A time-parallel framework for coupling finite element and lattice Boltzmann methods

Matteo Astorino, Franz Chouly, Alfio Quarteroni

A time-parallel framework for coupling finite element and lattice Boltzmann methods

M. Astorino[a], F. Chouly[b*], and A. Quarteroni[a,c]

[a] *CMCS, Chair of Modelling and Scientific Computing, MATHICSE, Mathematics Institute of Computational Science and Engineering, Ecole Polytechnique Fédérale de Lausanne, Station 8, CH-1015 Lausanne, Switzerland.*

[b] *Laboratoire de Mathématiques de Besançon - UMR CNRS 6623, Université de Franche Comté, 16 route de Gray, 25030 Besançon Cedex, France.*

[c] *MOX, Modeling and Scientific Computing, Department of Mathematics, Politecnico di Milano, Via Bonardi 9, 20133 Milano, Italy.*

Abstract: In this work we propose a new numerical procedure for the simulation of time-dependent problems based on the coupling between the finite element method and the lattice Boltzmann method. The procedure is based on the Parareal paradigm and allows to couple efficiently the two numerical methods, each one working with its own grid size and time-step size. The motivations behind this approach are manifold. Among others, we have that one technique may be more efficient, or physically more appropriate or less memory consuming than the other depending on the target of the simulation and/or on the sub-region of the computational domain. Furthermore, the coupling with finite element method may circumvent some difficulties inherent to lattice Boltzmann discretization, for some domains with complex boundaries, or for some boundary conditions. The theoretical and numerical framework is presented for parabolic equations, in order to describe and validate numerically the methodology in a simple situation.

KEY WORDS finite elements, lattice Boltzmann, coupling of numerical methods, Parareal, parallel-in-time domain decomposition

1 Introduction

Despite the advances in computer technology and breakthroughs in numerical methods, still there are important challenges in the simulation of physical phenomena either described by the coupling of different physical systems and/or that span across many spatiotemporal scales. Example of these problems can be found in diverse fields such as thermodynamics [55], material science [66, 39], fluid flows [24], fluid-structure interaction [17, 25], fluid-porous medium [18, 52], biology [28] and biomechanics [65], just to mention a few.

For these problems, the challenges associated with the development of the numerical frameworks are essentially twofold. On the one hand, the coupling of the different equations and variables describing the process must be mathematically consistent. Classical examples are FSI or geometrical multiscale. On the other hand, a coupling of different numerical methods may sometimes be required. This last point is typically experienced during the coupling of commercial solvers based on different numerical methods, or in the simulation of multiscale problems, when the spatiotemporal complexity of the problem cannot be taken into account by a single numerical method.

In the field of the lattice Boltzmann method (LBM), results focusing on the coupling with microscopic method have been presented for instance in nanoflows through disordered media [56], DNA translocation [28, 48] and dense fluids [19, 20]. On the contrary, the coupling between LBM and conventional macroscopic methods such as the Finite Element Method (FEM), the Finite Difference Method (FDM) and the Finite Volume Method (FVM) have been addressed only in a very few works. In [1, 2] a FDM-LBM coupling has been presented for the heat equation. The same kind of coupling is applied also to reaction-diffusion problems in [62] and to the Navier-Stokes equations in [40, Chapter 7]. For the same fluid equations a FVM-LBM coupling has been considered in [44, 67].

In contrast with LBM, that is based on a mesoscopic description of physical phenomena in terms of particles distribution functions, methods such as FDM, FVM or FEM approximate the physics directly through macroscopic variables and equations. Coupling LBM and macroscopic standard methods is interesting since each one can be chosen locally depending on its relevence to simulate the physics, its computational efficiency and its associated memory cost [1, 2, 40]. Indeed, when compared to macroscopic methods, LBM can be more efficient, or, conversely, more expensive, accordingly to the considered test case [44]. For instance, some physical quantities such as stress tensors or particle current are easily obtained from lattice Boltzmann methods [1, 2].

*Correspondence to: franz.chouly@univ-fcomte.fr

On the other hand, these methods are more memory consuming than their macroscopic counterparts, since they involve more state variables [1, 2, 62]. It can be therefore computationally prohibitive to discretize a whole complex domain with LBM and a very refined space/time discretization.
Coupling between LBM and FEM has received no attention to the best of our knowledge, though it can bring new possibilites to solve some issues related to LBM discretization. For instance Dirichlet, Neumann or Robin boundary conditions can be treated without much difficulty within the FEM framework, even for a domain with a curved boundary (see, e.g., [22]). However, these standard boundary conditions need to be implemented with care for LBM, because they must be translated in terms of distribution functions, and do not correspond directly to distribution function streaming from the boundary to the bulk (see, e.g., [68]). For curved boundaries the treatment is more involved and interpolation or extrapolation is needed (see, e.g., [64], and see, e.g., [13, Chapter 3] for a comparison between existing methods, for straight and curved boundaries). Though various methods have been proposed, this is is still an open issue, that is object of recent research for complex problems such as simulation of blood flows [50].
The existing works on coupling between LBM and FDM/FVM share two main features, specifically the derivation of the coupling conditions and the coupling technique. The coupling conditions have been obtained by means of a Chapman-Enskog expansion (up to the appropriate order) of the lattice Boltzmann equation (see for example [1, 67]). For the coupling algorithm an overlapping Schwarz domain decomposition technique is adopted: the computational domain is decomposed in two subregions, one for the LBM, the other for the macroscale method. An overlapping interface region is used to exchange the coupling information between the two methods (see Figure 1). Moreover, only the case of uniform structured meshes with a constant ratio (of 1 or 2) between the two mesh sizes has been considered (see e.g. [2]).

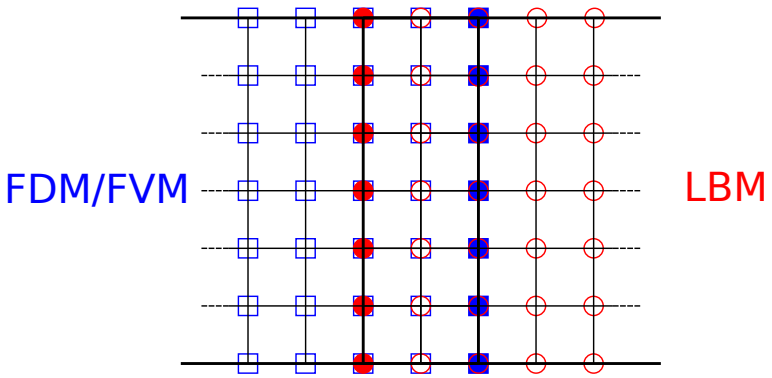


Figure 1: Sketch of the overlapping Schwarz method for the existing FDM/FVM coupling with LBM. Squares and circles identify the degrees of freedom (DOF) for the two methods, that are separately applied on the two regions. The filled squares and circles are the respective interface DOF.

In this work, we consider the coupling between the FEM and LBM. The choice of FEM is motivated by its well-assessed mathematical framework [54, 22] and by its flexibility in the treatment of complex boundary and interface conditions. To derive the transmission conditions between LBM and FEM, various techniques are available, among which, Chapman-Enskog expansion [1, 67], Grad's type distribution [14, 46] and asymptotic analysis [37, 11, 12]. We follow the works [37, 11, 12] on asymptotic analysis, though other choices would have left to the same results. The overall coupling algorithm introduces new features, namely:

1. the LBM is applied on a subregion of the whole computational domain where the problem is solved with FEM (Figure 2). This is motivated by possible applications to multi-scale problems, where multi-scale details may be localized in a patch, whereas a coarse grain (FEM) solver is applied globally. Additionally, the two spatial discretizations can be chosen independently (yielding non-conforming grids of arbitrary sizes). These characteristics are shared with numerical zoom methods, e.g. [30, 4, 32].
2. the coupling strategy relies on a time domain decomposition approach given by the Parareal algorithm [43, 23, 45, 29]. This naturally fits our evolutive problem and enhances the computational performance through parallelization in time. Different time discretizations may be chosen for the two methods provided that the timestep of FEM is a multiple of the LBM one.

In the aforementioned works, the time-coupling strategy is purely sequential. For simulations on long time intervals, this may be time-consuming, even more considering the restriction on the timestep of the LBM due to a CFL-type condition. This computational difficulty also appears in current numerical zoom techniques when applied to evolution equations, since multiple Schwarz iterations are required at each time step. Conversely, our

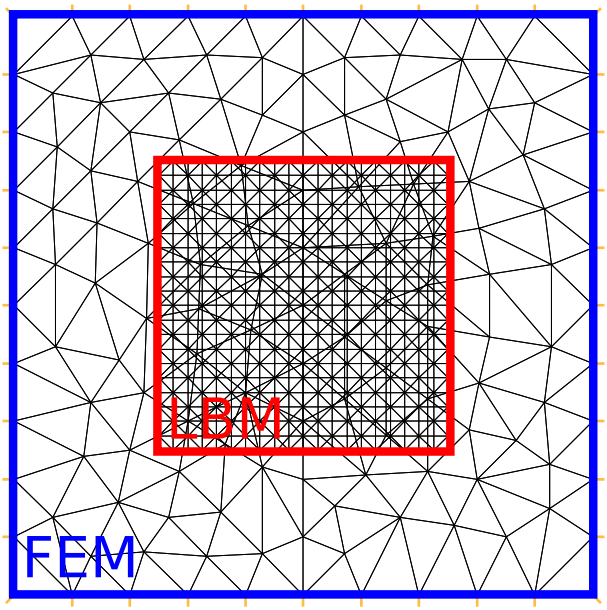


Figure 2: Example of the spatial discretization in our coupling between LBM and FEM.

method takes full advantage of the Parareal algorithm to overcome this problem. Indeed, the FEM problem is considered as a coarse prediction of the solution on the whole space/time domain, whereas the LBM problem serves as a fine description (in space and time) which locally improves the coarse one. The Parareal algorithm predicts first sequentially an approximation of the solution with the coarse FEM solver. Then, a precise computation with the fine LBM solver is carried out in parallel on each coarse time interval, in order to correct iteratively the first inaccurate prediction. In comparison to the previous approaches, this configuration reduces the size of the time interval for LBM computations. Instead of applying the LBM solver sequentially on the total time interval, we solve in parallel a finite number of LBM problems over smaller time intervals, each one of length equal to the coarse time-step.
In this study, following the pioneering papers on FDM/LBM coupling [1, 2], we consider linear diffusion as a prototype to focus on the methodology. This choice is motivated to describe and validate the method in a simple configuration, and to illustrate its properties and potentialities. It results that the LBM will be considered as a macroscopic solver, that approximates the same physics as the FEM (see also [62, 44] where the same kind of choice has been made). For this reason, we will adopt the terminology of FEM-LBM coupling, rather than multiscale or macro-meso coupling, in order to differentiate the macro and meso versions of LBM. Extension to more complex cases such as complex fluids and true multiscale problems, in which a different kind of physics is integrated into LBM, will be object of future works. The outline of the paper is as follows. The model problem, as well as its finite element and lattice Boltzmann approximations are described in Section 2. Section 3 presents the derivation of the spatial coupling between FEM and LBM, and defines the coarse and fine propagators as well as the required transmission operators between the two methods. In Section 4, we describe our version of the Parareal algorithm for coupling in time. Section 5 is dedicated to numerical experiments. Concluding remarks and perspectives are drawn in Section 6.

2 Model problem and setting

Let Ω_c be an open bounded domain of $\mathbb{R}^2$, with a Lipschitz-continuous boundary $\Gamma := \partial\Omega_c$. In the time interval $(0,T)$ $(T>0)$, let us consider the following model problem on Ω_c:
Find $u : \Omega_c \times (0,T) \to \mathbb{R}$ such that:

$$\begin{aligned} \frac{\partial u}{\partial t} - \mu\,\Delta u &= F \qquad \text{in} \quad \Omega_c \times (0,T), \\ u &= g \qquad \text{on} \quad \Gamma \times (0,T), \\ u(\cdot,0) &= u_0 \qquad \text{in} \quad \Omega_c, \end{aligned} \tag{1}$$

where Δ is the Laplace operator, $F : \Omega_c \times (0,T) \to \mathbb{R}$, $g : \Gamma \times (0,T) \to \mathbb{R}$ are given functions, u_0 is the initial condition and $\mu > 0$ is the diffusion coefficient. Note that the problem is formulated here with a Dirichlet

boundary condition, for the sake of simplicity, but other kind of boundary conditions can be taken into account (see §5.4 for such an example).
Let now Ω_{f} be a subset of Ω_{c} (see Figure 3). In the larger domain Ω_{c} the model problem (1) will be solved with FEM using a coarse mesh, while in the smaller domain Ω_{f} with LBM. In the following subsections §2.1 and §2.2, we recall the basics of each numerical method for problem (1).

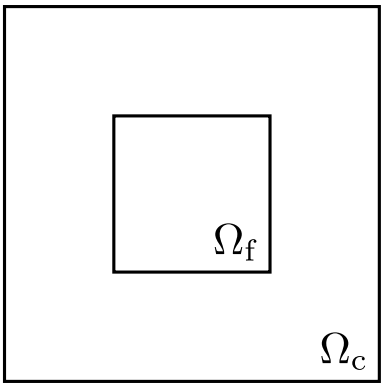


Figure 3: The two domains for FEM and LBM solving.

2.1 Finite element approximation

We first semi-discretize the problem (1) in space using finite elements. We note $\mathcal{T}^H$ a coarse mesh of the domain Ω_{c} ($H = \max_{K\in\mathcal{T}^H} H_K$, with H_K being the diameter of the mesh element K). The mesh is supposed regular, i.e. there exists $\sigma > 0$ such that $\forall K \in \mathcal{T}^H, H_K/\rho_K \leq \sigma$ where ρ_K denotes the radius of the inscribed circle in K. We choose standard continuous and piecewise affine finite element spaces:

$$V^H = \left\{ v_H \in C(\overline{\Omega}_{\mathrm{c}}) : v_H|_K \in \mathbb{P}_1(K), \forall K \in \mathcal{T}^H \right\},$$
$$V_0^H = \left\{ v_H \in V^H : v_H = 0 \text{ on } \Gamma \right\}.$$

The semi-discretized FEM problem reads, for all $t \in (0, T)$:

$$\begin{cases} \text{Find } u_H(t) \in V^H,\ u_H(t) = g_H(t) \text{ on } \Gamma, \text{ such that:} \\ \dfrac{d}{dt}\displaystyle\int_{\Omega_{\mathrm{c}}} u_H(t) v_H + \mu \int_{\Omega_{\mathrm{c}}} \nabla u_H(t) \cdot \nabla v_H = \int_{\Omega_{\mathrm{c}}} F(t) v_H, \qquad \forall\, v_H \in V_0^H, \\ u_H(0) = u_H^0, \end{cases} \tag{2}$$

where u_H^0 (resp. g_H) is a FE approximation (e.g, the interpolant) of u_0 (resp. g) on V^H.
For the complete discretization in space and time, let $\Delta t > 0$ be the time-step for the FEM problem, and consider a uniform discretization of the time interval $(0, T)$: $(t^0, \ldots, t^N)$, with $t^n = n\Delta t$, $n = 0, \ldots, N$. We choose for the sake of simplicity to semi-discretize in time the problem (2) using a backward Euler scheme (however this is not restrictive). We note u_H^{n-1} the resulting discretized solution of (2) at time step t^{n-1}. For $n \geq 1$, the fully discretized FEM problem reads:

$$\begin{cases} \text{Find } u_H^n \in V^H,\ u_H^n = g_H^n \text{ on } \Gamma, \text{ such that:} \\ \displaystyle\int_{\Omega_{\mathrm{c}}} \frac{u_H^n - u_H^{n-1}}{\Delta t} v_H + \mu \int_{\Omega_{\mathrm{c}}} \nabla u_H^n \cdot \nabla v_H = \int_{\Omega_{\mathrm{c}}} F^n v_H, \qquad \forall\, v_H \in V_0^H, \end{cases} \tag{3}$$

where $F^n = F(t^n)$ and $g_H^n = g_H(t^n)$ are values of source term F and boundary term g_H at time t^n.

