MOX–Report No. 47/2012

Multiscale coupling of finite element and lattice Boltzmann methods for time dependent problems

Astorino, M.; Chouly, F.; Quarteroni, A.

MOX, Dipartimento di Matematica "F. Brioschi"
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox@mate.polimi.it http://mox.polimi.it

Multiscale coupling of finite element and lattice Boltzmann methods for time dependent problems

M. Astorino[a], F. Chouly[b], 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 two methods are regarded as macroscale and mesoscale solvers, respectively.
The procedure is based on the Parareal paradigm and allows for a truly multiscale coupling between two numerical methods having optimal efficiency at different space and time scales. 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.
The theoretical and numerical framework is presented for parabolic equations even though its potential applicability is much wider (e.g. Navier-Stokes equations). Various numerical examples on the heat equation will validate the proposed procedure and illustrate its multiple advantages.

Keywords: finite element method, lattice Boltzmann method, multiscale coupling, 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 modeling and simulation of systems that exhibit a degree of complexity and diversity spanning across many spatiotemporal scales (e.g. macroscopic, mesoscopic and microcopic scales). Example of these *multiscale problems* can be found in diverse fields such as thermodynamics [46], material science [60], fluid flows [18], biology [22] and biomechanics [59], just to mention a few.

The challenges related to the development of multiscale frameworks are essentially twofold. On the one hand, the different equations and variables describing the process over the various scales must be consistently coupled (by e.g. weighted averages or homogenization techniques [43]). On the other hand, numerical procedures that take systematically into account all the scales have to be developed. This last point is often hardly achievable adopting a single numerical approach over all the spatiotemporal scales, either for lack of computational efficiency or for lack of physical relevance. For instance, molecular dynamics modeling is suitable for microscale problems but experience limitations in problems with computational domains larger than a few millimeters in size [30, 33, 58, 53].

Therefore none of the existing numerical method is currently acknowledged to be suited and optimally efficient for all the physical scales; but for each scale a number of suitable methods can be identified. For example, it is widely accepted that classical continuum methods such as the finite difference method (FDM) [50], the finite element method (FEM) [45] and the finite volume method (FVM) [16] are suited for macroscale problems. Similarly the lattice Boltzmann method (LBM) [51], the dissipative particle dynamics (DPD) [42] and the direct simulation Monte Carlo method (DSMC) [7] belong to the class of mesoscopic methods, while numerical approaches like molecular dynamics (MD) [25] and quantum dynamics (QD) [34, 56] can be regarded as microscopic methods.

A number of works dealing with multiscale problems proposed *hybrid approaches* based on a combination of various numerical techniques in order to overcome the above mentioned difficulty. These frameworks, also named *composite computational methods*, employ on each scale a different numerical method, aiming at the best numerical efficiency and physical accuracy. Examples of these approaches are numerous. Among others we recall the bridging scale method presented in [57, 54] for the atomistic-continuum coupling of MD and FEM. A similar technique adopted in [33] couples the mesoscopic and macroscopic scales using DPD and FEM. A full micro-meso-macro coupling, the so-called triple-decker, has been investigated in [18] with MD, DPD and FEM.

In the field of the lattice Boltzmann method, the general multiscale theory has been covered in [52]. For this mesoscopic method different hybrid approaches have been considered in various micro-meso applications. Results have been presented for instance in nanoflows through disordered media [47], DNA translocation [22, 41] and dense fluids [12, 13]. On the contrary, the coupling between LBM and macroscale methods has been addressed only in a few works. In [2, 3] a FDM-LBM coupling has been presented for the heat equation, the same coupling is applied also to reaction-diffusion problems in [55] and to the Navier-Stokes equations in [35, Chapter 7]. For the same fluid equations a FVM-LBM coupling has been considered in [38, 61].

Neglecting the differences related to the physical problem and to the macroscopic method considered, the cited works have two main elements in common, specifically the derivation of the coupling conditions and the coupling technique. The meso-macro coupling conditions have been obtained by means of a Chapman-Enskog expansion (up to the appropriate order) of the lattice Boltzmann equa-

tion (see for example [2, 61]). For the coupling 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. [3]).

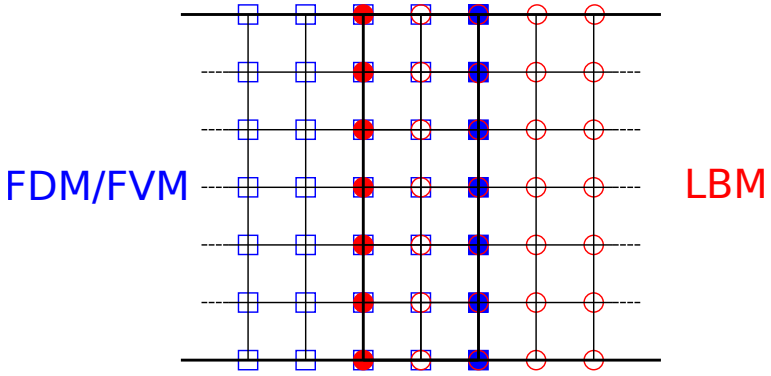


Figure 1: Sketch of the overlapping Schwarz method for the existing meso-macro coupling with LBM. Squares and circles identify the degrees of freedom (DOF) for the two methods, that are separately applied on the two regions. The filled squares and circles are the respective interface DOF.

In this work, we still consider a macro-meso coupling, but using FEM and LBM. The choice of FEM at a macroscale level is motivated by its well-assessed mathematical framework [45, 15] and by its flexibility in the treatment of complex boundary and interface conditions. As in the previous literature, our approach is also based on the Chapman-Enskog expansion to derive the transmission conditions between LBM and FEM, but the overall coupling algorithm is different and 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). In this perspective the LBM can be truly reinterpreted as a fine grain solver applied in a localized patch, whereas the FEM is a coarse grain solver 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. [24, 4, 27].
2. the coupling strategy relies on a time domain decomposition approach given by the Parareal algorithm [37, 17, 39, 23]. This naturally fits our multiscale evolutive problem and enhances the computational performance through parallelization in time. Different time discretizations may be chosen for the two scales provided that the timestep of FEM is a multiple of the LBM one.

