Scuola Normale Superiore
PISA

Department of Mathematics

Asymptotic Analysis of lattice Boltzmann method for Fluid-Structure interaction problems

ALFONSO CAIAZZO

Tesi di Perfezionamento in Matematica per la Tecnologia e l'Industria

Vom Fachbereich Mathematik
der Universität Kaiserslautern
zur Verleihung des akademischen Grades
Doktor der Naturwissenschaften
(**Dr. rer. nat.**) genehmigte Dissertation

REFEREES:
Prof. Dr. **Michael JUNK**
Prof. Dr. **Li-Shi LUO**
Prof. Dr. **Sauro SUCCI**
Prof. Dr. **Marcello ANILE**
Prof. Dr. **Axel KLAR**

Datum der Disputation: 06.02.2007

Acknowledgements

I am more than grateful to Prof. Michael Junk, with whom I learned a lot, and not only about the "Asymptotic Analysis" and the "lattice Boltzmann method". Thanks for all the time he spent with me in several discussions, held in different offices, different trains and through different telephones.

The work has been financially and technically supported by the *"Scuola Normale Superiore"* of Pisa, the *Technische Universität* of Kaiserslautern and the *Fraunhofer "Institut für Techno- und Wirtschaftsmathematik"* (ITWM) of Kaiserslautern. Particular thanks to Prof. Marcello Anile and Prof. Helmut Neunzert, who proposed me to complete the PhD in Germany, and to Prof. Axel Klar, who supervised me in Kaiserslautern.
My best gratitude goes also to the SKS department, in particular (in order of appearance) to Prof. Konrad Steiner, Dr. Dirk Kehrwald, and Dr. Guido Thömmes.

I would like to thank Prof. Li-Shi Luo, for having joint with me some interesting ideas and Prof. Sauro Succi, who helped me a lot at the beginning of the work. I thank also Martin Rheinländer and Zhaoxia Yang, for the cooperation and the useful discussion we had during these years.

However, the strongest support came from a lot of friends, whose complete list would probably be longer than the thesis itself. Special thanks to Anna and Monika, for the useful excursions far from work. I want to thank my parents, my sister and my family in Italy, for the logistic connections with Germany they set up for me, all the roommates and the friends I have met in Kaiserslautern, for everything we shared, Natalia, who has been my very personal sound track throughout the work, and the part of family I have in Colombia.

Abstract

The *lattice Boltzmann method* (LBM) is a numerical solver for the Navier-Stokes equations, based on an underlying molecular dynamic model. Recently, it has been extended towards the simulation of complex fluids.
We use the *asymptotic expansion technique* to investigate the standard scheme, the *initialization problem* and possible developments towards *moving boundary* and *fluid-structure interaction* problems. At the same time, it will be shown how the mathematical analysis can be used to understand and improve the algorithm.
First of all, we elaborate the tool "asymptotic analysis", proposing a general formulation of the technique and explaining the methods and the strategy we use for the investigation. A first standard application to the LBM is described, which leads to the approximation of the Navier-Stokes solution starting from the lattice Boltzmann equation.
As next, we extend the analysis to investigate origin and dynamics of initial layers. A class of *initialization algorithms* to generate accurate initial values within the LB framework is described in detail. Starting from existing routines, we will be able to improve the schemes in term of efficiency and accuracy.
Then we study the features of a simple *moving boundary* LBM. In particular, we concentrate on the initialization of new fluid nodes created by the variations of the computational fluid domain. An overview of existing possible choices is presented. Performing a careful analysis of the problem we propose a modified algorithm, which produces satisfactory results.
Finally, to set up an LBM for fluid structure interaction, efficient routines to evaluate forces are required. We describe the *Momentum Exchange algorithm* (MEA). Precise accuracy estimates are derived, and the analysis leads to the construction of an improved method to evaluate the interface stresses.
In conclusion, we test the defined code and validate the results of the analysis on several simple benchmarks.

From the theoretical point of view, in the thesis we have developed a general formulation of the asymptotic expansion, which is expected to offer a more flexible tool in the investigation of numerical methods. The main practical contribution offered by this work is the detailed analysis of the numerical method. It allows to understand and improve the algorithms, and construct new routines, which can be considered as starting points for future researches.

Table of Contents

Introduction

In recent years, the mathematical modeling of real-life problems has become a powerful tool in practical applications. The *Computational Fluid Dynamics* (CFD) is a subset of this branch, which groups a large variety of numerical methods, used to approximate the solution of systems involving fluid flows. Among them, we investigate the *Lattice Boltzmann method* (LBM).
The LBM was developed in the early 90s [2, 7, 33], as an evolution of the former rather inaccurate *Lattice Gas Cellular Automata* (LGCA) [13, 42]. The mathematical analysis of the LBM [18, 19] has shown its relation with the Navier-Stokes equations and its properties as approximate solver.
More precisely, the LBM can be considered as a *simple*, *low order accuracy* (first order in time, second order in space), *explicit integrator* of the Navier-Stokes equations. It is true that there are plenty of schemes, which can achieve more accurate results. Why then focus on the LBM?
The low accuracy is compensated by a favorable algorithmic formulation, which allows efficient implementations. Moreover, due to the existence of an underlying kinetic model, the LBM satisfies a set of symmetry properties and conservation constraints, which hold independently on the size of the discretization.

Scopes

The main aim of this thesis is to develop some new features of the LBM, towards the application to *moving boundary* (MB) and *fluid-structure interaction* (FSI) problems, preserving as much as possible the properties of the standard scheme.

Overview and main results

In **chapter 1**, the main background is described. Section 1.1 introduces the LBM, through short overview and derivation, also used to fix the notations.
In section 1.2 a general formulation of *asymptotic expansion* is proposed. First we deal with a formalization of the theoretical tools, setting up the background and the necessary ingredients. Starting from a general perturbed equation, we try to identify which elements are relevant in a general case and which steps can

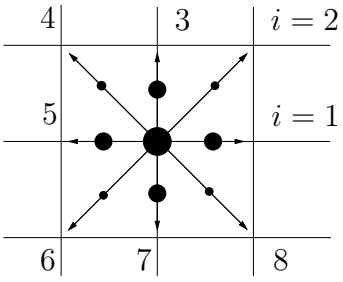


Figure 1: A two dimensional lattice Boltzmann discrete velocity model ($D2Q9$). From a kinetic scheme point of view, each node contains 9 particle distributions (one *rest* population and 8 moving populations).

be formalized on an abstract level. It will result in a *heuristic algorithm*, which describes the procedure used to perform the asymptotic analysis.
In section 1.3 the application of the procedure to the LBM is illustrated in detail, compared with the results of the regular analysis of the method [19]. In our intentions the goal of the analysis is to predict the behavior of the scheme. Calling $\hat{f}_h$ the numerical solution produced by the LBM on a regular grid of size h (small parameter), we consider an *ansatz*, i.e. a guess for a particular structure of $\hat{f}_h$. In the particular case of the LBM, we start with a power series in h,

$$\hat{f}_h = f^{(0)} + hf^{(1)} + h^2 f^{(2)} + \ldots,$$

where the coefficients $f^{(k)} = f^{(k)}(t, \mathbf{x})$ are functions of physical time and space. Inserting the ansatz into the algorithm, we are able to derive a prediction by expressing the coefficients $f^{(k)}$ in terms of Navier-Stokes solutions, which allows us to discuss the accuracy of the scheme (figure 2).

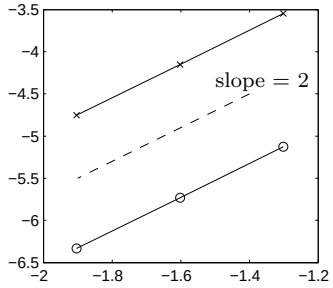


Figure 2: Double logarithmic plot of the maximum error in pressure (crosses $\times$) and velocity (circles $\circ$) versus grid size, in a particular LB simulation. The (averaged) slope of the curve gives the experimental order of accuracy. The dashed line indicates the slope predicted by the analysis.

In **chapter 2** we deal with initial layer phenomena, which are typical problems of LBM simulations. In such cases the prediction of the numerical results provided by the regular analysis is in general not satisfactory. We use more general expansions, where the coefficients depend on additional time variables.
In section 2.2 we define the *two-scales* expansion, introducing an additional faster time. With a more flexible ansatz, containing *three time scales* (section 2.3), we will be able to predict accurately the leading order of the oscillatory layers (figure 3). In conclusion, we relate the origin of an initial layer to an inaccurate setting of the *initial conditions*.

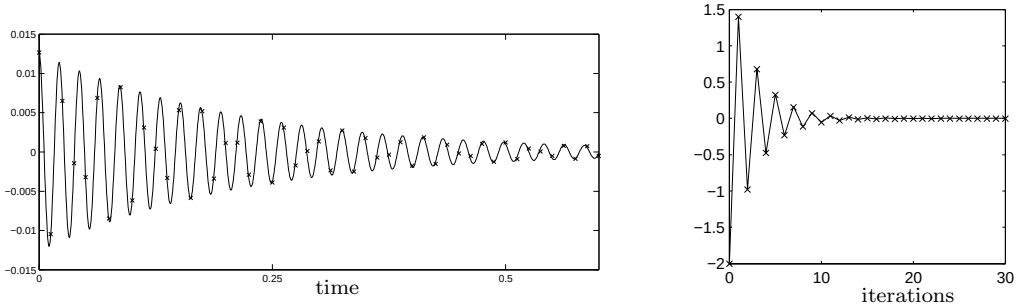


Figure 3: Prediction of the error simulating a particular Navier-Stokes solution with LBM. The continuous line is predicted by the analysis. The crosses ($\times$) indicate the numerical results. **Left:** Error in pressure. **Right:** Oscillations in the gradient of velocity.

Chapter 3 focuses on *LB-initializations*. To avoid initial layers, we consider some *pre-processors* to efficiently generate initial values for the LBM. Sections 3.2 and 3.3 describe some *initialization algorithms* [34]. First, for a simple periodic case, a detailed analysis is performed. After that, we discuss the construction of a general routine, suitable for boundary value problems. Employing these routines, we reduce the amplitude of initial layers (figure 4). Using the information obtained from the analysis, the *initialization routines* are improved both in accuracy and efficiency.

From **chapter 4**, we start to investigate a *moving boundary* version of the LBM (section 4.1), which has the following ingredients:

- standard LBM (fluid domain)
- boundary conditions
- force evaluation

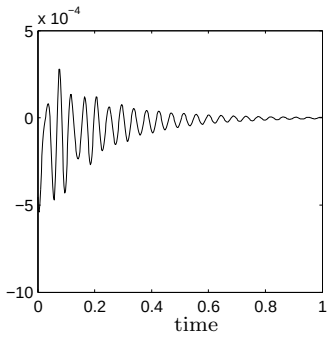


Figure 4: Same problem as in figure 3. The line shows the numerical error in pressure. Now, an initialization algorithm has been applied. The error has been reduced considerably.

- movement of the structure
- updating of the domains

In practice, the algorithm is tested on simple benchmarks. We use a family of problems based on the following model, which we call *cylinder-in-flow*. It consists of a disk (two dimensional section of a cylinder), surrounded by a fluid flow (figure 5). The ultimate task will be to simulate an interaction problem with the solid

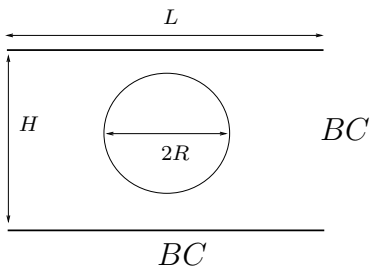


Figure 5: The *cylinder in flow* (CiF) benchmark. In a practical simulation, the parameters to be specified are: length (L) and width (H) of the domain, radius of the disk (R), boundary conditions, initial conditions for the fluid and the disk. The motion of the disk can be given (*moving boundary problem*) or determined by the interaction with the flow (*fluid-structure interaction*).

object moving through the fluid according to the force and the torque generated by the flow.
Step by step, we discuss and analyze the different aspects of the algorithm, starting from the *refill problem* (section 4.2). Since the fluid-solid boundary is moving through the fixed lattice, at certain time steps we will have to *initialize* the nodes entering the fluid domain from the solid one (figure 6). Using the hydrodynamic

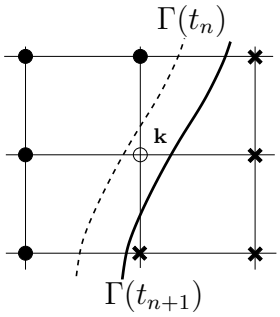


Figure 6: The movement of the boundary $\Gamma(t)$ through the grid creates and erases fluid nodes.

variables available at the neighboring fluid nodes and the data on the boundary, the task is to construct an initialization algorithm for a single node which does not spoil the accuracy of the standard scheme, and which is as stable and efficient as possible.
Existing algorithms are compared and analyzed, describing advantages and disadvantages. An improved routine is proposed, similar in spirit to the approach proposed in [16] to implement boundary conditions. With the help of the analysis, we construct a relatively simple approach, satisfactory in terms of accuracy, efficiency and suitability in connection with the LBM. The theoretical conclusions are validated on several numerical tests (figure 7 and 9).

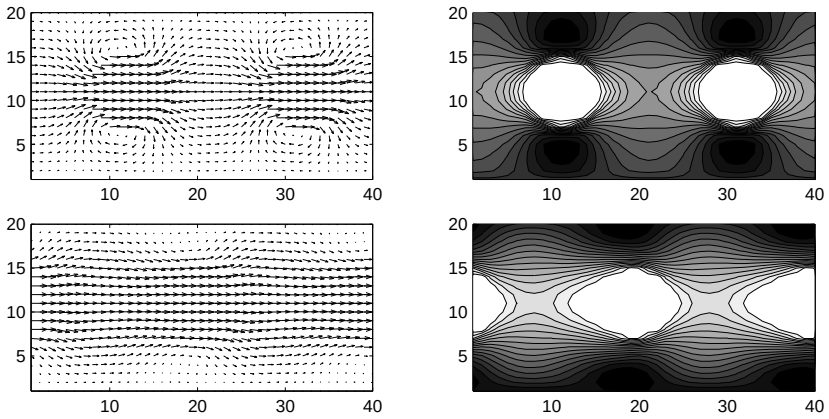


Figure 7: Examples of a flow around a structure. Simulation of (periodic) disks, moving horizontally in a channel. Arrows showing the velocity vectors and the corresponding contour lines of the horizontal velocity are drawn. On the top, the disks start to move, dragging the flow. At the bottom, the situation approaches a stationary flow, with a portion of space tending to a constant velocity.

To complete the investigation of *fluid-structure interactions*, we need a routine to couple the LBM with the evaluation of the interactions and a solver for the solid dynamic. **Chapter 5** deals with the evaluation of the interaction between fluid and solid object within the LBM.
In section 5.3 we describe, investigate and improve the *Momentum Exchange Algorithm* (MEA), proposed in [28].

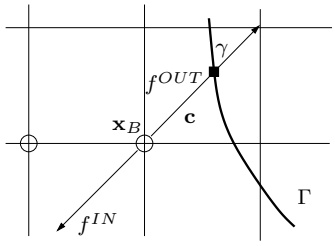


Figure 8: Sketch of the momentum exchange algorithm. For any fluid node $\mathbf{x}_B$, having a neighbor $\mathbf{x}_B + \mathbf{c}$ out of the fluid domain, the momentum exchanged with the boundary point γ is given by the difference of incoming and outgoing momenta in $\mathbf{x}_B$, along the direction $\mathbf{c}$. This quantity is considered as a force contribution acting on the solid body.

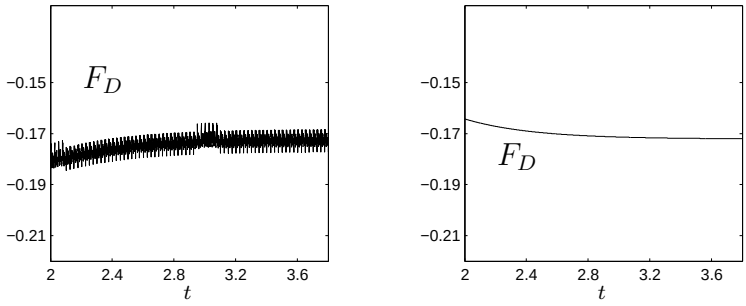


Figure 9: **Left:** Drag force $F_D(t)$ acting on a disk moving horizontally in a channel, approaching a stationary value. **Right:** Same situation, but simulated in a reference system moving with the disk. Now the walls move horizontally: this is a standard LB simulation, no special routines for moving boundaries are needed. The two tests produce a similar result for the stationary value (Galilean invariance in leading orders).

It was originally designed to be a simple method to evaluate the global force exerted on a solid body, as the sum of small contributions of momentum exchanged between the body and the fluid (figure 8). Performing an asymptotic analysis we introduce corrective terms, to improve the precision of the scheme. In this chapter a precise result concerning the accuracy of the algorithm for the evaluation of local and global forces is proven and validated on simple benchmarks. Namely,

we show that the scheme is inaccurate for the approximation of local interactions, although it is precise enough to approximate the global force.

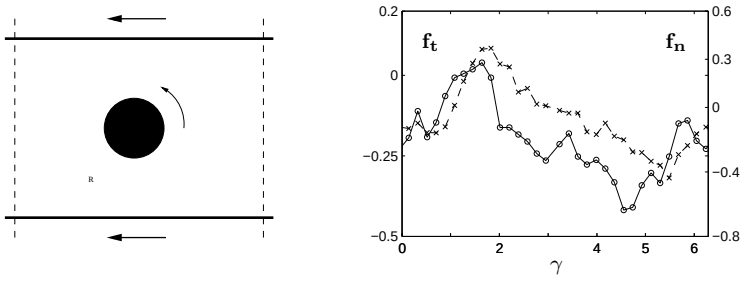


Figure 10: **Left:** Benchmark used to test the algorithm to evaluate the local forces. It consists of a rotating disk, between two plane, sliding walls. **Right:** Local forces, decomposed in tangential ($\mathbf{f_t}$ on the left y-axis, denoted with $\circ$) and normal component ($\mathbf{f_n}$ on the right y-axis, with $\times$). $\gamma \in [0, 2\pi)$ is the coordinate along the circle.

Hence, in section 5.4 we propose an improved algorithm to evaluate the local stresses, constructed starting from the results of the previous analysis. A satisfying accuracy (same as the accuracy of LBM) is achieved. Figure 10 shows a particular application.

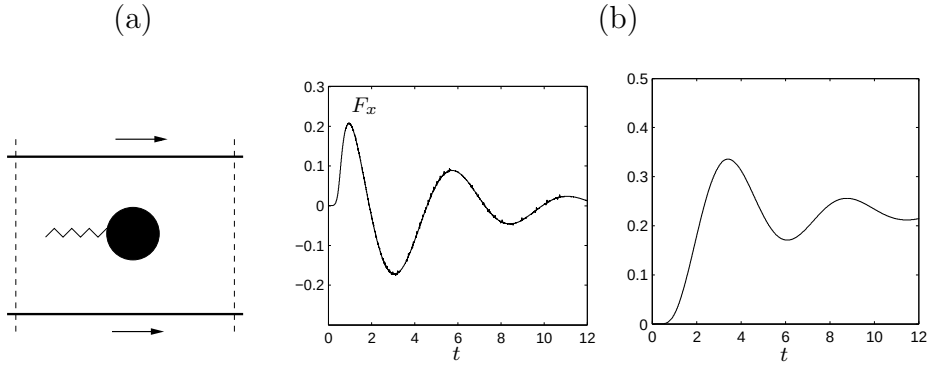


Figure 11: **(a)** Simple fluid-structure interaction model. The disk is placed between two sliding walls, and connected to a spring. **(b)** Result for the horizontal force on the disk, versus time (left) and displacement of the disk's center of mass, until an equilibrium position is reached (right).

In conclusion, we perform some numerical simulation of a simple *Fluid-Structure interaction* benchmark (some results are shown in figure 11), combining all the routines proposed and analyzed in the thesis.

Chapter 1

Asymptotic Analysis of the lattice Boltzmann Equation

We begin introducing the background, on which the rest of the thesis is built: the **asymptotic analysis** and the **lattice Boltzmann method**. Namely, this chapter contains the derivation of the numerical scheme and its analysis in the regular case. The final results of the analysis, that we recover here, have been published in the same form from M.Junk et al. in [19].

Our main task in this chapter is to set up a formulation of the asymptotic expansion technique based on a more formal point of view, to understand the procedure in depth. We attempt the construction of a *heuristic algorithm*, which can be used to perform the analysis in general situations.

In section 1.1 we provide an introductory overview of the lattice Boltzmann method (LBM) [13, 14, 33, 41, 42], deriving briefly the algorithm within the specific model we have used. As next, in section 1.2 we report the basic ideas of the asymptotic analysis [19, 27]. Besides we describe a generalized theory of the asymptotic expansion, while in section 1.3 we present the standard application to the LBM. The validity of theoretical results is discussed through some classical examples.

1.1 The Lattice Boltzmann Method

Among the numerical methods in computational fluid dynamics, the lattice Boltzmann schemes have the peculiarity of reproducing the dynamics of incompressible fluids, governed by the Navier-Stokes equations, by modeling the particle interactions on a mesoscopic scale. Historically, they originated as a development of the *Lattice Gas Cellular Automata* (LGCA) [13, 14, 33]. Later works have shown how they can be derived starting from a discretization of the Boltzmann Equation (BE) [17, 43]. However, from a mathematical point of view, the analysis presented in [18] recognizes the lattice Boltzmann models to be a particular class

of *finite differences schemes* for the Navier-Stokes equations.
We provide here a short derivation of the numerical scheme. For a detailed treatment, as well as overviews and classifications of the lattice Boltzmann schemes, we refer, for example, to [2, 7, 8, 10, 37, 41, 42].

1.1.1 Discrete kinetic framework

In the following, we consider the incompressible Navier-Stokes equation as a mathematical model to describe fluid flows on a domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$. The equations are of the form

$$\begin{cases} \nabla \cdot \mathbf{u} = 0 \\ \partial_t \mathbf{u} + \nabla p + \mathbf{u} \cdot \nabla \mathbf{u} = \nu \nabla^2 \mathbf{u} + \mathbf{G}, \qquad t > 0,\ \mathbf{x} \in \Omega \\ \mathbf{u}(0, \mathbf{x}) = \mathbf{u}_0(\mathbf{x}), \quad \mathbf{x} \in \Omega, \end{cases} \tag{1.1}$$

where $\mathbf{u}(t, \mathbf{x})$ represents the velocity of the flow field at time t and position $\mathbf{x}$, and $p(t, \mathbf{x})$ is the kinematic pressure, i.e. the dynamic pressure divided by the incompressible density, which is assumed to be equal to one. $\mathbf{G}$ is the volume force.
Within a kinetic model, a similar dynamics can be described by the *Boltzmann Equation* (BE)

$$\partial_t f(t, \mathbf{x}, \mathbf{v}) + \mathbf{v} \cdot \nabla_{\mathbf{x}} f(t, \mathbf{x}, \mathbf{v}) = J(f(t, \mathbf{x}, \mathbf{v})), \quad t > 0,\ \mathbf{x} \in \Omega, \mathbf{v} \in \mathbb{R}^d \tag{1.2}$$

where $f(t, \mathbf{x}, \mathbf{v})$ represents the probability of finding a particle at position $\mathbf{x}$ with velocity $\mathbf{v}$ at time t, while the *collision functional* $J(f)$ models the effects of collisions between particles.
The LBM discretizes the continuous problem in the framework of equation (1.2)[a].
The ingredients are a finite time step Δt and a *regular* (Cartesian) spatial grid

$$\mathcal{G}(\Delta x) \subset \mathbb{Z}^d, \tag{1.3}$$

whose nodes are defined by

$$\mathbf{j} \in \mathcal{G}(\Delta x) \overset{\text{def}}{\iff} \mathbf{x}_{\mathbf{j}} = \mathbf{j} \Delta x \in \Omega. \tag{1.4}$$

For the LBM, the two discretizations are related by the scaling

$$\Delta t = \Delta x^2. \tag{1.5}$$

[a]Historically, the LBM originated as a development of the former Lattice Gas Cellular Automata method [33, 41, 42]. However, we present here the shorter derivation obtained starting from the discrete Boltzmann Equation [17, 37, 41].

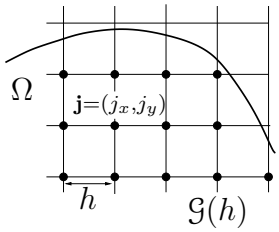


Figure 1.1: $\mathcal{G}(h)$ is a regular square grid, whose nodes are denoted with 2D integer vectors.

The implementation of the algorithm, for simplicity, is described using integer indices, n for time and $\mathbf{j}$ for space.
Regarding the velocity space, we choose a finite set

$$\mathbb{V} = \{\mathbf{c}_i \mid i = 1, \dots b\} \subset \mathbb{Z}^d, \tag{1.6}$$

which is symmetric

$$\mathbb{V} = -\mathbb{V}, \tag{1.7}$$

and has the property

$$\forall \mathbf{c}_i \in \mathbb{V}: \quad \mathcal{G}(\Delta x) + \mathbf{c}_i \subset \mathcal{G}(\Delta x) \tag{1.8}$$

i.e., the vectors $\mathbf{c}_i$ are node-to-node connections of the grid $\mathcal{G}(\Delta x)$, which has to be periodic, or to coincide with the whole space[b]. For square lattices, different velocity spaces $\mathbb{V}$ can be considered. We focus on the D2Q9 model in two dimensions (figure 1.2), with nine vectors (in lattice units)

$$\begin{array}{llll} \mathbf{c}_0 = (0,0), & & & \\ \mathbf{c}_1 = (1,0), & \mathbf{c}_2 = (1,1), & \mathbf{c}_3 = (0,1), & \mathbf{c}_4 = (-1,1) \\ \mathbf{c}_5 = (-1,0), & \mathbf{c}_6 = (-1,-1), & \mathbf{c}_7 = (0,-1), & \mathbf{c}_8 = (1,-1). \end{array} \tag{1.9}$$

Having defined the grid $\mathcal{G}(h)$ and the discrete velocities connecting the nodes, we introduce also

$$\overline{\mathcal{G}(h)} := \bigcup_{\mathbf{j} \in \mathcal{G}(h)} \left(\bigcup_{\mathbf{c}_i \in \mathbb{V}} \mathbf{j} + [0,1]\mathbf{c}_i \right) \cap \Omega, \tag{1.10}$$

i.e. the subset of Ω containing the lines (horizontal, vertical and diagonal) of the LB grid.
Approximating the advective derivatives

$$\frac{\partial}{\partial_t} + \mathbf{c}_i \cdot \nabla_{\mathbf{x}}$$

[b]To have simpler arguments, we restrict to the periodic case. Modifications regarding boundary value problems are described in section 1.1.3.

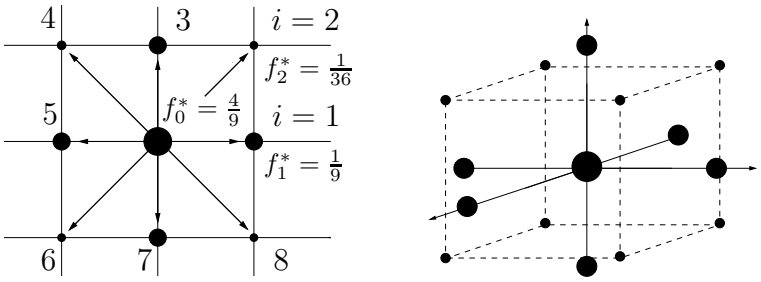


Figure 1.2: **Left**. Sketch of the lattice, the velocities $\mathbf{c}_i$, and the weights f_i^*, in the D2Q9 model. Bigger circles indicate larger weights. **Right.** A three-dimensional model (D3Q15). In both the cases, the velocity vectors connect only direct neighbors on the grid.

[17, 19, 41], we end up with the **lattice Boltzmann Equation** (LBE)

$$\hat{f}_i(n+1, \mathbf{j} + \mathbf{c}_i) = \hat{f}_i(n, \mathbf{j}) + J_i(\hat{f}(n, \mathbf{j})). \tag{1.11}$$

The functions[c] $\hat{f}_i : \mathbb{N} \times \mathcal{G}(h) \to \mathbb{R}$ indicate the numerical solutions for the density of particles associated with link $\mathbf{c}_i$ at the n-th iteration, and node $\mathbf{j}$. The collisions are modeled by the term $\hat{J}_i := J_i(\hat{f})$ which can be related, using suitable quadrature formulas with respect to the particular chosen space $\mathbb{V}$, to the collision integral in the Boltzmann equation (1.2) [17, 37, 41].
We call

$$\mathbb{D}_h := \mathbb{N} \times \mathcal{G}(h) \tag{1.12}$$

the domain of definition of the LB solutions $\hat{f}_i$. For shortness, we denote the set of functions between a set A and a set B as

$$\mathcal{F}(A, B) := \{\phi : A \to B\}. \tag{1.13}$$

So that

$$\hat{f}_{ih} \in \mathcal{F}(\mathbb{D}_h, \mathbb{R}).$$

In what follows, we will try to avoid the double subscript notation $\hat{f}_{ih}$, using, when possible, only $\hat{f}_i$ or $\hat{f}_h$.

1.1.2 Two-step dynamics and compact notation

In the practical implementation, the evolution step (1.11) is usually split in two sub-steps, the **local collision**:

$$\hat{f}_i^c(n, \mathbf{j}) = \hat{f}_i(n, \mathbf{j}) + J_i(\hat{f}(n, \mathbf{j})) \tag{1.14}$$

[c]With $\mathbb{N}$, we denote the set of all the natural numbers, including zero.

where $\hat{f}_i^c$ is the so called *post-collision* distribution, and the **advection**:

$$\hat{f}_i(n+1, \mathbf{j}+\mathbf{c}_i) = \hat{f}_i^c(n, \mathbf{j}). \tag{1.15}$$

In the LBE (1.11) $\hat{f}$ can be considered a vector in $\mathbb{R}^{b+1}$, whose components are the populations $\hat{f}_i$ (for $i = 0, \ldots, b$).
Introducing the notation

$$\hat{f}^n : \mathcal{G}(\Delta x) \to \mathbb{R}, \quad \hat{f}^n(\mathbf{j}) := \hat{f}(n, \mathbf{j}) \tag{1.16}$$

to indicate the numerical solution at a certain time step, and the grid-translation operator

$$S : f \mapsto Sf, \quad \text{with} \quad (Sf)_i(\mathbf{j}) = f_i(\mathbf{j}+\mathbf{c}_i), \tag{1.17}$$

equation (1.11) can be summarized as[d]

$$S\hat{f}^{n+1} = (I+J)(\hat{f}^n). \tag{1.18}$$

Finally, we introduce the *compact notation* for equation (1.18)

$$\forall (n, \mathbf{j}) \in \mathbb{D}_h : \mathrm{LBM}(h, \hat{f}_h)(n, \mathbf{j}) = 0, \tag{1.19}$$

where $\hat{f}_h \in X_h := \mathcal{F}(\mathbb{D}_h, \mathbb{R}^{b+1})$ and $\mathrm{LBM}(h, \cdot)$ is an operator

$$\mathrm{LBM}(h, \cdot) : X_h \to X_h, \tag{1.20}$$

which practically contains a set of rules to compute the function $\hat{f}_h$ on the discrete domain.
Within the vector notation, we can use the scalar product

$$\langle f, g \rangle = \sum_{i=0}^{b} f_i g_i, \quad f, g \in \mathbb{R}^{b+1}, \tag{1.21}$$

and the vectors

$$\mathbf{1} := (1, \ldots, 1) \in \mathbb{R}^{b+1},$$
$$\mathbf{c}_\alpha := (\mathbf{c}_{0\alpha}, \ldots, \mathbf{c}_{b\alpha}), \quad \alpha \in \{x, y\}$$

to abbreviate the moments taken with respect to the velocity vectors $\mathbf{c}_i$:

$$\begin{aligned} \rho(f) &:= \sum_{i=0}^{b} f_i = \langle f, \mathbf{1} \rangle, \\ \mathbf{u}_\alpha(f) &:= \sum_{i=0}^{b} \mathbf{c}_{i\alpha} f_i = \langle f, \mathbf{c}_\alpha \rangle. \end{aligned} \tag{1.22}$$

[d] On a periodic domain (satisfying (1.8)) the operator S is invertible:

$$S^{-1}(I + J_i)(\hat{f}^n).$$

scheme for advancing in time.

1.1.3 BGK approximation and basic algorithm

Different numerical methods are defined, specifying the form of J_i in equation (1.11). We choose an operator in a *linear relaxation* form,

$$J(f) = A\,(f^{eq} - f) + \mathbf{g}, \tag{1.23}$$

where A is a $b \times b$ positive semi definite matrix, with eigenvalues ω_i, which satisfy for stability reasons (for detailed theoretical and numerical treatment of stability, see [19, 25, 30])

$$0 < \omega_i < 2. \tag{1.24}$$

The additional vector $\mathbf{g} \in \mathbb{R}^{b+1}$ is used to include volume forces (see section 1.3). In (1.23) f_i^{eq} is the *equilibrium distribution*. Its form depends on the discrete velocity space. In general, it is a function of f through the **averaged density** ρ and the **averaged velocity u**, defined in equation (1.22):

$$f^{eq}(f) = H_i^{eq}(\rho(f), \mathbf{u}(f)). \tag{1.25}$$

For the D2Q9 model

$$H_i^{eq}(\rho, \mathbf{u}) = f_i^* \left(\rho + c_s^{-2} \mathbf{c}_i \cdot \mathbf{u} + \frac{c_s^{-4}}{2} \left(|\mathbf{c}_i \cdot \mathbf{u}|^2 - c_s^2 |\mathbf{u}|^2 \right) \right) \tag{1.26}$$

where $c_s = \frac{1}{\sqrt{3}}$ is the *lattice sound speed*, and

$$\begin{aligned} f_i^* &= \tfrac{4}{9}, \quad \text{for } i = 0 \\ f_i^* &= \tfrac{1}{9}, \quad \text{for } i = 1, 3, 5, 7 \\ f_i^* &= \tfrac{1}{36}, \quad \text{for } i = 2, 4, 6, 8. \end{aligned} \tag{1.27}$$

The function f_i^{eq} is chosen is such a way to satisfy a set of algebraic requirements and symmetries, in relation with the Maxwellian equilibrium of kinetic theory[e]. In the equilibrium H_i^{eq} we identify a linear and a quadratic part:

$$H_i^{eq}(\rho, \mathbf{u}) = H_i^{L(eq)}(\rho, \mathbf{u}) + H_i^{Q(eq)}(\mathbf{u}, \mathbf{u}). \tag{1.28}$$

Using the notation

$$(\mathbf{v} \otimes \mathbf{w})_{\alpha\beta} = \frac{1}{2} \left(v_\alpha w_\beta + v_\beta w_\alpha \right), \tag{1.29}$$

[e]More details can be found in [19, 41].

the following relations hold:

$$\begin{aligned}
&\sum_{i=1}^{b} H_i^{eq}(\rho, \mathbf{u}) = \sum_{i=1}^{b} H_i^{L(eq)}(\rho, \mathbf{u}) = \rho \\
&\sum_{i=1}^{b} \mathbf{c}_i H_i^{eq}(\rho, \mathbf{u}) = \sum_{i=1}^{b} \mathbf{c}_i H_i^{L(eq)}(\rho, \mathbf{u}) = \mathbf{u} \\
&\sum_{i=1}^{b} \mathbf{c}_i \otimes \mathbf{c}_i H_i^{L(eq)}(\rho, \mathbf{u}) = c_s^2 \rho \mathbf{I} \\
&\sum_{i=1}^{b} \mathbf{c}_i \otimes \mathbf{c}_i H_i^{Q(eq)}(\mathbf{u}, \mathbf{u}) = \mathbf{u} \otimes \mathbf{u}.
\end{aligned} \tag{1.30}$$

The first two identities in (1.30) are related to the conservation of mass and momentum. If the matrix A satisfies

$$\mathbf{1}^T A = 0$$

$$\mathbf{c}_\alpha^T A = 0$$

also the collision operator J (in the form (1.23)) conserves locally mass and momentum. Due to the properties of the equilibrium distribution, to have a momentum conserving collision operator it suffices that the vector $\mathbf{1}$ and $\mathbf{c}_\alpha$ are eigenvectors of the matrix A [19].
The most general form of the matrix A leads to the *Multiple Relaxation Time* model (MRT), proposed and investigated in the recent literature [11, 30]. Different relaxation parameters are defined for the f-moments of different orders. The simplest collision operator (*Single relaxation time*) is known as the *BGK approximation* (based on the homonym original model for the Boltzmann equation [3]). Formally, it is obtained by setting

$$A = \frac{1}{\tau}\mathbf{I}, \tag{1.31}$$

with one relaxation time $\tau > \dfrac{1}{2}$. We restrict to this case. An analysis of a wide class of linear collision operators, including also the MRT model, has been presented in [19]. Finally, the BGK-lattice Boltzmann method reads

$$\hat{f}_i(n+1, \mathbf{j} + \mathbf{c}_i) = \hat{f}_i(n, \mathbf{j}) + \frac{1}{\tau}\left(f_i^{eq}(\hat{f}) - \hat{f}_i\right)(n, \mathbf{j}). \tag{1.32}$$

Boundary conditions

The iteration (1.32) is well defined for each point of $\mathcal{G}(h)$ only if property (1.8) holds for the whole discrete lattice. In other words, if the domain Ω is periodic, or if it coincides with the whole space.

In presence of boundaries, some nodes will have one or more neighbors out of the domain (figure 1.3). For such points, the advection step cannot be defined as in (1.15). Additional algorithms, which incorporate the boundary conditions, are needed.
Formally, we define the *boundary set*

$$B = B(\Omega, h) := \{(\mathbf{k}, i) \in \mathcal{G}(h) \times \{1, \dots, b\} \mid \mathbf{x}_{\mathbf{k}+\mathbf{c}_i} \notin \Omega\} \tag{1.33}$$

containing *boundary couples*. Each couple identifies a node needing a different update rule[f]. For each $(\mathbf{k}, i) \in B$, we call the point $\mathbf{k}$ a *boundary node* and i an *outgoing direction* in $\mathbf{k}$. The direction i^*, such that $\mathbf{c}_{i^*} = -\mathbf{c}_i$ is called *incoming direction* at $\mathbf{k}$. To complete the algorithm (1.32), the distributions $f_{i^*}(n+1, \mathbf{k})$ have to be specified. We do not enter in details now. Some specific algorithms to include boundary conditions on pressure or velocity are described and analyzed in chapters 3 and 4, discussing their consistency within the method.
As notation, we write

$$\hat{f}_{i^*}(n+1, \mathbf{k}) = BC_{i^*}\left(\mathbf{d}^B, f^n, f^{n-1}, \dots; n, \mathbf{k}\right), \quad \text{for } (\mathbf{k}, i) \in B, \tag{1.34}$$

to indicate the boundary rule for the variable $\hat{f}_{i^*}$, as a functions of a certain set of data $\mathbf{d}^B$ (from the original Navier-Stokes problem) and the numerical solutions at the previous time steps.
Regarding initial conditions in time, we need an initialization rule:

$$\hat{f}(0, \mathbf{j}) = \mathcal{I}(\mathbf{d}^0; \mathbf{j}), \tag{1.35}$$

in accordance with certain initial data $\mathbf{d}^0$ provided by the original problem (1.1). Extending equation (1.19), we can use the general notation

$$\mathrm{LBM}(\mathbf{d}; h, \hat{f}_h) = 0, \tag{1.36}$$

which includes the set of rules for the interior domain, *plus* the additional ones (*initial value*, *boundary conditions*) and the data dependence.
A standard implementation of the lattice Boltzmann algorithm could be

Algorithm 1.1.
`set initial values:` ρ^0, $\mathbf{u}^0$, $\hat{f}_i(0, \mathbf{j})$,
  `DO` $n = 0, 1, \dots$
    `evaluate equilibrium` $\rightarrow H_i^{eq}(\rho^n, \mathbf{u}^n)$
    `collision step` $\rightarrow \hat{f}_i^c(n, \mathbf{j})$
    `IF (boundary_set` $\neq \emptyset$`)`
      { `boundary conditions` $\rightarrow \hat{f}_{i^*}(n+1, \mathbf{k})$ }
    `advection step` $\rightarrow \hat{f}_i(n+1, \mathbf{j})$
    `evaluate moments` $\rightarrow \rho^{n+1}, \mathbf{u}^{n+1}$
  `END`

[f] In general, for domains $\Omega(t)$ varying in time, we have to consider a time-depending set $B(\Omega(t_n), h)$ (section 4.1.2).

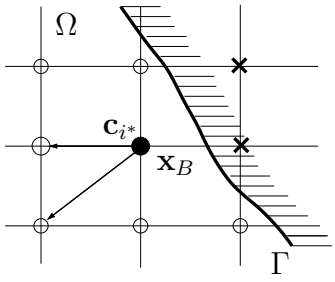


Figure 1.3: A boundary node is a lattice point $\mathbf{x}_B$ (•) with at least one neighbor $\mathbf{x}_B - h\mathbf{c}_{i^*}$ (×) out of the considered domain. In this case, $\mathbf{c}_{i^*}$ is an *incoming direction*, and the population f_{i^*} on $\mathbf{x}_B$ cannot be updated with the usual advection.

Properties of the method

Shortly, we summarize here the main characteristics of the LBM, and the advantages in using it in particular classes of problems.

Locality. In the implementation described before, the collision step (equation (1.14)) is the one which contains the biggest effort. Involving only variables at the node $(n, \mathbf{j})$, it is a completely local computation. The second step (advection) only performs an exchange of information between neighboring nodes.

Parallel computing. Therefore, the LBM is particularly suitable for parallel implementations. The structure of the scheme minimizes the need of *information exchange* between the processors, maximizing the *amount of operations* which has to be performed on a single computational node. The ratio between these two quantities expresses the *efficiency* of the method.

Symmetries The LBM is based on a simplified model of mesoscopic molecular dynamics. With respect to other numerical methods, it has a set of built-in properties, such as symmetries and conservation laws, which come from the underlying structure and which are satisfied independently of the size of the discretization.

Complex geometries. Of course, the concept of efficiency makes sense only compared with other approaches. In detail, due to its *mesoscopic formulation*, the LBM is used at its best in problems involving flows through complex geometries. In its simplest form, only a characterization of the nodes (*fluid-boundary*) is required.

Drawbacks. Despite of the listed qualities, the method is far from being optimal. Many points have still to be clarified and improved. Mainly, the property of allowing easy modeling of the geometry also leads to a low accuracy. In other words, to reach the accuracy provided by other schemes, very fine discretization (i.e. higher computational effort) is required.

Hence, the efficiency, presented as a merit, turns out to be a necessity, to make LBM competitive. One of our goals will be to improve and extend the LBM, preserving the advantages described above.

Main ideas of the expansion

As next, we study the behavior of the algorithm (1.32), deriving its relation with the system of partial differential equations (1.1), using methods of asymptotic analysis.
Practically, to investigate the scheme, we look for a representation of the LB solution in form of a power series

$$\hat{f}_h \approx f^{(0)} + hf^{(1)} + h^2 f^{(2)} + \dots.$$

Determining the coefficients $f^{(k)}$ order by order, we can characterize the scheme and predict the results. Such predictions are also used to improve the algorithm if its behavior is not satisfactory.

1.2 Asymptotic Analysis

We first introduce the topic in a general form, trying to develop a practical tool to be used to investigate a numerical scheme in general.
Later on, we focus on the particular application to the lattice Boltzmann algorithm.

1.2.1 Formal asymptotic: general background

The asymptotic analysis is a widely used tool, to investigate *equations depending on a small parameter* $h \ll 1$ [27].
Let us consider some simple examples. Typically, we can deal with **perturbed systems** of (ordinary or partial) *differential equations*, e.g.

$$x'' - hx' + x = f(x,t), \quad x \in C^2([0,T],\mathbb{R}), \tag{1.37}$$

or *algebraic equations*,

$$x^3 + hx^2 - x = 0, \quad x \in \mathbb{C}. \tag{1.38}$$

Besides, the technique has been also successfully applied to study the behavior of **numerical methods**, in which the small parameter is related to a certain *discretization*. For instance, let us consider the *explicit Euler scheme*:

$$\hat{x}_h(n+1) - \hat{x}_h(n) = hf(\hat{x}_h(n)), \quad n \in \mathcal{G}(h) \tag{1.39}$$

where $\mathcal{G}(h) = \{0, \dots, N\}$ are the indices of a partition $t_n = hn$ of an interval $[0, T] \subset \mathbb{R}$ and $f : \mathbb{R} \to \mathbb{R}$ is sufficiently regular. In this case, the solution is a *grid function*

$$\hat{x}_h : \mathcal{G}(h) \to \mathbb{R}.$$

Using the notation introduced in (1.13), we have $\hat{x}_h \in \mathcal{F}(\mathcal{G}(h), \mathbb{R})$, for the fixed $h \leq 1$. Compared with the previous class of problems, now the solution space

$$X_h = \mathcal{F}(\mathcal{G}(h), \mathbb{R})$$

itself depends on the small parameter.
In general, given a set $H \subset (0, 1]$ with *an accumulation point at* 0 (set of small parameters)[g] and a family of *normed spaces* $\{X_h\}_{h \in H}$, we can consider the sequence of problems

$$L(h, \hat{\mathbf{x}}_h) = 0, \quad h \in H, \tag{1.40}$$

expressing equations for the operators

$$L_h := L(h, \cdot) \, : \, X_h \to X_h,$$

with solutions $\hat{\mathbf{x}}_h \in X_h$.
As already indicated in equation (1.20) we are going to consider the LBM in this formulation.

Underlying framework

The starting point is now equation (1.40). Let be given

- the set $H \subset (0, 1]$ (with an accumulation point at 0),
- a family of *linear* and *normed* spaces $\mathcal{X} = \{(X_h, \| \|_{X_h}) \mid h \in H\}$,
- a family of operators $\mathcal{L} = \{L_h : X_h \to X_h \mid h \in H\}$, such that, for all $h \in H$, there exists $\hat{\mathbf{x}}_h$ such that $L_h(\hat{\mathbf{x}}_h) = 0$.

We call L_h the **algorithm on** X_h. The elements $\mathbf{x}_h$ which satisfy $L_h(\mathbf{x}_h) = 0$, are called **solutions** to L_h. The problem admits an elegant and compact formulation, introducing the set

$$X := \prod_{h \in H} X_h. \tag{1.41}$$

Remark. If, as in the examples (1.37), (1.38), $X_h = Y$, for all $h \in H$, then

$$X = \mathcal{F}(H, Y).$$

X inherits the properties of the family $\mathcal{X}$.

[g] The accumulation point is needed for practical reasons, if we aim at the limit $h \to 0$.

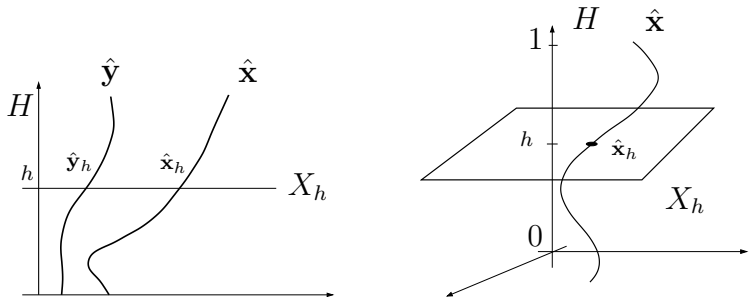


Figure 1.4: **Left:** Sketch of the space X, in the case $X_h = \mathbb{R}$. An element $\hat{\mathbf{x}} \in X$ is a curve in $\mathbb{R}^2$. **Right:** The case $X_h = \mathbb{R}^2$.

Lemma 1.1. *X is a* normed linear *space.*

Proof. To prove the lemma we introduce the *absolute value* of $\mathbf{x} \in X$:

$$|\mathbf{x}|(h) := \|\mathbf{x}(h)\|_{X_h} \in \mathbb{R}_+, \quad h \in H \tag{1.42}$$

(where $\mathbf{x}(h) = \mathbf{x}_h$). The linearity follows directly from the linearity of the components X_h. To conclude, using the definition in equation (1.42), it is easy to show that

$$\|\mathbf{x}\|_\infty := \sup_{h\in H} |\mathbf{x}|(h) \tag{1.43}$$

is a norm on X.
In fact, $\|\mathbf{x}\|_\infty = 0$ implies

$$\forall h \in H : \ \|\mathbf{x}(h)\|_{X_h} = 0$$

hence $\mathbf{x}(h) = 0$, for all $h \in H$. Then,

$$\|\lambda\mathbf{x}\|_\infty = \sup_{h\in H} \|\lambda\mathbf{x}(h)\|_{X_h} = |\lambda| \sup_{h\in H} \|\mathbf{x}(h)\|_{X_h} = |\lambda|\|\mathbf{x}\|_\infty.$$

For the triangular inequality, we have

$$\|\mathbf{x}+\mathbf{y}\|_\infty = \sup_h \|\mathbf{x}(h)+\mathbf{y}(h)\|_{X_h} \le \sup_h \|\mathbf{x}(h)\|_{X_h} + \sup_h \|\mathbf{y}(h)\|_{X_h}.$$

□

Remark. The norm $\|\mathbf{x}\|_\infty$ induces a topology on X, which is equivalent to the product topology of the spaces X_h.

1.2.2 Asymptotic analysis as optimization problem

Ansatz and approximate solutions

Having introduced the space X, we define the **algorithm** on X as the operator

$$\begin{array}{rccl} L : X & \to & X & \\ \mathbf{x} & \mapsto & L[\mathbf{x}], & \text{with } L[\mathbf{x}](h) := L(h, \mathbf{x}_h). \end{array} \tag{1.44}$$

An **exact solution** to L is an element $\hat{\mathbf{x}} \in X$ such that $L[\hat{\mathbf{x}}] = 0$, i.e. a solution for all $h \in H$.
To find an exact solution will not always be our goal. For example, assume that L is a numerical method, as the LBM in (1.32). In this case, for any *fixed* $h \in H$, we can compute an element $\hat{\mathbf{x}}_h \in X_h$, such that

$$L(h, \hat{\mathbf{x}}_h) = 0,$$

simply running the algorithm and taking the output.
The algorithm L can be seen as a "black box" which, given a specific value of the parameter h, produces the corresponding projection on X_h of the exact solution

$$h \xrightarrow{L} \hat{\mathbf{x}}_h : \; L(h, \hat{\mathbf{x}}_h) = 0.$$

Executing the algorithm we can have a *sample* of the solution through the family $\mathcal{X}$. In this case, our scope will rather be to *understand* the behavior of the solution $\hat{\mathbf{x}}_h$ for small values of h.
To do this, we look for an approximation of $\hat{\mathbf{x}}$ which depends only on *few* parameters:

$$\hat{\mathbf{x}}_h \approx F_h = F(h; \alpha_0, \dots, \alpha_M). \tag{1.45}$$

Formally, to perform the analysis we restrict to a subset $\mathcal{A} \subset X$, called **ansatz**. Within the ansatz we try to derive an easy representation of the numerical solution. The elements $F \in \mathcal{A}$ will be called **predictions**. To specify a prediction in $\mathcal{A}$ the finite set of free parameters has to be fixed. Summarizing, the scope of the asymptotic analysis is to find an *approximation of an exact solution to L for small h, in a particular class $\mathcal{A} \subset X$ which satisfies a set of constraints*, and whose elements have only a small number of degrees of freedom (figure 1.5).
The meaning of the symbol "$\approx$" in (1.45) has to be specified. To fix the ideas, let us consider a particular case, with the algorithm L **invertible** and with **inverse** L^{-1} **Lipschitz**. For $F \in X$, we can estimate

$$|F - \hat{\mathbf{x}}| = |L^{-1}L[F] - L^{-1}L[\hat{\mathbf{x}}]| \leq C_{L^{-1}}|L[F] - L[\hat{\mathbf{x}}]| = C_{L^{-1}}|L[F]|. \tag{1.46}$$

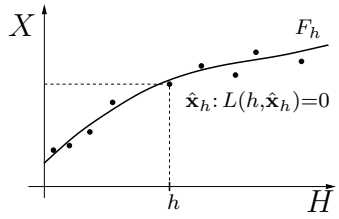


Figure 1.5: The philosophy of the asymptotic analysis. The circles • represent the projections $\hat{\mathbf{x}}_h$ of the solution to the algorithm L, obtained *experimentally* (running the algorithm) for specific values of $h \in H$. Aim of the analysis is to find a *good* prediction F_h, only depending on few parameters, able to approximate the exact solution for small h.

In this case[h]

$$|L[F]| \to 0 \Rightarrow |F - \hat{\mathbf{x}}| \to 0,$$

i.e. the element F approximates the solution $\hat{\mathbf{x}}$ for $h \in H$ small. We consider the following task:

- **Asymptotic solution to** L: find $F \in \mathcal{A}$ such that

$$\lim_{h\to 0} \|L(h, F(h))\|_{X_h} = 0. \tag{1.47}$$

Having in mind the estimate (1.46), the *smallness of the residue* $L[F]$ will be our indicator for the goodness of an approximation of the solution.

Example 1.1. *Suppose we want to understand the* h*-dependence of the solution* $\hat{x}_h$ *to*

$$L(h, x_h) := d^2 x_h^2 + h x_h - 1 = 0, \tag{1.48}$$

for some fixed $d \in \mathbb{C}$, $d \neq 0$.
Considering the elements F *and* G *such that*

$$\forall h \in H : \; F_h = \frac{1}{d}, \quad G_h = \frac{1}{d} - \frac{h}{2d^2},$$

we obtain

$$L(h, F_h) = d^2 \left(\frac{1}{d^2}\right) + \frac{h}{d} - 1 = \frac{h}{d},$$

[h]**Warning**. For a function F such that $|L[F]|(h) \to 0$, it is not true, in general, that

$$|F - \hat{\mathbf{x}}| \to 0,$$

$\hat{\mathbf{x}}$ being such that $L[\mathbf{x}] = 0$. It makes sense to discuss the implication only within a specific problem, knowing more in detail the properties of the operator L.

$$L(h, G_h) = d^2 \left(\frac{1}{d^2} + \frac{h^2}{4d^4} - \frac{h}{d^3} \right) + \frac{h}{d} - \frac{h^2}{2d^2} - 1 = -\frac{h^2}{4d^2}.$$

Hence, the residues $L[F]$ *and* $L[G]$ *are decreasing functions, for* $h \to 0$.
For the problem (1.48) *the exact solution can be analytically computed as*

$$\hat{x}_h^{\pm} = -\frac{h}{2d^2} \pm \frac{\sqrt{h^2 + 4d^2}}{2d^2} = -\frac{h}{2d^2} \pm \frac{1}{d}\sqrt{1 + \frac{h^2}{4d^2}}.$$

Actually, in this case F *and* G *are approximation of* $\hat{x}_h^+$.

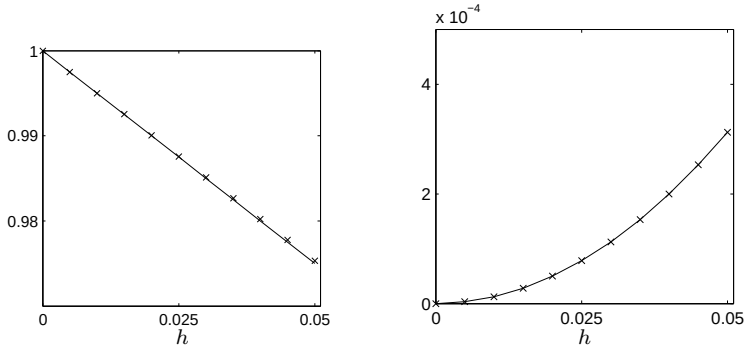


Figure 1.6: **Left:** Plot of the solution $\hat{x}_h^+$ of equation (1.48) (crosses, $\times$) and of the prediction G_h (solid line), for $h \leq 0.05$. **Right:** The difference $\left|\hat{x}_h^+ - G_h\right|$.

Problem data Generally a perturbed equation depends also on particular *data*, fixed by the specific problem. For example, one can think of the boundary condition of a differential problem, or of equations including some parameters, like the previous example (1.48). A proper notation would then be

$$L(d; h, \mathbf{x}_h) = 0.$$

We consider the analysis for *fixed data*. Thus, the dependence of L on d is not explicitly written, and the data is considered as a part of L itself.

The minimization procedure

Once we have fixed the background, we describe the strategy of investigation, explaining informally the main ideas of the approach we will follow in applying the asymptotic analysis.
The situation is the following. We have an algorithm L, i.e. an equation depending on a small parameter h. Through L we can derive the solution for particular values of h. However, we want to understand the behavior of the algorithm for small h. Within a suitable subset $\mathcal{A} \subset X$, called *ansatz*, we look only for an *approximation* of the solution, in the sense discussed above.

As shown in the previous example 1.1, there may be approximations of different accuracy. Naturally, our goal is to find the *best approximation* in $\mathcal{A}$. Of course, the notion of "best" requires an *order* on $\mathcal{A}$. The first task of the next section will be to introduce a relation $\prec$ on $\mathcal{A}$, so that, for $F, G \in \mathcal{A}$, $F \prec G$ means *"F is better than G"*. It can be done in a natural way (see example 1.1), based on the order of the residue.
In this framework, we can consider the asymptotic analysis as an *optimization problem*, where we have to *find the minimum in the set $\mathcal{A} \subset X$ with respect to the order $\prec$*.
Let us assume to have an initial ansatz $\mathcal{A}$, *totally ordered* with respect to $\prec$. We consider the following procedure to find the optimum.

Iterative optimization algorithm:

$$\begin{aligned} &\mathcal{A}^{(0)} = \mathcal{A} \\ &\texttt{for } k \geq 0 \\ &\quad \texttt{set } \mathcal{A}^{(k+1)} = \left\{ F \in \mathcal{A}^{(k)} \mid \exists G \in \mathcal{A}^{(k)} : \; F \prec G \right\} \\ &\texttt{repeat until } \mathcal{A}^{(k+1)} = \emptyset \end{aligned} \tag{1.49}$$

The idea is to exclude step by step some elements, to find the optimal one or to get closer to it.

1.2.3 Ordering the approximations

In the usual applications there is a natural way to define if an asymptotic solution is "better" than another (see also example (1.48)). In fact, $|L[F]|(h)$ is smaller than $|L[G]|(h)$ for small h if it converges *faster* to zero. Besides the property $|L(F)| \to 0$, the *speed of convergence* is also important. We classify the elements of X in that sense.

Definition 1.1. *Let $\omega \in \mathcal{F}(H, \mathbb{R}_+)$[i]. We say that the property $A(\omega)$ holds* **eventually** *(abbreviated* **evt.***) iff $\exists \bar{h} > 0$ such that $A(\omega|_{H \cap (0,\bar{h})})$ is true.*
For example, $\omega \leq K$ eventually, iff

$$\exists \bar{h} > 0 : \; \forall h \in H, h < \bar{h} : \omega(h) \leq K.$$

Lemma 1.2. *The conjunction of a finite number of eventually true statements, is also eventually true.*

[i] In the following, we use the notation

$$\mathbb{R}_+ = \{x \in \mathbb{R} \mid x \geq 0\}.$$

Proof. Let $N \in \mathbb{N}$, $N > 0$. Let $\omega \in \mathcal{F}(H, \mathbb{R}_+)$, such that $A_1(\omega), \ldots, A_N(\omega)$ hold eventually. Then, there exist $\bar{h}_1, \ldots, \bar{h}_N$ such that

$$\forall i \leq N : \; A_i \left(\omega|_{H\cap(0,\bar{h}_i)}\right) \text{ is true.}$$

Therefore, taking $\bar{h} = \min\{\bar{h}_1, \ldots, \bar{h}_N\}$,

$$(A_1 \wedge \ldots A_N)(\omega|_{H\cap(0,\bar{h})})$$

is true. □

Order comparisons (*Landau* symbols) Let $f, g \in \mathcal{F}(H, \mathbb{R}_+)$ be two real positive functions. We define [27]

$$f \in O(g), \text{ for } h \to 0 \overset{\text{def}}{\iff} \exists K : f \leq Kg \text{ evt.} \tag{1.50}$$

(f is *of the same order of* g). Additionally,

$$f \in o(g), \text{ for } h \to 0 \overset{\text{def}}{\iff} \forall \delta > 0 : f < \delta g \text{ evt.} \tag{1.51}$$

(f is *of order higher than* g).
The above conditions are standard notations in performing asymptotic investigations. They can easily be verified in some special cases (see [27]).

Lemma 1.3. *Let $f, g \in \mathcal{F}(H, \mathbb{R}_+)$, and assume $g > 0$ eventually. We have*

$$f \in o(g) \iff \limsup_{h\to 0} \frac{f(h)}{g(h)} = 0.$$

$$f \in O(g) \iff \exists C \in (0, +\infty) : \frac{f}{g} \leq C \iff \limsup_{h\to 0} \frac{f}{g} < \infty.$$

The proof is omitted.
This concept of order can be extended to the normed space X. Let $F, G \in X$. Using the absolute value $|F|$ defined in equation (1.42):

$$F \in O(G) \overset{\text{def}}{\iff} |F| \in O(|G|) \tag{1.52}$$

(and similarly for $F \in o(G)$).
After having introduced the standard background and concepts, we look for a criterion to order the elements of X, according to an algorithm L.

Definition 1.2 (Precision order). *Let $F, G \in X$. We say that F is more precise than G for L (denoted $F \prec_L G$) iff*

$$L[F] \in o(L[G]).$$

Remark. Assume we have two algorithms L and M defined on the same space X, such that

$$\forall h \in H : \; M_h = h^2 L_h.$$

Of course, any solution to L is also solution to M. In some sense, we can say that the two algorithm are *equivalent*. The definition 1.2 is constructed in such a way to be *invariant* with respect to such equivalence of algorithm. In fact,

$$F \prec_L G \iff F \prec_M G.$$

Note that, in contrast to this, notions depending explicitly on the order of the residue (like the *order of consistency*) are not invariant with respect to this equivalence.
In the example

$$d^2 x^2 + hx - 1 = 0, \tag{1.53}$$

considering the elements

$$G_h = \frac{1}{d} - \frac{h}{2d^2}, \quad F_h = \frac{1}{d}$$

we can say $G \prec_L F$, since

$$\lim_{h \to 0} \frac{|L[G]|}{|L[F]|} = 0.$$

As next, we analyze the properties of the defined order.

Lemma 1.4. *Let $E, F, G \in X$.*

$$\textbf{(a)} \quad E \prec_L F, \; F \prec_L G \quad \Longrightarrow E \prec_L G,$$

$$\textbf{(b)} \quad F \not\prec_L F.$$

Proof. Both the statements are direct consequences of definition (1.51). Concerning **(a)**, the hypothesis means

$$\forall \delta > 0 : \; |L[E]| < \delta |L[F]| \text{ evt.} \tag{1.54}$$

Analogously,

$$\forall \delta > 0 : \; |L[F]| < \delta |L[G]| \text{ evt.} \tag{1.55}$$

Hence, for any $\epsilon > 0$, it holds, choosing $\delta = \sqrt{\epsilon}$ in (1.54)-(1.55):

$$|L[E]| < \epsilon |L[G]| \text{ evt.}$$

which proves the statement.
Regarding **(b)**, we have to prove

$$L[F] \notin o(L[F]).$$

We assume the negation,

$$L[F] \in o(L[F]).$$

According to the definition, it means

$$\forall \delta > 0 : F < \delta F \text{ evt.},$$

but for $\delta = 1$ we have an absurd statement.
One remark is perhaps worth to do: despite the intuitive meaning of the relation $\prec_L$ which makes the statement **(b)** look obvious, we note that a slightly different definition, namely

$$f \in o(g), \quad \text{for } h \to 0 \;\Leftrightarrow\; \forall \delta > 0 : \; f \leq \delta g \text{ evt.} \tag{1.56}$$

(where the "$\leq$" appears on the right side, instead of "$<$") would not allow to conclude $F \not\prec_L F$. In particular, a solution $\hat{\mathbf{x}}$ to L would satisfy $\hat{\mathbf{x}} \prec_L \hat{\mathbf{x}}$, because $L[\hat{\mathbf{x}}] = 0$. □

The previous lemma proves the property of *anti-reflexivity* and *transitivity* of the relation $\prec_L$ on X. In other words,

Proposition 1.1. *The relation $\prec_L$ defines a partial ordering on X.*

However, we need a total ordering, which is not assured. As an example, let us consider again the equation

$$d^2x^2 + hx - 1 = 0, \tag{1.57}$$

and the elements

$$F'_h = \frac{1}{d} - h, \quad G'_h = \frac{1}{d} + h.$$

We have

$$L(h, F'_h) = h\left(-2d + \frac{1}{d}\right) + O(h^2), \quad L(h, G'_h) = h\left(2d + \frac{1}{d}\right) + O(h^2).$$

Therefore, neither $F' \prec_L G'$ nor $G' \prec_L F'$. In this case, the residues of F' and G' are of the same order (unless $d = \frac{1}{\sqrt{2}}$).

Definition 1.3. *Let $F, G \in X$. We say that F is as precise as G for L (denoted $F =_L G$) iff*

$$L[F] \in O(L[G]), \quad L[G] \in O(L[F]).$$

Using the definition in equation (1.50), we can prove the following

Lemma 1.5. *Let $F, G \in X$. Then, $F =_L G$ iff*

$$\exists K_1, K_2 > 0 : \; K_1^{-1}|L[F]| \leq |L[G]| \leq K_2|L[F]| \text{ evt.}$$

Proof. The conditions $L[F] \in O(L[G])$, $L[G] \in O(L[F])$ are equivalent to

$$(\mathbf{a})\exists K_1 > 0 : \ |L[F]| \leq K_1 |L[G]| \text{ evt.} \tag{1.58}$$
$$(\mathbf{b})\exists K_2 > 0 : \ |L[G]| \leq K_2 |L[F]| \text{ evt.} \tag{1.59}$$

Which proves the statement, because of lemma 1.2 (combination of a finite number of *eventually true* statement). □

Classes of precision

Proposition 1.2. *The relation $=_L$ is an equivalence relation on X.*

Proof. The reflexivity

$$F =_L F$$

is obvious. Also the symmetry is true by construction. Regarding the transitivity, let us assume to have $E, F, G \in X$, such that $E =_L F$, $F =_L G$. Using lemma 1.5, we have

$$\exists K_1, K_2 > 0 : \ K_1^{-1} |L[E]| \leq |L[F]| \leq K_2 |L[E]| \text{ evt.}, \tag{1.60}$$

as well as

$$\exists K_3, K_4 > 0 : \ K_3^{-1} |L[F]| \leq |L[G]| \leq K_4 |L[F]| \text{ evt.} \tag{1.61}$$

Therefore, there exist $K_5 = K_1 K_3$, $K_6 = K_2 K_4$, such that

$$K_5^{-1} |L[E]| \leq |L[G]| \leq K_6 |L[E]| \text{ evt.} \tag{1.62}$$

Applying again lemma 1.5, the last relation (1.62) implies $E =_L G$. □

Therefore, for $F \in X$, we can define the equivalence class

$$[F] := \{G \in X \mid G =_L F\},$$

which we call the *precision class* of F. Identifying predictions which have residues of the same order, we can come closer to a total ordering in the quotient set

$$X_{/L} := \{[F]_L \mid F \in X\}. \tag{1.63}$$

Definition 1.4. *Let $[\bar{F}], [\bar{G}] \in X_{/L}$. We define*

$$[\bar{F}] \prec_L [\bar{G}] \overset{def}{\iff} \bar{F} \prec_L \bar{G}.$$

For simplicity, we keep on using the same notation $\prec_L$ for the ordering, also when it is intended as a comparison of classes, on the quotient set.
To prove that $\prec_L$ is well defined on the classes, we have to show that it does not depend on the particular representing elements: given two classes $[\bar{F}]$ and $[\bar{G}]$, we can compare them choosing any $F \in [\bar{F}]$ and $G \in [\bar{G}]$.

Lemma 1.6 (semi-transitivity). *Let $E, F, G \in X$. Then,*

(a) $E \prec_L F, F =_L G \Rightarrow E \prec_L G$.

(b) $E =_L F, F \prec_L G \Rightarrow E \prec_L G$.

Proof. **(a)** From the definitions,

$$\forall \delta > 0 : |L[E]| < \delta |L[F]| \text{ evt.}$$

and

$$\exists K > 0 : \ |L[F]| < K |L[G]| \text{ evt.}$$

Hence, for all $\epsilon > 0$, taking $\delta = K^{-1}\epsilon$,

$$|L[E]| < \epsilon |L[G]| \text{ evt.}$$

Therefore, $L[E] \in o(L[G])$.
The statement **(b)** is proven using an analogous argument. □

Using lemma 1.6, we can demonstrate the following result.