2.2 Lattice Boltzmann approximation

The lattice Boltzmann formulation of problem (1) is a particular case of the lattice Boltzmann advection-diffusion model [15, 35], that can be retrieved from an appropriate discretization of the Boltzmann-Maxwell equation

$$\left(\partial_t + \mathbf{e} \cdot \boldsymbol{\nabla}_{\mathbf{x}} + \mathbf{F} \cdot \boldsymbol{\nabla}_{\mathbf{e}}\right) f(\mathbf{e}, \mathbf{x}, t) = J(f)(\mathbf{x}, t), \quad \mathbf{x} \in \Omega_{\mathrm{f}}, t > 0. \tag{4}$$

Equation (4) is a mesoscopic conservation equation for the *particle distribution function* $f(\mathbf{e}, \mathbf{x}, t)$, so that $f(\mathbf{e}, \mathbf{x}, t)\mathrm{d}\mathbf{e}\mathrm{d}\mathbf{x}$ is the total mass of particles inside the infinitesimal volume element $\mathrm{d}\mathbf{e}\mathrm{d}\mathbf{x}$ at a fixed time, position and velocity. The quantity $\mathbf{F}$ represents the external force while the term J, also called *collision operator*, takes into account the effects of inter-particle collisions.
In order to derive the lattice Boltzmann approximation of (1) from (4), a procedure similar to that presented in [57] can be adopted. First the velocity space is approximated by projecting the distribution function f onto a Hilbert subspace $\mathcal{H}^N$ spanned by the first N Hermite polynomials, where the order N is dictated by the macroscopic behavior one wants to recover. Then the resulting discrete velocity equation is integrated along the characteristics. An approximation with finite differences leads to the following lattice Boltzmann equation

$$f_i(\mathbf{x} + \mathbf{e}_i\delta x, t + \delta t) - f_i(\mathbf{x}, t) = J_i(f)(\mathbf{x}, t) + \delta t F_i \qquad \forall i = 0, ..., Q, \tag{5}$$

where δx and δt are respectively the LBM space- and time-step, while $E = \{\mathbf{e}_0, ..., \mathbf{e}_Q\}$ denotes the discrete velocity set obtained by the projection onto $\mathcal{H}^N$. Note that the discrete distribution function $f_i(\mathbf{x}, t)$ represents an approximation in space and time of $f(\mathbf{e}, \mathbf{x}, t)$, for a given velocity $\mathbf{e}_i \in E$, i.e. $f_i(\mathbf{x}, t) \equiv f(\mathbf{e}_i, \mathbf{x}, t)$. The discrete transport of the particles is balanced on the right hand side by the discrete forcing term F_i and by $J_i(f)$, a discrete approximation for the collision operator J.
Depending on the choice of the discrete collision operator $J_i(f)$, of the discrete force F_i and of the discrete velocity set E, different physical behaviors can be recovered from (5) [11, 40]. In the specific case of Equation (1), one can show that $J_i(f)$ and F_i take respectively the forms:

$$J_i(f) = J_i^{BGK}(f) = -\frac{1}{\tau}(f_i - f_i^{eq}) \qquad \forall i = 0, ..., Q, \tag{6}$$

$$F_i = w_i F \qquad \forall i = 0, ..., Q, \tag{7}$$

with F being the forcing term given in (1).
The term $J_i^{BGK}(f)$ is an approximation of the well-known single relaxation time Bhatnagar-Gross-Krook (BGK) collision operator [53], τ is the so-called *relaxation time* and f_i^{eq} is an appropriate discrete approximation of the Maxwell-Boltzmann equilibrium distribution [57]. For problem (1) this is given by

$$f_i^{eq} = w_i u_{LB}(\mathbf{x}, t) \qquad \forall i = 0, ..., Q, \tag{8}$$

where u_{LB} is the lattice Boltzmann approximation of the solution of problem (1) on the discrete lattice, while the weights w_i are suitable constants that depend on the choice of the discrete velocity set E.

Remark 2.1. *Equation* (8)*, requires the knowledge of* u_{LB} *to compute the associated* f_i^{eq}*. From a computational point of view, this evaluation is straightforward with the adoption of an explicit time-advancing scheme (see also Equation* (13)*).*

For problem (1) a common choice for E is the five-velocity square lattice structure represented in Figure 4. This structure, also referred as D2Q5 model, is characterized by the following lattice velocities $\mathbf{e}_i$

$$[\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4] = \begin{bmatrix} 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix}, \tag{9}$$

and by the weights w_i:

$$\begin{cases} w_0 = (1 - 2\eta), \\ w_i = 0.5\eta, \qquad \forall i = 0, ..., 4, \end{cases} \tag{10}$$

where $\eta = (0, 0.5]$ is a free positive parameter [36].

Remark 2.2. *The particular case* $\eta = 0.5$ *leads to a model in which the distribution function* f_0 *is neglected* ($w_0 = 0$)*. The resulting velocity structure is therefore sometimes indicated as the D2Q4 model. This model is adopted in the following sections.*

It can be shown that the numerical model defined by Equations (5)-(10) recovers asymptotically problem (1) with the unknown and the viscosity given respectively by

$$u_{LB}(\mathbf{x}, t) = \sum_{i=0}^{4} f_i(\mathbf{x}, t) = \sum_{i=0}^{4} f_i^{eq}(\mathbf{x}, t), \tag{11}$$

and

$$\mu = \eta(\tau - 0.5)\delta x^2/\delta t. \tag{12}$$

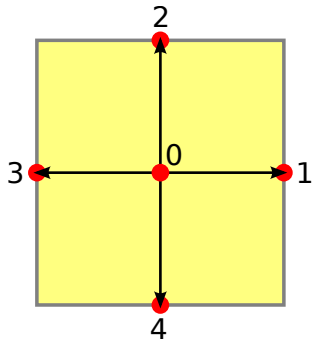


Figure 4: D2Q5 model. Dots represent the lattice nodes $\boldsymbol{x}_i, \forall i = 0, ..., 4$ in the unit cell. The arrows identify the links $\mathbf{e}_i, \forall i = 0, ..., 4$ in the lattice structure, their lengths are given by $\|\boldsymbol{x}_0 - \boldsymbol{x}_i\|, \forall i = 0, ..., 4$.

Remark 2.3. *Property* (11) *is issued from the collision invariant* $\sum_{i=0}^{4} J_i(f) = 0$*, which ensures the conservation of the zero-th order moment in the macroscopic system.*

From a computational point of view, one can think of Equation (5) being splitted into two parts:

$$\text{collision step :} \qquad \tilde{f}_i(\mathbf{x}, t) = f_i(\mathbf{x}, t) + J_i(f)(\mathbf{x}, t) + \delta t F_i \qquad \forall i = 0, ..., Q, \tag{13}$$

$$\text{streaming step :} \qquad f_i(\mathbf{x} + \mathbf{e}_i \delta x, t + \delta t) = \tilde{f}_i(\mathbf{x}, t) \qquad \forall i = 0, ..., Q. \tag{14}$$

The collision step is a local update of the distribution functions on each lattice node, while the streaming step moves the data across the lattice. This set of equations is eventually supplemented with appropriate initial and boundary conditions for the distribution functions, for which multiple formulations exist (see [47, 38, 41] and references therein). The treatment and implementation of complex initial and boundary conditions goes beyond the scope of this work. An accurate description and analysis of various boundary conditions for LBM can be found in [13].

3 Spatial coupling between FEM and LBM

In this section, we detail our methodological approach for the spatial coupling between FEM and LBM. First the continuous coupling conditions are derived. Then the notions of coarse and fine propagators are specified. Finally the transmission (interpolation) operators between the two spatial discretizations are introduced.

3.1 Derivation of the coupling conditions

As observed in [1, 67] the spatial coupling of the FEM and LBM processes relies on the definition of compression ($\mathcal{C}$) and reconstruction ($\mathcal{R}$) operations between the state variables of the two descriptions, respectively u_{LB} and $f_{i|i=0,...,Q}$:

$$u_{LB}(\mathbf{x}, t) = \mathcal{C}(f_{i|i=0,...,Q}(\mathbf{x}, t)), \tag{15}$$

$$f_{i|i=0,...,Q}(\mathbf{x}, t) = \mathcal{R}(u_{LB}(\mathbf{x}, t)). \tag{16}$$

While the compression operation is unique and defined by the local-ensemble average (11), the reconstruction operation is not. Note in fact that the variables $f_{i|i=0,...,Q}$ contain more information than the variable u_{LB}. Thus the reconstruction procedure leads to a one-to-many mapping.
The derivation of an analytic expression for (16) can be obtained, for instance, with a Chapman-Enskog expansion [1, 67]. We provide below an alternative derivation for the case of the D2Q4 lattice model, with zero external forces, that is based on asymptotic analysis, and inspired from [37, 11, 12]. For other models, the derivation is similar.

Theorem 3.1. *Let us consider the following lattice Bolzmann approximation of problem* (1) *(with* $F = 0$*):*

$$f_i(\mathbf{x} + \mathbf{e}_i \delta x, t + \delta t) - f_i(\mathbf{x}, t) = -\frac{1}{\tau}(f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)) \qquad \forall i = 1, ..., 4, \tag{17}$$

with $u_{LB}(\mathbf{x}, t) = \sum_{i=1}^{4} f_i(\mathbf{x}, t) = \sum_{i=1}^{4} f_i^{eq}(\mathbf{x}, t)$ *and* $f_i^{eq}(\mathbf{x}, t) = \frac{1}{4} u_{LB}(\mathbf{x}, t)$*. The associated zero-th order and first-order accurate reconstruction operators are respectively given by:*

$$f_{i|i=1,...,4}(\mathbf{x}, t) = \frac{1}{4} u_{LB}(\mathbf{x}, t), \tag{18}$$

and by

$$f_{i|i=1,\dots,4}(\mathbf{x},t) = \frac{1}{4}\left(u_{LB} - \tau\delta x\left(\mathbf{e}_i\cdot\boldsymbol{\nabla}\right)u_{LB}\right)(\mathbf{x},t). \tag{19}$$

Proof. Let ε be a small number, and set $\delta x = \varepsilon$, $\delta t = \varepsilon^2$, so that the diffusive scaling $\delta t = \delta x^2$ holds. We Taylor-expand the distribution functions f_i, f_i^{eq} $(i = 1, ..., 4)$ and the lattice Boltzmann solution u_{LB} as follows:

$$\begin{aligned} f_i(\mathbf{x},t) &= f_i^{[0]}(\mathbf{x},t) + \varepsilon f_i^{[1]}(\mathbf{x},t) + \varepsilon^2 f_i^{[2]}(\mathbf{x},t) + \mathcal{O}(\varepsilon^3), &(20)\\ f_i^{eq}(\mathbf{x},t) &= f_i^{eq,[0]}(\mathbf{x},t) + \varepsilon f_i^{eq,[1]}(\mathbf{x},t) + \varepsilon^2 f_i^{eq,[2]}(\mathbf{x},t) + \mathcal{O}(\varepsilon^3), &(21)\\ u_{LB}(\mathbf{x},t) &= u_{LB}^{[0]}(\mathbf{x},t) + \varepsilon u_{LB}^{[1]}(\mathbf{x},t) + \varepsilon^2 u_{LB}^{[2]}(\mathbf{x},t) + \mathcal{O}(\varepsilon^3). &(22)\end{aligned}$$

Next, we Taylor-expand the equation (17) at order 2:

$$\begin{aligned} & f_i(\mathbf{x},t)\\ &= f_i^{eq}(\mathbf{x},t) - \tau(f_i(\mathbf{x}+\mathbf{e}_i\varepsilon, t+\varepsilon^2) - f_i(\mathbf{x},t))\\ &= f_i^{eq}(\mathbf{x},t) - \tau\left(\varepsilon(\mathbf{e}_i\cdot\boldsymbol{\nabla})f_i(\mathbf{x},t) + \varepsilon^2\left(\frac{\partial}{\partial t} + \frac{1}{2}(\mathbf{e}_i\cdot\boldsymbol{\nabla})^2\right)f_i(\mathbf{x},t)\right) + \mathcal{O}(\varepsilon^3). \end{aligned} \tag{23}$$

We now omit the notation $(\mathbf{x}, t)$ to alleviate the formulas. We plug (20) and (21) into (23) and we get, after equating the terms at each order in ε:

$$\begin{aligned} f_i^{[0]} &= f_i^{eq,[0]} && \text{(order 0)} &(24)\\ f_i^{[1]} &= f_i^{eq,[1]} - \tau(\mathbf{e}_i\cdot\boldsymbol{\nabla})f_i^{[0]} && \text{(order 1)} &(25)\\ f_i^{[2]} &= f_i^{eq,[2]} - \tau\left((\mathbf{e}_i\cdot\boldsymbol{\nabla})f_i^{[1]} + \left(\frac{\partial}{\partial t} + \frac{1}{2}(\mathbf{e}_i\cdot\boldsymbol{\nabla})^2\right)f_i^{[0]}\right) && \text{(order 2)} &(26)\end{aligned}$$

At equilibrium, the distribution functions and the lattice Boltzmann solution are such that $f_i^{eq} = \frac{1}{4}u_{LB}$. This holds at each order in ε, so that (21) and (22) yield $f_i^{eq,[k]} = \frac{1}{4}u_{LB}^{[k]}$, $i = 1, \dots, 4$, $k = 0, 1, 2$. Therefore, we obtain respectively from (24) and (25):

$$\begin{aligned} f_i^{[0]} &= \frac{1}{4}u_{LB}^{[0]}, &(27)\\ f_i^{[1]} &= \frac{1}{4}\left(u_{LB}^{[1]} - \tau(\mathbf{e}_i\cdot\boldsymbol{\nabla})u_{LB}^{[0]}\right). &(28)\end{aligned}$$

Finally, we reconstruct each f_i, $i = 1, \dots, 4$, from $f_i = f_i^{[0]} + \varepsilon f_i^{[1]} + \mathcal{O}(\varepsilon^2)$, and then inserting (27), (28) and the approximation $u_{LB} = u_{LB}^{[0]} + \varepsilon u_{LB}^{[1]} + \mathcal{O}(\varepsilon^2)$:

$$\begin{aligned} f_i &= f_i^{[0]} + \varepsilon f_i^{[1]} + \mathcal{O}(\varepsilon^2)\\ &= \frac{1}{4}u_{LB}^{[0]} + \frac{\varepsilon}{4}\left(u_{LB}^{[1]} - \tau(\mathbf{e}_i\cdot\boldsymbol{\nabla})u_{LB}^{[0]}\right) + \mathcal{O}(\varepsilon^2)\\ &= \frac{1}{4}\left(\underbrace{u_{LB}^{[0]} + \varepsilon u_{LB}^{[1]}}_{u_{LB}+\mathcal{O}(\varepsilon^2)} - \varepsilon\tau(\mathbf{e}_i\cdot\boldsymbol{\nabla})\underbrace{u_{LB}^{[0]}}_{u_{LB}+\mathcal{O}(\varepsilon)}\right) + \mathcal{O}(\varepsilon^2)\\ &= \frac{1}{4}\left(u_{LB} - \varepsilon\tau(\mathbf{e}_i\cdot\boldsymbol{\nabla})u_{LB}\right) + \mathcal{O}(\varepsilon^2). \end{aligned}$$

Remembering that $\delta x = \varepsilon$, this proves the reconstruction formulas (18) and (19).