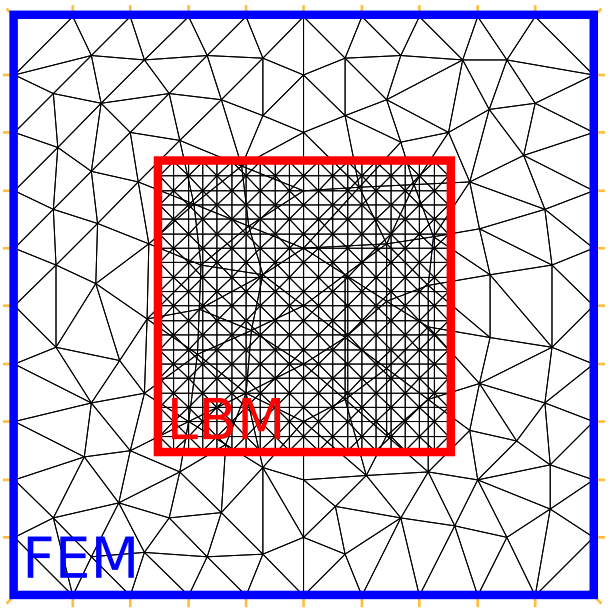


Figure 2: Example of the spatial discretization in our meso-macro coupling between LBM and FEM.

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 method takes full advantage of the Parareal algorithm to overcome this problem. Indeed, the macroscopic problem is considered as a coarse prediction of the solution on the whole space/time domain, whereas the mesoscopic 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 mesoscopic problems over smaller time intervals, each one of length equal to the coarse timestep.
In this study, for the sake of simplicity, we consider a simplified model problem: the two-dimensional heat equation. Extension to more complex cases will be object of future works. The outline of the paper is as follows. The model problem, its finite element and lattice Boltzmann approximations are described in Sec-

tion 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 scales. In Section 4, we describe our version of the Parareal algorithm for coupling in time. Section 5 is dedicated to numerical experiments. Finally, 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, $F : \Omega \times (0,T) \to \mathbb{R}$, $g : \Gamma \times (0,T) \to \mathbb{R}$ 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).

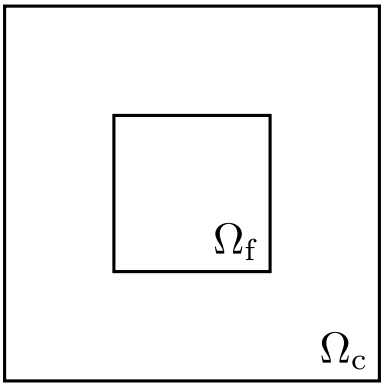


Figure 3: The two domains for FEM and LBM solving.

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:

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

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

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

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 macroscale 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 resctrictive). 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:

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

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 [11, 29], that can be retrieved from an appropriate discretization of the Boltzmann-Maxwell equation

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

Equation (4) is a mesoscopic conservation equation for the *particle distribution function* $f(\mathbf{e}, \mathbf{x}, t)$, so that $f(\mathbf{e}, \mathbf{x}, t)\mathbf{d}\mathbf{e}\mathbf{d}\mathbf{x}$ is the total mass of particles inside the

infinitesimal volume element d**e**d**x** at a fixed time, position and velocity. The quantity **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 [48] 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 mesoscopic 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) [9, 35]. 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 [44], τ is the so-called *relaxation time* and f_i^{eq} is an appropriate discrete approximation of the Maxwell-Boltzmann equilibrium distribution [48]. 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 exact 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 [31].

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.*

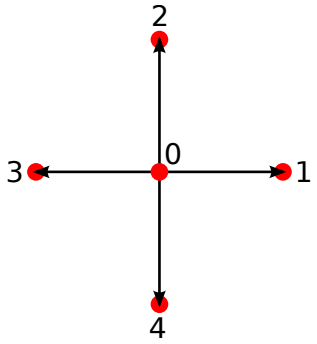


Figure 4: D2Q5 model. Dots represent the lattice nodes $\boldsymbol{x}_i, \forall i = 0, ..., 4$ in the unit cell. The arrows identify the links $\mathbf{e}_i, \forall i = 0, ..., 4$ in the lattice structure, their lengths are given by $\|\boldsymbol{x}_0 - \boldsymbol{x}_i\|, \forall i = 0, ..., 4$.

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

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

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:

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

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

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 [40, 32, 36] 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 [10].

3. Spatial coupling between FEM and LBM

In this section, we detail our methodological approach for the spatial coupling between the macroscale (FEM) and mesoscale (LBM). First the continuous coupling conditions are derived following the approach presented in [2, 3]. 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 [2, 61] the spatial coupling of the macro and meso processes relies on the definition of compression ($\mathcal{C}$) and reconstruction ($\mathcal{R}$) operations between the state variables of the two scales, respectively u_{LB} for the macroscale and $f_{i|i=0,...,Q}$ for the mesoscale:

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

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

While the compression operation is unique and defined by the local-ensemble average (11), the reconstruction operation is not. Note in fact that the mesoscale problem contains more information than the macroscale problem. Thus the reconstruction procedure leads to a one-to-many mapping.
The derivation of an analytic expression for (16) follows the approach proposed in [2, 61] and is reported below for the case of the D2Q4 lattice model, with zero external forces. For other models, the derivation is similar.

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

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

with $u_{LB}(\mathbf{x},t) = \sum_1^4 f_i(\mathbf{x},t) = \sum_1^4 f_i^{eq}(\mathbf{x},t)$ *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 operator are respectively given by:*

$$f_{i|i=0,...,4}(\mathbf{x},t) = \frac{1}{4}u_{LB}(\mathbf{x},t), \qquad \forall i = 0,...,4. \tag{18}$$

and by

$$f_{i|i=0,...,4}(\mathbf{x},t) = \frac{1}{4}(u_{LB} - \delta t\tau \mathbf{e}_i \cdot \boldsymbol{\nabla} u_{LB})(\mathbf{x},t), \qquad \forall i = 1,...,4. \tag{19}$$

Proof. In (17) let us assume δt small and let us Taylor expand $f_i(\mathbf{x}+\mathbf{e}_i\delta x, t+\delta t)$ around $f_i(\mathbf{x},t)$ up to the second order. This yelds:

$$\delta t D_i f_i(\mathbf{x},t) + \delta t^2 D_i^2 f_i(\mathbf{x},t) = -\frac{1}{\tau}(f_i - f_i^{eq}) \qquad \forall i = 1,...,4, \tag{20}$$

being $D_i = (\frac{\partial}{\partial t} + \mathbf{e}_i \cdot \boldsymbol{\nabla})$.
We now use in (20) the multiscale Chapman-Enskog expansion [21], that is we expand f_i in powers of the small parameter ε (also referred as the Knudsen number):

$$f_i(\mathbf{x},t) = f_i^{[0]}(\mathbf{x},t) + \varepsilon f_i^{[1]}(\mathbf{x},t) + \varepsilon^2 f_i^{[2]}(\mathbf{x},t) + ... \qquad \forall i = 1,...,4. \tag{21}$$