Lemma 1.7. *Let $\bar{F}, \bar{G} \in X$, such that $\bar{F} \prec_L \bar{G}$. Then,*

$$\forall F \in [\bar{F}], \forall G \in [\bar{G}] : \ F \prec_L G.$$

Proof. Take any $F \in [\bar{F}]$ and $G \in [\bar{G}]$. We have then

$$F =_L \bar{F}, \bar{F} \prec_L \bar{G}.$$

Using lemma 1.6, part **(b)**,

$$F \prec_L \bar{G}.$$

Similarly,

$$F \prec_L \bar{G}, \bar{G} =_L G$$

implies (lemma 1.6, part **(a)**)

$$F \prec_L G.$$

□

In conclusion, the definition 1.4 is well posed.
We can prove an analogous of lemma 1.4, also for the ordering on the quotient set, so that

Proposition 1.3. *The relation $\prec_L$ defines a partial ordering on $X_{/L}$.*

However, going over from X to $X_{/L}$ may not be enough to turn $\prec_L$ in a total ordering. It may happen that for two elements $F, G \in X$, we are not able to decide whether $[F] \prec_L [G]$ or $[G] \prec_L [F]$.

Example 1.2. *To construct a special case, we consider the simple algorithm*

$$L(h, x_h) = x_h,$$

where $x_h \in [0, 1]$. *The solution of*

$$L(h, \hat{x}_h) = 0$$

is trivially $\hat{x}_h = 0$. *Now, we choose the elements*

$$F_h := \begin{cases} h & \left[\frac{1}{2k}, \frac{1}{2k-1}\right), \quad k \geq 1 \\ h^2 & \left[\frac{1}{2k+1}, \frac{1}{2k}\right), \quad k \geq 1 \end{cases}$$

and

$$G_h := h^{\frac{3}{2}}.$$

In this case it holds

$$L[F] = F \notin o(G) = o(L[G]),$$

and similarly

$$L[G] \notin o(L[F]),$$

so that neither $F \prec_L G$ *nor* $G \prec_L F$. *Moreover, note that*

$$F \notin O(G), \quad G \notin O(F),$$

hence

$$[F] \neq [G].$$

None of the statement $[F] \prec_L [G]$ *nor* $[G] \prec_L [F]$ *nor* $[F] = [G]$ *are true, i.e. the two elements cannot be compared.*

Requiring the total ordering: reasonable ansatz

Consequently, to obtain a total ordering, we have to restrict the investigation to reasonable subsets of X (i.e. excluding peculiar cases). As discussed in section 1.2.2, performing the asymptotic analysis we look for *approximate solutions* to L in an ansatz $\mathcal{A} \subset X$ (whose elements are named predictions).

Remark. The quotient set $X_{/L}$ is defined as the set of the classes

$$[F] = \{G \in X \mid G =_L F\}.$$

Considering an ansatz $\mathcal{A} \subset X$ (a subset of X in general), it might happen that, for $F \in \mathcal{A}$,

$$[F] \not\subset \mathcal{A}.$$

The quotient $\mathcal{A}_{/L}$ is defined as the subset of $X_{/L}$ of the precision classes which have at least a representative element in $\mathcal{A}$:

$$\mathcal{A}_{/L} = \{[F] \mid F \in \mathcal{A}\}\,. \tag{1.64}$$

How to guarantee a total ordering, even on the subset $\mathcal{A}_{/L}$?We need a particular consideration. Since the role of the ansatz is to simplify the investigation, extreme cases as in example 1.2 are normally excluded. We assume the total ordering as a property *a priori* of the ansatz.

Definition 1.5. *An ansatz $\mathcal{A} \subset X$ is* **reasonable** *iff the quotient set $\mathcal{A}_{/L}$ is totally ordered with respect to $\prec_L$.*

At this point, the question might arise how to check whether a certain ansatz is reasonable, i.e. whether $\prec_L$ is a total ordering on it. We do not answer this question in general. The argument would require many technicalities, which are not reflecting the usual way to proceed. Typically, the ansatz is defined according to the problem we are interested in. Only after focusing on a particular ansatz for a specific algorithm, we can discuss whether it is reasonable or not.

The asymptotic expansions Having introduced the arguments from a general point of view, we do here some considerations regarding a particular ansatz, starting with a simple example. Let us consider again the algorithm

$$L[\mathbf{x}] = \mathbf{x}$$

on $X = \mathcal{F}(H, \mathbb{R})$, and the ansatz

$$\mathcal{A} = \{F \in X \mid \forall h \in H :\ F_h = h^\alpha,\ \alpha \in [0,1]\}\,. \tag{1.65}$$

In this case it is easy to check that $\mathcal{A}$ is reasonable. In fact, $\forall F, G \in \mathcal{A}$, assuming

$$\forall h \in H :\ F_h = h^\alpha,\ G_h = h^\beta,$$

we have

$$F \prec_L G \iff \alpha < \beta.$$

Now, assume we want to apply the iterative procedure defined in (1.49) to the ansatz $\mathcal{A}$. For simplicity, we identify a prediction $F \in \mathcal{A}$ using the real number

$$\alpha_F \in [0,1] :\ \forall h \in H :\ F_h = h^{\alpha_F}.$$

We have that

$$\forall F \in \mathcal{A} :\ \alpha_F < 1 \Rightarrow \exists G \in \mathcal{A} :\ F \prec_L G.$$

Therefore, the first iteration gives

$$\mathcal{A}^{(1)} = \{F \in X \mid \forall h \in H :\ F_h = h^\alpha,\ \alpha \in [0,1)\} \tag{1.66}$$

(only $\alpha = 1$ has been excluded).
Any successive iteration, will not restrict the ansatz anymore:

$$\forall k \geq 1 : \mathcal{A}^{(k)} = \mathcal{A}^{(1)} = \{F \in X \mid \forall h \in H : \; F_h = h^\alpha, \; \alpha \in [0,1)\}. \tag{1.67}$$

The example shows that, in practice, to use the algorithm (1.49) the ansatz has to be totally ordered and not dense[j].
In our applications we will often consider

$$\mathcal{A} = \left\{ \mathbf{x} \in X \mid \mathbf{x}_h = \sum_{k \in K} h^k x_h^{(k)}, \; x_h^{(k)} \in X_h \right\}, \tag{1.68}$$

i.e. the set of the formal power series in h, a coefficients in a *index set* K. Definition (1.68) can be seen as a particular case of a more general class of ansatz.

Definition 1.6. *A sequence of functions* (ω_k), $\omega_k \in \mathcal{F}(H, \mathbb{R}_+)$, *is called* **asymptotic sequence** *(for* $h \to 0$*) iff*

- $\lim_{h \to 0} \omega_k = 0$,
- $\omega_{k+1} \in o(\omega_k)$, *for* $h \to 0$.

Definition 1.7. *Let* (ω_k) *be an asymptotic sequence, and let* $\{\mathcal{C}_h\}$ *be a family of sets, with* $\mathcal{C}_h \subset X_h$, *for all* $h \in H$. *We call the ansatz*

$$\mathcal{A} = \left\{ F \in X \mid F = \sum_{k \in K} \omega_k(h) f_h^{(k)}, \quad f_h^{(k)} \in \mathcal{C}_h \right\}$$

the set of formal **asymptotic expansions** in ω_k, with coefficients in the family $\{\mathcal{C}_h\}$. $\mathcal{C}_h$ *are called* **spaces of coefficients**.

The elements of the asymptotic sequence play a role similar to a *numerable* basis in the space $\mathcal{A}$, while the coefficients represent the degrees of freedom. In other words, the *free parameters* of the ansatz.

[j] Without restricting to powers of h, the space X contains predictions with quite arbitrary convergence speeds. Namely, having chosen any two functions $\omega_1(h)$, $\omega_2(h)$ converging to zero, such that

$$\omega_2 \in o(\omega_1)$$

(ω_2 is faster than ω_1), there exists $\omega_*(h)$ such that

$$\omega_* \in o(\omega_1), \quad \omega_2 \in o(\omega_*).$$

Definition of the optimum

Within a totally ordered ansatz $\mathcal{A}$, we can look for the minimal element.

Definition 1.8. *Let $\mathcal{A} \subset X$. F^* is an L-minimum in $\mathcal{A}$ (simply, a minimum) iff*

$$\nexists\, G \in \mathcal{A} \;:\; G \prec_L F^*.$$

Analogously, the notion of minimum can be defined for the classes in the quotient set $\mathcal{A}_{/L}$.

The following lemma characterizes the solutions to the algorithm, through the relation $\prec_L$.

Lemma 1.8. *Let $\mathbf{x} \in X$ such that $L[\mathbf{x}] = 0$. Then,*

(a) $\forall G \in X : |L[G]| > 0$ *evt.* $\Rightarrow \mathbf{x} \prec_L G$ *(a solution is more precise than all the non-solutions).*

(b) $\nexists\, F \in X :\; F \prec_L \mathbf{x}$ *(a solution is a minimum on X).*

Proof. **(a)** Let us take a G such that $|L[G]| > 0$ eventually. Hence, for all $\delta > 0$ it holds

$$0 = |L[\mathbf{x}]| < \delta |L[G]| \text{ evt.} \quad \Rightarrow \mathbf{x} \prec_L G.$$

(b) Let us suppose that

$$\exists F \in \mathcal{A} : F \prec_L \mathbf{x}.$$

It means,

$$\forall \delta > 0 :\; |L[F]| < \delta |L[\mathbf{x}]| \text{ evt.}$$

But it is absurd, since $\delta |L[\mathbf{x}]| = 0$ and $|L[F]| \geq 0$. □

Therefore, if the ansatz $\mathcal{A}$ contains a solution to L, this will be minimum in $\mathcal{A}$.

1.2.4 The heuristic algorithm

Algorithm 1.2 (Optimization of the ansatz).

(1) $\mathcal{A}^{(0)} = \mathcal{A}$

`for` $k \geq 0$

(2) `set` $\mathcal{A}^{(k+1)} = \left\{ F \in \mathcal{A}^{(k)} \mid \exists G \in \mathcal{A}^{(k)} :\; F \prec_L G \right\}$

(3) `repeat until` $\mathcal{A}^{(k+1)} = \emptyset$

Remarks. The construction of $\mathcal{A}^{(k+1)}$ in the algorithm, does not need to employ the precision classes. Let us fix an element $F \in \mathcal{A}^{(k)}$. Holding

$$\exists G \in \mathcal{A}^{(k)} : [F] \prec_L [G] \iff \exists G \in \mathcal{A}^{(k)} : F \prec_L G,$$

the iteration in algorithm 1.2 is equivalent to

$$\mathcal{A}^{(k+1)} = \left\{ F \in \mathcal{A}^{(k)} \mid \exists G \in \mathcal{A}^{(k)} : [F] \prec_L [G] \right\}.$$

We use a *class-free* notation. However, it has to be kept in mind that each prediction represents its precision class. Moreover, observe that if the ansatz $\mathcal{A}^{(k)}$ contains a prediction $F \in \mathcal{A}$, it contains the subset $[F] \cap \mathcal{A}$ of the class of precision.

Lemma 1.9. *Let $\mathcal{A}$ be a reasonable ansatz. Let K be such that $\mathcal{A}^{(K)} \neq \emptyset$.*

$$\mathcal{A}^{(K+1)} = \emptyset \iff \exists F \in \mathcal{A}^{(K)} : \mathcal{A}^{(K)}_{/L} = \{[F]\}. \tag{1.69}$$

Proof. First of all, we note that if $\mathcal{A}$ is reasonable, i.e. the quotient set $\mathcal{A}_{/L}$ is totally ordered, any subset $\mathcal{A}^{(k)}$ will be reasonable as well.
$\Rightarrow$) If $\mathcal{A}^{(K+1)} = \emptyset$, we formally end up with a set $\mathcal{A}^{(K)}$ such that

$$\forall F \in \mathcal{A}^{(K)} : \nexists G \in \mathcal{A}^{(K)} : G \prec_L F, \tag{1.70}$$

by construction. Let us choose $F \in \mathcal{A}^{(K)}$. Equation (1.70) excludes, for all $G \in \mathcal{A}^{(K)}$, the relation $[F] \prec_L [G]$ and $[G] \prec_L [F]$ (otherwise we could define a set of better predictions). Therefore, since the set $\mathcal{A}^{(K)}_{/L}$ is totally ordered,

$$\forall G \in \mathcal{A}^{(K)} : [F] = [G].$$

Hence,

$$\mathcal{A}^{(K)} \subset [F],$$

which is true for all $F \in \mathcal{A}^{(K)}$. Considering the quotient set we have

$$\forall F \in \mathcal{A}^{(K)} : \mathcal{A}^{(K)}_{/L} = \{[F]\}.$$

Equivalently,

$$\mathcal{A}^{(k)} = [F] \cap \mathcal{A}.$$

$\Leftarrow$) If $\mathcal{A}^{(K)}$ consists of a single precision classes,

$$\forall G_1, G_2 \in \mathcal{A}^{(K)} : G_1 =_L G_2.$$

Hence, there are no better predictions to define a new ansatz $\mathcal{A}^{(K+1)}$. □

In conclusion the algorithm ends with a single precision class $\mathcal{A}^{(K)}$, i.e. the *minimum*. Inside it we can choose a representative element.
The iteration for $\mathcal{A}^{(k+1)}$ does not provide any operational rule. If only few properties of L are known, we will start with a rather general ansatz $\mathcal{A}^{(0)}$. To construct the set $\mathcal{A}^{(1)}$, we would need to check for *all* the elements $F \in \mathcal{A}^{(0)}$, whether there exists a worse element.
In practice the procedure is simplified using an *heuristic version* of the algorithm (see next paragraph 2). It may result in missing the minimum in the original $\mathcal{A}$, finding a "sub-optimal" solution, or a minimum in a subset $\tilde{\mathcal{A}} \subset \mathcal{A}$.

1. Selecting the initial ansatz To begin, we choose an ansatz $\mathcal{A}$ which is supposed to be large enough to contain sufficiently many elements of X, but also not too general, otherwise the analysis will become hard. The choice is made according to the properties of the algorithm, the space and the kind of solutions we are interested in. It can be modified and adapted to the result we get, step by step.
We do not check at this stage whether the ansatz is reasonable or not. We can start the algorithm assuming it as hypothesis. If we encounter problems regarding the total ordering, we include additional constraints, or explicitely remove from the ansatz the elements which could produce problems.

2. Iteration step Let assume we have found the ansatz $\mathcal{A}^{(k)}$. To perform the iteration $k+1$, we do not process the whole $\mathcal{A}^{(k)}$. We will rather select a suitably large subset

$$\mathcal{A}^{(k+1)} \subset \left\{ F \in \mathcal{A}^{(k)} \mid \exists G \in \mathcal{A}^{(k)} : \; F \prec_L G \right\},$$

with predictions for which we can easily find worse ones. If we can find a satisfactory set (where "*satisfactory*" is again depending on the problem we are considering), we will take this as new ansatz $\mathcal{A}^{(k+1)}$. We call the ansatz $\mathcal{A}^{(k+1)}$, the set of *selected better predictions* at the iteration $k+1$.

Remarks. Observe that the procedure could be related to the formal one at any time. Selecting $\mathcal{A}^{(k+1)}$ as a subset of the effective better prediction is equivalent to include, at the step k, additional restrictions:

$$\mathcal{A}^{(k+1)} = \left\{ F \in \mathcal{A}^{(k)} \mid \exists G \in \mathcal{A}^{(k)} : \; F \prec_L G \right\} \cap I^{(k+1)}.$$

In view of

$$\mathcal{A}^{(k+1)} = \bigcap_{j \leq k+1} \mathcal{A}^{(j)}, \tag{1.71}$$

the condition defining the ansatz $\mathcal{A}^{(k+1)}$ can be inserted as well in any of the previous ansatz $\mathcal{A}^{(j)}$.

Therefore, we would obtain the same result at this level, *restarting* the algorithm with the ansatz

$$\tilde{\mathcal{A}}^{(0)} = \mathcal{A}^{(0)} \cap \left(\bigcup_{j \leq k+1} I^{(j)} \right) \tag{1.72}$$

and performing the iteration as defined in 1.2.

3. Checking the *stopping criterion* The iterations continue until we find

$$\mathcal{A}^{(k+1)} = \emptyset.$$

This condition has to be carefully checked during the analysis, before we proceed to the next iteration.

The size of the final precision class

Let us consider the equation

$$L(h, x_h) = x_h^3 + h x_h^2 - x_h = 0, \quad x_h \in \mathbb{C}. \tag{1.73}$$

It has exact solutions

$$\hat{x}_h^z = 0, \quad \hat{x}_h^{\pm} = \frac{-h \pm \sqrt{h^2+4}}{2}. \tag{1.74}$$

However, assume we look for approximate solutions in the ansatz

$$\mathcal{A} = \{x \in \mathbb{R}^H \mid x_h = x_0 + h x_1 + h^2 x_2\}.$$

Inserting a general $x \in \mathcal{A}$ into the algorithm, we can find the general expression of the residue

$$\begin{aligned} L(h, x_h) &= (x_0^3 - x_0) + h(3x_0^2 x_1 + x_0^2 - x_1) + h^2(3x_0 x_1^2 + 2x_0 x_1 + 3x_0^2 x_2 - x_2) \\ &+ h^3(x_1^3 + 6x_0 x_1 x_2 + x_1^2 + 2x_0 x_2) + h^4(3x_0 x_2^2 + 3x_1^2 x_2 + 2x_1 x_2) \\ &+ h^5(3x_1 x_2^2 + x_2^2) + h^6 x_2^3 \end{aligned} \tag{1.75}$$

which we can write as

$$L(h, x_h) = \sum_{k=0}^{6} h^k r^{(k)}(x_0, x_1, x_2). \tag{1.76}$$

The ansatz is reasonable. In fact, taken any two predictions, we can compare them simply checking which one has the lowest-order non zero coefficient. Actually, the same property is also used to find predictions more precise than others. We

can proceed order by order selecting at each iteration only the elements which are solutions of the equations

$$r^{(k)}(x_0, x_1, x_2) = 0.$$

For $k = 0$, the residue is

$$r^{(0)} = x_0^3 - x_0.$$

Hence, the predictions in the set

$$\mathcal{A}^{(1)} = \left\{x \in \mathcal{A} \mid x_h = x_0 + hx_1 + h^2x_2,\, x_0 \in \{0, 1, -1\}\right\}$$

are more precise than, for example, the elements $y \in \mathcal{A}$ such that

$$\forall h \in H :\, y_h = 2.$$

In general, for all the remaining

$$y \in \mathcal{A} :\, y_0^3 - y_0 \neq 0$$

we cannot find worse approximations. They belong all to the same precision classes. In this simple situation, $\mathcal{A}^{(1)}$ the complete set of better predictions.
At this point we can observe that the ansatz contains a solution to L, namely $x = 0$. This will be a minimum. To investigate approximations to the other solutions we can develop a *tree*-algorithm, following separately the subsets specified in the definition of $\mathcal{A}^{(0)}$. This can be done introducing additional hypotheses on the initial ansatz (see equation (1.72)). For example, to exclude $x = 0$, we assume to start with

$$\mathcal{A} = \{x \in \mathbb{R}^H \mid x_h = x_0 + hx_1 + h^2x_2,\, x_0 \neq 0\}.$$

Now,

$$\mathcal{A}^{(1)} = \left\{x \in \mathcal{A} \mid x_h = x_0 + hx_1 + h^2x_2,\, x_0 \in \{1, -1\}\right\}.$$

We can continue solving

$$r^{(1)}(x_0, x_1, x_2) = 3x_0^2x_1 + x_0^2 - x_1 = 0$$

and finding the set

$$\mathcal{A}^{(2)} = \left\{x \in \mathcal{A}^{(1)} \mid\ x_h = x_0 + hx_1 + h^2x_2,\, x_1 = \frac{x_0^2}{1 - 3x_0^2}\right\} = \\ = \left\{x \in \mathcal{A} \mid x_h \in \left\{\pm 1 - \frac{h}{2} + h^2x_2\right\}\right\}. \quad (1.77)$$

It is a set of better predictions. For instance, for $y \in \mathcal{A}^{(1)}$ such that

$$y_h = 1 + h$$

we have

$$\forall x \in \mathcal{A}^{(2)} : x \prec_L y.$$

Analogously, in the following step we have

$$\mathcal{A}^{(3)} = \{x \in \mathcal{A}^{(2)} \mid 3x_0x_1^2 + 2x_0x_1 + 3x_0^2x_2 - x_2 = 0\} = \\ = \left\{x \in \mathcal{A} \mid x_h \in \left\{x_h^p = 1 - \frac{1}{2}h + \frac{1}{8}h^2,\ x_h^n = -1 - \frac{1}{2}h - \frac{1}{8}h^2\right\}\right\}. \quad (1.78)$$

The last set contains only two elements. Since

$$L(h, x_h^p) = \frac{h^4}{64} + O(h^5),\ L(h, x_h^n) = -\frac{h^4}{64} + O(h^5)$$

they belong to the same class of precision.
Observe that

$$\forall x \in \mathcal{A}^{(3)} : r^{(3)}(x_0, x_1, x_2) = 0$$

and

$$\forall x \in \mathcal{A}^{(3)} : r^{(4)}(x_0, x_1, x_2) \neq 0$$

Since $\mathcal{A}^{(3)}$ is already a subset of a precision class, we conclude

$$\mathcal{A}^{(4)} = \emptyset$$

and the algorithm stops.
In this case, the algorithm terminates with two *fully determined* elements of the initial ansatz $\mathcal{A}^{(0)}$. Note that expanding in Taylor series the roots of (1.73), around $h = 0$, we find

$$\hat{x}^{\pm} = \pm 1 - \frac{1}{2}h \pm \frac{1}{8}h^2 + O(h^4). \quad (1.79)$$

An improper ansatz In the same way we investigate the simple equation

$$x^2 - h = 0. \quad (1.80)$$

with the ansatz

$$\mathcal{A} = \{x \in \mathbb{R}^H \mid x_h = x_0 + hx_1 + h^2x_2,\ x_0, x_1, x_2 \in \mathbb{R}\}.$$

The residue has the general form

$$L(h, x_h) = x_0^2 + h(2x_0x_1 - 1) + h^2(x_1^2 + 2x_0x_2) + h^3x_1x_2 + h^4x_2.$$

Hence, we can conclude that $x_0 = 0$ defines a set of better predictions:

$$\mathcal{A}^{(1)} = \left\{x \in \mathcal{A} \mid x_h = hx_1 + h^2x^2\right\}.$$

However, we are already left with a single precision class, since

$$\forall x \in \mathcal{A}^{(1)} : L(h, x_h) = -h + O(h^2).$$

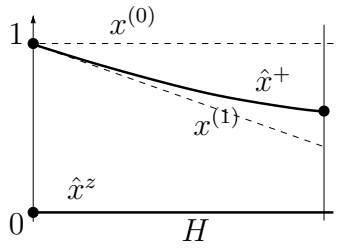


Figure 1.7: In the examples (1.73), one exact solution is $\hat{x}^+ = 1 - \frac{h}{2} + \sqrt{\frac{h^2}{4} + 1}$. It is drawn as bold line (together with the root $\hat{x}^z = 0$). With the considered ansatz $\mathcal{A}$, we found the predictions $x^{(0)} = 1$ and $x^{(1)} = 1 - \frac{h}{2}$ (dashed lines), which get close to the exact solution for h going to zero. Figure is not in scale.

Comparing the minimum classes These two simple applications show a relevant difference. In the first case, the minimum class (i.e. the class which is the minimum on $\mathcal{A}_{/L}$) is composed of two elements, x^p and x^n, which are two approximate solutions of equation (1.73). In the second example, the algorithm terminates with the ansatz $\mathcal{A}^{(1)}$. This is the minimum class, containing all the elements in the initial ansatz $\mathcal{A}^{(0)}$, which have a distance $O(h)$ from $x = 0$. *Intuitively*, this class has *size* $O(h)$.
The concept of size can be properly defined. Let be given a function $\omega \in \mathcal{F}(H, \mathbb{R}_+)$. For $F \in \mathcal{A}$, we can say that the class $[F]$ has size $O(\omega)$ in $\mathcal{A}$ if any other element $G \in \mathcal{A}$, such that $F - G \in O(\omega)$, belongs to the class $[F]$:

$$\forall G \in \mathcal{A} : |F - G| \in O(\omega) \Rightarrow F =_L G.$$

Equivalently, $[F]$ has size $O(\omega)$ in $\mathcal{A}$ iff

$$[F] + \mathcal{A} \cap O(\omega) \subset [F]. \tag{1.81}$$

Additionally, we can say that the class $[F]$ has size $o(\omega)$ in $\mathcal{A}$ iff

$$\exists \psi \in o(\omega) : [F] \text{ has size } O(\psi). \tag{1.82}$$

Since the technique is in principle applicable to a wide class of problems, we do not discuss the details of the definition.
The size of the minimum class in the initial ansatz indicates whether the ansatz is *adapt* to investigate the algorithm. If the last iteration provides a *too big* class, there are different possible choices for the best prediction, which give the same precision. In other words, we are taking into account too many degrees of freedom, more than how many we are actually able to fix.
Remark. Considering the ansatz of the asymptotic expansions with respect to the sequence (ω_k), definition (1.81) can be interpreted as follows. A class has size $O(\omega_k)$ if the precision does not depend on the k-th coefficient.

A different example: numerical schemes

We consider now the differential problem

$$x' = \lambda x, \quad x(0) = x_0,\ \lambda > 0 \tag{1.83}$$

for $x : [0,1] \to \mathbb{R}$, which we try to solve numerically using the *explicit Euler method*:

$$\begin{aligned} &\hat{x}_h(0) = x_0, \\ &\hat{x}_h(n+1) - \hat{x}_h(n) = h\lambda \hat{x}_h(n), \quad n \geq 0. \end{aligned} \tag{1.84}$$

For this problem (as for any numerical method for differential initial value problems), we take

$$H = \left\{ \frac{1}{N} \mid N \in \mathbb{N},\ N > 0 \right\},$$

which identifies, for each N, the N-points discretization of $[0,1]$. For $h \in H$,

$$\hat{x}_h(n) \in X_h = \mathcal{F}(\{0, 1, \ldots, N(h)\}, \mathbb{R}),$$

i.e. it is a function on an h-depending discrete set, defined according to a discretization of $[0,1]$ with nodes $t_n = nh$.
Viewing (1.84) as a problem in the form (1.40),

$$L(h, \hat{x}_h)(n+1) = \begin{cases} \hat{x}_h(0) - x_0 & n+1 = 0 \\ \hat{x}_h(n+1) - \hat{x}_h(n) - h\lambda \hat{x}_h(n) & n+1 > 0, \end{cases}$$

we apply the technique defined above, to approximate the solution $\hat{x}_h$, which corresponds, in this case, to the numerical solution provided by the scheme.
First, we chose the ansatz

$$\mathcal{A} = \left\{ y \in X \mid y_h(n) = \sum_{k \in K} h^k y^{(k)}(hn), \quad y^{(k)}(t) \in C^\infty(I) \right\}$$

(K being a finite set of indices). It is a set of the power series in h, with *smooth* coefficients. Let us consider the iteration for $n > 0$. Inserting the ansatz into the algorithm (1.84), we get

$$L(h, y_h)(n+1) = \sum_{k \in K} h^k y^{(k)}(t_{n+1}) - \sum_{k \in K} h^k y^{(k)}(t_n) - \lambda h \sum_{k \in K} h^k y^{(k)}(t_n).$$

Now, Taylor expanding the functions $y^{(k)}$ around $t = t_n$,

$$L(h, y_h) = h\left(y^{(0)\prime} - \lambda y^{(0)} \right) + h^2 \left(y^{(1)\prime} - \frac{1}{2} y^{(0)\prime\prime} - \lambda y^{(1)} \right) + O(h^3). \tag{1.85}$$

As done in the previous example, we define the sets $\mathcal{A}^{(k)}$ with the predictions which cancel the different orders in the residues (1.85). Regarding the initial conditions, the residue reads

$$L(h, y_h)(0) = y^{(0)}(0) - x_0 + \sum_{k \in K} h^k y^{(k)}(0). \tag{1.86}$$

Solving $L(h, y_h)(0) = 0$ order by order, equation (1.86) fixes the initial condition for the coefficients of the predictions.
Considering the leading order, we construct the set

$$\mathcal{A}^{(1)} = \{y \in \mathcal{A} \mid y^{(0)\prime} - \lambda y^{(0)} = 0,\ y^{(0)}(0) = x_0\},$$

of the better predictions. It contains a relation with the solutions of the original problem (1.83).
Performing a further step, we have

$$\mathcal{A}^{(2)} = \left\{ y \in \mathcal{A}^{(1)} \mid y^{(1)\prime} = \frac{1}{2} y^{(0)\prime\prime} - \lambda y^{(1)} = \frac{1}{2}\lambda^2 y^{(0)} - \lambda y^{(1)},\ y^{(1)}(0) = 0 \right\}.$$

The process could be continued. However, limiting the initial ansatz to

$$\tilde{\mathcal{A}} = \{y \in \mathcal{A} \mid y_h(n) = y^{(0)}(hn) + h y^{(1)}(hn), \quad y^{(0)}, y^{(1)} \in C^\infty(I)\}$$

(taking $K = \{0, 1\}$) we finish after two steps.
Remarks. Using the asymptotic analysis, we have been able to derive some information about the numerical method. Summarizing,

- the solution of the differential problem (1.83) does not satisfy the algorithm (1.84);
- the leading order of the asymptotic solution to (1.84), however, solves the same differential problem the method has been applied to;
- $\mathcal{A}^{(2)}$ contains the minimum on $\mathcal{A}_{/L}$. Whether the precision class contains a single element or not depends, in general, on the differential problem we are considering.

1.3 A *regular* case: from the LBE to Navier-Stokes

The asymptotic expansion has been very successfully applied to the lattice Boltzmann method, to study the behavior in periodic domains [19, 25]. Besides, boundary conditions [9, 20, 23, 39] and several other features, such as grid-refinement

[38], initial conditions [5], force evaluation [6] and fluid mixtures [1] have been investigated with the same principle.
We recover here some results, applying the technique described before. With the practical example, we show how how the analysis works and the conclusion it leads to. In particular, since in the rest of the thesis we make a wide use of asymptotic analysis, scope of this part is also to explain the general ideas of the practical procedure.
The computations are carried out for the particular BGK model. Details, more general results and overviews of the asymptotic expansion for the LBE can be found in [19].

The ingredients of the analysis

Continuous and discrete problems We have in mind to solve the motion of a fluid governed by an initial value incompressible Navier-Stokes problem

$$\begin{aligned} &\begin{cases} \nabla \cdot \mathbf{u} = 0 \\ \partial_t \mathbf{u} + \nabla p + \mathbf{u} \cdot \nabla \mathbf{u} = \nu \nabla^2 \mathbf{u} + \mathbf{G} \end{cases} \quad t \in [0,T],\ \mathbf{x} \in \Omega \\ &\mathbf{u}(0,\mathbf{x}) = \mathbf{u}_0(\mathbf{x}), \quad \mathbf{x} \in \Omega, \end{aligned} \tag{1.87}$$

on a *periodic domain* $\Omega \subset \mathbb{R}^2$. We refer to equations (1.87) as the **continuous problem**.
The spatial discretization is a grid $\mathcal{G}(\Delta x)$, as in equation (1.4). The time step is chosen according to the scaling

$$\Delta t = \Delta x^2 = h^2 \tag{1.88}$$

(called in literature *diffusive scaling*), which has been demonstrated[k] to be necessary condition [19] to recover the problem (1.87) in the limit $\Delta x, \Delta t \to 0$.
The **discrete problem** is the BGK-LBM in a periodic domain:

$$\hat{f}_i(n+1, \mathbf{j}+\mathbf{c}_i) = \hat{f}_i(n,\mathbf{j}) + \underbrace{\frac{1}{\tau}\left(f_i^{eq}(\hat{f}) - \hat{f}_i\right)(n,\mathbf{j}) + g_i(n,\mathbf{j})}_{\hat{J}_i(n,\mathbf{j})}. \tag{1.89}$$

The term

$$g_i(n,\mathbf{j}) = h^3 f_i^* c_s^{-2} \mathbf{c}_i \cdot \mathbf{G}(t_n, \mathbf{x}_\mathbf{j}), \tag{1.90}$$

is used to take into account the effect of the volume force $\mathbf{G}$ in equations (1.87). The origin of the expression for g_i will be clear at the end of the analysis. Observe that

$$\sum_i g_i = 0, \quad \sum_i g_i \mathbf{c}_i = h^3 \mathbf{G}. \tag{1.91}$$

[k] In the approach presented here, we assume (1.88) as a former relation. In [19], the condition results as a consequence of the analysis.

Calling $N(T,h)$ the number of time steps necessary to cover the time interval $[0,T]$ of problem (1.87), the numerical solution $\hat{f}_h$ is a function on the **discrete domain**

$$\mathbb{D}_h = \{0,1,\dots,N(T,h)\} \times \mathcal{G}(h). \tag{1.92}$$

Let us describe the LBM using the formalism introduced in section 1.1. In this case, H is the subset of $(0,1]$ containing the grid sizes h, which allow a periodic discretization of the continuous domain. Then, for all $h \in H$, we have

$$X_h = \mathcal{F}(\mathbb{D}_h, \mathbb{R}^{b+1}). \tag{1.93}$$

The interior algorithm For each $h \in H$ the interior LBM is defined by the set of equations (1.89)

$$LBM_i(h,\hat{f}_h)(n+1,\mathbf{j}) = \hat{f}_i(n+1,\mathbf{j}+\mathbf{c}_i) - \hat{f}_i(n,\mathbf{j}) + \hat{J}_i(n,\mathbf{j}) = 0$$

for[l] $n+1>0$. We consider

$$L(h,\hat{f}_h)(n,\mathbf{j}) = \mathrm{LBM}(h,\hat{f}_h)(n,\mathbf{j}) \quad n \in \{1,\dots,N(T,h)\},\ \mathbf{j} \in \mathcal{G}(h), \tag{1.94}$$

where $\mathbb{D}_h = \{1,\dots,N(T,h)\} \times \mathcal{G}(h)$ is the *interior discrete domain*, excluding the initial conditions for $\hat{f}_h$.
A solution[m] to L_h is a numerical result $\hat{f}_h$, which we obtain running the LBM on a grid of size h.
To apply the asymptotic analysis, following the strategy described in section 1.2.4, we need now to construct an ansatz.

1.3.1 The regular expansion

In the problem

$$L(h,\hat{f}_h) = 0,$$

the $\hat{f}_h \in \mathcal{F}(\mathbb{D}_h, \mathbb{R}^{b+1})$ are functions on an h-discrete domain $\mathbb{D}_h$. Dealing with continuous and discrete problems, the grid is related to a continuous domain D, through a function

$$\eta_h : \mathbb{D}_h \to D, \tag{1.95}$$

which we call **discretization map**.

Example 1.3. *For the LBM, we have*

$$\eta_h^d(n,\mathbf{j}) = (nh^2, h\mathbf{j}), \tag{1.96}$$

defined according to the (diffusive) scaling (1.88).

[l] We use the index $n+1$ to define the algorithm, since equation (1.89) expresses an update rule for the variables $\hat{f}_i$ at time $n+1$.

[m] To completely define a solution, proper initial (and boundary) conditions are needed. The interior algorithm does not contain information regarding the starting value for $n=0$.

Definition 1.9 (Coordinate map). *Let E be a set and*

$$Y_h(E) := \mathcal{F}(\mathbb{D}_h, E).$$

An element $y \in Y(E) = \prod_{h \in H} Y_h(E)$ is called a coordinate map *(from $\mathbb{D}_h$) on E.*

So, the discretization map is a particular coordinate map on D. Given a coordinate map y on E, we have a recipe to construct a particular subset of X (for the LBM, X_h is defined in equation (1.93)). Namely (diagram in figure 1.8), for a function $f : E \to \mathbb{R}^d$, we have

$$f(y_h) \in X_h.$$

Defining

$$F := f(y), \quad \text{with } f(y)(h) := f(y_h),$$

then $F \in X$.

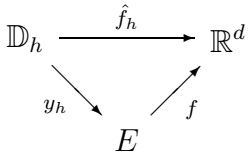


Figure 1.8: Diagram of the relevant spaces, cited in the definition 1.9 of a coordinate map. The numerical solution $\hat{f}_h$ is a general function defined on the discrete domain. We choose a particular ansatz, constructed through a coordinate map.

Definition 1.10. *Let y be a coordinate map on a given set E, and $\mathcal{C}$ a space of functions defined on E. The ansatz of* **power series with coefficients in** $\mathcal{C}$ *is the set*

$$\mathcal{A}(y, \mathcal{C}) := \left\{ F \in X \mid F_h = \sum_{k \in K} h^k f^{(k)}(y_h),\ f^{(k)} \in \mathcal{C} \right\},$$

where K is a finite set of indices.

Definition 1.11. *The class $\mathcal{A}(y, \mathcal{C})$ is called* **regular ansatz** *iff*

$$\begin{array}{l} \textbf{(1)}\ y \text{ is a coordinate on } D \\ \textbf{(2)}\ \mathcal{C} = C^\infty(D) \end{array} \tag{1.97}$$

The **regular expansion** *is the analysis performed within a regular ansatz.*

Note that in a regular expansion the spaces of coefficients do not depend on h. Actually, the hypothesis of $C^\infty(D)$ coefficients could be replaced with a less restrictive requirement. This assumption allows to simplify the investigation. The regularity we need to perform the analysis can be fixed *a posteriori*.

Coefficients of the regular predictions

Let us consider the ansatz $\mathcal{A}(y, \mathcal{C})$ of the expansions through the coordinate map y. Let K be the set of indices, and let us call

$$\bar{k} = \max\{k \mid k \in K\}.$$

Starting from a set of functions $\left\{f^{(k)} \mid f^{(k)} \in \mathcal{C},\ k \leq \bar{k}\right\}$ it is immediate to construct the prediction $F \in \mathcal{A}$ with coefficients $f^{(k)}$, defining

$$\forall h \in H:\ F_h(f^{(0)}, \ldots, f^{(\bar{k})}) = \sum_{k \leq \bar{k}} h^k f^{(k)}(y_h). \tag{1.98}$$

On the other hand, given $F \in \mathcal{A}$, how can we pick up the coefficient of order k? In words, we are looking for a map

$$\begin{array}{rccc} f_i^{(k)}: & \mathcal{A} & \to & \mathcal{C} \\ & F & \mapsto & f_i^{(k)}(F) \end{array} \tag{1.99}$$

which gives "*the k-th coefficient of* F_h", such that

$$\forall h \in H:\ F_h = \sum_{k \leq \bar{k}} h^k f^{(k)}(F)(y_h).$$

The previous equation has to hold at each point of the discrete domain. Therefore, the functions $f_i^{(k)}(F)$ can be defined *pointwise* starting from the prediction F. Let us focus on the regular ansatz for the LBM, where $\mathbb{D}_h = \mathbb{N} \times \mathcal{G}(h)$ and

$$y_h(n, \mathbf{j}) = (nh^2, h\mathbf{j}) = (t_n, \mathbf{x}_\mathbf{j}).$$

However, the argument can be extended also in more general cases (see 2.2). For fixed coordinates $(t, \mathbf{x}) \in D$, we need to consider a proper limit processes. Namely, we choose a sequence of parameters $(h_m)_{m \geq 0}$, tending to zero,

$$h_m \overset{m \to \infty}{\longrightarrow} 0,$$

and two sequences of indeces (n_{h_m}) and $(\mathbf{j}_{h_m})$ such that

$$h_m^2 n_{h_m} \overset{m \to \infty}{\longrightarrow} t, \quad h_m \mathbf{j}_{h_m} \overset{m \to \infty}{\longrightarrow} \mathbf{x}.$$

Then, the coefficients in $(t, \mathbf{x})$ can be defined iteratively:

$$\begin{aligned} f^{(0)}(F)(t, \mathbf{x}) &= \lim_{m \to \infty} F_{h_m}(n_{h_m}, \mathbf{j}_{h_m}), \\ \forall k > 0:\ f^{(k)}(F)(t, \mathbf{x}) &= \lim_{m \to \infty} \frac{1}{h_m^k} \left(F_{h_m}(n_{h_m}, \mathbf{j}_{h_m}) - \sum_{l \leq k-1} h_m^l f^{(l)}(F)(t, \mathbf{x}) \right). \end{aligned} \tag{1.100}$$

1.3.2 Preparing the regular ansatz for the LBM

In this section we describe the application of the heuristic algorithm 1.2 to search for a prediction for the interior algorithm (1.94). At the beginning, many steps will be done on a not totally rigorous level, taking into account some considerations derived from the known properties of the method, and from already existent analysis [19] of it. Step by step, the results will be compared with the standard analysis presented in [19].
A general prediction in the regular ansatz $\mathcal{A}(\eta^d, C^\infty(D))$ for the LBM has the form

$$F_h(n, \mathbf{j}) = f^{(0)}(t_n, \mathbf{x_j}) + hf^{(1)}(t_n, \mathbf{x_j}) + h^2 f^{(2)}(t_n, \mathbf{x_j}) + \dots \tag{1.101}$$

with *smooth* coefficients $f^{(k)}$. As remarked before, the series expressing F does not consist of an infinite number of terms, but it is extended to a finite set K. However, the number of considered coefficients can be fixed in a second moment. Similarly to what we have done in the previous examples, we start inserting the expansion (1.101) into the algorithm (1.94)

$$\begin{aligned}\mathrm{LBM}_i(h, F_h)(n+1, \mathbf{j}) = \sum_{k\in K} h^k f_i^{(k)}(t_{n+1}, \mathbf{x_{j+c}}_i) - \sum_{k\in K} h^k f_i^{(k)}(t_n, \mathbf{x_j}) - \\ \frac{1}{\tau}\left(f_i^{eq}\left(\sum_{k\in K} h^k f^{(k)}(t_n, \mathbf{x_j})\right) - \sum_{k\in K} h^k f_i^{(k)}(t_n, \mathbf{x_j})\right) - g_i(n, \mathbf{j}).\end{aligned}$$

We assume that also the term $g_i(n, \mathbf{j})$ can be expanded in power series[n]

$$g_i(n, \mathbf{j}) = \sum_{k\in K} h^k g_i^{(k)}(t_n, \mathbf{x_j})$$

(choosing a suitable K).
The right hand side contains smooth functions on the continuous domain D. Using a Taylor expansion around the point $(t_n, \mathbf{x_j})$, we get

$$\begin{aligned}\mathrm{LBM}_i(h, F_h)(n+1, \mathbf{j}) = \sum_{k\in K} h^k \left(h^2\partial_t + h\mathbf{c}_i\cdot\nabla + \dots\right) f_i^{(k)}(t_n, \mathbf{x_j}) - \\ \sum_{k\in K} h^k \left[\frac{1}{\tau}\left(f_i^{eq,(k)}(F_h(n, \mathbf{j})) - f_i^{(k)}(t_n, \mathbf{x_j})\right) - g_i^{(k)}(t_n, \mathbf{x_j})\right] + \dots,.\end{aligned} \tag{1.102}$$

grouping in $f^{eq,(k)}$ all the terms of order h^k of the equilibrium function. The dots at the end of the equation allude to a remainder, which contains all the term of

[n] With definition (1.90), we have

$$g_i^{(3)} = f_i^* c_s^{-2}\mathbf{c}_i\cdot\mathbf{G}, \quad g_i^{(k)} = 0, \quad \text{for } k \neq 3.$$

the expansion with indices $k \notin K$. We do not deal with this technical problem. In the considered case the set K will always contain the relevant orders, while the remainder will be asymptotically ignorable.

Equilibrium of order k To derive an explicit expression of $f^{eq,(k)}$ we consider the equilibrium as a function of the moments of the particle distribution (equation (1.25)):

$$f_i^{eq}(F_h) = H_i^{eq}\left(\rho(F_h), \mathbf{u}(F_h)\right),$$

where ρ and $\mathbf{u}$ are defined as (equation (1.22))

$$\rho(F_h) = \sum_i F_{hi}, \quad \mathbf{u}(F_h) = \sum_i F_{hi}\mathbf{c}_i. \tag{1.103}$$

Considering F in the form (1.101), analogous expansions are inherited from the moments:

$$\rho(F_h) = \sum_{k\in K} h^k \rho^{(k)}(F), \quad \mathbf{u}(F_h) = \sum_{k\in K} h^k \mathbf{u}^{(k)}(F), \tag{1.104}$$

where the coefficients of order k (h-independent) are obtained taking the moments of the coefficients of F (defined in equation (1.100)) of the corresponding order:

$$\rho^{(k)}(F) = \sum_i f_i^{(k)}(F), \quad \mathbf{u}^{(k)}(F) = \sum_i f_i^{(k)}(F)\mathbf{c}_i. \tag{1.105}$$

Introducing the expression (1.26) for H_i^{eq}, the *equilibrium of order k* is defined isolating (splitting H_i^{eq} as in equation (1.28))

$$f_i^{eq,(k)}(F) = H_i^{L(eq)}(\rho^{(k)}(F), \mathbf{u}^{(k)}(F)) + \sum_{m+l=k} H_i^{Q(eq)}(\mathbf{u}^{(m)}(F), \mathbf{u}^{(l)}(F)). \tag{1.106}$$

Observe that $f_i^{eq,(k)}$ depends only on the coefficients of the prediction F, through equations (1.105).
In the leading orders we have

$$f^{eq,(0)} = f_i^*\rho^{(0)} + c_s^{-2} f_i^* \mathbf{c}_i \cdot \mathbf{u}^{(0)} + H_i^{Q(eq)}(\mathbf{u}^{(0)}, \mathbf{u}^{(0)}), \tag{1.107}$$

$$f^{eq,(1)} = f_i^*\rho^{(1)} + c_s^{-2} f_i^* \mathbf{c}_i \cdot \mathbf{u}^{(1)} + 2H_i^{Q(eq)}(\mathbf{u}^{(0)}, \mathbf{u}^{(1)}), \tag{1.108}$$

$$f^{eq,(2)} = f_i^*\rho^{(2)} + c_s^{-2} f_i^* \mathbf{c}_i \cdot \mathbf{u}^{(2)} + H_i^{Q(eq)}(\mathbf{u}^{(1)}, \mathbf{u}^{(1)}) + 2H_i^{Q(eq)}(\mathbf{u}^{(0)}, \mathbf{u}^{(2)}). \tag{1.109}$$

Using (1.106), we can eventually write equation (1.102) in the form

$$\mathrm{LBM}_i(h, F_h)(n+1, \mathbf{j}) = \sum_{k\in K} h^k r_i^{(k)}(F)(\eta_h^d(n, \mathbf{j})), \tag{1.110}$$

where $r^{(k)}(F)(\eta_d)$ is, for each $k \geq 0$, the residue of order h^k, which depends only on the continuous functions $f_i^{(k)}(F)$ (treating the expansion coefficients of the force term $\mathbf{g}$ as a constant).

Explicitely,

$$r_i^{(k)}(F) = \sum_{l+m=k,\, l>0} D_{l,i} f_i^{(m)}(F) - \frac{1}{\tau}\left(f_i^{eq,(k)}(F) - f_i^{(k)}(F)\right) + g_i^{(k)}. \tag{1.111}$$

As in [19], we have introduced the operator

$$D_{l,i} = \sum_{a+b=l} \frac{\partial_t{}^a}{a!} \frac{(\mathbf{c}_i \cdot \nabla)^b}{b!}, \tag{1.112}$$

which groups all the derivatives giving order l in h.
To perform the iteration to find the minimum, we proceed trying to construct, order by order, the coefficients $f^{(k)}(F)$ such that

$$r^{(k)}(F) = 0, \tag{1.113}$$

which expresses a **constraint** on the coefficients of the prediction F.

Selecting the better predictions The solutions for the coefficients are derived recursively, starting from $k = 0$ and inserting in the recovered solutions in the following orders.
We start from a regular ansatz $\mathcal{A}$. Since the expansion of F_h begins from $k = 0$, we assume

$$\forall k < 0 : \; f_i^{(k)}(F) = 0.$$

Additionally, to make more comfortable the reading, we index the iterations of the algorithm 1.2 starting from $k = 0$, instead of with $k = 1$.

[$\mathbf{k = 0}$] In the leading order, we have

$$r^{(0)}(F) = f^{(0)}(F) - f^{eq,(0)}(F). \tag{1.114}$$

First, we search a prediction which does not cancel the residue. Let us consider the element $G \in \mathcal{A}$ such that

$$\forall h \in H : \; G_i := |\mathbf{c}_i|^2 - \frac{\sum_m |\mathbf{c}_m|^2}{\sum_m 1}, \tag{1.115}$$

for which we have $f_i^{(0)}(G) = G$. The moments are

$$\rho^{(0)}(G) = 0, \quad \mathbf{u}^{(0)}(G) = 0,$$

and the equilibrium of order zero is (using equation (1.107)) $f_i^{eq,(0)}(G) = 0$. As a consequence the residue (1.114) reads

$$r^{(0)}(G) = G \neq 0. \tag{1.116}$$

Therefore, taking a subset of the solutions of (1.114) we can select a set of predictions which are better than G. Defining

$$\mathcal{A}^{(0)} = \{F \in \mathcal{A} \mid f_i^{(0)}(F) = f_i^*\}, \tag{1.117}$$

(which is trivially not empty), it is easy to show that

$$\forall F \in \mathcal{A}^{(0)} : \rho^{(0)}(F) = 1, \quad \mathbf{u}^{(0)}(F) = 0. \tag{1.118}$$

Hence $f_i^{eq,(0)} = f_i^*$, which implies

$$\forall F \in \mathcal{A}^{(0)} : r^{(0)}(F) = 0$$

and $F \prec_L G$. The set $\mathcal{A}^{(0)}$ is the subset of the *selected better predictions* in the ansatz $\mathcal{A}$.

[$\mathbf{k = 1}$] We proceed in a similar way. Equation (1.111) for $k = 1$ reads

$$r^{(1)}(F) = f^{(1)}(F) - f^{eq,(1)}(F). \tag{1.119}$$

Now, we consider the prediction $G^{(1)} \in \mathcal{A}^{(0)}$ such that

$$\forall h \in H : G_i^{(1)} := f_i^* + hG_i, \tag{1.120}$$

G being the prediction defined in (1.115). As before

$$f^{(1)}(G^{(1)}) = G, \quad \rho^{(1)}(G^{(1)}) = 0, \quad \mathbf{u}^{(1)}(G^{(1)}) = 0, \tag{1.121}$$

which give (equation (1.108)) $f_i^{eq,(1)} = 0$ and

$$r^{(1)}(G^{(1)}) = G \neq 0.$$

A set of better predictions can be constructed defining

$$\mathcal{A}^{(1)} = \{F \in \mathcal{A}^{(0)} \mid f_i^{(1)}(F) = c_s^{-2} f_i^* \mathbf{c}_i \cdot \mathbf{v}^{(1)},\ \mathbf{v}^{(1)} \in (C^\infty(D))^d\}. \tag{1.122}$$

In fact, since

$$\forall F \in \mathcal{A}^{(1)} : \rho^{(1)}(F) = 0, \quad \mathbf{u}^{(1)}(F) = \mathbf{v}^{(1)}, \tag{1.123}$$

we have $f^{eq,(1)}(F) = f^{(1)}(F)$ and the residue vanishes.
In the definition (1.122) there are no special conditions on the field $\mathbf{v}^{(1)}$ appearing in $f_i^{(1)}(F)$. In fact,

$$f_i^{(1)}(F) = c_s^{-2} f_i^* \mathbf{c}_i \cdot \mathbf{v}^{(1)} \tag{1.124}$$

is only a constraint on the structure of the coefficients[o] $f_i^{(1)}$.
Obviously, $\mathcal{A}^{(1)} \neq \emptyset$.

[o]Due to the symmetry properties of the lattice, it holds

$$\forall \mathbf{v}^{(1)} \in \mathcal{F}(D, \mathbb{R}^d) : \mathbf{v}^{(1)} = \sum_i \mathbf{c}_i \left(c_s^{-2} f_i^* \mathbf{c}_i \cdot \mathbf{v}^{(1)}\right),$$

Example 1.4 (Stationary first order). *For example, we could restrict the analysis to an even smaller subset of the better predictions, choosing* $\mathbf{v}^{(1)} = 0$*:*

$$\mathcal{A}^{(1)} = \left\{ F \in \mathcal{A}^{(0)} \mid f^{(1)}(F) = 0 \right\}.$$

Despite the restrictive choice, it is still possible to perform another iteration of the algorithm. More details are given in the following derivation.

[$\mathbf{k = 2}$] In the following order we obtain

$$r_i^{(2)}(F) = f_i^{(2)}(F) - \left(f_i^{eq,(2)}(F) - \tau \mathbf{c}_i \cdot \nabla f_i^{(1)}(F) \right). \tag{1.125}$$

It can be proven that $r^{(2)}$ differs from zero choosing the prediction $G^{(2)} \in \mathcal{A}^{(1)}$ defined by

$$\forall h \in H : G^{(2)} := f_i^* + h^2 G. \tag{1.126}$$

To cancel the second order residue we select the elements $F \in \mathcal{A}^{(1)}$ such that

$$f_i^{(2)}(F) = f_i^* q^{(2)} + c_s^{-2} f_i^* \mathbf{c}_i \cdot \mathbf{v}^{(2)} - \tau \mathbf{c}_i \cdot \nabla f_i^{(1)}(F), \tag{1.127}$$

for smooth fields $q^{(2)}$ and $\mathbf{v}^{(2)}$.
We have

$$\rho^{(2)}(F) = q^{(2)} - \tau \nabla \cdot \mathbf{u}^{(1)}(F), \quad \mathbf{u}^{(2)}(F) = \mathbf{v}^{(2)}. \tag{1.128}$$

Inserting (1.128) into the equilibrium (1.109) yields

$$r^{(2)}(F) = -\tau f_i^* \nabla \cdot \mathbf{u}^{(1)}(F).$$

Therefore, for predictions $F \in \mathcal{A}^{(1)}$, the residue $r^{(2)}(F)$ vanishes only if $\mathbf{u}^{(1)}(F)$ is *divergence-free*. The set

$$\begin{aligned} \mathcal{A}^{(2)} = \{ F \in \mathcal{A}^{(1)} \mid f_i^{(2)}(F) = f_i^* q^{(2)} + c_s^{-2} f_i^* \mathbf{c}_i \cdot \mathbf{v}^{(2)} - \tau \mathbf{c}_i \cdot \nabla f_i^{(1)}(F), \\ q^{(2)} \in C^\infty(D),\ \mathbf{v}^{(2)} \in (C^\infty(D))^d,\ \nabla \cdot \mathbf{u}^{(1)}(F) = 0 \} \end{aligned} \tag{1.129}$$

contains better predictions.
In view of (1.123), to have

$$\mathcal{A}^{(2)} \neq \emptyset,$$

the field $\mathbf{v}^{(1)}$, which could be freely chosen to define the ansatz $\mathcal{A}^{(1)}$, has to be divergence-free. All the elements in $\mathcal{A}^{(1)}$ such that $\nabla \cdot \mathbf{u}^{(1)}(F) \neq 0$ will be excluded at this stage.
Observe that the particular choice $\mathbf{v}^{(1)} \equiv 0$ (example 1.4) satisfies the condition. Equivalently, we could have selected at the previous step (equation (1.71)) the set

$$\mathcal{A}^{(1)} = \{ F \in \mathcal{A}^{(0)} \mid f_i^{(1)}(F) = c_s^{-2} f_i^* \mathbf{c}_i \cdot \mathbf{v}^{(1)},\ \mathbf{v}^{(1)} \in (C^\infty(D))^d,\ \nabla \cdot \mathbf{v}^{(1)} = 0 \}. \tag{1.130}$$

[$\mathbf{k=3}$] Continuing in the same way, to cancel the following order we consider the predictions $F \in \mathcal{A}^{(2)}$ such that

$$f_i^{(3)}(F) = f_i^* q^{(3)} + c_s^{-2} f_i^* \mathbf{c}_i \cdot \mathbf{v}^{(3)} - \tau \left(\partial_t f_i^{(1)}(F) + \mathbf{c}_i \cdot \nabla f_i^{(2)}(F) - g_i^{(3)} \right), \quad (1.131)$$

for smooth fields $q^{(3)}$ and $\mathbf{v}^{(3)}$. Computing the third order moments, evaluating equation (1.111) for $k=3$ and the equilibrium of order three (details in [19]), we have $r^{(3)}(F)=0$ iff

$$\begin{aligned} &\partial_t \mathbf{u}^{(1)}(F) + c_s^2 \nabla \rho^{(2)}(F) + \nabla \cdot (\mathbf{u}^{(1)} \otimes \mathbf{u}^{(1)})(F) = c_s^2 \left(\tau - \frac{1}{2} \right) \nabla^2 \mathbf{u}^{(1)}(F) + \mathbf{G} \\ &\nabla \cdot \mathbf{u}^{(2)}(F) = 0. \end{aligned} \quad (1.132)$$

Hence, we define the set

$$\begin{aligned} \mathcal{A}^{(3)} = \Big\{ F \in \mathcal{A}^{(2)} \mid f_i^{(3)}(F) &= f_i^* q^{(3)} + c_s^{-2} f_i^* \mathbf{c}_i \cdot \mathbf{v}^{(3)} \\ -\tau \left(\mathbf{c}_i \cdot \nabla f_i^{(2)}(F) + \partial_t f_i^{(1)}(F) - g_i \right), \; &q^{(3)} \in C^\infty(D), \; \mathbf{v}^{(3)} \in (C^\infty(D))^d, \\ &\mathbf{u}^{(1)}(F), \; \rho^{(2)}(F), \; \mathbf{u}^{(2)}(F) \text{ satisfy } (1.132) \Big\}. \end{aligned} \quad (1.133)$$

The procedure can be pursued for any order $k \geq 0$ in the same way. However, to show how the optimization algorithm works for the LBM, it is enough what we have done so far. Formally, the procedure would terminate at this level if we start with the ansatz

$$\mathcal{A}_3 := \{ F \in \mathcal{A}(\eta^d, C^\infty) \mid \forall k \geq 4 : \; f_i^{(k)}(F) = 0 \}.$$

Equations (1.132) already shows a relation with Navier-Stokes, which also justifies the definition (1.90) of $\hat{g}_i$ in connection to the volume forces.

Hydrodynamic equations

Having understood how to proceed, we summarize the main results obtained so far and suggest a shortcut to perform some further step.
To define the ansatz $\mathcal{A}^{(k)}$, we have used a condition on the structure of the coefficients derived from equation (1.111):

$$f_i^{(k)}(F) = f_i^{eq,(k)}(F) - \tau \left(\sum_{l+m=k, l>0} D_{l,i} f_i^{(m)}(F) - g_i^{(k)} \right). \quad (1.134)$$

In order to have $r^{(k)}(F) = 0$, we got supplementary conditions involving the moments of previous orders, in form of a PDE system.

Without describing the computation in detail (see [19] for general results), we give explicitely only the equations for the leading order moments.

In what follows we define the *pressure coefficients*

$$p^{(k)}(F) = c_s^2 \rho^{(k)}(F). \tag{1.135}$$

Additionally, in view of equation (1.132), we fix the relaxation time in order to have $\nu = c_s^2(\tau - \frac{1}{2})$ (ν being the viscosity in the original Navier-Stokes problem). We find that $\mathbf{u}^{(2)}, p^{(3)}$ (suppressing the argument F) solve a homogeneous *Stokes equation*

$$\begin{cases} \nabla \cdot \mathbf{u}^{(2)} = 0 \\ \partial_t \mathbf{u}^{(2)} + \nabla p^{(3)} - \nu \nabla^2 \mathbf{u}^{(2)} = 0. \end{cases} \tag{1.136}$$

For the following order, the couple $(\mathbf{u}^{(3)}, p^{(4)})$ has to be a solution to the following *inhomogeneous Oseen-type problem*:

$$\begin{cases} \nabla \cdot \mathbf{u}^{(3)} = -c_s^{-2} \partial_t p^{(2)} + \dfrac{1}{2} \nabla \cdot \mathbf{G} \\ \partial_t \mathbf{u}^{(3)} + \nabla p^{(4)} + 2\nabla \cdot \left(\mathbf{u}^{(1)} \otimes \mathbf{u}^{(3)} \right) = \nu \nabla^2 \mathbf{u}^{(3)}. \end{cases} \tag{1.137}$$

Remark. The lack of initial data in the previous PDE systems is due to the fact that we are no considering the initial conditions for the algorithm.

Particular choices of the moments Defining the coefficients $f^{(0)}$ and $f^{(1)}$ we did not have any particular restriction on the moments. Nevertheless, we have considered only particular forms of the coefficients, which lead to

$$\rho^{(0)}(F) = 1, \ \mathbf{u}^{(0)}(F) = 0, \ \rho^{(1)}(F) = 0. \tag{1.138}$$

Different definitions are possible, to select the better predictions for $k = 0$ and $k = 1$. However, we observe that the adopted restrictions lead to a relation with the Navier-Stokes equations, which could not be true in more general cases. In the following analysis it has to be checked whether the restrictions (1.138) are fulfilled and do not lead to inconsistencies in the higher orders.

The interior prediction

We define the ansatz $\overset{\circ}{\mathcal{A}}_3$ of the *inner expansions up to the order* h^3, collecting the predictions

$$
\begin{aligned}
&F \in \mathcal{A} \mid \forall t > 0,\ \forall \mathbf{x} \in \Omega : \\
&\quad f_i^{(0)}(F)(t,\mathbf{x}) = f_i^*, \\
&\quad f_i^{(1)}(F)(t,\mathbf{x}) = f_i^* c_s^{-2} \mathbf{c}_i \cdot \mathbf{u}(t,\mathbf{x}), \\
&\quad f_i^{(2)}(F)(t,\mathbf{x}) = f_i^* c_s^{-2} p(t,\mathbf{x}) + f_i^* c_s^{-2} \mathbf{c}_i \cdot \mathbf{v}^{(2)}(t,\mathbf{x}) + H_i^{Q(eq)}(\mathbf{u},\mathbf{u})(t,\mathbf{x}) \\
&\qquad\qquad - \tau f_i^* c_s^{-2} (\mathbf{c}_i \cdot \nabla) \mathbf{c}_i \cdot \mathbf{u}(t,\mathbf{x}), \\
&\quad f_i^{(3)}(F)(t,\mathbf{x}) = f_i^* c_s^{-2} q^{(3)}(t,\mathbf{x}) + f_i^* c_s^{-2} \mathbf{c}_i \cdot \mathbf{v}^{(3)}(t,\mathbf{x}) \\
&\qquad - \tau \left[(\mathbf{c}_i \cdot \nabla) f_i^{(2)}(F)(t,\mathbf{x}) + \left(\frac{(\mathbf{c}_i \cdot \nabla)^2}{2} + \partial_t \right) f_i^{(1)}(F)(t,\mathbf{x}) - g_i^{(3)}(t,\mathbf{x}) \right] \\
&\quad f_i^{(k)}(F)(t,\mathbf{x}) = 0, \quad \text{for } k \geq 4,
\end{aligned}
\tag{1.139}
$$

where the functions $\mathbf{u}$, p are a smooth solution to the Navier-Stokes problem

$$
\begin{cases}
\nabla \cdot \mathbf{u} = 0 \\
\partial_t \mathbf{u} + \nabla p + \mathbf{u} \cdot \nabla \mathbf{u} = \nu \nabla^2 \mathbf{u} + \mathbf{G},
\end{cases}
\tag{1.140}
$$

the couple $(\mathbf{v}^{(2)}, q^{(3)})$ solves the system (1.136)

$$
\begin{cases}
\nabla \cdot \mathbf{v}^{(2)} = 0 \\
\partial_t \mathbf{v}^{(2)} + \nabla q^{(3)} - \nu \nabla^2 \mathbf{v}^{(2)} = 0,
\end{cases}
\tag{1.141}
$$

and $\mathbf{v}^{(3)}$ satisfies

$$
\begin{cases}
\nabla \cdot \mathbf{v}^{(3)} = -c_s^{-2} \partial_t p + \dfrac{1}{2} \nabla \cdot \mathbf{G} \\
\partial_t \mathbf{v}^{(3)} + 2 \nabla \cdot \left(\mathbf{u} \otimes \mathbf{v}^{(3)} \right) = \nu \nabla^2 \mathbf{v}^{(3)}
\end{cases}
\tag{1.142}
$$

((1.137) using $p^{(4)} = 0$).

Theorem 1.1 (LBM inner expansion). *Assume that there exist smooth solutions to the problems* (1.140), (1.141), (1.142). *Then, the ansatz* $\overset{\circ}{\mathcal{A}}_3$ *of the inner expansions is not empty, and it holds*

$$
\forall F \in \overset{\circ}{\mathcal{A}}_3 :\ LBM[F] \in o(h^3),
\tag{1.143}
$$

i.e. the minimum class in $\mathcal{A}$ *is of size* $o(h^3)$.

Proof. The previous derivation proves property (1.143) for the predictions in $\overset{\circ}{\mathcal{A}}_3$. Performing further iterations of the optimization algorithm, the minimum class

will be constructed taking a subset of $\overset{\circ}{\mathcal{A}}_3$. Since the coefficients coincide up to the third order, the difference between two predictions in a subset of $\overset{\circ}{\mathcal{A}}_3$ will be of order $o(h^3)$, and so the size of the resulting minimum class. □

At this point, the problem is still too general to make a more precise statement regarding the classes of precisions. In particular, to have equations determining the coefficients[p] up to $k \leq 3$, assures only that the classes of precision are smaller than $O(h^3)$. The residue of order h^4 reads

$$r_i^{(4)}(F) = -\tau \left[\left(\partial t (\mathbf{c}_i \cdot \nabla) + \frac{(\mathbf{c}_i \cdot \nabla)^3}{6} \right) f_i^{(1)}(F) + \left(\partial_t + \frac{(\mathbf{c}_i \cdot \nabla)^2}{2} \right) f_i^{(2)}(F) + \mathbf{c}_i \cdot \nabla f_i^{(3)}(F) \right]. \quad (1.144)$$

Without more specifications on the moments we are not able to discuss whether there are predictions in $\overset{\circ}{\mathcal{A}}_3$ able to cancel the residue $r^{(4)}$.
For example, observe that the if problems (1.140), (1.141), (1.142) admit zero solutions, then a prediction such that

$$\forall k > 0,\ f_i^{(k)}(F) = 0$$

satisfies equation (1.144).

1.3.3 Asymptotic of the initialization

The analysis of the LBM has to be completed treating the influence of the initial conditions, to derive a precise relation with the Navier-Stokes problem (1.87). Formally, the algorithm to be analyzed is

$$L(h, \hat{f}_h)(n, \mathbf{j}) := \begin{cases} IC(\mathbf{u}_0; h, \hat{f}_h)(\mathbf{j}) & n = 0,\ \mathbf{j} \in \mathcal{G}(h) \\ \mathrm{LBM}(h, \hat{f}_h)(n, \mathbf{j}) & n > 0,\ \mathbf{j} \in \mathcal{G}(h). \end{cases} \quad (1.145)$$

The initialization IC depends on the initial data of the continuous problem, in this case the velocity $\mathbf{u}_0$. We assume that it is consistent with it, i.e.

$$\hat{\mathbf{u}}(0, \mathbf{j}) = h\mathbf{u}_0(\mathbf{x}_\mathbf{j}). \quad (1.146)$$

Note. According to the scaling $\Delta x = h$, $\Delta t = h^2$, using a lattice units reference system (where Δx and Δt are unitary), the physical velocity has to be rescaled by h.

[p] *Determining a coefficient* means, for us, to have a precise equation for it. Observe that if there are multiple solutions for the differential problems (1.140), (1.141), (1.142) we could fall into the technical problems in defining the coefficients.