□

Remark 3.2. *Using the asymptotic expansions derived in the above proof, it can also be shown that (see A):*

$$\frac{\partial u_{LB}}{\partial t} - \frac{1}{2}(\tau - \frac{1}{2})\Delta u_{LB} = \mathcal{O}(\varepsilon).$$

So the lattice Boltzmann solution is consistent at (least at) first order in ε with the heat equation (1) *provided that $\mu = \frac{1}{2}(\tau - \frac{1}{2})$ and $\delta t = \delta x^2$ (see also* (12)*).*

3.2 Coarse and fine propagators

For evolution problems, it is useful to reinterpret our solvers as propagators, i.e. semigroup operators that compute the solution at time $t + \Delta t$ given an initial condition at time t.
For the FEM, we define a coarse propagator $\mathcal{G}_{\Delta t}(t)$ from time t to time $t + \Delta t$ as follows:

$$\mathcal{G}_{\Delta t}(t) : \left\{ \begin{array}{lcl} V^H & \to & V^H \\ u_H(t) & \mapsto & u_H(t + \Delta t), \end{array} \right. \tag{29}$$

where $u_H(t + \Delta t)$ is the solution of (2) on time interval $(t, t + \Delta t)$ with $u_H(t)$ as initial condition.
For the LBM we define also a fine propagator $\mathcal{F}_{\Delta t}(t)$ from time t to time $t + \Delta t$:

$$\mathcal{F}_{\Delta t}(t) : \left\{ \begin{array}{lcl} V^h \times \left(W^h\right)^2 & \to & V^h \\ (u_h(t), (w_h, \nabla w_h)|_{\partial\Omega_{\rm f}}(t), (w_h, \nabla w_h)|_{\partial\Omega_{\rm f}}(t + \Delta t)) & \mapsto & u_h(t + \Delta t). \end{array} \right. \tag{30}$$

Here V^h is a finite element space of piecewise linear continuous functions

$$V^h = \left\{ v_h \in C(\overline{\Omega}_{\rm f}) : v_h|_K \in \mathbb{P}_1(K), \forall K \in \mathcal{T}^h \right\},$$

built on a mesh $\mathcal{T}^h$ whose nodes are those of the lattice Boltzmann grid. This allows to introduce a discrete fine solution $u_h(t) \in V^h$ which is directly obtained from $u_{LB}(t)$ through linear interpolation (note that the Lattice Boltzmann solution $u_{LB}(t)$ is only defined at the nodes of the lattice Boltzmann grid). Therefore, the output $u_h(t + \Delta t)$ of this fine propagator is obtained from the LBM solver described in §2.2.
The space W^h is the discrete trace space on the boundary $\partial\Omega_{\rm f}$ of functions in V^h and their (piecewise constant) gradients :

$$W^h = \left\{ (v_h, \nabla v_h)|_{\partial\Omega_{\rm f}} : v_h \in V^h \right\}.$$

The quantities $(w_h, \nabla w_h)|_{\partial\Omega_{\rm f}}(t)$ and $(w_h, \nabla w_h)|_{\partial\Omega_{\rm f}}(t + \Delta t)$ are (known) boundary conditions that are imposed on $\partial\Omega_{\rm f}$.
The fine propagator $\mathcal{F}_{\Delta t}(t)$ results from the composition of three operators issued from equations (18) (or (19)), (5) and (11). For the sake of clarity, the algorithm associated to (30) is given in Figure 5.
In Figure 5, the parameter α ($\alpha = 0, 1$) allows to consider the two possible orders for the reconstruction operation. Remember that for the LBM solver, the choice of the time-step is restricted by a CFL-type condition, while this is not necessarily the case for the FEM solver, so we need to allow different time scales between FEM and LBM. For this reason, we introduced the parameter p which quantifies the ratio between the coarse time-step Δt and the small time-step δt associated to LBM. This results in a loop in order to carry out p LBM computations during the fine propagation.
One critical point concerns the boundary conditions on $\partial\Omega_{\rm f}$, which should be transmitted by the FEM coarse propagator. To this purpose, we first interpolate linearly in time between the boundary conditions $(w_h, \nabla w_h)(t)$ and $(w_h, \nabla w_h)(t + \Delta t)$, for better accuracy. From this interpolation $(w_\partial^j, \nabla w_\partial^j)$ of the physical quantity, the reconstruction operation is carried out once again in order to obtain the distribution functions f_i on the boundary.

3.3 Transmission operators

Since the two spatial discretizations (for FEM and LBM) are different, two transmission operators are needed in order to transfer the discrete solution from one mesh to the other. We denote by $\Pi_{\rm F}^{\rm L}$ the operator that transfers the FEM solution to the lattice Boltzmann space. Reciprocally, the operator $\Pi_{\rm L}^{\rm F}$ transfers the LBM solution to the finite element space. In practice, $\Pi_{\rm F}^{\rm L}$ (resp. $\Pi_{\rm L}^{\rm F}$) is simply a linear interpolation operator from V^H to V^h (resp. from V^h to V^H) [33]. Due to the inclusion of the domains ($\Omega_f \subset \Omega_c$), the interpolation $\Pi_{\rm L}^{\rm F} v \in V^H$ of a function $v \in V^h$ is in fact the interpolation of the zero-extension of v on Ω_c.
As a result, with the choice of linear interpolation, the transmission operation between meshes involves only a matrix multiplication. Moreover a direct calculation shows that the coefficients the matrix $\Pi_{\rm F}^{\rm L}$ can be expressed as follows

$$\Pi_{{\rm F}\,ij}^{\rm L} = \varphi_j(\mathbf{x}_i),$$

where $\mathbf{x}_i$ is the node number i of the mesh $\mathcal{T}^h$ and φ_j is the shape function associated to the node of number j of V^H (an identical formula can be derived for $\Pi_{\rm L}^{\rm F}$). Algorithms for construction of such rectangular matrices are now implemented in numerous finite element librairies (see e.g. the function `interpolate` in FreeFEM++ [33]).

Fine propagator $\mathcal{F}_{\Delta t}(t)$:
Input : $u_h(t), (w_h, \nabla w_h)|_{\partial\Omega_\mathrm{f}}(t), (w_h, \nabla w_h)|_{\partial\Omega_\mathrm{f}}(t+\Delta t)$.

1. *Initialization:* $u_h^0 = u_h(t)$.
2. *From physical quantities to distribution functions:*
   - Reconstruction operation
     $f_{i|i=1,\dots,4}(\mathbf{x},t) = \frac{1}{4}(u_h^0(\mathbf{x}) - \alpha\,\tau\delta x(\mathbf{e}_i\cdot\boldsymbol{\nabla})u_h^0(\mathbf{x}))$,
     for grid points $\mathbf{x}\in\Omega_\mathrm{f}$.
3. *Loop on small time steps* $\delta t = \Delta t/p$:
   $j = 1,\dots,p;\quad t^j = t + j\delta t$.
   (a) *Linear interpolation in time of boundary conditions:*
   $$w_\partial^j = (1-\frac{j}{p})w_h|_{\partial\Omega_\mathrm{f}}(t) + \frac{j}{p}w_h|_{\partial\Omega_\mathrm{f}}(t+\Delta t),$$
   $$\nabla w_\partial^j = (1-\frac{j}{p})\nabla w_h|_{\partial\Omega_\mathrm{f}}(t) + \frac{j}{p}\nabla w_h|_{\partial\Omega_\mathrm{f}}(t+\Delta t).$$
   (b) *Reconstruction of distribution functions on the boundary:*
   - Reconstruction operation
     $f_{i|i=1,\dots,4}(\mathbf{x},t^j) = \frac{1}{4}(w_\partial^j(\mathbf{x}) - \alpha\,\tau\delta x(\mathbf{e}_i\cdot\boldsymbol{\nabla})w_\partial^j(\mathbf{x}))$,
     for grid points $\mathbf{x}\in\partial\Omega_\mathrm{f}$.
   (c) *Evolution of the distribution functions according to:*
   - Collision step:
     $\tilde{f}_i(\mathbf{x},t^j) = f_i(\mathbf{x},t^{j-1}) + J_i(f)(\mathbf{x},t^{j-1}) + \delta t F_i$,
   - Streaming step:
     $f_i(\mathbf{x}+\mathbf{e}_i\delta x, t^j) = \tilde{f}_i(\mathbf{x},t^j)$,
     for grid points $\mathbf{x}\in\overline{\Omega}_\mathrm{f}$ and for $i = 1,...,4$.
4. *End loop.*
5. *Back from distribution functions to physical quantities:*
   - Compression: $u_h(\mathbf{x},t+\Delta t) = \sum_{i=1}^4 f_i(\mathbf{x},t^p)$.

Figure 5: The algorithm for $\mathcal{F}_{\Delta t}(t)$ in pseudo-code.

4 Time coupling between FEM and LBM using Parareal

In this part, we first explain the ideas of our coupling strategy, and then describe the complete algorithm of coupling between FEM and LBM with Parareal. This section is ended with some considerations on speed-up and system efficiency.

4.1 Basics of the Parareal approach

Here we recall the main ideas of the Parareal method for a simple model. It may be useful for the reader who is unfamiliar to this topic, and aims to make easier the understanding of our Parareal method for FEM-LBM coupling. For more details, we refer to, e.g., [43, 7, 23, 45, 29].
Suppose that we want to solve the following differential system:
Find $\boldsymbol{q} : [0,T] \to \mathbb{R}^m$ such that

$$\begin{aligned} \frac{d\boldsymbol{q}}{dt} &= \boldsymbol{A}(t,\boldsymbol{q}) \quad \text{in } (0,T), \\ \boldsymbol{q}(0) &= \boldsymbol{q}_0, \end{aligned} \tag{31}$$

where $m \geq 1$, for given $T > 0$ and $\boldsymbol{q}_0 \in \mathbb{R}^m$. First, using an inaccurate but cheap method, such as backward Euler on large time-steps Δt, we build a coarse propagator $\mathcal{G}_{\Delta t}$ which serves to predict a sketch of the time-evolution of the quantity $\boldsymbol{q}$:

$$\boldsymbol{q}^{n+1,0} = \mathcal{G}_{\Delta t}(t^n)\boldsymbol{q}^{n,0},\ n = 0,\dots,N\,;\boldsymbol{q}^{0,0} = \boldsymbol{q}_0,$$

where $t^n = n\Delta t$, $N = T/\Delta t$ and $\boldsymbol{q}^{n,0} \simeq \boldsymbol{q}(t^n)$. This can serves to predict on each time-slab $(t^n; t^{n+1})$ an accurate solution, with for instance a high-order time-integration method, or a simple time-integration method with a smaller time-step $\delta t \ll \Delta t$. We introduce for this purpose a fine propagator $\mathcal{F}_{\Delta t}$:

$$\hat{\boldsymbol{q}}^{n+1,0} = \mathcal{F}_{\Delta t}(t^n)\boldsymbol{q}^{n,0},\ n = 0, \ldots, N\,; \hat{\boldsymbol{q}}^{0,0} = \boldsymbol{q}_0.$$

Observe that, since all the sequence $(\boldsymbol{q}^{n,0})_n$ is known, this step can be fully parallelized, and each (expensive) fine propagation can be effectuated by one processor in a parallel architecture.
However due to the difference of accuracy between $\mathcal{F}_{\Delta t}$ and $\mathcal{G}_{\Delta t}$ we expect that $\hat{\boldsymbol{q}}^{n,0} \neq \boldsymbol{q}^{n,0}$ for $n = 1, \ldots, N$, i.e., there are jumps at each time-step t^n between coarse and fine predictions of $\boldsymbol{q}(t^n)$. These jumps can be, hopefully, attenuated using a prediction-correction (or defect-correction [60]) step:

$$\boldsymbol{q}^{n+1,1} = \mathcal{G}_{\Delta t}(t^n)\boldsymbol{q}^{n,1} + \mathcal{F}_{\Delta t}(t^n)\boldsymbol{q}^{n,0} - \mathcal{G}_{\Delta t}(t^n)\boldsymbol{q}^{n,0}$$

for $n = 0, \ldots, N$ and still with $\boldsymbol{q}^{0,1} = \boldsymbol{q}_0$. In the above formula, the new coarse prediction $\mathcal{G}_{\Delta t}(t^n)\boldsymbol{q}^{n,1}$ is corrected by the difference between the fine and coarse predictions already computed. As a result, we obtain a new sequence $(\boldsymbol{q}^{n,1})_n$ which should approximate more accurately the solution of (31).
Then we repeat the previous steps to obtain sequences of solutions $(\boldsymbol{q}^{n,k})_n$, $(\boldsymbol{q}^{n,k+1})_n$, ... that should converge to a solution $(\boldsymbol{q}^{n,\infty})_n$ which verifies:

$$\boldsymbol{q}^{n+1,\infty} = \mathcal{G}_{\Delta t}(t^n)\boldsymbol{q}^{n,\infty} + \mathcal{F}_{\Delta t}(t^n)\boldsymbol{q}^{n,\infty} - \mathcal{G}_{\Delta t}(t^n)\boldsymbol{q}^{n,\infty} = \mathcal{F}_{\Delta t}(t^n)\boldsymbol{q}^{n,\infty}.$$

So $(\boldsymbol{q}^{n,\infty})_n$ is the solution obtained through (expensive) sequential computations with the accurate numerical method. Note that the parallel acceleration is effective provided that $K \ll N$ where K is the number of Parareal iterations necessary for convergence, and provided that the coarse computation is much cheaper than the fine one (this case occurs for instance if $\delta t \ll \Delta t$).
To give an illustration, we display Figure 6 different Parareal solutions after each iteration k for the simple differential equation:

$$\frac{dq}{dt} + \alpha(t)q(t) = \beta(t) \quad \text{in } (0,T), \quad q(0) = 1,$$

where $\alpha(t) = 1 + \sin(5t)$, $\beta(t) = \frac{1}{2}(3 + \sin(t))$ (here $m = 1$ and $\boldsymbol{A}(t, \boldsymbol{q}) = \beta(t) - \alpha(t)\boldsymbol{q}$). We chose backward Euler scheme for each propagator $\mathcal{G}_{\Delta t}$ and $\mathcal{F}_{\Delta t}$, and smaller time-steps for the fine propagator $\mathcal{F}_{\Delta t}$. Parameters are then: $T = 10$, $\Delta t = 1$, $\delta t = 0.01$ (so $N = \frac{T}{\Delta t} = 10$ and $p = \frac{\Delta t}{\delta t} = 100$).
After the first coarse and fine propagations ($k = 0$), there are huge jumps between coarse time slabs, that are reduced progressively when k is increased, since prediction-correction iterations are made. After three Parareal iterations ($k = 3$), the jumps are no more visible and the quality of fine solution is almost recovered on the whole time-interval $(0, 10)$.