In a similar way, let us introduce the expansions in time and space:

$$\frac{\partial}{\partial t} = \varepsilon \frac{\partial}{\partial t^{[1]}} + \varepsilon^2 \frac{\partial}{\partial t^{[2]}} \quad \text{and} \quad \boldsymbol{\nabla} = \varepsilon \boldsymbol{\nabla}^{[1]}. \tag{22}$$

Plugging (21) and (22) in (20) and keeping all the terms up to second order in ε we obtain

$$\begin{aligned} &\varepsilon \mathbf{e}_i \cdot \boldsymbol{\nabla}^{[1]} f_i^{[0]} + \varepsilon^2 \boldsymbol{\nabla}^{[1]} f_i^{[1]} + \varepsilon^2 \frac{\partial f_i^{[0]}}{\partial t^{[1]}} + \frac{\delta t}{2}\varepsilon^2 \mathbf{e}_i^2 \Delta^{[1]} f_i^{[0]} \\ &\qquad = -\frac{1}{\delta t \tau}(f_i^{[0]} + \varepsilon f_i^{[1]} + \varepsilon^2 f_i^{[2]} - f_i^{eq}) \qquad \forall i = 1,...,4. \end{aligned} \tag{23}$$

This equation holds for each order separately. Equating the zero-th and first order contributions, $f_i^{[0]}$ and $f_i^{[1]}$ can be rewritten as

$$f_i^{[0]} = f_i^{eq} = \frac{u_{LB}}{4}, \tag{24}$$

$$\varepsilon f_i^{[1]} = -\delta t \tau \mathbf{e}_i \varepsilon \cdot \boldsymbol{\nabla}^{[1]} f_i^{[0]} = -\frac{\delta t \tau}{4} \mathbf{e}_i \cdot \boldsymbol{\nabla} u_{LB}. \tag{25}$$

Therefore the particle distribution functions can be reconstructed at the zero-th order with

$$f_i(\mathbf{x},t) = f_i^{[0]}(\mathbf{x},t) = \frac{1}{4}u_{LB}(\mathbf{x},t) \qquad \forall i = 1,...,4, \tag{26}$$

and at the first order with

$$f_i(\mathbf{x},t) = f_i^{[0]}(\mathbf{x},t) + \varepsilon f_i^{[1]}(\mathbf{x},t) = \frac{1}{4}(u_{LB} - \delta t\tau \mathbf{e}_i \cdot \boldsymbol{\nabla} u_{LB})(\mathbf{x},t) \quad \forall i = 1,...,4. \tag{27}$$

□

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:

$$\mathcal{G}_{\Delta t}(t) : \left\{ \begin{array}{lcl} V^H & \to & V^H \\ u_H(t) & \mapsto & u_H(t + \Delta t), \end{array} \right. \tag{28}$$

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 $t+\Delta t$:

$$\mathcal{F}_{\Delta t}(t) : \left\{ \begin{array}{lcl} V^h \times V^h & \to & V^h \\ (u_h(t), u_h^*(t + \Delta t)) & \mapsto & u_h(t + \Delta t), \end{array} \right. \tag{29}$$

where V^h is the finite element space of piecewise linear continuous functions

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

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 Section 2.2. There are in fact two inputs: $u_h(t)$ is the initial condition given at time t, and $u_h^*(t + \Delta t)$ is a (possibly inaccurate) prediction of the solution at time $t + \Delta t$ extracted from the coarse propagator. This last input is of interest for interpolation of the boundary conditions on $\partial\Omega_{\mathrm{f}}$. This point as well as other major features of $\mathcal{F}_{\Delta t}(t)$ are detailed below.

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 (29) 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 $\partial\Omega_{\mathrm{f}}$, which are unknown *a-priori* since the LBM domain is only a portion included into the FEM domain. To this purpose, we first interpolate linearly in time the numerical solution between $u_h(t)$ and $u_h^*(t + \Delta t)$, for better accuracy. From this interpolation u_b^j of the physical quantity, the reconstruction operation is carried out once again for obtention of the distribution functions f_i on the boundary.

Fine propagator $\mathcal{F}_{\Delta t}(u_h(t), u_h^*(t+\Delta t))$:

1. *Initialization:* $u_h^0 = u_h(t)$.
2. *From physical quantities to distribution functions:*
   - Reconstruction operation
     $f_{i|i=0,...,4}(\mathbf{x}, t) = \frac{1}{4}(u_h^0(\mathbf{x}) - \alpha\, \delta t \tau \mathbf{c}_i \cdot \boldsymbol{\nabla} u_h^0(\mathbf{x}))$,
     for grid points $\mathbf{x} \in \Omega_\mathrm{f}$ and for $i = 1, ..., 4$.
3. *Loop on small time steps* $\delta t = \Delta t/p$:
   $j = 1, \ldots, p; \quad t^j = t + j\delta t$.
   (a) *Linear interpolation in time:*
   $$u_b^j = (1 - \frac{j}{p}) u_h(t) + \frac{j}{p} u_h^*(t + \Delta t).$$
   (b) *Reconstruction of distribution functions on the boundary:*
   - Reconstruction operation
     $f_{i|i=0,...,4}(\mathbf{x}, t^j) = \frac{1}{4}(u_b^j(\mathbf{x}) - \alpha\, \delta t \tau \mathbf{c}_i \cdot \boldsymbol{\nabla} u_b^j(\mathbf{x}))$,
     for grid points $\mathbf{x} \in \partial\Omega_\mathrm{f}$ and for $i = 1, ..., 4$.

   (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: $u_h(\mathbf{x}, t + \Delta t) = \sum_{i=0}^{4} f_i(\mathbf{x}, t^p)$.

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

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 $\Pi_\mathrm{F}^\mathrm{L}$ the operator that transfers the FEM solution to the lattice Boltzmann space. Reciprocally, the operator $\Pi_\mathrm{L}^\mathrm{F}$ transfers the LBM solution to the finite element space. In practice, $\Pi_\mathrm{F}^\mathrm{L}$ (resp. $\Pi_\mathrm{L}^\mathrm{F}$) is simply a linear interpolation operator from V^H to V^h (resp. from V^h to V^H) [28]. Due to the inclusion of the domains ($\Omega_f \subset \Omega_c$), the interpolation $\Pi_\mathrm{L}^\mathrm{F} v \in V^H$ 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 $\Pi_{\mathrm{F}}^{\mathrm{L}}$ can be expressed as follows

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

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 $\Pi_{\mathrm{L}}^{\mathrm{F}}$). Algorithms for construction of such rectangular matrices are now implemented in numerous finite element librairies (see e.g. the function `interpolate` in FreeFEM++ [28]).

