice-Boltzmann's convergence deteriorates to the first order for $\mathrm{Pe_g} \to 0$; (*ii*) having the same (*viz.*, second) order of convergence, the central finite-difference performs worse than the Lattice-Boltzmann for $10 < \mathrm{Pe_g} < 100$ because of the onset of oscillations (see Fig. 10(a)), and this is predictable by comparing Eq. (25) to Eq. (38); (*iii*) the upwind finite-difference model is inferior to the Lattice-Boltzmann because of its numerical diffusivity (see Fig. 11(a)).

A comparison between Figs. 7(a), 10(a) and 11(a) shows that the Lattice-Boltzmann and the central finite-difference models rapidly lose their predictive power when $\mathrm{Pe_g} > 100$ because of the oscillations, whereas the upwind finite-difference remains stable up to $\mathrm{Pe_g} \to \infty$. Finally, Fig. 13 shows that the introduction of the negative-diffusivity correction brings about a significant reduction of the numerical diffusivity error. However, oscillations are re-introduced if the weight of the correction is too high, *i.e.*: $\alpha \to 1$. Therefore, the optimum value of α must be chosen in order to reduce the error as much as possible, whilst maintaining system stability through oscillation removal. The proposed method of numerically finding the minimum of the error in function of α is shown to successfully provide this balance. However, the magnitude of the oscillations – and therefore the optimum value of α – depends on the values of Pe; this is exemplified by the fact that three different optima were found for three different values of Pe. Consequently, the operation of finding the optimum value of α by finding the minimum value of the error should be repeated for each application of the method, and each set of data if possible.

8. Conclusions

The Lattice-Boltzmann advection–diffusion model is shown to possess advantages in terms of parallelizability comparable to pure Lattice-Boltzmann, and to exhibit second order of convergence for smooth initial conditions. For non-smooth initial conditions, the order of convergence is shown to primarily depend on $\mathrm{Pe_g}$. As this work is primarily concerned on assessing the dependence of Lattice-Boltzmann and explicit finite-difference models for advection–diffusion from Pe and $\mathrm{Pe_g}$ with the explicit case Pe, $\mathrm{Pe_g} \to \infty$ in mind, the dependence of Lattice-Boltzmann advection–diffusion's order of convergence from Co and κ^* has not been thoroughfully assessed. Further work is required to address this specific gap and will also (*i*) investigate Lattice-Boltzmann's order of convergence for $\mathrm{Pe_g} \to 0$, and (*ii*) find the critical value for $\mathrm{Pe_g}$ in the smooth-initial condition case.

Based on the considerations reported in Section 7 on the mutual comparison between central finite-difference, upwind finite-difference and Lattice-Boltzmann advection–diffusion models, the following recommendations are given.

a. $\mathrm{Pe_g} < 10$: use explicit finite-difference with the central differentiation scheme for advection.
b. $10 < \mathrm{Pe_g} < 100$: use Lattice-Boltzmann.
c. $100 < \mathrm{Pe_g} < \infty$: use explicit finite-difference with the upwind differentiation scheme for advection and the negative-diffusivity correction. Tune the parameter α in order to strike the optimum balance between stability and precision (*viz.*, find the minimum of the error as a single-variable function of α).

Further work will also include the application of the explicit finite-difference models to the continuous boundary condition benchmark, and the extension of the comparative work to models implementing multiple-relaxation-time and two-relaxation-time Lattice-Boltzmann for advection–diffusion.

CRediT authorship contribution statement

Davide Dapelo: Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Writing - original draft, Visualization, Funding acquisition. **Stephan Simonis:** Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Writing - review & editing, Visualization. **Mathias J. Krause:** Software, Resources, Data curation, Writing - review & editing. **John Bridgeman:** Conceptualization, Resources, Writing - review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

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 Bradford High Performance Computing (HPC) Service, where part of the computational work was carried out. The authors would also like to thank the Steinbuch Centre for Computing at KIT for providing access to their high performance computer ForHLR II, where part of the computations were carried out as part of the CPE project. The research leading to these results has received funding from the German Research Foundation (DFG 436212129).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.jocs.2021.101363.