4.2 Coupling in time between coarse and fine propagators

Let us now adapt the Parareal methodology described previously in §4.1 for coupling FEM and LBM. To solve Problem (1), we first compute a coarse FEM approximation of the solution u on the whole time interval $(0, T)$. We keep the notations already introduced in §2.1. From the initial condition u_H^0 at time $t^0 = 0$, we iterate as follows:

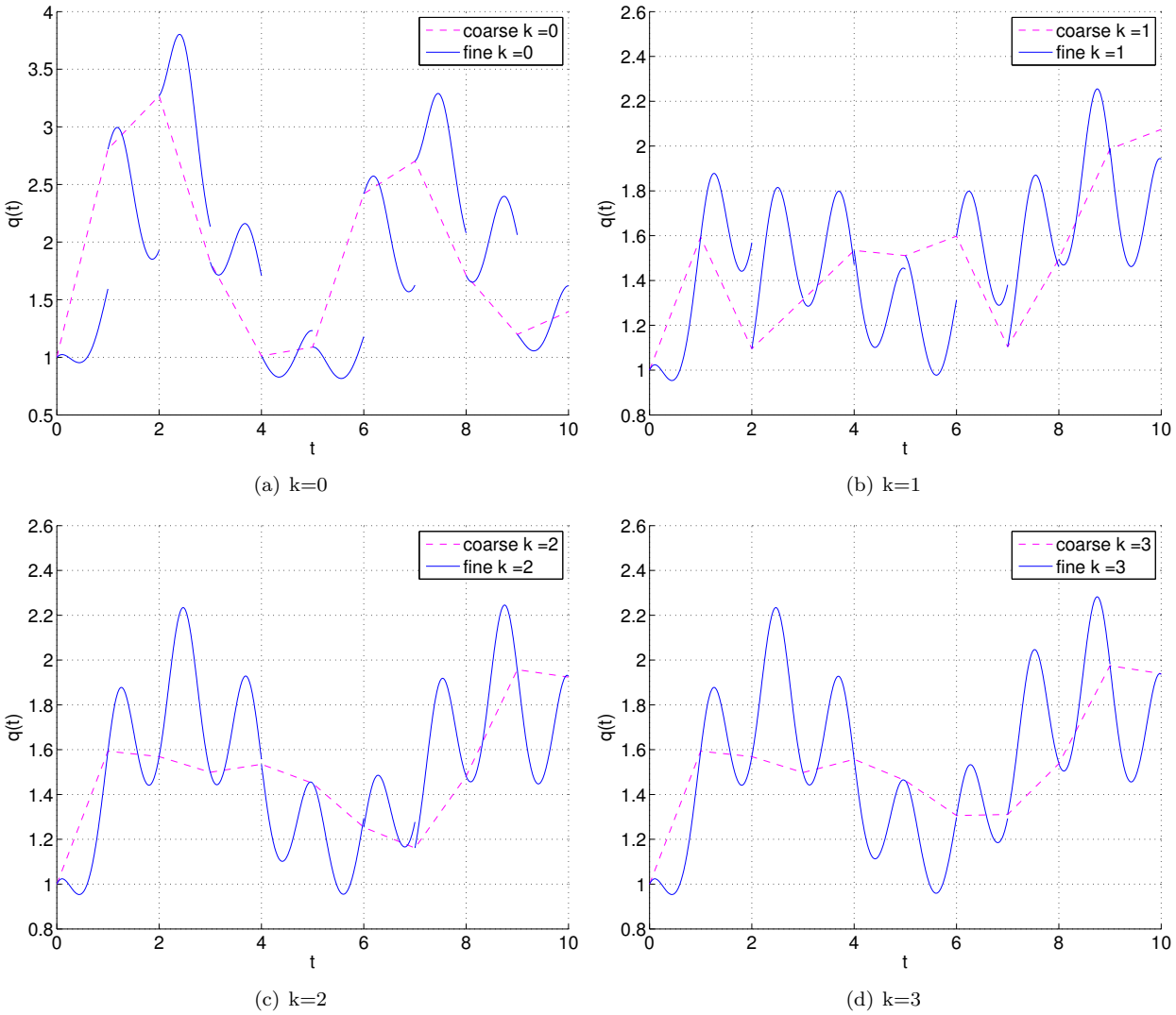
$$u_H^{n+1,0} := \mathcal{G}_{\Delta t}(t^n)(u_H^{n,0}), \quad n = 0, \ldots, N-1,$$

where $u_H^{n,0}$ is a first coarse approximation of $u(t^n)$. Then, from the sequel $(u_H^{n,0})_n$ of these coarse approximations, a fine computation on each time interval (t^n, t^{n+1}) with LBM is done:

$$u_h^{n+1,0} := \mathcal{F}_{\Delta t}(t^n)(\Pi_{\mathrm{F}}^{\mathrm{L}} u_H^{n,0}, [u_H^{n,0}]_{\partial\Omega_{\mathrm{f}}}, [u_H^{n+1,0}]_{\partial\Omega_{\mathrm{f}}}), \quad n = 0, \ldots, N-1,$$

where we use the abbreviated notation $[w_H]_{\partial\Omega_{\mathrm{f}}} := (\Pi_{\mathrm{F}}^{\mathrm{L}} w_H, \nabla\Pi_{\mathrm{F}}^{\mathrm{L}} w_H)|_{\partial\Omega_{\mathrm{f}}}$ for any $w_H \in V^H$.
One first fundamental observation is that these fine LBM computations allow to correct the first coarse prediction, in order to obtain a better approximation of $u(t^n)$. Indeed, the initial value which serves for each coarse computation with $\mathcal{G}_{\Delta t}$ can be updated taking into account the LBM solution on the small domain Ω_{f}. This requires a prediction-correction procedure (see §4.1):

$$u_h^{n+1,1} := \underbrace{\Pi_{\mathrm{F}}^{\mathrm{L}} \mathcal{G}_{\Delta t}(t^n)\left(\chi(\Pi_{\mathrm{L}}^{\mathrm{F}} u_h^{n,0}) + (1-\chi)u_H^{n,1}\right)}_{\text{Prediction}} + \underbrace{u_h^{n+1,0} - \Pi_{\mathrm{F}}^{\mathrm{L}} u_H^{n+1,0}}_{\text{Correction}},$$



(a) k=0 (b) k=1

(c) k=2 (d) k=3

Figure 6: Coarse and fine solutions for iterations $k = 0, \ldots, 3$ of the Parareal method.

for $n = 0, \ldots, N-1$. In this step, the new coarse prediction is corrected by the difference between the previous fine prediction $u_h^{n+1,0}$ and the previous coarse prediction $u_H^{n+1,0}$. The new coarse prediction takes into account the previous fine computation $u_h^{n,0}$ as initial condition at each time t^n. The characteristic function of the LBM domain Ω_{f} is denoted by the symbol χ, and is introduced due to the relationship $\Omega_{\mathrm{f}} \subsetneq \Omega_{\mathrm{c}}$. Outside of the sub-domain Ω_{f} a new coarse solution $u_H^{n,1}$ at time t^n is used. Note that in this step the transmission operators $\Pi_{\mathrm{L}}^{\mathrm{F}}$ and $\Pi_{\mathrm{F}}^{\mathrm{L}}$ (see §3.3) are also needed due to the difference of discretization between FEM and LBM.

The second fundamental observation is that, as in the Parareal algorithm, the fine computations with LBM on each time interval (t^n, t^{n+1}) can be parallelized. As a result, it is possible to define a Parareal-type algorithm for efficient coupling in time of FEM and LBM, that produces at each iteration k a sequence of solutions $(u_h^{n,k})_n$. Increasing k enhances the coupling degree between the two methods.

4.3 An adaptation of the Parareal algorithm for FEM-LBM coupling

Suppose that, at iteration k, we dispose of a sequence of couples of coarse and fine solutions on the whole time interval: $(u_H^{n,k}, u_h^{n,k})_n$. The key ingredient of our Parareal FEM-LBM coupling procedure is the following

prediction-correction iteration:

$$
\begin{aligned}
u_h^{n+1,k+1} := & \underbrace{\Pi_{\mathrm{F}}^{\mathrm{L}} \mathcal{G}_{\Delta t}(t^n)\left(\tilde{u}_{h/H}^{n,k+1}\right)}_{\text{Prediction}} \\
& + \underbrace{\mathcal{F}_{\Delta t}(t^n)(u_h^{n,k}, [u_H^{n,k}]_{\partial\Omega_{\mathrm{f}}}, [u_H^{n+1,k}]_{\partial\Omega_{\mathrm{f}}}) - \Pi_{\mathrm{F}}^{\mathrm{L}} \mathcal{G}_{\Delta t}(t^n)\left(\tilde{u}_{h/H}^{n,k}\right)}_{\text{Correction}},
\end{aligned}
\tag{32}
$$

for $0 \leq n \leq N-1$. In the above formula we introduced the following abbreviated notation for the composed FEM-LBM solution on the whole domain Ω_{c}:

$$
\tilde{u}_{h/H}^{n,k} := \chi(\Pi_{\mathrm{L}}^{\mathrm{F}} u_h^{n,k-1}) + (1-\chi) u_H^{n,k}.
$$

The complete Parareal FEM-LBM algorithm in pseudo-code is given Figure 7.

Parareal FEM-LBM coupling:

1. *Initialization step (–):*
   - First coarse prediction:
   $$
   \begin{aligned}
   u_H^{0,0} = u_H^0 \,;\, u_H^{n+1,0} &:= \mathcal{G}_{\Delta t}(t^n)(u_H^{n,0}), \quad n = 0, \ldots, N-1, \\
   \tilde{u}_{h/H}^{n,0} &:= u_H^{n,0}, \qquad\qquad n = 0, \ldots, N, \\
   u_h^{n,0} &:= \Pi_{\mathrm{F}}^{\mathrm{L}} u_H^{n,0}, \qquad\qquad n = 0, \ldots, N.
   \end{aligned}
   $$
   - Init the index of the first time interval for solving: $n_0 = 0$.
   - Init the index of the Parareal iteration: $k = 0$.
2. *While* $(n_0 < N)$ *:*
   (a) *Fine computation (//):*
   $$
   \hat{u}_h^{n+1,k} := \mathcal{F}_{\Delta t}(t^n)(u_h^{n,k}, [u_H^{n,k}]_{\partial\Omega_{\mathrm{f}}}, [u_H^{n+1,k}]_{\partial\Omega_{\mathrm{f}}}),\ n = n_0, \ldots, N-1.
   $$
   (b) *Error evaluation (//):*
   $$
   \varepsilon_n^k = \|\hat{u}_h^{n+1,k} - u_h^{n+1,k}\| / \|u_h^{n+1,k}\|,\ n = n_0, \ldots, N-1.
   $$
   (c) *Update* n_0 *:* $n^* \to n_0$ with n^* computed as:
   $$
   n^* = \min\{n \mid n_0 \leq n < N; \varepsilon_n^k > \varepsilon^*\}.
   $$
   (d) *Prediction-correction (–):*
   compute the new values $(u_H^{n+1,k+1}, u_h^{n+1,k+1})_n$, $n = n_0, \ldots, N-1$.
   $$
   \begin{aligned}
   \tilde{u}_{h/H}^{n,k+1} &:= \chi(\Pi_{\mathrm{L}}^{\mathrm{F}} u_h^{n,k}) + (1-\chi) u_H^{n,k+1}, \\
   u_H^{n+1,k+1} &:= \mathcal{G}_{\Delta t}(t^n)(\tilde{u}_{h/H}^{n,k+1}), \\
   u_h^{n+1,k+1} &:= \Pi_{\mathrm{F}}^{\mathrm{L}}(u_H^{n+1,k+1} - u_H^{n+1,k}) + \hat{u}_h^{n+1,k}.
   \end{aligned}
   $$
   (e) Update the index of the Parareal iteration: $k \leftarrow k+1$.
3. *End while.*

Figure 7: The Parareal algorithm for FEM-LBM coupling in pseudo-code. The symbol (//) means that the step can be processed in parallel, while the symbol (–) means that the step needs to be processed sequentially.

Once the initial values have been computed with the coarse propagator $\mathcal{G}_{\Delta T}$, the main loop starts until the coupled FEM-LBM solution is approximated with a sufficient accuracy ε^*. During each step of the main loop, the fine computations are first effectuated in parallel. Then, the prediction-correction formula (32) is applied sequentially.

At each step, the fine solution is recovered with required accuracy on the first $0, \ldots, n_0 - 1$ intervals, with $n_0 \geq 1$. As a result, and as in the version presented in [23], only the $n_0, \ldots, N-1$ intervals in which convergence has not been reached are integrated into the computations. The algorithm stops when convergence has been attained for all the coarse time intervals.

Remark 4.1. *At convergence, i.e., when $k \to +\infty$, we obtain the following system at the limit:*

$$\begin{aligned}
\hat{u}_h^{n+1,\infty} &:= \mathcal{F}_{\Delta t}(t^n)(u_h^{n,\infty}, [u_H^{n,\infty}]_{\partial\Omega_{\mathrm{f}}}, [u_H^{n+1,\infty}]_{\partial\Omega_{\mathrm{f}}}),\\
\tilde{u}_{h/H}^{n,\infty} &:= \chi(\Pi_{\mathrm{L}}^{\mathrm{F}} u_h^{n,\infty}) + (1-\chi) u_H^{n,\infty},\\
u_H^{n+1,\infty} &:= \mathcal{G}_{\Delta t}(t^n)(\tilde{u}_{h/H}^{n,\infty}),\\
u_h^{n+1,\infty} &:= \Pi_{\mathrm{F}}^{\mathrm{L}}(u_H^{n+1,\infty} - u_H^{n+1,\infty}) + \hat{u}_h^{n+1,\infty}.
\end{aligned}$$

This simplifies into:

$$\begin{aligned}
u_h^{n+1,\infty} &:= \mathcal{F}_{\Delta t}(t^n)(u_h^{n,\infty}, [u_H^{n,\infty}]_{\partial\Omega_{\mathrm{f}}}, [u_H^{n+1,\infty}]_{\partial\Omega_{\mathrm{f}}}),\\
u_H^{n+1,\infty} &:= \mathcal{G}_{\Delta t}(t^n)(\chi(\Pi_{\mathrm{L}}^{\mathrm{F}} u_h^{n,\infty}) + (1-\chi) u_H^{n,\infty}).
\end{aligned}$$

Note that our formulation differs from the standard Parareal strategy in that the fine solver needs the coarse solver for boundary conditions, whence a reference solution can not be obtained by running the fine solver sequentially. It results that our solution can indeed be regarded as a FEM-LBM coupled solution, with a fine LBM prediction on the required sub-region.