4. Time coupling between FEM and LBM using Parareal

In this part, we first explain the idea 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. Coupling in time between coarse and fine propagators

To solve Problem (1), we first compute a coarse FEM approximation of the solution u on the whole time inteval $(0, T)$. We keep the notations already introduced in Section 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:

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

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 (or defect-correction [49]) procedure [37, 23]:

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

for $n = 0, \ldots, N-1$. In this step, the new coarse prediction is corrected by the difference between the previous fine prediction and the previous coarse prediction. 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} the 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 Section 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, which produces at each iteration k a sequence of solutions $(u_h^{n,k})_n$. Increasing k enhances the coupling degree between the two methods.

4.2. 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: $(u_H^{n,k}, u_h^{n,k})_n$. Let us introduce 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 key ingredient of our Parareal FEM-LBM coupling procedure is then the following prediction-correction iteration:

$$\begin{aligned} u_h^{n+1,k+1} := & \underbrace{\Pi_{\mathrm{F}}^{\mathrm{L}} \mathcal{G}_{\Delta t}(t^n)\left(\tilde{u}_{h/H}^{n,k+1}\right)}_{\text{Prediction}} \\ & + \underbrace{\mathcal{F}_{\Delta t}(t^n)(u_h^{n,k}, u_h^{n+1,k}) - \Pi_{\mathrm{F}}^{\mathrm{L}} \mathcal{G}_{\Delta t}(t^n)\left(\tilde{u}_{h/H}^{n,k}\right)}_{\text{Correction}}, \end{aligned} \tag{30}$$

for $0 \leq n \leq N-1$. The complete Parareal FEM-LBM algorithm in pseudo-code is given Figure 6.
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 (30) 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 [17], only the $n_0, \ldots, N$ 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. *As the algorithm 6 involves two distinct spatial discretizations and associated transmission operators, it follows the work originally proposed in [19] where a Parareal method with two spatial grids is applied to solve the Navier-Stokes equations.*

Parareal FEM-LBM coupling:

1. *Initialization step (–):*
   - First coarse prediction:
$$u_H^{0,0} = u_H^0 \,;\, u_H^{n+1,0} := \mathcal{G}_{\Delta t}(t^n)(u_H^{n,0}), \quad n = 0, \ldots, N-1,$$
$$\tilde{u}_{h/H}^{n,0} := u_H^{n,0}, \qquad n = 0, \ldots, N,$$
$$u_h^{n,0} := \Pi_{\mathrm{F}}^{\mathrm{L}} u_H^{n,0}, \qquad n = 0, \ldots, N.$$
   - 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 (//):*
$$\hat{u}_h^{n+1,k} := \mathcal{F}_{\Delta t}(t^n)(u_h^{n,k}, u_h^{n+1,k}),\ n = n_0, \ldots, N-1.$$
   (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:
$$n^* = \min\{n \mid n_0 \le n < N; \varepsilon_n^k > \varepsilon^*\}.$$
   (d) *Prediction-correction (–):*
   compute the new values $(u_h^{n,k+1})_n$, $n = n_0, \ldots, N-1$.
$$\tilde{u}_{h/H}^{n,k+1} := \chi(\Pi_{\mathrm{L}}^{\mathrm{F}} u_h^{n,k}) + (1-\chi) u_H^{n,k+1},$$
$$u_H^{n+1,k+1} := \mathcal{G}_{\Delta t}(t^n)(\tilde{u}_{h/H}^{n,k+1}),$$
$$u_h^{n+1,k+1} := \Pi_{\mathrm{F}}^{\mathrm{L}}(u_H^{n+1,k+1} - u_H^{n+1,k}) + \hat{u}_h^{n+1,k}.$$
   (e) Update the index of the Parareal iteration: $k \leftarrow k+1$.
3. *End while.*

Figure 6: 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.

Remark 4.2. *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 multiscale coupled solution, with a fine LBM prediction on the required sub-region.*

Remark 4.3. *When the evolution equation is non-linear the last equation of step 2 (d) in Figure 6 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 6 we took advantage of the linearity of our model problem (1) *to factorize the transmission operator, which involves less computations.*

4.3. 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).
For this purpose, let us first use the notation $\mathrm{C_{FEM}}$ 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 Section 3.2 and Figure 5), we detail the cost of one fine computation, that we note $\mathrm{C_{LBM}}$:

$$\mathrm{C_{LBM}} = \mathrm{C_{RC}} + p\,\mathrm{C_{LBMS}}, \tag{31}$$

where $\mathrm{C_{RC}}$ stands for the cost of reconstruction and compression operations (steps 2 and 5 of Figure 5) and $\mathrm{C_{LBMS}}$ 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 $\mathrm{C_{LBM}}$ 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). In this following study, for the sake of simplicity, we will neglect the computational cost associated to the transmission operators $\Pi_{\mathrm{F}}^{\mathrm{L}}$ and $\Pi_{\mathrm{L}}^{\mathrm{F}}$. We first evaluate the computational cost of a sequential FEM-LBM coupling, denoted by $\mathrm{C_{seq}}$ and which can be expressed as follows:

$$\mathrm{C_{seq}} \simeq k_s N(\mathrm{C_{FEM}} + \mathrm{C_{LBM}}). \tag{32}$$

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 6, the computational cost can be estimated as:

$$\mathrm{C_{par}} \simeq N\mathrm{C_{FEM}} + k_p(\mathrm{C_{LBM}} + N\mathrm{C_{FEM}}) = k_p\mathrm{C_{LBM}} + N(1+k_p)\mathrm{C_{FEM}}. \tag{33}$$

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 (32) and (33), the speed-up $\mathrm{S_P}$ is obtained in a straightforward way:

$$\begin{aligned}\mathrm{S_P} =& \frac{\mathrm{C_{seq}}}{\mathrm{C_{par}}} \\ \simeq& \frac{k_s N(\mathrm{C_{FEM}} + \mathrm{C_{LBM}})}{k_p \mathrm{C_{LBM}} + N(1+k_p)\mathrm{C_{FEM}}} \\ \simeq& \frac{k_s N(\mathrm{C_{FEM}} + \mathrm{C_{RC}} + p\,\mathrm{C_{LBMS}})}{k_p(\mathrm{C_{RC}} + p\,\mathrm{C_{LBMS}}) + N(1+k_p)\mathrm{C_{FEM}}},\end{aligned} \tag{34}$$

where in the last inequality we used (31). At this stage, one reasonable assumption we can make 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 iterations 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:

$$\mathrm{S_P} \simeq \frac{N(\mathrm{C_{FEM}} + \mathrm{C_{LBM}})}{\mathrm{C_{LBM}} + N\mathrm{C_{FEM}}} \simeq \frac{N(\mathrm{C_{FEM}} + \mathrm{C_{RC}} + p\,\mathrm{C_{LBMS}})}{\mathrm{C_{RC}} + p\,\mathrm{C_{LBMS}} + N\mathrm{C_{FEM}}}. \tag{35}$$