Equilibrium initial conditions

In this section we focus on the *equilibrium initialization*, defined setting

$$\hat{f}_h(0,\mathbf{j}) = H^{eq}(1, h\mathbf{u}_0(\mathbf{x_j})). \tag{1.147}$$

Observe that (1.147) satisfies (1.146).
This is one of the most popular choices to initialize a lattice Boltzmann simulation. However, the procedure we use can be extended to any kind of initial values. Some example will be considered in the following chapters.
We introduce

$$IC^{EQ}(\mathbf{j})(\mathbf{u}_0; h, f_h) := \hat{f}_h(0,\mathbf{j}) - H^{eq}(1, h\mathbf{u}_0(\mathbf{x_j})) \tag{1.148}$$

into the algorithm (1.145):

$$L(h,\hat{f}_h)(n,\mathbf{j}) := \begin{cases} \hat{f}_h(0,\mathbf{j}) - H_i^{eq}(1, h\mathbf{u}_0(\mathbf{x_j})) & n = 0,\ \mathbf{j} \in \mathcal{G}(h) \\ \mathrm{LBM}(h,\hat{f}_h)(n,\mathbf{j}) & n > 0,\ \mathbf{j} \in \mathcal{G}(h). \end{cases} \tag{1.149}$$

The sequence of better predictions can be constructed similarly as before, considering separately the residues of the two algorithms. However, since theorem 1.1 already gives a precise result regarding the precision of the predictions in the ansatz $\overset{\circ}{\mathcal{A}}_3$ with respect to the interior LBM, if we choose $\overset{\circ}{\mathcal{A}}_3$ as initial ansatz we only need to check the order of residue of the initialization rule.
Since

$$H_i^{eq}(1, h\mathbf{u}_0) = 1 + hc_s^{-2} f_i^* \mathbf{c}_i \cdot \mathbf{u}_0 + h^2 H_i^{Q(eq)}(\mathbf{u}_0, \mathbf{u}_0), \tag{1.150}$$

inserting a prediction from a regular ansatz

$$F_h(n,\mathbf{j}) = \sum_{k\in K} h^k f^{(k)}(t_n, \mathbf{j})$$

into the algorithm (1.148), we found the residue

$$\begin{aligned} IC^{EQ}(\mathbf{u}_0; h, F_h)(\mathbf{j}) = &\left(f_i^{(0)}(F)(0,\mathbf{x_j}) - f_i^*\right) \\ &+ h\left(f_i^{(1)}(F)(0,\mathbf{x_j}) - c_s^{-2} f_i^* \mathbf{c}_i \cdot \mathbf{u}_0(\mathbf{x_j})\right) \\ &+ h^2\left(f_i^{(2)}(F)(0,\mathbf{x_j}) - H_i^{Q(eq)}(\mathbf{u}_0,\mathbf{u}_0)(\mathbf{x_j})\right) \\ &+ \sum_{k\in K,\, k\geq 3} h^k f^{(k)}(F)(0,\mathbf{x_j}) = \\ &= \sum_{k\in K} h^k r_{IC}^{(k)}(F)(\eta_h^d(0,\mathbf{j})). \end{aligned} \tag{1.151}$$

The residue is canceled completely with a prediction $F \in \mathcal{A}$ such that

$$\begin{aligned} f_i^{(0)}(F)(0,\mathbf{x}) &= f_i^* \\ f_i^{(1)}(F)(0,\mathbf{x}) &= f_i^* c_s^{-2} \mathbf{c}_i \cdot \mathbf{u}_0(\mathbf{x}) \\ f_i^{(2)}(F)(0,\mathbf{x}) &= H_i^{Q(eq)}(\mathbf{u}_0,\mathbf{u}_0)(\mathbf{x}), \\ f_i^{(r)}(F)(0,\mathbf{x}) &= 0, \quad r \geq 3, \end{aligned} \tag{1.152}$$

which is therefore a *solution*[q] to the initialization algorithm(1.148).
Substituting order by order the solution for the interior coefficients (1.139) into the residue (1.151) we have

$$\forall F \in \overset{\circ}{\mathcal{A}}_3 : \; r_{IC}^{(0)}(F) = 0, \tag{1.153}$$

$$\forall F \in \overset{\circ}{\mathcal{A}}_3 : \; r_{IC}^{(1)}(F)(\mathbf{x}) = c_s^{-2} f_i^* (\mathbf{u}^{(1)}(0,\mathbf{x}) - \mathbf{u}_0(\mathbf{x})). \tag{1.154}$$

Therefore, the first order residue vanishes in the set

$$\mathcal{A}^{IC,(1)} = \{F \in \overset{\circ}{\mathcal{A}}_3 \mid f_i^{(1)}(F) = f_i^* c_s^{-2} \mathbf{c}_i \cdot \mathbf{u}, \mathbf{u}(0,\mathbf{x}) = \mathbf{u}_0(\mathbf{x})\}, \tag{1.155}$$

i.e. $\mathbf{u}$ is the velocity which solves the Navier-Stokes problem (1.140), with initial condition $\mathbf{u}_0$.
For $k = 2$, we obtain

$$\forall F \in \mathcal{A}^{IC,(1)} : \; r_{IC}^{(2)}(F)(\mathbf{x}) = f_i^* c_s^{-2} p(0,\mathbf{x}) - \tau f_i^* c_s^{-2} \mathbf{c}_i \cdot \nabla \mathbf{u}_0(\mathbf{x}), \tag{1.156}$$

where p is the Navier-Stokes pressure for the interior domain and the condition $\mathbf{u}(0,\mathbf{x}) = \mathbf{u}_0(\mathbf{x})$ has been used.

Initial data depending minimum class We cannot use a general argument anymore, since the result depends on the *initial data* $\mathbf{u}_0$. In fact, $r_{IC}^{(2)}(F) = 0$ only if the initial data are such that

$$p(0,\mathbf{x}) = 0, \quad \nabla \mathbf{u}_0 = 0. \tag{1.157}$$

In case (1.157) is not satisfied, we cannot find better predictions in $\mathcal{A}^{IC,(1)}$, and the optimization algorithm stops. Extending theorem 1.1, we have now

$$\forall F \in \mathcal{A}^{IC,(1)} : \; L[F] \in O(h^2). \tag{1.158}$$

In other words, for the LBM combined with equilibrium initial conditions, there are no way to choose the coefficient $f^{(2)}$ to define a better predictions in the

[q]The fact that we can compute exactly the exact solution of an algorithm should not be surprising, since the initialization rule (1.148) is given as input of the algorithm. However, there are cases (see chapter 3, where the initial values are obtained as result of initialization routines) when an explicit solution for the initial condition is not known.

regular ansatz, i.e. $\mathcal{A}^{IC,(2)} = \emptyset$. The minimum precision class in a regular ansatz, including the initialization rule, has size $O(h^2)$. It has grown, respect to the result obtained with the simple interior LBM. Equivalently, the precision does not depend on the coefficient $f^{(2)}(F)$.
However, if (1.157) holds, then

$$\forall F \in \mathcal{A}^{IC,(1)} : L[F] \in o(h^2). \tag{1.159}$$

***Ideal* initializations**

It would be natural to say, that the equilibrium initial condition does not represent a good choice to initialize the LBM. From the theoretical point of view, one can think of initialization rules which limit the size of the precision classes. For example, we define

Definition 1.12. *An initialization rule IC is a* **second order initialization** *for the interior LBM iff*

- *the minimum class of the resulting algorithm has size $o(h^2)$ in $\mathcal{A}$,*

i.e. the prediction which includes $f^{(2)}$ as specified in the ansatz $\overset{\circ}{\mathcal{A}}_3$ is better than the simple expansion only up to $f^{(1)}$.

The definition implies that, for a second order initialization, the coefficient $f^{(2)}$ defined as the interior coefficient evaluated at time $t = 0$,

$$f_i^{(2)}(F)(0,\mathbf{x}) = f_i^* c_s^{-2} p(0,\mathbf{x}) + H_i^{Q(eq)}(\mathbf{u}_0,\mathbf{u}_0)(\mathbf{x}) - \tau f_i^* c_s^{-2} \mathbf{c}_i \cdot \nabla \mathbf{u}_0(\mathbf{x}) \cdot \mathbf{c}_i, \tag{1.160}$$

selects a set of better predictions.
Similarly, *third order* and *higher order initializations* could be defined. The 1.12 is not an empty definition. Here, we assume the existence of such algorithms. In chapters 2 and 3, we will discuss how to construct better initializations for the LBM.

1.3.4 Prediction of the LB solution

For our purpose, it suffices to have derived the coefficients for $k \leq 3$, and the equations for the hydrodynamic moments appearing in them.

Corollary 1.1 (LBM prediction). *Let denote with $(\mathbf{u}_{NS}, p_{NS})$ a solution to the (periodic) Navier-Stokes problem* (1.87)*, with initial condition*

$$\mathbf{u}_{NS}|_{t=0} = \mathbf{u}_0. \tag{1.161}$$

Let LBM^{IC} be the algorithm combining the LBM and an initialization rule IC. Let assume that, for the initial data $\mathbf{u}_0$, IC is a second order initialization. Defining the ansatz

$$\mathcal{A}^{LBM} = \left\{ F \in \mathcal{A} \mid \forall k \leq 2 : \ f^{(k)}(F) = f^{LBM,(k)}, \quad \forall k > 2 : \ f^{(k)}(F_h) = 0 \right\}, \tag{1.162}$$

with

$$\begin{aligned} f_i^{LBM,(0)} &= f_i^*, \\ f_i^{LBM,(1)} &= f_i^* c_s^{-2} \mathbf{c}_i \cdot \mathbf{u}_{NS}, \\ f_i^{LBM,(2)} &= f_i^* c_s^{-2} p_{NS} + H_i^{Q(eq)}(\mathbf{u}_{NS}, \mathbf{u}_{NS}) - \tau f_i^* c_s^{-2} (\mathbf{c}_i \cdot \nabla) \mathbf{c}_i \cdot \mathbf{u}_{NS}, \end{aligned} \tag{1.163}$$

then,

$$\forall F \in \mathcal{A}^{LBM} : \ LBM^{IC}[F] = o(h^2),$$

i.e. the minimum class in $\mathcal{A}$ with respect to LBM^{IC} has size $o(h^2)$.

Proof. The proof is a straightforward conclusion of the previously derived relations. A second order initialization realizes the conditions (taking the moments of the coefficients (1.163) for $n = 0$)

$$\mathbf{u}^{(1)}(F)(0, \mathbf{x}) = \mathbf{u}_0(\mathbf{x}), \ \mathbf{u}^{(2)}(F)(0, \mathbf{x}) = 0, \tag{1.164}$$

$$p^{(2)}(F)(0, \mathbf{x}) = p_{NS}(0, \mathbf{x}), \ p^{(3)}(F)(0, \mathbf{x}) = 0. \tag{1.165}$$

These relations have to be inserted in the PDEs for the moments derived in theorem 1.1, checking then the order of the resulting residue.
Holding the initial conditions (1.164), (1.165), the field $\mathbf{u}_{NS}, p_{NS}$ satisfy (1.87) (with the initial condition (1.161)). Moreover, the system (1.141) with homogeneous initial condition has only a trivial solution. Since the structure of $f^{(2)}$ and $f^{(3)}$ is such that $\mathbf{u}^{(2)}(F) = 0$ and $\rho^{(3)}(F) = 0$, the hypothesis of theorem 1.1 regarding the moments are satisfied. Thus, the leading orders of the residue of the algorithm LBM vanishes up to h^2.
Performing further steps of the optimization algorithm, the precision of a prediction in $\mathcal{A}$, can only be improved modifying the third order coefficient. Hence, the minimal size of the precision classes is $o(h^2)$. Since this conclusion holds for the initialization rule (by assumption), the statement is true also for the combined algorithm. □

Remark. Observe that if the initial data are such that the problem (1.87) admits a trivial solution, the prediction ansatz such that

$$f^{(0)}(F) = f^*, \quad \forall k > 0 : \ f^{(k)}(F) = 0$$

is a solution to the algorithm.

The truncated expansion

The last result means that, applying the algorithm to find the minimum to the regular ansatz $\mathcal{A}$ (truncated at some order K), we would arrive (but, in general, do not stop) at the ansatz $\mathcal{A}^{\mathrm{LBM}}$. A more detailed analysis would be necessary, to check whether we can find better predictions in the ansatz $\mathcal{A}^{\mathrm{LBM}}$. However, for our first applications it is enough to have assured that the minimal precision class is of size $o(h^2)$. In fact, having relations regarding the coefficients up to the second order, we can consider the *truncated expansion*

$$\hat{F}_i(n, \mathbf{j}) = \sum_{k=0}^{2} h^k f_i^{LBM,(k)}(t_n, \mathbf{x_j}) \tag{1.166}$$

(a general prediction in the ansatz $\mathcal{A}^{\mathrm{LBM}}$), neglecting the successive terms which would specify the class of the minimum in the ansatz.
The idea is to use the truncated expansion as a *prediction* (which is meant now, as a function which *can predict*) of the lattice Boltzmann *solution* $\hat{f}$.
In particular, since suitable moments of $\hat{F}_i$ yield the Navier-Stokes solution $\mathbf{u}_{NS}$, p_{NS}, we conclude that the corresponding moments of $\hat{f}_i$ approximate these fields:

$$\frac{1}{h}\hat{\mathbf{u}} = \frac{1}{h}\sum_i \hat{f}_i \mathbf{c}_i = \mathbf{u}_{NS} + O(h^2) \tag{1.167}$$

$$\hat{p} := c_s^2 \frac{\sum_i \hat{f}_i - 1}{h^2} = p_{NS} + O(h). \tag{1.168}$$

Since we have stopped at the second order coefficients, the uncertainty on the third order coefficient appears on the remainders of equations (1.167)-(1.168).
As additional result, we can extract

$$\hat{\mathbf{S}}[\mathbf{u}] = -\frac{1}{\tau c_s^2 h^2} \sum_i \mathbf{c}_i \otimes \mathbf{c}_i \left(\hat{f}_i - f_i^{eq}(\hat{f}) \right) \tag{1.169}$$

as a first order approximation of the (*rate-of-strain*) tensor

$$\mathbf{S}[\mathbf{u}] \equiv \nabla \mathbf{u} + \nabla \mathbf{u}^T.$$

Numerical tests - the *Taylor-Vortex* solution

To test the procedure and the results derived before, we use a simple analytic smooth solution of the Navier-Stokes equations (1.87), on a periodic square.
Namely, the test problems are based on the two-dimensional *periodic Taylor vortex field*. Given a, $b \in \mathbb{R}$, on the periodic domain

$$\Omega = \left[0, \frac{2\pi}{a}\right] \times \left[0, \frac{2\pi}{b}\right],$$

we consider

$$\begin{aligned} u_{TV}^{a,b}(t,x,y) &= -\frac{U_0}{a}\cos(ax)\sin(by)\exp\left(-\nu(a^2+b^2)t\right) \\ v_{TV}^{a,b}(t,x,y) &= \frac{U_0}{b}\sin(ax)\cos(by)\exp\left(-\nu(a^2+b^2)t\right) \end{aligned} \tag{1.170}$$

$$p_{TV}^{a,b}(t,x,y) = -\frac{U_0^2}{4}\left(\frac{1}{a^2}\cos(2ax)+\frac{1}{b^2}\cos(2by)\right)\exp\left(-2\nu(a^2+b^2)t\right). \tag{1.171}$$

as solutions to different problems:

- TV-NS: Navier-Stokes,
- TV-ST: Stokes, using a *linearized* equilibrium function[r] and *adding a volume force* $\mathbf{G} = \nabla p_{TV}$ in equation (1.87).

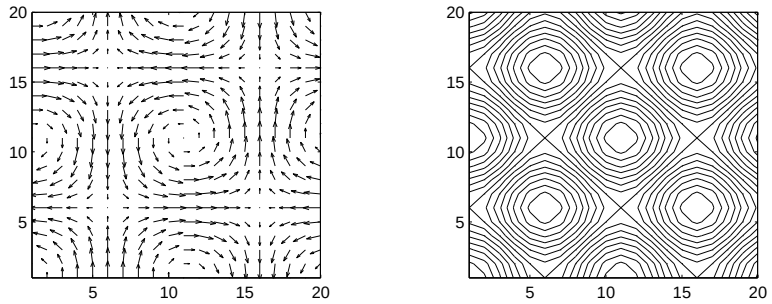


Figure 1.9: **Left**. Streamlines of the vortex field $\mathbf{u}_{TV}$, for $a = b = 2\pi$. **Right**. Level curves of the initial pressure (1.171).

The initial conditions in the practical simulations are defined through the *equilibrium initialization* (equation (1.148))

$$\hat{f}_i(0,\mathbf{j}) = H_i^{eq}(1, h\mathbf{u}_0(\mathbf{j})) \tag{1.172}$$

(the initial condition of problem (1.87) has been rescaled by h in the lattice unities reference system).
As we have already observed, the hypothesis of corollary 1.1 combining initial data and initialization rule is not satisfied, using (1.172), unless the initial pressure p_0 and the initial velocity $\mathbf{u}_0$ satisfy

$$p_0 \equiv 0, \quad \nabla \mathbf{u}_0 = 0. \tag{1.173}$$

Therefore, we perform two different tests.

[r] An analogous derivation shows that, using a linear equilibrium

$$f^{L(eq)}(f) = H_i^{L(eq)}(\rho(f)), \mathbf{u}(f))$$

yields to the solution to a Stokes problem.

Homogeneous initial conditions Consider the problem [19]

$$\begin{cases} \nabla \cdot \mathbf{u} = 0 \\ \partial_t \mathbf{u} + \nabla p + \mathbf{u} \cdot \nabla \mathbf{u} = \nu \nabla^2 \mathbf{u} + \mathbf{G}_\alpha \\ \mathbf{u}(0, \mathbf{x}) = 0, \end{cases} \tag{1.174}$$

in $\Omega \times [0, T]$, where

$$\mathbf{G}_\alpha = \dot{\alpha} \mathbf{u}_{TV} + \alpha(\alpha - 1)\mathbf{u}_{TV} \cdot \nabla \mathbf{u}_{TV}, \tag{1.175}$$

and

$$\alpha(t) = t^3. \tag{1.176}$$

An analytic solution is given by

$$\mathbf{u}_\alpha(t, \mathbf{x}) = \alpha(t)\mathbf{u}_{TV}(t, \mathbf{x}), \quad p_\alpha(t, \mathbf{x}) = \alpha(t) p_{TV}(t, \mathbf{x}). \tag{1.177}$$

Relations (1.173) holds.
The following pictures show the errors

$$E^{\mathbf{u}}(t_n) = \max_{\mathbf{j} \in \mathcal{G}(h)} \|\hat{\mathbf{u}}(n, \mathbf{j})) - \mathbf{u}_\alpha(t_n, \mathbf{x}_\mathbf{j})\|, \quad E^p(t_n) = \max_{\mathbf{j} \in \mathcal{G}(h)} \|\hat{p}(n, \mathbf{j}) - p_\alpha(t_n, \mathbf{x}_\mathbf{j})\|$$

in velocity and in pressure.

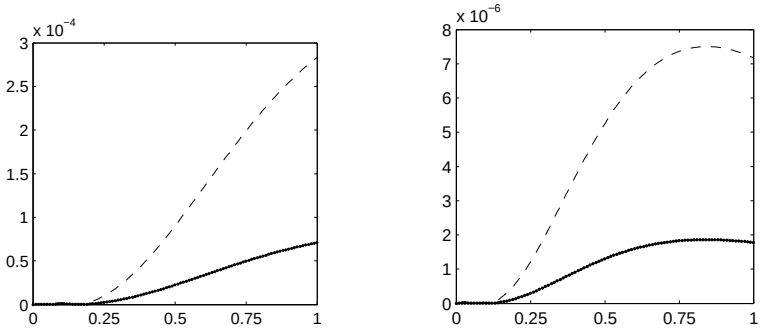


Figure 1.10: Comparison between the numerical solutions for $\mathbf{u}$ and p and the moments of the prediction F of equation (1.166), for the problem (1.174). Two grids, $h = 0.05$ (dashed line) and $h = 0.025$ (bold line) are compared. **Left:** Maximum error $E^{\mathbf{u}}$ in velocity versus time. **Right:** Maximum error in pressure E^p.

The behavior of the errors in figure 1.10 and the order plot in figure 1.11 show that the results are second order accurate. Actually, for the pressure we achieved an even better accuracy than the one predicted by (1.168). It depends (see [19]) on the problem and on the particular choice of the initial condition.

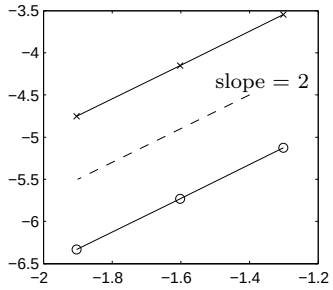


Figure 1.11: Tests over several grids, shows the experimental order of convergence of the numerical results. Double logarithmic plots (in a $\log_{10}$ scale) of maximum error versus grid size, for velocity ($\times$) and pressure ($\circ$) are drawn, together with a reference line of slope 2.

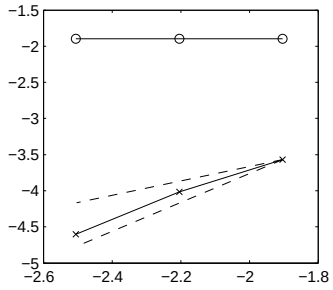


Figure 1.12: LBM used to simulate the TV-NS solution, without the smoothing factor $\alpha(t)$. As in figure 1.11, the order plot for velocity ($\times$) and pressure ($\circ$) are drawn. Now, velocity is less accurate, pressure is inconsistent. Reference slopes 1 and 2 are also drawn (dashed lines).

General IC: arising of initial layers Now we look at the case $\mathbf{u}_0 \not\equiv 0$.
Figures 1.12 shows a clear reduction of the accuracy (first order in velocity and inconsistent pressure). The maximum error in pressure, in figure 1.13, is affected by an *initial layer*, which is responsible for the loss of precision.
In the following chapters, we investigate more deeply the influence of the initialization on the accuracy of the results.

1.3.5 Outline of the asymptotic expansion technique

As a conclusion, we summarize the main ideas used to perform the asymptotic analysis, providing a set of practical tools regarding the application to the LBM. We have derived precise results concerning the ansatz $\mathcal{A}^{\mathrm{LBM}}$. The idea is to use it as *basis ansatz* for further investigations and extensions of the standard scheme. Namely, if we can justify the use of the prediction F^{LBM}, we will recover the

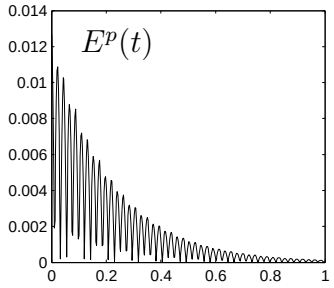


Figure 1.13: Maximum error in pressure E^p, versus time. Compared to figure 1.10, an *initial layer* affects the result.

accuracies for pressure and velocity predicted for the interior domain.
The effects of new algorithms will be discussed with respect to the consequences they have on the ansatz $\mathcal{A}^{\mathrm{LBM}}$, the predictions and the classes of precision.

Coupling of LBM with other algorithms For example, let assume to have an algorithm

$$L(h,\hat{f}_h)(\mathbf{j}) := \begin{cases} \mathrm{LBM}(h,\hat{f}_h)(\mathbf{j}) & \mathbf{j}\in\mathbb{D}_h^1 \\ M(h,\hat{f}_h)(\mathbf{j}) & \mathbf{j}\in\mathbb{D}_h^2. \end{cases} \tag{1.178}$$

which combines the LBM and another scheme M, on two subset of the discrete domain $\mathbb{D}_h$.
The analysis of the initialization rules in equation (1.145) is an example, where we use the interior LBM, for $n>0$, plus an additional algorithm for $n=0$. We want to generalize the procedure used in section 1.3.3.
We start the analysis with the ansatz $\mathcal{A}^{\mathrm{LBM}}$. Practically, we *insert* the expressions of the coefficients $f^{\mathrm{LBM},(k)}$ into the expressions for the residues

$$r_M^{(k)}(F) = 0 \tag{1.179}$$

of the algorithm M. If, for some $k\le 2$,

$$\nexists F\in\mathcal{A}^{\mathrm{LBM}} :\ r_M^{(k)}(F)=0,$$

the optimization algorithm terminates at step[s] k.

A theoretical result Observe that if

$$M[F^{\mathrm{LBM}}] = 0$$

[s]As for the initialization, concerning the algorithm L defined in equation (1.178), we can say that M is a *coupling of order* k for the interior LBM, iff the interior coefficients evaluated in $\mathbb{D}_h^2$ defines a set of better predictions for L.

we do not need to perform additional analysis on the algorithm M, since the total residue, for the prediction F^{LBM}, coincides with the residue

$$\mathrm{LBM}[F].$$

Lemma 1.10 (Self-compatibility). *For all $F \in X$, there exists an algorithm L_F such that*

$$L_F[F] = 0.$$

Proof. Simply, set

$$L_F(h, \hat{f}_h) = \hat{f}_h - F_h.$$

□

Despite the apparently trivial proof, the lemma contains useful elements, and suggests an analytical approach towards the construction of new features for the LBM.
According to lemma 1.10, we can construct an algorithm whose results can be predicted using $F^{LBM}(n, \mathbf{j})$ just setting the value of the previous prediction point by point.

Corollary 1.2. *Let $(n_*, \mathbf{j}_*) \in \mathbb{D}_h$, we define*

$$LBM^*(h, \hat{f}_h)(n, \mathbf{j}) := \begin{cases} LBM(h, \hat{f}_h)(n, \mathbf{j}) & (n, \mathbf{j}) \neq (n_*, \mathbf{j}_*) \\ \displaystyle\sum_{k \leq 2} h^k f^{LBM,(k)}(t_{n_*}, \mathbf{x}_{\mathbf{j}_*}) & (n, \mathbf{j}) = (n_*, \mathbf{j}_*). \end{cases}$$

Then, the precision classes in the regular ansatz respect to LBM^* are of size $o(h^2)$ (as in corollary 1.1).

In practice we will follow this idea, trying to construct extension to the LBM in order to have the same prediction as the interior algorithm, up to the second order.

Example 1.5. *For instance, regarding the initial conditions, we have that*

$$IC^{LBM}(h, \hat{f}_h)(\mathbf{j}) = \hat{f}(0, \mathbf{j}) - F^{LBM}(0, \mathbf{j})$$

is a second order initialization*. In other words, the coefficients up to the second order are initialized according to the interior ones evaluated at time $t = 0$ (see equation* (1.160)*).*

Correction of the coefficients As a final remark, let us assume to have derived a prediction

$$\hat{F}_h = \sum_k h^k f^{(k)},$$

for the solution to a certain algorithm L in a considered $\mathcal{A}$.
Suppose that the coefficients do not correspond to our expectations.
Instead, we would prefer

$$\tilde{F}_h = \sum_k h^k \tilde{f}^{(k)}.$$

Can we find an algorithm $\tilde{L}$, in such a way to obtain $\tilde{F}$ as prediction?
The scheme (1.180) sketches the way to follow. To construct a new *corrected* scheme $\tilde{L}$, we act directly on the equations determining the coefficients, defining the modifications in the relations for $f^{(k)}(F)$, in such a way that they are now satisfied by $\tilde{F}$, instead of by F.

$$\begin{array}{ccc} L_h & & \tilde{L}_h \\ \downarrow & & \uparrow \\ f^{(k)} & \leftrightarrow & \tilde{f}^{(k)} \end{array} \tag{1.180}$$

Subsequently, walking back the procedure used to go from L to $f^{(k)}$, we try to translate such modifications in a *corrected algorithm* $\tilde{L}$.
Regarding the LBM, this *target prediction* corresponds to the truncated expansion F^{LBM}. However, the idea of diagram (1.180) will be successfully applied also in other situations.

Summary

In this chapter we have set the basis of the asymptotic expansion technique using a formal description, which clarifies the objects we are working with. Although the whole machinery could appear complicate, reductions and simplifications are in many cases possible.
As an example, the presented argument has been applied to the particular case of the LBM for periodic boundary, recovering the approximation of the Navier-Stokes equations as it appears in the standard literature.
However, with the present formulation we are able to extend, in a straightforward way, the application of the asymptotic analysis also to other features of the scheme.
In particular, we remark the results presented in corollary 1.2, which offers a hint to develop the LBM towards new features. Its application will be shown later on in the thesis.

Chapter 2

Initial layers and Multiscale expansion

The behaviors observed in the previous benchmarks motivate a further refinement of the asymptotic for the LBE. Actually, the appearance of initial layer is a common phenomenon related to LB simulations. Recently, a class of *initialization routines* (see [34], the analysis reported in [5] and in the following chapter 3) has been developed, trying to improve the numerical results reducing the size of the oscillating layers via a better LB initialization. In this chapter we focus on the characteristics of the initial layers.

Such oscillations have a numerical origin and are due to the artificial compressibility of the LBM. The regular asymptotic analysis of the LBM [19], (whose results have been presented in the previous chapter) is restricted to the scaling $\Delta t = \Delta x^2$, which leads in the continuous limit to the approximation of the *incompressible* Navier-Stokes equations. To catch compressibility effects (even if of numerical origin) further improvements are needed.

Scope of this chapter is to illustrate how, modifying the regular ansatz, we can apply in this case the asymptotic expansion technique to understand in depth the cause and the behavior of the initial layers affecting the LB results. The approach shows the flexibility and the variety of possibilities offered by the asymptotic analysis to describe the numerical method.

In section 2.1, we introduce the definition of layers and irregular expansions. Through some examples, we will show in which sense the regular ansatz lacks of elements. The notion of ansatz will be extended more in general, enlarged to *irregular functions*, in order to allow the prediction of compressible phenomena. To explain the method, we perform a detailed two-scales expansion in section 2.2, and a three-scales in section 2.3, predicting in practical examples the initial layers arising in the numerical results.

2.1 Need of irregular expansions

We classify as *irregular behaviors* any phenomenon which the analysis based on the regular ansatz (1.101) is not able to catch. In other words, the predictions in the regular ansatz are only quite poor approximations of $\hat{f}$ (big sizes of the precision classes).
For example, if the initial conditions are *inaccurate*, we observed additional oscillations of *numerical origin* which mix with the results. We took this example as a starting point to extend further the asymptotic analysis.

2.1.1 Equilibrium initial conditions

Without describing the topic from a general point of view, we illustrate the technique focusing on the equilibrium initial conditions (reporting equation (1.172))

$$\hat{f}_h(0, \mathbf{j}) = H^{eq}(1, h\mathbf{u}_0(\mathbf{x_j})) \tag{2.1}$$

(where $\mathbf{u}_0$ is the initial data for the velocity in physical units).
The results using initialization (2.1) in the problem TV-NS (same test as in the previous chapter) are shown in figure 2.1-2.2.
The analysis of the equilibrium initial conditions presented in the previous chapter (section 1.3.3), reveals that the trouble are due to the the inconsistent initialization of the second order coefficient. To have a second order initialization it should be initialized according to the interior coefficient $f^{LBM,(2)}$ at time $t = 0$:

$$f_i^{LBM,(2)}(0, \mathbf{x}) = H_i^{eq,(2)}(1 + h^2 c_s^{-2} p_0(\mathbf{x}), h\mathbf{u}_0(\mathbf{x})) - \tau f_i^* c_s^{-2} (\mathbf{c}_i \cdot \nabla)\mathbf{c}_i \cdot \mathbf{u}_0(\mathbf{x}),$$

p_0 being the initial Navier-Stokes pressure.

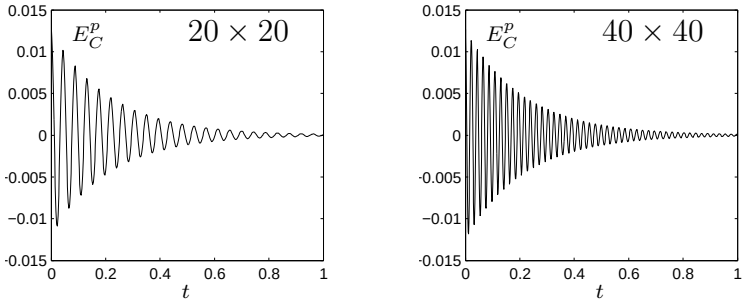


Figure 2.1: Initial layer in pressure. $E_C^p(t)$ is the value $\hat{p} - p_{TV}$ in the central point $\mathbf{x}_C$ of the unit square (where p_{TV} takes its maximum), initializing with (2.1). The exact initial value of pressure is ~ 0.012. The initial layer is of the same order of the pressure, since it does not decrease refining the grid.

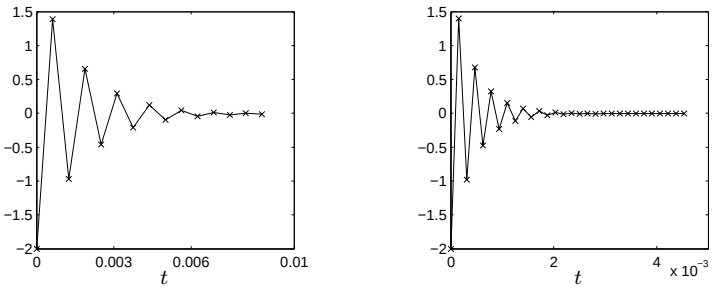


Figure 2.2: Typical initial layers in $\mathbf{S}[\mathbf{u}]$, initializing with (2.1). Error $E_M^{S_{xx}}(t_n)$, in the component S_{xx} evaluated at the point $x_M = (0.25, 0.25)$, is drawn, computed with grid 40×40 (**left**) and 80×80 (**right**). The crosses ($\times$) show the value at every time step. The damping depends on the number of iterations, not on the physical time.

For general initial data, initialization (2.1) (as well as *any initialization* based on the equilibrium function) will not be a second order initialization. As a consequence, the precision of a prediction does not depend on the second order coefficient (corollary 1.1 does not hold). Therefore, the approximation of pressure and stress tensor (contained in the second order coefficient) are no longer assured. At this stage, there are two possible ways to approach the problem. From a practical point of view, having identified the cause in the inconsistent initial conditions, a better initialization could be set up. This topic will be discussed in chapter 3. On the other hand, one could try to understand the irregular behavior of the scheme in more detail, looking for better predictions in a bigger ansatz.

2.1.2 Extending the regular ansatz

So far, we only regarded the LBM as a numerical solver for the Navier-Stokes equations. In the regular ansatz the coefficients of the predictions only depend on the physical variables. Using the asymptotic expansion approach, we now try to describe also the *compressible* scale, even if the resulting predictions do not correspond to physical behaviors.

Keeping the structure $\mathcal{A} = \mathcal{A}(y, \mathcal{C})$, we can include more time and space variables, through a different coordinate-map (instead of $(n, \mathbf{j}) \mapsto (t_n, \mathbf{x_j})$) or extend the the class $\mathcal{C}$. Any ansatz, constructed differently from definition 1.11, will be named **irregular expansion**. A simple sketch of the relevant domains is given in figure 2.3.

Observe that, without the diffusive coupling, in the transformation between discrete and continuous variables, the arising functions are not well defined in the limit $h \to 0$, performed according to $\Delta t = h^2 \Delta x$, even if, for any $h > 0$, they are.

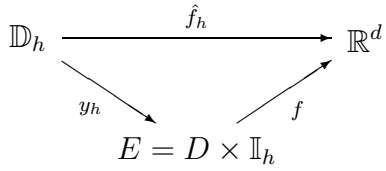


Figure 2.3: Extension of the diagram in figure 1.8 for the irregular ansatz. Allowing more freedom for the map y_h, we do not assume the mapping into D. In general, the sets $y_h(\mathbb{D}_h)$ will only *contain* the continuous domain, times additional *irregular* spaces $\mathbb{I}_h$.

Example 2.1. *Let us consider*

$$\eta_h(n, \mathbf{j}) := \big(n, nh, nh^3, nh^2, h\mathbf{j}\big) = (n, s_n, r_n, t_n, \mathbf{x}_\mathbf{j}),$$

within a LBM. In a physical simulation for time $t \leq T$:

$$D = [0, T] \times \Omega,$$

the algorithm is defined for $N_T = Th^{-2}$ time steps, and we have $\mathbb{D}_h = [0, N_T] \times \mathcal{G}(h)$. Hence, for any h,

$$\eta_h(\mathbb{D}_h) \subset \left[0, \frac{T}{h^2}\right] \times \left[0, \frac{T}{h}\right] \times [0, Th] \times D = \mathbb{I}_h \times D.$$

Note that, for $h \to 0$, some components of the sets $\mathbb{I}_h$ are growing, while others are shrinking.

2.2 Two-scales expansion

We want to investigate the dynamic of the oscillatory layer appearing in the numerical pressure as shown in figure 2.1. Inspired by the test case, we modify the original expansion adding a dependence on a *faster* time $s = hn$.

Definition 2.1. *Using $E = \mathbb{R} \times \mathbb{R} \times \Omega$ and the coordinate map*

$$\begin{array}{rccc} \eta_h^2(n, \mathbf{j}): & \mathbb{D}_h & \to & E \\ & (n, \mathbf{j}) & \mapsto & (h^2 n, hn, h\mathbf{j}) \quad := (t_n, s_n, \mathbf{x}_\mathbf{j}), \end{array} \tag{2.2}$$

we define the **two-scales ansatz** $\mathcal{A}^2 = \mathcal{A}(\eta^2, C^\infty(E))$*, whose predictions have the general form*

$$F_h(n, \mathbf{j}) \equiv \sum_{k \in K} h^k f_i^{(k)}(\eta_h^2(n, \mathbf{j})), \tag{2.3}$$

with functions $f_i^{(k)}(t, s, \mathbf{x})$ smooth.

As we did before, given a prediction $F \in \mathcal{A}^2$, we introduce the projections

$$\begin{array}{rccc} f^{(k)}: & \mathcal{A}^2 & \to & C^\infty \\ & F & \mapsto & f^{(k)}(F) \end{array}$$

such that $f^{(k)}(F)$ is the k-th coefficient of the expansion of F. The functions $f^{(k)}(F)$ are defined generalizing to the coordinate map (2.2) the limit processes used in equation (1.100) for the regular case.

Residue for the new ansatz

Formally, we consider the algorithm

$$L(h, \hat{f}_h)(n, \mathbf{j}) := \begin{cases} \hat{f}_h(0, \mathbf{j}) - H^{eq}(1, h\mathbf{u}_0(\mathbf{x}_\mathbf{j})) & n = 0,\ \mathbf{j} \in \mathcal{G}(h) \\ \mathrm{LBM}(h, \hat{f}_h)(n, \mathbf{j}) & n > 0,\ \mathbf{j} \in \mathcal{G}(h) \end{cases} \tag{2.4}$$

inserting a general two-scales expansion expansion into it and sorting the different order in h. Regarding the interior part, for $n > 0$, we obtain (compare with (1.102))

$$\mathrm{LBM}_i(h, F_h)(n+1, \mathbf{j}) = \sum_{k \in K} h^k \left(h^2 \partial_t + h\partial_s + h\mathbf{c}_i \cdot \nabla + \ldots\right) f_i^{(k)}(F)(t_n, s_n, \mathbf{x}_\mathbf{j}) -$$

$$\sum_{k \in K} h^k \left[\frac{1}{\tau}\left(f_i^{eq,(k)}(F)(t_n, s_n, \mathbf{x}_\mathbf{j}) - f_i^{(k)}(F)(t_n, s_n, \mathbf{x}_\mathbf{j})\right) - g_i^{(k)}(t_n, \mathbf{x}_\mathbf{j})\right] + \ldots \tag{2.5}$$

As before, the equilibrium of order k reads

$$f_i^{eq,(k)} = H_i^{L(eq)}(\rho^{(k)}, \mathbf{u}^{(k)}) + \sum_{l+m=k} H_i^{Q(eq)}(\mathbf{u}^{(l)}, \mathbf{u}^{(m)}),$$

$\rho^{(k)}(F)$ and $\mathbf{u}^{(k)}(F)$ being defined as the moments of the coefficient $f^{(k)}(F)$. Introducing

$$\hat{D}_{l,i} = \sum_{2a+b=l} \frac{\partial_t^a}{a\,!} \frac{(\partial_s + \mathbf{c}_i \cdot \nabla)^b}{b!}, \tag{2.6}$$

which is a straightforward extension of the previous operator (1.112) in presence of an additional time variable, we can write equation (2.5) in the form

$$\mathrm{LBM}_i(h, F_h)(n+1, \mathbf{j}) = \sum_{k \in K} h^k r_i^{(k)}(F)(\eta^2(n, \mathbf{j})) \tag{2.7}$$

with[a]

$$r_i^{(k)}(F) = \sum_{l+m=k,\, l>0} \hat{D}_{l,i} f_i^{(m)}(F) - \frac{1}{\tau}\left(f_i^{eq,(k)}(F) - f_i^{(k)}(F)\right) + g_i^{(k)}. \tag{2.8}$$

[a]Originating from the same algorithm, equations (1.111) and (2.8) exemplify the dependence of the residues on the coordinate map.

Besides the equations for the interior part, we have to consider the initialization rule of (2.4). The computation has been already performed in section 1.3.3. Using the notation

$$IC^{EQ}(h,\hat{f}_h)(\mathbf{j}) = \hat{f}_h(0,\mathbf{j}) - H^{eq}(1, h\mathbf{u}_0(\mathbf{x_j})),$$

the residue of a prediction F reads

$$\begin{aligned} IC_i^{EQ}(h,F_h)(\mathbf{j}) &= f_i^{(0)}(F)(0,0,\mathbf{x_j}) - f_i^* \\ &+ h\left(f_i^{(1)}(F)(0,0,\mathbf{x_j}) - f_i^* c_s^{-2}\mathbf{c}_i\cdot\mathbf{u}_0(\mathbf{x_j})\right) \\ &+ h^2\left(f_i^{(2)}(F)(0,0,\mathbf{x_j}) - H_i^{Q(eq)}(\mathbf{u}_0,\mathbf{u}_0)(\mathbf{x_j})\right) \\ &+ \sum_{k\geq 3} h^k f_i^{(k)}(F)(0,0,\mathbf{x_j}), \end{aligned} \tag{2.9}$$

which we can rewrite as

$$IC^{EQ}(h,F_h)(\mathbf{j}) = \sum_{k\in K} h^k r_{IC}^{(k)}(F)(\mathbf{x_j}). \tag{2.10}$$

2.2.1 Projected algorithm

In this section, we want to focus on the phenomena affecting the numerical results for pressure (i.e. density) and velocity. Therefore, instead of the LBM (2.4) we will rather consider an algorithm whose output are the moments of the variable $\hat{f}_h$. It can be described by the following sequence of operations:

$$\begin{aligned} \hat{f}_i(n+1,\mathbf{j}+\mathbf{c}_i) &= \hat{f}_i(n,\mathbf{j}) + J_i(\hat{f}(n,\mathbf{j})), \\ \hat{\rho}_h(n+1,\mathbf{j}) &= \rho(\hat{f}_h(n+1,\mathbf{j})) = \sum_i \hat{f}_i(n+1,\mathbf{j}), \\ \hat{\mathbf{u}}_h(n+1,\mathbf{j}) &= \mathbf{u}(\hat{f}_h(n+1,\mathbf{j})) = \sum_i \hat{f}_i(n+1,\mathbf{j})\mathbf{c}_i, \end{aligned} \tag{2.11}$$

(denoting with $J_i(\hat{f})$ the collision operator). In words, we run the standard LBM, extracting then $\hat{\rho}$ and $\hat{\mathbf{u}}$ projecting the results on the moment space.
Using a compact notation (obtained substituting the first equation of (2.11) into the others), we define the *projected algorithm*

$$\begin{aligned} \mathrm{MOM}(h,\hat{\rho}_h,\hat{\mathbf{u}}_h,\hat{f}_h) &= \begin{bmatrix} \mathrm{MOM}_\rho \\ \mathrm{MOM}_{\mathbf{u}} \end{bmatrix} = \\ &= \begin{bmatrix} \hat{\rho}_h(n+1,\mathbf{j}) - \sum_i \hat{f}_i(n,\mathbf{j}-\mathbf{c}_i) + J_i(\hat{f}(n,\mathbf{j}-\mathbf{c}_i)) \\ \hat{\mathbf{u}}_h(n+1,\mathbf{j}) - \sum_i \left(\hat{f}_i(n,\mathbf{j}-\mathbf{c}_i) + J_i(\hat{f}(n,\mathbf{j}-\mathbf{c}_i))\right)\mathbf{c}_i \end{bmatrix}. \end{aligned} \tag{2.12}$$

Observe that, since $\hat{\rho}_h = \rho(\hat{f}_h)$ and $\hat{\mathbf{u}}_h = \mathbf{u}(\hat{f}_h)$, the principal variables are still the distributions $\hat{f}_h$. A prediction for $\hat{\rho}$ (or $\hat{\mathbf{u}}$) will be derived taking the moment of a prediction F for the standard LBM:

$$\rho_h(F) = \rho(F_h) = \sum_i F_{ih}.$$

If F_h is an asymptotic expansion, so will be also the predictions $\rho_h(F)$ and $\mathbf{u}_h(F)$. In fact

$$\rho_h(F) = \sum_i \left(\sum_{k\in K} h^k f_i^{(k)}(F) \right) = \sum_{k\in K} h^k \rho^{(k)}(F)$$

shows that the coefficients of the inherited expansion for the moments are the moments of the coefficients of F_h:

$$\rho^{(k)}(F) = \sum_i f_i^{(k)}(F).$$

Projected residues

In practice, we are interested in expansions able to cancel the residue of the algorithm MOM. It can be related to the LBM-residue. We have

$$\mathrm{LBM}_i(h, F_h)(n+1, \mathbf{j}) = \sum_{k\in K} h^k r_i^{(k)}(F)(n, \mathbf{j}),$$

which means, for all $(n, \mathbf{j}) \in \mathbb{D}_h$,

$$F_{ih}(n+1, \mathbf{j}) = F_{ih}(n, \mathbf{j} - \mathbf{c}_i) + J_i(F_h(n, \mathbf{j} - \mathbf{c}_i)) + \sum_{k\in K} h^k r_i^{(k)}(F). \tag{2.13}$$

Taking the moment of order zero of both sides leads to

$$\rho_h(F)(n+1, \mathbf{j}) - \sum_i F_{ih}(n, \mathbf{j} - \mathbf{c}_i) + J_i(F_h(n, \mathbf{j} - \mathbf{c}_i)) = \sum_i \sum_{k\in K} h^k r_i^{(k)}(F). \tag{2.14}$$

The left hand side is the residue of the prediction $\rho(F)$ in the algorithm MOM_ρ. Thus we can write

$$\mathrm{MOM}_\rho(h, \rho_h(F)) = \sum_{k\in K} r_\rho^{(k)}(F), \tag{2.15}$$

where

$$r_\rho^{(k)}(F) = \sum_i r_i^{(k)}(F). \tag{2.16}$$

Analogously, for the velocity we have

$$\mathrm{MOM}_{\mathbf{u}}(h, \mathbf{u}_h(F)) = \sum_{k\in K} r_{\mathbf{u}}^{(k)}(F), \tag{2.17}$$

with

$$r_{\mathbf{u}}^{(k)}(F) = \sum_i r_i^{(k)}(F)\mathbf{c}_i. \tag{2.18}$$

In conclusion,

Proposition 2.1. *The residue of the projected LBM is the projection of the LBM-residue.*

We call the vector

$$r_{\mathrm{MOM}}^{(k)}(F) = \begin{bmatrix} r_\rho^{(k)}(F) \\ r_{\mathbf{u}}^{(k)}(F) \end{bmatrix} \tag{2.19}$$

the *projected residue* of order k.
Investigating the algorithm for the moments, we are interested only in finding better prediction for MOM. Hence, we will look for predictions which cancel the projected residue.
Remark. Observe that, since $r_{\mathrm{MOM}}^{(k)}$ is a projection on the moments space of $r^{(k)}(F)$, it holds

$$r^{(k)}(F) = 0 \Rightarrow r_{MOM}^{(k)}(F) = 0,$$

but the vice versa is not true. It might happen that a certain prediction F cancels the projected residues, but not the single components.
Taking the moments of equation (2.8) for a general k, we obtain PDE systems for the coefficients of the moments in the interior domain:

$$\begin{aligned} \partial_t \rho^{(k-2)} + \nabla \cdot \mathbf{u}^{(k-1)} + \partial_s \rho^{(k-1)} + \frac{1}{2}\left(\nabla \otimes \nabla \mathbf{p}^{(k-2)} + \partial_s^2 \rho^{(k-2)}\right) + \\ + \partial_s(\nabla \cdot \mathbf{u}^{(k-2)}) = \sum_i \left(g_i^{(k)} - \sum_{\substack{l+m=k \\ m<k-2}} \hat{D}_l f_i^{(m)}(F) \right) \end{aligned} \tag{2.20}$$

$$\begin{aligned} \partial_t \mathbf{u}^{(k-2)} + \nabla \cdot \mathbf{p}^{(k-1)} + \partial_s \mathbf{u}^{(k-1)} + \frac{1}{2}\sum_i \mathbf{c}_i (\mathbf{c}_i \cdot \nabla)^2 f_i^{(k-2)} + \partial_s^2 \mathbf{u}^{(k-2)} + \\ + \partial_s(\nabla \cdot \mathbf{p}^{(k-2)}) = \sum_i \mathbf{c}_i \left(g_i^{(k)} - \sum_{\substack{l+m=k \\ m<k-2}} \hat{D}_l f_i^{(m)}(F) \right). \end{aligned} \tag{2.21}$$

In the previous, we have introduced the pressure tensor of order k:

$$\mathbf{p}^{(k)} := \sum_i f_i^{(k)} \mathbf{c}_i \otimes \mathbf{c}_i. \tag{2.22}$$

Similarly, projecting the residue (2.10) for the initialization yields

$$\begin{aligned} r_{IC,\rho}^{(0)}(F) &= \rho^{(0)}(F) - 1, \quad \forall k > 0 : r_{IC,\rho}^{(k)}(F) = 0 \\ r_{IC,\mathbf{u}}^{(1)}(F) &= \mathbf{u}_0, \quad \forall k \neq 1 : r_{IC,\mathbf{u}}^{(k)}(F) = 0. \end{aligned} \tag{2.23}$$

Remark. If for a certain k the projected residues depends only on the moments of the coefficients (as it happens for the lower orders), the prediction which cancels it will be a function only of the moments as well.

2.2.2 Preparing the ansatz

To process the two-scale ansatz for the projected LBM, we apply the optimization algorithm as we did for the regular case. Some initial considerations allow to simplify the analysis. Observe that in addition to the regular case, the residue (2.8) contains also the derivatives with respect to the faster time variable. Moreover, if some coefficient does not depend on s we can partially recycle the results of the computation performed in the previous case.

$\mathbf{k=0}$, $\mathbf{k=1}$. Keeping the regularity

Let us denote with $(\mathbf{u}_{NS}, p_{NS})$ a solution of a periodic incompressible Navier-Stokes, with initial condition $\mathbf{u}_{NS}(0,\mathbf{x}) = \mathbf{u}_0(\mathbf{x})$. We start with a regular *two-scales* ansatz $\mathcal{A}^2$ (equation (2.3)). Observe that the predictions in the ansatz (not s-depending)

$$\mathcal{A}^{2,(1)} = \{F \in \mathcal{A}^2 \mid f_i^{(0)}(F) = f_i^*,\ f_i^{(1)}(F)(t,\mathbf{x}) = f_i^* c_s^{-2} \mathbf{c}_i \cdot \mathbf{u}_{NS}(t,\mathbf{x})\},$$

cancel the leading orders of the residue (2.8) for the LBM (as in section 1.3.2) and for the equilibrium initial conditions (section 1.3.3). Therefore, the projected residues vanish as well. We have

$$\begin{aligned} \forall F \in \mathcal{A}^{2,(1)} : \rho^{(0)}(F) = 1,\ \rho^{(1)}(F) = 0, \\ \mathbf{u}^{(0)}(F) = 0,\ \mathbf{u}^{(1)}(F)(t,\mathbf{x}) = \mathbf{u}_{NS}(t,\mathbf{x}), \end{aligned} \tag{2.24}$$

which are the same conditions on the leading order moments which we had in the regular case. The choice of such initial ansatz can be justified by the fact that the initial layer we want to investigate appears first in the second order in h.

$\mathbf{k=2}$. The irregular moments

Regarding the following order, an argument analogous to the one discussed in section the 1.3.2 can be used to show that the interior LBM-residue is removed by a prediction $F \in \mathcal{A}^{2,(1)}$ such that

$$\begin{aligned} f_i^{(2)}(F)(t,s,\mathbf{x}) = f_i^* c_s^{-2} \left(p_{NS}(t,\mathbf{x}) + p_e^{(2)}(t,s,\mathbf{x})\right)) + f_i^* c_s^{-2} \mathbf{c}_i \cdot \mathbf{u}_e^{(2)}(t,s,\mathbf{x}) \\ + H_i^{Q(eq)}(\mathbf{u}_{NS}, \mathbf{u}_{NS})(t,\mathbf{x}) - \tau \mathbf{c}_i \cdot f_i^{(1)}(F)(t,\mathbf{x}). \end{aligned} \tag{2.25}$$

where $p_e^{(2)}$ and $\mathbf{u}_e^{(2)}$ are smooth field and p_{NS} is the Navier-Stokes pressure.

The coefficient in equation (2.25) has the same structure as the one defined in equation (1.127) for the regular ansatz. The only difference is the dependence on the new time scale, which adds a new parameter to the predictions. We have split the pressure in $f_i^{(2)}$ to evidence the Navier-Stokes part p_{NS} and the additional error p_e, which depends on s. It holds

$$\rho^{(2)}(F)(t,s,\mathbf{x}) = c_s^{-2}\left(p_{NS}(t,\mathbf{x}) + p_e^{(2)}(t,s,\mathbf{x})\right).$$

Concerning the initial condition, the projected residue reads

$$\begin{aligned} r_{IC,\rho}^{(2)}(F)(\mathbf{x}) &= \rho^{(2)}(F)(0,0,\mathbf{x}) = c_s^{-2}\left(p_{NS}(0,\mathbf{x}) + p_e^{(2)}(0,0,\mathbf{x})\right), \\ r_{IC,\mathbf{u}}^{(2)}(F)(\mathbf{x}) &= \mathbf{u}_e^{(2)}(0,0,\mathbf{x}). \end{aligned} \tag{2.26}$$

Therefore, we select the better predictions defining the ansatz

$$\begin{aligned} \mathcal{A}^{2,(2)} = \Big\{ F \in \mathcal{A} \mid f_i^{(2)}(F)(t,s,\mathbf{x}) &= f_i^* c_s^{-2}\left(p_{NS}(t,\mathbf{x}) + p_e^{(2)}(t,s,\mathbf{x})\right) \\ &+ f_i^* c_s^{-2}\mathbf{c}_i \cdot \mathbf{u}_e^{(2)}(t,s,\mathbf{x}) + H_i^{Q(eq)}(\mathbf{u}_{NS},\mathbf{u}_{NS})(t,\mathbf{x}) - \tau \mathbf{c}_i \cdot f_i^{(1)}(F)(t,\mathbf{x}), \\ p_e^{(2)} \in C^\infty(E),\ \mathbf{u}_e^{(2)} \in (C^\infty(E))^d, p_e^{(2)}(0,0,\mathbf{x}) &= -p_{NS}(0,\mathbf{x}),\ \mathbf{u}_e^{(2)}(0,0,\mathbf{x}) = 0 \Big\}. \end{aligned} \tag{2.27}$$

Remark. Within the extended ansatz, using the *new degrees of freedom* $p_e^{(2)}$ and $\mathbf{u}_e^{(2)}$, we are able to find a suitable second order coefficient to define better predictions (which reduces the size of the minimum class).
In particular, the function $p_e^{(2)}$ allows to set a *zero initial pressure* in the second order coefficient.

2.2.3 Initial layer equations

We are interested in the prediction of the new parameters $p_e^{(2)}$ and $\mathbf{u}_e^{(2)}$. They are fixed by the differential equations (2.20)-(2.21) recovered taking the projection of the residues of higher orders, with initial conditions (2.23).

k = 3. Dynamic of initial layers

Using properties (2.24), equations (2.21)-(2.21) for $k = 3$ give

$$\begin{aligned} \partial_s p_e^{(2)} &= -c_s^2 \nabla \cdot \mathbf{u}_e^{(2)} \\ \partial_s \mathbf{u}_e^{(2)} &= -\left(\partial_t \mathbf{u}^{(1)} + \nabla p_{NS}^{(2)} + \nabla \cdot (\mathbf{u}^{(1)} \otimes \mathbf{u}^{(1)}) - \nu \nabla^2 \mathbf{u}^{(1)} - \mathbf{G}\right) = -\nabla p_e^{(2)} \end{aligned} \tag{2.28}$$

(where $\mathbf{u}^{(1)} = \mathbf{u}_{NS}$ has been used).

Decoupling the previous system, we have two *wave-type* equations:

$$\begin{aligned} \partial_s^2 p_e^{(2)} &= c_s^2 \nabla^2 p_e^{(2)} \\ \partial_s^2 (\nabla \cdot \mathbf{u}_e^{(2)}) &= c_s^2 \nabla^2 (\nabla \cdot \mathbf{u}_e^{(2)}). \end{aligned} \tag{2.29}$$

We focus on the equation for $p_e^{(2)}$.
The ansatz defined in equation (2.27) defines the initial conditions

$$\begin{aligned} p^{(2)}(0,0,\mathbf{x}) &= -p_{NS}(0,\mathbf{x}) = -p_0(\mathbf{x}), \\ \partial_s p^{(2)}(0,0,\mathbf{x}) &= \nabla \cdot \mathbf{u}_e^{(2)}(0,0,\mathbf{x}) = 0. \end{aligned} \tag{2.30}$$

However, to close problem (2.29), conditions for $p^{(2)}(0,t,\mathbf{x})$ (initial values in s) and the first s-derivative $\partial_s p^{(2)}(0,t,\mathbf{x})$ for general $t > 0$ are needed. We introduce **unknown t-initial conditions**:

$$\begin{aligned} p_e^{(2)}(0,t,\mathbf{x}) &= P_{in}(t,\mathbf{x}), \\ \partial_s p_e^{(2)}(0,t,\mathbf{x}) &= Q_{in}(t,\mathbf{x}), \end{aligned} \tag{2.31}$$

with P_{in} and Q_{in} to be determined. According to (2.30), we have

$$P_{in}(0,\mathbf{x}) = -p_0(\mathbf{x}). \tag{2.32}$$

$$Q_{in}(0,\mathbf{x}) = 0. \tag{2.33}$$

The unknown initial values appear because the time scales t, s are coupled in the expansion, since $t_n = hs_n$. In particular, at the initial time step $n = 0$, both take the value zero. However, in performing the analysis they are independent variables. Additional relations for the new unknowns will be derived from the equation for the coefficients of higher orders.
Remark. Equation (2.32) shows that an inconsistent initial condition in pressure is related to a source of an oscillating error $p_e^{(2)}$. In practice, one should therefore use the correct initial pressure. Obviously, if (2.31) were both homogeneous, the system (2.29) would have the zero solution.

Solutions in a particular case

For particular structures of the initial data of the problem, and consequently of the functions $P_{in}(t,\mathbf{x})$ and $Q_{in}(t,\mathbf{x})$, some solution of (2.29) can be constructed. Let us consider an initial pressure p_0 such that

$$p_0(\mathbf{x}) = \sum_{r=1}^{M} \epsilon_r^0(\mathbf{x}) \tag{2.34}$$

where the $\epsilon_r^0(\mathbf{x})$ are *eigenfunctions* of the Laplacian with (different) negative eigenvalues, such that

$$\forall r = 1, \ldots, M : \ \nabla^2 \epsilon_r^0(\mathbf{x}) = -\lambda_r^2 \epsilon_r^0(\mathbf{x}) \tag{2.35}$$

$$\forall s \neq r : \ \lambda_s \neq \lambda_r. \tag{2.36}$$

Recalling the relation

$$P_{in}(0, \mathbf{x}) = -p_0(\mathbf{x}),$$

we assume the unknown initial data to be of the form

$$P_{in}(t, \mathbf{x}) = \sum_{r=1}^{M} \epsilon_r(t, \mathbf{x}) \tag{2.37}$$

where the $\epsilon_r(t, \mathbf{x})$ are eigenfunctions of the *spatial* Laplacian with negative eigenvalues.

Definition 2.2. *A function satisfying (2.35)-(2.37), will be called an* **M-frequency** *function, with frequencies* $(\lambda_1, \ldots, \lambda_M)$.

Definition 2.3. *An M-frequency function* $P(t, \mathbf{x})$ *is called* **separable**, *if it can be expressed as*

$$P(t, \mathbf{x}) = \sum_{r=1}^{M} \Phi_r(t) p_r(\mathbf{x}). \tag{2.38}$$

We have in mind the construction of solution for initial pressure formally expandable in a Fourier series in space,

$$p_0(\mathbf{x}) = \sum \hat{p}_{\mathbf{k}} \mathrm{Re}\left(\exp\left(-i\mathbf{k} \cdot \mathbf{x}\right)\right),$$

extending the sum (2.34) for $M \to \infty$. However, to keep the argument free of technicalities, only finite sums will be considered.

Single-frequency solution If the initial Navier-Stokes pressure is an eigenfunction of the Laplacian:

$$\nabla^2 p_{NS}(0, \mathbf{x}) = -\lambda^2 p_{NS}(0, \mathbf{x}),$$

we look for solutions in the form

$$p_e^{(2)}(t, s, \mathbf{x}) = h(s, t)\epsilon(t, \mathbf{x}), \tag{2.39}$$

where

$$\nabla^2 \epsilon = -\lambda^2 \epsilon.$$

In this case, the initial values for $s = 0$,

$$\begin{aligned} P_{in}(t, \mathbf{x}) &= h(0, t)\epsilon(t, \mathbf{x}) \\ Q_{in}(t, \mathbf{x}) &= \partial_s h(0, t)\epsilon(t, \mathbf{x}), \end{aligned} \tag{2.40}$$

are 1-frequency functions. From the first equation of (2.29), $h(s,t)$ solves

$$\partial_s^2 h(s,t) = -c_s^2 \lambda^2 h(s,t). \tag{2.41}$$

Hence, the *single frequency solution* is

$$p_e^{(2)}(t,s,\mathbf{x}) = \cos(\eta s + \Theta(t))\epsilon(t,\mathbf{x}), \quad \text{with } \eta = c_s\lambda. \tag{2.42}$$

Including the constraints (2.32)-(2.33) for $t=0$:

$$\begin{array}{l} P_{in}(0,\mathbf{x}) = -p_{NS}(0,\mathbf{x}) = \cos(\Theta(0))\epsilon(0,\mathbf{x}) \\ Q_{in}(0,\mathbf{x}) = 0 = \sin(\Theta(0))\epsilon(0,\mathbf{x}), \end{array} \tag{2.43}$$

we have

$$\epsilon(0,\mathbf{x}) = -p_{NS}(0,\mathbf{x}), \quad \Theta(0) = 0 \tag{2.44}$$

(of course, it can be chosen $\epsilon(0,\mathbf{x}) = p_{NS}(0,\mathbf{x})$, $\Theta(0) = \pi$).

Multiple-frequency solution More generally, let us assume

$$p_{NS}(0,\mathbf{x}) = \sum_{r=1}^{M} p_r(\mathbf{x}), \quad \text{with } \nabla^2 p_r = -\lambda_r^2 p_r. \tag{2.45}$$

In this case, we search a solution of (2.29) in which the pressure is a sum

$$p_e(t,s,\mathbf{x}) = \sum_{r=1}^{M} h_r(s,t)\epsilon_r(t,\mathbf{x}) \tag{2.46}$$

and

$$\nabla^2 \epsilon_r = -\lambda_r^2 \epsilon_r.$$

The s-initial conditions P_{in}, Q_{in} are now M-frequency functions.
Since equation (2.29) is linear, we can work distinctly on the different frequencies, solving systems analogous to (2.41) for the different $r = 1, \ldots, M$. The *multiple frequency solution* reads

$$p_e^{(2)}(t,s,\mathbf{x}) = \sum_{r=1}^{M} \cos(\eta_r s + \Theta_r(t))\epsilon_r(t,\mathbf{x}), \quad \text{with } \eta_r = c_s\lambda_r. \tag{2.47}$$

From (2.28) the velocity $\mathbf{u}_e^{(2)}$ can be computed:

$$\mathbf{u}_e^{(2)}(t,s,\mathbf{x}) = -\sum_{r=1}^{M} \frac{1}{\eta_r} \sin(\eta_r s + \Theta_r(t))\nabla\epsilon_r(t,\mathbf{x}). \tag{2.48}$$

Now, t-initial values give

$$\begin{aligned} P_{in}(0,\mathbf{x}) = -\sum_{r=1}^{M} p_r(\mathbf{x}) = \sum_{r=1}^{M} \cos(\Theta_r(0))\epsilon_r(0,\mathbf{x}) \\ Q_{in}(0,\mathbf{x}) = 0 = \sum_{r=1}^{M} \sin(\Theta_r(0))\epsilon_r(0,\mathbf{x}), \end{aligned} \tag{2.49}$$

which can be solved by

$$\Theta_r(0) = 0, \quad \epsilon_r(0,\mathbf{x}) = -p_r(\mathbf{x}), \quad r = 1,\dots,M. \tag{2.50}$$

Slowly increasing terms To recover the functions $\epsilon_r(t,\mathbf{x})$, $\Theta_r(t)$, we have to derive further information from the following order equations. The ansatz $\mathcal{A}^{2,(3)}$ has to be constructed choosing $p_e^{(2)}$ and $\mathbf{u}_e^{(2)}$ which cancel the projected residue $r_{\mathrm{MOM}}^{(3)}(F)$. However, it can be not enough in a general case. For example, if the time interval is *unbounded*, it has to be assured also that the following order residue $r_{\mathrm{MOM}}^{(4)}$ does not increase up to the order $O\left(\frac{1}{h}\right)$, producing a residue in the algorithm of the same order as $r_{\mathrm{MOM}}^{(3)}$. In the present case, it might happen that P_{in}, Q_{in} generate slowly varying t-depending terms , of size increasing in time, which become $O\left(\frac{1}{h}\right)$ in a long time range $t \leq T \in O\left(\frac{1}{h}\right)$ (**secular terms**).

Example 2.2. *Let us consider the equation*

$$\partial_s^2 y^{(3)} = -y^{(3)} - \Phi \sin s. \tag{2.51}$$

A solution of (2.51) is given by

$$y^{(3)}(t,s) = Y_0 \sin s + \frac{\Phi}{2} s \cos s.$$

For $\Phi \neq 0$, $y^{(3)}$ contains a secular term.

Hence, we look for conditions on the functions $P_{in}(t,\mathbf{x})$ and $Q_{in}(t,\mathbf{x})$ in order to keep the residues of the following order bounded.

k = 4. Secular terms

Without enter into the details of the computations, we consider equations (2.20)-(2.21), for $k = 4$. First, we recall a result regarding the ansatz $\overset{\circ}{\mathcal{A}}_3$ constructed for the interior LBM in the regular case (theorem 1.1). We define, for $F \in \overset{\circ}{\mathcal{A}}_3$,

$$\mathbf{u}_{reg}^{(3)}(F) = \sum_i f_i^{(3)}(F),$$

as the moments of the regular third order coefficient. We have (equation (1.142)) that

$$\begin{aligned} &\partial_t p_{NS}^{(2)} + c_s^2 \nabla \cdot \mathbf{u}_{reg}^{(3)} = \frac{1}{2} \nabla \cdot \mathbf{G} \\ &\partial_t \mathbf{u}_{reg}^{(3)} + 2\nabla \cdot \left(\mathbf{u}_{NS} \otimes \mathbf{u}_{reg}^{(3)} \right) = \nu \nabla^2 \mathbf{u}_{reg}^{(3)}. \end{aligned} \tag{2.52}$$

Coming back to the irregular expansion, we define the field

$$\mathbf{u}_e^{(3)}(t, s, \mathbf{x}) = \mathbf{u}^{(3)}(t, s, \mathbf{x}) - \mathbf{u}_{reg}^{3)}(t, \mathbf{x}), \tag{2.53}$$

which contains the s-depending part of the third order velocity.
Using relation (2.52), and calling

$$p_e^{(3)} = c_s^2 \rho^{(3)}$$

we end up with the following system of PDEs for the irregular functions:

$$\begin{aligned} &\partial_s p_e^{(3)} + \partial_t p_e^{(2)} + c_s^2 \nabla \cdot \mathbf{u}_e^{(3)} = 0 \\ &\partial_s \mathbf{u}_e^{(3)} + \partial_t \mathbf{u}_e^{(2)} + \nabla p_e^{(3)} + 2\nabla \cdot \left(\mathbf{u}_{NS} \otimes \mathbf{u}_e^{(2)} \right) = \nu \left(\nabla^2 \mathbf{u}_e^{(2)} + \nabla (\nabla \cdot \mathbf{u}_e^{(2)}) \right). \end{aligned} \tag{2.54}$$

Combining the equations (2.54) and inserting also relations (2.28) between $p_e^{(2)}$ and $\mathbf{u}_e^{(2)}$, we have

$$\partial_s^2 p_e^{(3)} + c_s^2 \nabla^2 p_e^{(3)} = -2\partial_s \left(\partial_t p_e^{(2)} - \nu \nabla^2 p_e^{(2)} \right) + 2c_s^2 \nabla \cdot \mathbf{U}_{1,2}. \tag{2.55}$$

We have substituted

$$\mathbf{U}_{1,2} = \nabla \cdot \left(\mathbf{u}_{NS} \otimes \mathbf{u}_e^{(2)} \right).$$

Excluding the sources We do not need to solve equation (2.55). To avoid secular terms, it is enough to force the right hand side to be zero. It can be proven that the solutions of the resulting system are bounded in time. Hence, we derive explicitly the sources for the secular terms in equation (2.55), considering the *M-frequency solution* (2.47)-(2.48):

$$\begin{aligned} 2c_s^2 \nabla \cdot \mathbf{U}_{1,2} &= 2 \left(\nabla \mathbf{u}_{NS} : (\nabla \mathbf{u}_e^{(2)})^T + \mathbf{u}_{NS} \cdot \nabla (\nabla \cdot \mathbf{u}_e^{(2)}) \right) = \qquad (2.56) \\ &= \sum_{r=1}^{M} \eta_r \sin(\eta_r s + \Theta_r) 2\lambda_r^{-2} \left(\nabla \mathbf{u}_{NS} : \nabla \otimes \nabla \epsilon_r - \lambda_r^2 \mathbf{u}_{NS} \cdot \nabla \epsilon_r \right) = \\ &= 2 \sum_{r=1}^{M} \eta_r \sin(\eta_r s + \Theta_r) 2\lambda_r^{-2} \left(\nabla \cdot (\mathbf{u}_{NS} \cdot \nabla \otimes \nabla \epsilon_r) \right) \qquad (2.57) \end{aligned}$$

(since $\mathbf{u}_{NS}$ is divergence-free, and using the properties of an M-frequency solution),

$$\begin{aligned} -2\partial_s\left(\partial_t - \nu\nabla^2\right) p_e^{(2)} &= 2\sum_{r=1}^{M} \eta_r \sin(\eta_r s + \Theta_r)\left(\partial_t \epsilon_r + \nu\lambda_r^2 \epsilon_r\right) \\ &- 2\sum_{r=1}^{M} \eta_r \dot{\Theta}_r \cos(\eta_r s + \Theta_r)\epsilon_r. \end{aligned} \tag{2.58}$$

Imposing the right hand side of equation (2.55) to be zero, frequency by frequency, and including the relations (2.50) for $t = 0$, we get the systems:

$$\begin{aligned} &\partial_t \epsilon_r = -\nu\lambda_r^2 \epsilon_r + 2\lambda_r^{-2}\left(\nabla \cdot (\mathbf{u}_{NS} \cdot \nabla \otimes \nabla \epsilon_r)\right), \\ &\epsilon_r(0, \mathbf{x}) = -p_r(0, \mathbf{x}) \end{aligned} \tag{2.59}$$

$$\dot{\Theta}_r = 0, \quad \Theta(0) = 0. \tag{2.60}$$

Regarding the phases, it allows to conclude

$$\Theta_r(t) = 0, \quad r = 1, \ldots, M \tag{2.61}$$

hence $Q_{in}(t, \mathbf{x}) = 0$. Therefore, the problem is reduced to the determination of

$$P_{in} = \sum_{r=1}^{M} \epsilon_r,$$

recovering the remaining unknowns from the PDEs (2.59)[b].