Remark 4.2. *As the algorithm 7 involves two distinct spatial discretizations and associated transmission operators, it follows the work originally proposed in [26] where a Parareal method with two spatial grids is applied to solve the Navier-Stokes equations.*

Remark 4.3. *When the evolution equation is non-linear the last equation of step 2 (d) in Figure 7 should be rewritten as*

$$u_h^{n+1,k+1} := \Pi_{\mathrm{F}}^{\mathrm{L}} u_H^{n+1,k+1} + \hat{u}_h^{n+1,k} - \Pi_{\mathrm{F}}^{\mathrm{L}} u_H^{n+1,k}.$$

In Figure 7 we took advantage of the linearity of our model problem (1) *to factorize the transmission operator, which involves less computations.*

4.4 Speed-up due to parallelization

This section is dedicated to the estimation of the speed-up, which quantifies the gain provided by parallelization. The speed-up is defined as the ratio between the computational cost of a sequential FEM-LBM coupling and that of parallelized FEM-LBM coupling (the computational cost being the total number of elementary operations carried out when the whole algorithm is run one time). In the following analysis, we make the assumptions below:

1. We neglect the time to communicate data. This is a frequent assumption in the literature (see, e.g., [23, 51]). It is valid for instance when the time of a fine computation is far above communication time.
2. We neglect the computational cost associated to the transmission operators $\Pi_{\mathrm{F}}^{\mathrm{L}}$ and $\Pi_{\mathrm{L}}^{\mathrm{F}}$.

Let us first use the notation $\mathrm{C}_{\mathrm{FEM}}$ for the cost of one coarse computation $\mathcal{G}_{\Delta t}$ with FEM, on an interval of length Δt. This is supposed to be a known constant, which of course depends on the number of degrees of freedom induced by the FEM discretization (mesh size and finite element type) and on the type of the time-marching scheme that is chosen for one coarse propagation.
Since the structure of the fine propagator $\mathcal{F}_{\Delta t}$ is more complex (see §3.2 and Figure 5), we detail the cost of one fine computation, that we note $\mathrm{C}_{\mathrm{LBM}}$:

$$\mathrm{C}_{\mathrm{LBM}} = \mathrm{C}_{\mathrm{RC}} + p\, \mathrm{C}_{\mathrm{LBMS}}, \tag{33}$$

where C_{RC} stands for the cost of reconstruction and compression operations (steps 2 and 5 of Figure 5) and $\mathrm{C}_{\mathrm{LBMS}}$ is the combined cost of steps (a)-(b)-(c) in the loop 3 of Figure 5, particulary of the collision step and streaming step needed for the evolution of distribution functions. As in the FEM case, the cost $\mathrm{C}_{\mathrm{LBM}}$ is supposed to be known, and is also highly dependent on the number of degrees of freedom (type of cells and resolution of the lattice-Boltzmann grid).

We first evaluate the computational cost of a sequential FEM-LBM coupling, denoted by $\mathrm{C_{seq}}$ and which can be expressed as follows:

$$\mathrm{C_{seq}} \simeq k_s N(\mathrm{C_{FEM}} + \mathrm{C_{LBM}}). \tag{34}$$

The above formula simply states that for each of the N coarse time-steps of length Δt, k_s iterations are carried out to couple the FEM and LBM solvers. We supposed that the number of iterations needed is approximatively the same at each time-step.
Instead, for the Parareal FEM-LBM coupling algorithm of Figure 7, the computational cost can be estimated as:

$$\mathrm{C_{par}} \simeq N\mathrm{C_{FEM}} + k_p(\mathrm{C_{LBM}} + N\mathrm{C_{FEM}}) = k_p\mathrm{C_{LBM}} + N(1+k_p)\mathrm{C_{FEM}}. \tag{35}$$

This formula is obtained observing that the initialization implies N coarse computations, and then, at each of the k_p correction iterations of Parareal, the LBM computations are done in parallel, while the prediction-correction step implies again N coarse computations with FEM.
From equations (34) and (35), the speed-up $\mathrm{S_P}$ is obtained as:

$$\begin{aligned} \mathrm{S_P} =& \frac{\mathrm{C_{seq}}}{\mathrm{C_{par}}} \\ \simeq& \frac{k_s N(\mathrm{C_{FEM}} + \mathrm{C_{LBM}})}{k_p\mathrm{C_{LBM}} + N(1+k_p)\mathrm{C_{FEM}}} \\ \simeq& \frac{k_s N(\mathrm{C_{FEM}} + \mathrm{C_{RC}} + p\,\mathrm{C_{LBMS}})}{k_p(\mathrm{C_{RC}} + p\,\mathrm{C_{LBMS}}) + N(1+k_p)\mathrm{C_{FEM}}}, \end{aligned} \tag{36}$$

where in the last line we used (33). At this stage, one reasonable assumption is that the numbers k_s, k_p (and $k_p + 1$) are approximatively the same. Indeed, in practice, only a few iterations are needed for both sequential and parallel versions. So if the number of coarse propagations N is moderate, this assumption may hold. Conversely, this assumption may be unreasonable if N is too high. In the case the above assumption $k_s \simeq k_p \simeq k_p + 1$ holds, we obtain the following simplified formula:

$$\mathrm{S_P} \simeq \frac{N(\mathrm{C_{FEM}} + \mathrm{C_{LBM}})}{\mathrm{C_{LBM}} + N\mathrm{C_{FEM}}} \simeq \frac{N(\mathrm{C_{FEM}} + \mathrm{C_{RC}} + p\,\mathrm{C_{LBMS}})}{\mathrm{C_{RC}} + p\,\mathrm{C_{LBMS}} + N\mathrm{C_{FEM}}}. \tag{37}$$

One interesting case that can be considered for practical purposes is when p is sufficiently large so as to neglect the cost of reconstruction and compression operations C_{RC} and we then obtain:

$$\mathrm{S_P} \simeq \frac{N\mathrm{C_{FEM}} + Np\,\mathrm{C_{LBMS}}}{N\mathrm{C_{FEM}} + p\,\mathrm{C_{LBMS}}}. \tag{38}$$

This case is realistic since an implicit scheme in the FEM solver (backward Euler for instance) allows large time-steps Δt. The above formula (38) highlights the gain rate due to parallelism: the cost associated to LBM computations is reduced from Np to p.
At last, another asymptotic formula can be obtained from (36) for long-time simulations, which corresponds to the case $N \to +\infty$:

$$\begin{aligned} \mathrm{S_P} \simeq& \frac{k_s(\mathrm{C_{FEM}} + \mathrm{C_{LBM}})}{(1+k_p)\mathrm{C_{FEM}}} \\ \simeq& \frac{k_s(\mathrm{C_{FEM}} + \mathrm{C_{RC}} + p\,\mathrm{C_{LBMS}})}{(1+k_p)\mathrm{C_{FEM}}}. \end{aligned} \tag{39}$$

In this situation, one could speculate that the maximal speed-up is obtained by taking p as large as possible and the FEM solver as fast as possible, but this is balanced by the fact that this may strongly increase the number of iterations k_p necessary for convergence. Note that furthermore, k_p is also increased when N is large.

4.5 System efficiency

The system efficiency $\mathrm{S_E}$ is the ratio between the speed-up and the number of processors involved in the parallel computation. Ideally this number must be close to 1, while maintaining a speed-up as high as possible. From formula (36), it is given by:

$$\mathrm{S_E} = \frac{\mathrm{S_P}}{N} \simeq \frac{k_s(\mathrm{C_{FEM}} + \mathrm{C_{LBM}})}{k_p\mathrm{C_{LBM}} + N(1+k_p)\mathrm{C_{FEM}}}. \tag{40}$$

Note that the algorithm becomes less efficient if N is too high, due to the sequential correction step involving FEM. However if N is moderate and the FEM solver very cheap ($\mathrm{C_{FEM}} \ll \mathrm{C_{LBM}}$), the system efficiency is quite preserved.

5 Numerical results

The FEM-LBM coupling algorithm of Figure 7 is implemented with help of FreeFEM++, an open-source finite element library [33]. For the FEM method, standard $\mathbb{P}_1$ continuous finite elements have been used and the backward Euler time-marching scheme given in (3) is implemented. For the LBM method, the D2Q4 lattice structure is always considered (see §2.2).

5.1 A preliminary test case

The main purpose of this experiment is to illustrate the solution of Problem (1) with the proposed FEM-LBM solver on a simple configuration. The same problem will be used in the following section to analyze the convergence properties of our method. The solutions in the coarse and fine domains are reported for different timesteps. The geometry considered is given in Figure 8 (a). For the FEM-LBM solver, the FEM domain is given by $\Omega_c = (-\frac{1}{4}; \frac{3}{4}) \times (-\frac{1}{4}; \frac{3}{4})$ whereas the LBM domain is defined by $\Omega_f = (0; \frac{1}{2}) \times (0; \frac{1}{2})$. For the reference FEM solution only Ω_c is considered.

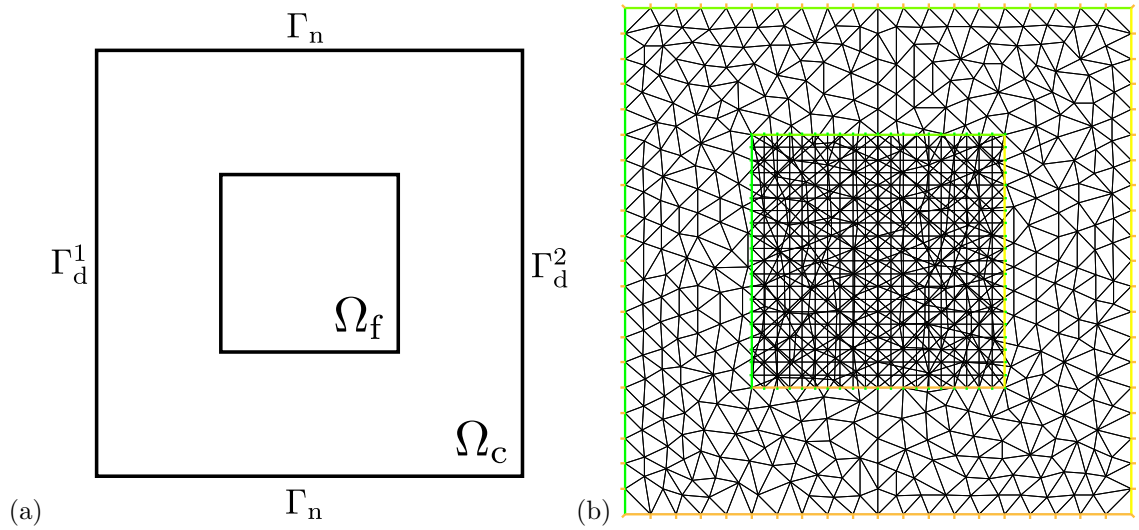
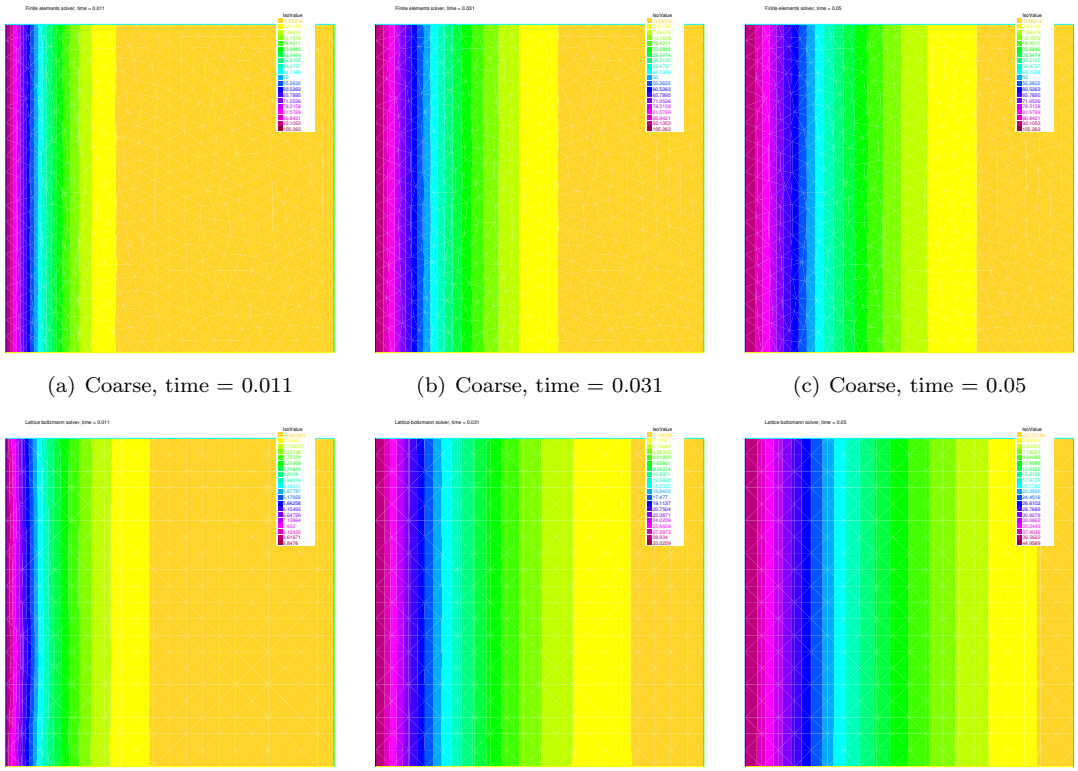


Figure 8: (a) Geometry for FEM domain Ω_c and LBM patch Ω_f. (b) Spatial discretization of the computational domain: coarse mesh for FEM and LB cells on a patch.

For Problem (1) the value of the diffusion coefficient is $\mu = 1$ and the source term F is equal to 0. On Ω_c we impose two Dirichlet boundary conditions: $u = 100$ on Γ_d^1 and $u = 0$ on Γ_d^2. On the remaining boundaries Γ_n a zero Neumann condition is imposed. The whole time interval is $(0, T)$ - with $T = 0.05$.
For the fine solver (LBM) Equation (19) of Theorem 3.1 is adopted to reconstruct the distribution functions. The fine domain is discretized with 20×20 cells. For the coarse solver (FEM) the coarse domain is discretized with 942 triangular elements. Note that the fine and coarse discretizations are non-conforming, see Figure 8 (b). For the fine and coarse solvers the timesteps are $\delta t = 10^{-5}$ and $\Delta t = 10^{-3}$, respectively. For this test case, the number of processors of the Parareal algorithm of Figure 7 is $N = 50$ (corresponding to the total number of coarse timesteps) and the convergence criterion is $\varepsilon^* = 10^{-5}$.
The algorithm converges in 3 iterations. The evolution of the fine and coarse solutions at different timesteps is given for each iteration in Figures 9, 10 and 11. The converged coarse solution is reported in Figure 12 for the same timesteps. Note on Figure 10 the effect of the coupling due to the first prediction-correction iteration.