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

$$\mathrm{S_P} \simeq \frac{N\mathrm{C_{FEM}} + Np\,\mathrm{C_{LBMS}}}{N\mathrm{C_{FEM}} + p\,\mathrm{C_{LBMS}}}. \tag{36}$$

This case is realistic since an implicit scheme in the FEM solver (Euler implicit for instance) allows large time-steps Δt. The above formula (36) 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 (34) for long-time simulations, which corresponds to the case $N \to +\infty$:

$$\begin{aligned}\mathrm{S_P} \simeq& \frac{k_s(\mathrm{C_{FEM}} + \mathrm{C_{LBM}})}{(1+k_p)\mathrm{C_{FEM}}} \\ \simeq& \frac{k_s(\mathrm{C_{FEM}} + \mathrm{C_{RC}} + p\,\mathrm{C_{LBMS}})}{(1+k_p)\mathrm{C_{FEM}}}.\end{aligned} \tag{37}$$

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.

4.4. 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 maintening a speed-up as high as possible. From formula (34), it is given by:

$$\mathrm{S_E} = \frac{\mathrm{S_P}}{N} \simeq \frac{k_s(\mathrm{C_{FEM}} + \mathrm{C_{LBM}})}{k_p \mathrm{C_{LBM}} + N(1+k_p)\mathrm{C_{FEM}}}. \tag{38}$$

Note that the algorithm becomes less efficient if N is too big, which is 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 6 is implemented with help of FreeFEM++, an open-source finite element library [28]. 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 multiscale solver on a relatively simple configuration. The same problem will be used in the following section to analyze the convergence properties of our multiscale approach. The solutions in the coarse and fine domains are reported for different timesteps. The geometry considered is given in Figure 7 (a). For the FEM-LBM solver, the FEM domain is given by $\Omega_\mathrm{c} = (-\frac{1}{4}; \frac{3}{4}) \times (-\frac{1}{4}; \frac{3}{4})$ whereas the LBM domain is defined by $\Omega_\mathrm{f} = (0; \frac{1}{2}) \times (0; \frac{1}{2})$. For the reference FEM solution only Ω_c is considered.
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 7 (b).
For the mesoscopic and macroscopic 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 6 is $N = 50$ (corresponding to the total number of coarse timesteps) and the convergence criterion is $\varepsilon^* = 10^{-5}$.
The multiscale algorithm converges in 3 iterations. The evolution of the fine and coarse solutions at different timesteps is given for each multiscale iteration

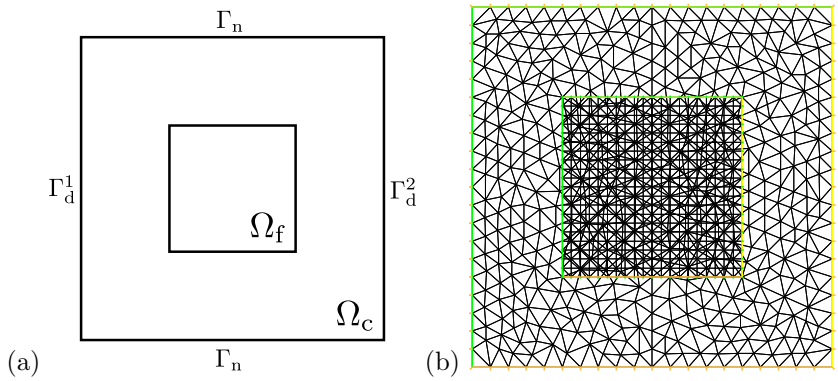


Figure 7: (a) Geometry for FEM domain Ω_c and LBM patch Ω_f. (b) Spatial discretization of the computational domain: coarse mesh for FEM and LB cells on a patch.

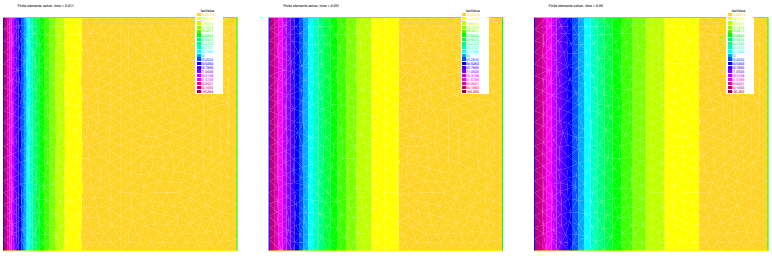
in Figures 8, 9 and 10. The converged coarse solution is reported in Figure 11 for the same timesteps. Note on Figure 9 the effect of the coupling due to the first prediction-correction iteration.

5.2. Convergence behavior for the FEM-LBM coupling

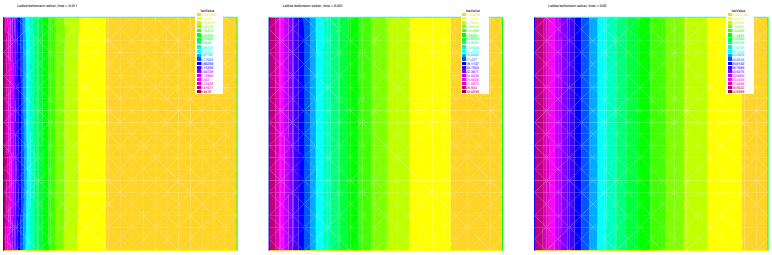
In this experiment, we solve the same physical problem considered in §5.1. For this test case the right end of time interval is $T = 0.04$. For the LBM, both order 0 (equation (18)) and 1 (equation (19)) formulas of Theorem 3.1 have been considered for the reconstruction of the distribution functions. 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 6 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 12. For a given 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 13. The global convergence behaviour 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 behaviour of the curves is slightly more complex.