Correct initial pressure Using an initialization which includes the correct initial pressure, is equivalent to set homogeneous initial conditions for the functions ϵ_r in the system (2.59), which has therefore zero solution, i.e. $P_{in} = Q_{in} = 0$. As a consequence, also the initial condition (2.31) for the system (2.29) are homogeneous and the irregular pressure $p_e^{(2)}$ vanishes.

Linear problem

A particular solution can be found restricting the analysis to the Stokes equation. Here, the quadratic term $\mathbf{U}_{1,2}$ drops out and equation (2.59) becomes

$$\partial_t \epsilon_r = -\nu\lambda_r^2 \epsilon_r, \quad r = 1, \ldots, M\,. \tag{2.63}$$

[b]For general initial conditions, the needed equations have to be derived from the condition of zero sources

$$\partial_t(\nabla \cdot \mathbf{u}^{(2)}) = \nu\nabla^2(\nabla \cdot \mathbf{u}_e^{(2)}) + 2\left(\nabla \mathbf{u}_{NS} : (\nabla \mathbf{u}_e^{(2)})^T + \mathbf{u}_{NS} \cdot \nabla(\nabla \cdot \mathbf{u}_e^{(2)})\right). \tag{2.62}$$

Involving only the time derivative, such equation can be solved for unknown initial conditions in a *separable* form

$$P_{in}(t,\mathbf{x}) = \sum_{r=1}^{M} \Phi_r(t) p_r(\mathbf{x}).$$

In fact, the second order solution reads

$$p_e^{(2)}(t,s,\mathbf{x}) = \sum_{r=1}^{M} \cos(\eta_r s)\Phi_r(t) p_r(\mathbf{x}), \tag{2.64}$$

$$\mathbf{u}_e^{(2)}(t,s,\mathbf{x}) = -\sum_{r=1}^{M} \frac{1}{\eta_r} \sin(\eta_r s)\Phi_r(t) \nabla p_r(\mathbf{x}). \tag{2.65}$$

Equations (2.50)-(2.63), which yield

$$\dot{\Phi}_r(t) = -\nu\lambda_r^2 \Phi_r(t), \quad \Phi(0) = -1, \tag{2.66}$$

fix the time-depending functions

$$\Phi_r(t) = -\exp(-\nu\lambda_r^2 t), \quad r = 1,\ldots,M. \tag{2.67}$$

In conclusion, we find the prediction for the initial layer

$$p_e^{(2)}(t,s,\mathbf{x}) = -\sum_{r=1}^{M} \cos(\eta_r s)\exp(-\nu\lambda_r^2 t) p_r(\mathbf{x}). \tag{2.68}$$

Dynamic of initial layer. Equation (2.29), derived at the beginning, connects the dynamic of the oscillatory layer to the spatial variation in general. In the particular case of equation (2.68) we have a complete description of the initial layer. It is composed by s-oscillatory terms and viscosity dependent t-decaying factors, both related to the spatial frequencies of the initial error

2.2.4 Prediction of initial layers

The Taylor vortex solution

As a test for the results obtained in section 2.2.3, we use the vortex solution of problems TV-ST or TV-NS (introduced in section 1.3.4). The exact initial pressure can be written as

$$p_0^{TV}(\mathbf{x}) = p_a(x) + p_b(y), \tag{2.69}$$

with

$$\nabla^2 p_a(x) = -4a^2 p_a(x), \quad \nabla^2 p_b(y) = -4b^2 p_b(y). \tag{2.70}$$

Note that it is a one-frequency function if $a = b$, with $\lambda = 2a$ and a two-frequencies functions if $a \neq b$.
Initializing the lattice Boltzmann with the equilibrium distribution (2.1), we have

$$p_e^{(2)}(0, 0, \mathbf{x}) = -p_0^{TV}(\mathbf{x}). \tag{2.71}$$

Navier-Stokes problem: single frequency For the sake of simplicity, regarding the full Navier-Stokes problem, we only experiment with the test case $a = b = 2\pi$, on the (periodic) domain $\Omega = [0, 1)^2$. The initial pressure has the property

$$\left(\nabla \otimes \nabla p_0^{TV}\right)_{\alpha,\beta} = -4a^2 p_0^{TV} \delta_{\alpha,\beta}. \tag{2.72}$$

As a consequence

$$\nabla \cdot (\mathbf{u}_{NS}(0, \mathbf{x}) \cdot \nabla \otimes \nabla \epsilon(0, \mathbf{x})) = -4a^2 p_0^{TV} \nabla \cdot \mathbf{u}_{NS}(0, \mathbf{x}) = 0, \tag{2.73}$$

which allows us to simplify equation (2.59) as

$$\partial_t \epsilon(t, \mathbf{x}) = -4a^2 \nu \, \epsilon(t, \mathbf{x}). \tag{2.74}$$

The latter admits a solution of the form $\epsilon(t, \mathbf{x}) = \Phi(t) p_0^{TV}(\mathbf{x})$, where (from (2.71))

$$\begin{aligned} \dot{\Phi} &= -4a^2 \nu \Phi \\ \Phi(0) &= -1. \end{aligned} \tag{2.75}$$

In conclusion, we have the prediction, for the initial layer in pressure:

$$p_e^{(2)}(s, t, \mathbf{x}) = -\cos(2ac_s s) \exp(4a^2 \nu t) p_0^{TV}(\mathbf{x}). \tag{2.76}$$

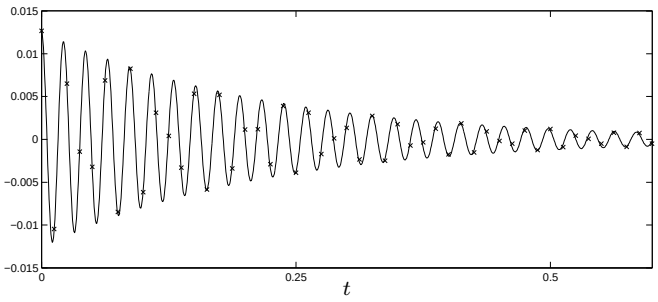


Figure 2.4: Problem TV-NS, with $a = b = 2\pi$. Prediction of the error in pressure (in the central point, as shown in figure 2.1), for $\tau = 0.56$, and related viscosity $\nu = 0.02$, after initialization (2.1). The continuous line represents the predicted value, using (2.76). Computation on grid 40×40, with numerical results ($\times$) sampled every $10th$ iteration of the algorithm.

The results are shown in figure 2.4. The behavior of the error (in a selected point) is drawn, together with the prediction (2.76). We observe a good agreement between them.

Stokes problem: M-frequency In the linear problem TV-ST, we use the initial pressure

$$p_0^{TV,a,b}(\mathbf{x}) = p_a(x) + p_b(y), \tag{2.77}$$

as particular case of

$$p_0(\mathbf{x}) = \sum_{q=1}^{M} p_0^{TV,a_q,b_q}(\mathbf{x}) = \sum_{r=1}^{2M} p_r(\mathbf{x}), \tag{2.78}$$

summing vortices with different frequencies.
According to (2.64), the behavior of the initial layer is described by

$$p^{(2)}(t,s,\mathbf{x}) = -\cos(2ac_s s)\mathrm{e}^{-4a^2\nu t}p_a(\mathbf{x}) - \cos(2bc_s s)\mathrm{e}^{-4b^2\nu t}p_b(\mathbf{x}). \tag{2.79}$$

In figure 2.5, the prediction (2.79) is compared with the numerical result after an equilibrium initialization.
Moreover, the known leading order term of the error can be subtracted from the numerical results, to improve the approximation. Defining

$$\hat{\hat{p}}(n,\mathbf{j}) = \hat{p}(n,\mathbf{j}) - p_e^{(2)}(h^2 n, hn, h\mathbf{j}) \tag{2.80}$$

(doing the same for velocity), figure 2.6 shows that a better accuracy is reached.

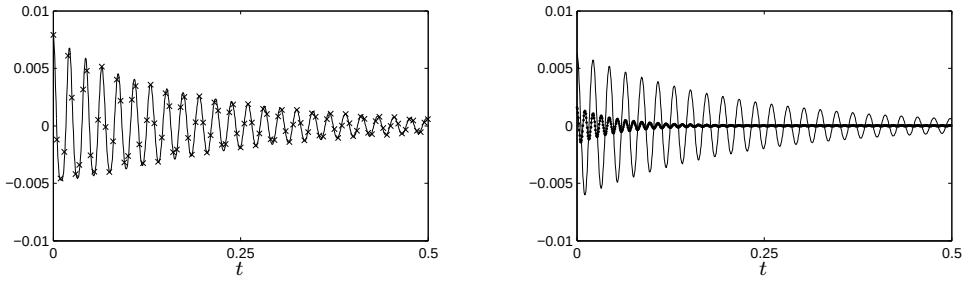


Figure 2.5: Layer in pressure for problem TV-ST, with $a = 2\pi$, $b = 4\pi$. The prediction (2.79) (solid lines) reproduces the numerical error (sampled by $\times$, at iterations h^{-1}, $2h^{-1}$,...). Grid is 40×40. Note the presence of another frequency (compare with figure 2.4) in the fast oscillations, which is rapidly damped. On the right, the two components of the prediction are superimposed.

Finite modes-Stokes problem

Following the same path, we perform some numerical test using a more general M-frequency initial pressure.
We consider

$$p(t,\mathbf{x}) = \sum_{\mathbf{k}\in I} \hat{p}_{\mathbf{k}}(t)\mathrm{e}^{i\mathbf{k}\cdot\mathbf{x}}, \tag{2.81}$$

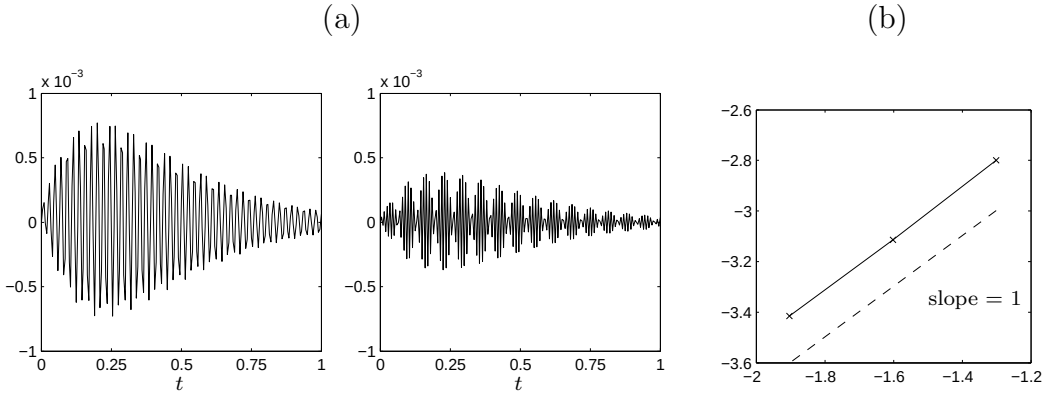


Figure 2.6: **(a)** Problem TV-NS: Error in pressure after the result has been corrected as proposed in equation (2.80). Plot for grid 40×40 (**left**) and 80×80 (**right**). **(b)** Double logarithmic plot of the error in pressure versus grid size, removing the predicted initial layer. Accuracy has been improved.

written as a complex sum of eigenfunctions of the spatial Laplacian, for a finite set $I \subset \mathbb{Z}^2$. It can be viewed as a solution of a Stokes problem including a volume force

$$\mathbf{G}(t, \mathbf{x}) = \sum_{\mathbf{k} \in I} \hat{\mathbf{g}}_{\mathbf{k}}(t) \mathrm{e}^{i\mathbf{k}\cdot\mathbf{x}} \tag{2.82}$$

(*M-frequency force*) which satisfies

$$\nabla^2 p = \nabla \cdot \mathbf{G}, \tag{2.83}$$

i.e. with coefficients $\hat{\mathbf{g}}_{\mathbf{k}}$ such that

$$|\mathbf{k}|^2 \hat{p}_{\mathbf{k}} = -i(\mathbf{k} \cdot \hat{\mathbf{g}}_{\mathbf{k}}). \tag{2.84}$$

Assuming equilibrium initialization with zero pressure, the prediction of the initial layer gives

$$p_e^{(2)}(t, s, \mathbf{x}) = \sum_{\mathbf{k} \in I} -\cos(c_s |\mathbf{k}| s) \hat{p}_{\mathbf{k}}(t) \mathrm{e}^{-\nu|\mathbf{k}|^2 t} \mathrm{e}^{i\mathbf{k}\cdot\mathbf{x}}. \tag{2.85}$$

We test the previous result on a Stokes problem with exact pressure

$$p_0(x, y) = \frac{(\sin(2\pi(x+y)) - \cos(2\pi(x+y)))}{4\pi} - \frac{\cos(6\pi x)}{2\pi},$$

which is generated by a force

$$\mathbf{G}(x, y) = \begin{pmatrix} \cos(2\pi(x+y)) + 3\sin(6\pi x) \\ \sin(2\pi(x+y)) \end{pmatrix}.$$

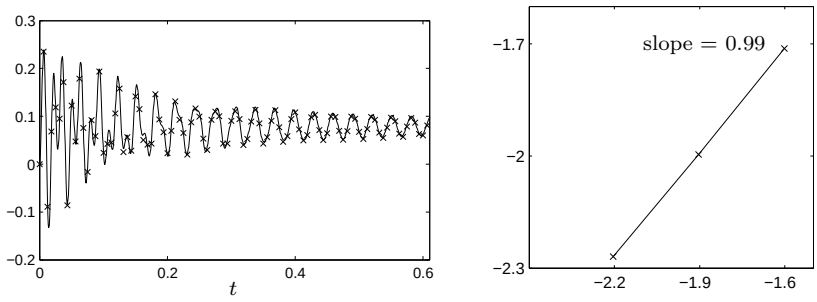


Figure 2.7: **Left:** Error in pressure (central point of the unit square), for a *finite-modes Stokes problem*, sampled with $\times$. Oscillations with different frequencies can be generated. The prediction (2.64) (solid lines), follows the numerical results with good agreement. **Right:** Double logarithmic plot of the error in pressure versus grid size, using the correction (2.80). The indicated slope refers to an approximated straight line.

Equation (2.85) predicts

$$p_e^{(2)} = \frac{1}{4\pi}\left(\cos\left(2\pi(x+y)\right) - \sin\left(2\pi(x+y)\right)\right)\cos\left(2\sqrt{2}\pi\ ac_s s\right)\exp\left(-8\pi^2\nu t\right) + \\ + \frac{1}{2\pi}\cos\left(6\pi x\right)\cos\left(6\pi c_s s\right)\exp\left(-36\pi^2\nu t\right), \quad (2.86)$$

i.e. an initial layer sum of two fast oscillating function, differently damped in time t. Results of this benchmark are shown in figure 2.7.

2.3 Discrete time scale

The previous two-scales analysis has been limited to the lower order moments of $\hat{f}$. Actually, since the initial layer affects the variable $\hat{f}$, focusing on pressure and velocity does not describe completely the initial layer problem. For example, we did not get information about the layer appearing in the stress tensor (shown in figure 2.2), using the equilibrium initialization.
According to the definition (1.169) for $\hat{\mathbf{S}}$, this layer comes from the *non-equilibrium* part of $\hat{f}_i$, which gives by construction (because of the conservation constraints (1.30) of the equilibrium distribution) zero contribution to pressure and velocity.

2.3.1 Lack of non equilibrium

Remark. If the initialization lacks in the **non equilibrium** of $f_i^{(2)}$, any asymptotic expansion ansatz does not contain predictions with smooth coefficients able to cancel the second order residue.

If fact, let us consider the lattice Boltzmann equation written in the form

$$\hat{f}_i(n+1, \mathbf{j}+\mathbf{c}_i) - \hat{f}_i(n, \mathbf{j}) = -\frac{1}{\tau}\hat{f}_i^{neq}(n, \mathbf{j}), \tag{2.87}$$

and a prediction

$$F_h(n, \mathbf{j}) = f^{(0)} + hf^{(1)}(t_n, \mathbf{x_j}) + h^2 f^{(2)}(y_h(n, \mathbf{j})) + O(h^3), \tag{2.88}$$

with $f^{(0)}$ and $f^{(1)}$ defined as in the regular case (as in the previous analysis). The coordinate y_h can be more general than the diffusive one.
Using a Taylor expansion in equation (2.87), we end up with the condition

$$-\tau \mathbf{c}_i \cdot \nabla f_i^{(1)}(t_n, \mathbf{x_j}) = f_i^{neq,(2)}(y_h(n, \mathbf{j})). \tag{2.89}$$

For the equilibrium initialization $f^{neq} = 0$ and the last relation evaluated at $n = 0$ gives

$$-\tau c_s^{-2} f_i^* \mathbf{c}_i \cdot \nabla \mathbf{u}_0(\mathbf{x_j}) \cdot \mathbf{c}_i = f_i^{neq,(2)}(F_h)(0, \mathbf{j}) = 0. \tag{2.90}$$

Equation (2.90) does not have solutions, if $\nabla \mathbf{u}_0 \neq 0$. Hence, we have to use a basically different ansatz, in order to cancel the LBM-residue of second order for an equilibrium initialization. This is an example where an order of the projected residue can be canceled (with the two-scale ansatz), but the single components can not.

Periodicity in space: Initial layer in stress tensor

We derive a prediction for the initial layer in the stress tensor, assuming to be in a periodic domain. For simplicity, we consider the initialization

$$\hat{f}_i(0, \mathbf{j}) = H_i^{eq}(1 + h^2 p_0(\mathbf{x_j}), h\mathbf{u}_0(\mathbf{x_j})), \tag{2.91}$$

which avoids the initial layer in pressure and velocity analyzed before.

Definition 2.4. *We define the* **grid-scale ansatz** *as*

$$\mathcal{A}^3 = \left\{ F \in X \mid F_h(n, \mathbf{j}) = \sum_{k \leq 2} h^k f_i^{LBM,(k)}(t_n, \mathbf{x_j}) + h^2 \sigma_i(n, \mathbf{x_j}) \right\} \tag{2.92}$$

where $f^{LBM,(k)}$ are the interior coefficient for the regular case (1.163) *and σ_i are functions smooth in space.*

The discrete time scale can be introduced in more general ways. However, definition 2.4 allows a clearer application of the analysis to this case.
In a general prediction

$$F_{ih}(n, \mathbf{j}) = \sum_{k \leq 2} h^k f_i^{LBM,(k)}(t_n, \mathbf{x_j}) + h^2 \sigma_i(n, \mathbf{x_j}), \tag{2.93}$$

we have added to the regular expansion the degree of freedom $\sigma_i(n, \mathbf{x_j})$, which depends on the *discrete time scale*. We compute the residue of the LBM, looking for the condition on σ_i which allows to cancel the leading order.
Inserting the ansatz (2.93) into the LBE (2.87), we have

$$f_i^{(2)}(t_{n+1}, n+1, \mathbf{x_j} + h\mathbf{c}_i) - f_i^{(2)}(t_n, n, \mathbf{x_j}) = -\frac{1}{\tau} f_i^{neq,(2)}(t_n, n, \mathbf{x_j}). \tag{2.94}$$

Taylor expanding in t and $\mathbf{x}$,

$$\begin{aligned} \sigma_i(n+1, \mathbf{x_j}) + \mathbf{c}_i \cdot \nabla f_i^{LBM,(1)}(t_n, \mathbf{x_j}) - \sigma_i(n, \mathbf{x_j})) = \\ -\frac{1}{\tau}\left(f_i^{eq,(2)}(t_n, \mathbf{x_j}) - f_i^{LBM,(2)}(t_n, \mathbf{x_j}) - \sigma_i(n, \mathbf{x_j})\right). \end{aligned} \tag{2.95}$$

Note the difference between equation (2.95) and the previous (2.90). The discrete component balances the initial lack in the non equilibrium. Since the function σ has been assumed *smooth in space*, it has been expanded as well. Some considerations which excludes this hypothesis will be done in the next section.
Being

$$f_i^{eq,(2)} - f_i^{LBM,(2)} = -\tau \mathbf{c}_i \cdot \nabla f_i^{LBM,(1)}$$

we have the *difference equation*

$$\sigma_i(n+1, \mathbf{x_j}) = \left(1 - \frac{1}{\tau}\right) \sigma_i(n, \mathbf{x_j}). \tag{2.96}$$

From the residue of the initial conditions

$$r_{IC}^{(2)}(F)(\mathbf{x}) = \sum_{k \le 2} h^k f_i^{LBM,(k)}(0, \mathbf{x}) + h^2 \sigma_i(0, \mathbf{x}) - H_i^{(eq)}(1 + h^2 p_0(\mathbf{x}), h\mathbf{u}_0(\mathbf{x})),$$

we obtain the initial condition

$$\sigma_i(0, \mathbf{x}) = \tau c_s^{-2} f_i^* \mathbf{c}_i \cdot \nabla \mathbf{u}_0(\mathbf{x}) \cdot \mathbf{c}_i, \tag{2.97}$$

which yields

$$\sigma_i(n, \mathbf{x}) = \left(1 - \frac{1}{\tau}\right)^n \tau c_s^{-2} f_i^* \mathbf{c}_i \cdot \nabla \mathbf{u}_0(\mathbf{x}) \cdot \mathbf{c}_i. \tag{2.98}$$

Regarding the stress tensor, taking the second order $\mathbf{c}_i$-moment of σ

$$\mathbf{\Sigma} = \mathbf{c}_i \otimes \mathbf{c}_i \, \sigma, \tag{2.99}$$

we obtain a prediction of the layer in the non-equilibrium part:

$$\mathbf{\Sigma}(n, \mathbf{x}) = \lambda(\tau)^n \tau c_s^2 \, \mathbf{S}\left[\mathbf{u}_0(\mathbf{x})\right], \quad \lambda(\tau) = \left(1 - \frac{1}{\tau}\right). \tag{2.100}$$

The layer is decreasing ($\|\boldsymbol{\Sigma}(n)\| \to 0$) when $|\lambda| < 1$, i.e. for $\tau > \dfrac{1}{2}$, it oscillates when $\lambda < 0$ ($\tau < 1$) and it is over damped when $\lambda > 0$ ($\tau > 1$). Note that if $\nabla \mathbf{u}_0 = 0$, there is no additional layer in the non equilibrium part.

The comparisons between the prediction (2.100) and the initial layer for the problem TV-NS, are shown in figure 2.8. Additionally, in figures 2.9 and 2.10 we test the order of accuracy of the prediction and the effects of different values of τ.

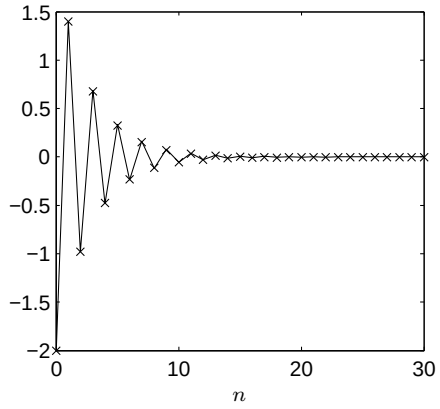


Figure 2.8: Initial layer in $\mathbf{S}[\mathbf{u}]$ ($\times$) in problem TV-NS. Maximum amplitude oscillation of the component S_{xx} is drawn ($\times$), compared with the prediction (2.100) (solid line) as solution of a difference equation. Grid is 80×80. In contrast to figure 2.2, the abscissae indicates the iteration steps n.

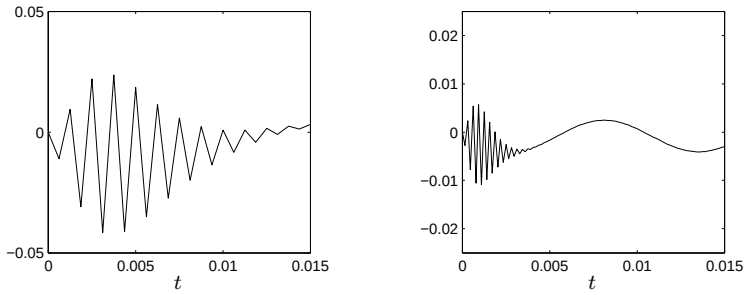


Figure 2.9: The predicted initial layer (2.100) is removed from the result $\hat{\mathbf{S}}$, obtained with initialization (2.91). Grid 40×40 (**left**) and 80×80 (**right**) are compared in time.

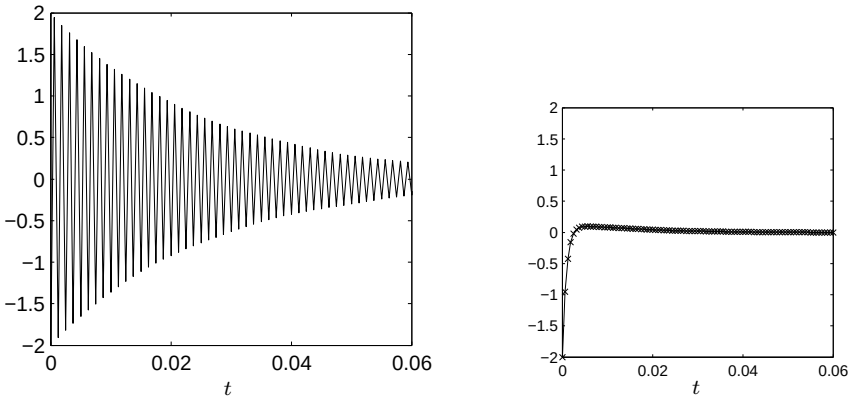


Figure 2.10: Behavior of the initial layer in $\mathbf{S}$ for different τ. On the left, for $\tau = 0.505$, the oscillations persist over a long time scale. On the right, for $\tau = 2$, they decay very fast.

Pointwise arise of error

To derive the prediction (2.100), we have used the hypothesis of regularity in $\mathbf{x}$. The result

$$\sigma_i(n, \mathbf{x_j}) = \lambda(\tau)^n \tau c_s^{-2} f_i^* \mathbf{c}_i \cdot \nabla \mathbf{u}_0(\mathbf{x_j}) \cdot \mathbf{c}_i$$

is a smooth function which oscillates separately in each node $\mathbf{j}$ of the grid. Because of the symmetry properties of the discrete velocity model, since

$$\sum_i \sigma_i(n, \mathbf{x_j}) = 0, \quad \sum_i \mathbf{c}_i \sigma_i(n, \mathbf{x_j}) = \mathbf{0}, \tag{2.101}$$

such initial layer evolves in a space orthogonal to the vectors

$$(1, \ldots, 1),\ (\mathbf{c}_{ix})_{1 \le i \le b},\ (\mathbf{c}_{iy})_{1 \le i \le b} \in \mathbb{R}^b.$$

and will therefore be invisible in pressure and in velocity. In fact, we have been able to treat the irregular behaviors in pressure and velocity excluding the oscillations in the non-equilibrium part.
This property only holds if the layer arises uniformly over all the domain Ω. If the error appears on a limited subset of Ω, for example at a single node $\mathbf{x}_P$ of the grid, the oscillations in the non equilibrium part spoil also in the moments ρ and $\mathbf{u}$. In fact as the populations propagate to the next nodes, the neighbors will be unsymmetrically affected. Moreover, if the spatial regularity is no longer true, the previous procedure cannot be applied.
This is a typical trouble, which arises in moving boundary problems (presented in detail in chapter 4). Here, we perform a qualitative analysis. Taking as basic behavior what we found for the periodic case, we investigate, using some simple benchmarks, the dynamics of the error arising in a single point.

The test problems we consider are based on the TV-NS model, used also for the previous numerical tests. Let us fix the domain

$$\Omega = [0,1)^2$$

and define two sets

$$X_E = \left\{ \mathbf{k}_r^E \mid r = 1, \dots R \right\} \subset \mathcal{G}(h), \quad T_E = \left\{ n_r^E \mid r = 1, \dots R \right\} \subset \mathbb{N}$$

of nodes and times. At the time step n_r^E we add artificially an error, performing a *re-equilibration* at the node $\mathbf{k}_r$, replacing

$$\hat{f}_i(n_r^E, \mathbf{k}_r^E) \to H_i^{eq}(\hat{\rho}(n_r^E, \mathbf{k}_r^E), \hat{\mathbf{u}}(n_r^E, \mathbf{k}_r^E)). \tag{2.102}$$

Two different cases will be investigated.

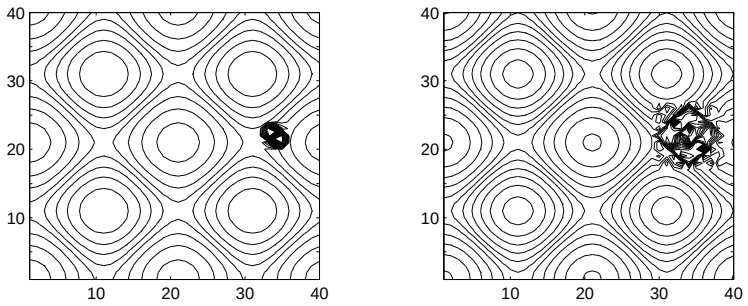


Figure 2.11: Examples of a field of pressure (Taylor vortex) where an *artificial* error has been inserted in a single node, through a re-equilibration. Pressure and velocity on the node are not changed. The picture shows the contour plots of pressure one step after the re-equilibration (**left**) and some time steps later (**right**). Using a low viscosity ($\nu = 0.03$) the error in pressure propagates to a rather wide neighborhood.

Simple pointwise error (SIMPLE) The sets X_E, T_E are generated without particular relation between two successive replacements. As expected, in figure 2.12 we observe that the error in the second order coefficient affects the pressure, as a layer in the discrete time scale. However, it is quickly reduced in time after a few oscillations on the discrete time scale. It depends on n, with a structure similar to (2.99). The damping factor depends on the viscosity. Responsible for the claimed loss of accuracy are the highest peaks, generated immediately after a re-equilibration.

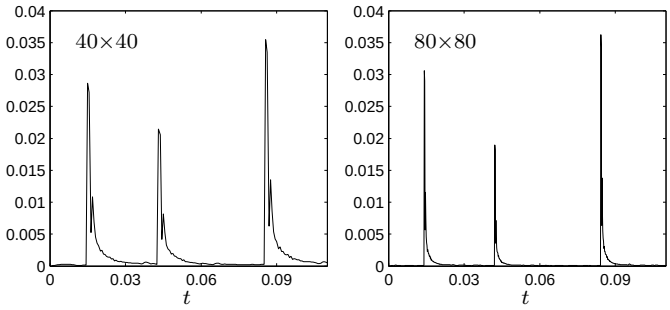


Figure 2.12: Run of the SIMPLE re-equilibration benchmark, error in pressure is shown for grids 40×40 and 80×80. The peaks in the error are of zeroth order, but the error is fast decreasing to zero. The damping factor depends on the viscosity. Here, $\nu = 0.03$.

Error arising along a curve (CURVE) Let $\Gamma \subset \Omega$ be a closed curve, parameterized with a coordinate $\gamma \in [0, 2\pi)$, counterclockwise oriented (see figure 2.13). We identify with Ω_{int} the closed part of the domain, inside the curve Γ, and $\Omega_{ext} = \Omega \backslash \Omega_{int} \cup \Gamma$ (the point on Γ are considered part of the computational domain). The set X_E is defined by

$$\mathbf{k} \in X_E \iff \mathbf{x}_\mathbf{k} \in \Omega_{ext} \wedge \exists i \,:\, \mathbf{x}_\mathbf{k} + h\mathbf{c}_i \in \Omega_{int}, \tag{2.103}$$

i.e. $\mathbf{x}_\mathbf{k}$ is a node in Ω_{ext} which has at least a neighbor in Ω_{int}.

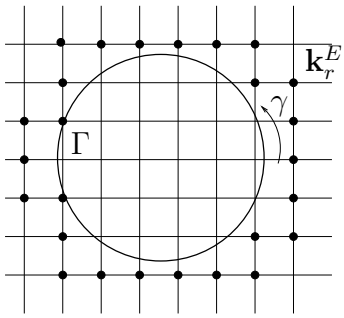


Figure 2.13: Sketch of how the benchmark for the CURVE re-equilibration is constructed. Once defined a closed curve $\Gamma \subset \Omega$, here a circle, the nodes having a neighbor inside the curve (•) are numbered, according to a coordinate γ on Γ. Re-equilibrations are performed following this ordering.

Numbering $X_E = \left\{\mathbf{k}_r^E, \quad r = 1, \ldots, R(\Gamma)\right\}$, we define

$$T_E = \left\{ n_r := \frac{1}{h^2}\frac{1}{R(\Gamma)} r, \quad r = 1, \ldots, R(\Gamma) \right\},$$

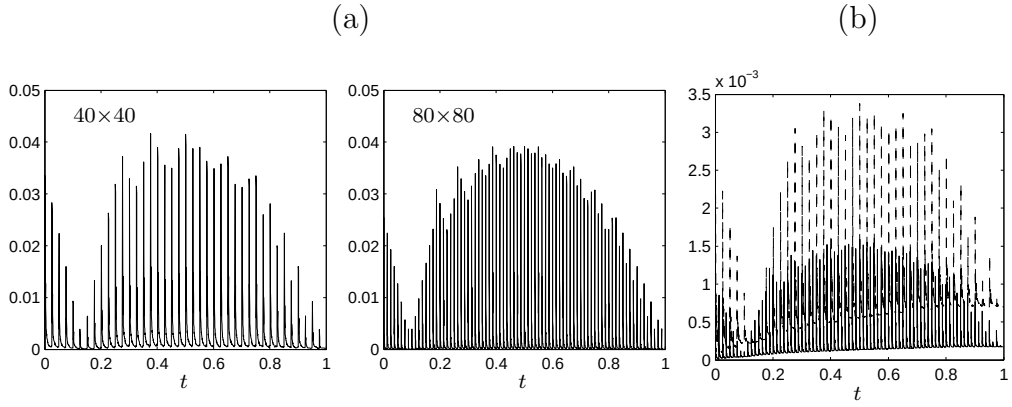


Figure 2.14: **(a)** Maximum error in pressure re-equilibrating the problem CURVE. Two different grid sizes are compared. Now, the introduced perturbations interact during the propagation. Besides the peaks of order zero, the whole profile of the error increases in time, in contrast to the benchmark SIMPLE. **(b)** Superimposed maximal errors in velocity, for $h = 0.025$ (dashed) and $h = 0.0125$ (solid). The Result is only first order accurate.

so that all the nodes are processed in a *physical time* $\overline{T} = 1$.

Note that, now, the curve Γ and time $\overline{T}$ are fixed, *not grid-depending*. Hence, decreasing the grid size (i.e. increasing the cardinality of X_E), more re-equilibrations take place.

The results show a strong decrease of the accuracy in pressure (figure 2.14). The error does consists of more and more dense peaks, which do not reduce, decreasing the grid size h. Analogously, we observe a first order accurate velocity.

2.3.2 TRT models

In this section, we investigate shortly a slightly different lattice Boltzmann model. Instead of the BGK approximation in equation (1.11), we consider the so called TRT model (*Two Relaxation Times*).

In general, for a function $F \in \mathcal{F}(\mathbb{V}, \mathbb{R}^b)$ on the velocity space $\mathbb{V}$, we can define the **even** part:

$$F_i^+ = \frac{F(\mathbf{c}_i) + F(\mathbf{c}_{i^*})}{2}, \quad \text{with } \mathbf{c}_{i^*} = -\mathbf{c}_i, \tag{2.104}$$

and the **odd** part:

$$F_i^- = \frac{F(\mathbf{c}_i) - F(\mathbf{c}_{i^*})}{2}. \tag{2.105}$$

The TRT-LBM is defined as

$$\hat{f}_i(n+1,\mathbf{j}+\mathbf{c}_i) = \hat{f}_i(n,\mathbf{j}) + \frac{1}{\tau^+}\left(f_i^{eq}(\hat{f}) - \hat{f}_i\right)^+(n,\mathbf{j}) + \frac{1}{\tau^-}\left(f_i^{eq}(\hat{f}) - \hat{f}_i\right)^-(n,\mathbf{j}). \quad (2.106)$$

A general asymptotic analysis of the LBM, which includes a wide class of collision operators has been performed in [19]. Here, our purpose is to analyze the effect of a second relaxation time on a multiscale ansatz. In particular, on the *two-scale* and on the *discrete-time* ansatz.
The consequence of distinguishing between odd and even quantities, is that in each coefficient $f_i^{(k)}$ the odd and the even part *of the non equilibrium* will behave differently. Observe that for $f^{(0)}$, $f^{(1)}$, which are already at the equilibrium state, nothing will change. For the second order $f^{(2)}$ (odd function), the non equilibrium part relaxes with τ^+. For a general discrete multi-scale ansatz

$$F_h(n,\mathbf{j}) = \sum_{k<2} f^{LBM,(k)}(h^2 n, h\mathbf{j}) + \sum_{k\in K, k\geq 2} h^k \left(f^{(k)}(hn, h^2 n, h\mathbf{j}) + \sigma^{(k)}(n, h\mathbf{j})\right),$$

we can derive partial differential equations for the oscillating terms in the second and third order, and difference equation for the discrete functions $\sigma^{(k)}$.
Without entering in the details of the computation, we point out the effect of the double relaxation time in a special case. Let us assume we use a second order initialization, i.e. that the nodes are initialized according to (1.163), up to the second order. The ansatz we consider reads

$$F_{ih}(n,\mathbf{j}) = f_i^{LBM,(0)} + h f_i^{LBM,(1)}(t_n,\mathbf{x_j}) + h^2 f_i^{LBM,(2)}(t_n,\mathbf{x_j}) + h^3\left(f_i^{(3)}(t_n,s_n,n,\mathbf{x_j}) + \sigma_i^{(3)}(n,\mathbf{x_j})\right) \quad (2.107)$$

Performing the Taylor expansion as in section 2.3.1, we find eventually the prediction

$$\sigma_i^{(3)}(n+1,\mathbf{x_j}) = \left(1 - \frac{1}{\tau^-}\right)\sigma_i^{(3)}(n,\mathbf{x_j}). \quad (2.108)$$

The odd relaxation time differently relaxes some components of the discrete-time oscillation in higher orders.

Summary

In this chapter the results regarding the approximation of Navier-Stokes through the LBM have been extended, including the analysis of *irregular behaviors*, to predict initial layers and grid oscillations.
The analysis helped to understand the main properties of the oscillating errors appearing in the numerical simulations.

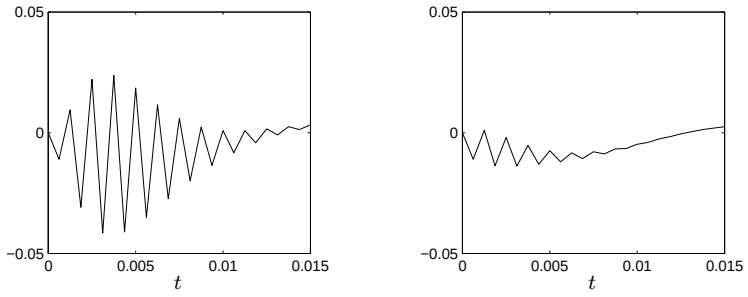


Figure 2.15: Effect of a TRT model. Monitoring the quantity $\hat{S}_{xx} - \mathbf{\Sigma}_{xx}$ in the point $\mathbf{x}_M$ (simulation of problem TV-NS), we observe the error in the order h^3. Simulations are performed on a 40×40 grid. On the left, we use a single $\tau = 0.56$ (same as in figure 2.9). On the right, choosing $\tau_+ = 0.56$, $\tau_- = 1$, the additional initial oscillations have been reduced.

In particular, we obtained a relation between the dynamic of the oscillations (frequency and amplitude) and spatial frequencies of the initial pressure. Numerical tests validate the technique and the derived predictions.
Additionally, we have performed some introductory numerical experiments concerning the initialization of LB variables on a single point. This topic will be of interests dealing with the moving boundary problems in chapter 4.

Next scopes

Our goal is to investigate possible extension of the lattice Boltzmann method towards the fluid-structure interaction problems.
In the following chapters the asymptotic analysis will be used as a tool to *define* new algorithms. The prediction for the interior LBM will be a criteria to *decide* whether a variant of the original algorithm could or could not be satisfactory for our purpose.
The treatment proposed so far, will result useful to have an idea of the general behavior of the errors (fast-time oscillations, grid-scale, symmetry, etc.) within an LB implementation.

Chapter 3

Analysis of initialization algorithms

We have presented a deep asymptotic analysis of the LBM in presence of wrong initial conditions, covering the origin and the prediction of oscillatory layers in time. Now, we approach the problem from a different angle. With the acquired knowledge about inaccurate initializations, our scope is to avoid initial layers by employing some *pre-processor* to the LBM, able to generate better initial values in an efficient way. An initialization algorithm for the LBM (using the multi-relaxation time model) has been first published in [34]. It produces a second order accurate initialization in periodic domains.

We found that a detailed asymptotic analysis of the method helps in understanding its properties. In this chapter, we start from the algorithm proposed in [34] (employed here for the BGK model) for the periodic case, showing how the analysis can be used to improve the accuracy and the efficiency in general. Additionally, we explain how an extension to boundary value problem can be constructed. Finally, we discuss and investigate further variants of initialization routines, and the reliability of high order initializations.

In section 3.1 we present once more the problem, giving some example and defining the benchmarks. A solution for the periodic case [34] is described and analyzed in section 3.2. In section 3.3, we extend the algorithm to boundary value problems. Section 3.4 contains further developments, remarks and some corollary results. The results concerning the analysis of the periodic case have been published by the author in [5].

3.1 The initial layer problem

As usual, the starting point is an incompressible Navier-Stokes initial boundary value problem

$$\begin{aligned} &\begin{cases} \nabla \cdot \mathbf{u} = 0 \\ \partial_t \mathbf{u} + \nabla p + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = \nu \nabla^2 \mathbf{u} + \mathbf{G} \end{cases} \qquad t > 0,\ \mathbf{x} \in \Omega \\ &\mathbf{u}(t, \mathbf{x}) = \mathbf{u}_B(t, \mathbf{x}), \quad t > 0,\ \mathbf{x} \in \partial\Omega \\ &\mathbf{u}(t, \mathbf{x}) = \mathbf{u}_0(\mathbf{x}), \quad t = 0,\ \mathbf{x} \in \Omega \end{aligned} \tag{3.1}$$

on a given domain $\Omega \subset \mathbb{R}^2$.

3.1.1 Test problems

The numerical simulations are based on simple solutions of (3.1).
Namely, we simulate a particular *Taylor vortex field* (section 1.3.4), on the square $\Omega = [0, 1)^2$, with

$$\mathbf{u}_{TV}(t, x, y) = \begin{pmatrix} -\dfrac{1}{2\pi} \cos(2\pi x) \sin(2\pi y) \\ \dfrac{1}{2\pi} \sin(2\pi x) \cos(2\pi y) \end{pmatrix} \exp(-8\pi^2 \nu t), \tag{3.2}$$

$$p_{TV}(t, x, y) = -\frac{1}{16\pi^2} (\cos(4\pi x) \cos(4\pi y)) \exp(-16\pi^2 \nu t). \tag{3.3}$$

Both the problems Navier-Stokes (TV-NS) or Stokes (TV-ST) (using a linearized equilibrium and adding a volume force $\mathbf{G} = \nabla p_{TV}$), are considered.
Such problems are naturally implemented with **periodic boundary**. However, also a bounded domain can be considered, assigning the exact velocity

$$\mathbf{u}_B(t, \mathbf{x}) = \mathbf{u}_{TV}(t, \mathbf{x}),\ \mathbf{x} \in \partial\Omega$$

on the boundary of the square as **boundary condition**.
Besides, we consider another *linear* problem with *Dirichlet boundary conditions* (denoted with DBC):

$$\begin{aligned} &\begin{cases} \nabla \cdot \mathbf{u} = 0 \\ \partial_t \mathbf{u} + \nabla p = \nu \nabla^2 \mathbf{u} + \mathbf{G}^{\mathrm{DBC}} \end{cases} \qquad t > 0,\ \mathbf{x} \in \Omega = [0, 1] \times \mathbb{R} \\ &\mathbf{u}(0, \mathbf{x}) = \mathbf{u}_0^{\mathrm{DBC}}(\mathbf{x}), \quad \mathbf{x} \in \Omega \\ &\mathbf{u}(t, \mathbf{x}) = \mathbf{u}_B(t, \mathbf{x}), \quad t > 0,\ \mathbf{x} \in \partial\Omega \end{aligned} \tag{3.4}$$

with

$$\mathbf{G}^{\mathrm{DBC}}(t, \mathbf{x}) = \begin{pmatrix} \sin 2\pi x \\ -\sin 2\pi y \end{pmatrix} - \begin{pmatrix} \mathrm{e}^{-t} \\ 0 \end{pmatrix} \tag{3.5}$$

and velocity on the boundary

$$\mathbf{u}_B(t,0,y) = \mathbf{u}_B(t,1,y) = \begin{pmatrix} \mathrm{e}^{-t} \\ 0 \end{pmatrix}, \; y \in \mathbb{R}. \tag{3.6}$$

Exact solutions are

$$u^{\mathrm{DBC}}(t,\mathbf{x}) = \mathrm{e}^{-t}, \quad v^{\mathrm{DBC}}(t,\mathbf{x}) = 0, \tag{3.7}$$

$$p^{\mathrm{DBC}}(t,\mathbf{x}) = \frac{1}{2\pi}\left(\cos 2\pi y - \cos 2\pi x\right). \tag{3.8}$$

3.1.2 Lattice Boltzmann initial conditions

As an introduction, we present a short overview of possible choices for the initial values, describing their main features. The simplest way consists of initializing with *equilibrium values*

$$\hat{f}_i(0,\mathbf{j}) = H_i^{eq}(1, h\mathbf{u}_0(\mathbf{x_j})). \tag{3.9}$$

As already remarked before, it is not completely in accordance with the problem. In view of (1.168), a constant initial density $\rho_0 = 1$, is equivalent to an initial pressure $p_0 = 0$. However, for general $\mathbf{u}_0$ and $\mathbf{G}$, the physical initial pressure obeys

$$\nabla^2 p_0 = -\nabla\cdot(\nabla\cdot(\mathbf{u}_0\otimes\mathbf{u}_0)) + \nabla\cdot\mathbf{G}|_{t=0}. \tag{3.10}$$

which follows by taking the divergence of the Navier-Stokes equation at time $t = 0$.

Figure 3.1 shows the results of the LBM for the problem TV-NS, using the initial values (3.9). As observed and analyzed in chapter 2, the initial discrepancy in pressure produces an initial layer, which does not vanish for $h \to 0$.

To include the initial pressure, we could use an *additional Poisson solver*, estimating $\tilde{p}$ from equation (3.10) and defining

$$\hat{f}_i(0,\mathbf{j}) = H_i^{eq}(1 + h^2 c_s^{-2}\tilde{p}(\mathbf{j}), h\mathbf{u}_0(\mathbf{x_j})). \tag{3.11}$$

With initialization (3.11), the initial layer still appears in the tensor $\mathbf{S}[\mathbf{u}]$. An initialization which includes also the initial *non equilibrium* part, and leads to better results, was first proposed in [40]:

$$\hat{f}_i(0,\mathbf{j}) = H_i^{eq}(1 + h^2 c_s^{-2}\tilde{p}(\mathbf{j}), h\mathbf{u}_0(\mathbf{x_j})) - h^2\tau c_s^{-2} f_i^*\left(\mathbf{c}_i\otimes\mathbf{c}_i : \tilde{\mathbf{S}}(\mathbf{j})\right). \tag{3.12}$$

using a *numerical approximation* $\tilde{\mathbf{S}}$, of the tensor $\mathbf{S}[\mathbf{u}_0]$.

At this point, it seems that to construct better initializations, more and more expensive routines are required. However, the approach presented in [34], which we analyze and generalize in this chapter, allows to achieve initialization (3.12) completely within the LB framework.

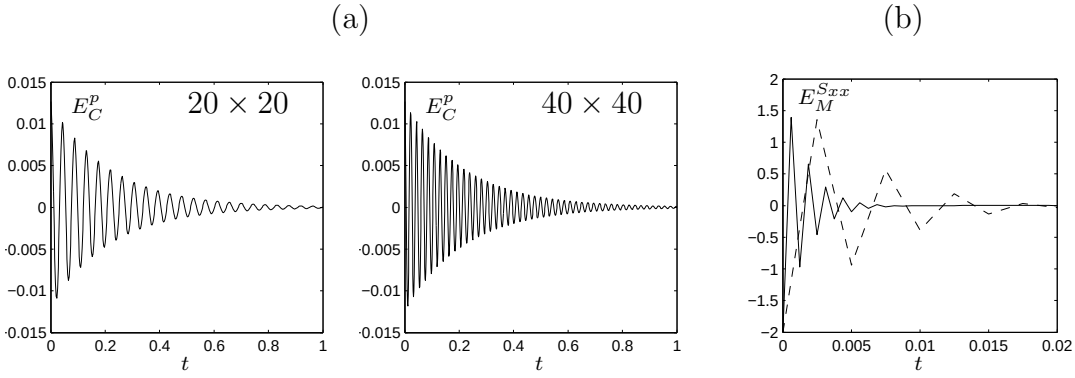


Figure 3.1: Qualitative behavior of initial layers. **(a)** Error $E_C^p(t)$ in pressure for the problem TV-NS, in the central point $\mathbf{x}_C$ of the unit square. Initialization (3.9), comparison between different grids. **(b)** Superimposed oscillatory initial layers in $\mathbf{S}[\mathbf{u}]$, with initial values (3.11). The quantity $E_M^{S_{xx}} = \hat{S}_{xx} - \mathbf{S}_{xx}$ is shown, in the point of maximum amplitude oscillation, for $h = 0.05$ (dashed line) and $h = 0.025$ (solid).

3.2 LB Initialization routines (periodic case)

3.2.1 A first initialization algorithm

We focus on a periodic domain, without boundaries. The following algorithm has been proposed in [34][a] to initialize LBM according to (3.12).

It has the same structure as the classical LBM. Only, in the collision step, the velocity is kept fixed and equal to $\mathbf{u}_0$ in the equilibrium function. We assume to initialize the procedure with equilibrium initial conditions (3.9).

Algorithm 3.1 ((periodic) LB-initialization).
`initialize` $\hat{\rho}(0,\mathbf{j}) = 1, \quad \hat{f}(0,\mathbf{j}) = H_i^{eq}(1, h\mathbf{u}_0(\mathbf{x_j}))$
`DO WHILE` $||\hat{\rho}(n+1,\cdot) - \hat{\rho}(n,\cdot)|| > \epsilon$ (fixed by tolerance criterion)
  $\mathbf{u}_0$`-collision:` $\hat{f}_i^c(n,\mathbf{j}) = \hat{f}_i(n,\mathbf{j}) + \frac{1}{\tau}(H_i^{eq}(\hat{\rho}(n,\mathbf{j}), h\mathbf{u}_0(\mathbf{x_j})) - \hat{f}_i(n,\mathbf{j})) + \hat{g}_i(0,\mathbf{j})$
  `advection:` $\hat{f}_i(n+1,\mathbf{j}+\mathbf{c}_i) = \hat{f}_i^c(n,\mathbf{j})$
  $\hat{\rho}(n+1,\cdot) = \sum_i \hat{f}_i(n+1,\cdot)$
`END`

[a] Using the MRT model.

Asymptotic analysis

Defining

$$\begin{aligned}\mathrm{LBM}_i^0(h,\hat{f}_h)(n+1,\mathbf{j}+\mathbf{c}_i) &= \hat{f}_i(n+1,\mathbf{j}+\mathbf{c}_i) - \hat{f}_i(n,\mathbf{j}) \\ &- \left(\frac{1}{\tau}\left(H_i^{eq}(\rho(n,\mathbf{j}),h\mathbf{u}_0(\mathbf{x}_\mathbf{j})) - \hat{f}_i(n,\mathbf{j})\right) + \hat{g}_i(0,\mathbf{j})\right) \quad (3.13)\end{aligned}$$

(LBM with fixed collision velocity), we analyze the algorithm

$$L(h,\hat{f}_h)(n,\mathbf{j}) = \begin{cases}\hat{f}(0,\mathbf{j}) - H^{eq}(1,h\mathbf{u}_0(\mathbf{x}_\mathbf{j})) & n=0,\ \mathbf{j}\in\mathcal{G}(h)\\ \mathrm{LBM}^0(h,\hat{f}_h)(n,\mathbf{j}) & n>0,\ \mathbf{j}\in\mathcal{G}(h)\end{cases} \quad (3.14)$$

using a *regular ansatz*, for which the predictions have the form (1.101):

$$F_{ih}(n,\mathbf{j}) = \sum_{k\in K} h^k f_i^{(k)}(h^2 n, h\mathbf{j}).$$

Since in what follows the time t is not the "real" time, i.e. the same appearing in the Navier-Stokes equation (3.1) (the algorithm is used only to initialize the populations, keeping the initial velocity and the initial force fixed), we will call it *pseudotime*, even if we indicate it with the letter t. The frozen quantities are denoted with subscript 0, like $\mathbf{u}_0$ or $\mathbf{G}_0$.
As in the previous chapter, particular attention will be given to the moments. In particular, since the velocity is kept fixed, we focus on the pressure, computed through (see equation (1.168))

$$\hat{p}_h(n,\mathbf{j}) = c_s^{-2}\frac{\hat{\rho}_h(n,\mathbf{j})-1}{h^2}. \quad (3.15)$$

Therefore we perform the analysis introducing a variant of the *projected algorithm* MOM_ρ (defined in equation (2.12), section 2.2.1) for the density, which gives as output the numerical pressure:

$$\mathrm{MOM}_p(h,\hat{p}_h,\hat{f}_h) = \hat{p}_h - c_s^{-2}\frac{\hat{\rho}_h - 1}{h^2}, \quad (3.16)$$

where $\hat{\rho}_h = \rho(\hat{f}_h)$.
To find a prediction for the results initialization algorithm, we apply the heuristic optimization 1.2.4 starting from a regular ansatz $\mathcal{A}$. The argument which leads to the derivation of the coefficients is totally analogous to the one used in section 1.3.2 for the standard LBM. Without repeating the computations, eventually we find that the minimum class is contained in the subset (compare with (1.139))

$$
\begin{aligned}
&F \in \mathcal{A} \mid \\
&f_i^{(0)}(F) = f_i^*, \\
&f_i^{(1)}(F)(\mathbf{x}) = f_i^* c_s^{-2} \mathbf{c}_i \cdot \mathbf{u}_0(\mathbf{x}), \\
&f_i^{(2)}(F)(t,\mathbf{x}) = f_i^* c_s^{-2} p(t,\mathbf{x}) + H_i^{Q(eq)}(\mathbf{u}_0,\mathbf{u}_0)(\mathbf{x}) - \tau f_i^* c_s^{-2} (\mathbf{c}_i \cdot \nabla)\mathbf{c}_i \cdot \mathbf{u}_0(\mathbf{x}), \\
&f_i^{(3)}(F)(t,\mathbf{x}) = f_i^* c_s^{-2} p^{(3)}(t,\mathbf{x}) \\
&\qquad\qquad - \tau \left((\mathbf{c}_i \cdot \nabla) f_i^{(2)}(F)(t,\mathbf{x}) + \frac{(\mathbf{c}_i \cdot \nabla)^2}{2} f_i^{(1)}(F)(\mathbf{x}) - \tau g_{0,i}^{(3)}(\mathbf{x}) \right),
\end{aligned}
\tag{3.17}
$$

with smooth fields p and $p^{(3)}$. Note that $f^{(0)}$ and $f^{(1)}$ are constant functions. The predictions defined in (3.17) cancel the third order residue of LBM^0 if the field p satisfies

$$
\begin{cases}
\nabla p + \nabla \cdot (\mathbf{u}_0 \otimes \mathbf{u}_0) = \nu \nabla^2 \mathbf{u}_0 + \mathbf{G}_0 - \dfrac{1}{\tau}\mathbf{w} \\
c_s^{-2} \partial_t p + \nabla \cdot \mathbf{w} + \dfrac{1}{2}(\nabla^2 p + \nabla \cdot (\mathbf{u}_0 \otimes \mathbf{u}_0)) = 0.
\end{cases}
\tag{3.18}
$$

In (3.18) the field $\mathbf{w}$ is the first order moment of the coefficient $f_i^{(3)}$,

$$
\mathbf{w} = \sum_{i=1}^{b} \mathbf{c}_i f_i^{(3)}, \tag{3.19}
$$

which can be recovered from the first equation and inserted into the second, obtaining

$$
\partial_t p = \nu(\nabla^2 p + \nabla \cdot (\nabla \cdot (\mathbf{u}_0 \otimes \mathbf{u}_0)) - \nabla \cdot \mathbf{G}_0) + \frac{c_s^2}{2} \nabla \cdot \mathbf{G}_0. \tag{3.20}
$$

To fix the initial condition for p we consider the residue of the initialization rule (3.9). As remarked before (section 3.1.2), it leads to

$$
p(0,\mathbf{x}) = 0, \ \mathbf{x} \in \Omega\,. \tag{3.21}
$$

From the asymptotic expansion of F, we can derive an expansion for the pressure:

$$
\hat{p}(F_h) = p + h p^{(3)} + h^2 p^{(4)} + h^3 p^{(5)} + \dots. \tag{3.22}
$$

Using the definition (3.15) of $\hat{p}$, the coefficients (for $k \geq 2$) reads

$$
p^{(k)}(F) = c_s^{-2} \rho^{(k)}(F). \tag{3.23}
$$

Note that the expansion (3.22) starts from the second order, since

$$
\rho^{(0)}(F) = 1, \quad \rho^{(1)}(F) = 0
$$

if F is defined as in (3.17). Since equation (3.15) is scaled by h^2, the indexes of the expansion coefficients in (3.22). have been shifted. For simplicity, we have called p the second order coefficient.
Equation (3.22) expresses the relation between a prediction F for the LB-variables and a prediction for the pressure $\hat{p}_h$. To fix the coefficients $p^{(k)}$ we investigate the following order residues for the projected algorithm (3.16). Introducing the field

$$\mathbf{w}^{(4)}(F) := \sum_i \mathbf{c}_i f_i^{(4)}(F), \tag{3.24}$$

we obtain, for the coefficient $p^{(3)}$, the system

$$\begin{cases} \nabla p^{(3)} = -\dfrac{1}{\tau}\mathbf{w}^{(4)} \\ c_s^{-2}\partial_t p^{(3)} + \nabla \cdot \mathbf{w}^{(4)} + \dfrac{1}{2}\nabla^2 p^{(3)} = 0 \end{cases} \tag{3.25}$$

which can be reduced to

$$\partial_t p^{(3)} = \nu \nabla^2 p^{(3)}. \tag{3.26}$$

Inserting the initial condition

$$p^{(3)}(0, \mathbf{x}) = 0, \tag{3.27}$$

required to cancel the third order projected residue of the initialization in algorithm 3.1, equation (3.26) has solution $p^{(3)} = 0$.

An LB Poisson solver

We use the coefficients (3.17) to set up the truncated expansion

$$\hat{F} = f^{(0)} + hf^{(1)} + h^2 f^{(2)} + h^3 f^{(3)}, \tag{3.28}$$

which is employed to investigate the behavior of the initialization routine.
Now we analyze equation (3.20) in more detail. If $\nabla \cdot \mathbf{G}_0 = 0$, it shows that p solves the Poisson equation (3.10) at the steady state (in pseudotime). Hence, according to (3.22) we can extract from F a second order accurate pressure, since (as a consequence of (3.26))

$$\hat{p}_h - p \in O(h^2)$$

(the higher order coefficients have not been fixed).
On the other hand, the procedure does not work if the force has non zero divergence, because equation (3.20) is then different from (3.10). Figure 3.2 exemplifies the problem. Looking at the pressure during the initialization algorithm for a TV-ST problem (where an additional *non divergence-free* volume force is present) we see an error in the asymptotic value increasing like ν^{-1} (this relationship is also explained by equation (3.20)). Fixing the viscosity and refining the grid

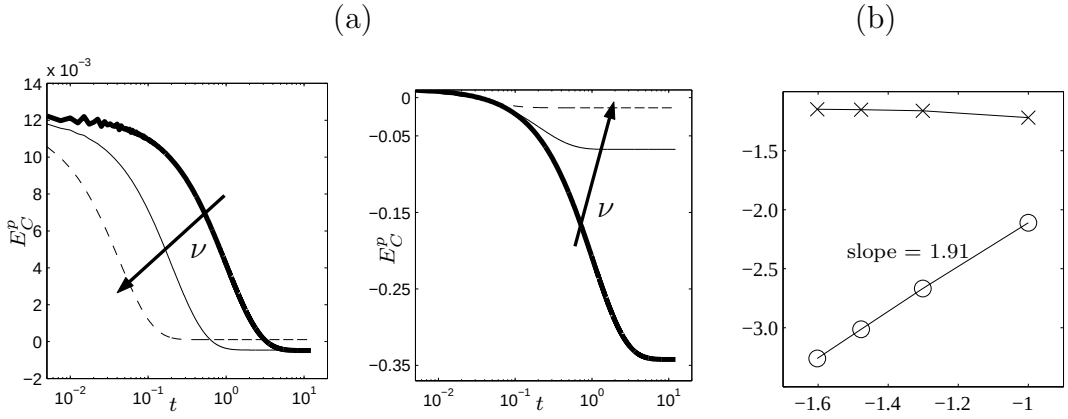


Figure 3.2: **(a)** Algorithm 3.1 applied to vortex solution (20×20 grid), for viscosities $\nu_1 = 0.006$ (bold line), $\nu_2 = 5\nu_1 = 0.030$ (solid), $\nu_3 = 5\nu_2 = 0.15$ (dashed) (the arrow in the plots shows the increasing viscosity). Left: error $E_p^C(t)$ of central point pressure in *logarithmic* pseudotime during initialization for TV-NS problem. Right: TV-ST problem (with additional *non divergence-free* force). The error is increasing like ν^{-1}. **(b)** Double logarithmic plot of maximum error in initial pressure versus grid size in TV-ST, with ($\circ$) and without ($\times$) the corrected force term g^∇.

(figure 3.2b), the error in the initial pressure (solution of (3.10)) does not reduce (slope ~ 0 in the double logarithmic plot).
To cure this anomaly we modify the collision step, replacing $\hat{g}_i$ with

$$g_i^\nabla \equiv h^3 c_s^{-2} f_i^* \mathbf{c}_i \cdot \mathbf{G} + h^4 f_i^* \frac{\nabla \cdot \mathbf{G}}{2}, \tag{3.29}$$

which produces a new term in (3.20) able to remove the undesired source. Results of this modified routine confirm that the corrected force g_i^∇ allows to recover a second order accurate initial pressure (slope ~ 2) solving equation (3.10) only by using LB-type iterations.
Other aspects of the (modified) algorithm 3.1 follow from the analysis performed in section 3.2.1. Actually, it is more than a LB Poisson solver, since it does not even require an approximation of $\nabla \mathbf{u}_0$ to set correctly also the non equilibrium part of $f_i^{(2)}$, according to the interior coefficient of the standard LBM (1.163) at time $t = 0$ (as initialization (3.12) does). It represents therefore a *second order initialization*.
On the other hand, we observe that the number of pseudotime steps needed to reach the steady state of equation (3.20) is a function ($\sim \frac{1}{\nu}$) of viscosity, i.e. of τ (it can be seen in figure 3.2, showing the error in pressure approaching a steady value). Therefore, once guaranteed the accuracy in pressure for a general force field, our next aim is to see whether it is possible to reduce the computational effort to get it. Within the MRT model, a solution has been proposed in [34].

Here, employing the BGK lattice Boltzmann, we show that the efficiency can be improved even in this much more simple model.

3.2.2 Accelerated initialization

The idea is the following: since the Poisson equation (3.10) does not depend on ν, to have a faster procedure, it should be possible to run the algorithm with a higher, *faster*, viscosity (i.e. using a different τ). This allows the pressure to get closer to its limit in less iterations. Unfortunately, the simple increasing of viscosity leads to a wrong initial tensor $\mathbf{S}[\mathbf{u}]$, in which we still see a zeroth order initial layer (figure 3.3, 3.4). This is due to a wrong initialization of the (τ-depending) non-equilibrium term in $f_i^{(2)}$, which is different from the interior coefficient $f_i^{LBM,(2)}$. However, even initializing with a different value of τ, we can

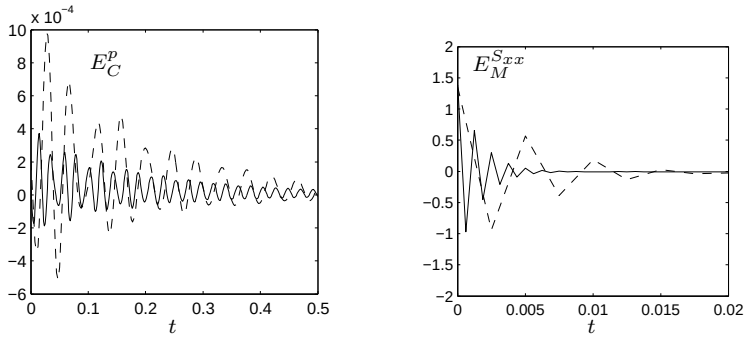


Figure 3.3: Same simulation as in figure 3.1, the error in p and S_{xx} are now shown after the modified viscosity initialization. It reduces the order of initial layer in pressure (**left**) but not in the tensor $\mathbf{S}[\mathbf{u}]$ (**right**). Dashed line: 20×20 grid. Solid line: 40×40 grid.

use the definitions of the coefficients (3.17) to derive a recipe to correct the initial populations, re-obtaining an algorithm whose residue vanishes if $f_i^{(2)}(F)$ coincide with the interior coefficient $f^{LBM,(2)}|_{t=0}$. Practically, calling $\tilde{\tau}$ the new relaxation time and

$$r \equiv \frac{\tau}{\tilde{\tau}},$$

we first *isolate* (up to order h) the term we are interested in,

$$f_i^{(2),noneq,\tilde{\tau}} = \frac{1}{h^2}\left(\hat{f}_i^{\tilde{\tau}} - H_i^{eq}(1+h^2\hat{p}, h\mathbf{u}_0) + O(h^3)\right),$$

subtracting from the output $\hat{f}_i^{\tilde{\tau}}$ of the accelerated routine the previous order terms. Then, we *reconstruct* "by hand" an approximation of the correct initial

second order,

$$\bar{f}_i^{(2)} = (1-r)\left(f_i^* c_s^{-2}\hat{p} + H_i^{Q(eq)}(\mathbf{u}_0, \mathbf{u}_0)\right) + r\left(\frac{f_i^{\tilde{\tau}} - f_i^* - h f_i^* c_s^{-2}\mathbf{c}_i \cdot \mathbf{u}_0}{h^2}\right). \quad (3.30)$$

The corrected distributions are then

$$\bar{f}_i^{\tilde{\tau}} = f_i^* + h f_i^{(1)} + h^2 \bar{f}_i^{(2)} = (1-r)H_i^{eq}(1 + h^2 c_s^{-2}\hat{p}, h\mathbf{u}_0) + r\hat{f}_i^{\tilde{\tau}}. \quad (3.31)$$

Note that equation (3.31) also provides a simple formula to correct the initial values. Performing the iterative algorithm, the correction can be easily implemented as

$$\bar{f}_i^{\tilde{\tau}} = (1-r)H_i^{eq}(\hat{\rho}, h\mathbf{u}_0) + r\hat{f}_i^{\tilde{\tau}}, \quad (3.32)$$

where both the quantities appearing on the right hand side do not need to be evaluated (the equilibrium can be taken from the collision step of the last iteration).