5.2 Convergence behavior for the FEM-LBM coupling

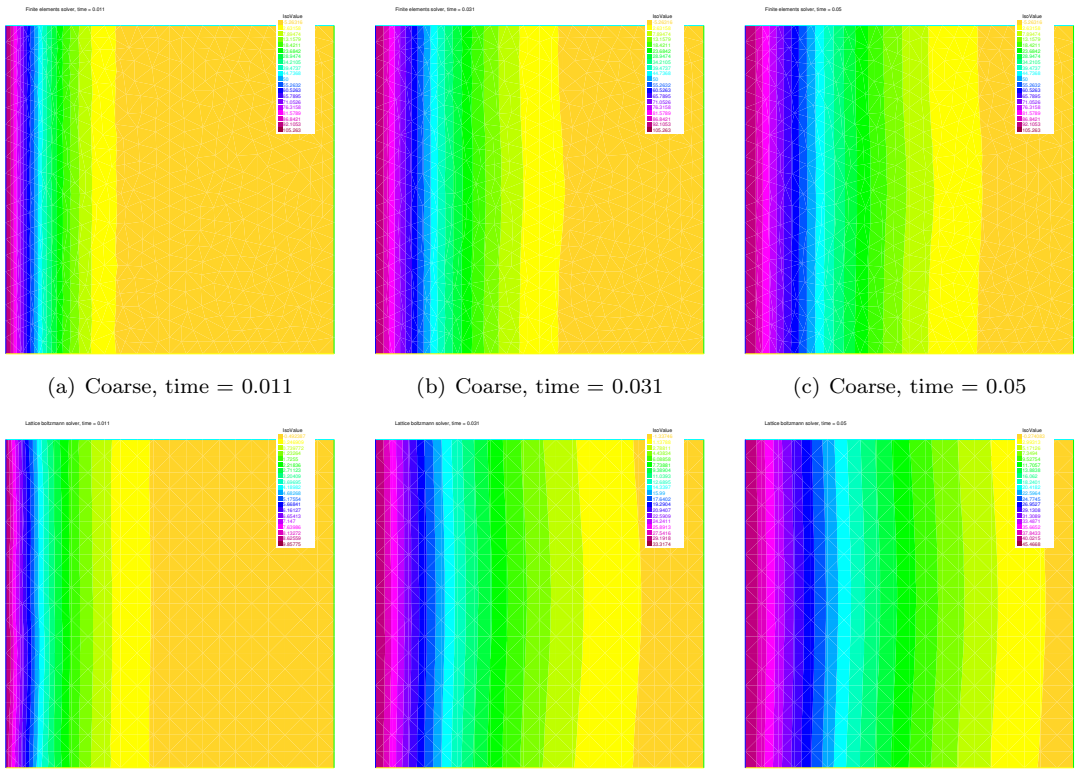
In this experiment, we solve the same physical problem as described in §5.1. For this test case the right end of time interval is $T = 0.025$. For the LBM, both order 0 (equation (18)) and 1 (equation (19)) formulas of Theorem 3.1 have been considered for the reconstruction of the distribution functions. The FEM domain Ω_c has been meshed with an unstructured mesh, the number of mesh nodes on each edge of the boundary $\partial\Omega$ being $n_c = 20$. The number of LB cells on each row of the LB domain Ω_f is $n_f = 40$.
The parameters of the Parareal FEM-LBM coupling algorithm are as follows: number of coarse time-steps $N = 50$, ratio coarse/fine time-steps $p = \Delta t/\delta t = 50$, time-step for the fine solver $\delta t = 10^{-5}$.
We carry out a simulation with a very small convergence criterion ($\varepsilon^* = 10^{-10}$) in order to assess the convergence properties of the algorithm. For order 0 reconstruction, the residuals ε_n^k are depicted in Figure 13. For a given



(a) Coarse, time = 0.011 (b) Coarse, time = 0.031 (c) Coarse, time = 0.05

(d) Fine, time = 0.011 (e) Fine, time = 0.031 (f) Fine, time = 0.05

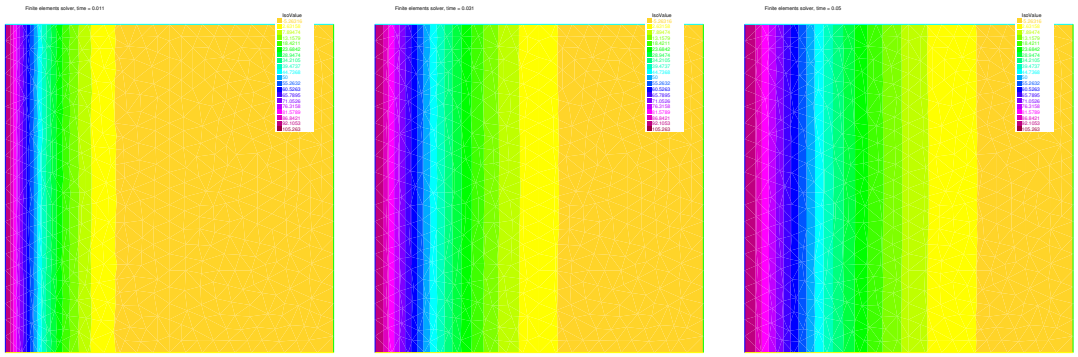
Figure 9: Coarse and fine solution at different timesteps for iteration 0 of the FEM-LBM coupling.



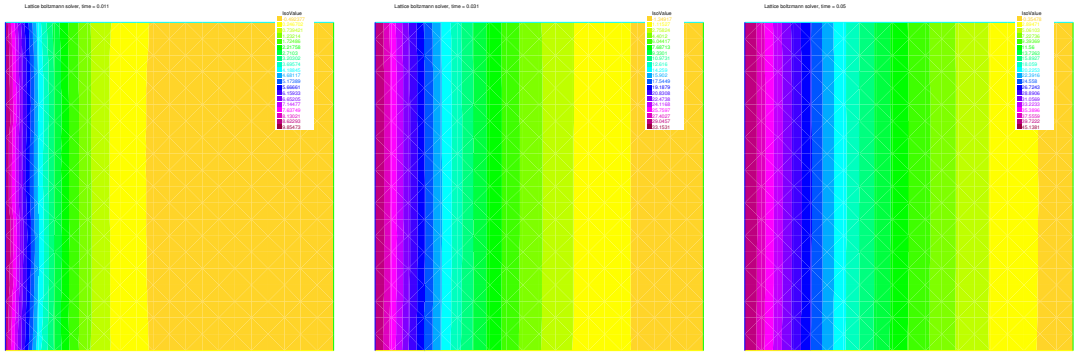
(a) Coarse, time = 0.011 (b) Coarse, time = 0.031 (c) Coarse, time = 0.05

(d) Fine, time = 0.011 (e) Fine, time = 0.031 (f) Fine, time = 0.05

Figure 10: Coarse and fine solution at different timesteps for iteration 1 of the FEM-LBM coupling.

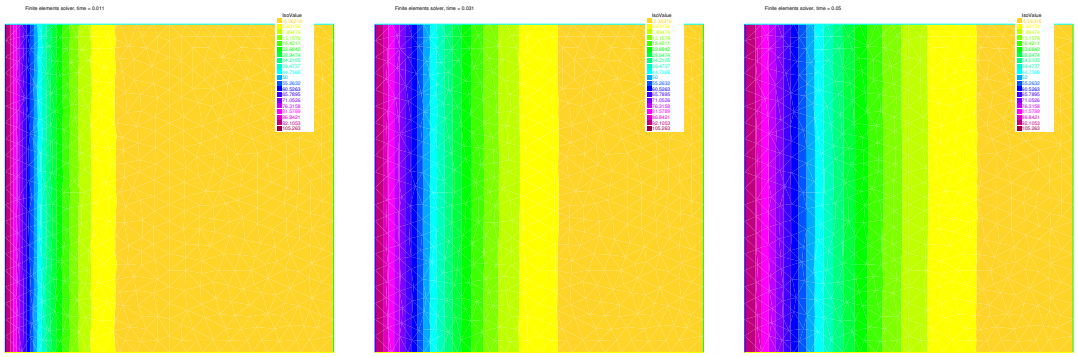


(a) Coarse, time = 0.011 (b) Coarse, time = 0.031 (c) Coarse, time = 0.05



(d) Fine, time = 0.011 (e) Fine, time = 0.031 (f) Fine, time = 0.05

Figure 11: Coarse and fine solution at different timesteps for iteration 2 of the FEM-LBM coupling.



(a) Coarse, time = 0.011 (b) Coarse, time = 0.031 (c) Coarse, time = 0.05

Figure 12: Converged coarse solution at different timesteps.

value of the correction iteration k, the residuals first increase with n, which is due to the correction process that gives a better approximation on the first time-steps. Then it attains a maximum and decreases, which may be due to the diffusive process. When k is increased, the residuals are decreased monotonically, and the algorithm stops when all the residuals ε_n^k are below the value $\varepsilon^* = 10^{-10}$. For order 1 reconstruction, the residuals are depicted in Figure 14. The global convergence behavior is comparable to that of order 0 reconstruction, but the whole algorithm converges much faster than with order 0 reconstruction (13 iterations instead of 18), which confirms the interest of taking into account the gradient of the FEM solution in the reconstruction formula. Note also that, for small values of n, the behavior of the curves is slightly more complex.

The computed fine solution $u(t, x_c, y_c)$ at the center of the domain $x_c = \frac{1}{4}, y_c = \frac{1}{4}$ in function of time t and for various correction iterations k is depicted in Figure 15. The first observation is that the final coupled FEM-LBM solution is really different from the first coarse prediction ($k = 0$) which involved only the coarse FEM solver. The other observation is that a few correction iterations (2 in this case) are sufficient to obtain a good approximation of the final coupled solution.

Finally we also compared the Parareal FEM-LBM coupling with a Parareal FEM-FEM coupling, in which the

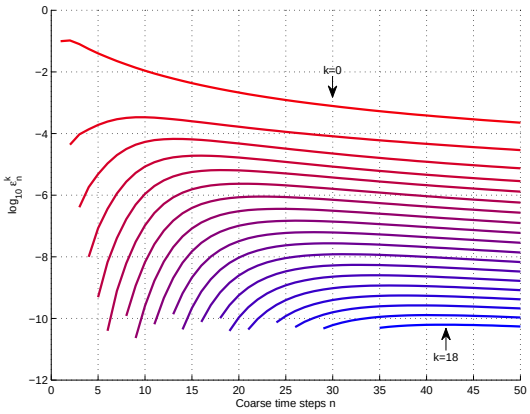


Figure 13: Order 0 reconstruction. The residuals ε_n^k (in log-scale) as a function of the coarse time-steps n for various values of k. Colors from red to blue correspond to increasing values of k.

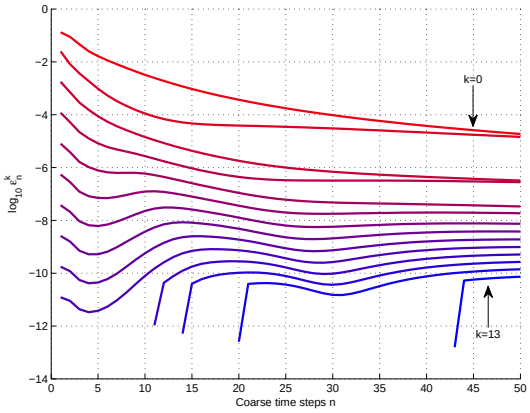


Figure 14: Order 1 reconstruction. The residuals ε_n^k (in log-scale) as a function of the coarse time-steps n for various values of k. Colors from red to blue correspond to increasing values of k.

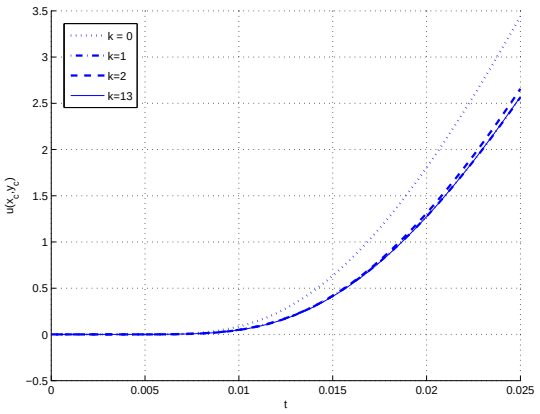


Figure 15: Evolution of the fine solution at the center of the domain, for various values of k.

LBM solver on the patch has been replaced by a FEM solver. The FEM on the patch is built identically as the FEM on the global domain (with $\mathbb{P}_1$-continuous elements, and backward Euler time-marching scheme), while the discretization parameters δt and h have been chosen identical to the LBM parameters. The final (converged) solutions are displayed Figure 16, and compare well.

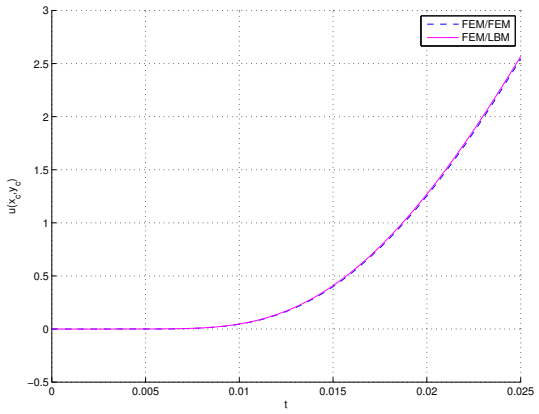


Figure 16: Evolution of the fine solution at the center of the domain. Comparison of FEM-LBM coupling and FEM-FEM coupling.

5.3 Influence of discretization in space and time

In this section we study the influence of main discretization parameters, namely the ratio between the coarse and fine meshes n_c/n_f and the ratio between the coarse and fine time-steps p $(= \Delta t/\delta t)$. We start from the same configuration as in §5.2. Since results are better with the reconstruction formula of order 1, we keep this choice. The mesh of the FEM domain Ω_c is unstructured and is kept unchanged, with also $n_c = 20$. The other parameters are as follows: number of coarse time-steps $N = 50$, time-step for the fine solver $\delta t = 10^{-5}$ and convergence criterion $\varepsilon^* = 10^{-5}$.
The number of total iterations k_p (starting from $k = 0$) to reach convergence for different values of the couple (p, n_f) is given in Table 1.

p \ n_f	10	20	30	40	60	80	100
30	8	7	5	4	3	3	4
50	6	5	4	4	3	2	4
100	5	4	4	3	3	3	3
150	4	3	3	3	3	3	3
200	4	3	3	3	2	2	3
250	5	4	3	2	2	2	2
300	4	4	3	2	2	2	2

Table 1: Influence of numerical parameters n_f and p on the total number k_p of correction iterations necessary for convergence $(\varepsilon^* = 10^{-5})$.

The first observation is that the algorithm converges after a few iterations: eight iterations are needed in the worst case $(p = 30, n_f = 10)$ and this number is lower for other values of the parameters. The number of iterations is not subjected to significant variations for a wide range of the parameters. Note that the convergence is very good for high values of p and n_f: only two or three iterations are needed. For values of n_f greater than 100, the CFL in the LBM solver is not respected anymore with the choice of the fine time-step $\delta t = 10^{-5}$, which prevents the algorithm to converge.