References

[1] B. Andersson, R. Andersson, L. Håkansson, M. Mortensen, N. Defence, R. Sudiyo, B. van Wachem, Computational Fluid Dynamics for Engineers, Cambridge University Press, Cambridge, UK, 2012.
[2] S. Succi, Lattice Boltzmann 2038, Europhys. Lett. 109 (5) (2015) 50001.
[3] P. Lallemand, L.s. Luo, M. Krafczyk, W.a. Yong, The lattice Boltzmann method for nearly incompressible flows, J. Comput. Phys. (2020) 109713.
[4] M.J. Krause, A. Kummerlaender, S.J. Avis, H. Kusumaatmaja, D. Dapelo, F. Klemens, M. Gaedtke, N. Hafen, A. Mink, R. Trunk, J.E. Marquardt, M.-L. Maier, M. Haussmann, S. Simonis, OpenLB-Open source lattice Boltzmann code, Comput. Math. Appl. 81 (2021) 258–288.
[5] D. Dapelo, R. Trunk, M.J. Krause, J. Bridgeman, Towards Lattice–Boltzmann modelling of unconfined gas mixing in anaerobic digestion, Comput. & Fluids 180 (2019) 11–21.
[6] D. Dapelo, F. Alberini, J. Bridgeman, Euler–Lagrange CFD Modelling of unconfined gas mixing in anaerobic digestion, Water Res. 85 (2015) 497–511.
[7] E.G. Flekkoy, Lattice Bhatnagar–Gross–Krook models for miscible fluids, Phys. Rev. E (1993) http://dx.doi.org/10.1103/PhysRevE.47.4247.
[8] D. Wolf-Gladrow, A lattice Boltzmann equation for diffusion, J. Stat. Phys. 79 (5–6) (1995) 1023–1032.
[9] T. Krüger, H. Kasumaatmaja, A. Kuzmim, O. Shardt, G. Silva, E.M. Viggen, The Lattice Boltzmann Method, No. March 2015, Springer, 2017, pp. 1–18, http://dx.doi.org/10.1007/978-3-319-44649-3, arXiv:arXiv:1011.1669v3.
[10] I. Ginzburg, Equilibrium-type and link-type lattice Boltzmann models for generic advection and anisotropic-dispersion equation, Adv. Water Resour. 28 (11) (2005) 1171–1195.
[11] N.I. Prasianakis, S.S. Chikatamarla, I.V. Karlin, S. Ansumali, K. Boulouchos, Entropic lattice Boltzmann method for simulation of thermal flows, Math. Comput. Simulation 72 (2–6) (2006) 179–183.
[12] L. Zheng, B. Shi, Z. Guo, Multiple-relaxation-time model for the correct thermohydrodynamic equations, Phys. Rev. E 78 (2) (2008) http://dx.doi.org/10.1103/PhysRevE.78.026705.
[13] L. Deng, Y. Zhang, Y. Wen, B. Shan, H. Zhou, A fractional-step thermal lattice Boltzmann model for high Peclet number flow, Comput. Math. Appl. 70 (5) (2015) 1152–1161.
[14] B. Servan-Camas, F.T. Tsai, Lattice Boltzmann method with two relaxation times for advection-diffusion equation: Third order analysis and stability analysis, Adv. Water Resour. 31 (8) (2008) 1113–1126.
[15] B. Servan-Camas, F.T. Tsai, Non-negativity and stability analyses of lattice Boltzmann method for advection-diffusion equation, J. Comput. Phys. 228 (1) (2009) 236–256.
[16] J.P. Weiss, Numerical Analysis of LBMs for the Heat Equation on a Bounded Interval (Ph.D. thesis), Universitätsverlag Karlsruhe, 2006, p. 208.
[17] M.J. Krause, S. Avis, D. Dapelo, N. Hafen, M. Haußmann, M. Gaedtke, F. Klemens, A. Kummerländer, M.-L. Maier, A. Mink, J. Ross-Jones, S. Simonis, R. Trunk, OpenLB Release 1.3: Open Source Lattice Boltzmann Code, Zenodo, 2019, http://dx.doi.org/10.5281/zenodo.3625967.
[18] P.L. Bhatnagar, E.P. Gross, M. Krook, A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems, Phys. Rev. 94 (3) (1954) 511–525.
[19] S. Simonis, S. Simonis, M. Frank, M.J. Krause, On relaxation systems and their relation to discrete velocity Boltzmann models for scalar advection-diffusion equations, Phil. Trans. R. Soc. A (2020) 20190400.
[20] X.D. Niu, C. Shu, Y.T. Chew, T.G. Wang, Investigation of stability and hydrodynamics of different lattice Boltzmann models, J. Stat. Phys. 117 (3–4) (2004) 665–680.
[21] C. Hirsch, Numerical Computations of Internal and External Flows, Vol. 1, Butterworth-Heinemann, Oxford, 2007.
[22] A. Mojtabi, M.O. Deville, One-dimensional linear advection-diffusion equation: Analytical and finite element solutions, Comput. & Fluids 107 (2015) 189–195.
[23] D. Kuzmin, A Guide to Numerical Methods for Transport Equations, University Erlangen-Nuremberg, 2010, https://www.mathematik.uni-dortmund.de/~kuzmin/Transport.pdf.

Davide Dapelo After completing his degree in theoretical physics in Genoa, Italy, Davide started his Ph.D. in civil engineering in Birmingham in 2012 and completed it in 2016. Since 2015, Davide has been employed in a number of postdoctoral positions in Birmingham and in Bradford. Davide's scientific interest has been numerical modelling of multiphase non-Newtonian flow applied in water managing and, more specifically, anaerobic digestion for wastewater treatment. After using and developing finite-volume CFD models, Davide has focussed his attention to developing new models and methodologies for the Lattice-Boltzmann method.

Stephan Simonis is a doctoral student at the Institute for Applied and Numerical Mathematics, at Karlsruhe Institute of Technology (KIT), and contributes to the open source C++ library OpenLB. He obtained a B.Sc. and M.Sc. in Mathematics at KIT in Germany and KTH Royal Institute of Technology in Sweden. His research focuses on the mathematical background and numerical aspects of discrete velocity models, relaxation systems and lattice Boltzmann methods.

Mathias J. Krause studied mathematics with economics at the Universität Karlsruhe (TH) in Germany and the Cardiff University in Wales (2006). In 2010, he received a doctorate at the Karlsruhe Institute of Technology (KIT). Since 2013, he heads the interdisciplinary Lattice Boltzmann Research Group (LBRG) at KIT. His research interests are mainly dedicated to the fields of applied mathematics, applied computer science with focus on HPC, CFD and optimization under the constraints of PDE. He is initiator and main author of OpenLB. His work was honoured with several prizes and a membership as WIN-Kollegiat at the Heidelberg Academy of Science.

John Bridgeman is Deputy Vice-Chancellor (Research, Innovation & Engagement) and Professor of Environmental Engineering at the University of Bradford, UK. He is a Chartered Civil Engineer and a Chartered Scientist, and a Fellow of the Institution of Civil Engineers, the Chartered Institution of Water and Environmental Management, the International Water Association and the Institute of Mathematics and its Applications. Bridgeman's research focuses on experimental and numerical approaches to address water security and resource efficiency. Areas of interest include CFD and lattice Boltzmann modelling of water and wastewater treatment processes, and the development of novel water quality assessment tools.