Algorithm 3.2 (Accelerated (periodic) LB-initialization).
(given initial data $\mathbf{u}_0$ and force $\mathbf{G_0}$)
`compute` $\nabla \cdot \mathbf{G}_0$ (at least first order accurately), g^{∇}
`run algorithm 3.1 with` $\tilde{\tau}$
`compute pressure` $\hat{p}$ `and equilibrium` $H_i^{eq}(\hat{\rho}, h\mathbf{u}_0)$
`initialize LBM using` (3.32)

Using equations (3.30)-(3.31), a prediction and the coefficients defined in (3.17) $\bar{F}_h^{\tilde{\tau}}$ for the accelerated initialization can be constructed. We can estimate the difference between $F^{\tilde{\tau}}$ and the initial interior prediction

$$F_h^{LBM,0} = F_h^{LBM}|_{t=0} = \sum_{k\leq 3} h^k f^{LBM,(k)} \quad (3.33)$$

(constructed from (1.163)) of the standard LBM as

$$F_h^{LBM,0} - \bar{F}_h^{\tilde{\tau}} = h^3\left(-r f^{(3),\tilde{\tau}} + f^{LBM,(3)}\right) + O(h^4). \quad (3.34)$$

We conclude that *the accelerated initialization procedure 3.2 second order initialization for the interior LBM.*
The next numerical tests (figure 3.4) compare results of the original initialization routine, with the accelerated one. The original viscosity is $\nu \sim 0.03$ is achieved with $\tau = 0.59$. As faster relaxation time we use $\tilde{\tau} = 1$, which allows also to simplify the implementation of the LB collision step. The tolerance criterion in algorithm 3.1 is based on the difference between the pressure in two successive pseudotime iterations (related to an approximated $\partial_t p$) and the gain in CPU time is about 65%. Initial layers in pressure and viscous stress tensor have been compared. We get first order accuracy for $\hat{p}$ and second order (only after correction (3.31)) for $\hat{\mathbf{S}}$, even if, for pressure, after the accelerated procedure the initial layer amplitude may be slightly bigger[b].

[b]It happens mainly because we modify the $f_i^{(3)}$. Using the evaluation of error (3.34), we can

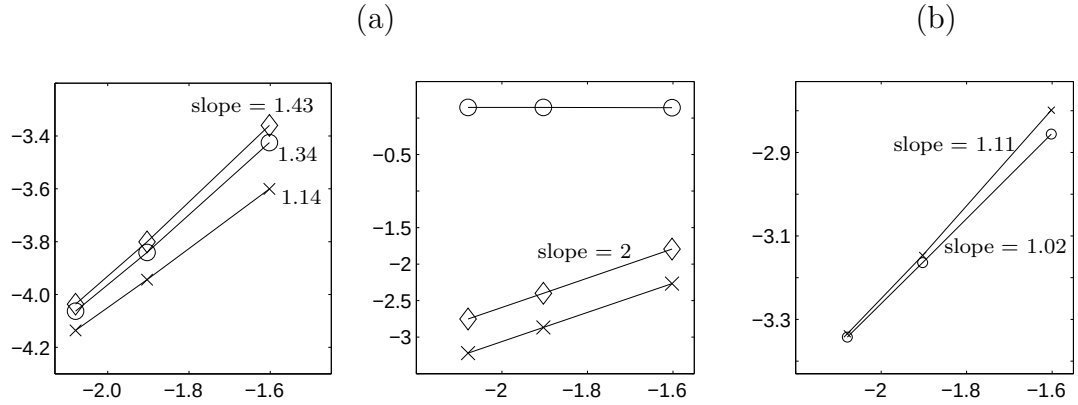


Figure 3.4: **(a)** Double logarithmic plot of maximum error in initial pressure (left) and $\mathbf{S}[\mathbf{u}]$ (right) for TV-NS, versus grid size ($\tau = 0.59$, viscosity $\sim 3 \cdot 10^{-2}$). The curves are obtained with original-viscosity routine ($\times$), algorithm 3.1 run with $\tau = 1$ ($\circ$), and accelerated routine, inclusive of correction ($\diamond$). **(b)** Error in pressure versus grid size for original ($\times$) and accelerated ($\circ$) initialization applied to TV-ST.

Special viscosity

Performing a further step of asymptotic analysis, equating to zero the projected residue of the next orders, we find an equation for the coefficient $p^{(4)}$ of expansion (3.22):

$$\begin{cases} A(\tau)\partial_t p^{(4)} + B(\tau)\partial_t^2 p = \nabla^2 p^{(4)} + \dfrac{1}{\tau}\phi(\tau)\mathcal{F}(\mathbf{u}_0, \mathbf{G}_0) + \gamma(\tau)\nabla^2(\nabla \cdot \mathbf{G}_0) \\ p^{(4)}(0, \mathbf{x}) = 0, \quad \mathbf{x} \in \Omega. \end{cases} \tag{3.36}$$

The initial condition is fixed by the equilibrium initialization. The operator $\mathcal{F}$ involves fourth and sixth order derivatives of the initial data. The function $\phi(\tau)$ is a second order polynomial with two real roots, $\tau_\pm^* = \dfrac{1}{2} \pm \dfrac{1}{\sqrt{6}}$. This means that if $\nabla \cdot \mathbf{G}_0 = 0$, with the special value $\tau_+^* \sim 0.9089$, $p^{(4)}$ vanishes at the steady state, and the expansion predicts a fourth order initial pressure (the coefficient $p^{(5)}$ behaves like the previous odd term $p^{(3)}$, see [19]). Note that the third order

express the modification occurring in the third order coefficients (as given in (3.17)) running algorithm 3.2:

$$E_i^{(3)} := f_i^{(3),\tau} - \bar{f}_i^{(3),\tilde{\tau}} = (\tau - \tilde{\tau})\tau c_s^2 f_i^*(\mathbf{c}_i \cdot \nabla)^2(\mathbf{c}_i \cdot \mathbf{u}_0). \tag{3.35}$$

This part is only responsible of the increasing of the amplitude (figure 3.4a) of the initial layer in pressure. It arises even using the original viscosity routine, because the expression of $f_i^{(3),\tau}$ differs anyway from the initial values of $f^{LBM,(3)}$ given in (1.139) (section 1.3). In particular, a term involving $\partial_t \mathbf{u}|_{t=0}$ is missing, affecting our prediction $\bar{F}_h^{\tilde{\tau}}$ from order h^3. Note that in the linear problem, original and accelerated routines lead to similar results (equation (3.35) only involves quadratic terms), figure 3.4b.

error in the coefficients has not been removed (it is just "invisible" in pressure) and that the fourth order pressure will, in general, become of second order once starting the actual LBE iteration. Figure 3.5 validates the theoretical conclusions.

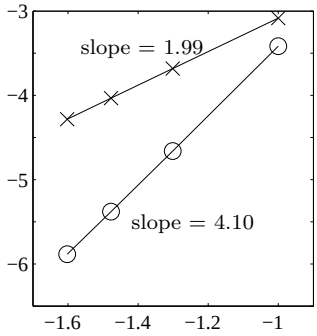


Figure 3.5: Double logarithmic plot of maximum error in the *initial pressure* after initialization routine, for $\tau = \tau_+^*$ ($\circ$) and $\tau = 1$ ($\times$).

3.3 Initialization for boundary value problems

3.3.1 The Neumann condition for pressure

Considering a bounded domain, through a boundary rule we introduce additional boundary conditions for the pressure in equations (3.18). Projecting the Navier-Stokes equations at the initial time along the direction of the outgoing normal $\mathbf{n}$ to the boundary, we have

$$\frac{\partial p_0}{\partial \mathbf{n}} = -\mathbf{n}[(\mathbf{u}\cdot\nabla)\mathbf{u}] + \nu(\nabla^2\mathbf{u})\cdot\mathbf{n} + \mathbf{G}\cdot\mathbf{n} - (\partial_t\mathbf{u})\cdot\mathbf{n}. \tag{3.37}$$

Rewriting the first part of equation (3.18) as

$$\nabla p + (\mathbf{u}_0\cdot\nabla)\mathbf{u}_0 = \nu\nabla^2\mathbf{u}_0 + \mathbf{G} - \frac{1}{\tau}\mathbf{w} \tag{3.38}$$

and imposing $\dfrac{\partial p}{\partial \mathbf{n}} = \dfrac{\partial p_0}{\partial \mathbf{n}}$ we obtain a relation involving the field $\mathbf{w}$ (on the boundary):

$$\mathbf{w}\cdot\mathbf{n} = \tau\partial_t\mathbf{u}|_{t=0}\cdot\mathbf{n}. \tag{3.39}$$

Assuming a Dirichlet condition for the velocity in the Navier-Stokes problem, our aim is to construct a boundary rule for the algorithm 3.1, which achieves the conditions

$$\mathbf{u}^{(1)}(\mathbf{x}) = \mathbf{u}_B(\mathbf{x}), \quad \mathbf{x}\in\partial\Omega. \tag{3.40}$$

$$\mathbf{w}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) = \sum_i f_i^{(3)}(\mathbf{x}) \mathbf{c}_i \cdot \mathbf{n}(\mathbf{x}) = \tau \partial_t \mathbf{u}|_{t=0}(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega. \tag{3.41}$$

There are many ways [4, 20, 24] to include a specific Dirichlet or Neumann condition for *the velocity* in an LB algorithm. None of them, however, has been built to deal with a condition on the moment $\mathbf{w}$ as (3.41).

3.3.2 LB-boundary condition algorithms

Within the LB implementation, we consider the first order BFL-rule[c]. Namely, for $\mathbf{x_k} \in \Omega$, which has a direction i^* crossing the boundary (*outgoing* direction), i.e. $\mathbf{x_k} + h\mathbf{c}_{i^*} \notin \Omega$, the boundary rule defines the population occupying the link i, with $\mathbf{c}_i = -\mathbf{c}_{i^*}$ (*incoming direction*), which is not updated by the advection. The idea of the BFL is to interpolate such distribution by using the value of neighboring nodes after performing the collision step. According to the notations in figure 3.6:

$$\begin{aligned} \hat{f}_i(n+1, \mathbf{k}) = C_1(q) \hat{f}^c_{i^*}(n, \mathbf{k}) + C_2(q) \hat{f}^c_{i^*}(n, \mathbf{k} + \mathbf{c}_i) + \\ C_3(q) \hat{f}^c_i(n, \mathbf{k}) + h2D(q) f^*_i c_s^{-2} \mathbf{u}_B(\mathbf{b}_{i^*}(\mathbf{k})) \cdot \mathbf{c}_i, \end{aligned} \tag{3.42}$$

($\hat{f}^c_i$ indicates the distribution *after collision*).

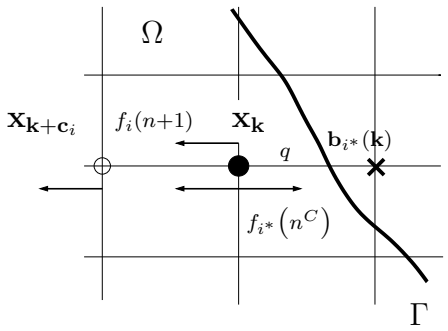


Figure 3.6: Sketch of the BFL boundary conditions. The bold line is the boundary, $\mathbf{c}_i$ is an *incoming* direction at the boundary node $\mathbf{x_k}$, $\mathbf{c}_{i^*}$ is outgoing and connects the fluid node (circles) to the first boundary one (cross). With q we denote the distance (in lattice units) between the boundary node and the point $\mathbf{b}_{i^*}(\mathbf{k}) \in \Gamma$ (intersection along the link $\mathbf{c}_{i^*}$). The new population (top arrow) is computed interpolating between the boundary node and the left one (bottom arrows) .

In the previous equation, $\mathbf{b}_{i^*}(\mathbf{k})$ is the point where the link $\mathbf{c}_{i^*}$ intersects the boundary, and $q \in [0, 1)$ is the distance in *lattice units* between the boundary

[c]From *M.Bouzidi, M.Firdaouss, P.Lallemand*, reference [4].

and last fluid node, along the link $\mathbf{c}_i$:

$$q = \frac{|\mathbf{b}_{i^*}(\mathbf{k}) - \mathbf{x}_\mathbf{k}|}{h}$$

The additional momentum on the right hand side is used to incorporate the Dirichlet condition on the velocity. The coefficients C_i and D have different expression, according to $q \leq \frac{1}{2}$ or $q > \frac{1}{2}$. This algorithm will be described in chapter 4 in more detail (see also [20], for a general overview of boundary rules). In the simple case $q = 0.5$, we recover the popular *bounce-back* algorithm:

$$\hat{f}_i(n+1, \mathbf{k}) = \hat{f}^c_{i^*}(n, \mathbf{k}) + 2hc_s^{-2} f^*_i \mathbf{c}_i \cdot \mathbf{u}_B(\mathbf{b}_i(\mathbf{k}))\,. \tag{3.43}$$

It is known ([20], summarized in section 4.1.2) that the boundary conditions (3.42) are consistent with the boundary velocity $\mathbf{u}_B$, for any value of $q \in [0, 1)$. The analysis of algorithm

$$L(h, \hat{f}_h)(n, \mathbf{j}) := \begin{cases} \mathrm{LBM}_i(h, \hat{f}_h)(n, \mathbf{j}) & (n, \mathbf{j}) \in \Omega : \mathbf{x}_\mathbf{j} - h\mathbf{c}_i \in \Omega \\ BFL_i(h, \hat{f}_h)(n, \mathbf{j}) & \text{otherwise} \end{cases} \tag{3.44}$$

produces a minimum class in the regular ansatz whose predictions have the same coefficients, for $k = 0, 1, 2$, as the interior LBM, with the additional constraint that the field $\mathbf{u}^{(1)}$ is the particular solution to Navier-Stokes which satisfies

$$\mathbf{u}^{(1)}(\mathbf{x}) = \mathbf{u}_B(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega.$$

Therefore, we can use the prediction which gives second order accurate velocity and first order pressure also for LBM^{BFL}. Nevertheless, this does not assure the applicability of the boundary rule also for the initialization algorithm.

Initialization with BFL rule

Figures 3.7-3.9 show some examples of application of the rule (3.42), implemented on the boundary during the initialization algorithm.
We consider first the linear problem DBC, with the grid placed at distance $q = \frac{1}{2}$ from the boundary (bounce-back is used). Then we simulate the Taylor vortex TV-NS, forcing the Dirichlet conditions on $x = 0$, $x = 1$, placing the grid inside $[0, 1)^2$ in such a way to have the left and the right boundaries at distances (resp.) $q = 0.75$ and $q = 0.25$. To measure the error in pressure, we take the *overall maximum* in space

$$E^p(t_n) := \max_{\mathbf{j} \in \mathcal{G}(h)} \|\hat{p}(n, \mathbf{j})) - p(t_n, \mathbf{x}_\mathbf{j})\|\,. \tag{3.45}$$

The procedure does not work, because an inconsistent pressure is produced.

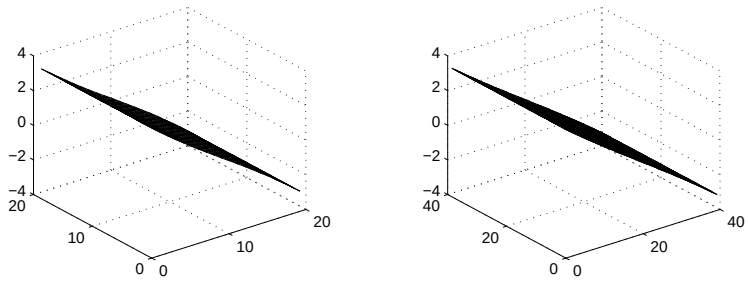


Figure 3.7: Examples of the error in pressure after initialization routine for the problem DBC, applying the bounce-back rule (3.43), for $h = 0.05$ (**left**) and $h = 0.025$ (**right**). The algorithm has been run until it showed convergence. However, the limit does not approximate the initial pressure.

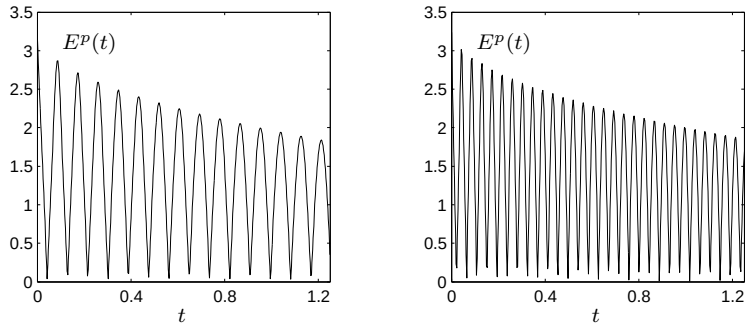


Figure 3.8: Initial layer in pressure simulating the benchmark DBC, after having run an initialization algorithm. The quantity $E^p(t)$ is shown, versus time t. Grid size: $h = 0.05$ (**left**), $h = 0.025$ (**right**). The viscosity is $\nu \sim 0.03$ ($\tau = 0.56$).

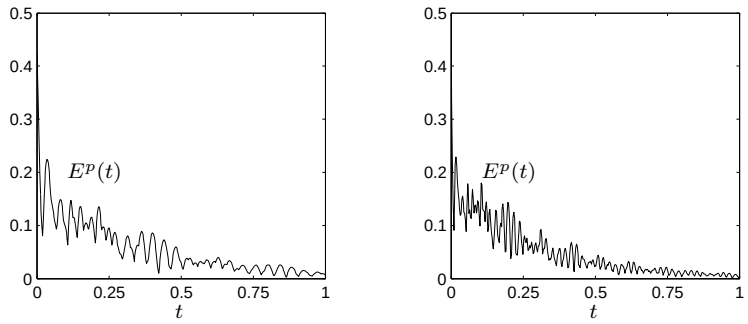


Figure 3.9: Maximum error in pressure $E^p(t)$ in the problem TV-NS, initializing the LBM with algorithm 3.2+BFL rule. Simulation with $h = 0.05$ (**left**) and with $h = 0.025$ (**right**).

Usually, analyzing the classical LBM through a truncated expansion, we are interested in the results for pressure and velocity, whose approximations are contained in the coefficients $f^{(k)}$ for $k \leq 2$. Thus, a truncated expansion (chapter 1)

$$\hat{F} = f^{(0)} + hf^{(1)} + h^2 f^{(2)}$$

is enough to predict an approximate Navier-Stokes solution. The present case is different, since we deal with additional relations for the coefficients $f^{(3)}$ (through the moment $\mathbf{w}$), which influence the field p. In the shown examples, the initialization fails, because the boundary algorithm (3.42) is not consistent with equation (3.41).
We have to understand how the coupling of algorithm 3.1- equation (3.42) acts on the coefficients $f_i^{(3)}$, and, if possible, to introduce a correction term.

3.3.3 Corrected initialization

Enforcing consistent boundary conditions As the initialization rule, the boundary algorithm represents a scheme which is coupled with the interior LBM. Hence, to justify the usage of the interior prediction also for the boundary rule, the coefficients of the expansions have to coincide up to order three, since we want to predict also the field $\mathbf{w}$.
Remark. In other words, to state a result analogous to corollary 1.1 also for the non periodic initialization algorithm, we need *third order boundary conditions*.
Furthermore, we want to achieve conditions (3.41).

Simple case: Bounce-Back We look more carefully at the behavior of the initialization algorithm when a bounce-back boundary condition (3.43), in the case $q = \frac{1}{2}$, is applied.
According to the notations in equation (3.42), let i be an *incoming* direction on the node $\mathbf{x_k}$, so that the last point on the LB grid can be written as $\mathbf{x_k} = \mathbf{b}_{i^*}(\mathbf{k}) + \frac{h}{2}\mathbf{c}_i$. We use a Taylor expansion in relation (3.43), around $\mathbf{b}_{i^*}(\mathbf{k})$, sorting then the different orders in h. As previously remarked, $f_i^{(1)}$ and $f_i^{(2)}$ are consistent with the interior expansion. Therefore, we focus only on the third order coefficient.

The strategy is to first define a target expansion, based on the interior coefficients (3.17), restricted with (3.40)-(3.41) on the boundary. Then, to look for a correction (section 1.3.5) to the boundary condition rule, so that the resulting prediction coincides with the target one.
First of all, the coefficients $f^{(3)}$ at the node updated with the boundary algorithm have to coincide, up to the third order, with the inner coefficient (3.17) of the initialization routine. Observe that with an LB-boundary rule is only applied for the *incoming directions* at the boundary, which satisfy the condition $\mathbf{c}_i \cdot \mathbf{n} < 0$, being $\mathbf{n}$ the outgoing normal.

Inserting the values for $f_i^{(0)}$, $f_i^{(1)}$, $f_i^{(2)}$ into $f_i^{(3)}$, we have for algorithm (3.14)

$$f_i^{INIT,(3)} = f_i^* c_s^{-2} \left(p^{(3)} - \tau \mathbf{c}_i \cdot \left(\nabla p + \nabla Q_i^0 - \left(\tau - \frac{1}{2} \right) \mathbf{D}_i^0 - \mathbf{G}_0 \right) \right). \quad (3.46)$$

We have used

$$Q_i^0 = \frac{1}{2} \left(c_s^{-2} |\mathbf{c}_i \cdot \mathbf{u}_0|^2 - |\mathbf{u}_0|^2 \right), \quad \mathbf{D}_i^0 = (\mathbf{c}_i \cdot \nabla)^2 \mathbf{u}_0.$$

Note that Q_i^0 and $\mathbf{D}_i^0$ are constant during the initialization algorithm, since they only depend on $\mathbf{u}_0$.
Additionally, we enforce the following condition for the gradient of pressure:

$$\nabla p = \nabla p_0. \quad (3.47)$$

Therefore we end up with a third order coefficient (called *exact*), which looks like

$$\begin{aligned} f_i^{EX,(3)} &= f_i^* c_s^{-2} p^{(3)} \\ &- \tau f_i^* c_s^{-2} \mathbf{c}_i \cdot \left(-\partial_t \mathbf{u}|_{t=0} + \nu \nabla^2 \mathbf{u}_0 - \nabla \cdot (\mathbf{u}_0 \otimes \mathbf{u}_0) + \nabla Q_i^0 - \left(\tau - \frac{1}{2} \right) \mathbf{D}_i^0 \right). \end{aligned} \quad (3.48)$$

On the other hand, expanding algorithm (3.43), inserting the post-collision distribution, for $\hat{f}_{i^*}^c(n, \mathbf{k})$, and isolating the third order term, we found (the superscript BB denote the bounce-back rule)

$$\begin{aligned} f_i^{BB,(3)} &= f_i^* c_s^{-2} p^{(3)} \\ &+ f_i^* c_s^{-2} \mathbf{c}_i \cdot \left((\tau - 1) \nabla p + \left(\tau - 1 - \frac{1}{2\tau} \right) \nabla Q_i^0 - \left(\tau^2 - \frac{3}{2}\tau + \frac{1}{4} \right) \mathbf{D}_i^0 - \tau \mathbf{G}_0 \right). \end{aligned} \quad (3.49)$$

All the quantities are evaluated at boundary point $\mathbf{b}_{i^*}(\mathbf{k})$ and pseudotime t_n. Comparing (3.49) and (3.48), the condition to be satisfied (to have the required third order coefficient in the prediction for the boundary rule) is

$$\forall i : \ \mathbf{c}_i \cdot \mathbf{n}(\mathbf{b}_{i^*}(\mathbf{k})) < 0, \ f_i^{BB,(3)}(t_n, \mathbf{b}_{i^*}(\mathbf{k})) = f_i^{EX,(3)}(t_n, \mathbf{b}_{i^*}(\mathbf{k})). \quad (3.50)$$

It is, in general, not true.
To force (3.50), we introduce the following modification. Through the $\mathbb{R}^b$ vector

$$\mathbf{W}_i = (1 - 2\tau) \nabla p_0 - \frac{1}{\tau} \left(2\tau^2 - \tau - \frac{1}{2} \right) \nabla Q_i^0 + \lambda(\tau) \mathbf{D}_i^0 + 2\tau \mathbf{G}_0, \quad (3.51)$$

where $\lambda(\tau) = \tau^2 - \frac{3}{2}\tau + \frac{1}{4}$, we define the *corrected bounce-back*

$$\hat{f}_i(n+1, \mathbf{k}) = \hat{f}_{i^*}^c(n, \mathbf{k}) + 2hc_s^{-2} f_i^* \mathbf{c}_i \cdot \mathbf{u}_B(\mathbf{b}_{i^*}(\mathbf{k})) + h^3 f_i^* c_s^{-2} \mathbf{c}_i \cdot \mathbf{W}_i, \quad (3.52)$$

which leads, by construction, to the equality (3.50).

Numerical tests We test the corrected routine on the simple problem DBC, with boundaries parallel to the y-axes, located in $x = 0$ and $x = 1$. The last fluid nodes are placed at half-grid space far from the boundary.
Since

$$\mathbf{D}_i^0 = Q_i^0 = \nabla^2 \mathbf{u}_0 = \nabla p(t = 0, \mathbf{x})|_{\partial\Omega} = 0,$$

the corrected boundary algorithm reads in this case

$$\hat{f}_i(n+1, \mathbf{k}) = \hat{f}^c_{i^*}(n, \mathbf{k}) + h 2 f_i^* c_s^{-2} \mathbf{c}_i \cdot \mathbf{u}_0(\mathbf{b}_{i^*}(\mathbf{k})) + h^3 2\tau f_i^* c_s^{-2} \mathbf{c}_i \cdot \mathbf{G}_0. \qquad (3.53)$$

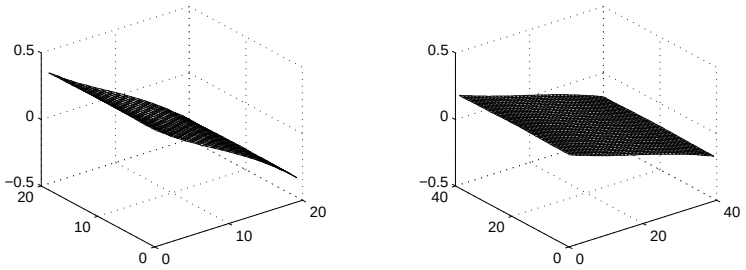


Figure 3.10: Problem DBC. We simulate the flow field on a periodic strip of unitary length and width (boundary on $x = 0$ and $x = 1$) for different grid sizes. Viscosity is $\nu = 0.03$. The figures show the errors in pressure field using the corrected bounce-back rule (3.53) (compare with figure 3.7), for $h = 0.05$ (**left**) and $h = 0.025$ (**right**). The accuracy is now first order. The leading term in the error can be predicted from the coefficient $p^{(3)}$ of the expansion for pressure. Due to the simplicity of the problem, the leading order error is a linear function. Observe that in this case the prediction for the leading order error $p^{(3)}$ is a solution to $\nabla^2 p^{(3)} = 0$.

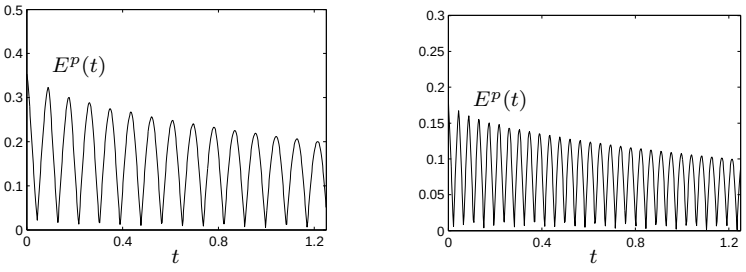


Figure 3.11: Error $E^p(t)$ after corrected *bounce-back*. Comparisons between different grid, $h = 0.05$ (dashed) and $h = 0.025$ (solid).

The error in the initial pressure and the initial layer in time, in figures 3.10-3.11, after the corrected initialization have been reduced by one order in h.

General correction

In its general form, the BFL can be corrected, proceeding in the same way as the bounce-back. The algorithm (3.42) has to be modified as

$$\begin{aligned}\hat{f}_i(n+1,\mathbf{k}) = C_1(q)\hat{f}^c_{i^*}(n,\mathbf{k}) + C_2(q)\hat{f}^c_{i^*}(n,\mathbf{k}+\mathbf{c}_i) + C_3(q)\hat{f}^c_i(n,\mathbf{k}) \\ + hD_i(q)f^*_i c_s^{-2}\mathbf{u}_B(\mathbf{b}_{i^*}(\mathbf{k}))\cdot\mathbf{c}_i + h^3 c_s^{-2} f^*_i \mathbf{c}_i \cdot \mathbf{W}_i(q,\tau,\mathbf{u}_0)(\mathbf{b}_{i^*}(\mathbf{k})),\end{aligned} \quad (3.54)$$

in order to match the expansion (3.17), for the interior domain, and the gradient ∇p_0 on the boundary.
Performing the computation[d], to satisfy a relation analogous to (3.50), we found for the correction the expression

$$\mathbf{W}_i(q,\tau,\mathbf{u}_0) = A(q,\tau)\nabla p_0 + F(q,\tau)\mathbf{G}_0 + Q(q,\tau)\nabla Q^0_i + D(q,\tau)\mathbf{D}^0_i, \quad (3.55)$$

where

$$\begin{aligned}A(q,\tau) &= \begin{cases} 2q-2\tau & q\le\frac{1}{2} \\ -1+\frac{1-\tau}{q} & q>\frac{1}{2}\end{cases}, \\ F(q,\tau) &= \begin{cases} 2\tau & q\le\frac{1}{2} \\ \frac{\tau}{q} & q>\frac{1}{2}\end{cases}, \\ Q(q,\tau) &= \begin{cases} 2q-2\tau+\frac{1-q}{\tau} & q\le\frac{1}{2} \\ -\frac{\tau}{q}-1+\frac{1}{q}+\frac{q}{\tau} & q>\frac{1}{2}\end{cases}, \\ D(q,\tau) &= \begin{cases} 2\tau^2+\tau(-1-2q)+q-q^2 & q\le\frac{1}{2} \\ \frac{1}{q}\tau^2+\tau\left(1-\frac{3}{2q}\right)+\frac{1}{2q}(1-q-q^2) & q>\frac{1}{2}\end{cases} \equiv \kappa_q(\tau).\end{aligned} \quad (3.56)$$

In practice, the correction term $\mathbf{W}_i$ can be computed starting from the initial data $\mathbf{u}_0$, analytically (if possible), or approximating the needed derivatives. Concerning the gradient of the initial pressure ∇p_0, it can be recovered from the Navier-Stokes equation, as a function of $\mathbf{u}_0$ and $\partial_t\mathbf{u}|_{t=0}$. On the boundary, both the functions are known. In the next section 3.4.1 some efficient procedure to evaluate these and other terms involving the initial data $\mathbf{u}_0$ will be analyzed.

Numerical tests The second set of numerical tests regards the vortex solution TV-NS, defined in section 3.1.1, including two straight vertical boundaries, placed at distances $q_W = 0.25$ (west boundary) and $q_E = 0.75$ (east boundary) from the first and the last point in horizontal direction.
The results of the simulations, where the correction (3.51) has been used, are shown in figure 3.12. The terms appearing in $\mathbf{W}_i$ have been computed analytically.

[d]The computation has been carried out with MAPLE.

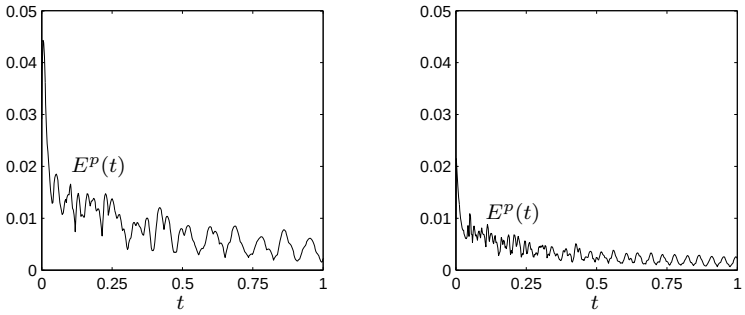


Figure 3.12: Error $E^p(t)$ in pressure using corrected boundary conditions, for the same benchmark as in figure 3.9. Grid $h = 0.05$ (dashed) and $h = 0.025$ (solid) are compared.

Remark. Note that in the problem TV-NS the correction consists only of the term $\mathbf{D}_i^0$ (the other functions vanish on the boundary), which appears multiplied by the polynomial $\kappa_q(\tau)$. In the case of $q = 0.5$ (bounce-back),

$$\kappa_{\frac{1}{2}}(\tau) = 2\tau^2 - 2\tau + \frac{1}{4} = \lambda(\tau), \tag{3.57}$$

which has two real roots, $\tau_\pm = \frac{1}{2} + \frac{1}{\sqrt{8}}$, with $\tau_+ \sim 0.854$. The performed analysis predicts the existence of a special value of viscosity for which, theoretically, no correction is needed, to match the desired condition.
As additional validation of the asymptotic expansion technique, figure 3.13 confirms this expectation. The error (overall maximum) in pressure, drawn as a function of τ, takes its minimum near τ_+.

3.3.4 Accelerated routines

Another important aspect of the initialization algorithm concerns its efficiency. The convergence of p towards the initial pressure p_0 depends on viscosity and can become very slow, for small ν. As in the periodic case, we want to speed up the procedure, by using a different value of τ, also in presence of boundaries.
Running the initialization with a different relaxation time τ^I (related viscosity ν^I), the coefficients of the expansion are (compare with (3.17)):

$$\begin{aligned}
f_i^{\tau^I,(1)} &= f_i^* c_s^{-2} \mathbf{c}_i \cdot \mathbf{u}_0 \\
f_i^{\tau^I,(2)} &= f_i^* c_s^{-2} p^I + H_i^{Q(eq)}(\mathbf{u}_0, \mathbf{u}_0) - \tau^I f_i^* c_s^{-2} (\mathbf{c}_i \cdot \nabla)\mathbf{c}_i \cdot \mathbf{u}_0, \\
f_i^{\tau^I,(3)} &= f_i^* c_s^{-2} \left(p^{(3)} - \tau^I \mathbf{c}_i \cdot \left(\nabla p^I + \nabla Q_i^0 - \left(\tau^I - \tfrac{1}{2}\right) \mathbf{D}_i^0 - \mathbf{G}_0 \right) \right)
\end{aligned} \tag{3.58}$$

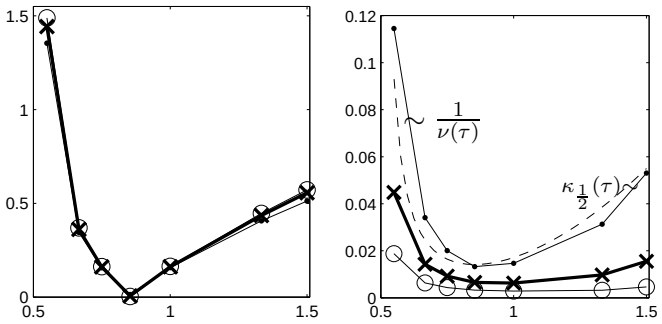


Figure 3.13: Problem TV-NS, boundaries placed at $q = \frac{1}{2}$. Value of $\max_{\mathbf{j}\in\mathcal{G}(h)} \left|\hat{p}(0,\mathbf{j}) - p_{\text{TV-NS}}(0,\mathbf{x}_\mathbf{j})\right|$ after initialization, as function of τ. **Left:** Error with the bounce-back rule, for $h = 0.05$ (points), $h = 0.025$ (crosses) and $h = 0.0125$ (circles). The error is not decreasing, and it takes its minimum close to τ_2^*. **Right:** Now the correction is used, first order accuracy is reached. The error behaves like $\frac{1}{\nu}$, for $\nu \ll 1$, and like $\kappa_{\frac{1}{2}}(\tau)$ far from $\frac{1}{2}$ (dotted line).

where p^I is solution of a system of PDE analogous to (3.18), with τ^I and ν^I replacing the original τ and ν:

$$\begin{aligned} &\nabla p^I + (\mathbf{u}_0\cdot\nabla)\mathbf{u}_0 = \nu^I\nabla^2\mathbf{u}_0 + \mathbf{G}_0 - \frac{1}{\tau^I}\mathbf{w} \\ &c_s^{-2}\partial_t p^I + \nabla\cdot\mathbf{w} + \frac{1}{2}(\nabla^2 p^I + \nabla\cdot(\mathbf{u}_0\otimes\mathbf{u}_0)) = \frac{1}{2}\nabla\cdot\mathbf{G}_0 \end{aligned} \tag{3.59}$$

In the interior domain, an *accelerated* equation for p^I is obtained:

$$\partial_t p^I = \nu^I\nabla^2(p^I - p_0). \tag{3.60}$$

So far, everything works just as in the previous periodic boundary situation (section 3.2.2). If we proceed similarly as in section 3.3.3, the corrected BFL rule (3.54) involves now the vector

$$\mathbf{W}_i(q,\tau,\tau^I,\mathbf{u}_0) = A(q,\tau^I)\nabla p_0(\nu) + F(q,\tau^I)\mathbf{G}_0 + Q(q,\tau^I)\nabla Q_i^0 + D(q,\tau^I)\mathbf{D}_i^0. \tag{3.61}$$

It is analogous to the previous (3.51), only the dependence on the original viscosity ν is present in the term ∇p_0.
Combining the correction for $f_i^{(2)}$ derived before, (section 3.2.2) and the one for $f_i^{(3)}$, we can formally define

Algorithm 3.3 (General LB-Initialization Algorithm).
given initial data $\mathbf{u}_0$ and force $\mathbf{G_0}$
`compute` $O(h)$-`approximation of` $\nabla\cdot\mathbf{G}_0$, g^∇

```
compute O(h)-approximation of ∇p_0 on the boundary
set fast relaxation time → τ^I
compute O(h)-approximation of W_i
DO n = 1, 2, ...
  interior:  algorithm 3.1
  BC: rule (3.54)
compute pressure p and equilibrium from p and u_0
initialize LBM using (3.32)
```

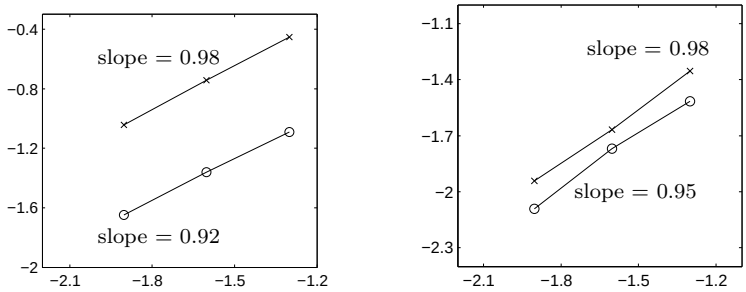


Figure 3.14: Order plot of maximum error in pressure, for DBC problem (**left**) and vortex field (**right**). Comparisons between *original viscosity initialization*, $\nu \sim 0.03$ ($\tau = 0.59$), (crosses) and the *accelerated routines*, $\tau^I = 1$ (circles). The routines have been run under a convergence criterion based on the difference between two successive iteration of pressure. The gain in computational time is about 70% (in both the cases). The slopes refer to 1-degree polynomials which are least squares approximations of the discrete data.

Initial time derivative

Using the results of the analysis concerning the moments, other interesting relations can be derived. Comparing the equations

$$\nabla p + (\mathbf{u}_0 \cdot \nabla)\mathbf{u}_0 = \nu^I \nabla^2 \mathbf{u}_0 + \mathbf{G}_0 - \frac{1}{\tau^I}\mathbf{w},$$

relating p and $\mathbf{w}$ in algorithm 3.3, and

$$\partial_t \mathbf{u}|_{t=0} + \nabla p_0 + (\mathbf{u}_0 \cdot \nabla)\mathbf{u}_0 = \nu \nabla^2 \mathbf{u}_0 + \mathbf{G}_0$$

(Navier-Stokes at initial time), an explicit expression for $\mathbf{w}$ is recovered:

$$\mathbf{w} = \tau^I \left(\partial_t \mathbf{u}|_{t=0} + (\nu^I - \nu)\nabla^2 \mathbf{u}_0 \right). \tag{3.62}$$

Assuming that our prediction approximates the numerical solution $\hat{f}$, in view of definition (3.24), the field $\mathbf{w}$ can be extracted as:

$$\hat{\mathbf{w}} = \frac{\sum_i \mathbf{c}_i \hat{f}_i - h\mathbf{u}_0}{h^2} = \mathbf{w} + O(h). \tag{3.63}$$

Equation (3.62) predicts that, after the initialization procedure, the populations contain an approximated initial time derivative of velocity:

$$\frac{\sum_i \mathbf{c}_i \hat{f}_i - h\mathbf{u}_0}{\tau^I h^2} - (\nu^I - \nu)\nabla^2 \mathbf{u}_0 = \partial_t \mathbf{u}|_{t=0} + O(h). \tag{3.64}$$

Observe that an estimate of $\nabla^2 \mathbf{u}_0$ is also needed, using a relaxation time $\tau^I \neq \tau$ (see section 3.4.1).

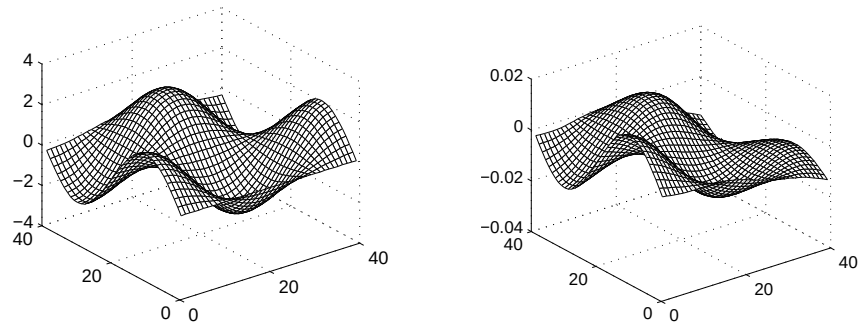


Figure 3.15: Approximation of the initial time derivative $(\partial_t \mathbf{u}_{0x})|_{t=0}$, through the field $\frac{1}{\tau^I}\hat{\mathbf{w}}_x$, from (3.62). Grid size is $h = 0.025$, Original viscosity $\nu = 0.03$ ($\tau = 0.59$). Accelerated routine has been run with $\nu^I = \frac{1}{6}$ ($\tau^I = 1$). **Left:** Surface plot of the approximation. **Right:** Surface plot of the error.

3.4 Further remarks

The asymptotic expansion approach has successfully predicted the behavior of the LBM. In particular, we have corrected the original algorithm starting from the prediction we have been looking for (lemma 1.10).
In this conclusive section, we use the expansion in the opposite way: given an algorithm, if we are able to predict the results it produces, useful routines can be constructed.

3.4.1 Overfrozen LBM

So far, we have constructed a solution to the Poisson equation (3.10) as well as the time derivative $\partial_t \mathbf{u}|_{t=0}$, only using a *modified equilibrium* in the lattice Boltzmann equation, with a **frozen** velocity field.

Keeping also the density ρ_0 fixed in the equilibrium function, we define now the **overfrozen LBE**:

$$\hat{f}_i(n+1, \mathbf{j}+\mathbf{c}_i) = \hat{f}_i(n, \mathbf{j}) + \frac{1}{\tau}\left(f_i^{eq}(\rho_0(\mathbf{x_j}), h\mathbf{u}_0(\mathbf{x_j})) - \hat{f}_i(n, \mathbf{j})\right) + g_i^0(\mathbf{j}). \quad (3.65)$$

Observe that the operator,

$$J_i^{00}(f) = \frac{1}{\tau}\left(H_i^{eq}(\rho_0, h\mathbf{u}_0) - f_i\right) \quad (3.66)$$

does not conserve neither the mass nor the velocity.
Performing the asymptotic expansion starting with a regular ansatz of type, we find the coefficients

$$\begin{aligned} f_i^{00,(1)} &= f_i^* c_s^{-2} \mathbf{c}_i \cdot \mathbf{u}_0 \\ f_i^{00,(2)} &= f_i^* p_0 + H_i^{Q(eq)}(\mathbf{u}_0, \mathbf{u}_0) - \tau f_i^* c_s^{-2} (\mathbf{c}_i \cdot \nabla)\mathbf{c}_i \cdot \mathbf{u}_0, \\ f_i^{00,(3)} &= -\tau c_s^{-2} f_i^* \left(\mathbf{c}_i \cdot \nabla p_0 + (\mathbf{c}_i \cdot \nabla) Q_i^0 - \left(\tau - \frac{1}{2}\right)(\mathbf{c}_i \cdot \nabla)^2 \mathbf{c}_i \cdot \mathbf{u}_0 - \mathbf{G}_0\right), \end{aligned} \quad (3.67)$$

p_0 being an initial given pressure field.
The coefficients (3.67) do not depend on time. However, the prediction is stationary only if we start with a second order initialization, which already realizes conditions (3.67). Therefore, we will analyze algorithm (3.65) in case of starting with *inaccurate* equilibrium initial values. Of course, this overfrozen method could not be regarded as a solver for hydrodynamical problem. Nevertheless, we will show some simple applications hidden in it.

Fast non-equilibrium set up

We have computed (section 2.3) a prediction for the initial oscillatory layer in the stress tensor appearing using an equilibrium initialization, due to the grid scale oscillation (equation (2.98))

$$\sigma_i(n, \mathbf{j}) = \left(1 - \frac{1}{\tau}\right)^n \tau c_s^2 \mathbf{c}_i \cdot \nabla \mathbf{u}_0(\mathbf{x_j}) \cdot \mathbf{c}_i \quad (3.68)$$

which affects the coefficient $f_i^{(2)}$. The derivation of equation (3.68) works also in the framework of the overfrozen LBM, initializing with (3.9) or (3.11).
Now, let assume we run the method (3.65), performing linear iterations ($H^{Q(eq)} = 0$), with $\tau = 1$ and density $\rho_0 = 0$. The predicted additional layer σ_i vanishes after a single time step, and the relation

$$f_i^{00,(2)}(n, \mathbf{j}) = f_i^* c_s^{-2} (\mathbf{c}_i \cdot \nabla)\mathbf{c}_i \cdot \mathbf{u}_0(\mathbf{x_j}) \quad (3.69)$$

holds for $n \geq 2$. It is therefore possible to set up the non-equilibrium part of the initial distribution in a fast and efficient way.

Algorithm 3.4 (Non-Equilibrium set up).
set $\rho_0 = 0$, $\tau = 1$
$\hat{f}(0, \mathbf{j}) = H_i^{L(eq)}(0, h\mathbf{u}_0(\mathbf{x_j}))$
`linear overfrozen LBM:`
$\hat{f}_i(n+1, \mathbf{j} + \mathbf{c}_i) = H_i^{L(eq)}(0, h\mathbf{u}_0(\mathbf{x_j}))$
`extract second order:`
$\hat{f}_i^{(2)}(\mathbf{j}) = \hat{f}_{ih}(n+1, \mathbf{j}) - hc_s^{-2} f_i^* \mathbf{c}_i \cdot \mathbf{u}_0(\mathbf{x_j})$
initial non-equilibrium : $f_i^{neq,(2)} = \tau \hat{f}_i^{(2)} + O(h)$

Stress tensor and Laplacian evaluations

Running the overfrozen method, on a periodic domain, with

$$\rho_0 = 0, \quad \tau = 1, \quad \mathbf{G} = 0, \quad H^{Q(eq)} = 0 \text{ (linear)},$$

the predicted coefficients read

$$\begin{aligned} f_i^{(1)} &= f_i^* c_s^{-2} \mathbf{c}_i \cdot \mathbf{u}_0 \\ f_i^{(2)} &= -f_i^* c_s^{-2} (\mathbf{c}_i \cdot \nabla) \mathbf{c}_i \cdot \mathbf{u}_0, \\ f_i^{(3)} &= \frac{1}{2} c_s^{-2} f_i^* (\mathbf{c}_i \cdot \nabla)^2 \mathbf{c}_i \cdot \mathbf{u}_0, \end{aligned} \tag{3.70}$$

for $n \geq 2$.
Again, despite of wrong initial conditions, equations (3.70) predict the behavior of $\hat{f}$ for $n > 1$, since the arising initial layers relaxes to zero after a single iteration, due to the choice $\tau = 1$. Taking the second order moment we conclude:

$$\frac{1}{h^2} \sum_i \hat{f}_i \mathbf{c}_i \otimes \mathbf{c}_i \hat{f} = c_s^2 (\nabla \mathbf{u}_0 + \nabla \mathbf{u}_0^T) + O(h^2), \tag{3.71}$$

which gives a second order approximation of the symmetric part of the gradient of the initial velocity field.
Result (3.71) is already contained in the previous section 3.4.1. Additionally, evaluating the first order moment of the population,

$$\frac{1}{h} \sum_i \hat{f}_i \mathbf{c}_i = \mathbf{u}_0 + \frac{1}{2} c_s^2 h^2 \nabla^2 \mathbf{u}_0 + O(h^3), \tag{3.72}$$

we obtain an approximation of the Laplacian of $\mathbf{u}_0$:

$$\hat{\mathbf{L}}[\mathbf{u}_0] := \frac{1}{h^3} 2 c_s^{-2} \left(\sum_i \hat{f}_i \mathbf{c}_i - h\mathbf{u}_0 \right) = \nabla^2 \mathbf{u}_0 + O(h). \tag{3.73}$$

In conclusion, given a field $\mathbf{u}_0 : \Omega \to \mathbb{R}$ on a periodic domain Ω, we have:

Algorithm 3.5 ($\mathbf{u}_0$-Derivatives evaluation).
`set parameters:` $\rho = 0$, $\tau = 1$
`initial conditions:` $\hat{f}(0, \mathbf{j}) = H^{L(eq)}(0, h\mathbf{u}_0(\mathbf{x_j}))$
`linear overfrozen LBM:`
$\hat{f}_i(n+1, \mathbf{j} + \mathbf{c}_i) = H_i^{L(eq)}(0, h\mathbf{u}_0(\mathbf{x_j}))$
`extract stress tensor:`
$$\hat{\mathbf{S}} = \frac{c_s^{-2}}{h^2} \sum_i \hat{f}_i \mathbf{c}_i \otimes \mathbf{c}_i$$
`extract Laplacian:`
$$\hat{L}(\mathbf{u}_0) = \frac{1}{h^3} 2c_s^{-2} \left(\sum_i \hat{f}_i \mathbf{c}_i - h\mathbf{u}_0 \right)$$

In general, for bounded domains, the above procedure 3.5 could fail *on the boundary*. However, it can still be regarded as an efficient routine, to be coupled with other approaches (e.g. finite differences, etc.) on the boundary nodes.
In figure 3.16 we check the approximations of $\mathbf{S}(\mathbf{u}_0)$ and $\nabla^2 \mathbf{u}_0$ for $\mathbf{u}_0 = \mathbf{u}_{TV}$ over the periodic domain $\Omega = [0, 1)^2$, employing algorithm 3.5. The double logarithmic plot, of the maximum overall errors, shows first order accuracy in h.

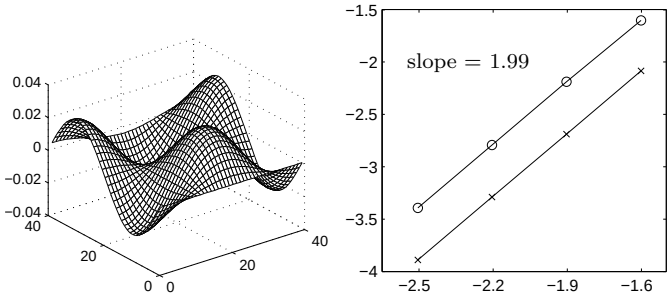


Figure 3.16: **Left:** Approximation of $(\nabla^2 \mathbf{u}_0)_x$ with (3.73), for $\mathbf{u}_0 = \mathbf{u}_{TV}$. The difference $\hat{\mathbf{L}}(\mathbf{u}_0) - \nabla^2 \mathbf{u}_0$ is shown, for a grid $h = 0.025$. **Right:** Double logarithmic plot, for the accuracy of the approximation of $\mathbf{S}[\mathbf{u}_0]$ (sampled with $\times$) and $\nabla^2 \mathbf{u}_0$ ($\circ$).

Finite differences interpretation

Note that equation (3.65) has the form

$$\hat{f}_i(n+1, \mathbf{j} + \mathbf{c}_i) = \left(1 - \frac{1}{\tau}\right) \hat{f}_i(n, \mathbf{j}) + H_i^{eq}(\rho_0(\mathbf{x_j}), h\mathbf{u}_0(\mathbf{x_j})) + \hat{g}_i^0(\mathbf{j}). \tag{3.74}$$

If $\rho_0 = 0$, $\mathbf{G} = 0$, $\tau = 1$, it is a simple *one-step* algorithm.
Removing the dependence on n, we can write the result after a single iteration:

$$\hat{f}(\mathbf{j}) = h c_s^{-2} f_i^* \mathbf{c}_i \cdot \mathbf{u}_0(\mathbf{x}_{\mathbf{j}-\mathbf{c}_i}), \tag{3.75}$$

so that

$$\hat{\mathbf{S}}(\mathbf{j}) = \frac{1}{h} c_s^{-2} \sum_i f_i^* \mathbf{c}_i \otimes \mathbf{c}_i (\mathbf{c}_i \cdot \mathbf{u}_0(\mathbf{x}_{\mathbf{j}-\mathbf{c}_i})), \tag{3.76}$$

and

$$\hat{\mathbf{L}}(\mathbf{u}_0)(\mathbf{j}) = \frac{1}{h^2} 2 c_s^{-2} \left[\sum_i f_i^* c_s^{-2} \mathbf{c}_i (\mathbf{c}_i \cdot \mathbf{u}_0(\mathbf{x}_{\mathbf{j}-\mathbf{c}_i})) - \mathbf{u}_0(\mathbf{x}_\mathbf{j}) \right]. \tag{3.77}$$

We develop equation (3.77) for the D2Q9 model, to establish a connection with finite difference schemes. In this framework, it is useful a compact notation [21], denoting the sum of a discrete function over a 2D-stencil, around a node $\mathbf{j} = (j_x, j_y)$, as a matrix 3×3:

$$\begin{bmatrix} a_{-1,1} & a_{0,1} & a_{1,1} \\ a_{-1,0} & a_{00} & a_{1,0} \\ a_{-1,-1} & a_{0,-1} & a_{1,-1} \end{bmatrix} \mathbf{u}_0(\mathbf{x}_\mathbf{j}) := \sum_{l,m=-1}^{1} a_{l,m} \mathbf{u}_0(\mathbf{x}_{j_x+l, j_y+m}). \tag{3.78}$$

Inserting the value of the entries, we obtain

$$\hat{L}_x(\mathbf{u}_0) = \frac{1}{h^2} \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} \\ 2 & -6 & 2 \\ \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix} u_0(\mathbf{x}_\mathbf{j}) + \begin{bmatrix} -\frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 \\ \frac{1}{2} & 0 & -\frac{1}{2} \end{bmatrix} v_0(\mathbf{x}_\mathbf{j}), \tag{3.79}$$

where we recognize a first order stencil for the Laplacian $\nabla^2 u_0$ and a weighted averaged of values of v_0, which vanishes up to the order h^3. Similarly:

$$\hat{L}_y(\mathbf{u}_0) = \frac{1}{h^2} \begin{bmatrix} -\frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 \\ \frac{1}{2} & 0 & -\frac{1}{2} \end{bmatrix} u_0(\mathbf{x}_\mathbf{j}) + \begin{bmatrix} \frac{1}{2} & 2 & \frac{1}{2} \\ 0 & -6 & 0 \\ \frac{1}{2} & 2 & \frac{1}{2} \end{bmatrix} v_0(\mathbf{x}_\mathbf{j}) \tag{3.80}$$

approximate the y-component.
In conclusion, the results (3.71)-(3.73) express nothing else but finite difference approximations of the gradient and the Laplacian of the field $\mathbf{u}_0$, contained in the special LB iteration. Analogous considerations have been done in [21] for the classical LB method.

3.4.2 Higher order initialization

After the analysis, the question might arise, whether it is possible to further improve the initial value, defining initializations of higher orders.
Let us consider the periodic case. For the third order, we would need to construct an initialization consistent with the definition of the coefficient $f^{(3)}$ of (1.139) (interior algorithm):

$$\begin{aligned} f_i^{(3)}(0, \mathbf{x}) = f_i^* c_s^{-2} \mathbf{c}_i \cdot \mathbf{u}^{(3)}(0, \mathbf{x}) - \\ \tau c_s^{-2} f_i^* \mathbf{c}_i \cdot \left(\partial_t \mathbf{u}(t, \mathbf{x})|_{t=0} + \nabla p_0(\mathbf{x}) + \nabla Q_i^0(\mathbf{x}) - \left(\tau - \tfrac{1}{2}\right) \mathbf{D}_i^0(\mathbf{x}) - \mathbf{G}_0(\mathbf{x}) \right). \end{aligned} \tag{3.81}$$

As before, the only data we have is the initial velocity field $\mathbf{u}_0$. To set up a higher order initialization, we need to compute, at least with first order accuracy, all the fields appearing in (3.81).
Manipulating the initialization routines described throughout the chapters, and conventional approximation formulas (e.g. finite differences), the quantities derived from the initial data $\mathbf{u}_0$ and p_0 can be extracted. With the procedure of section 3.3.4, also the initial time derivative is available.
An additional equation for the field $\mathbf{u}^{(3)}(t, \mathbf{x})$ is recovered in order to cancel the projected residue (equation (1.142) recovered analyzing the standard LBM, section 1.3.2).
To have a third order initialization, $\mathbf{u}^{(3)}$ should satisfy

$$\nabla \cdot \mathbf{u}^{(3)}(0, \mathbf{x}) = -c_s^2 \partial_t p^{(2)}(0, \mathbf{x}) + \frac{c_s^2}{2} \nabla \cdot \mathbf{G}_0, \quad \mathbf{x} \in \Omega. \tag{3.82}$$

In practice we need additional solvers, which would only lead to a small improvement, in the whole LBM method. Furthermore, in presence of boundaries, using the initialization algorithm 3.3 also a higher-order boundary rule is required to avoid layers in $p^{(3)}$.
At this stage, it seems that the gain in the accuracy we could reach does not actually compensate the required computational effort.

Summary

In this chapter we have analyzed how the LBM can be improved using initialization routines. Starting from an existing algorithm [34] for the periodic boundary, we have discussed its improvement in efficiency and accuracy, using the results of the asymptotic analysis.
Moreover, the extension to boundary value problems, the requirements and the reliability in that case have been investigated.
Finally, we have analyzed further properties of modified initialization routines. It has been illustrated how the analysis can be employed to understand the scheme in depth. Starting from a simple LB-type iteration, we have been able to recover approximations of first and second order derivatives of the velocity field in an efficient way. These results are a simple corollary of the present chapter. However, they might be of interest for future research.

Chapter 4

Moving boundary problems

Using some of the results derived so far, we begin the investigations of possible extensions of the LBM towards moving boundary problems. Since the standard LBM is based on a fixed Cartesian grid, dealing with moving boundaries an additional algorithm is needed, to *initialize* the variables on the new nodes created by the variations of the computational domain.
The problem has been mainly investigated numerically. Different algorithms to approximate the missing information on the new nodes have been proposed and numerically tested in previous works [31, 15]. These schemes are described in the present chapter. Besides, we provide also a short overview of the possible choices of LBM-boundary condition algorithms and the main results of the related asymptotic analysis, as presented in [20, 22].
Our issue is now to use the asymptotic analysis to understand the effect of moving boundaries in the LBM. Modifying the examples proposed in [31] we show how a reliable algorithm, which preserve the accuracy and efficiency of the scheme, can be constructed in a simple way. The results of the analysis will be in the end validated numerically through various benchmarks.
Section 4.1 begins defining a basic LB code for moving boundary problems. Section 4.2 deals with the *refill problem*, analyzing two existing algorithms [31]. A rigorous asymptotic allows to point out the critical steps, advantages and disadvantages of the different approaches. As a result, in section 4.3 we propose and test an improved procedure, based on a similar idea used to implement boundary conditions in [16].

4.1 A Moving boundary lattice Boltzmann

4.1.1 Fixed lattice and moving boundaries

To model a moving boundary problem, we divide a domain $\Omega \subset \mathbb{R}^2$ as

$$\Omega = \Omega_F(t) \cup \Gamma(t) \cup \Omega_S(t),$$

in a fluid part, a solid part and the interface Γ between them. In the fluid subdomain, we consider the following incompressible Navier-Stokes problem with moving boundaries (MB-NS):

$$\begin{cases} \nabla \cdot \mathbf{u} = 0 & t > 0,\ \mathbf{x} \in \Omega_F(t) \\ \partial_t \mathbf{u} + \nabla p + \mathbf{u} \cdot \nabla \mathbf{u} = \nu \Delta \mathbf{u} + \mathbf{G} & t > 0,\ \mathbf{x} \in \Omega_F(t) \\ \mathbf{u}(t, \mathbf{x}) = \mathbf{u}_B(t, \mathbf{x}) \qquad \mathbf{x} \in \Gamma(t) \\ \mathbf{u}(0, \mathbf{x}) = \mathbf{u}_0(\mathbf{x}) \qquad \mathbf{x} \in \Omega_F(0). \end{cases} \tag{4.1}$$

The domain $\Omega_S(t)$ is supposed to be known (*moving boundary problem, without interaction*).
The computational lattice covers both the fluid and the solid part. A *flag* is added to each node of the grid to indicate whether it is a fluid or a solid one. The movement of the boundary can swap the flags, erasing fluid nodes or creating new ones.

Sets of nodes

Within the grid $\mathcal{G}(h)$, we define different classes of points. Running the algorithm, they are processed differently.

Definition 4.1. *Introducing the* generalized *fluid domain*

$$\overline{\Omega}_F = \Omega_F \cup \Gamma, \tag{4.2}$$

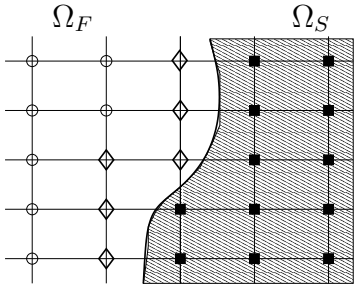
at every time step $t_n = nh^2$ *we define*

- $I_F(n) := \left\{ \mathbf{k} \in \mathcal{G}(h) \mid \mathbf{x}_\mathbf{k} \in \overline{\Omega}_F(t_n) \right\}$, *the* **fluid nodes**,
- $I_{int}(n) := \left\{ \mathbf{k} \in I_F(n) \mid \forall \mathbf{c}_i \in \mathbb{V} :\ \mathbf{x}_{\mathbf{k}+\mathbf{c}_i} \in \overline{\Omega}_F(t_n) \right\}$, *the* **interior nodes**,
- $I_b(n) := \{ \mathbf{k} \in I_F(n) \mid \exists\ \mathbf{c}_i \in \mathbb{V} :\ \mathbf{x}_{\mathbf{k}+\mathbf{c}_i} \in \Omega_S(t_n) \}$, *the* **boundary nodes**.

According to definition (4.2), a node will be *fluid* if belonging to the fluid domain *or* to the interface. In words, $I_F(n)$ contains the nodes which are relevant for the fluid dynamics. Among them, the interior nodes have all neighbors in the fluid domain and are updated with the classical collision+advection algorithm (1.32). The boundary nodes have at least one neighbor in the solid domain.
For the implementation, we associate to each node the **grid flag**:

$$\mathbf{m}_n(\mathbf{k}) = \begin{cases} 1 & \mathbf{k} \in I_{int}(n) \\ 0 & \mathbf{k} \in I_b(n) \\ -1 & \mathbf{k} \notin I_F(n) \end{cases} \tag{4.3}$$

Furthermore, we distinguish the discrete velocities in a boundary node.



1	1	0	-1	-1
1	1	0	-1	-1
1	0	0	-1	-1
1	0	-1	-1	-1
1	0	-1	-1	-1

Figure 4.1: **Left:** At each time step t_n, the nodes have different roles. In the figure, the circles (○) are *internal* nodes, the diamonds (◇) are the *boundary* and the black squares (■) represent the *solid* points, out of the domain $\Omega(t_n)$. **Right:** The translation in terms of **grid flag** (4.3).

Definition 4.2. *Let* $\mathbf{k} \in I_b$. *We define*

- $\mathbb{V}_O(n, \mathbf{k}) = \{\mathbf{c}_i \in \mathbb{V} \mid \mathbf{x}_{\mathbf{k}+\mathbf{c}_i} \in \Omega_S(t_n)\}$, outgoing *directions;*
- $\mathbb{V}_I(n, \mathbf{k}) = \{\mathbf{c}_i \in \mathbb{V} \mid \mathbf{x}_{\mathbf{k}-\mathbf{c}_i} \in \Omega_S(t_n)\}$, incoming *directions;*
- $\tilde{\mathbb{V}}(n, \mathbf{k}) = \mathbb{V} - \mathbb{V}_O - \mathbb{V}_I$.

Note that the set $\tilde{\mathbb{V}}$ is symmetric:

$$\tilde{\mathbb{V}} = -\tilde{\mathbb{V}}. \tag{4.4}$$

Additionally, we collect in a set the **boundary couples**, i.e. couples formed by a boundary node and one of the related outgoing direction:

$$B(\Omega(t_n), h) := \left\{(\mathbf{k}, i) \in \mathcal{G}(h) \times \{1, \dots, b\} \mid \mathbf{x}_{\mathbf{k}} \in \overline{\Omega}_F(t_n),\ \mathbf{x}_{\mathbf{k}+\mathbf{c}_i} \in \Omega_S(t_n)\right\} \tag{4.5}$$

(already introduced in section 1.1). It depends on the structure of the domain, the interface, the grid and the time step. For simplicity, we will denote it as

$$B^n = B(\Omega(t_n), h).$$

It holds

$$(\mathbf{k}, i) \in B^n \iff \mathbf{k} \in I_b(n) \text{ and } \mathbf{c}_i \in \mathbb{V}_O(n, \mathbf{k}).$$

In future, the preference of one notation rather than the other will depend on the context.
Equivalently

$$(\mathbf{k}, i) \in B^n \iff \exists q \in [0, 1) \ :\ \mathbf{x}_{\mathbf{k}} + qh\mathbf{c}_i \in \Gamma(t_n).$$

Note that since the node on the interface are considered fluid, the variable q can assume the value zero. There exists a natural correspondence

$$\begin{array}{ccc} B^n & \rightarrow & \Gamma(t_n) \\ (\mathbf{k}, i) & \mapsto & \mathbf{b}_i^n(\mathbf{k}) \end{array} \tag{4.6}$$

where

$$\mathbf{b}_i^n(\mathbf{k}) := \mathbf{x}_\mathbf{k} + qh\mathbf{c}_i, \tag{4.7}$$

is the boundary point related to the couple $(\mathbf{k}, i)$. In some cases, we will use the set

$$\mathcal{B}^n = \{\mathbf{b}_i^n(\mathbf{k}), \mid (\mathbf{k}, i) \in B^n\} \subset \Gamma, \tag{4.8}$$

which collects the intersections of grid and interface. Calling $\overline{\mathcal{G}(h)}$ the subset of Ω made of the lattice links (equation (1.10)):

$$\mathcal{B}^n = \overline{\mathcal{G}(h)} \cap \Gamma\,. \tag{4.9}$$

If we consider the interface to be a *closed, regular*, curve (in two dimensions), we can introduce a parameterization

$$\begin{array}{ccc} \Gamma & \rightarrow & [0, 2\pi) \\ \mathbf{b} & \mapsto & \gamma \end{array} \tag{4.10}$$

which is a one-to-one correspondence In the following, we will use both the notations $\mathbf{b}(\gamma)$ and $\gamma(\mathbf{b})$ to denote a point on the interface and (resp.) its coordinate.