5.4 A more complex test-case

In this last test we take advantage of the flexibility of FEM to illustrate the use of our FEM-LBM coupling in a more complex computational problem. Let us consider the simulation of temperature evolution of a steel airfoil such as the one represented in Figure 17. The inner squared domain represents the region where the LBM method is applied. The airfoil domain Ω_c is characterized by an initial temperature $u_0 = 298.15°\mathrm{K}$ and surrounded by air at $u_s = 253.15°\mathrm{K}$ on Γ_R. An imposed temperature $u_d = 348.15°\mathrm{K}$ is applied on a small portion Γ_D of the boundary, at the bottom of the airfoil $(\partial\Omega = \Gamma_R \cup \Gamma_D)$. The steel is characterized by a density $\rho = 7860\ \mathrm{kg/m^3}$, a specific heat $c = 502\ \mathrm{J/Kg\,K}$ and a thermal conductivity $\kappa = 60\mathrm{W/m\,K}$. For the air a convection heat transfer

coefficient $\alpha = 45$ W/m^2 K is assumed. The total simulation time is $T = 1000$ s. The length L of the airfoil is 0.8 m.
This problem is described by the following equations:

$$\begin{aligned} \rho c \frac{\partial u}{\partial t} - \kappa \, \Delta u = 0 \qquad & \text{in} \quad \Omega_{\mathrm{c}} \times (0, T), \\ u(\cdot, 0) = u_0 \qquad & \text{in} \quad \Omega_{\mathrm{c}}, \\ \kappa \frac{\partial u}{\partial n} + \alpha u = \alpha u_s \qquad & \text{on} \quad \Gamma_{\mathrm{R}}, \\ u = u_d \qquad & \text{on} \quad \Gamma_{\mathrm{D}}. \end{aligned} \tag{41}$$

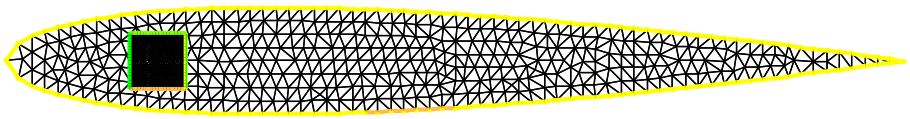


Figure 17: Computational domain for the airfoil test case.

The Parareal FEM-LBM coupling algorithm is applied with the following parameters: mesh sizes $n_{\mathrm{c}} = 20$ and $n_{\mathrm{f}} = 20$, number of coarse time-steps $N = 50$, ratio coarse/fine time-steps $p = 100$, time-step for the fine solver $\delta t = 0.2$ and convergence criterion $\varepsilon^* = 10^{-5}$. The number of correction iterations necessary for convergence is $k_p = 2$. The final FEM and LBM solutions are displayed in Figures 18 and 19.

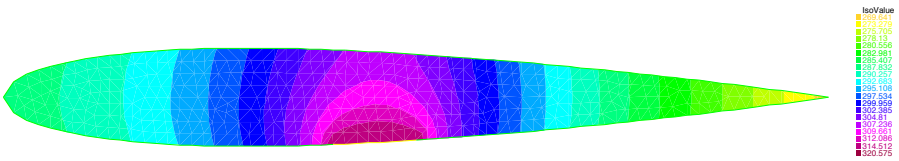


Figure 18: Final FEM solution (temperature in K) for the airfoil test case.

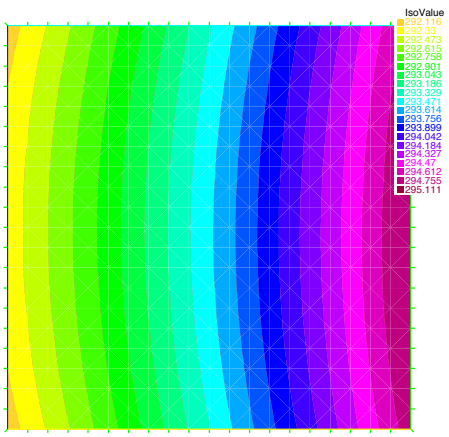


Figure 19: Final LBM solution (temperature in K) for the airfoil test case.

6 Discussion

6.1 Summary on the method and numerical experiments

We proposed a method for time-dependent problems based on the coupling of FEM and LBM. Its efficiency derive mostly from the Parareal framework where the FEM solver is adopted as a coarse predictor for the LBM fine solver. Although other numerical methods can be chosen, in this work we have considered the FEM and the LBM for their intrinsic properties. The LBM, derived from the Boltzmann equation, can be more appropriate to simulate physics at some mesoscopic level (see, e.g., [61, 34, 3, 58]), or to compute some specific quantities

[1, 2]. The FEM, strengthened by the variational framework, is extremely flexible for macroscopic descriptions associated with complex geometries and boundary conditions. The Parareal framework for the coupling between FEM and LBM leads to a simple approach, attractive from the computational point of view. On the one hand, the fine computations are parallelized in time, therefore the use of a small time-step, typically needed in LBM computations, is limited to smaller time-slab. On the other hand for the FEM a larger time-step can be chosen provided that an implicit time-marching scheme is implemented. Additionally the LBM is highly parallelizable in space [5], therefore a fine mesh in the mesoscopic scale can be foreseen. For the FEM, a coarser space discretization is sufficient since it only serves to provide a coarse prediction. For the spatial coupling between both methods, various techniques exist for the derivation of the transmission conditions. In this work we used the asymptotic analysis of [37, 11, 12] but other techniques such as Chapman-Enskog expansion or Grad's type distribution would have been valuable as well and would have left to the same kind of results.

Remark furthermore that no numerical instability has been observed due to interpolation. This should be confirmed by further numerical experiments on more complex test cases, but note already that any spurious oscillation coming from the fine propagator may be filtered when transmitted to the coarse one, due to the interpolation on the coarser mesh.

In previous works the coupling between LBM and FDM/FVM is operational but it lacks of full overlap typically needed in some multiscale problems, where fine details are searched only in a localized patch (as in numerical zoom methods, see, e.g., [30, 4, 32]). Moreover the previous methods are inspired from classical Schwarz iterations, that are purely sequential and well-fitted for stationary problems. Yet, they may be less natural and interesting for time-evolution problems.

Numerical experiments confirm the good behavior of the overall algorithm on a simple model problem (the heat equation) and illustrate its flexibility. For a wide range of the physical parameters, the convergence is attained within a few iterations (less than 4). This point is important since the speed-up due to parallelization is strongly dependent on the number of correction iterations, which should be the lowest possible. Nevertheless, increasing too much the critical variables for spatial and temporal discretization (namely p and the ratio $n_{\mathrm{f}}/n_{\mathrm{c}}$) will end up with a decrease of the speed-up since it will also increase the cost of the fine computations.

As a final remark, it is important to note that our coupling approach can also be considered as a valid alternative among the multiple techniques proposed in the LBM literature for the imposition of macroscopic boundary conditions. As can be observed from the last experiment, the implementation of some complex boundary conditions finds a natural formulation in the FEM. On the contrary the application of similar boundary conditions to the LBM would have been more challenging.

6.2 Some remarks on Parareal and the numerical zoom

The Parareal algorithm thanks to its versatility and genericity is suitable for time-parallel time-integration of a wide class of evolution problems described by ODE's/PDE's [43, 23, 29]. It involves a fine propagator and a coarse propagator. The latter serves to predict quickly an inaccurate solution during the (sequential) correction iterations. There is freedom to design a coarse propagator provided that it delivers an approximation of the fine solution.

The most obvious and simple choice for the coarse propagator is thus to implement the same method as in the fine solver, but with a coarser spatio-temporal discretization to obtain a cheaper resolution. An alternative choice consists of using a different, and simpler, mathematical model than in the fine solver: for instance a simpler differential equation or a reduced set of equations. After an earlier remark in this direction (see e.g. [7]), this possibility has not been so extensively explored until now, apart from a few exceptions. In [21], Parareal is applied to multiscale stochastics chemical kinetics, and while the fine solver uses stochastic simulation at mesoscopic scales, the coarse predictor uses a reaction rate equation at macroscopic scale. Also in the context of chemical kinetics, the authors of [10] design a coarse propagator using a reduction method: a reduced set of chemical reactions is considered for the coarse prediction. At last, in reference [31] the possibility of using reduced basis methods for design of a cheapest coarse solver is explored. Let us finally mention some recent extensions of the Parareal paradigm to multiscale problems, in the ODE context [42] and for numerical zoom methods [16]. Our present work goes into this direction as well and tends to confirm that the Parareal methodology is well suited for problems such as FEM/LBM coupling.

From the point of view of numerical zoom, as pointed out in the introduction, most of the existing techniques are devoted to stationary problems to the best of our knowledge. In this work we extended the framework of classical Parareal so that it might be adapted for numerical zoom purposes. In the original version of Parareal, the coarse and fine solvers must work on the same domain, whereas the Parareal FEM-LBM algorithm described in Figure 7 allows a coarse solver on a larger domain.

6.3 Some perspectives

The model problem (1) has been chosen for simplicity reasons and in order to focus on methodological aspects, as in, e.g., [1, 2]. Thus the LBM solver has been considered merely as a macro-scale solver, and neither intrinsic multi-scale effects nor richer physics have been integrated into the lattice Boltzmann computations.
In future works, application of the proposed method to more realistic problems may be considered. For instance, there have been recent interest to use LBM for discretization of the Fokker-Planck equation, with application to non-newtonian fluids, such as polymer dilute solutions [3, 58]. In these situation LBM reveals to be interesting in comparison to stochastic simulation, since it is easily implemented and quite accurate. More generally for complex fluids (non-newtonian fluids, heterogeneous, multi-scale or multi-component fluids) there is still some need to combine various methods for different physics [34], and LBM seems to be promising and relevant for phenomena at the mesoscopic level [61], while FEM could remain an appropriate method for the macroscopic scale. For biological fluids such as blood flows, both FEM and LBM methods have been tested with convincing results, though obviously they have their pros and cons (see, e.g., [27, 50, 8]). Therefore a coupling method that would allow to combine them (almost) optimally could be of interest for blood flow simulation.
In such a perspective, the derivation of coupling conditions between FEM and LBM needs to be proceeded carefully for more complex models. In addition to well known techniques, some new numerical algorithms for approximating these conditions, such as point-wise iterative schemes [62] or numerical Chapman-Enskog expansion [63] may be beneficial and represent a direction that can be further explored.
Additionally, issues related to efficient implementation on parallel architectures, such as GPU, need to be studied carefully. Various works deal with efficient implementations of the Parareal method, and could serve to inspire an improved implementation of the algorithm of Figure 7, see, e.g., [59, 49, 6, 9]. In [6] a distributed version of Parareal has been designed, and the author provide some numerical evidence that the algorithm can be well-fitted for heterogeneous architectures, with a low communication overhead. Moreover in [51] a GPU implementation of Parareal has been designed and tested. These preliminary tests reveal once more that Parareal can perform better than classical (spatial) domain decomposition in some situations, and that communication and latency issues have less influence on speed-up. So an idea would be to combine a algorithm such as presented in [51, Algorithm 2, p.103] with massively parallel GPU implementation of lattice Boltzmann methods, such as in [8, 5].

Acknowledgements

First, the authors thank Alexei Lozinski and Alfonso Caiazzo for stimulating discussions and for some constructive remarks that allowed to improve this work. Moreover the second author would like to thank Miguel Fernández, Jean-Frédéric Gerbeau, Sidi Mahmoud Kaber, Yvon Maday, Joaquín Mura, Frédéric Legoll and Florian Lemarié for very encouraging and interesting discussions on the Parareal algorithm.

A A remark on the consistency of LBM

Using the asymptotic analysis in the proof of Theorem 3.1, we can check that the LBM discretization is consistent with the heat equation (1) (see Remark 3.2). We detail this point below.
Let us sum (26) for each $i = 1, \dots, 4$:

$$\sum_{i=1}^{4} f_i^{[2]} = \sum_{i=1}^{4} f_i^{eq,[2]} - \tau \left(\sum_{i=1}^{4} (\mathbf{e}_i \cdot \boldsymbol{\nabla}) f_i^{[1]} + \sum_{i=1}^{4} \left(\frac{\partial}{\partial t} + \frac{1}{2} (\mathbf{e}_i \cdot \boldsymbol{\nabla})^2 \right) f_i^{[0]} \right). \tag{42}$$

Note first that

$$\sum_{i=1}^{4} f_i^{[2]} = \sum_{i=1}^{4} f_i^{eq,[2]} = u_{LB}^{[2]}. \tag{43}$$

Then, using equation (28) we rewrite:

$$\begin{aligned} \sum_{i=1}^{4} (\mathbf{e}_i \cdot \boldsymbol{\nabla}) f_i^{[1]} &= \frac{1}{4} \sum_{i=1}^{4} (\mathbf{e}_i \cdot \boldsymbol{\nabla}) (u_{LB}^{[1]} - \tau (\mathbf{e}_i \cdot \boldsymbol{\nabla}) u_{LB}^{[0]}) \\ &= \frac{1}{4} \left(\sum_{i=1}^{4} (\mathbf{e}_i \cdot \boldsymbol{\nabla}) \right) u_{LB}^{[1]} - \frac{\tau}{4} \left(\sum_{i=1}^{4} (\mathbf{e}_i \cdot \boldsymbol{\nabla})^2 \right) u_{LB}^{[0]} \\ &= -\frac{\tau}{2} \Delta u_{LB}^{[0]}. \end{aligned} \tag{44}$$

The last equation comes from the structure of the D2Q4 lattice (symmetry) which implies $\sum_{i=1}^{4}(\mathbf{e}_i \cdot \boldsymbol{\nabla}) = 0$ and $\sum_{i=1}^{4}(\mathbf{e}_i \cdot \boldsymbol{\nabla})^2 = 2\Delta$.
With help of equation (27) and relationship $\sum_{i=1}^{4}(\mathbf{e}_i \cdot \boldsymbol{\nabla})^2 = 2\Delta$ we transform:

$$\begin{aligned}\sum_{i=1}^{4}\left(\frac{\partial}{\partial t}+\frac{1}{2}(\mathbf{e}_i\cdot\boldsymbol{\nabla})^2\right)f_i^{[0]} &= \left(\sum_{i=1}^{4}\left(\frac{\partial}{\partial t}+\frac{1}{2}(\mathbf{e}_i\cdot\boldsymbol{\nabla})^2\right)\right)\frac{1}{4}u_{LB}^{[0]}\\ &= \frac{\partial u_{LB}^{[0]}}{\partial t}+\frac{1}{8}\left(\sum_{i=1}^{4}(\mathbf{e}_i\cdot\boldsymbol{\nabla})^2\right)u_{LB}^{[0]}\\ &= \frac{\partial u_{LB}^{[0]}}{\partial t}+\frac{1}{4}\Delta u_{LB}^{[0]}. \end{aligned} \tag{45}$$

Plugging relationships (43)–(45) into (42) yields:

$$\frac{\partial u_{LB}^{[0]}}{\partial t}-\frac{1}{2}(\tau-\frac{1}{2})\Delta u_{LB}^{[0]}=0.$$

Since $u_{LB} = u_{LB}^{[0]} + \mathcal{O}(\varepsilon)$, we finally obtain:

$$\frac{\partial u_{LB}}{\partial t}-\frac{1}{2}(\tau-\frac{1}{2})\Delta u_{LB}=\mathcal{O}(\varepsilon).$$

So the lattice Boltzmann solution is consistent at (least at) first order in ε with the heat equation (1) provided that $\mu = \frac{1}{2}(\tau - \frac{1}{2})$ and $\delta t = \delta x^2$.