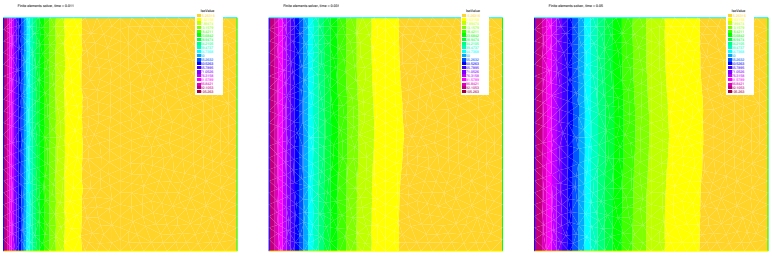


(a) Coarse, time = 0.011 (b) Coarse, time = 0.031 (c) Coarse, time = 0.05

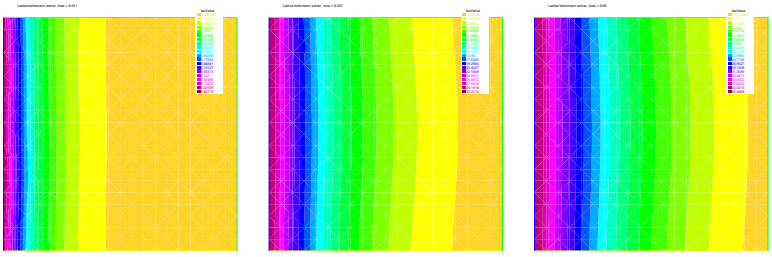


(d) Fine, time = 0.011 (e) Fine, time = 0.031 (f) Fine, time = 0.05

Figure 8: Coarse and fine solution at different timesteps for iteration 0 of the multiscale coupling.

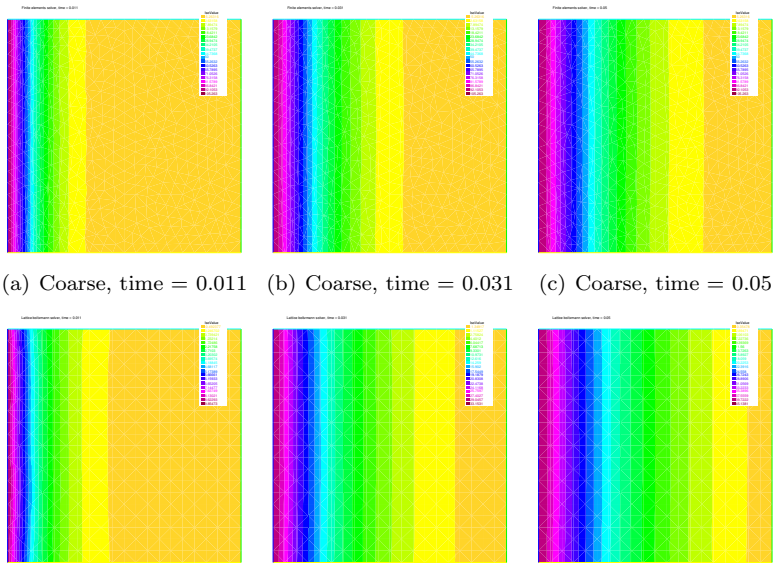


(a) Coarse, time = 0.011 (b) Coarse, time = 0.031 (c) Coarse, time = 0.05



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

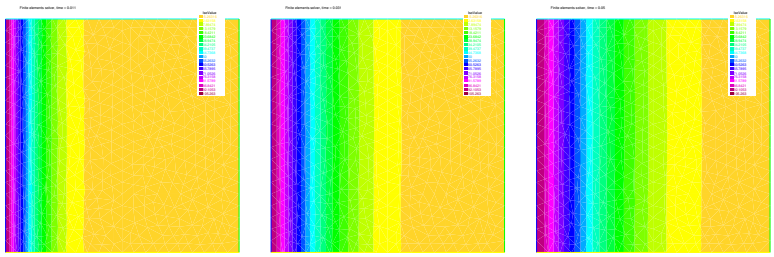
Figure 9: Coarse and fine solution at different timesteps for iteration 1 of the multiscale coupling.



(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 2 of the multiscale coupling.



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

Figure 11: Converged coarse solution at different timesteps.

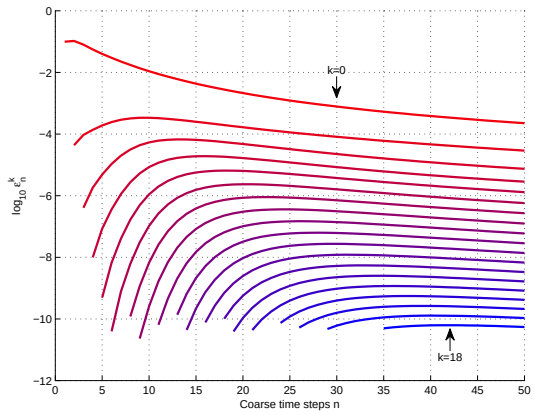


Figure 12: Order 0 reconstruction. The residuals ε_n^k (in log-scale) as a function of the coarse time-steps n for various values of k. Colors from red to blue correspond to increasing values of k.

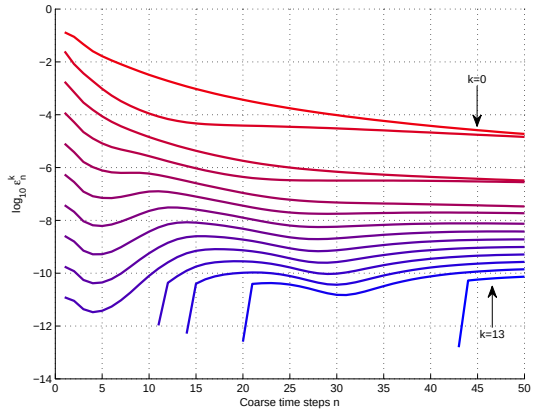


Figure 13: Order 1 reconstruction. The residuals ε_n^k (in log-scale) as a function of the coarse time-steps n for various values of k. Colors from red to blue correspond to increasing values of k.

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 14. 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 sufficent to obtain a good approximation of the final coupled solution.

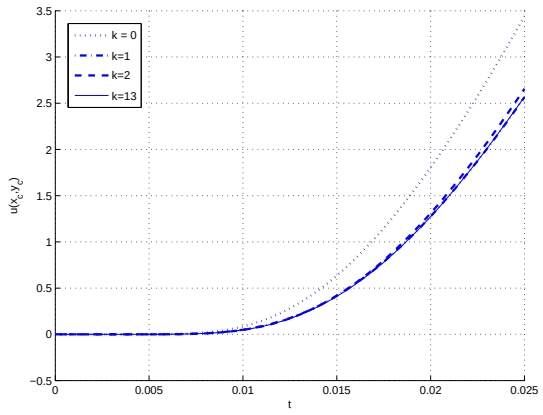


Figure 14: Evolution of the fine solution at the center of the domain, for various values of k.