Refill step The characterization of the grid nodes (4.3) is time depending. According to the movement of the boundary, the nodes can move from the boundary to the internal set, appear or disappear as fluid node. If a node becomes fluid only at time t_{n+1}, i.e.

$$\mathbf{x}_\mathbf{k} \in \overline{\Omega}_F(t_{n+1}),\ \mathbf{x}_\mathbf{k} \notin \overline{\Omega}_F(t_n),$$

the distributions $\hat{f}_i(n+1, \mathbf{k})$ have to be defined. We will call the process dealing with *initialization* of a new fluid node a **refill**.

4.1.2 Description and analysis of the algorithm

According to the previous notations, we define

Algorithm 4.1 (Moving Boundary LB).

```
Initialize variables
DO n = 1, ...
  INTERIOR-LBM: for (n, j, i) : j ∈ I_int(n),   j ∈ I_b(n) ∧ c_i ∉ V_I(n, j):
     collision + advection
```

```
  BOUNDARY CONDITIONS: for (k, i) ∈ B^n:
      boundary condition rule for  f̂_{i*}(n+1, k) (with c_{i*} = −c_i)
  UPDATE DOMAIN AND FLAGS :
      Ω_S(t_n) → Ω_S(t_{n+1}) (known) ⇒ m_n → m_{n+1}
  IF new_fluid_nodes
    {REFILL}
END
```

Algorithm 4.1 is defined in an extremely general way. Going through it, we describe more in detail some possible choices for *boundary condition algorithms* and *refill approaches*.

Use of the Asymptotic analysis

In this case, we want to investigate the algorithm composed by

$$\mathrm{LBM}_i^{MB}(h,\hat{f}_h)(n,\mathbf{j}) := \begin{cases} IC_i(h,\hat{f}_h)(\mathbf{j}) & n=0 \\ L(h,\hat{f}_h)(n,\mathbf{j}) & n>0,\ (\mathbf{j},i^*)\notin B^n \\ BC_i(h,\hat{f}_h)(n,\mathbf{j}) & n>0,\ (\mathbf{j},i^*)\in B^n \\ REF_i(h,\hat{f}_h)(n,\mathbf{j}) & n>0,\ \mathbf{m}_{n-1}(\mathbf{j})<0 \wedge \mathbf{m}_n(\mathbf{j})\geq 0 \end{cases} \tag{4.11}$$

(where i^* is such that $\mathbf{c}_{i^*} = -\mathbf{c}_i$). Here, IC is an initialization rule, LBM is the standard lattice Boltzmann algorithm for the interior domain, BC is the algorithm used for the boundary conditions and REF the one to deal with the refill step.
We want to use the same ansatz $\mathcal{A}^{\mathrm{LBM}}$ obtained for the interior domain, which is represented by the prediction (corollary 1.1 of chapter 1)

$$F_h^{\mathrm{LBM}}(n,\mathbf{j}) = \sum_{k=0}^{2} h^k f^{LBM,(k)}(t_n,\mathbf{x}_\mathbf{j}), \tag{4.12}$$

with coefficients

$$\begin{aligned} f_i^{LBM,(0)} &= f_i^*, \\ f_i^{LBM,(1)} &= f_i^* c_s^{-2}\mathbf{c}_i\cdot\mathbf{u}, \\ f_i^{LBM,(2)} &= f_i^* c_s^{-2} p + H_i^{Q(eq)}(\mathbf{u},\mathbf{u}) - \tau f_i^* c_s^{-2}(\mathbf{c}_i\cdot\nabla)\mathbf{c}_i\cdot\mathbf{u}, \end{aligned} \tag{4.13}$$

where $\mathbf{u}$ and p solve the Navier-Stokes moving boundary problem.
To check whether it is possible to use the prediction F^{LBM} in this case, we analyze the algorithm (4.11) starting from the ansatz $\mathcal{A}^{\mathrm{LBM}}$. As explained in section 1.3.5 (and already applied to the analysis of the initial condition), to do this we insert the coefficients (4.13) (which cancel the leading orders of the LBM-residue) into the expression of the residues of the moving boundary LBM (4.11).

Remark. Once we have justified the use of the prediction (4.12), we can again conclude that the lattice Boltzmann solution $\hat{f}$ approximates the solution of Navier-Stokes, as the interior solution does. In section 1.3.4, using (4.12), we had

$$h^{-1}\hat{\mathbf{u}} = h^{-1}\sum_i \mathbf{c}_i \hat{f}_i = \mathbf{u} + O(h^2), \quad \hat{p} = c_s^2 \frac{\sum_i \hat{f}_i - 1}{h^2} = p + O(h). \tag{4.14}$$

Boundary conditions

The implementation of the boundary conditions of problem (4.1), within a kinetic scheme, suffers from the same difficulties encountered with the initialization. The LBM involves a set of variables which is typically larger than the number of conditions provided by the hydrodynamical problem. Therefore, the boundary conditions of the original problem are not enough to fix all degrees of freedom of the kinetic scheme. The additional constraints have to be defined carefully, in order to avoid the appearance of wrong behavior on the hydrodynamical level.
The topic is not treated in detail here. Description of reliable choices can be found in [4, 12, 15, 16, 22, 35]. An overview of different approaches, analyzed with the asymptotic expansion technique has been presented in [20]. We will only summarize shortly the asymptotic analysis and the main results, regarding the method proposed in [4], which we use in the numerical simulations.

BFL rule Figure 4.2 (left) exemplifies the neighborhood of a boundary node $\mathbf{x_k}$. We have to define the population

$$\hat{f}_i(n+1, \mathbf{k}),$$

for the velocity $\mathbf{c}_i$ which are incoming directions at a node $\mathbf{k} \in I_b(n)$, according to the velocity $\mathbf{u}_B$ prescribed on boundary.
The BFL-rule[a] interpolates the incoming population at the node $\mathbf{x_k}$, using the values of the neighboring nodes after collision (see figure 4.2), and adding an amount of momentum to achieve the required Dirichlet condition on velocity.
If i^* indexes the link such that $\mathbf{c}_{i^*} = -\mathbf{c}_i$, then $(\mathbf{k}, i^*) \in B^n$. The algorithm reads

$$\hat{f}_i(n+1, \mathbf{k}) = C_1(q(n))\hat{f}^c_{i^*}(n, \mathbf{k}) + C_2(q(n))f^c_{i^*}(n, \mathbf{k} + \mathbf{c}_i) + C_3(q(n))f^c_i(n, \mathbf{k}) + \\ + 2D(q(n))f^*_i c_s^{-2} h \mathbf{u}_B(\mathbf{b}^n_{i^*}(\mathbf{k})) \cdot \mathbf{c}_i, \tag{4.15}$$

where f^c_i is the population after collision, $q \in [0, 1)$ is the distance between $\mathbf{x_k}$ and the boundary, along the link $\mathbf{c}_i$ (in lattice units) and $\mathbf{b}^n_{i^*}(\mathbf{k})$ is the point belonging to Γ defined in (4.7).

[a] From *M.Bouzidi, M.Firdaouss, P.Lallemand*, ref. [4].

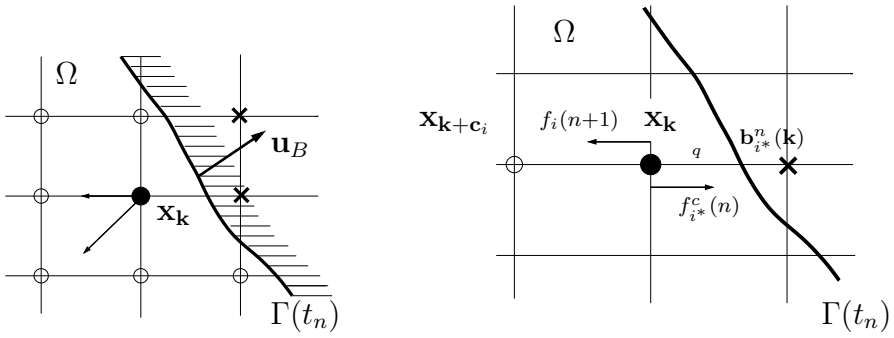


Figure 4.2: **Left:** The point $\mathbf{x_k}$ is a boundary node at the time t_n, the directions $i^* = 1$, $i^* = 2$ are *outgoing*. Therefore, the corresponding incoming $i = 5$, $i = 6$ have to be defined, in accordance with the Dirichlet condition $\mathbf{u}_B$. **Right:** Sketch of the BFL boundary conditions. The bold line is the boundary, i is an *incoming* direction at the boundary node $\mathbf{x_k}$ (circles), i^* is the related outgoing direction connecting $\mathbf{x_k}$ to a solid node (cross). q is the distance (in lattice units) between $\mathbf{x_k}$ and the point $\mathbf{b}^n_i(\mathbf{k}) \in \Gamma$ (see equation (4.7)).

The interpolation coefficients (time depending through q) have the expressions

$$C_1(q) = \begin{cases} 2q & q < \frac{1}{2} \\ \frac{1}{2q} & q \geq \frac{1}{2} \end{cases}, \; C_2(q) = \begin{cases} (1-2q) & q < \frac{1}{2} \\ 0 & q \geq \frac{1}{2} \end{cases}, \; C_3(q) = \begin{cases} 0 & q < \frac{1}{2} \\ \frac{2q-1}{2q} & q \geq \frac{1}{2} \end{cases} \tag{4.16}$$

while

$$D(q) = \begin{cases} 1 & q < \frac{1}{2} \\ \frac{1}{2q} & q \geq \frac{1}{2} \end{cases}. \tag{4.17}$$

For $q = \frac{1}{2}$, equation (4.15) reduces to the well known *bounce-back* rule (see also section 3.3.2),

$$\hat{f}_i(n+1, \mathbf{k}) = \hat{f}^c_{i^*}(n, \mathbf{k}) + 2h f^*_i c_s^{-2} \mathbf{u}_B(\mathbf{b}^n_{i^*}(\mathbf{k})) \cdot \mathbf{c}_i. \tag{4.18}$$

Asymptotic analysis First, we compute the residue for a regular expansion in the ansatz $\mathcal{A}^{\mathrm{LBM}}$ inserting the prediction (4.12) into the algorithm (4.15). The different orders are sorted, using a Taylor expansion around the point[b] $(t_n, \mathbf{b}^n_{i^*}(\mathbf{k})) \in \Gamma(t_n)$ and the relations

$$\begin{aligned} \mathbf{x_k} &= \mathbf{b}^n_{i^*}(\mathbf{k}) + hq(n)\mathbf{c}_i \\ \mathbf{x_{k+c}}_i &= \mathbf{b}^n_{i^*}(\mathbf{k}) + h(1+q(n))\mathbf{c}_i \,. \end{aligned} \tag{4.20}$$

[b]To analyze the boundary conditions we have to choose a sequence of the center $(\bar{t}_h, \bar{\mathbf{x}}_h)$ of the Taylor expansion such that

$$\forall h \in H : \; \bar{\mathbf{x}}_h \in \Gamma(\bar{t}_h). \tag{4.19}$$

In the resulting equation, we insert the expression

$$f_i^{(1)} = c_s^{-2} f_i^* \mathbf{c}_i \cdot \mathbf{u},$$

coming from the interior algorithm. We find the following condition (abbreviating $\mathbf{b} = \mathbf{b}_{i^*}^n(\mathbf{k})$)

$$(1 + C_1 + C_2 - C_3)\, \mathbf{c}_{i^*} \cdot \mathbf{u}(t_n, \mathbf{b}) = 2D\, \mathbf{c}_{i^*} \cdot \mathbf{u}_B(t_n, \mathbf{b}). \tag{4.21}$$

Since

$$\forall q \in [0,1) :\ 1 + C_1(q) + C_2(q) - C_3(q) = 2D(q), \tag{4.22}$$

equation (4.21) reduces to

$$\mathbf{c}_{i^*} \cdot \mathbf{u}(t_n, \mathbf{b}) = \mathbf{c}_{i^*} \cdot \mathbf{u}_B(\mathbf{b}). \tag{4.23}$$

In substance, it is reasonable to add a constraint on the field $\mathbf{u}$, and modify the ansatz (equation (1.155), for the interior LBM and the initialization rule) as

$$\mathcal{A}^{(1)} = \{F \in \mathcal{A}^{\mathrm{LBM}} \mid f_i^{(1)}(F_h) = f_i^* c_s^{-2} \mathbf{c}_i \cdot \mathbf{u}(F_h), \nabla \cdot \mathbf{u} = 0, \\ \mathbf{u}(0, \mathbf{x_j}) = \mathbf{u}_0(\mathbf{x_j}),\ \mathbf{u}(t_n, \mathbf{b}) = \mathbf{u}_B(\mathbf{b})\}. \tag{4.24}$$

It is a restriction reflecting Dirichlet boundary condition in problem (4.1). It means, that the velocity has to be a solution to Navier-Stokes, satisfying the proper initial and boundary condition.
In the second order, an analogous procedure leads to [20]

$$\mathbf{c}_{i^*} \cdot \mathbf{u}^{(2)}(F) = (-q + C_1(1-q) - qC_2 + C_3(1+q))\, \mathbf{c}_{i^*} \cdot \nabla \mathbf{u} \cdot \mathbf{c}_{i^*}, \tag{4.25}$$

where

$$\mathbf{u}^{(2)}(F) = \sum_i \mathbf{c}_i f_i^{(2)}(F). \tag{4.26}$$

Since

$$\forall q \in [0,1) :\ -q + C_1(1-q) - qC_2 + C_3(1+q) = 0, \tag{4.27}$$

it follows

$$\mathbf{u}^{(2)}(F)(t_n, \mathbf{b}) = 0, \tag{4.28}$$

i.e., we find a boundary condition consistent with the solution

$$\mathbf{u}^{(2)} = 0$$

for the interior domain[c]. In virtue of equations (4.23), (4.28), we can select the expansion up to the second order with coefficients $f^{\mathrm{LBM},(k)}$. The scheme preserves the predicted second order in velocity and first order in pressure[d].

[c] Trying to do the same for the $k = 3$, we find a relation which cannot be satisfied within both by the interior and the boundary algorithm. Theoretically, the regular ansatz could be modified [20] to predict also the irregular behaviors in higher orders.

[d] As remarked at the beginning (section 1.2.1), it does not assure

$$|\hat{F}^{BFL} - \hat{f}| = O(h^3),$$

Local boundary conditions The boundary rule (4.15) can be implemented in a simple way, but not in general situations. Namely, it requires at least one additional fluid node in the incoming direction. Boundaries with special curvatures, or corners (figure 4.3), are not covered by this approach.

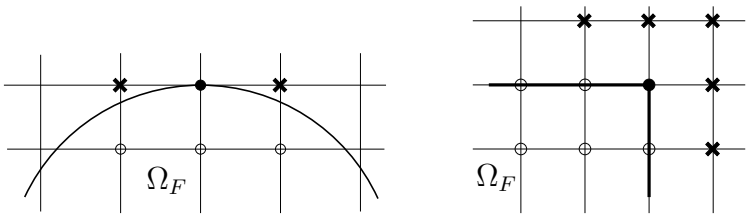


Figure 4.3: Two situations in which BFL cannot be applied, in the node indicated with •. **Left:** The fluid domain is inside a circle, the upper node, considered belonging to the fluid, does not have horizontal fluid neighbors, for any $h > 0$. **Right:** The critical node is located in a corner, no diagonal neighbors are available.

In practice, the BFL rule might be replaced, in such special points, by the simpler bounce-back rule (4.18). However, this leads to a loss in accuracy. A possible solution has been presented in [22]. It is based on the principle of lemma 1.10 (*self-compatibility*), directly approximating the coefficient given for the interior LBM. Through a linear combinations of the different $f_m(n, \mathbf{k})$, for $\mathbf{c}_m \in \mathbb{V}_I(n, \mathbf{k})$, an approximation of the *predicted* second order coefficient $f_i^{\tilde{LBM},(2)}$ is constructed, defining the new population as the sum

$$\begin{aligned} \hat{f}_i(n+1, \mathbf{k}) &= f_i^{LBM,(0)} + h f_i^{LBM,(1)} + h^2 f_i^{\tilde{LBM},(2)} = \\ &= f_i^{LBM,(0)} + h f_i^{LBM,(1)} + h^2 f_i^{LBM,(2)} + O(h). \end{aligned} \tag{4.29}$$

Periodic pressure drop In some simulations, problems involving pressure boundary conditions can be considered. The following algorithm [9] can deal with a directional pressure drop applied on a periodic box.
Let us assume, for simplicity, to have $\Omega = [0, 1)^2$ and the condition

$$p(t, \mathbf{x})|_{x=0} - p(t, \mathbf{x})|_{x=1} = \Delta P(t), \tag{4.30}$$

besides the periodicity in all directions.
The idea is to re-inject the distributions exiting from the side $x = 1$ after having modified the coefficients in order to achieve the difference Δp. Briefly, the

for any $t > 0$, since the BFL-rule may suffer from instability. In particular, algorithm (4.15) does not conserve the mass, in contrast to the *bounce-back* condition (4.18), whose stability has been proven [23, 26]. On the other hand, the bounce-back rule is not able to reach the same accuracy.

Periodic Pressure Drop condition is implemented as follows:

$$\begin{aligned}
&\texttt{DO } j_y = 1, 2, \ldots, N_y \\
&\quad \texttt{periodic BC: } f_i(n+1,(1,j_y)) = f_i(n,(N,j_y)) \\
&\quad \texttt{evaluate } \overline{\Delta P} = c_s^2 \left[\rho(n+1,(1,j_y)) - \rho(n,(N,j_y))\right] \\
&\quad \texttt{impose } \Delta P\texttt{: } f_i(n+1,(1,j_y)) = f_i(n+1,(1,j_y)) + f_i^* c_s^{-2} h^2 \left[\Delta P - \overline{\Delta P}\right] \\
&\texttt{END}
\end{aligned} \tag{4.31}$$

The periodicity assures mass conservation during the simulation, since the quantities added/subtracted on both the sides are balanced.

4.2 Refill methods

Figure 4.4 illustrates an example of a node which has become part of the fluid region at a certain time step. In that point, the LBM populations have to be initialized. We want to approach the problem from the point of view of asymptotic analysis. In view of lemma 1.10, it would be enough to define an algorithm by *approximating* the coefficients $f^{LBM,(k)}$ in the different orders. However, a direct approximation can be difficult, since it contains unknown functions (for example, the Navier-Stokes solution). Therefore we will rather consider *first-attempt* algorithms, perform the analysis in the usual way and look at the conditions on the coefficients and for the corrections we need to achieve the desired prediction (see section 1.3.5).

The starting point are two algorithms, proposed in [31], which will be described, analyzed and improved in this section.

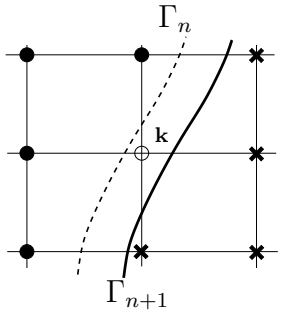


Figure 4.4: The boundary has moved from the time t_n (dashed line) to t_{n+1} (solid line), over the node $\mathbf{x_k}$. Populations $\hat{f}_i(n+1,\mathbf{k})$ need to be defined.

4.2.1 Benchmarks

We define simple benchmarks, which allow to *easily construct analytical solutions* and to *better focus on the problem* we are interested in. Once the algorithms have been fixed, we will consider more general test cases (section 5.5).

Plane Moving Walls

First, we set $\Omega = \mathbb{R}^2$, and choose

$$\Omega_F(t) = [a_L(t), a_R(t)] \times \mathbb{R}, \tag{4.32}$$

where a_L and a_R are given function, in the form

$$a_L(t) = l_0 + \Phi_W(t), \quad a_R(t) = r_0 + \Phi_W(t). \tag{4.33}$$

The model describes the space between two plane infinite walls, parallel to the y-axis and moving horizontally with the same *given* velocity,

$$\dot{a_i}(t) = \dot{\Phi}_W(t) = \varphi_W(t), \quad i = L, R \tag{4.34}$$

with incompressible flow in between. We call this benchmark the *Moving Walls* (MW) problem (figure 4.5).

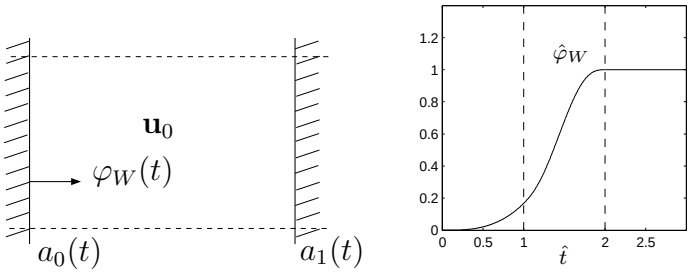


Figure 4.5: **Left:** Sketch of the *Moving Walls* model.The distance $a_R(t) - a_L(t)$ is fixed in time, both the walls move with the same velocity. **Right:** Wall's speed $\hat{\varphi}_W(\hat{t})$ (equation (4.36)), as a function of (non dimensional) time. To avoid initial layers in leading orders, we use a *smooth start*, approaching a constant velocity with a C^2-profile.

Smooth start The idea of the benchmark is to simulate a steady constant horizontal movement of the walls. We initialize the simulations using *equilibrium initial conditions*

$$\hat{f}_i(0, \mathbf{j}) = H_i^{eq}(1, h\mathbf{u}_0(\mathbf{x_j})).$$

As discussed in chapter 1, such choice is reliable only for particular initial conditions of the flow. Therefore, we start with a rest configuration, moving the walls according to a function φ_W such that

$$\begin{aligned} &\varphi_W(0) = 0, \\ &\varphi_W(t) = 1, \forall t \geq T_1, \\ &\varphi_W(t) \in C^2, \end{aligned} \tag{4.35}$$

for a given $T_1 > 0$. Explicitly, we use $\varphi_W(t) = \hat{\varphi}_W(\hat{t})$, with $t = \frac{T_1}{2}\hat{t}$ and

$$\hat{\varphi}_W(\hat{t}) = \begin{cases} \frac{1}{6}\hat{t}^3 & \text{for } \hat{t} \in [0,1] \\ 3\hat{t}^5 - 22\hat{t}^4 + \frac{371}{6}\hat{t}^3 - 83\hat{t}^2 + 54\hat{t} - \frac{41}{3} & \text{for } \hat{t} \in [1,2] \\ 1 & \text{for } \hat{t} \geq 2. \end{cases} \tag{4.36}$$

The C^2-regularity is needed to avoid troubles like boundary and initial layers in the leading orders.

We consider two different cases. In the *Simple Moving Walls* (SMW) model, the boundaries start to move with the fluid at rest. Due to incompressibility, in the exact solutions the fluid and the walls move together as a rigid structure:

$$u_{\text{SMW}}(t,x) = \varphi_W(t), \quad v_{\text{SMW}} = 0 \tag{4.37}$$

The pressure can be chosen up to a time depending function. The field

$$p_{\text{SMW}}(t,x) = -\dot{\varphi}_W(t)\left(x - \frac{a_L(t) + a_R(t)}{2}\right). \tag{4.38}$$

satisfies Navier-Stokes plus the *zero-average* condition[e]

$$\forall t \geq 0 : \ \langle p_{\text{SMW}} \rangle(t) = \int_{\Omega_F(t)} p_{\text{SMW}}(t, \mathbf{x}) d\mathbf{x} = 0. \tag{4.39}$$

All the quantities are only function of x.
A variant of this problem is obtained forcing a non trivial flow inside the walls. Considering the 2π-Taylor vortex solution (see section 1.3.4),

$$\mathbf{u}_{TV}(t,x,y) = \begin{pmatrix} -\frac{1}{2\pi}\cos(2\pi x)\sin(2\pi y) \\ \frac{1}{2\pi}\sin(2\pi x)\cos(2\pi y) \end{pmatrix} \exp\left(-8\pi^2\nu t\right) \tag{4.40}$$

$$p_{TV}(t,x,y) = -\frac{1}{16\pi^2}\left(\cos(4\pi x) + \cos(4\pi y)\right)\exp\left(-16\pi^2\nu t\right), \tag{4.41}$$

[e]Such choice is justified by the initial condition, $\rho = 1$. In view of equation (1.168), used to extract the pressure, at the initial time we have $\langle p \rangle(0) = 0$.

in the *Vortex Moving Walls* (VMW) benchmark, we add, to the problem (4.1), the boundary conditions

$$\mathbf{u}(t, a_i(t)) = \varphi_W(t) + \mathbf{u}_{TV}(t, a_i(t)), \quad i = L, R \tag{4.42}$$

and the volume force

$$\mathbf{G}_{VMW}(t, \mathbf{x}) = \varphi_W(t) \nabla \mathbf{u}_{TV}(t, \mathbf{x}). \tag{4.43}$$

Analytically, we have the solutions

$$u_{VMW}(t, \mathbf{x}) = \varphi_W(t) + u_{TV}(t, \mathbf{x}),\ v_{VMW} = v_{TV}(t, \mathbf{x}) \tag{4.44}$$

$$p_{VMW}(t, s) = p_{TV}(t, \mathbf{x}) + p_{\mathrm{SMW}}(t, x). \tag{4.45}$$

Forced renormalization of pressure A pressure which solves Navier-Stokes is fixed up to a function of time. Since we have chosen the zero average solution (4.39), this condition has to be forced in the algorithm, too. Therefore, after each LB-time step the populations are renormalized:

$$\hat{f}_i(n, \mathbf{j}) = \hat{f}_i(n, \mathbf{j}) - \frac{1}{|I_F(n)|} \left(\sum_{\mathbf{j} \in I_F(n)} \hat{f}_i(n, \mathbf{j}) - 1 \right) \tag{4.46}$$

($|I_F(n)|$ being the number of generalized fluid nodes), forcing a zero average pressure.

Handmade refill process

To isolate the refill step within an LB algorithm, another particular benchmark is used, retracing the benchmark CURVE of section 2.3.1 (*pointwise arise of error*). We consider the curve

$$\Gamma = \{\mathbf{x} \in \Omega \mid \|\mathbf{x} - \mathbf{x}_C\| = R\} = \{\mathbf{x}_C + R(\cos\gamma, \sin\gamma),\ \gamma \in [0, 2\pi)\} \tag{4.47}$$

as the interface between a fluid domain and a solid disk of radius R centered at $\mathbf{x}_C$.
However, instead of simulating a moving boundary problem, we perform *virtual*, prearranged refills, without moving the interface. More precisely, in correspondence with the set

$$I_b(\Gamma) = \{\mathbf{k}_m \mid m = 1, \ldots N_\Gamma\} \tag{4.48}$$

of the boundary nodes numbered from 1 to N_Γ, we define a set of time steps

$$T(\Gamma) = \left\{ n_m := m \frac{h^{-2}}{N_\Gamma} \mid m = 1, \ldots N_\Gamma \right\}. \tag{4.49}$$

The benchmarks proceeds as follows: at time step n_m the values $\hat{f}_i(n_m, \mathbf{k}_m)$ are erased, trying the reconstruction as it would be a refill step. At time $T = 1$, each node in $I_b(\Gamma)$ will have been processed once. Unlike the problem of section 2.3.1, we now include also zero velocity boundary conditions on Γ.

4.2.2 Equilibrium refill (EQ)

This approach [31] reconstructs the populations on the new node using the equilibrium distribution for approximate density and velocity.
Calling $\mathbf{k}$ the node to be refilled at time step $n+1$ (see figure 4.4), first we compute approximations of density and velocity on the new node:

$$\tilde{\rho}_{n+1,\mathbf{k}} = \mathcal{F}^{\rho}(\hat{\rho}, \mathbf{k}), \quad \tilde{\mathbf{u}}_{n+1,\mathbf{k}} = \mathcal{F}^{\mathbf{u}}(\hat{\mathbf{u}}, \mathbf{k}), \tag{4.50}$$

through some extrapolation rules $\mathcal{F}^{\rho}$, $\mathcal{F}^{\mathbf{u}}$, setting then

$$\hat{f}_i(n+1, \mathbf{k}) := H_i^{eq}(\tilde{\rho}_{n+1,\mathbf{k}}, \tilde{\mathbf{u}}_{n+1,\mathbf{k}}). \tag{4.51}$$

In other words, the **equilibrium refill** reads

$$REF_i^{EQ}(h, \hat{f}_h)(n+1, \mathbf{k}) := \hat{f}_i(n+1, \mathbf{k}) - H_i^{eq}(\tilde{\rho}_{n+1,\mathbf{k}}, \tilde{\mathbf{u}}_{n+1,\mathbf{k}}). \tag{4.52}$$

The choice of the extrapolation might depend on the particular flow and motion of the boundary. For example, if the boundary is moving parallel to the x-axis (figure 4.4, left), a three-points backward approximation (along the x-axis) can be used. The density can be extrapolated looking at the last three nodes,

$$\tilde{\rho}_{n+1,k} = 3\hat{\rho}(n+1, \mathbf{k} - \mathbf{c}_1) - 3\hat{\rho}(n+1, \mathbf{k} - 2\mathbf{c}_1) + \hat{\rho}(n+1, \mathbf{k} - 3\mathbf{c}_1), \tag{4.53}$$

and the velocity interpolated between the boundary and the previous two nodes:

$$\tilde{\mathbf{u}}_{n+1,k} = 2\frac{h\mathbf{u}_B(t_{n+1}, \mathbf{b}_1^n)}{q^2+3q+2} + 2q\frac{\hat{\mathbf{u}}(n+1, \mathbf{k} - \mathbf{c}_1)}{q+1} - 2q\frac{\hat{\mathbf{u}}(n+1, \mathbf{k} - 2\mathbf{c}_1)}{q+2} \tag{4.54}$$

(the velocity $\mathbf{u}_B$ has to be rescaled by h, since it is expressed in physical unities), where

$$q = \frac{1}{h}\left(\mathbf{b}_1^{n+1}(\mathbf{k}) - \mathbf{x}_{\mathbf{k}}\right)$$

is the distance from the boundary (at time t_{n+1}) in lattice units. According to the choice of the direction $-\mathbf{c}_1$ for the interpolation, we have used the boundary point $\mathbf{b}_1^{n+1}(\mathbf{k})$ in the previous formulas.

Numerical tests In figure 4.6, we compare the maximum overall errors for pressure and velocity for the moving walls model. The *Equilibrium refill* shows unsatisfactory results for pressure.
Although in the SMW the results are still first order accurate, in the VMW the pressure is inconsistent, since the error does not decrease refining the discretization.
Regarding the velocity (figure 4.7), we achieve only first order accuracy in the problem VMW.

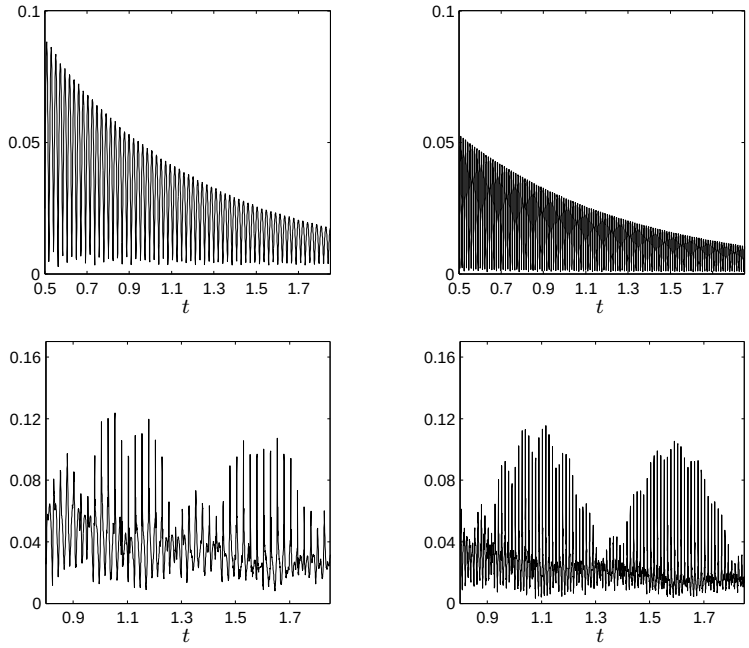


Figure 4.6: Maximum errors in pressure $E^p(t)$. On the **top**, the problems SMW, for which the results on the grids 40×40 (left side) and 80×80 (right side) are compared. On the **bottom**, the more general VMW.

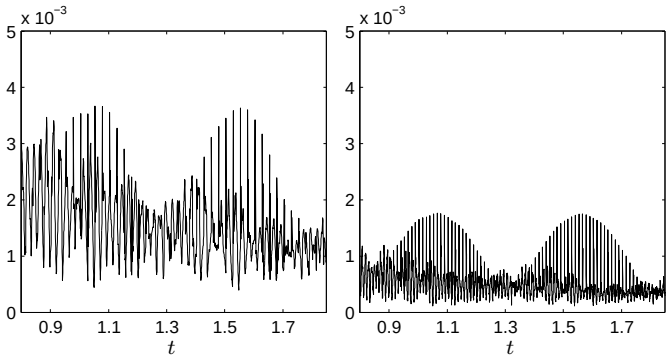


Figure 4.7: Maximum error in velocity simulating the problem VMW. Two grids are compared, 40 on the left, 80×80 on the right. The error decreases like h.

Asymptotic analysis We insert the interior coefficients (4.13) into the algorithm (4.52), checking the resulting conditions to cancel the residue.
Since we are using a second order interpolation, we assume that (4.14) holds for the leading orders of $\tilde{\rho}$ and $\tilde{\mathbf{u}}$. Therefore, $\tilde{\rho}$ and $\tilde{\mathbf{u}}$ approximate the hydrodynamics on the boundary:

$$\tilde{\rho} = 1 + h^2 c_s^{-2} p^{(2)} + O(h^3), \quad \tilde{\mathbf{u}} = h\mathbf{u}_B + O(h^3)$$

(evaluated in $(t_n, \mathbf{b}_i^n(\mathbf{k}))$). We can conclude that a condition arising from the equilibrium refill is

$$f_i^{(k)}(F_h) = f_i^{eq,(k)}(F_h), \quad \text{for } k = 0, 1, 2. \tag{4.55}$$

Compared to the interior LBM, the *non-equilibrium* is missing. For a flow such that

$$f^{\mathrm{LBM},(2)} = f^{eq,(2)}$$

(such as the SMW), the size of the minimum precision class remains small enough ($o(h^2)$) also including the refill step. Hence, we can use the truncated expansion up to the second order, which predicts first order pressure.
Otherwise, we can restrict to

$$\hat{F} = f^{(0)} + h f^{(1)},$$

which does not contains information about the pressure, but is enough to predict a first order accurate velocity. This expectation is confirmed by the result shown in figure 4.7.
Actually, the refill step produces analogous phenomena as those described in chapter 2, where wrong initial conditions for the second order coefficient have been analyzed. In particular, we refer to the theoretical investigation and to the numerics performed in section 2.3.1 (*lack of non-equilibrium*). We have shown how the accuracy in pressure and velocity could be spoiled, even if they are correctly approximated (as for algorithm (4.52)) in the equilibrium function.

4.2.3 Interpolation + Advection refill (IA)

In view of the previous analysis, refill (4.52) cannot be satisfactory. The following approach, proposed in [31], combines different algorithms, according to the different roles the links play in the LBM. Using the sets $\mathbb{V}_O$ (*outgoing directions*), $\mathbb{V}_I$ (*incoming directions*), and $\tilde{\mathbb{V}}$ of definition 4.1, we initialize

$$\hat{f}_i(n+1, \mathbf{k}) = \begin{cases} \hat{f}_i(n, \mathbf{k} - \mathbf{c}_i), & \mathbf{c}_i \in \mathbb{V}_O(\mathbf{k}, n+1) \quad \text{(advected)} \\ \mathcal{E}(\hat{f}_i, n+1, \mathbf{k}, \mathbf{c}_i) & \mathbf{c}_i \in \mathbb{V}_I(\mathbf{k}, n+1) \quad (i\text{-interpolated}) \\ \mathcal{F}(\hat{f}_i, n+1, \mathbf{k}, \mathbf{u}_B) & \mathbf{c}_i \in \tilde{\mathbb{V}}(\mathbf{k}, n+1) \quad (\mathbf{u}\text{-interpolated}) \end{cases} \tag{4.56}$$

where $\mathcal{E}$ is the second order extrapolation

$$\mathcal{E}(\phi, n, \mathbf{k}, \mathbf{c}_i) = 3\phi(n, \mathbf{k} + \mathbf{c}_i) - 3\phi(n, \mathbf{k} + 2\mathbf{c}_i) + \phi(n, \mathbf{k} + 3\mathbf{c}_i) \tag{4.57}$$

along the direction $\mathbf{c}_i$, and $\mathcal{F}$ is an approximation routine to be chosen according to the problem. For example, we can use an extrapolation along a particular direction $\mathbf{c}_m^{ex}$, relevant for the problem. In the problems SMW or VMW we use a backward extrapolation along the x-axis, taking $\mathbf{c}_m^{ex} = \mathbf{c}_5$, since the walls are moving horizontally[f] (as we did for the equilibrium refill).

Numerical tests Simulating the same benchmarks used to test the equilibrium refill (4.52), we observe now that the troubles in pressure have been solved (figure 4.8). The accuracy in velocity has been improved to second order, and in the pressure to first order in h.

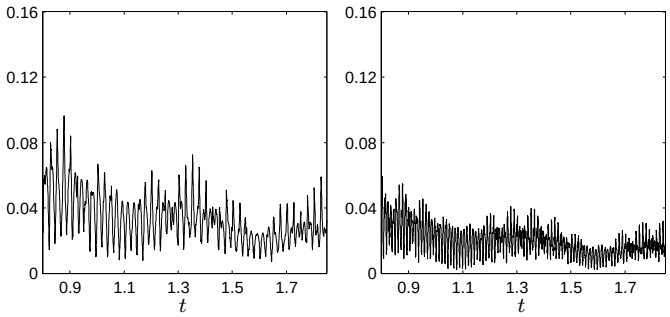


Figure 4.8: Same tests as in figure 4.6, but using the refill (4.56). Errors in pressure $E^p(t)$ for the problems VMW. Results are first order accurate.

Regarding the benchmark CURVE, in chapter 2, simulating the pointwise re-equilibration we have already tested an algorithm analogous to the equilibrium refill. We obtained an inconsistent pressure (figure 2.14). Figure 4.11 shows that the accuracy has been improved, employing the refill (4.56).

Remark. Also in the numerical test performed in [31], algorithm (4.56) seemed to produce qualitatively better results, respect to the equilibrium refill (4.52).

Asymptotic analysis Now, the three different rules in (4.56) coupled with the internal lattice Boltzmann have to be separately analyzed.
Since the *outgoing* directions are updated with the interior LBM, the residue will vanish automatically using the interior coefficients.

[f] In general, different approaches are possible. For example, one could choose to extrapolate along the direction which minimizes the scalar product with the velocity $\mathbf{u}_B$ at a closed boundary point.

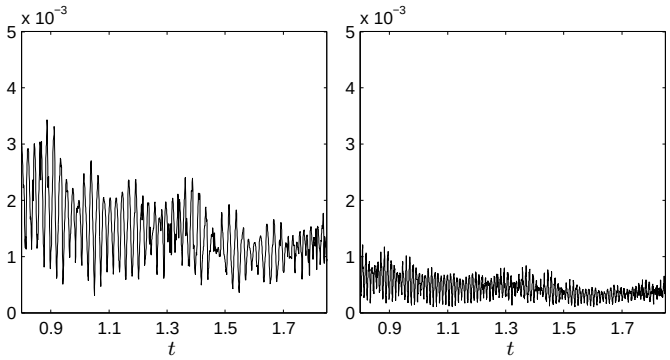


Figure 4.9: Errors in velocity for the refill method (4.56). **Left:** Grid 40×40. **Right:** Grid 80×80.

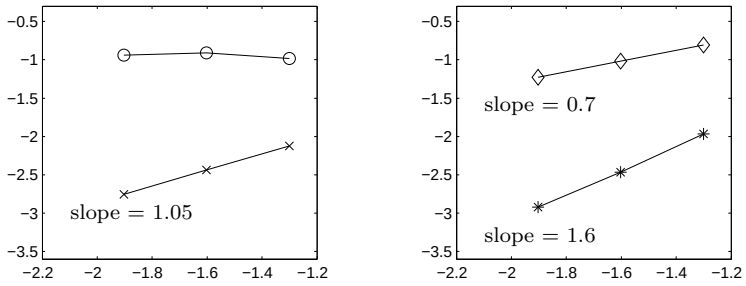


Figure 4.10: Order plot of the velocity and the pressure, testing the refills on different grids. Legend: ($\circ$) : EQ refill, pressure. ($\times$): EQ refill, velocity . ($\diamond$) : IA refill, pressure . ($*$) : IA refill, velocity .

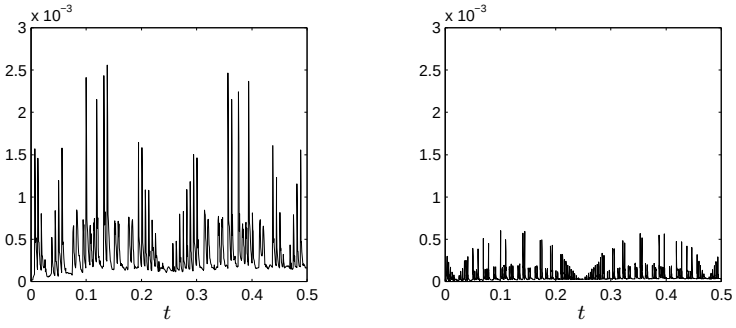


Figure 4.11: Interpolation+Advection refill used in the benchmark CURVE. Maximum error in pressure is shown. **Left:** Grid 40×40. **Right:** Grid 80×80.

Regarding the *incoming* directions, we expand the relations

$$\hat{f}_i(n+1,\mathbf{k}) = 3\hat{f}_i(n+1,\mathbf{k}+\mathbf{c}_i) - 3\hat{f}_i(n+1,\mathbf{k}+2\mathbf{c}_i) + \hat{f}_i(n+1,\mathbf{k}+3\mathbf{c}_i). \quad (4.58)$$

Included in these formula, are the relations for the BC for $\hat{f}_i(n+1,\mathbf{k}+\mathbf{c}_i)$, since $\mathbf{k}+\mathbf{c}_i$ was a boundary node at time t_n. The update rule is actually

$$\begin{aligned}\hat{f}_i(n+1,\mathbf{k}) = 3(C_1(q_n)\hat{f}^c_{i^*}(n,\mathbf{k}+\mathbf{c_i}) + C_2(q_n)\hat{f}^c_{i^*}(n,\mathbf{k}+2\mathbf{c}_i) + C_3(q_n)\hat{f}^c_i(n,\mathbf{k}) + \\ + D_i(q_n,\mathbf{u}_B)) - 3\hat{f}_i(n+1,\mathbf{k}+2\mathbf{c}_i) + \hat{f}_i(n+1,\mathbf{k}+3\mathbf{c}_i), \quad (4.59)\end{aligned}$$

where $q_n = q(n)$ is the distance at the *previous* time step. Obviously, only the case $q_n > \frac{1}{2}$ is considered[g] (before the node $\mathbf{k}$ had entered the fluid region). Inserting a regular expansion for $\hat{f}_i(n+1,\mathbf{k})$ and the interior coefficients on the right hand side of (4.59) (where only *interior* nodes are involved), we find

$$f_i^{(k)} = f_i^{\mathrm{LBM}},\ k = 0,1,2.$$

Regarding the directions $\mathbf{c}_i \in \tilde{\mathbb{V}}$, we insert the interior coefficients into (4.59), since these populations belong to the inner solution. Since the refilled population are constructed using a second order approximation of the solution in the interior domain, the resulting residue is removed up to the second order.
Eventually, we can use a prediction

$$\hat{F}_h = f^{\mathrm{LBM},(0)} + hf^{\mathrm{LBM},(1)} + h^2 f^{\mathrm{LBM},(2)} \quad (4.60)$$

based on the coefficients (4.13), which predicts the same accuracy as the interior LBM and boundary conditions (4.15).
Stability remarks. As observed deriving the asymptotic expansion technique (in section 1.2.1), (4.60) does not assure

$$|\hat{F} - \hat{f}| = O(h^3).$$

In particular, the extrapolation rules used to get the approximations could produce instability. In fact, unlike the equilibrium refill (4.52), the algorithm (4.56) does not interpolate $\mathbf{u}$ and p on the new nodes. It can affect the mass conservation, already spoiled by the boundary condition rule. However, we do not enter into details of a stability analysis.

[g]Even more, the analysis can be reduced to

$$q_n = 1 - O(h).$$

4.3 A compromise: Equilibrium + Non equilibrium refill (EnE)

After the previous analysis and the numerical tests, we can figure out the advantages and the disadvantages of using the two approaches.
The *equilibrium refill* is easily implemented with interpolation **restricted to the hydrodynamics**, but it gives inaccurate results. In particular, we observe a smooth hydrodynamics right after the refill step (due to the interpolation), which becomes highly inaccurate (inconsistent pressure) in the following iterations. It is due to the missing non equilibrium. As we discussed in chapter 2, such error appears in pressure and velocity as a *grid scale* oscillations, after one propagation step, damped in a viscosity dependent number of iterations.
The second approach theoretically solves that problem. The reconstruction involves **the population**, either transported from the inner domain or extrapolated. As a consequence, both equilibrium and non equilibrium are recovered, with enough accuracy. However, the implementation of this algorithm is not local and needs the communication with several neighboring nodes. It can become complicated in a general case.

Idea In view of the previous considerations, the aim is to construct a refill step

- well fitting to LB implementation,
- based on interpolation of p and $\mathbf{u}$,
- able to reconstruct equilibrium and non equilibrium with enough accuracy to keep the precision of the scheme.

According to the analysis, it is enough to *complete* the reconstruction of the populations copying the non equilibrium part from a neighbor of the new fluid node, i.e. choosing a simple first order extrapolation. It has to be mentioned that a similar idea was used in [16], to implement the Dirichlet boundary condition. In practice, we propose the following algorithm:

Algorithm 4.2 (EQ+non EQ (EnE) Refill).
`choose an extrapolation direction` $\mathbf{c}_m^{ex}$ *(incoming)*
`interpolate:`
$\tilde{\rho} = \mathcal{F}^{\rho}(\rho, n+1, \mathbf{k}, \mathbf{c}_m^{ex})$, $\tilde{\mathbf{u}} = \mathcal{F}^{\mathbf{u}}(\mathbf{u}, n+1, \mathbf{k}, \mathbf{c}_m^{ex})$
`use approximated equilibrium`
$\hat{f}_i(n+1, \mathbf{k}) = H_i^{eq}(\tilde{\rho}, \tilde{\mathbf{u}}),\ i = 1, \ldots, b$
`add approximated non equilibrium`
$\hat{f}_i(n+1, \mathbf{k}) = f_i(n+1, \mathbf{k}) + f_i^{neq}(n+1, \mathbf{k} + \mathbf{c}_m^{ex}),\ i = 1, \ldots, b$

Asymptotic Analysis We can recycle the results of the previous analysis. By construction, the new populations contain a second order correct equilibrium part, and a first order correct non-equilibrium part (approximation from a neighboring node). In conclusion, with an algorithm which requires little additional work compared to the EQ-refill, we are able to apply lemma 1.10, i.e. the prediction for the moving boundary algorithm achieves the same accuracy in pressure and velocity as the interior LBM.

4.3.1 Numerical simulations

We use the simple test cases VMW and CURVE, to check the validity of the prediction for the EnE-refill. In figure 4.12, the error in pressure for the problem VMW is shown, comparing two different grids. With the double logarithmic plot, we measured experimentally the accuracy order of pressure and velocity. The results for the error behavior are similar to the *interpolation+advection* refill.

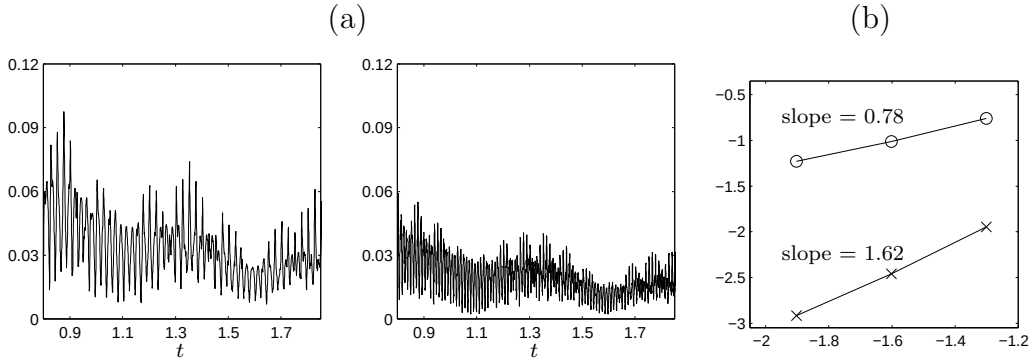


Figure 4.12: Benchmark VMW, using *Equilibrium-non Equilibrium* refills. **(a)** Maximum error in pressure. The grid 40×40 and 80×80 are compared. **(b)** Double logarithmic plot, of the maximum errors (in space and time) in pressure and velocity, with approximate orders of accuracy.

Figure 4.13 shows the results for the benchmark CURVE. The first order accuracy in pressure is achieved.

Summary

We have presented a detailed treatment of moving boundary simulations with LBM. Besides a description of possible existing approaches, a rigorous asymptotic analysis has been applied to understand the problem.
The chapter also shows how the analysis can be used in practice to define new routines. In this particular case we have constructed a new refill algorithm by

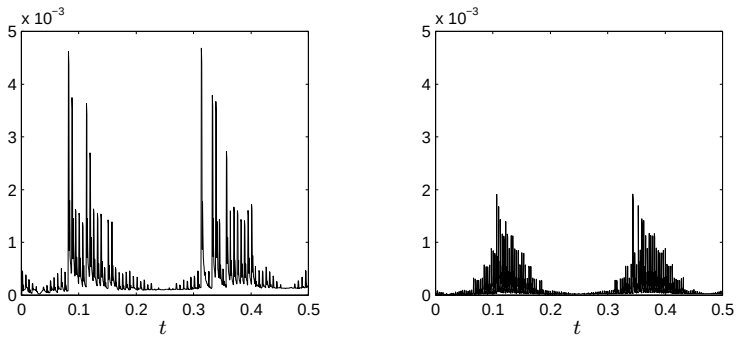


Figure 4.13: Results for the benchmark CURVE, refilling the nodes with the *Equilibrium-non Equilibrium* method. Maximum error in pressure versus time, for the grids 40×40 (**left**) and 80×80 (**right**). In this case, the error decreases faster than h.

approximating the leading order of the prediction. The method preserves the main advantages of the LBM with a simple implementation.

Chapter 5

Fluid structure interaction

Due to the particular suitability of the LBM in modeling flows through complex geometries, the interaction between fluid and structure became an interesting field of application of the method. Such problems, however, require special routines to evaluate the *boundary forces* due to the fluid flow.

The best characteristics of the LBM lie in its *efficiency*. Therefore, any additional algorithm should be able to preserve such a property, in order to keep the LBM numerically competitive. We focus on the *Momentum Exchange algorithm* (MEA, proposed in [28]), which models the fluid-boundary interaction based on simple particle dynamics, and which requires only a low additional computational effort. In fact, this algorithm has been used in numerical simulations [29, 32, 36] but, to our knowledge, has not been investigated theoretically in depth.

In this chapter, besides some comparison between different standard approaches, we present an *asymptotic analysis* of the algorithm. The main result is represented by an accuracy estimate for the MEA, concerning the evaluation of local and global forces. Furthermore, we propose, discuss and test an improved routine to evaluate the local interaction, which achieves first order accuracy.

In section 5.1 we set up the model problem and some benchmarks. Some methods to evaluate the forces, based on the direct extrapolation of the stresses, are shortly presented, tested and commented in section 5.2. The MEA is described in section 5.3, with examples of applications, analysis and comparisons between theoretical and numerical results. The accuracy estimate is stated. In section 5.4 an improved algorithm for the evaluation of the local interaction is proposed, shortly discussed and tested on some benchmark. Finally, section 5.5 summarizes the steps performed so far and presents the numerical results for simple interaction problems.

A shorter version of the theoretical results discussed in this chapter has been presented by the author and M. Junk in [6].

5.1 The flow model

We consider the domain Ω decomposed in a fluid and a solid part, separated by an interface Γ:

$$\Omega = \Omega_F(t) \cup \Gamma(t) \cup \Omega_S(t).$$

At the beginning, we assume that $\Omega_S(t)$ moves with a given velocity along a prescribed path. This will help us to formulate simple test problems for which the exact boundary forces are known. If the state of the system is known at the initial time $t = 0$, the dynamics of the fluid can be described by an initial boundary value incompressible Navier-Stokes problem,

$$\begin{aligned} &\begin{cases} \nabla \cdot \mathbf{u} = 0 \\ \partial_t \mathbf{u} + \nabla p + \mathbf{u} \cdot \nabla \mathbf{u} = \nu \nabla^2 \mathbf{u} + \mathbf{G} \end{cases} \quad t > 0,\ \mathbf{x} \in \Omega_F(t) \\ &\mathbf{u}(t, \mathbf{x}) = \mathbf{u}_B(t, \mathbf{x}),\ \mathbf{x} \in \Gamma(t) \\ &\mathbf{u}(0, \mathbf{x}) = \mathbf{u}_0(\mathbf{x}), \quad \mathbf{x} \in \Omega_F(0). \end{aligned} \tag{5.1}$$

where $\mathbf{u}_B(t, \mathbf{x})$ is the given velocity of the interface point $\mathbf{x} \in \Gamma(t)$ at time t and $\mathbf{G}$ is the volume force acting on the fluid.
We are interested in the *total boundary force*[a] defined according to

$$\mathbf{F}_S(t) = \int_{\Gamma(t)} (-p(t, \mathbf{x})\mathbf{I} + \mathbf{S}^\nu(t, \mathbf{x})) \cdot \mathbf{n}(\mathbf{x}) d\gamma(\mathbf{x}), \tag{5.2}$$

where

$$\mathbf{S}^\nu = \mathbf{S}^\nu[\mathbf{u}] = \nu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T \right) \tag{5.3}$$

is the viscous stress tensor and $\mathbf{n}$ the normal vector to Γ, pointing out of the solid domain. If $\mathbf{t}$ denotes the tangential vector to Γ, which is obtained by following the interface counterclockwise, the **local stresses** are

$$f_\mathbf{t} := (\mathbf{S}^\nu \cdot \mathbf{n}) \cdot \mathbf{t}, \quad f_\mathbf{n} := -p + (\mathbf{S}^\nu \cdot \mathbf{n}) \cdot \mathbf{n}. \tag{5.4}$$

A coordinate $\gamma \in [0, 2\pi)$ to parameterize the interface is introduced (equation (4.10)). To denote a point on Γ we use both the expression $\mathbf{b}$ and $\mathbf{b}(\gamma)$, or indicating its coordinate with $\gamma(\mathbf{b})$.

[a] The total torque acting on the solid,

$$\mathbf{T}(t) = \int_{\Gamma(t)} (\mathbf{x} - \mathbf{x}_{CM}(t)) \times [(-p(t, \mathbf{x})\mathbf{I} + \mathbf{S}^\nu(t, \mathbf{x})) \cdot \mathbf{n}(\mathbf{x})]\ d\gamma(\mathbf{x}),$$

could be considered in the same way.

The motion of the structure

In the present work, we do not focus on a general dynamical of the structure. Specifically, only rigid bodies (disks) are considered, for which Newton's equations

$$\begin{cases} \ddot{\mathbf{x}}_{CM} = \overline{\mathbf{F}}_S = \dfrac{\mathbf{F}_S}{M} \\ \dot{\mathbf{\Omega}} = \overline{\mathbf{T}}_S = \dfrac{\mathbf{T}_S}{I} \end{cases} \tag{5.5}$$

(M and I being the mass and the moment of inertia) are solved with a simple *explicit* integration in time (first order Euler method):

$$\begin{aligned} \hat{\mathbf{u}}_{CM}^{n+1} - \hat{\mathbf{u}}_{CM}^{n} &= h^2 \overline{\mathbf{F}}_S^n \\ \hat{\mathbf{\Omega}}^{n+1} - \hat{\mathbf{\Omega}}^{n} &= h^2 \overline{\mathbf{T}}_S^n \\ \hat{\mathbf{x}}_{CM}^{n+1} - \hat{\mathbf{x}}_{CM}^{n} &= h^2 \left(\hat{\mathbf{u}}_{CM}^{n+1} + \hat{\mathbf{u}}_{CM}^{n} \right). \end{aligned} \tag{5.6}$$

5.1.1 Benchmark: the *cylinder-in-flow*

In order to set up a lattice Boltzmann method including the boundary force evaluation, we consider a disk $\Omega_S(t)$ with radius R as a solid body in the rectangle $\Omega = [0, L] \times [0, 1]$ (see figure 5.1).

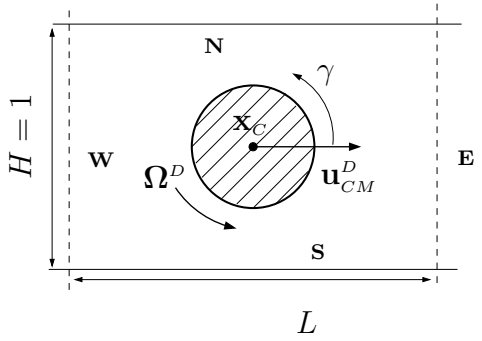


Figure 5.1: The model CiF. The parameter L regulates the ratio between the length and the width of the domain. The solid $\Omega_S(t) = \{\mathbf{x} \in \Omega \mid \|\mathbf{x} - \mathbf{x}_C\| < R\}$ is placed in the center. We refer to the boundary with *North* (N), *South* (S), *East* (E), *West* (W). Furthermore, $\gamma \in [0, 2\pi)$ is the coordinate on the interface.

The benchmark is defined adding to the Navier-Stokes equation the Dirichlet boundary condition on the disk:

$$\mathbf{u}(t, \mathbf{x}) = \mathbf{u}_{CM}^D(t) + \mathbf{\Omega}^D(t) \times (\mathbf{x}_{CM}(t) - \mathbf{x}), \quad t > 0,\ \mathbf{x} \in \Gamma(t), \tag{5.7}$$

$\mathbf{u}_{CM}^D$ and $\mathbf{\Omega}^D$ being the translational and rotational velocity of the disk. The motion of the disk is assumed to be known.

We will consider periodic boundary conditions as well as the motion in a channel, including

$$\begin{aligned} \mathbf{u}(t, x, y = 1) &= \mathbf{u}_N(t),\ t > 0 \\ \mathbf{u}(t, x, 0) &= \mathbf{u}_S(t),\ t > 0 \end{aligned} \tag{5.8}$$

Physically, this situation models a cross section of a flow around a periodic array of long cylinders. We refer to the problem as *cylinder-in-flow* (CiF). To test the algorithm, we use two simple exact solutions of (5.1) around a cylinder of radius $R = 0.2$ in the unit square $\Omega = [0, 1)^2$. We denote these benchmark as CiF_0 and CiF_1.

In the case of CiF_0, we move the disk with constant velocity $\mathbf{u}_0$ in a flow with the same constant velocity $\mathbf{u}(t, \mathbf{x}) = \mathbf{u}_0$, zero pressure, and vanishing body force $\mathbf{G} = 0$. As a consequence, the local stresses vanish and the total boundary force is zero.

In the case CiF_1, we again move the body with a constant velocity $\mathbf{u}_0$ in the constant flow field $\mathbf{u}(t, \mathbf{x}) = \mathbf{u}_0$. However, to obtain a non-trivial local force, we choose a periodic function p_0 and define the body force $\mathbf{G} = \nabla p_0$ which generates a pressure $p(t, \mathbf{x}) = p_0(\mathbf{x})$. For the particular choice

$$p_0(x, y) = \sin(2\pi x)\cos(2\pi y) \tag{5.9}$$

we obtain

$$\begin{aligned} f_{\mathbf{t}}(t, \gamma) &= 0, \\ f_{\mathbf{n}}(t, \gamma) &= \underbrace{-\sin(2\pi(x_C(t) + R\cos\gamma))\cos(2\pi(y_C(t) + R\sin\gamma))}_{-p_0(x(t,\gamma), y(t,\gamma))}, \end{aligned} \tag{5.10}$$

with $\gamma \in [0, 2\pi]$.

In the future, to quickly identify the parameters of a benchmark of type CiF, we will use a notation as in figure 5.2.

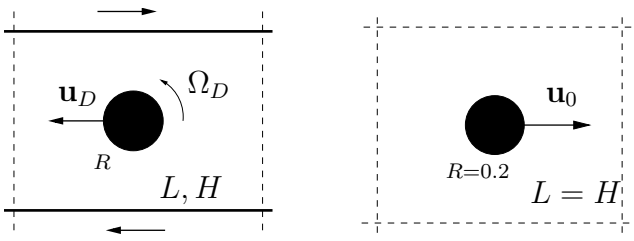


Figure 5.2: We use this simple diagram to refer to a benchmark. Dashed lines represent periodic boundaries, bold lines are solid walls. On the right, the sketch for CiF_0 and CiF_1 is depicted.

5.2 Extrapolation approaches

Before describing in detail the Momentum Exchange algorithm, we present a short overview of other possible approaches to evaluate the interaction. They will serve for future comparison.
In principle the force (5.2), as well as the local interaction (5.4), could be evaluated using a quadrature on the interface. Schematically, using the abbreviations $p(\gamma) = p(\mathbf{b}(\gamma))$, $\mathbf{S}^\nu(\gamma) = \mathbf{S}^\nu(\mathbf{b}(\gamma))$,

$$\hat{\mathbf{F}}_S = \sum_{r=1}^{G} w_i \left(-p(\gamma_i)\mathbf{n}(\gamma_i) + \mathbf{S}^\nu(\gamma_i)\cdot\mathbf{n}(\gamma_i)\right)(\gamma_{i+1} - \gamma_i). \tag{5.11}$$

The previous formula is based on a partition

$$\{\gamma_1, \ldots, \gamma_G\} \subset [0, 2\pi),$$

i.e. a partition

$$P_\Gamma = \{\mathbf{b}(\gamma_r) \mid \gamma_r \in [0, 2\pi),\ r = 1, \ldots, G\} \subset \Gamma$$

of the interface, and uses the values of p and $\mathbf{S}^\nu$ on those points. Unfortunately, approximations of pressure and stress tensor, $\hat{p}$ and $\hat{\mathbf{S}^\nu}$, are only available at the lattice nodes. To have an operational formula within a LBM, (5.11) has to be combined with additional routines, to *extrapolate* the values on the interface.
Several choices are available at this stage. Without too many details, we describe some examples, particularly fitting in the LB-framework.

Boundary Node approximations Within the LBM, we have introduced (equation (4.5)) the set of *boundary couples*

$$B^n := \left\{(\mathbf{k}, i) \in \mathcal{G}(h) \times \{1, \ldots, b\} \mid \mathbf{x}_\mathbf{k} \in \overline{\Omega}_F(t_n),\ \mathbf{x}_{\mathbf{k}+\mathbf{c}_i} \in \Omega_S(t_n)\right\}. \tag{5.12}$$

Each boundary couple $(\mathbf{k}, i) \in B^n$ defines a point $\mathbf{b}_i^n(\mathbf{k}) \in \Gamma$, when the link crosses the interface.
Hence, we can construct the set of the intersections grid-boundary at time t_n

$$\mathcal{B}^n = \overline{\mathcal{G}(h)} \cap \Gamma(t_n) = \{\mathbf{b}_i^n(\mathbf{k}) \mid (\mathbf{k}, i) \in B^n\}$$

(introduced in equation (4.8)).
Identifying each point of Γ with its coordinate in $[0, 2\pi)$, the set $\mathcal{B}^n$ can be ordered

$$\mathcal{B}^n = \{\mathbf{b}(\gamma_r) = \mathbf{b}_{i_r}^n(\mathbf{k}_r), \mid r = 1, \ldots, |B|\}$$

(and consequently B^n). The same set can be seen as a discretization of the interface:

$$P_\Gamma = \mathcal{B}^n.$$

The simplest approximation for pressure and stress tensor, is done using the values at the corresponding **boundary node**:

$$\tilde{p}_{BN}(\mathbf{b}_i^n(\mathbf{k})) = \hat{p}(\mathbf{k}) \tag{5.13}$$

(and similarly for the components of $\mathbf{S}^\nu$).

Lattice-Link extrapolation Otherwise, with the same partition P_Γ, the approximated value can be obtained via a linear (or quadratic) **extrapolation following the link $\mathbf{c}_i$**:

$$\tilde{p}_{LL}(\mathbf{b}_i^n(\mathbf{k})) = (1+q)\hat{p}(\mathbf{k}) - q\hat{p}(\mathbf{k} - \mathbf{c}_i) \tag{5.14}$$

being $q = h^{-1}\|\mathbf{x}_\mathbf{k} - \mathbf{b}_i^n(\mathbf{k})\|$ (in lattice unities). A disadvantage of these approaches, is that the partition is unstructured. Of course, simplified variants are possible, which only consider intersection with horizontal, or vertical, or diagonal links.

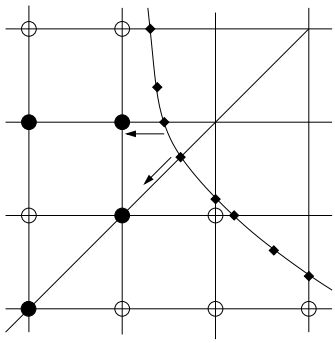


Figure 5.3: Using the partition defined by the intersections grid-interface, the pressure on the boundary point ($\diamond$) can be extrapolated with the nodes ($\bullet$) encountered following, $\mathbf{c}_i$-backward, the intersecting link.

Bilinear extrapolation Differently, we can define a LB-independent discretization $\{\mathbf{b}(\gamma_r)\} \subset \Gamma$, and take the pressure and stress tensor at the **closest lattice node**. Calling

$$\mathbf{k}^{\gamma_r} := \text{ node such that } \quad \|\mathbf{x}_{\mathbf{k}^{\gamma_r}} - \mathbf{b}(\gamma_r)\| = \min_{\mathbf{j}\in I_F(n)} \{\|\mathbf{x}_\mathbf{j} - \mathbf{b}(\gamma_r)\|\}, \tag{5.15}$$

we can use the approximation

$$\tilde{p}(\mathbf{b}(\gamma_r)) = \hat{p}(\mathbf{k}^{\gamma_r}). \tag{5.16}$$

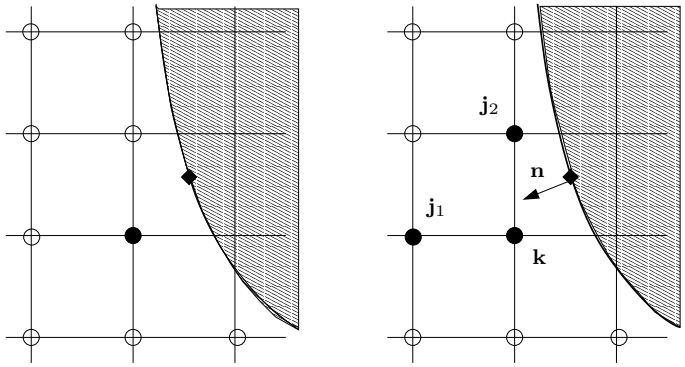


Figure 5.4: Taking the values at the closest lattice nodes, approximations of p and $\mathbf{S}^\nu$ on a boundary point ($\diamond$) can be constructed. **Left:** A simple way, is to take the value on the closest node ($\bullet$). **Right:** More complicated, is a bilinear extrapolation, based on three not aligned nodes ($\bullet$).