References

[1] P. Albuquerque, D. Alemani, B. Chopard, and P. Leone, *Coupling a lattice Boltzmann and a finite difference scheme*, in Proceedings of ICCS 2004: 4th International Conference on Computational Science (Kraków, 2004), M. Bubak, G.D. van Albada, P.M.A. Sloot, J.J. Dongarra, eds., Vol. 3039, Part IV, Springer Berlin/Heidelberg, 2004, pp. 540–547.

[2] ——, *A hybrid lattice Boltzmann finite difference scheme for the diffusion equation*, Int. J. Mult. Comp. Eng., 4 (2006), pp. 209–219.

[3] A. Ammar, *Lattice boltzmann method for polymer kinetic theory*, J. Non-Newton. Fluid., 165 (2010), pp. 1082 – 1092.

[4] J.-B. Apoung Kamga and O. Pironneau, *Numerical zoom for multiscale problems with an application to nuclear waste disposal*, J. Comput. Phys., 224 (2007), pp. 403–413.

[5] M. Astorino, J. Becerra Sagredo, and A. Quarteroni, *A modular lattice Boltzmann for GPU computing processors*, SeMA Journal, 59 (2012), pp. 53–78.

[6] E. Aubanel, *Scheduling of tasks in the parareal algorithm*, Parallel Comput., 37 (2011), pp. 172–182.

[7] L. Baffico, S. Bernard, Y. Maday, G. Turinici, and G. Zérah, *Parallel-in-time molecular-dynamics simulations.*, Phys. Rev. E: Stat. Nonlinear Soft Matter Phys., 66 (2002), p. 057701.

[8] M. Bernaschi, M. Fatica, S. Melchionna, S. Succi, and E. Kaxiras, *A flexible high-performance lattice Boltzmann GPU code for the simulations of fluid flows in complex geometries*, Concurrency-Pract. Ex., 22 (2010), pp. 1–14.

[9] L. A. Berry, W. Elwasif, J. M. Reynolds-Barredo, D. Samaddar, R. Sanchez, and D. E. Newman, *Event-based parareal: A data-flow based implementation of parareal*, J. Comput. Phys., 231 (2012), pp. 5945–5954.

[10] A. Blouza, L. Boudin, and S. Kaber, *Parallel in time algorithms with reduction methods for solving chemical kinetics*, Commun. Appl. Math. Comput. Sci., 5 (2010), pp. 241–263.

[11] A. Caiazzo, *Asymptotic Analysis of lattice Boltzmann method for Fluid-Structure Interaction problems.*, PhD thesis, Technische Universität Kaiserslautern, Scuola Normale Superiore Pisa, 2007.

[12] A. Caiazzo and M. Junk, *Asymptotic analysis of lattice Boltzmann methods for flow-rigid body interaction*, Progress in Computational Physics Volume 3: Novel Trends in Lattice-Boltzmann Methods, 3 (2013), p. 91.

[13] P. Chen, *The lattice Boltzmann method for fluid dynamics: theory and applications*, Master's thesis, EPFL, 2011.

[14] S. S. Chikatamarla, S. Ansumali, and I. V. Karlin, *Grad's approximation for missing data in lattice Boltzmann simulations*, Europhys. Lett., 74 (2006), pp. 215–221.

[15] B. Chopard, J. L. Falcone, and J. Latt, *The lattice Boltzmann advection-diffusion model revisited*, The European Physical Journal - Special Topics, 171 (2009), pp. 245–249. 10.1140/epjst/e2009-01035-5.

[16] F. Chouly and A. Lozinski, *Parareal multi-model numerical zoom for parabolic multiscale problems*, C. R. Math. Acad. Sci. Paris, 352 (2014), pp. 535–540.

[17] A. Dervieux, *Fluid-structure interaction*, Hermes, Paris, 2000.

[18] M. Discacciati and A. Quarteroni, *Navier-Stokes/Darcy coupling: modeling, analysis, and numerical approximation*, Revista Matemática Complutense, 22 (2009), pp. 315–426.

[19] A. Dupuis, E. Kotsalis, and P. Koumoutsakos, *Coupling lattice Boltzmann and molecular dynamics models for dense fluids*, Phys. Rev. E, 75 (2007), p. 046704.

[20] A. Dupuis and P. Koumoutsakos, *Effects of atomistic domain size on hybrid lattice Boltzmann-molecular dynamics simulations of dense fluids*, Int. J. Mod. Phys. C, 18 (2007), pp. 644–641.

[21] S. Engblom, *Parallel in time simulation of multiscale stochastic chemical kinetics*, Multiscale Model. Simul., 8 (2009), pp. 46–68.

[22] A. Ern and J.-L. Guermond, *Theory and practice of finite elements*, vol. 159 of Applied Mathematical Sciences, Springer-Verlag, New York, 2004.

[23] C. Farhat and M. Chandesris, *Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid-structure applications*, Int. J. Num. Meth. Eng., 58 (2003), pp. 1397–1434.

[24] D. Fedosov and G. Karniadakis, *Triple-decker: Interfacing atomistic-mesoscopic-continuum flow regimes*, J. Comput. Phys., 228 (2009), pp. 1157–1171.

[25] M. A. Fernández, L. Formaggia, J.-F. Gerbeau, and A. Quarteroni, *The derivation of the equations for fluids and structure*, in Cardiovascular mathematics, vol. 1 of MS&A. Model. Simul. Appl., Springer Italia, Milan, 2009, pp. 77–121.

[26] P. Fischer, F. Hecht, and Y. Maday, *A parareal in time semi-implicit approximation of the Navier-Stokes equations*, Domain decomposition methods in science and engineering, (2005), pp. 433–440.

[27] L. Formaggia, A. Quarteroni, and A. Veneziani, *Cardiovascular Mathematics*, MS&A: Modeling, Simulation and Applications, Springer, 2009.

[28] M. Fyta, S. Melchionna, E. Kaxiras, and S. Succi, *Multiscale coupling of molecular dynamics and hydrodynamics: Application to translocation through a nanopore*, Multiscale Model. Simul., 5 (2006), pp. 1156–1173.

[29] M. Gander and S. Vandewalle, *Analysis of the parareal time-parallel time-integration method*, SIAM J. Sci. Comp., 29 (2007), pp. 556–578.

[30] R. Glowinski, J. He, A. Lozinski, J. Rappaz, and J. Wagner, *Finite element approximation of multi-scale elliptic problems using patches of elements*, Numer. Math., 101 (2005), pp. 663–687.

[31] L. He, *The reduced basis technique as a coarse solver for parareal in time simulations*, J. Comput. Math., 28 (2010), pp. 676–692.

[32] F. Hecht, A. Lozinski, and O. Pironneau, *Numerical zoom and the Schwarz algorithm*, in Domain decomposition methods in science and engineering XVIII, vol. 70 of Lect. Notes Comput. Sci. Eng., Springer, Berlin, 2009, pp. 63–73.

[33] F. Hecht, O. Pironneau, J. Morice, A. Le Hyaric, and K. Ohtsuka, *Freefem++ documentation, third edition, version 3.19-1*, 2012. Webpage: http://www.freefem.org/ff++.

[34] D. M. Heyes, J. Baxter, U. Tüzün, and R. S. Qin, *Discrete-element method simulations: from micro to macro scales*, Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci., 362 (2004), pp. 1853–1865.

[35] A. Hiorth, U. H. a Lad, S. Evje, and S. M. Skjaeveland, *A lattice Boltzmann-BGK algorithm for a diffusion equation with Robin boundary condition–application to NMR relaxation*, Int. J. Num. Meth. Fl., 59 (2009), pp. 405–421.

[36] H. Huang, X. Lu, and M. Sukop, *Numerical study of lattice Boltzmann methods for a convection–diffusion equation coupled with Navier–Stokes equations*, J. Phys. A-Math. Gen., 44 (2011), p. 055001.

[37] M. Junk, A. Klar, and L.-S. Luo, *Asymptotic analysis of the lattice Boltzmann equation*, J. Comput. Phys., 210 (2005), pp. 676–704.

[38] P. Kao and R. Yang, *An investigation into curved and moving boundary treatments in the lattice Boltzmann method*, J. Comput. Phys., 227 (2008), pp. 5671–5690.

[39] P. Ladevèze, J.-C. Passieux, and D. Néron, *The LATIN multiscale computational method and the proper generalized decomposition*, Comput. Methods Appl. Mech. Engrg., 199 (2010), pp. 1287–1296.

[40] J. Lätt, *Hydrodynamic limit of lattice Boltzmann equations*, PhD thesis, Geneva University, 2007.

[41] J. Lätt, B. Chopard, O. Malaspinas, M. Deville, and A. Michler, *Straight velocity boundaries in the lattice Boltzmann method*, Phys. Rev. E, 77 (2008), p. 056703.

[42] F. Legoll, T. Lelièvre, and G. Samaey, *A micro-macro parareal algorithm: application to singularly perturbed ordinary differential equations*, SIAM J. Sci. Comput., 35 (2013), pp. A1951–A1986.

[43] J.-L. Lions, Y. Maday, and G. Turinici, *Résolution d'EDP par un schéma en temps "pararéel"*, C. R. Math. Acad. Sci. Paris, 332 (2001), pp. 661–668.

[44] H. B. Luan, H. Xu, L. Chen, D. L. Sun, and W. Q. Tao, *Numerical illustrations of the coupling between the lattice Boltzmann method and finite-type macro-numerical methods*, Numer. Heat Tr. B-Fund., 57 (2010), pp. 147–171.

[45] Y. Maday and G. Turinici, *The Parareal in time iterative solver: a further direction to parallel implementation*, in Domain decomposition methods in science and engineering, vol. 40 of Lect. Notes Comput. Sci. Eng., Springer, Berlin, 2005, pp. 441–448.

[46] O. Malaspinas, B. Chopard, and J. Latt, *General regularized boundary condition for multi-speed lattice Boltzmann models*, Comput. Fluids, 49 (2011), pp. 29–35.

[47] R. Mei, L.-S. Luo, P. Lallemand, and D. d'Humières, *Consistent initial conditions for lattice Boltzmann simulations*, Comput. Fluids, 35 (2006), pp. 855–862.

[48] S. Melchionna, M. Fyta, E. Kaxiras, and S. Succi, *Exploring DNA translocation trough a nanopore via a multiscale lattice-Boltzmann molecular-dynamics methodology*, Int. J. Mod. Phys. C, 18 (2007), pp. 685–692.

[49] M. L. Minion, *A hybrid parareal spectral deferred corrections method*, Commun. Appl. Math. Comput. Sci., 5 (2010), pp. 265–301.

[50] R. W. Nash, H. B. Carver, M. O. Bernabeu, J. Hetherington, D. Groen, T. Krüger, and P. V. Coveney, *Choice of boundary condition for lattice-Boltzmann simulation of moderate-Reynolds-number flow in complex domains*, Phys. Rev. E, 89 (2014), p. 023303.

[51] A. S. Nielsen, *Feasibility study of the parareal algorithm*, 2012. MSc thesis, Technical University of Denmark.

[52] K. Perktold, M. Prosi, and P. Zunino, *Mathematical models of mass transfer in the vascular walls*, in Cardiovascular mathematics, vol. 1 of MS&A. Model. Simul. Appl., Springer Italia, Milan, 2009, pp. 243–278.

[53] Y. Qian, D. d'Humieres, and P. Lallemand, *Lattice BGK models for Navier-Stokes equation*, EPL (Europhysics Letters), 17 (1992), p. 479.

[54] A. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential equations*, Springer, 1997.

[55] G. Romano, *Multiscale thermal models of nanostructured devices*, PhD thesis, Università degli Studi di Roma "Tor Vergata", 2010.

[56] J. Russo, J. Horbach., F. Sciortino, and S. Succi, *Nanoflows through disordered media: A joint lattice Boltzmann and molecular dynamics investigation*, EPL (Europhysics Letters), 89 (2010), p. 44001.

[57] X. Shan, X.-F. Yuan, and H. Chen, *Kinetic theory representation of hydrodynamics: a way beyond the Navier-Stokes equation*, J. Fluid Mech., 550 (2006), pp. 413–441.

[58] S. Singh, G. Subramanian, and S. Ansumali, *Lattice fokker planck for dilute polymer dynamics*, Phys. Rev. E, 88 (2013), p. 013301.

[59] A. Srinivasan and N. Chandra, *Latency tolerance through parallelization of time in scientific applications*, Parallel Comput., 31 (2005), pp. 777–796.

[60] H. Stetter, *The defect correction principle and discretization methods*, Numer. Math., 29 (1978), pp. 425–443.

[61] S. Succi, O. Filippova, G. Smith, and E. Kaxiras, *Applying the lattice Boltzmann equation to multiscale fluid problems*, Comput. Sci. Eng., 3 (2001), pp. 26–37.

[62] P. Van Leemput, C. Vandekerckhove, W. Vanroose, and D. Roose, *Accuracy of hybrid lattice Boltzmann/finite difference schemes for reaction-diffusion systems*, Multiscale Model. Sim, 6 (2007), pp. 838–857.

[63] Y. Vanderhoydonc and W. Vanroose, *Numerical extraction of a macroscopic PDE and a lifting operator from a lattice Boltzmann model*, Multiscale Model. Simul., 10 (2012), pp. 766–791.

[64] J. C. Verschaeve and B. Müller, *A curved no-slip boundary condition for the lattice Boltzmann method*, J. Comput. Phys., 229 (2010), pp. 6781–6803.

[65] E. Weinberg, D. Shahmirzadi, and M. Mofrad, *On the multiscale modeling of heart valve biomechanics in health and disease*, Biomech. Model. Mechanobiol., 9 (2010), pp. 373–387. 10.1007/s10237-009-0181-2.

[66] P. Wriggers and M. Hain, *Micro-meso-macro modelling of composite materials*, in III European Conference on Computational Mechanics, C. Motasoares, J. Martins, H. Rodrigues, J. Ambrosio, C. Pina, C. Motasoares, E. Pereira, and J. Folgado, eds., Springer Netherlands, 2006, pp. 37–37.

[67] H. Xu, H. Luan, Y. He, and W. Tao, *A lifting relation from macroscopic variables to mesoscopic variables in lattice Boltzmann method: Derivation, numerical assessments and coupling computations validation*, Comput. Fluids, 54 (2012), pp. 92 – 104.

[68] D. P. Ziegler, *Boundary conditions for lattice Boltzmann simulations*, J. Stat. Phys., 71 (1993), pp. 1171–1177.