5.3. Influence of discretization in space and time

In this section we study the influence of some important discrezation parameters, namely the ratio between the coarse and fine meshes $n_{\mathrm{c}}/n_{\mathrm{f}}$ and the ratio between the coarse and fine time-steps $p\ (= \Delta t/\delta t)$. We start from the same configuration as in the previous section 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 $n_{\mathrm{c}} = 20$. 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 ($p = 30$, $n_{\mathrm{f}} = 10$) 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 macro-meso coupling in a more complex computational problem. Let us consider the multiscale simulation of temperature evolution of a steel airfoil such as the one represented in Figure 15. The inner squared domain represents the region where the mesoscopic method is applied. The airfoil domain Ω is characterized by an initial temperature $u_0 = 298.15°\mathrm{K}$ and surrounded by air at $u_{\mathrm{s}} = 253.15°\mathrm{K}$ 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 ($\partial\Omega = \Gamma_{\mathrm{R}} \cup \Gamma_{\mathrm{D}}$). 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 $\alpha = 45\ \mathrm{W/m^2\,K}$ 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:

$$\begin{aligned} \rho c \frac{\partial u}{\partial t} - \kappa\, \Delta u = 0 \qquad & \text{in} \quad \Omega \times (0,T), \\ u(\cdot,0) = u_0 \qquad & \text{in} \quad \Omega, \\ \kappa \frac{\partial u}{\partial n} + \alpha u = \alpha u_s \qquad & \text{on} \quad \Gamma_{\mathrm{R}}, \\ u = u_d \qquad & \text{on} \quad \Gamma_{\mathrm{D}}. \end{aligned} \tag{39}$$

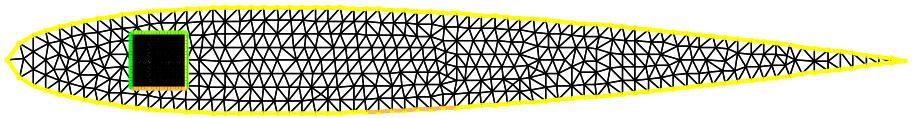


Figure 15: Computational domain for the airfoil test case.

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 16 and 17.

6. Discussion

6.1. Summary on the method and numerical experiments

We proposed an efficient multiscale method for time-dependent problems based on the coupling of FEM and LBM. Its efficiency derive mostly from the Parareal

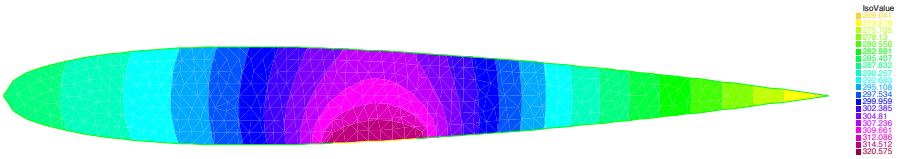


Figure 16: Final FEM solution (temperature in K) for the airfoil test case.

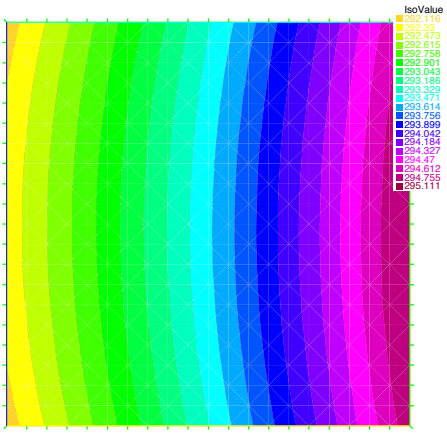


Figure 17: Final LBM solution (temperature in K) for the airfoil test case.

framework where the FEM solver is adopted as a coarse predictor (macro) for the LBM fine solver (meso). 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, naturally fits the mesoscopic scale. The FEM, strengthened by the variational framework, is extremely flexible for macroscopic problems characterized by complex geometries and boundary conditions. The Parareal framework for the multiscale coupling between FEM and LBM leads to a simple approach, very 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 one can use larger time-step if 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, the Chapman-Enskog expansion provides a satisfactory technique for the derivation of the transmission conditions.

In previous works the multiscale coupling is operational but it lacks of effective full overlap typically needed in multiscale problems (see numerical zoom).

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 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 very suitable for time-parallel time-integration of a wide class of evolution problems described by ODE's/PDE's [37, 17, 23]. It involves a fine propagator and a coarse propagator. The latter serves to predict quickly an inaccurate solution during the (sequential) correction iterations. An interesting feature is that the choice of this coarse propagator is left completely free to the user, it should only serve the purpose of providing a suitable 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. [6]), this possibility has not been so extensively explored until now, apart from a few exceptions. In [14], 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 [8] design a coarse propagator using a reduction method: a reduced set of chemical reactions is considered for the coarse prediction. At last, in reference [26] the possibility of using reduced basis methods for design of a cheapest coarse solver is explored. Our present work also goes into this direction and tends to confirm that the Parareal methodology is well suited for coupled multiscale 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 can be fitted for numerical zoom purposes. In its original version, the coarse and fine solvers must work on the same domain, whereas the Algorithm 6 allows a coarse solver on a larger domain.

6.3. Some perspectives

The model problem (1) in a two-dimensional domain has been chosen for simplicity reasons and in order to focus on methodological aspects. Of course, the proposed method can be extended quite easily to more general (linear) elliptic operators and to three-dimensional domains. It only requires an appropriate derivation of the lattice Boltzmann approximation and of associated transmission conditions.

In future works, application of the proposed method to more realistic problems may be considered: for instance reaction-diffusion equations in biological systems and Navier-Stokes equations in the context of blood flows [20]. Also, the method could be object of parallel implementation on GPU's architectures, following the work in [5].

Extension to other type of couplings may be also an interesting - and challenging - perspective: for instance in a recent work [1] a coupling between full Botzmann equations, BGK-ES equations and Euler equations has been realized using a domain decomposition framework.

Acknowledgements

The second author would like to thank particularly Miguel Fernández, Jean-Frédéric Gerbeau, Sidi Mahmoud Kaber and Yvon Maday for very encouraging and interesting discussions on the Parareal algorithm.

References

[1] A. Alaia, G. Puppo, A hybrid method for hydrodynamic-kinetic flow – Part II – Coupling of hydrodynamic and kinetic models, J. Comput. Phys. 231 (2012) 5217 – 5242.

[2] P. Albuquerque, D. Alemani, B. Chopard, 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, pp. 540–547.

[3] P. Albuquerque, D. Alemani, B. Chopard, P. Leone, A Hybrid Lattice Boltzmann Finite Difference Scheme for the Diffusion Equation, Int. J. Mult. Comp. Eng. 4 (2006) 209–219.