A bit more complicated is a **three-points bilinear** extrapolation (an example is drawn in figure 5.4), obtained by searching, for any γ in the partition, three nodes

$$\mathbf{k}^\gamma \in \mathcal{G}(h):\ \|\mathbf{x}_{\mathbf{k}^\gamma} - \mathbf{b}(\gamma)\| = \min_{\mathbf{j}\in I_F(n)} \{\|\mathbf{x}_\mathbf{j} - \mathbf{b}(\gamma)\|\},$$
$$\mathbf{j}^1, \mathbf{j}^2\ : \|\mathbf{k}^\gamma - \mathbf{j}^r\| \text{ is minimum}, \quad \mathbf{k}^\gamma - \mathbf{j}^1,\ \mathbf{k}^\gamma - \mathbf{j}^2 \text{ linearly independent}$$

and by approximating bilinearly

$$\tilde{p}_{BIL}(\mathbf{b}(\gamma)) = \frac{\hat{p}(\mathbf{j}^1) - \hat{p}(\mathbf{k}^\gamma)}{\mathbf{j}^1_x - \mathbf{k}^\gamma_x}\,(x(\gamma) - \mathbf{k}^\gamma_x) + \frac{\hat{p}(\mathbf{j}^2) - \hat{p}(\mathbf{k}^\gamma)}{\mathbf{j}^2_y - \mathbf{k}^\gamma_y}\left(y(\gamma) - \mathbf{k}^\gamma_y\right) + \hat{p}(\mathbf{k}^\gamma) \tag{5.17}$$

(assuming $\mathbf{j}^1_y = \mathbf{k}^\gamma_y$, $\mathbf{j}^2_x = \mathbf{k}^\gamma_x$, as in figure 5.4).

LLFQ algorithm The LLFQ[b] algorithm has been proposed and tested in [32]. The partition P_Γ is pre-defined on the interface, and for each $\mathbf{b}(\gamma_r) \in P_\Gamma$, we proceed as follows. According to the notation in figure 5.5, we consider the lattice node $\mathbf{D}_r$, close to $\mathbf{b}(\gamma_r)$, the point $\mathbf{C}_r$, intersection between the grid and the line $\mathbf{bD}_r$ and the nodes $\mathbf{A}_r$ and $\mathbf{B}_r$, located in opposite sides of $\mathbf{C}_r$. From the populations $\hat{f}_i(\mathbf{A}_r)$, $\hat{f}_i(\mathbf{B}_r)$, approximations for $\hat{f}_i(\mathbf{C}_r)$ are constructed:

$$\tilde{f}_i(\mathbf{C}_r) = \hat{f}_i(\mathbf{A}_r)\overline{\mathbf{B}_r\mathbf{C}_r} + \hat{f}_i(\mathbf{B}_r)\overline{\mathbf{A}_r\mathbf{C}_r},$$

then, using linear extrapolation,

$$\tilde{f}_i^{LLFQ}(\mathbf{b}(\gamma_r)) = \frac{1}{\overline{\mathbf{D}_r\mathbf{C}_r}}\left(\hat{f}_i(\mathbf{D}_r)\overline{\mathbf{b}_r\mathbf{C}_r} - \tilde{f}_i(\mathbf{C}_r)\overline{\mathbf{b}_r\mathbf{D}_r}\right) \tag{5.18}$$

[b]From the authors *H.Li, X.Lu, H.Fang, Y.Qian.*

is defined. From the resulting distributions, $\tilde{p}$ and $\tilde{\mathbf{S}^{\nu}}$ are extracted.

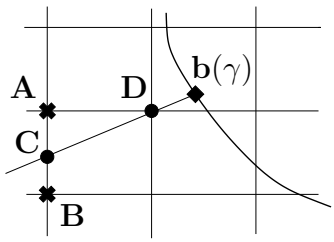


Figure 5.5: The approximation in [32] is constructed by extrapolating the LB population on a pre-selected boundary point $\mathbf{b}(\gamma)$ (diamond) from a close node $\mathbf{D}$ and a point $\mathbf{C}$ (black circles). In turn, the populations on $\mathbf{C}$ have to be interpolated from two nodes $\mathbf{A}$ and $\mathbf{B}$ (crosses). Furthermore, an averaged over the possible choices of $\mathbf{D}$ is considered.

Many choices of the node D_r are possible. In fact, an arithmetic average between more points is performed. For more details, see [32].

Numerical tests

As a preliminary test for the defined algorithms, we perform the computation of the stresses in the problem CiF_1 defined in section 5.1. Only normal stresses, due to the pressure, are present. Figure 5.6 shows a comparison between the different approaches. In this simple case, all of them produce similar and regular results. About the accuracy, assuming to get a first order accurate pressure running the LBM, we cannot expect to go beyond that precision. It is experimentally confirmed in the order plot in figure 5.7.
In figure 5.9, we compare the extrapolations to evaluate the stresses around the cylinder in the problem CiF summarized in the diagram in figure 5.8, for which the analytic solution is unknown.
The LB-grid defined discretization, consisting of a bigger amount of nodes, shows noisy and irregular behaviors. More smooth are the bilinear and LLFQ approximations.

Remarks

In a general situation, the implementation of the previous method can be awkward, especially dealing with *irregular boundaries* and *complex geometries*. In fact, the nodes containing the information used in equations (5.13)-(5.18) have to be searched, case by case, around the boundary point. The resulting routines reduce the efficiency of the LBM. The property of **locality** of operations is lost. Concerning the accuracy, it has to be remarked that the LBM provides only *low order* pressure and stress tensor (first order in h). Therefore, for general problems,

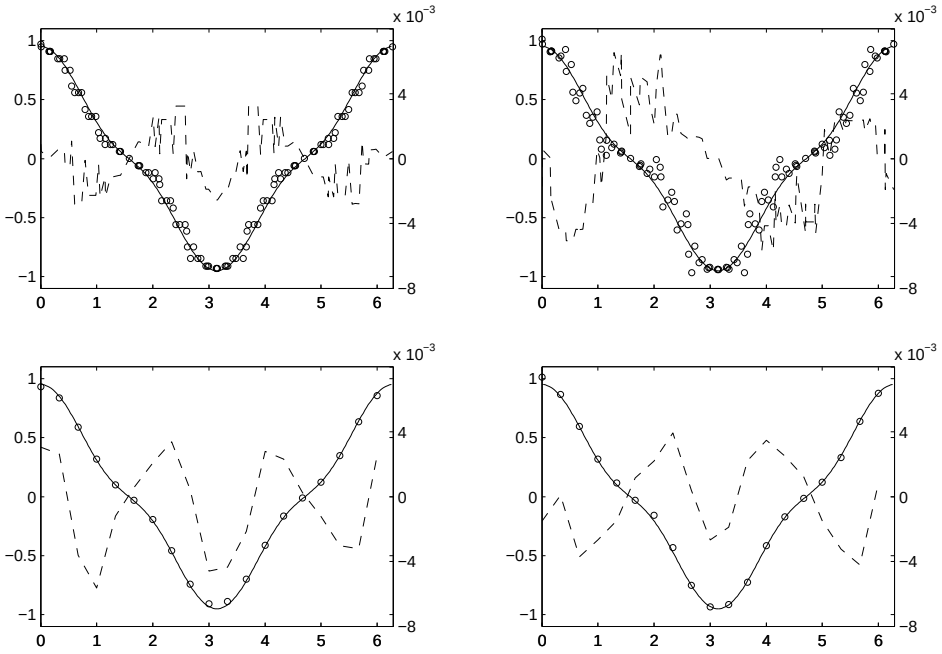


Figure 5.6: The presented extrapolation methods are compared, applied to the problem CiF_1. We sampled with small circles (ordinate on the left y-axis) the pressure, while the exact solution (5.9) is superimposed as a bold line. The dashed line (right y-axis) is the error in the result for the stress $\mathbf{S}^{\nu}_{xx}$ (exact solution $\mathbf{S}^{\nu}_{xx} = 0$). **Top-left:** *boundary node* approximation. **Top-right:** *lattice-link* extrapolation. **Bottom-left:** *bilinear* approximation. **Bottom-right:** LLFQ algorithm.

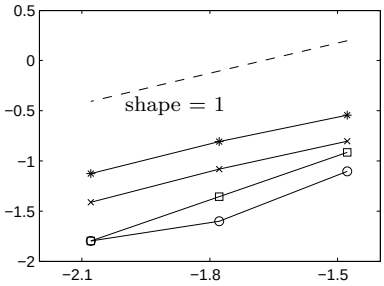


Figure 5.7: Investigation of the pressure accuracy order, for the approximation methods (pressure) described above: *boundary node* ($\times$), *bilinear* ($*$), *lattice-link* ($\circ$), LLFQ ($\square$). The dashed reference lines indicates first order accuracy.

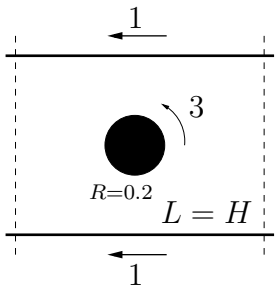


Figure 5.8: A further CiF benchmark. Results of force evaluations are drawn in figure 5.9.

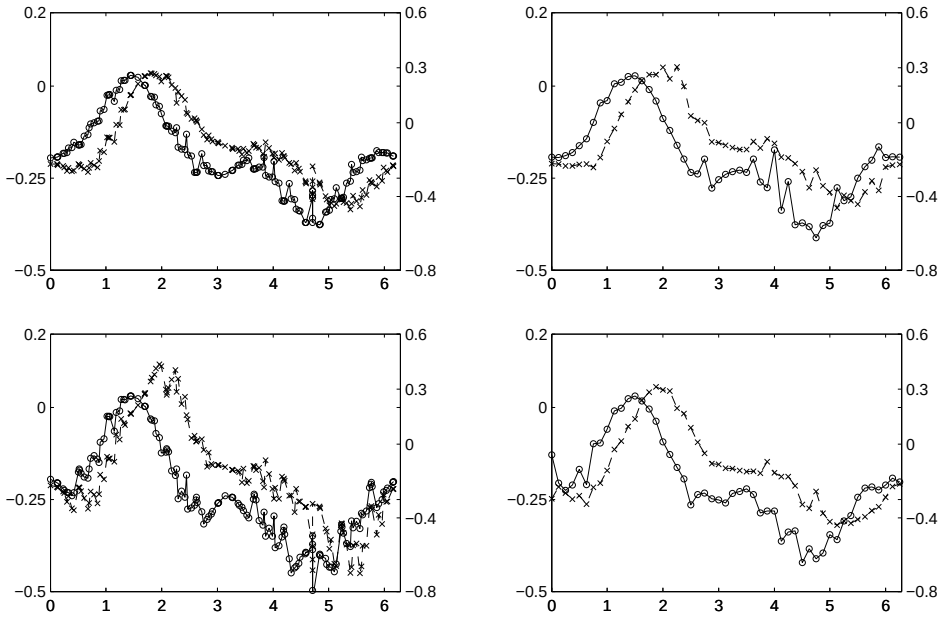


Figure 5.9: As in figure 5.6, the different extrapolation approaches are compared, on the CiF problem, simulating a rotating cylinder (figure 5.8), between two walls moving horizontally. The circles sample the tangential stress $\hat{f}_{\mathbf{t}}$ (y-axis on the left), while the dashed-crosses line represents the normal one $\hat{f}_{\mathbf{n}}$ (y-axis on the right). **Top-left:** *boundary node* approximation. **Top-right:** *lattice-link* extrapolation. **Bottom-left:** *bilinear* approximation. **Bottom-right:** LLFQ algorithm.

increasing the complexity and the order of the approximations (for example going from linear to quadratic, etc.), does not assure any gain in precision.
In view of these considerations, a good solution consists of a rather simple scheme, well-fitting in the lattice framework, which does not require too much exchange of information and, as a consequence, is not supposed to produce high accuracy results.

5.3 Momentum exchange algorithm

We investigate the features of the *Momentum Exchange Algorithm* (MEA), proposed in its original form by A. Ladd in [28], which allows to evaluate the interaction between fluid and boundary using directly the variables of LBM.

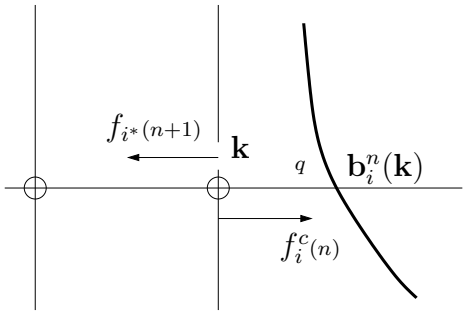


Figure 5.10: Applying the boundary rule on a boundary couple $(\mathbf{k}, i)$, the momentum exchanged in the point $\mathbf{b}_i^n(\mathbf{k})$ is defined as the difference between the incoming $f_{i^*}(n+1, \mathbf{k})$ and the population *after collision* $f_i^c(n, \mathbf{k})$.

The idea is to consider the momentum transferred to the solid from each *boundary fluid node* $\mathbf{x_k}$ interacting with the boundary along a link $\mathbf{c}_i$ (see figure 5.10). The net momentum is given by the sum of momentum due to the particles moving with opposite velocities $\mathbf{c}_i$ and $\mathbf{c}_{i^*}$.

$$\phi_i(n, \mathbf{k}) \equiv \mathbf{c}_i \hat{f}_i^c(n, \mathbf{k}) - \mathbf{c}_{i^*} \hat{f}_{i^*}(n+1, \mathbf{k}) = \mathbf{c}_i \left(\hat{f}_{i^*}(n+1, \mathbf{k}) + \hat{f}_i^c(n, \mathbf{k}) \right). \quad (5.19)$$

(f_i^c is the distribution *after collision*). Denoting with B^n the set of boundary couples (equation (5.12)) the force is approximated by

$$\hat{\mathbf{F}}(n) = \sum_{(\mathbf{k},i) \in B^n} \phi_i(n, \mathbf{k}). \quad (5.20)$$

In practice, the algorithm can be summarized as

Algorithm 5.1 (Momentum Exchange Algorithm).
Given the boundary set $B^n = \{(\mathbf{k}_r, i_r),\ r = 1, \dots |B|\}$
`initialize` $\hat{\mathbf{F}} = 0$
`DO over` B^n
  `LB-collision:` $\rightarrow \hat{f}_i^c(n, \mathbf{k})$
  `boundary condition:` $\rightarrow \hat{f}_{i^*}(n+1, \mathbf{k})$
  `momentum exchanged:` $\phi_i(n, \mathbf{k}) = \mathbf{c}_i \left(\hat{f}_{i^*}(n+1, \mathbf{k}) + \hat{f}_i^c(n, \mathbf{k}) \right)$
  `update:` $\hat{\mathbf{F}} = \hat{\mathbf{F}} + \phi_i(n, \mathbf{k})$
`end`

5.3.1 Numerical tests and asymptotic analysis

Algorithm 5.1 is now tested on the problem CiF_0. In absence of pressure, we compare the results for the local stresses, by evaluating the momentum exchanged *point by point* along the boundary, when the flow and the cylinder are fixed ($\mathbf{u}_0 = 0$), or both moving with the same velocity $\mathbf{u}_0 = (5, 0)$ (figure 5.11). Despite the trivial exact solution $f_{\mathbf{t}} = f_{\mathbf{n}} = 0$, we observe the presence of local forces in relevant orders, *different* in the two cases and *highly irregular*, even if both pressure and velocity are exact in the domain.

Remark: aim of the analysis. Since the boundary forces are represented by an integral over the interface, our primary interest is to investigate the properties of the Momentum Exchange Algorithm intended as *integration routine*. Thus, in order to decouple the errors coming from the approximation of the fluid fields from the one due to the approximate quadrature along the boundary, we use the simple test cases CiF_0 and CiF_1, where the flow is easily solved correctly. In other words, the fields on the boundary are correctly reproduced by the LBM and the arising errors in the force computation are due exclusively to the integration properties of the MEA[c]. Even in the first simple benchmark, when the boundary stresses vanish, we do not obtain correct results.

Also, it has to be remarked that the observed problem appears in the leading orders of the results independently on the size of the velocity $\mathbf{u}_0$. In fact, the relatively high value $\mathbf{u}_0 = (5, 0)$ (in physical units) has been employed only to amplify the error, in order to make the inconsistency better visible.

To investigate the properties of the algorithm, we analyze the scheme

$$\mathrm{MEA}(h, \hat{\mathbf{F}}_h, \hat{\mathbf{f}}_h, \hat{f}_h)(n) = \begin{bmatrix} \hat{\mathbf{F}}_h(n) - \sum_{(\mathbf{k},i)\in B^n} \phi_i(n, \mathbf{k}), \\ \hat{\mathbf{f}}_h(n) - \sum_{(\mathbf{k},i)\in B'} \phi_i(n, \mathbf{k}), \end{bmatrix} \tag{5.21}$$

[c]It must be excluded also the dependence of the results on the choice between single and multiple relaxation time (MRT) collision operator, as well as on the particular LB-model.

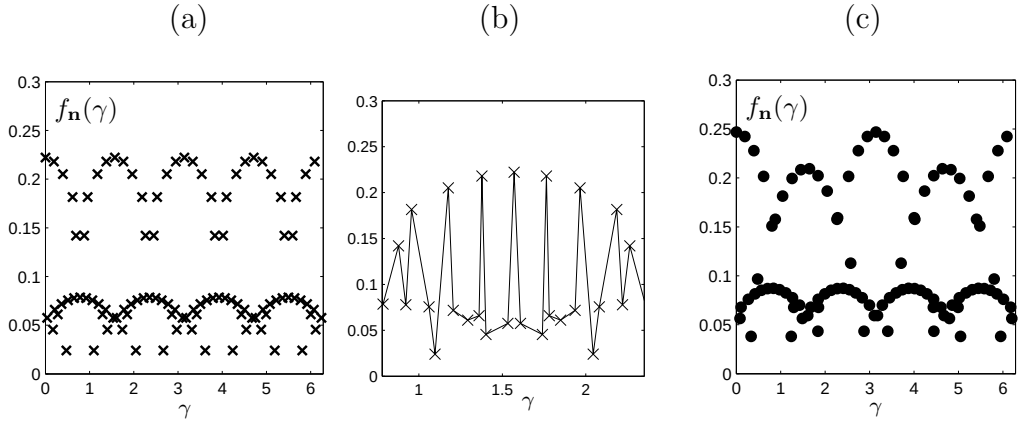


Figure 5.11: Results of the MEA for the normal stress $f_{\mathbf{n}}$, simulating CiF_0 on a 25×25 grid. For each boundary point $\mathbf{b}_i^n(\mathbf{k})$ the value computed with (5.19) at the corresponding node $\mathbf{x}_\mathbf{k}$ is drawn. The exact solution is $f_\mathbf{n} = 0$. **(a)** Flow at rest, $\mathbf{u}_0 = (0, 0)$. The results show strong oscillations. Note that consecutive points ($\times$) are not connected by lines for clarity. The discrete data appear to be $\frac{\pi}{2}$-periodic. **(b)** Zoom on a small part of the boundary of amplitude $\frac{\pi}{2}$ around the north pole $\gamma = \frac{\pi}{2}$ now with connecting lines to demonstrate the oscillation. **(c)** Same model as in (a), but with $\mathbf{u}_0 = (5, 0)$. The MEA breaks the Galilean invariance (in the relevant order).

which computes the approximate *global force* according to (5.20) and the *local forces* using a sum of boundary couples restricted to a proper subset $B' \subset B^n$. The variables $\hat{\mathbf{F}}_h$ and $\hat{\mathbf{f}}_h$ are actually derived from the LB-output $\hat{f}_h$:

$$\hat{\mathbf{F}}_h = \mathbf{F}(\hat{f}_h), \quad \hat{\mathbf{f}}_h = \mathbf{f}(\hat{f}_h).$$

Therefore, to perform the analysis and search a prediction which cancel the residue of (5.21), we use the result obtained with the the *regular ansatz* (chapter 1, whose consistency has been checked also in chapter 4 for the moving boundary algorithm), assuming that the LB solution is approximated by a prediction

$$F_h^{LBM} = \sum_{k=0}^{2} h^k f^{LBM,(k)},$$

with coefficients(1.163):

$$\begin{aligned} f_i^{LBM,(0)} &= f_i^*, \\ f_i^{LBM,(1)} &= f_i^* c_s^{-2} \mathbf{c}_i \cdot \mathbf{u}, \\ f_i^{LBM,(2)} &= f_i^* c_s^{-2} p + H_i^{Q(eq)}(\mathbf{u}, \mathbf{u}) - \tau f_i^* c_s^{-2} (\mathbf{c}_i \cdot \nabla) \mathbf{c}_i \cdot \mathbf{u}, \end{aligned} \tag{5.22}$$

where $\mathbf{u}$ and p solve the Navier-Stokes problem (5.1).

By definition, an asymptotic expansion for the force is the sum over the grid point $(\mathbf{k}, i)$ in B^n of the asymptotic expansions for the momentum exchange $\phi_i(\mathbf{k})$.
A prediction for $\phi_i(\mathbf{k})$ is derived inserting the prediction for $\hat{f}_i$, in equations (5.19) and using the relation (4.15) for the population updated with the boundary conditions. We have (dropping the time dependence for brevity):

$$\phi_i(\mathbf{k}) = \phi_i^{(0)}(\mathbf{b}_i(\mathbf{k})) + h^2\phi_i^{(2)}(\mathbf{b}_i(\mathbf{k})) + O(h^3), \tag{5.23}$$

with

$$\begin{aligned}\phi_i^{(0)} &= 2f_i^*\mathbf{c}_i \\ \phi_i^{(2)} &= 2f_i^*c_s^{-2}\left(p + \frac{c_s^{-2}}{2}\left(|\mathbf{c}_i\cdot\mathbf{u}_B|^2 - c_s^2\mathbf{u}_B^2\right) - c_s^{-2}\nu\mathbf{c}_i\cdot\nabla\mathbf{u}_B\cdot\mathbf{c}_i\right)\mathbf{c}_i.\end{aligned} \tag{5.24}$$

All the quantities in equations (5.23)-(5.24) are evaluated at the boundary point $\mathbf{b}_i^n(\mathbf{k})$.

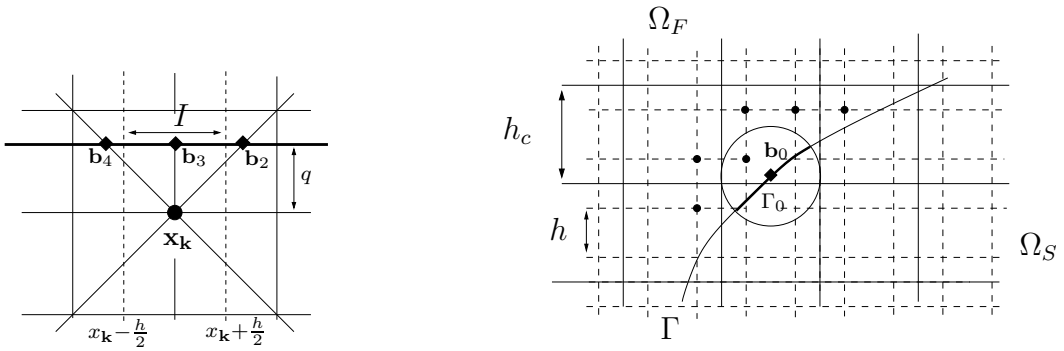


Figure 5.12: **Left:** Formula (5.24) for horizontal boundary. The points on the boundary ($\diamond$) where the functions are evaluated can be interpreted as nodes of a quadrature rule for the integral (5.2) over an interval I of length h. The location of such nodes depends on the distance q. **Right:** Meaning of the *coarser grid* h_c introduced in theorem 5.1. For a point $\mathbf{g}_0 \in \Gamma$ ($\diamond$), the local boundary $\Gamma_0(\mathbf{g}_0, h_c)$ (bold line inside the circle) can be identified as a ball centered in $\mathbf{g}_0$ and diameter h_c, intersected with the interface. The momentum exchange is evaluated at the points interacting with Γ_0 ($\bullet$).

Plane horizontal boundary To better understand equation (5.24), we look at a simple example, with an horizontal boundary on the top of the fluid flow (figure 5.12 left).
In our convention for the discrete velocities of the model D2Q9 (figure 1.2), the last row of fluid nodes interacts with the solid along the directions $i = 2, 3, 4$. Computing explicitly the sum $\Phi(\mathbf{k}) = \phi_2(\mathbf{k}) + \phi_3(\mathbf{k}) + \phi_4(\mathbf{k})$ with equation (5.24),

for a particular boundary node $\mathbf{x_k}$, we have (suppressing the argument $\mathbf{k}$)

$$\Phi = \frac{1}{3}\begin{pmatrix} 0 \\ 1 \end{pmatrix} + h^2 \begin{pmatrix} -\frac{1}{2}[\mathbf{S}^\nu_{xy}(\mathbf{b}_2) + \mathbf{S}^\nu_{xy}(\mathbf{b}_4)] \\ \frac{1}{6}[p(\mathbf{b}_2) + 4p(\mathbf{b}_3) + p(\mathbf{b}_4)] - \mathbf{S}^\nu_{yy}(\mathbf{b}_3) \end{pmatrix} + \\ + h^2 \begin{pmatrix} \frac{1}{2}[u_B(\mathbf{b}_2)v_B(\mathbf{b}_2) + u_B(\mathbf{b}_4)v_B(\mathbf{b}_4)] \\ \frac{1}{6}[v_B(\mathbf{b}_2) + 4v_B(\mathbf{b}_3)^2 + v_B(\mathbf{b}_4)^2] \end{pmatrix} + O(h^3) \quad (5.25)$$

The zero order term predicts a surplus of pressure[d], and it is not related to the integral (5.2). The second order is a combination of quadrature formulas over a small interval on the boundary for the functions p, $\mathbf{S}^\nu$, plus a quadratic function of velocity, which breaks the *Galilean invariance* (as happened in the test problem CiF_0, figure 5.11).

5.3.2 Corrected and averaged momentum

After discovering the unwanted terms in expression (5.24), we can use the expansion to define a *corrected* momentum exchange algorithm, based on the values

$$\overline{\phi}_i(\mathbf{k}) = \phi_i(\mathbf{k}) - 2f_i^*\mathbf{c}_i - h^2 f_i^* c_s^{-4}\left(|\mathbf{c}_i \cdot \mathbf{u}_B(\mathbf{b}_i^n(\mathbf{k}))|^2 - c_s^2 \mathbf{u}_B(\mathbf{b}_i^n(\mathbf{k}))^2\right)\mathbf{c}_i. \quad (5.26)$$

Using this modification, the simple test problem CiF_0 with zero boundary stresses is now solved correctly.
We continue our analysis with problem CiF_1 where a prescribed pressure distribution appears on the boundary. Results obtained with the modified MEA are shown in figure 5.13. Obviously, the approximation of the local stresses is still unsatisfactory[e]. In the special case of horizontal boundary the sum of momentum exchange in a boundary node (5.25) had a clear relation with an approximate integration rule. However, for general curved boundary the distribution of these points and of the outgoing directions along the interface is extremely *irregular*, and the momentum exchange in a single boundary node might not be directly related to an approximation of the stresses on the interface. Moreover, the ME-interaction is *discrete*, i.e. using directly the momentum exchange $\overline{\phi}_i(\mathbf{k})$ as approximation of the stress in the point $\mathbf{b}_i^n(\mathbf{k})$ allows only to define the boundary interaction in special points (the intersections between grid and lattice). In other words, the MEA does not allow to define the force acting on an arbitrary $\mathbf{b} \in \Gamma$.

[d]Unless (as pointed out in [36]) we define the pressure using

$$\hat{p} := c_s^2 \hat{\rho}.$$

The term of order zero in equation (5.24) would be than encompassed into the pressure. This, however, does not suffice to correct the original algorithm in a general case.

[e]It should be remarked that the MEA in its original form (algorithm 5.1) has not been designed for local stress evaluation. However, for our purpose it represents just an initial step, to be further improved.

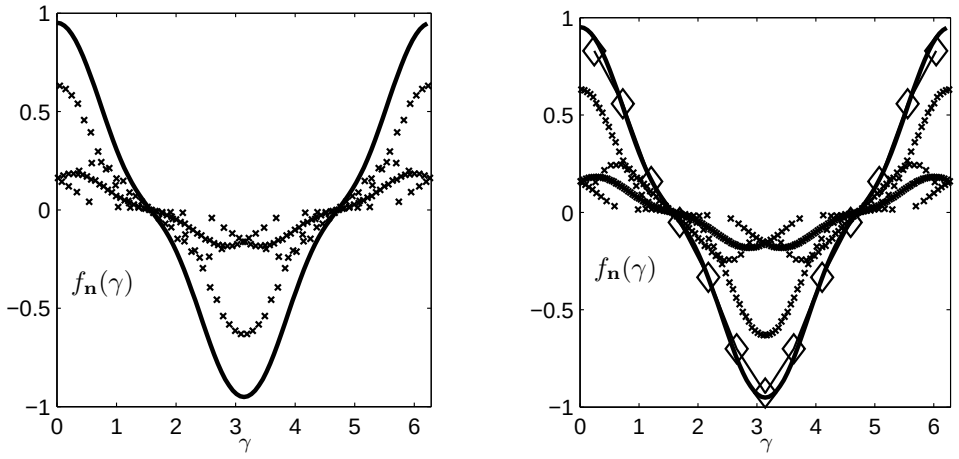


Figure 5.13: Problem CiF_1 (pressure different from 0). **Left:** The symbols ($\times$) denote the values (5.26) for each boundary couple $(\mathbf{k}, i)$ versus the related point $\mathbf{b}_i^n(\mathbf{k}) \in \Gamma$, identified by $\gamma \in [0, 2\pi)$, for a 50×50 grid. The solid line is the exact solution (5.10). **Right:** Results using a 100×100 grid. The approximation on the fine scale (crosses) is more noisy, but does not improve the approximation of the local stresses. Averaged values, computed grouping the points according to a grid $h_c = h^{0.5}$ ($\diamond$), are indicated by diamonds.

To overcome these problems, we have analyzed an *averaged* value of the momentum exchanged along small intervals on the boundary. In practice, we choose a partition $\{\mathbf{b}_m\} \subset \Gamma$ based on a coarse grid size $h_c > h$. The approximation of the local force in $\mathbf{b}_m$ is computed summing all the momentum exchange contributions (with a proper weight relating h and h_c) of the couples $(\mathbf{k}, i) \in B(\Gamma)$ such that the corresponding $\mathbf{b}_i^n(\mathbf{k})$ belongs to an h_c neighborhood of $\mathbf{b}_m$.

Algorithm 5.2 (Coarsening procedure).

let be given:
the set $\mathcal{B} = \{\gamma_r = \gamma(\mathbf{b}_{i_r}(\mathbf{k}_r)) \mid (\mathbf{k}_r, i_r) \in B\}$ (intersections grid-boundary)
the values $ME := \{\phi_i(\mathbf{k}) \mid (\mathbf{k}, i) \in B\}$
`set the coarse grid parameter` h_c
`define partition of the interface:`
$\{\mathbf{b}_m = \mathbf{b}(\gamma_m^c) \mid \gamma_m^c = mh_c,\ m = 1, \ldots, M\}$
`DO` $l = 1, M$
`initialize the averaged forces` $\mathbf{F}_m^c = 0$
`DO` $j = 1, |\mathcal{B}|$
`IF` $(m - \frac{1}{2})h_c < \gamma_r < (m + \frac{1}{2})h_c$
$\mathbf{F}_m^c = \mathbf{F}_m^c + \phi(\mathbf{k}_r)$
`END IF`
`END DO`

`normalize` $\mathbf{F}_m^c = \mathbf{F}_m^c \dfrac{h}{h_c}$

`END DO`

This algorithm leads to better results (figure 5.13, right). From a theoretical point of view, the improvements can be rigorously stated using the asymptotic prediction for the momentum exchanged. The following result is valid also for the three dimensional $D3Q15$ model.

Theorem 5.1.
Let $\mathbf{b}_0 \in \Gamma$ *be a point on a smooth d-dimensional interface* ($d = 1, 2$) *of finite length. Given the LB-grid size* h*, we consider a* coarser *grid* h_c*, such that* $h \in o(h_c)$*, and the related interval* $\Gamma_0(\mathbf{b}_0, h_c) = \left\{ \mathbf{b} \in [0, 2\pi) : \|\mathbf{b} - \mathbf{b}_0\| < \dfrac{h_c}{2} \right\}$.
Defining local averages of the exact and approximate normal stress in Γ_0

$$\mathcal{I}(\mathbf{b}_0, h_c) = \frac{1}{h_c^d} \int_{\Gamma_0} (-p\mathbf{I} + \mathbf{S}) \cdot \mathbf{n} \, d\sigma, \quad \overline{\Phi}(\mathbf{b}_0, h_c) = \left(\frac{h}{h_c}\right)^d \sum_{(\mathbf{k},i):\mathbf{b}_i(\mathbf{k}) \in \Gamma_0} \frac{\overline{\phi}_i(\mathbf{k})}{h^2},$$

the following estimate holds

$$\left| \mathcal{I}(\mathbf{b}_0, h_c) - \overline{\Phi}(\mathbf{b}_0, h_c) \right| \in O\left(h_c + \frac{h}{h_c}\right). \tag{5.27}$$

The detailed proof is given in appendix B. Shortly, it is based on writing the sum $\bar{\Phi}(\mathbf{b}_0, h_c)$ in terms of the functions p and $\mathbf{S}^\nu$, using equation (5.24) combined with a Taylor expansion around the node $\mathbf{b}_0$. The resulting expressions can be viewed as approximate integration rules on the interface. Unfortunately, the weights of the arising quadrature formulas do not sum up exactly to one at every node, which rules out first order accuracy. However, using some arithmetical properties of the weights, it can be shown that the deviation from one goes to zero, if the weights are summed over subsets of the interface which are large compared to the grid size h of the regular grid. On the other hand, the Taylor approximation is less accurate if it is used on a coarse mesh of typical distance h_c. Hence, a balance between fine and coarse grid arises in equation (5.27), and to obtain an optimal error bound, a good compromise is required.

Corollary 5.1 (Optimal coarsening).
Choosing $h_c = \sqrt{h}$*, equation (5.27) gives*

$$\left| \mathcal{I}(\mathbf{b}_0, \sqrt{h}) - \bar{\Phi}\left(\mathbf{b}_0, \sqrt{h}\right) \right| \in O\left(\sqrt{h}\right). \tag{5.28}$$

To validate the result of theorem 5.1, we compare the momentum exchange evaluated along the boundary point in problem CiF_1, averaged according to several coarser grids of type $h_c = h^\alpha$. The order plot in figure 5.14 confirms the predicted result. The best rate is obtained choosing $h_c = \sqrt{h}$.
It should be noted that the averaging does not affect the accuracy of the global force evaluation.

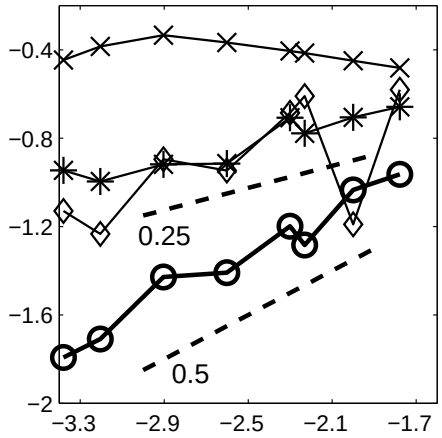


Figure 5.14: Double logarithmic plot of the error in the local forces versus the grid size h. Comparisons of different coarser grids $h_c = h^\alpha$, with $\alpha = 0.25(\diamond), 0.5(\circ), 0.75(*), 0.9(\times)$. The dashed lines represent reference slopes.

Theorem 5.2. *Let Γ be a smooth d-dimensional interface of finite length. Denoting with $\overline{\Phi}(\Gamma) = \sum_{(\mathbf{k},i)\in B(\Gamma)} \frac{\overline{\phi}_i(\mathbf{k})}{h^2}$ the total sum of corrected momentum exchange, it holds*

$$\left| \int_\Gamma (-p\mathbf{I} + \mathbf{S}) \cdot \mathbf{n}\, d\sigma - h^d \overline{\Phi}(\Gamma) \right| \in O(h). \tag{5.29}$$

The result is proven in appendix B.

Test on the cylinder in flow In figure 5.15 we show the result (double logarithmic plot) for the MEA used to compute the total force on the disk in the problem CiF_1. The value for the exact solution is obtained using a high accuracy numerical quadrature for the pressure (5.9). The plot confirms the expected first order accuracy.

5.4 The ME-stress extraction

In a practical simulations, we need the force acting on the structure to solve the dynamics. In the case of flows around rigid bodies the result of theorem 5.2 authorizes to use the MEA to evaluate the global force, since the predicted accuracy is the same as for the pressure (first order in h). Nevertheless, there is a wide class of problem (typically, interaction with *deformable structure* or *multiphase* flows, when the computation involves local stresses along the interface) where the local evaluation with the *averaged MEA* is not satisfactory. Any approximate

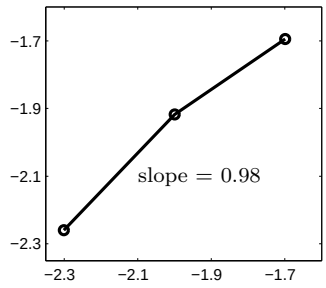


Figure 5.15: Double logarithmic plot (logarithm of error *vs* logarithm of the grid size) for the MEA, used to compute the force in the problem CiF_1. The value used as exact solution is a high order numerical approximation of the integral (5.2).

solver for the movement of the interface would carry an error of *at least* $O(\sqrt{h})$, consequently spoiling the accuracy of the LBM.

5.4.1 Improving the stress evaluation

The origin of the loss of accuracy for the boundary forces, can be identified retracing the proof of theorem 5.1. Actually, in the first example (horizontal boundary) presented in section 5.3 everything works fine. First order accurate result are produced, for *both* local and global forces. In that special case, the *extremely regular* structure of the geometry, allows to define naturally a $O(h)$-discretization of the interface and gives the *local* sum

$$\Phi(\mathbf{k}) = \sum_{i:(\mathbf{k},i)\in B} \phi_i(\mathbf{k})$$

additional properties, which are in general true only after averaging over a large number of points.
To fix the ideas, let us consider the predicted expression for $i = 1$

$$\phi_1(\mathbf{k}) = \left(\frac{2}{3}p(\mathbf{b}_1) - \mathbf{S}^{\nu}_{xx}(\mathbf{b}_1)\right) + O(h). \tag{5.30}$$

Assuming first order accuracy for p and $\mathbf{S}^{\nu}$, (5.30) contains already an approximate combination of stresses. The problem is that p and $\mathbf{S}^{\nu}$ are multiplied by different constants. In some situations, like horizontal boundaries, the resulting combination of ϕ_i is balanced exactly when summing the contributions, which leads to an immediate relation with

$$-p\mathbf{n} + \mathbf{S}^{\nu} \cdot \mathbf{n}. \tag{5.31}$$

However, this does not happen for general boundaries.

The scope of the averaging is to smooth out the differences between the discrete sum and the integrand (5.31). However, this is by far not the only solution to the problem.

The ME based extrapolation Since ϕ_i (as in (5.30)) contains the quantities p and $\mathbf{S}^\nu$, which we are interested in, we will try to construct an extrapolation rule based on the MEA. Once more, we will show how the knowledge acquired with the asymptotic analysis is crucial for improving an algorithm. Everything we will do is based on the *predicted* coefficient $\phi^{(2)}$.
Namely, our scope is to find a sequence of operations in order to extract a first order approximation of the pressure and the stress tensor from ϕ_i.

Simple algorithm to extract the stresses

We assume that running the LBM we get a first order correct pressure $\hat{p}$. The idea at the basis of the algorithm is straightforward. A first order approximation of the pressure (as well as of the stress tensor) in $\mathbf{b}_i^n(\mathbf{k})$, is given by the value

$$\hat{p}(\mathbf{k}) = p(\mathbf{x}_\mathbf{k}) + O(h) = p(\mathbf{b}_i^n(\mathbf{k})) + O(h)$$

at the boundary node, to which $\mathbf{b}_i^n(\mathbf{k})$ and $\phi_i(\mathbf{k})$ are related.
Therefore, from

$$\sigma_i(\mathbf{k}) = \bar{\phi}_i(\mathbf{k}) - h^2 \mathbf{a}^{p,i} \hat{p}(\mathbf{k}) = h^2 \left(\mathbf{a}^{\mathbf{S}^\nu_{\alpha,\beta},i} \mathbf{S}^\nu_{\alpha,\beta}[\mathbf{u}(\mathbf{k})] + O(h) \right) \tag{5.32}$$

we obtain a first order approximation of the stress tensor:

$$\widehat{\mathbf{S}^\nu}_{\alpha\beta}[\mathbf{u}(\mathbf{k})] = \frac{\sigma_i}{h^2 \mathbf{a}^{\mathbf{S}^\nu_{\alpha,\beta},i}} = \mathbf{S}^\nu_{\alpha,\beta}[\mathbf{u}(\mathbf{k})] + O(h).$$

The procedure to approximate the local interaction is based on the idea

$$\begin{gathered} \text{MEA (corrected) + approximation of pressure + extraction of stress tensor} \\ = \\ \text{local interaction} \end{gathered}$$

Of course, the other way around is also possible, subtracting from ϕ_i the stress tensor on the node $\mathbf{k}$. However, the choice of the pressure leads to a cheaper algorithm.
Notice that from (5.32), we can extract the approximation of the component $\mathbf{S}^\nu_{\alpha,\beta}$, for α, β, i such that $\mathbf{a}^{\mathbf{S}^\nu_{\alpha,\beta},i} \neq 0$. In practice

$$\begin{aligned} &\phi_1,\ \phi_5 \to \mathbf{S}^\nu_{xx} \\ &\phi_2,\ \phi_4,\ \phi_6,\ \phi_8 \to \mathbf{S}^\nu_{xy} \\ &\phi_3,\ \phi_7 \to \mathbf{S}^\nu_{yy}\,. \end{aligned}$$

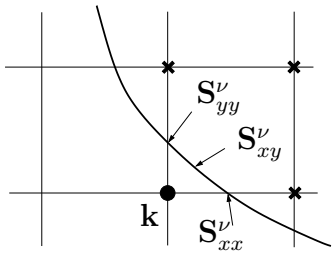


Figure 5.16: Points along the interface in which the stresses are evaluated. The different partitions, used in the proof of lemma 5.2, are used to integrate the different component of $\mathbf{S}^{\nu}$.

Hence, for each couple $(\mathbf{k}, i) \in B^n$, only a single component of the stress tensor is available, at the point $\mathbf{b}_i^n(\mathbf{k}) \in \Gamma$. Concerning the pressure, in each intersection $\mathbf{b}_i^n(\mathbf{k})$, we take the value in the node $\mathbf{k}$.
Formally, we propose the following algorithm:

Algorithm 5.3 (ME-stress extraction).
boundary set: $B = \{(\mathbf{k}_r, i_r)\}$, intersections: $\left\{\gamma_r = \gamma(\mathbf{b}_{i_r}^n(\mathbf{k}_r))\right\} \subset [0, 2\pi)$
`corrected momentum exchange:` $\rightarrow \overline{\phi}_{i_r}(\mathbf{k}_r)$
`pressure evaluation:` $\rightarrow p(\mathbf{k}_r)$
`compute` $\sigma_{i_r}(\mathbf{k}_r) = \overline{\phi}_{i_r}(\mathbf{k}_r) - h^2 \mathbf{a}^{p,i} p(\mathbf{k})$
`EXTRACT STRESSES:`
`IF` $i_r = 1, 5$, $\widehat{\mathbf{S}^{\nu}}_{xx}(\gamma_r) = h^{-2}\sigma_{i_r}(\mathbf{k}_r)$
`IF` $i_r = 2, 6$, $\widehat{\mathbf{S}^{\nu}}^{a}_{xy}(\gamma_r) = h^{-2} 2\sigma_{i_r}(\mathbf{k}_r)$
`IF` $i_r = 3, 7$, $\widehat{\mathbf{S}^{\nu}}_{yy} = h^{-2}\sigma_{i_r}(\mathbf{k}_r)$
`IF` $i_r = 4, 8$, $\widehat{\mathbf{S}^{\nu}}^{b}_{xy} = h^{-2} 2\sigma_{i_r}(\mathbf{k}_r)$
`construct partitions` $\mathcal{B}^{(i,i^*)} := \{\mathbf{b}_{i_r}(\mathbf{k}_r) \mid i_r = i, i^*\} = \left\{\mathbf{b}\left(\gamma_r^{(i,i^*)}\right)\right\}$

Having stored the values on the interface on the different partitions $\mathcal{B}^{(i,i^*)}$, we need to post-process them, in order to calculate the local stresses, which combines pressure and stress tensor. To do this, we first define a regular partition on the interface of size $\Delta\gamma$ (with $\Delta\gamma \in O(h)$):

$$\overline{\gamma} = (\overline{\gamma}_m), \quad \overline{\gamma}_m := m\Delta\gamma;$$

then, we collect for each interval $\left[\overline{\gamma}_m - \dfrac{\Delta\gamma}{2}, \overline{\gamma}_m + \dfrac{\Delta\gamma}{2}\right]$ the points of the i-partitions which fall into it, averaging between the corresponding values of p and $\mathbf{S}^{\nu}$ (figure 5.17).
Remarks. Such computation of the stress tensor along the interface, has been inspired by the proof of theorem 5.2 (appendix B), treating differently the different discrete velocity links.

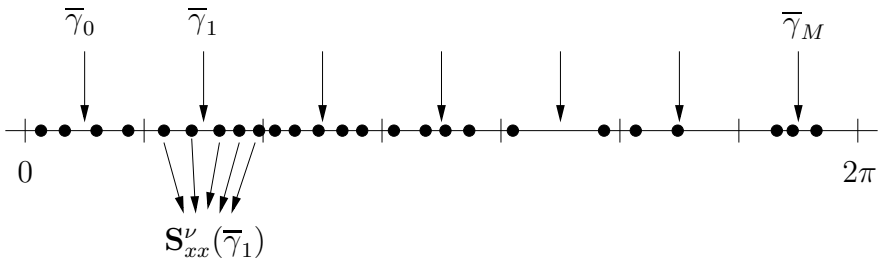


Figure 5.17: Sketch of the operation done in algorithm 5.4. On the interface $\Gamma \sim [0, 2\pi]$, we define the regular grid $\overline{\gamma}_m$. The circles • are the node in which we have stored the stresses running algorithm 5.3. The values of p and $\mathbf{S}^{\nu}$ in a node $\overline{\gamma}_m$, are obtained as the average of those related to the nodes belonging to the cell $\left[\overline{\gamma}_m - \frac{\Delta\gamma}{2}, \overline{\gamma}_m + \frac{\Delta\gamma}{2}\right]$.

Algorithm 5.4 (MEA-local stresses).

`define a regular partition:` $\overline{\gamma}_m = m\Delta\gamma$

`pressure evaluation:`

`initialize` $n_{int} = 0$, $\overline{p}_m = 0$

`FOR` $r = 1, \ldots, |\mathcal{B}|$

$\{$ `IF` $|\gamma_r - \overline{\gamma}_m| \leq \dfrac{\Delta\gamma}{2}$

$\overline{p}_m = \overline{p}_m + \hat{p}(\mathbf{k}_r)$

$n_{int} = n_{int} + 1$ $\}$

`END`

$\overline{p}_m = \dfrac{\overline{p}_m}{n_{int}}$

`stress tensor:`

$n_{int} = 0$

`FOR` $r = 1, \ldots, |\mathcal{B}^{(1,5)}|$

$\{$ `IF` $|\gamma_r^{(1,5)} - \overline{\gamma}_m| \leq \dfrac{\Delta\gamma}{2}$

$\overline{\mathbf{S}^{\nu}_{xxm}} = \overline{\mathbf{S}^{\nu}_{xxm}} + \hat{\mathbf{S}^{\nu}}_{xx}(\gamma_r^{(1,5)})$

$n_{int} = n_{int} + 1$ $\}$

`END`

$\overline{\mathbf{S}^{\nu}_{xxm}} = \dfrac{\overline{\mathbf{S}^{\nu}_{xxm}}}{n_{int}}$

`IF` $n_{int} = 0, \Rightarrow \overline{\mathbf{S}^{\nu}_{xxm}} = 0$

(analogous for $\mathbf{S}^{\nu}_{xy}$, $\mathbf{S}^{\nu}_{yy}$)

`LOCAL STRESSES:`

`tangential:` $\left(\hat{f}_{\mathbf{t}}^{\,MEA}\right)_m = \left(\overline{\mathbf{S}^{\nu}}_m \cdot \mathbf{n}\right) \cdot \mathbf{t}$

`normal:` $\left(\hat{f}_{\mathbf{n}}^{\,MEA}\right)_m = -\overline{p}_m + \left(\overline{\mathbf{S}^{\nu}}_m \cdot \mathbf{n}\right) \cdot \mathbf{n}$

Regarding the off diagonal component $\mathbf{S}^{\nu}_{xy}$, two different approximations (in two different sets of points) are computed, one for the couple of direction $(2, 6)$, one for the couple of direction $(4, 8)$, and the arithmetic average is used.

5.4.2 Numerical tests and comparisons

To test the MEA and the procedure 5.4, we use the benchmark drawn in figure 5.9, simulating a rotating cylinder between two walls.
The problem is simulated on a channel with $L = H = 1$, grid 40×40, and choosing

$$\begin{aligned} &\mathbf{u}_N(t) = \mathbf{u}_S(t) = (-1, 0) \\ &\|\mathbf{\Omega}^D(t)\| = 3 \\ &R = 0.2. \end{aligned} \tag{5.33}$$

In figure 5.18 we observe the velocity and pressure field after a steady state is reached.

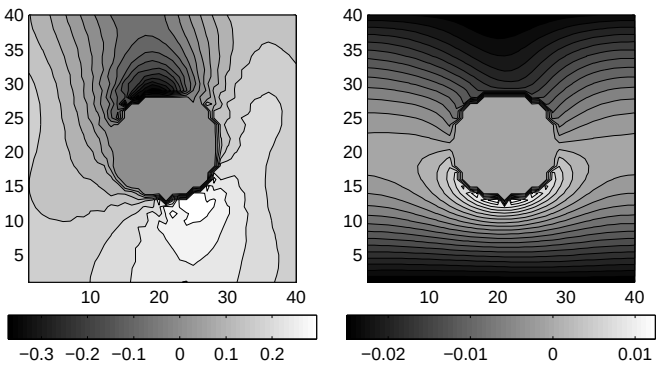


Figure 5.18: Contour lines of pressure (**left**) and horizontal velocity (**right**) fields, at time $T = 1$, for the problem CiF in a channel with non-zero angular velocity. The viscosity is $\nu = 0.03$.

Using the MEA, we compute the lift (figure 5.19) and the drag (figure 5.20) force on the disk.
Regarding the local forces, we compare the *ME-stress extraction* and the bilinear extrapolation, described in section 5.2. The results are shown in figures 5.21-5.22.

5.5 Ultimate numerical simulations

5.5.1 Moving cylinder

The next benchmark involves the refill algorithm and the force computation. Aim of the simulations is to validate the *Equilibrium + non Equilibrium* refill for the

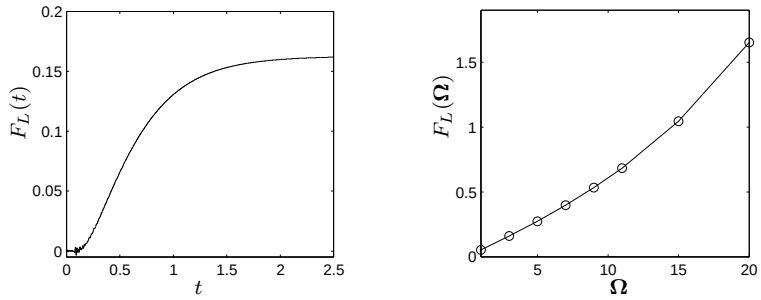


Figure 5.19: **Left:** Lift force on the cylinder versus time, for viscosity $\nu = 0.3$. After an oscillatory initial behavior, it follows a smooth profile into a steady state. **Right:** Stationary values of the lift force for different angular velocities, $0.5 \leq \Omega \leq 20$.

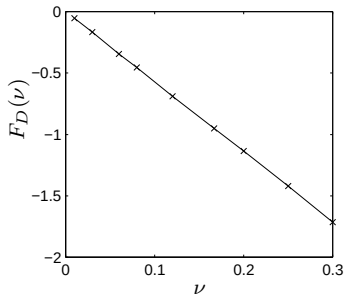


Figure 5.20: Stationary values of the drag force for different viscosities, in the range $0.01 < \nu < 0.3$.

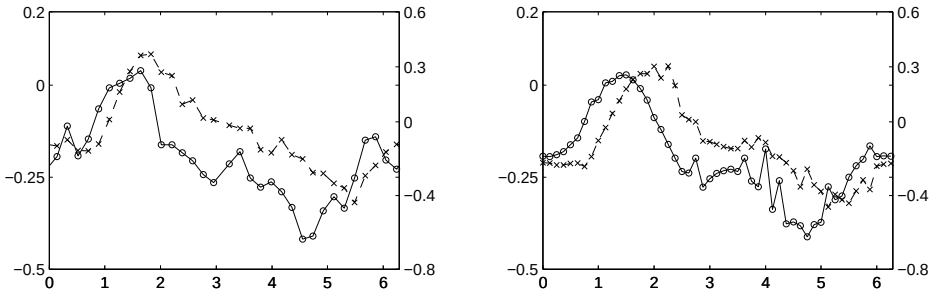


Figure 5.21: **Left:** Usage of the MEA to obtain the local forces on the interface, with the procedure 5.4. Tangential stress $\hat{f}_{\mathbf{t}}^{\text{MEA}}$ ($\circ$, left y-axis) and normal stress $\hat{f}_{\mathbf{n}}^{\text{MEA}}$ ($\times$, right y-axis) are superimposed, as functions of the coordinate $\gamma \in [0, 2\pi)$ parameterizing the interface. Compare with figure 5.9. **Right:** The evaluation using *bilinear* extrapolation is reported. Results appear more regular, but the algorithm requires higher computational effort.

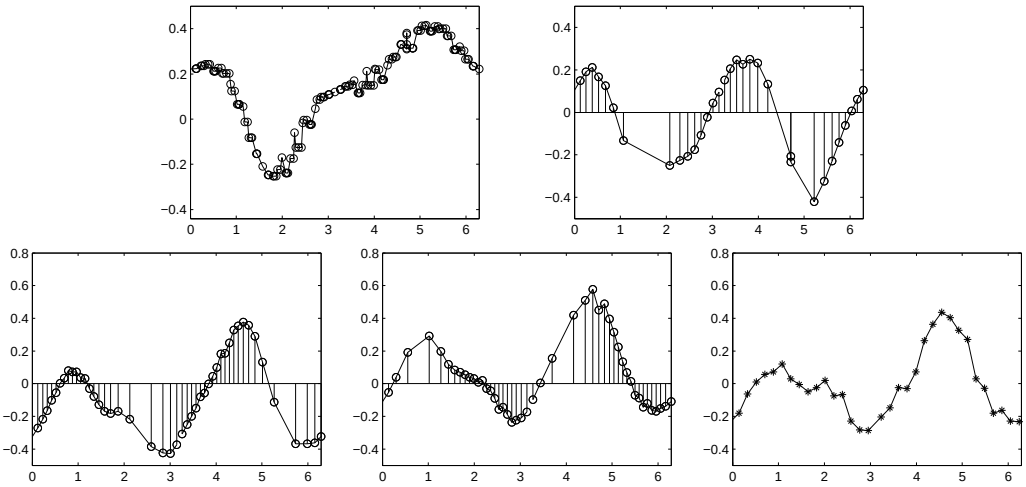


Figure 5.22: The quantities p, $\mathbf{S}^{\nu}_{xx}$, $\mathbf{S}^{\nu}_{xy}$, used in algorithm 5.4 are drawn along the interface (as functions of $\gamma \in [0, 2\pi)$). The lines evidence the structures of the different grids. **Top-left**: pressure, evaluated as in the *closest node* extrapolation. **Top-right**: $\mathbf{S}^{\nu}_{xx}$. **Bottom**: $\mathbf{S}^{\nu}_{xy}$, the grid for the links $(2, 6)$ and $(4, 8)$ are displayed separately (circles $\circ$). An unified plot would show irregular behavior (up-down), which is smoothed out performing the average in algorithm 5.4 (rightmost plot, data are sampled with asterisks $*$). Grid size $h = \frac{1}{40}$.

moving boundary problems, and to observe the effect of refilling on the force evaluation.
The disk is placed on the midline of a channel of length and width equal to one. Periodic boundary conditions are used on the left and on the right. We consider the CiF models represented in figure 5.23. The moving boundary test is performed comparing two different situations:

$$\begin{aligned} (\mathbf{a}) &\begin{cases} u_N(t) = 1, \quad u_S(t) = 1, \\ \mathbf{u}^D_{CM} = \mathbf{0}, \quad \Omega^D = 0 \end{cases} \\ (\mathbf{b}) &\begin{cases} u_N(t) = 0, \quad u_S(t) = 0, \\ \mathbf{u}^D_{CM} = (-1, 0), \quad \Omega^D = 0 \end{cases} \end{aligned} \tag{5.34}$$

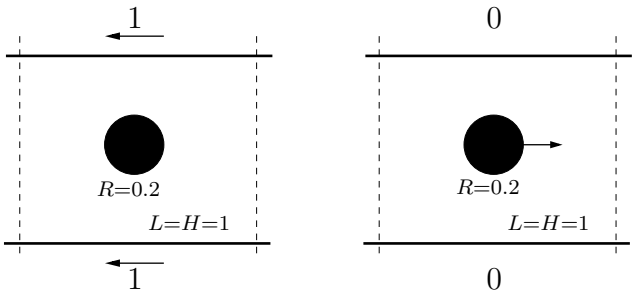


Figure 5.23: Benchmarks used to test the refill algorithms and the force computation. A disk of radius $R = 0.2$ is placed in a channel of unitary length and width. The flow viscosity is $\nu = 0.03$.

We measure the drag force acting on the disk, comparing the results in the two cases, which are at the steady state theoretically equal. For the simulations, BFL boundary conditions are used on the disk and on the walls.
We test the three different refill approaches. The stationary drag force computed on the moving disk approximates the result obtained moving the walls and keeping the disk fixed (figure 5.24). However, in the moving boundary simulations it is affected by jumps and fast frequency oscillations, which are investigated in figure 5.25 in more detail, for the different refill approaches. The results validate the theoretical conclusions of chapter 4. Using the *Equilibrium refill* the observed jumps in the force do not decrease after refining the grid. Better precision (jumps reducing as $O(h)$) is achieved with the *Interpolation+Advection* and the *Equilibrium+non Equilibrium* refill.

5.5.2 Periodic elastic fiber

To simulate a simple fluid-structure interaction problem, we consider the following variation of the CiF. The disk $\Omega_S(t)$ is located initially on the center line of a

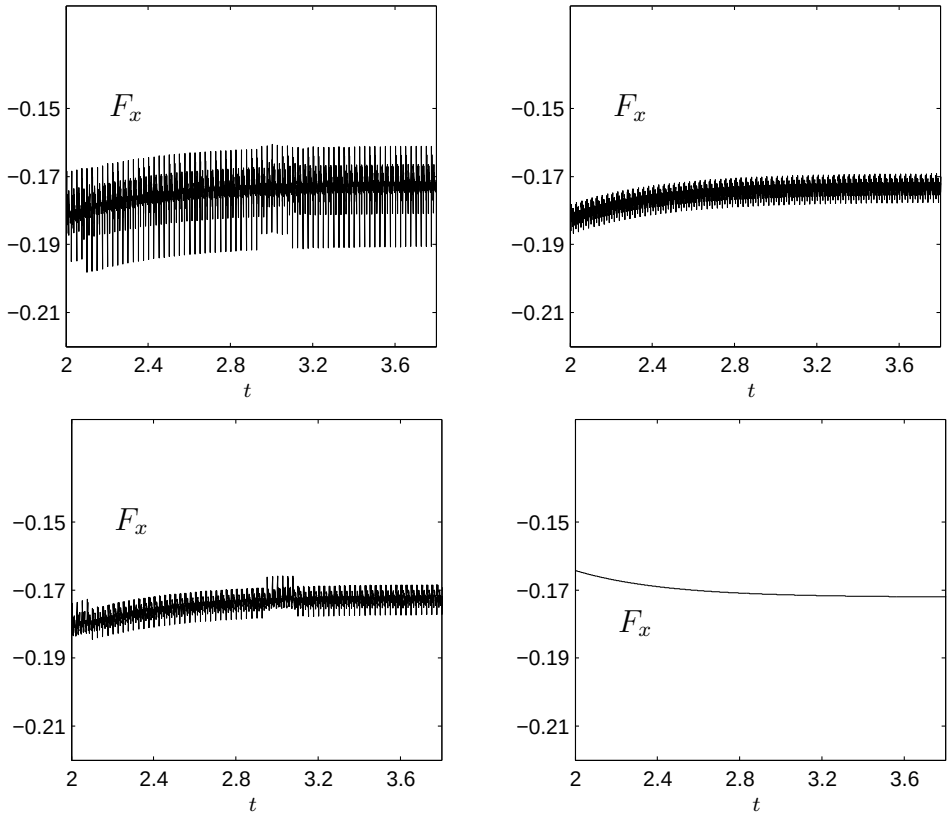


Figure 5.24: On the problem CiF in figure 5.23 we compute the drag force using the EQ-refill (**top-left**), the IA-refill (**top-right**) and the EnE-refill (**bottom-left**). Situation approaching a stationary state is shown, compared with the stationary state of a simulation in a reference system fixed with the cylinder (**bottom-right**).Grid size $h = \frac{1}{40}$.

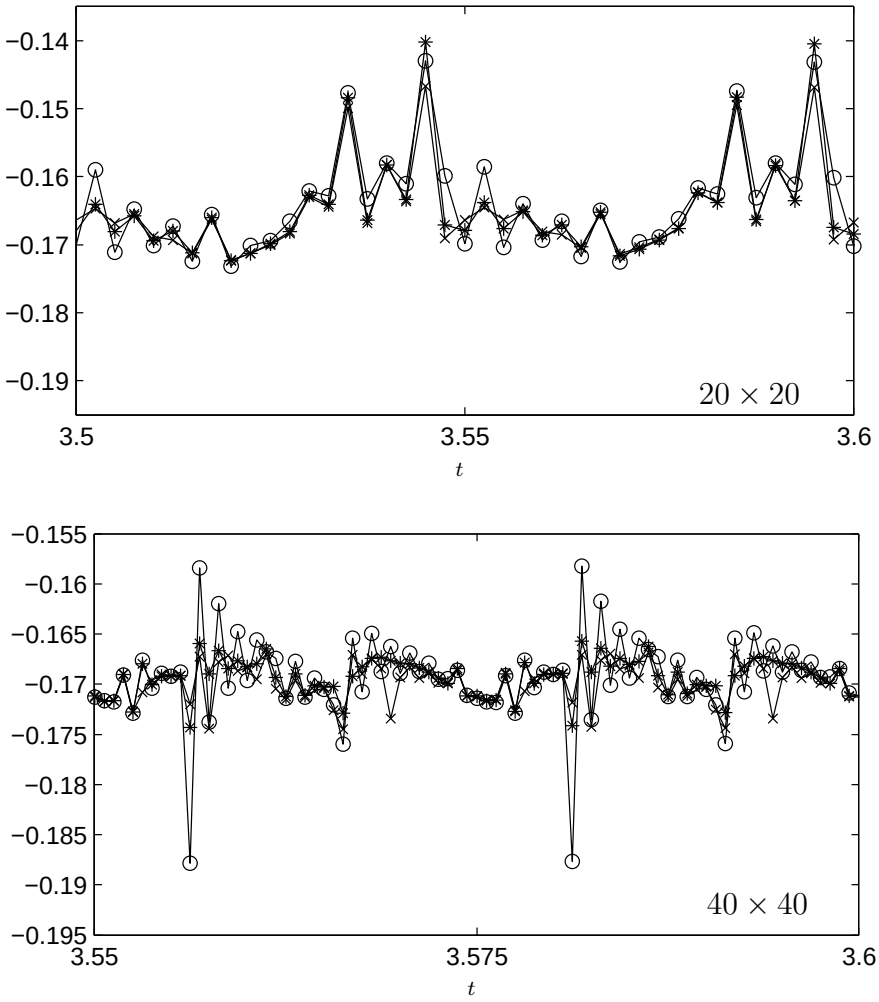


Figure 5.25: The jumps in the force computation, obtained using the different refills, are compared on two different grids, $h = \frac{1}{20}$ and $h = \frac{1}{40}$. The circles ($\circ$) sample the value for the EQ-refill, the crosses ($\times$) the IA refill and the stars ($\star$) the EnE-refill.

rectangular channel. The center of the disk $\mathbf{x}_C(t)$ is connected to a spring whose rest length is equal to zero. It can be seen as a model for the motion of a long cylindric fiber, driven by a fluid flow. We call this benchmark *periodic elastic fiber* (PEF).

Parameters and set up of the algorithm In figure 5.26 we indicate the parameters needed to set up a simulation of the benchmark PEF. To define the Navier-Stokes problem, we have to set up the domain (ratio between length L and width H), the boundary conditions (channel or fully periodic) and the initial conditions for the fluid. The parameters concerning the solid structure are the properties of the fiber (static and dynamic parameters, radius and elastic constant) and its initial motion.

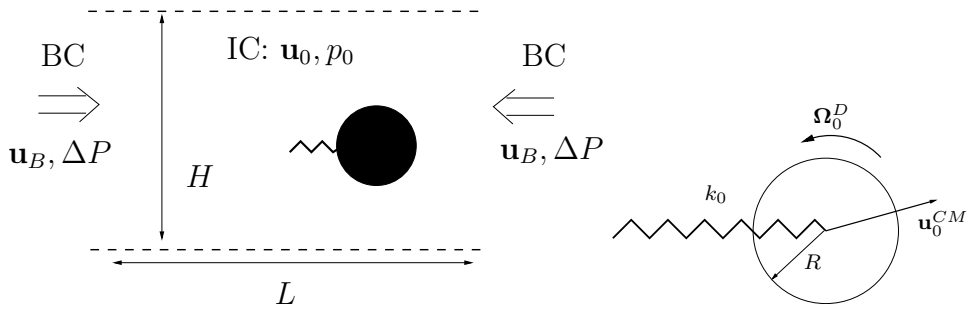


Figure 5.26: Relevant parameters to set up a PEF simulation. On the right, the solid structure.

Numerical results

PEF-1 We perform two simulations. First (figure 5.27, left), we consider the disk in a channel whose top and bottom walls move horizontally with velocity

$$\mathbf{u}_B(t) = \varphi_W(t),$$

where $\varphi_W(t)$ is the smooth profile used in chapter 4 (equation (4.36)). It has the properties

$$\begin{aligned} &\varphi_W(0) = 0, \\ &\varphi_W(t) = 1, \ t \geq 0.5 \\ &\varphi_W \in C^2. \end{aligned}$$

Initially, fluid and disk are at rest,

$$\mathbf{u}_0 = 0, \quad \mathbf{u}_0^{CM} = 0, \quad \mathbf{\Omega}_0^D = 0.$$

We fix the width $H = 1$ of the channel and choose a double length $L = 2$. The disk has radius $R = 0.2$ and it is initially placed on the midline $y = \frac{1}{2}$. This condition allows to check the correctness of the results for the lift force which has to be zero because of the symmetry. The problem is called PEF-1.

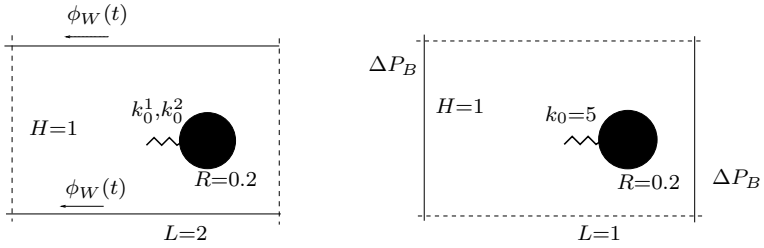


Figure 5.27: **Left:** Benchmark PEF-1. In a channel of length $L = 2$ and width $H = 1$, a disk of radius $R = 0.2$ is initially placed on the midline. Two values of the elastic constant are used, $k_0^1 = 2.5$ and $k_0^2 = 10$. The flow viscosity is $\nu = 0.055$ ($\tau = 0.67$). **Right:** Benchmark PEF-2. The domain is a periodic unit square, the disk has radius $R = 0.2$ and the elastic constant is $k_0 = 5$. A horizontal pressure drop $\Delta P_B(t)$ is applied from left to right.

The results for two different values of the elastic constant are compared, using a grid parameter $h = \frac{1}{80}$. We consider $k_0^1 = 2.5$ and $k_0^2 = 10 = 4k_0^1$. Figures 5.28-5.29 show results for drag and lift forces. The force F_x has an oscillatory profile, with exponentially damped amplitude. The jumps appear in relation with the refills. However, they are rapidly absorbed. Similarly, we observe jumps and peaks in F_y.
In the simulation with $k_0 = 10$, the oscillation period is half of the one observed for $k_0 = 2.5$. Moreover, in this second case the disk moves less, and a lower number of refills is performed. We obtain less jumps in the forces. Sizes, however, are comparable.
In figures 5.30 we compare the motion of the disk (trajectory of the center of mass is shown) inside the channel. The displacement is mainly along the x-axis. Along the y-direction, where for symmetry reasons the disk should not move, we observe only variations of very small size. The final value can be used to compute the drag force of the disk in the flow:

$$F_D = k_0(x_{CM}(t) - x_{CM}(0)).$$

The displacements, along x and y reduce by a factor four, using $k_0 = 10$. The values of the drag force in the two simulations are approximatively the same.
We observe that also the final errors in the lift forces are similar.
Finally, the difference of pressure ΔP_D at opposite sides of the disk (leftmost and rightmost point) is monitored. Figures 5.31 show these quantities as a function of time. We observe an oscillatory profile, similar to the other quantities.