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

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

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

[7] G. Bird, Molecular Gas Dynamics and the Direct Simulation of Gas Flows, Claredon, Oxford, 1994.

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

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

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

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

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

[13] A. Dupuis, P. Koumoutsakos, Effects of Atomistic Domain Size on Hybrid Lattice Boltzmann-Molecular Dynamics Simulations of Dense Fluids, Int. J. Mod. Phys. C 18 (2007) 644–641.

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

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

[16] R. Eymard, T. Gallouët, R. Herbin, The finite volume method, in: P. Ciarlet, J. Lions (Eds.), Handbook of Numerical Analysis, volume VII, North-Holland, 2000, pp. 713–1020.

[17] C. Farhat, 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) 1397–1434.

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

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

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

[21] U. Frisch, D. d'Humières, B. Hasslacher, P. Lallemand, Y. Pomeau, J. Rivet, Lattice gas hydrodynamics in two and three dimensions, Complex Systems 1 (1987) 649–707.

[22] M. Fyta, S. Melchionna, E. Kaxiras, S. Succi, Multiscale Coupling of Molecular Dynamics and Hydrodynamics: Application to Translocation through a Nanopore, Multiscale Model. Simul. 5 (2006) 1156–1173.

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

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

[25] M. Griebel, S. Knapek, G. Zumbusch, Numerical Simulation in Molecular Dynamics, Springer, Berlin, 2007.

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

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

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

[29] A. Hiorth, U.H. a Lad, S. Evje, 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) 405–421.

[30] T.Y. Hou, X.H. Wu, A multiscale finite element method for elliptic problems in composite materials and porous media, Journal of Computational Physics 134 (1997) 169 – 189.

[31] H. Huang, X. Lu, M. Sukop, Numerical study of lattice boltzmann methods for a convection–diffusion equation coupled with navier–stokes equations, Journal of Physics A: Mathematical and Theoretical 44 (2011) 055001.

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

[33] M. Kojic, N. Filipovic, A. Tsuda, A mesoscopic bridging scale method for fluids and coupling dissipative particle dynamics with continuum finite element method, Comp. Meth. Appl. Mech. Eng. 197 (2008) 821–833.

[34] R. Kosloff, Time-dependent quantum-mechanical methods for molecular dynamics, The Journal of Physical Chemistry 92 (1988) 2087–2100.

[35] J. Lätt, Hydrodynamic limit of lattice Boltzmann equations, Ph.D. thesis, Geneva University, 2007.

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

[37] J.L. Lions, Y. Maday, G. Turinici, Résolution d'EDP par un schéma en temps "pararéel", Comptes Rendus de l'Académie des Sciences, Série I 332 (2001) 661–668.

[38] H.B. Luan, H. Xu, L. Chen, D.L. Sun, W.Q. Tao, Numerical Illustrations of the Coupling Between the Lattice Boltzmann Method and Finite-Type Macro-Numerical Methods, Numerical Heat Transfer, Part B: Fundamentals 57 (2010) 147–171.

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

[40] R. Mei, L.S. Luo, P. Lallemand, D. d'Humières, Consistent initial conditions for lattice Boltzmann simulations, Comp. Fluids 35 (2006) 855–862.

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

[42] E. Moeendarbary, T. Ng, M. Zangeneh, Dissipative particle dynamics: introduction, methodology and complex fluid applications – A review, Int. J. Appl. Mechanics 4 (2009) 737–764.

[43] G. Pavliotis, A. Stuart, Multiscale methods: averaging and homogenization, Springer Verlag, 2008.

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

[45] A. Quarteroni, A. Valli, Numerical Approximation of Partial Differential equations, Springer, 1997.

[46] G. Romano, Multiscale thermal models of nanostructured devices, Ph.D. thesis, Università degli Studi di Roma "Tor Vergata", 2010.

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

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

[49] H. Stetter, The Defect Correction Principle and Discretization Methods, Numer. Math. 29 (1978) 425–443.

[50] J. Strikwerda, Finite difference schemes and partial differential equations, Society for Industrial Mathematics, 2004.

[51] S. Succi, The Lattice Boltzmann Equation for Fluid Dynamics and Beyond, Oxford University Press, 2008.

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

[53] S. Succi, E. Weinan, E. Kaxiras, Lattice boltzmann methods for multiscale fluid problems, Handbook of Materials Modeling (2005) 2475–2486.

[54] S. Tang, T. Hou, W. Liu, A mathematical framework of the bridging scale method, Int. J. Numer. Method. Eng. 65 (2006) 1688–1713.

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

[56] G. Vidal, Efficient simulation of one-dimensional quantum many-body systems, Phys. Rev. Lett. 93 (2004) 040502.

[57] G. Wagner, W. Liu, Coupling of atomistic and continuum simulations using a bridging scale decomposition, J. Comput. Phys. 190 (2003) 249–274.

[58] E. Weinan, B. Engquist, Multiscale modeling and computation, Notices of the AMS 50 (2003) 1063–1070.

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

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

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

MOX Technical Reports, last issues

Dipartimento di Matematica "F. Brioschi",
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

47/2012 Astorino, M.; Chouly, F.; Quarteroni, A.
Multiscale coupling of finite element and lattice Boltzmann methods for time dependent problems

46/2012 Dassi, F.; Perotto, S.; Formaggia, L.; Ruffo, P.
Efficient geometric reconstruction of complex geological structures

45/2012 Negri, F.; Rozza, G.; Manzoni, A.; Quarteroni, A.
Reduced basis method for parametrized elliptic optimal control problems

44/2012 Fumagalli, A.; Scotti, A.
A numerical method for two-phase flow in fractured porous media with non-matching grids

43/2012 Secchi, P.; Vantini, S.; Vitelli, V.
A Case Study on Spatially Dependent Functional Data: the Analysis of Mobile Network Data for the Metropolitan Area of Milan

42/2012 Lassila, T.; Manzoni, A.; Quarteroni, A.; Rozza, G.
Generalized reduced basis methods and n width estimates for the approximation of the solution manifold of parametric PDEs

41/2012 Chen, P.; Quarteroni, A.; Rozza, G.
Comparison between reduced basis and stochastic collocation methods for elliptic problems

40/2012 Lombardi, M.; Parolini, N.; Quarteroni, A.
Radial basis functions for inter-grid interpolation and mesh motion in FSI problems

39/2012 Ieva, F.; Paganoni, A.M.; Ziller, S.
Operational risk management: a statistical perspective

38/2012 Antonietti, P.F.; Bigoni, N.; Verani, M.
Mimetic finite difference approximation of quasilinear elliptic problems