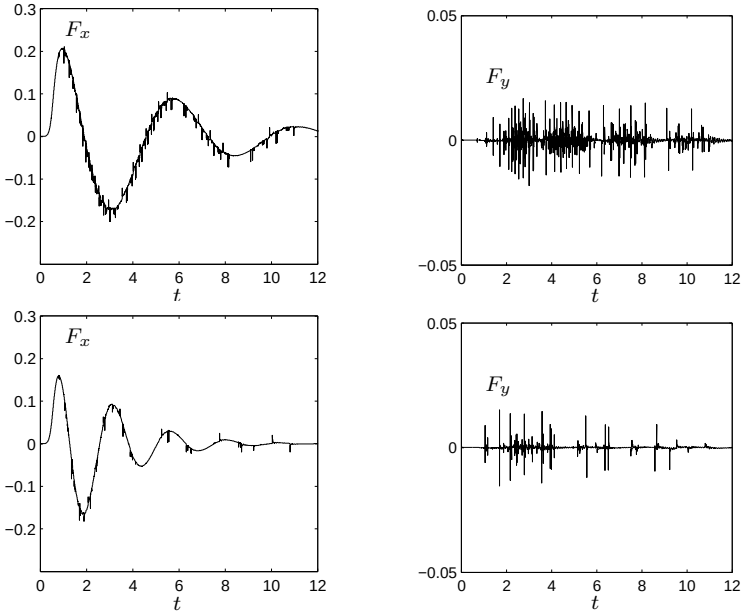


Figure 5.28: Benchmark PEF-1. The approximated drag (left) and lift (right) forces on the fiber section are shown. **Top:** The case $k_0 = 2.5$. **Bottom:** The case $k_0 = 10$. In both situations, the exact lift force is zero due to the symmetry of the problem. The grid size is $h = \frac{1}{40}$.

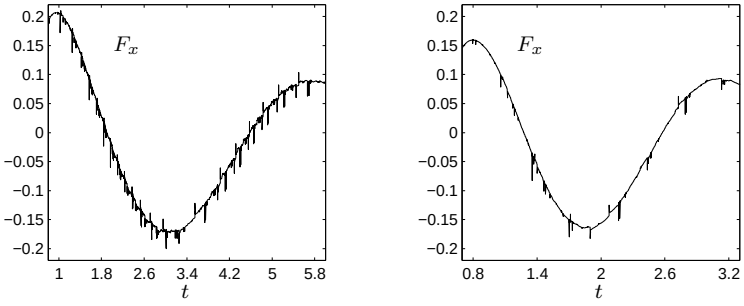


Figure 5.29: Zoom of the previous figure 5.28, on a single period of oscillation of the forces F_x. **Left:** $k_0 = 2.5$. **Right:** $k_0 = 10$.

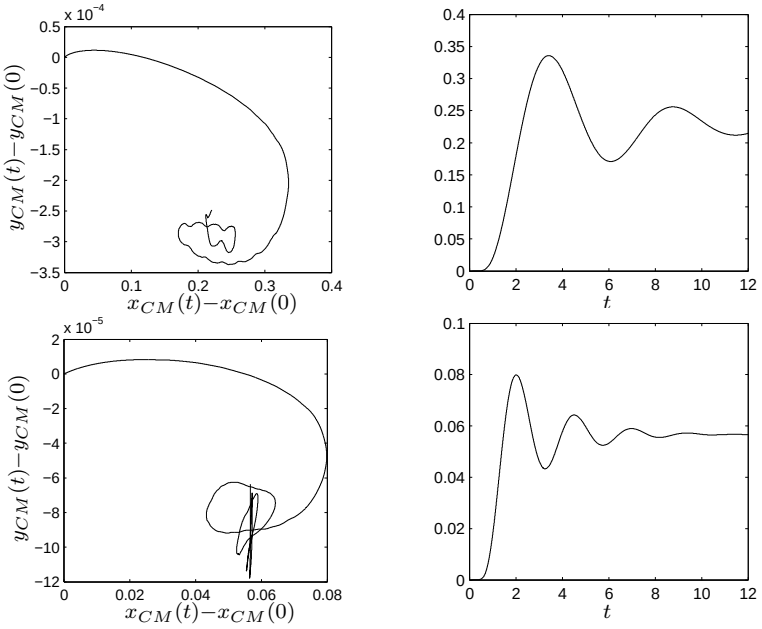


Figure 5.30: Displacement of the center of mass, with respect to the initial position. **Top:** Elastic constant $k_0 = 2.5$. On the left, the motion of the center of mass is shown. On the right, the displacement of the x-coordinate as a function of time. **Bottom:** Result for $k_0 = 10$. The higher elasticity constant has a stabilizing effect. The oscillations in the trajectory (left pictures) have smaller amplitude. Grid size: $h = \frac{1}{40}$.

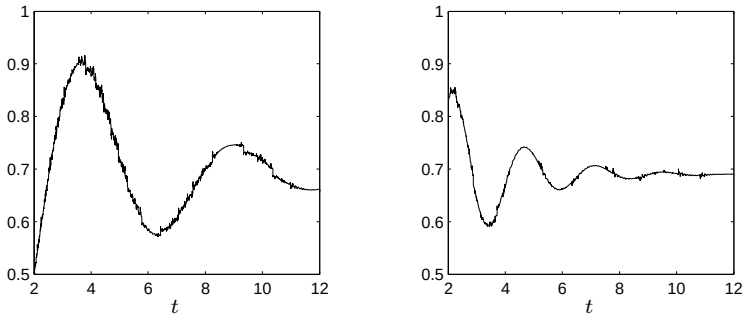


Figure 5.31: Problem PEF-1. **Left:** Difference of pressure on the opposite side of the horizontal diameter of the disk (left side minus right side), for $k_0 = 2.5$. **Right:** $k_0 = 10$. Grid size: $h = \frac{1}{40}$.

The results are affected by discontinuities due to the refills. This happens because at each refill step, we need to reconstruct the population on the new fluid node. It can be done only up to a certain order. Therefore, it is important to check the order of the jump amplitude. We simulate the same problem (PEF-1, with elastic constant $k_0 = 2.5$) on a finer grid $h = \dfrac{1}{80}$ which is half of the previous one. Results are shown in figure 5.32. The horizontal force has a more regular profile, and the peaks in the lift force (representing an error measure) visibly decrease, although the number of performed refills increases (finer grid size).

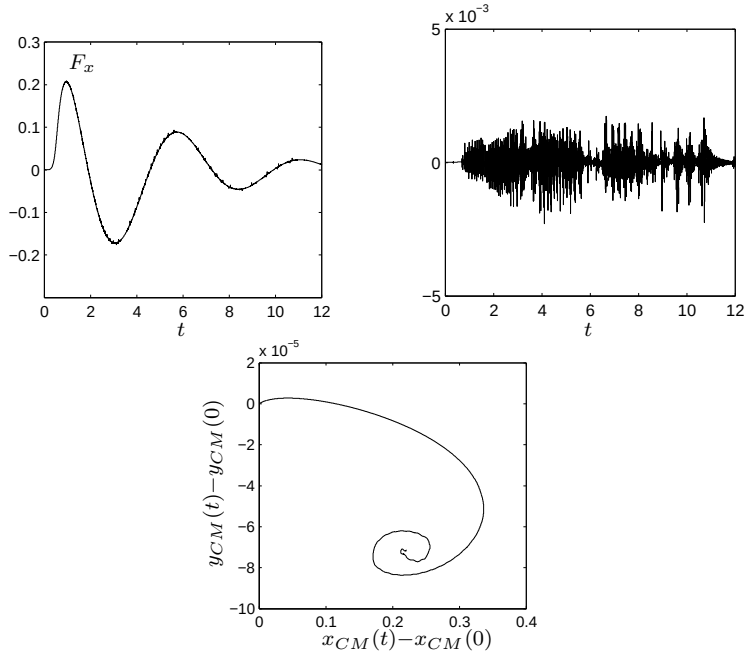


Figure 5.32: Simulation of the problem PEF-1, with elastic constant $k_0 = 2.5$ (as in figure 5.28). A grid of size $h = \frac{1}{80}$ has been used. **Top:** forces exerted on the disk. The size of the jumps has been reduced, as well as the error in F_y. **Bottom:** Trajectory of the center of mass. The displacement along the y-axis is approximatively one fourth of the one obtained with the coarser grid.

In figure 5.33 we compare the sizes of the jumps arising after the refill step, for two different grids. Since the decreasing rate is higher than the one of the grid size, we conclude that the jumps do not affect the leading order of the force. In other words, the leading orders of the hydrodynamics are continuously approximated by the refill.

PEF-2 In the second case, the domain is a periodic unit square, $H = L = 1$. The disk is placed in the center. On the boundary, we add a periodic pressure

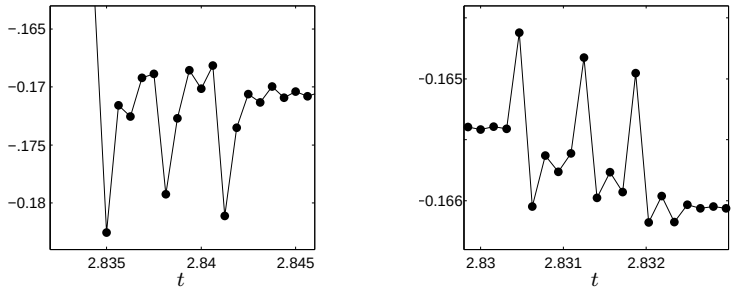


Figure 5.33: Drag force on the disk, zoom on a sequence of jumps arising after the refill steps, close to the minimum point of F_x. Each circle • represents an iteration step. **Left:** Grid $h = \frac{1}{40}$. **Right:** Grid $h = \frac{1}{80}$.

drop (section 4.1.2), forcing an oscillating gradient

$$\Delta P_B(t) = \Delta P_B^0 \sin \vartheta t.$$

For the pulsation the value $\vartheta = \dfrac{2\pi}{0.3}$ is used, and the benchmark is simulated for a time $T = 12$. At the initial time, there is no flow and disk is at rest. The elastic constant is $k_0 = 5$. We call this benchmark PEF-2 (figure 5.27, right).

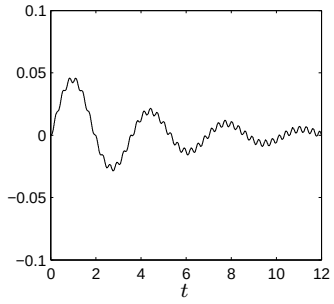


Figure 5.34: Problem PEF-2. The plot shows the abscissa of the center of mass *versus* time.

In figures 5.34, 5.35 the numerical results for $h = \dfrac{1}{40}$ are shown. We compute the position of the center of mass and the two components of its velocity, as functions of time.
The movement along the x-axis shows a transient flow, with the presence of oscillations depending on the spring constant. Later, we see a steady state governed by the gradient of pressure. The motion has the same period as $\Delta P_B(t)$.
Also the velocity shows two superimposed oscillations. In the vertical components v and y_{CM} no appreciable variations are visible. In this benchmark the initial symmetry configuration is preserved.

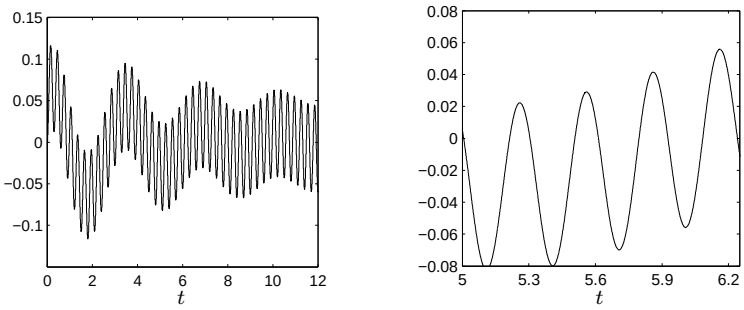


Figure 5.35: **Left:** Velocity of the center of mass, in the benchmark PEF-2. **Right:** The figure is zoomed, to evidence the period of the forced oscillations.

Summary

The force evaluation via Momentum Exchange Algorithm has been described, discussed and analyzed, compared with different approaches.
Using the results of the asymptotic analysis for the interior LBM, we have related the algorithm to the physical quantities and derived precise accuracy estimates, which justify the usage of the algorithm for *global force* computations. Namely, we show that the MEA provides in this case first order accuracy, while it is not able to achieve in general the same precision also for the local interaction.
Therefore, an improved local stresses evaluation based on the results of the analysis has been proposed and tested on simple benchmarks.

Chapter 6

Conclusions

6.1 Summary of the results

Through the thesis we have performed a detailed analysis of the LBM, starting from the properties of the standard algorithm and investigating some possible extensions.
In particular, we focused on *initialization* routines, *moving boundary* algorithms and *fluid-structure interaction* problems. Step by step, we have analyzed separately different topics, searching for suitable solution from both the theoretical and practical point of view.
The strategy of our investigation has been based on the *asymptotic analysis*. The scope of the thesis has been actually double. Besides the practical issues of developing the scheme, the particular case of the LBM has been used to explain and to show the features of the asymptotic analysis, how the technique work and its applicability.
In the following paragraphs we summarize the main results we have obtained.

Analysis of LBM: understand the scheme and the numerical results
In chapter 1 we have proposed a more general formulation of the asymptotic expansion technique. The method has been applied to the LBM, recovering, as in the standard asymptotic analysis [19], the relation with the Navier-Stokes equation and *predicting* the behavior of the numerical solution. The procedure allows to better understand the scheme and estimate theoretically the accuracy. The expectations have been validated on simple benchmarks.
Starting from the results of the first numerical tests, in chapter 2 we have developed the analysis method proposed in chapter 1 also for a more general situation, investigating in detail the arising of oscillating errors in LB simulations. As a result we obtained a set of equations to describe the dynamic of the initial layers. In some particular cases analytical solutions have been computed, able to predict the leading order error as a function of the initial conditions. The prediction has

been successfully tested in some numerical simulations.

Initialization routines: correct and improve the algorithms In chapter 3 we have described how to use the results of the analysis also to derive *corrections* to existing algorithms. In particular, we dealt with LB-initialization routines, which are used to generate initial condition for LB simulations. Starting from an algorithm for periodic domains [34], we developed some correction to improve the efficiency and the accuracy, and to extend the routine to boundary value problems.

Moving boundary and force evaluation: propose new routines To approach a LBM for moving boundary problems (chapter 4), we began considering a simple algorithm, combining the interior scheme, a boundary update rule and a refill step, i.e. a method to initialize the fluid nodes created by the movement of the boundary through the computational domain. An overview of existing algorithms [31] have been presented.
Making use of the results of the analysis, we have constructed a refill algorithm able to achieve the same accuracy as the LBM with a rather simple implementation. It has to be mentioned that a similar idea has been proposed in [16] to implement the boundary condition.
Finally, in chapter 5 we studied different choices to evaluate the forces on obstacles in flow in a LB simulation. We focused on the Momentum Exchange algorithm [28].
A detailed analysis of the algorithm has been performed, which yielded to an accuracy estimate, proven for the case of smooth boundaries. It represents the main theoretical result of this chapter. The conclusions have been validated on a simple model. According to our predictions, the MEA does not achieve enough accuracy in the approximation of the local stresses along the boundary. Therefore, we have *constructed* an improved algorithm, based on the theoretical prediction for the numerical solution.

Numerical validations In conclusion, we presented simple numerical simulations to validate the proposed extensions to the LBM. For the basic algorithm we set up, we have derived a list of properties, accuracy results and main difficulties.

6.2 Further ideas: what needs to be done and what can be done

On a theoretical level, the presented formulation of the asymptotic analysis offers a wider field of application and represents an interesting topic to be developed in future.

Concerning the practical applications, the presented algorithms have to be considered as a *starting point* to be developed in particular applications. Further validations on more general test cases are necessary, to observe how the different routines interact together in general situations.
An important point concerns the stability. Using the asymptotic expansion as we did (up to the second order in h), we are able to recover the information regarding the stability only partially. Even to investigate the interior LBM in this sense, deeper expansions are needed [23]. A detailed stability analysis of special algorithms like boundary condition rules or refills step becomes a large topic, which is left to future researches.
Moreover, in our analysis we only focused on the construction of valid schemes, consistent with the standard LBM, not dealing in detail with the global efficiency. In a real implementation of the code, a sequence of algorithms as the one presented here might not be optimal.

Outlook: fluid-structure interaction and free interface problems

The algorithm to evaluate the interface stresses can theoretically achieve the same accuracy as the LBM. It is reliable for coupled applications, since it will not reduce the precision of the solver for the hydrodynamics.
In principle, it is therefore possible to model flows through deformable structures, such as filters or obstacles with elastic interfaces.
More in general, the stress extraction algorithm can be used in the simulations of *free interface* problems. It allows to approach LB models for multiphase flows or mixtures, which are of great interests in several practical applications.

Appendix A

List of principal notations

General

$\mathbb{C}, \mathbb{R}, \mathbb{Q}, \mathbb{Z}, \mathbb{N}$	Complex, Real, Rational, Integer, Natural numbers
$\mathbf{x}, (x, y, z)$	point in Euclidean space, scalar coordinates
$\Delta, \nabla, \nabla\cdot$	Laplace operator, Gradient, Divergence
h	grid size (small parameter)
$\mathcal{F}(A, B)$	functions from the set A to the set B
χ_A	characteristic function of the set A

Hydrodynamics and Navier-Stokes equations

$\Omega, \Omega_F, \Omega_S$	Domain, fluid domain, solid domain
Γ	Boundary, interface
γ	coordinate on Γ
$\mathbf{u}$	velocity
u, v	component of velocity (in 2D)
$\mathbf{u}_0, \mathbf{u}_B$	initial and boundary velocities
ρ	density
p	pressure
$\mathbf{S}[\mathbf{u}], S_{xx}, S_{xy}, S_{yy}$	*rate-of-strain* tensor, component (in 2D)
$\mathbf{S}^\nu[\mathbf{u}]$	stress tensor
$\mathbf{G}$	volume force in Navier-Stokes equations
ν	viscosity

Asymptotic Analysis

L	general notation for algorithm
H	set of small parameters, $H \subset [0,1)$
X	domain of definition of L
$\hat{\mathbf{x}}$	solution to L
D	continuous domain
$\mathbb{D}_h$	h-discretized domain
$\mathbf{d}_h$	element of $\mathbb{D}_h$
$\mathcal{A}$	ansatz
F	prediction (element of an ansatz)
$f^{(k)}, f^{(k)}(F)$	expansion coefficients, coefficients of a $F \in \mathcal{A}$
$r^{(k)}(F)$	residue of order k for $F \in X$

Lattice Boltzmann Method

LBM	Lattice Boltzmann Method
$\hat{f}, \hat{f}_h, \hat{f}_i$	LB solution (different notations)
n	time step, state after collision
$\mathcal{G}(h)$	discrete grid of size h
$\mathbf{j}$	general grid node
$\mathbb{V}$	discrete velocity space
$\mathbf{c}_i, c_{ix}, c_{iy}$	discrete velocity, scalar components
$J(\hat{f}), \hat{J}$	LB-collision operator
f_i^c	post-collision distribution
τ	BGK relaxation time
ω	BGK relaxation frequence
$f^{eq}(\hat{f}_h)$	equilibrium distribution (function of $\hat{f}$)
$f^{L(eq)}(\hat{f}_h), f^{Q(eq)}(\hat{f}_h)$	linear and quadratic part of $f^{eq}(\hat{f}_h)$
$H^{eq}(\rho, \mathbf{u}), H^{L(eq)}(\rho, \mathbf{u}), H^{Q(eq)}(\rho, \mathbf{u})$	equilibrium as function of the moments
f_i^*	LB-weights
c_s	lattice sound speed
g_i	force term in LBE
$(\mathbf{k}, i)$	boundary couple
$\mathbf{b}_i^n(\mathbf{k})$	boundary point related to $(\mathbf{k}, i)$

Appendix B

Accuracy properties of the MEA

Theorem B.1 (5.1).
Let $\mathbf{b}_0 \in \Gamma$ *be a point on a smooth* d*-dimensional interface* ($d = 1, 2$) *of finite length. Given the LB-grid size* h*, we consider a* coarser *grid size* h_c*, such that* $h \in o(h_c)$*, and the related interval* $\Gamma_0(\mathbf{b}_0, h_c) = \left\{ \mathbf{b} \in [0, 2\pi) : \|\mathbf{b} - \mathbf{b}_0\| < \dfrac{h_c}{2} \right\}$. *Defining local averages of the exact and approximate normal stress in* Γ_0

$$\mathcal{I}(\mathbf{b}_0, h_c) = \frac{1}{h_c^d} \int_{\Gamma_0} (-p\mathbf{I} + \mathbf{S}) \cdot \mathbf{n} \, d\sigma, \quad \overline{\Phi}(\mathbf{b}_0, h_c) = \left(\frac{h}{h_c}\right)^d \sum_{(\mathbf{k},i):\mathbf{b}_i(\mathbf{k}) \in \Gamma_0} \frac{\overline{\phi}_i(\mathbf{k})}{h^2},$$

the following estimate holds

$$\left| \mathcal{I}(\mathbf{b}_0, h_c) - \overline{\Phi}(\mathbf{b}_0, h_c) \right| \in O\left(h_c + \frac{h}{h_c} \right). \tag{B.1}$$

The proof of theorem 5.1 proceeds as follows. First, in section B.1 we introduce notations and quantities which will be useful during the demonstration. Then, we analyze in section B.2 and B.3 the effect of averaging the momenta exchanged along the interface, proving the convergence toward the local integral. In section B.4 we extend the proof to the total force evaluation.
We perform a detailed computation for $d = 1$, i.e. with a one-dimensional interface in a domain $\Omega \subset \mathbb{R}^2$. The case $d = 2$ is a straightforward extension.

B.1 Preliminary definitions

Like in chapter 5, we introduce the set of boundary couples

$$B(\Gamma) = \left\{ (\mathbf{k}, i) \in \mathcal{G}(h) \times \{1, \ldots, 8\} \mid \mathbf{x}_\mathbf{k} \in \overline{\Omega}_F, \ \mathbf{x}_\mathbf{k} + h\mathbf{c}_i \in \Omega_S \right\}. \tag{B.2}$$

It depends on the domain Ω, the grid $\mathcal{G}(h)$ and the shape of the boundary Γ. In the notation of equation (B.2), we stress the influence of Γ. Each boundary

couple identifies a point on the interface:

$$\begin{array}{rcl} B(\Gamma) & \to & \mathcal{B} \subset \Gamma \\ (\mathbf{k}, i) & \mapsto & \mathbf{b}_i(\mathbf{k}) \end{array}$$

i.e. the intersection between the interface Γ and the link $\mathbf{c}_i$ starting from the node $\mathbf{k}$ (figure B.1). The base of the proof is the expansion derived for the corrected momentum exchange[a] ϕ (equations (5.24) and (5.26)):

$$\bar{\phi}_i(\mathbf{k}) = h^2 \phi_i^{(2)}(\mathbf{b}_i(\mathbf{k})) + O(h^3), \tag{B.4}$$

where

$$\phi_i^{(2)}(\mathbf{b}_i(\mathbf{k})) = \left[2 f_i^* c_s^{-2} \left(p(\mathbf{b}_i(\mathbf{k})) - c_s^{-2} \nu \mathbf{c}_i \cdot \nabla \mathbf{u}_B(\mathbf{b}_i(\mathbf{k})) \cdot \mathbf{c}_i\right)\right] \mathbf{c}_i. \tag{B.5}$$

Those expressions, for different i, are composed by evaluating a *macroscopic* function ζ (like p or $\mathbf{S}^\nu_{\alpha,\beta}$) at the boundary point $\mathbf{b}_i(\mathbf{k})$ related to the couple $(\mathbf{k}, i) \in B$, with particular weights. To generalize the notations, we introduce the vector $\mathbf{a}^{\zeta,i} = (a_x^{\zeta,i}, a_y^{\zeta,i})$ which indicates the two-dimensional weight multiplying the function ζ in ϕ_i:

$$\phi_i^{(2)}(\mathbf{b}) = \sum_\zeta \mathbf{a}^{\zeta,i} \zeta(\mathbf{b}). \tag{B.6}$$

Example B.1. *Since* $\phi_1^{(2)} = \left(\frac{2}{3}p - \mathbf{S}^\nu_{xx}, 0\right)$, *we have*

$$\mathbf{a}^{p,1} = \left(\frac{2}{3}, 0\right), \quad \mathbf{a}^{\mathbf{S}^\nu_{xx},1} = (-1, 0).$$

Grid-interface projections

For the boundary set $B(\Gamma)$ defined in (B.2), we consider the projections

$$\Pi_{\mathcal{G}} : B(\Gamma) \to \mathcal{G}(h), \quad \Pi_{\mathbb{V}} : B(\Gamma) \to \{1, \ldots, 8\}. \tag{B.7}$$

We indicate with $X(\Gamma) = \Pi_{\mathcal{G}}(B(\Gamma))$ the set of boundary *grid points*.
To have a more direct notation for the objects under consideration, we also define a map on the set $X(\Gamma)$, which gives all the outgoing directions on a boundary node $\mathbf{k} \in X$:

$$\begin{array}{rccl} \mathbb{P}: & X(\Gamma) & \to & \mathcal{P}(\{1, \ldots, 8\}) \\ & \mathbf{k} & \mapsto & \Pi_{\mathbb{V}}\left(\Pi_{\mathcal{G}}^{-1}(\mathbf{k})\right). \end{array} \tag{B.8}$$

This map is obviously strictly related to the grid and the shape of the boundary, as the set B is.

[a] However, without additional effort, we can consider a statement similar to theorem 5.1, considering the second order coefficient of the *original* ME (section B.3.4)

$$\phi_i^{(2)} = 2 f_i^* c_s^{-2} \left(p + \frac{c_s^{-2}}{2}\left(|\mathbf{c}_i \cdot \mathbf{u}_B|^2 - c_s^2 \mathbf{u}_B^2\right) - c_s^{-2} \nu \mathbf{c}_i \cdot \nabla \mathbf{u}_B \cdot \mathbf{c}_i\right) \mathbf{c}_i. \tag{B.3}$$

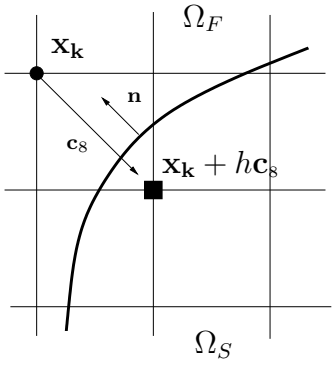


Figure B.1: Sketch of a boundary node (circle) in the fluid domain Ω_F, close to the interface Γ. The couple $(\mathbf{k}, 8) \in B$, being the point $\mathbf{x_k} + h\mathbf{c}_8 \in \Omega_S$.The vector $\mathbf{n}$ is the outgoing (with respect to the solid) normal to the interface.

B.2 Straight boundaries

First, we consider theorem 5.1 in the special case of *straight boundaries*. Performing the proof will help in understanding the way to proceed in more general cases.
We assume the interface Γ to be a straight line, with constant inclination $\theta \in [0, 2\pi)$. Without loss of generality, we consider (figure B.2) $\Omega = [0, 1] \times [0, \tan\theta]$ and the solid domain

$$\Omega_S = \{(x, y) \in \Omega \mid y < x \tan\theta\}. \tag{B.9}$$

In other words, the boundary (named θ-*boundary* in the future) is inclined by θ with respect to the x axis. In this particular situation, the outgoing normal to the interface is $\mathbf{n} = (-\sin\theta, \cos\theta)$.
Actually, we also assume $0 \leq \theta \leq \frac{\pi}{4}$ (note that the case $\theta = 0$ has been already solved in the example of flat boundaries in chapter 5, equation (5.25)). Otherwise, a similar proof can be repeated rotating the reference system, i.e. exchanging the role of x and y or of the discrete links in the following formulas.

Frequency of appearance

The structure of $\mathbb{P}(\mathbf{k})$ (outgoing direction in the node $\mathbf{k}$) can be described through some algebraic relations. Let us consider a node $\mathbf{k} = (k_x, k_y)$. From

$$\mathbf{x_k} \in \Omega_F \cup \Gamma \iff k_y \geq \tan\theta k_x$$

and

$$\begin{aligned} i \in \mathbb{P}(\mathbf{k}) &\Leftrightarrow \mathbf{x_k} + h\mathbf{c}_i \in \Omega_S \\ &\Leftrightarrow hk_y + hc_{iy} < \tan\theta(hk_x + hc_{ix}) \\ &\Leftrightarrow hk_y < \tan\theta hk_x + h(\tan\theta c_{ix} - c_{iy}), \end{aligned}$$

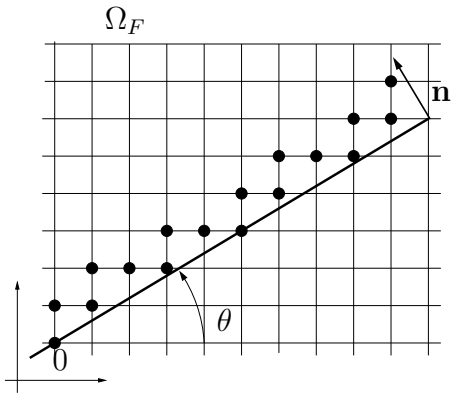


Figure B.2: Boundary points for a straight boundary with inclination θ. Here $\tan\theta = \frac{3}{5}$. According to (B.9), the fluid region is on the top. $\mathbf{n}$ is the solid-outgoing normal.

we conclude

$$i \in \mathbb{P}(\mathbf{k}) \Leftrightarrow 0 \leq k_y - \tan\theta k_x < \tan\theta c_{ix} - c_{iy}. \tag{B.10}$$

Inequality (B.10) has solutions only if the right hand side is greater than zero. We call **boundary crossing link** any discrete velocity $\mathbf{c}_i$ such that $\tan\theta c_{ix} - c_{iy} > 0$. To fix the ideas, we look at the two extreme situations.
If $\theta = 0$, relation (B.10) reduces to

$$i \in \mathbb{P}(\mathbf{k}) \Leftrightarrow 0 \leq k_y < -c_{iy} \ (k_y \geq 0). \tag{B.11}$$

In this case the boundary points (see figure B.3, left) are the nodes $\mathbf{k}$ such that $k_y = 0$. From the condition (B.11), we have $c_{iy} < 0$, i.e. $i = 6, 7, 8$. If $\theta = \dfrac{\pi}{4}$ (figure B.3, right), the condition is $0 < k_y - k_x < c_{ix} - c_{iy}$. It is satisfied by $i = 1, 7$ and $k_x = k_y$, or by $i = 8$ and $k_y = k_x, k_x + 1$. Notice that there are two couples with the same k_x, containing $i = 8$, and $\tan\theta c_{8x} - c_{8y} = 2$, while only one with $i = 1$ or $i = 7$ and $\tan\theta c_{1x} - c_{1y} = \tan\theta c_{7x} - c_{7y} = 1$. In these examples the quantity $c_{ix}\tan\theta - c_{iy}$ estimates how many k_y satisfy the inequality for a fixed k_x. This suggests a further characterization of the θ-boundary.

Definition B.1 (Frequency of appearance). *If the interface is a θ-boundary, we define the* frequency of appearance *of the link i in the set $\mathbb{P}(\mathbf{k})$ as*

$$m_{\theta,i} \equiv \max\left(\tan\theta c_{ix} - c_{iy}, 0\right). \tag{B.12}$$

As remarked in (B.10),

$$i \in \mathbb{P}(\mathbf{k}) \Leftrightarrow m_{\theta,i} > 0. \tag{B.13}$$

Within the restriction $0 \leq \theta \leq \frac{\pi}{4}$, we have

$$\tan\theta c_{ix} - c_{iy} > 0 \Leftrightarrow i \in \{1, 6, 7, 8\}.$$

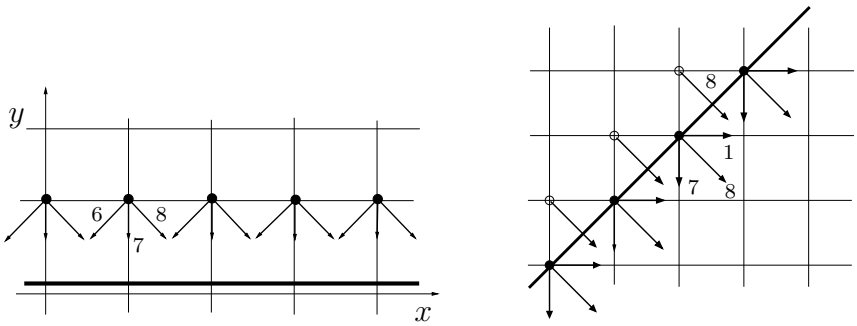


Figure B.3: Sketch of the boundary direction for the cases $\theta = 0$, $\theta = \frac{\pi}{4}$. In the second example, for a fixed k_x, the link $i = 8$ appears twice (nodes $\circ$ and $\bullet$), with respect to the directions $i = 1, 7$ (nodes $\bullet$).

Remark. Since $m_{\theta,i} \propto (\mathbf{c}_i \cdot \mathbf{n})$, $\mathbf{c}_i$ has to be an *outgoing* direction. Such condition has been already announced. However, the function $m_{\theta,i}$ contains more information. In some sense, $m_{\theta,i}$ *counts* the boundary couples. This concept will be formalized in general in the following part. Through $m_{\theta,i}$ a set of properties relating the macroscopical weights $\mathbf{a}^{\zeta,i}$ to the integral (5.2) can be proven.

Lemma B.1. *Let us define*

$$\mathbf{W}^{\theta}(\zeta) = \sum_i m_{\theta,i} \begin{pmatrix} a_x^{i,\zeta} \\ a_y^{i,\zeta} \end{pmatrix}. \tag{B.14}$$

It holds

$$\begin{aligned} &\mathbf{W}^{\theta}(p) = \begin{pmatrix} \tan\theta \\ -1 \end{pmatrix}, \mathbf{W}^{\theta}(\mathbf{S}_{xx}^{\nu}) = \begin{pmatrix} -\tan\theta \\ 0 \end{pmatrix}, \\ &\mathbf{W}^{\theta}(\mathbf{S}_{xy}^{\nu}) = \begin{pmatrix} 1 \\ -\tan\theta \end{pmatrix}, \ \mathbf{W}^{\theta}(\mathbf{S}_{yy}^{\nu}) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \end{aligned} \tag{B.15}$$

Proof. We will proof only the relation for $W_x^{\theta}(p)$ (others are similar). Direction by direction, we have

$$\begin{aligned} \mathbf{a}^{p,1} &= \left(\frac{2}{3}, 0\right) = -\mathbf{a}^{p,5}, \ \mathbf{a}^{p,2} = \left(\frac{1}{6}, \frac{1}{6}\right) = -\mathbf{a}^{p,6}, \\ \mathbf{a}^{p,3} &= \left(0, \frac{2}{3}\right) = -\mathbf{a}^{p,7}, \ \mathbf{a}^{p,4} = \left(-\frac{1}{6}, \frac{1}{6}\right) = -\mathbf{a}^{p,8}. \end{aligned}$$

Hence,

$$\begin{aligned}
&\sum_i a_x^{p,i} \max\left(\tan\theta c_{ix} - c_{iy}, 0\right) = \\
= \;& \frac{2}{3}\max\left(\tan\theta, 0\right) + \frac{1}{6}\max\left(\tan\theta - 1, 0\right) - \frac{1}{6}\max\left(-\tan\theta - 1, 0\right) - \\
- \;& \frac{2}{3}\max\left(-\tan\theta, 0\right) - \frac{1}{6}\max\left(-\tan\theta + 1, 0\right) + \frac{1}{6}\max\left(\tan\theta + 1, 0\right) = \\
= \;& \frac{2}{3}\tan\theta + \frac{1}{6}\left(\tan\theta - 1\right) + \frac{1}{6}\left(\tan\theta + 1\right) = \tan\theta.
\end{aligned}$$

□

A direct consequence is the following

Lemma B.2. *Let* $\mathbf{b} \in \Gamma$. *It holds*

$$\sum_\zeta \mathbf{W}^\theta(\zeta)\zeta(\mathbf{b}) = \frac{1}{\cos\theta}\mathbf{n}(\mathbf{b}) \cdot \left(-p(\mathbf{b}) + \mathbf{S}^\nu(\mathbf{b})\right). \tag{B.16}$$

The last relation expresses a connection between the *discrete sum* of the ME ϕ and the *continuous integral* of the hydrodynamical fields. Furthermore, it serves to identify where the integral measure come from. Heuristically, let us consider an integral

$$\int_{\Delta\Gamma} (-p + \mathbf{S}^\nu) \cdot \mathbf{n} d\gamma$$

which corresponds to the force acting on a subset $\Delta\Gamma$ of a θ-inclined interface. The increment $d\gamma$ can be expressed as

$$d\gamma = d\gamma_x \frac{1}{\cos\theta} \tag{B.17}$$

being $d\gamma_x$ the increment on the x-axis. In case of curved boundary, equation (B.17) would be a first order approximation (in h) of the increment.

Remark. The particular dependence on the abscissae comes from the restriction $0 \leq \theta < \frac{\pi}{4}$. In general, the axis which spans the angle θ with the interface is involved.

B.2.1 Averaged momentum exchanged

For a point $\mathbf{b}_0 \in \Gamma$ and a parameter h_c, we set

$$I(\mathbf{b}_0, h_c) = \left\{ \mathbf{x} \in \Gamma \mid \|x - \mathbf{b}_{0x}\| \leq \frac{h_c}{2} \right\}. \tag{B.18}$$

It is the subset of Γ whose projection on the x-axis is an $\frac{h_c}{2}$-neighborhood of $\mathbf{b}_{0x}$.

For this subset of the interface, we consider the set

$$B(\mathbf{b}_0, h_c) := \{(\mathbf{k}, i) \in B(\Gamma) \mid \mathbf{b}_i(\mathbf{k}) \in I(\mathbf{b}_0, h_c)\} \subset B(\Gamma),$$

of the boundary couples which identify a point in $I(\mathbf{b}_0, h_c)$, and the set

$$X(\mathbf{b}_0, h_c) = \Pi_{\mathcal{G}}\left(B(\mathbf{b}_0, h_c)\right),$$

of boundary grid nodes exchanging momentum with the subset $I(\mathbf{b}_0, h_c)$.
According to the statement of theorem 5.1, we need a coarse parameter $h_c(h)$ such that

$$h \in o(h_c).$$

For example, we could consider

$$h_c = h^\alpha, \quad 0 \leq \alpha < 1.$$

In what follows, we define a coarse grid as

$$h_c(h) = L(h)h, \tag{B.19}$$

$L(h)$ being the number of finer grid nodes belonging to a coarse grid step. Since $h \in o(h_c)$, we have

$$\lim_{h \to 0} L(h) = +\infty\,. \tag{B.20}$$

The idea is to prove a set of estimates regarding the boundary set and the momentum exchange which leads to the continuous integral after taking the average over several grid nodes. To do this, we relate the boundary set to the shape of the boundary, through the frequency of appearance $m_{\theta,i}$.
Formally, introducing the *characteristic function* of a set A:

$$\chi_A(x) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases},$$

we have

Lemma B.3.

$$\frac{1}{L} \sum_{\mathbf{k} \in X(\mathbf{b}_0, Lh)} \chi_{\mathbb{P}(\mathbf{k})}(i) = m_{\theta,i} + O\left(\frac{1}{L}\right).$$

Thus, $m_{\theta,i}$ represents *asymptotically* the *number of times per unity of length* a link i appears as outgoing direction in a set of points of length L on the x-axis.

Proof. The sum

$$\sum_{\mathbf{k} \in X(\mathbf{b}_0, Lh)} \chi_{\mathbb{P}(\mathbf{k})}(i)$$

expresses the number of times the boundary is crossed by a link $\mathbf{c}_i$ outgoing from nodes $\mathbf{k} \in X(\mathbf{b}_0, Lh)$.
We restrict to $0 \leq \tan\theta \leq 1$. Moreover, for simplicity we can shift the axis in order to have $\mathbf{b}_0 = h\dfrac{L}{2}$, i.e.

$$\forall \mathbf{k} \in X(\mathbf{b}_0, h_c) : \ 0 \leq k_x \leq L-1 \,. \tag{B.21}$$

Of course, the links $i = 2, 3, 4, 5$ satisfy trivially the property, holding $m_{\theta,i} = 0$. Now, let us consider the directions such that $m_{\theta,i} > 0$. Calling $\mathbf{k} = (k_x, k_y)$, the condition $\chi_{\mathbb{P}(\mathbf{k})}(i) = 1$ means

$$k_x \tan\theta \leq k_y < (k_x + c_{ix}) \tan\theta - c_{iy}. \tag{B.22}$$

Since $k_x, k_y \in \mathbb{Z}$, we can count the integer solutions of inequality (B.22). Namely, for fixed $k_x \in \{0, \ldots, L-1\}$ and $\mathbf{c}_i \in \mathbb{V}$, we have

$$\lceil (k_x + c_{ix}) \tan\theta - c_{iy} \rceil - \lceil k_x \tan\theta \rceil \tag{B.23}$$

possible choices for k_y, such that $\chi_{\mathbb{P}(\mathbf{k})}(i) = 1$ ($\mathbf{k}$ is a boundary node and $\mathbf{c}_i$ is an outgoing direction). In equation (B.23), $\lceil\ \rceil$ denotes the *upper integer part*. For $\alpha \in \mathbb{R}$:

$$\lceil \alpha \rceil = \min_{n \in \mathbb{Z}} \{ n \mid n \geq \alpha \}. \tag{B.24}$$

So that, summing up for $k_x \leq L-1$,

$$\sum_{\mathbf{k} \in X(\mathbf{b}_0, Lh)} \chi_{\mathbb{P}(\mathbf{k})}(i) = \sum_{k_x=0}^{L-1} \big(\lceil (k_x + c_{ix}) \tan\theta - c_{iy} \rceil - \lceil k_x \tan\theta \rceil \big). \tag{B.25}$$

To complete the proof we use some simple arithmetic relations.

Sub-lemma B.3.1. *Let $\alpha \in \mathbb{R}$, $n \in \mathbb{Z}$. Then*

$$(\mathbf{A}) \lceil \alpha + n \rceil = \lceil \alpha \rceil + n$$
$$(\mathbf{B}) \lceil \alpha \rceil = \alpha + O(1).$$

(Proof is omitted)
Since $c_{iy} \in \mathbb{Z}$, using sublemma B.3.1, part (A), we can write sum (B.25) as

$$\sum_{k_x=0}^{L-1} \big(\lceil (k_x + c_{ix}) \tan\theta - c_{iy} \rceil - \lceil k_x \tan\theta \rceil \big) =$$
$$= -c_{iy} L + \sum_{k_x=c_{ix}}^{L-1+c_{ix}} \lceil k_x \tan\theta \rceil - \sum_{k_x=0}^{L-1} \lceil k_x \tan\theta \rceil. \tag{B.26}$$

We distinguish between three different situations.
(a): $c_{ix} = 0$**.** In this case, the two sums on the right hand side of equation (B.26) are equal, i.e.

$$\sum_{\mathbf{k}\in X(\mathbf{b}_0,Lh)} \chi_{\mathbb{P}(\mathbf{k})}(i) = -c_{iy}L. \tag{B.27}$$

(b): $c_{ix} = 1$**.** In the expression

$$\sum_{\mathbf{k}\in X(\mathbf{b}_0,Lh)} \chi_{\mathbb{P}(\mathbf{k})}(i) = -c_{iy}L + \sum_{k_x=1}^{L} \lceil k_x \tan\theta \rceil - \sum_{k_x=0}^{L-1} \lceil k_x \tan\theta \rceil$$

the two sums over k_x can be grouped in a single one, which has telescopic properties, i.e. the addenda balance themselves. Only the lowest one, for $k_x = 0$ (which vanishes), and the highest one, for $k_x = L$, remains, giving

$$\sum_{\mathbf{k}\in X(\mathbf{b}_0,Lh)} \chi_{\mathbb{P}(\mathbf{k})}(i) = -c_{iy}L + c_{ix}\lceil L \tan\theta \rceil = -c_{iy}L + c_{ix}L\tan\theta + O(1). \tag{B.28}$$

(c): $c_{ix} = -1$**.** In the last case, we have

$$\sum_{\mathbf{k}\in X(\mathbf{b}_0,Lh)} \chi_{\mathbb{P}(\mathbf{k})}(i) = -c_{iy}L + \sum_{k_x=-1}^{L-2} \lceil k_x \tan\theta \rceil - \sum_{k_x=0}^{L-1} \lceil k_x \tan\theta \rceil = \\ - c_{iy}L - \lceil \tan\theta \rceil - \lceil (L-1)\tan\theta \rceil \tag{B.29}$$

(it is again a telescopic sum). Applying sub-lemma B.3.1, part (B)

$$-\lceil \tan\theta \rceil - \lceil (L-1)\tan\theta \rceil = -\tan\theta + O(1) - (L-1)\tan\theta + O(1),$$

so that

$$\sum_{\mathbf{k}\in X(\mathbf{b}_0,Lh)} \chi_{\mathbb{P}(\mathbf{k})}(i) = -c_{iy}L + c_{ix}L\tan\theta + O(1). \tag{B.30}$$

Summarizing, for $c_{ix} \in \{-1, 0, 1\}$ it holds

$$\sum_{\mathbf{b}\in X(\mathbf{b}_0,Lh)} \chi_{\mathbb{P}(\mathbf{k})}(i) = Lm_{\theta,i} + O(1). \tag{B.31}$$

Dividing by L the proof of lemma B.3 is complete. □

As a consequence, we obtain an estimate for the number of points in the boundary set.

Corollary B.1.

$$\frac{|X(\mathbf{b}_0, Lh)|}{L} = 1 + \tan\theta + O\left(\frac{1}{L}\right).$$

Proof. Within the restriction $0 \leq \tan\theta \leq 1$, we have that

$$\forall \mathbf{k} \in X(\mathbf{b}_0, Lh) : \ \chi_{\mathbb{P}(\mathbf{k})}(8) = 1,$$

i.e. the link $\mathbf{c}_8$ is outgoing in each boundary node. Thus,

$$|X(\mathbf{b}_0, Lh)| = \sum_{\mathbf{b} \in X(\mathbf{b}_0, Lh)} \chi_{\mathbb{P}(\mathbf{k})}(8).$$

Since $m_{\theta,8} = 1 + \tan\theta$, the corollary is just a particular case of the previous lemma B.3. □

Coarse grids

Lemmas B.2 and B.3 motivate the origin of the *coarser* grid. In fact, considering a discretization parameter $h_c(h) = L(h)h$ such that (equation (B.20))

$$\lim_{h \to 0} L(h) = +\infty,$$

for small h we can translate the previous properties in asymptotic estimates. Let $\mathbf{b}_0 \in \Gamma$ be given. We define the coarse interval

$$I(\mathbf{b}_0, h_c) = \left\{ (x, y) \in \Gamma \mid \|x - \mathbf{b}_{0x}\| \leq \frac{h_c}{2} \right\} \tag{B.32}$$

(considering the distance along the x-axis) and focus on the subintegral[b]

$$\mathfrak{I}(\mathbf{b}_0, h_c) = \int_{I(\mathbf{b}_0, h_c)} \mathbf{n} \cdot (-p + \mathbf{S}^\nu)\, d\gamma := \int_{I(\mathbf{b}_0, h_c)} \mathbf{Z} d\gamma. \tag{B.33}$$

[b]In practice, the parameter h_c can be used to define a coarse partition of the interface

$$\{\mathbf{b}_0^r \mid r = 1, \dots R\} \subset \Gamma$$

such that

$$\Gamma = \bigcup_{r=1}^{R} I(\mathbf{b}_0^r, h_c).$$

Consequently, the original integral along Γ can be divided in *sub-integrals*

$$\mathfrak{I} = \sum_{r=1}^{R} \mathfrak{I}(\mathbf{b}_0^r, h_c),$$

where

$$\mathfrak{I}(\mathbf{b}_0^r, h_c) = \int_{I(\mathbf{b}_0^r, h_c)} \mathbf{n} \cdot (-p + \mathbf{S}^\nu)\, d\gamma := \int_{I(\mathbf{b}_0^r, h_c)} \mathbf{Z} d\gamma.$$

In the theorem, we focus on a single coarse interval.

Using a Taylor expansion we have

$$\mathcal{I}(\mathbf{b}_0, h_c) = \frac{h_c}{\cos\theta}\mathbf{Z}(\mathbf{b}_0) + O(h_c^2). \tag{B.34}$$

At the same time,

$$\Phi(\mathbf{b}_0, h_c) = \sum_{(\mathbf{k},i)\in B(\mathbf{b}_0,h_c)} \bar{\phi}_i(\mathbf{k}) = \sum_{i=1}^{8} \left(\sum_{\mathbf{k}\in X(\mathbf{b}_0,h_c)} \chi_{\mathbb{P}(\mathbf{k})}(i)\bar{\phi}_i(\mathbf{k}) \right). \tag{B.35}$$

Inserting the prediction (5.24) for (the corrected) $\bar{\phi}_i$ and its Taylor expansion around $\mathbf{b}_0$, the previous formula becomes

$$\Phi(\mathbf{b}_0, h_c) = h^2 \left(\sum_{i=1}^{8} \bar{\phi}_i^{(2)}(\mathbf{b}_0) \sum_{\mathbf{k}\in X(\mathbf{b}_0,h_c)} \chi_{\mathbb{P}(\mathbf{k})}(i) + O(h_c) \right). \tag{B.36}$$

Remark. We average over a coarse grid cell, which contains L nodes of the original grid along the x-axis. Moreover, since the quantities in which we are interested are of order h^2 in $\overline{\Phi}(\mathbf{b}_0, h_c)$, we consider an averaged and rescaled sum[c]

$$\overline{\Phi}(\mathbf{b}_0, h_c) = \frac{1}{L}\frac{\Phi(\mathbf{b}_0, h_c)}{h^2}. \tag{B.37}$$

We have

$$\begin{aligned} \overline{\Phi}(\mathbf{b}_0, h_c) &= \sum_{i=1}^{8} \phi_i^{(2)}(\mathbf{b}_0) \sum_{\mathbf{k}\in X(\mathbf{b}_0,h_c)} \left(\frac{\chi_{\mathbb{P}(\mathbf{k})}(i)}{L} + O(L^{-1}h_c) \right) = \\ &= \sum_{i=1}^{8} m_{\theta,i}\phi_i^{(2)}(\mathbf{b}_0) + O(\big|X(\mathbf{b}_0, h_c)\big|L^{-1}h_c). \end{aligned}$$

Estimating the right hand side using lemma B.3 and corollary B.1 we find

$$\overline{\Phi}(\mathbf{b}_0, h_c) = \frac{1}{\cos\theta}\mathbf{Z}(\mathbf{b}_0) + O\left(\frac{h}{h_c}\right) + O(h_c). \tag{B.38}$$

Being $L = \dfrac{h_c}{h}$, the relation between (B.34) and (B.36) is now

$$\left| \frac{\mathcal{I}(\mathbf{b}_0, h_c)}{h_c} - \overline{\Phi}(\mathbf{b}_0, h_c) \right| \in O\left(h_c + \frac{h}{h_c}\right). \tag{B.39}$$

[c] In the case of a two-dimensional interface, the relation between the points of fine and coarse grid involves $\left(\frac{h}{h_c}\right)^2$. Equation (B.37) contains in general a factor $\left(\frac{h}{h_c}\right)^d$.

In the enunciate of theorem 5.1, we deal with the coarse set

$$\Gamma_0(\mathbf{b}_0, h_c) = \left\{ \mathbf{b} \in \Gamma \mid \left\| \mathbf{b} - \mathbf{b}_0 \right\| < \frac{h_c}{2} \right\}$$

which is slightly different from the $I(\mathbf{b}_0, h_c)$ considered in equation (B.39). However, for straight interfaces these two sets are asymptotically equal, in the sense that their lengths are of the same order, and the number of nodes of difference (belonging to one but not to another) goes to zero as L goes to infinity. Hence, the validity of the argument is not affected, i.e. estimate (B.39) proves the theorem for straight boundaries.

B.3 Generalization for curved boundaries

To generalize the proof, we follow the path which yielded equation (B.39), investigating the modification we need in a more general case.

B.3.1 Interface as smooth function of x

Let us assume that the interface Γ is described by the graph of a *smooth* function:

$$\exists g \text{ smooth } : \ \Gamma = \{(x, y) \in \Omega \mid y = g(x)\}, \tag{B.40}$$

In this case, the fluid and solid domains are identified by

$$\Omega_F = \{(x, y) \in \Omega \mid y > g(x)\}, \ \Omega_S = \{(x, y) \in \Omega \mid y < g(x)\} \tag{B.41}$$

(compare with equation (B.9)).

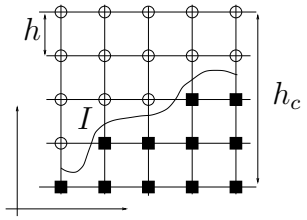


Figure B.4: Example of a coarse cell $h_c \times h_c$ in a lattice of size h. The squares represent the solid node, the circles the fluid ones. The set $I = I(\mathbf{b}_0, h_c)$ (bold line) is the portion of interface contained in the cell.

Assuming to have chosen a point $\mathbf{b}_0 \in \Gamma$ and a coarse parameter h_c, defining the related coarse interval $I(\mathbf{b}_0, h_c)$ as in equation (B.32), we focus on a single cell $h_c \times h_c$ (figure B.4).

B.3.2 Averaged inclination

We look for a geometric condition on a couple $(\mathbf{k}, i)$ belonging to the set $B(\mathbf{b}_0, h_c)$. According to (B.41),

$$(\mathbf{k}, i) \in B(\mathbf{b}_0, h_c) \iff \begin{array}{l} hk_y \geq g(hk_x) \\ h(k_y + c_{iy}) < g(hk_x + hc_{ix}). \end{array} \tag{B.42}$$

If $g(x)$ is smooth, we can write (fundamental theorem of calculus)

$$g(hk_x + hc_{ix}) = g(hk_x) + \int_{hk_x}^{hk_x + hc_{ix}} g'(s)ds. \tag{B.43}$$

Using $s = h(k_x + c_{ix}\sigma)$, we have

$$g(hk_x + hc_{ix}) = g(hk_x) + hc_{ix} \int_0^1 g'(hk_x + hc_{ix}\sigma)d\sigma \tag{B.44}$$

which allows to rewrite (B.42) as

$$(\mathbf{k}, i) \in B(\mathbf{b}_0, h_c) \iff 0 \leq k_y - \frac{g(hk_x)}{h} < c_{ix} \int_0^1 g'(hk_x + hc_{ix}\sigma)d\sigma - c_{iy}. \tag{B.45}$$

In view of (B.45), having fixed the node $\mathbf{k}$, the direction $\mathbf{c}_i$ intersects the interface iff

$$c_{ix} \int_0^1 g'(hk_x + hc_{ix}\sigma)d\sigma - c_{iy} > 0. \tag{B.46}$$

Different interpretations of (B.45) are possible. In words, equation (B.46) expresses the fact that the vector $\mathbf{c}_i$ from the node $\mathbf{k}$ is a **boundary crossing link** in terms of the average of the function $g'(x)$ in the range $h[k_x, k_x + c_{ix}]$.
Let us consider a boundary couple $(\mathbf{k}, i) \in B(\mathbf{b}_0, h_c)$. We can denote with

$$\langle g' \rangle (\mathbf{b}_i(\mathbf{k})) = \int_0^1 g'(hk_x + hc_{ix}\sigma)d\sigma. \tag{B.47}$$

the average of $g'(x)$ in the x-cell containing the intersection $\mathbf{b}_i(\mathbf{k})$ (figure B.5). As well, we can introduce the angle $\langle\theta\rangle \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ such that

$$\tan\langle\theta\rangle(\mathbf{b}_i(\mathbf{k})) = \langle g' \rangle (\mathbf{b}_i(\mathbf{k})), \tag{B.48}$$

i.e. the *average inclination* of the interface in $h[k_x, k_x + c_{ix}]$.

The definition of $\langle\theta\rangle$ can be extended to any point of the interface[d], i.e.

$$\forall \mathbf{b} \in \Gamma : \ \langle\theta\rangle(\mathbf{b}) = \text{ average inclination in the } x\text{-cell containing } \mathbf{b}.$$

In other words, we are looking at the interface Γ as a piecewise straight line, with inclinations equal to the average on the corresponding cell (figure B.5).
Remark. Condition (B.45) is purely geometric, and related to a property of the grid and the interface. The averaged inclination (B.49) will be used to transport the results proven for straight boundaries also in the case of a curved interface.

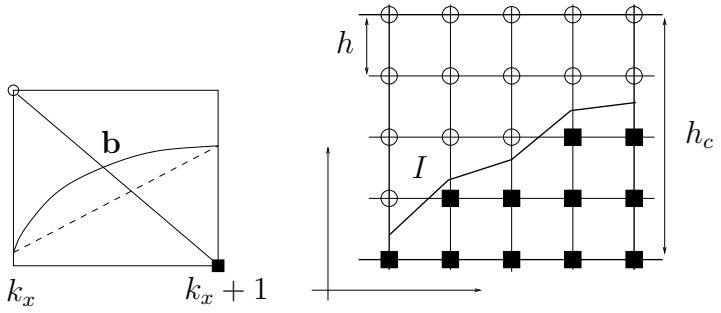


Figure B.5: **Left:** Averaged inclination of the interface between the abscissae k_x and $k_x + 1$. Formula (B.47) expresses the averaged inclination of the interface, over the interval of size h containing the intersection lattice-boundary $\mathbf{b}$. The dashed line is a straight boundary with inclination equal to the averaged one. **Right:** In a $h_c \times h_c$ cell (figure B.4), using (B.47) we approximate the interface with a piecewise straight approximation.

Generalized frequency of appearance

Relation (B.45) can be rewritten as

$$\forall (\mathbf{k}, i) \in B(\mathbf{b}_0, h_c) : \ c_{ix} \tan\langle\theta\rangle(\mathbf{b}_i(\mathbf{k})) - c_{iy} > 0. \tag{B.50}$$

It generalizes the previous relation (B.10) in case of curved boundaries, where the inclination is no longer constant.

[d]Formally, any $\mathbf{b} \in \Gamma$ is contained in one and only one x-cell:

$$\exists ! \, \mathbf{k}(\mathbf{b}) \in \mathcal{G}(h) : \ \mathbf{b} \in h[k_x(\mathbf{b}), k_x(\mathbf{b}) + 1) \times h[k_y(\mathbf{b}), k_y(\mathbf{b}) + 1),$$

and we can define

$$\langle\theta\rangle(\mathbf{b}) = \tan^{-1}\left(\int_0^1 g'(hk(\mathbf{b})_x + h\sigma)d\sigma\right). \tag{B.49}$$

In analogy with the previous results, we introduce the *generalized frequency of appearance*

$$\langle m\rangle_{(\mathbf{k},i)} = \max\left\{c_{ix}\int_0^1 g'(hk_x + hc_{ix}\sigma)d\sigma - c_{iy}, 0\right\}. \tag{B.51}$$

In case of straight boundary with constant inclination θ, $g'(x) = \tan\theta$, $\langle m\rangle$ does not depend on $\mathbf{k}$ and it coincides with the frequency of appearance previously defined.

Lemma B.4.

$$\forall \mathbf{k} \in X(\mathbf{b}_0, h_c) : \langle m\rangle_{(\mathbf{k},i)} = m_{\langle\theta\rangle(\mathbf{b}_0),i} + O(h_c).$$

Proof. Let us take $(\mathbf{k}, i) \in \mathcal{G}(h) \times \{1, \ldots, 8\}$. We have

$$\langle m\rangle_{(\mathbf{k},i)} = \begin{cases} c_{ix}\tan\langle\theta\rangle(\mathbf{b}_i(\mathbf{k})) - c_{iy} & (\mathbf{k}, i) \in B(\Gamma) \\ 0 & (\mathbf{k}, i) \notin B(\Gamma). \end{cases} \tag{B.52}$$

Taylor expanding g around $\mathbf{b}_{0x}$, for the points $\mathbf{b}_i(\mathbf{k})$ in the considered coarse cell, it holds

$$\tan\langle\theta\rangle(\mathbf{b}_i(\mathbf{k})) = \tan\langle\theta\rangle(\mathbf{b}_0) + O(h_c).$$

Hence,

$$\langle m\rangle_{(\mathbf{k},i)} = \begin{cases} c_{ix}\tan\langle\theta\rangle(\mathbf{b}_0) - c_{iy} + O(h_c) & (\mathbf{k}, i) \in B(\Gamma) \\ 0 & (\mathbf{k}, i) \notin B(\Gamma). \end{cases} \tag{B.53}$$

The straight frequency of appearance reads

$$m_{\langle\theta\rangle(\mathbf{b}_0),i} = \max\{c_{ix}\tan\langle\theta\rangle(\mathbf{b}_0) - c_{iy}, 0\}.$$

It might happen, that for $(\mathbf{k}, i) \in B(\mathbf{b}_0, h_c)$ we have $\langle m\rangle_{(\mathbf{k},i)} > 0$ but $m_{\langle\theta\rangle(\mathbf{b}_0),i} = 0$, or vice versa. In words, $(\mathbf{k}, i)$ is a boundary couple for Γ, but not for a straight approximation with inclination $\langle\theta\rangle(\mathbf{b}_0)$. However, in this case

$$\begin{aligned} &c_{ix}\tan\langle\theta\rangle(\mathbf{b}_0) \le 0 \\ &c_{ix}\tan\langle\theta\rangle(\mathbf{b}_0) + O(h_c) > 0, \end{aligned}$$

which imply

$$\langle m\rangle_{(\mathbf{k},i)},\ m_{\langle\theta\rangle(\mathbf{b}_0),i} \in O(h_c).$$

Hence,

$$\langle m\rangle_{(\mathbf{k},i)} = m_{\langle\theta\rangle(\mathbf{b}_0),i} + O(h_c).$$

□

Therefore, in the considered $h_c \times h_c$ cell, up to an error $O(h_c)$ we can consider the frequency of appearance of a straight interface with inclination equal to the average in the coarse cell (figure B.6).
Obviously, the boundary couples of a straight approximation of the interface do not coincide with the boundary couples of the original one. However, the lemma tells that this happens only when the frequency of appearance is of order $O(h_c)$. In other words the difference between the boundary set of the interface Γ and of its straight approximation is *small*, for small h_c. Formally,

Lemma B.5.

$$\frac{1}{L}\sum_{\mathbf{k}\in X(\mathbf{b}_0,Lh)} \chi_{\mathbb{P}(\mathbf{k})}(i) = m_{\langle\theta\rangle(\mathbf{b}_0),i} + O\left(\frac{1}{L}\right) + O(h_c).$$

Proof. Let $\mathbf{k} \in X(\mathbf{b}_0, Lh)$ and $\mathbf{c}_i \in \mathbb{V}$ be fixed. As before (lemma B.3), we estimate the sum counting, for each k_x, how many choices for k_y we have, such that $\mathbf{k} = (k_x, k_y) \in X(\mathbf{b}_0, Lh)$ and $i \in \mathbb{P}(\mathbf{k})$.
According to condition (B.42), it has to hold

$$\frac{g(hk_x)}{h} \leq k_y < \frac{g(hk_x + hc_{ix})}{h} - c_{iy}.$$

Hence, there are

$$\left\lceil \frac{g(hk_x + hc_{ix})}{h} - c_{iy} \right\rceil - \left\lceil \frac{g(hk_x)}{h} \right\rceil$$

possible solutions $k_y \in \mathbb{Z}$. For simplicity, since $h_c = Lh$, we assume (see lemma B.3)

$$\forall \mathbf{k} \in X(\mathbf{b}_0, Lh) : \ 0 \leq k_x \leq L-1$$

(it can be done shifting the interface along the x-axis).
We have

$$\begin{aligned}\sum_{\mathbf{k}\in X(\mathbf{b}_0,Lh)} \chi_{\mathbb{P}(\mathbf{k})}(i) &= \sum_{k_x=0}^{L-1} \left\lceil \frac{g(hk_x + hc_{ix})}{h} - c_{iy} \right\rceil - \left\lceil \frac{g(hk_x)}{h} \right\rceil = \\ &= -c_{iy}L + \sum_{k_x=0}^{L-1} \left\lceil \frac{g(hk_x + hc_{ix})}{h} \right\rceil - \left\lceil \frac{g(hk_x)}{h} \right\rceil.\end{aligned} \tag{B.54}$$

With an argument analogous to the one used to prove lemma B.3, we split the cases $c_{ix} = 0, 1, -1$.
If $c_{ix} = 0$, equation (B.54) reduces to

$$\sum_{\mathbf{k}\in X(\mathbf{b}_0,Lh)} \chi_{\mathbb{P}(\mathbf{k})}(i) = -c_{iy}L. \tag{B.55}$$

If $c_{ix} = 1$ we have a telescopic sum, which yields

$$\begin{aligned}\sum_{\mathbf{k}\in X(\mathbf{b}_0,Lh)} \chi_{\mathbb{P}(\mathbf{k})}(i) &= -c_{iy}L + \left\lceil \frac{g(Lh)}{h} \right\rceil - \left\lceil \frac{g(0)}{h} \right\rceil = \\ &= -c_{iy}L + \frac{g(Lh) - g(0)}{h} + O(1).\end{aligned} \tag{B.56}$$

From

$$g(Lh) - g(0) = \int_0^{Lh} g'(x)dx = Lh \int_0^1 g'(Lh\sigma)d\sigma,$$

using the Taylor expansion for $g'(x)$ we can estimate

$$\frac{g(Lh) - g(0)}{h} = L\left(\langle g' \rangle(\mathbf{b}_0) + O(h_c)\right),$$

which yields

$$\sum_{\mathbf{k}\in X(\mathbf{b}_0,Lh)} \chi_{\mathbb{P}(\mathbf{k})}(i) = Lm_{\langle\theta\rangle(\mathbf{b}_0),i} + O(Lh_c) + O(1). \tag{B.57}$$

The case $c_{ix} = -1$ is analogous, so that we can conclude

$$\forall c_{ix} \in \{0, 1, -1\} : \frac{1}{L} \sum_{\mathbf{k}\in X(\mathbf{b}_0,Lh)} \chi_{\mathbb{P}(\mathbf{k})}(i) = m_{\langle\theta\rangle(\mathbf{b}_0),i} + O(h_c) + O\left(\frac{1}{L}\right).$$

The result is analogous to lemma B.3, only using $\langle\theta\rangle(\mathbf{b}_0)$ instead of θ, and with an additional term $O(h_c)$. □

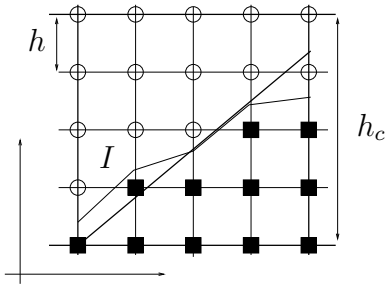


Figure B.6: From the subset of interface $I(\mathbf{b}_0, h_c) \subset \Gamma$ of figure B.4, we reduce to a straight interface with inclination equal to the average in the LB cell (of size h) containing $\mathbf{b}_0$ (bold line). Differences in the structure of the boundary node set of Γ and of the straight approximation appear only if the frequency of appearance $\langle m \rangle_{(\mathbf{k},i)}$ is small.

From discrete grid to continuous integral

Let us define

$$\mathbf{W}^{\langle\theta\rangle(\mathbf{b})}(\zeta) = \sum_i m_{\langle\theta\rangle(\mathbf{b}),i}\mathbf{a}^{i,\zeta}. \tag{B.58}$$

Applying lemma B.1, using the averaged angle $\langle\theta\rangle(\mathbf{b}_0)$ instead of the constant θ, it holds

$$\begin{aligned}
\mathbf{W}^{\langle\theta\rangle(\mathbf{b}_0)}(p) &= \begin{pmatrix} \tan\langle\theta\rangle(\mathbf{b}_0) \\ -1 \end{pmatrix} + O(h_c),\\
\mathbf{W}^{\langle\theta\rangle(\mathbf{b}_0)}(\mathbf{S}^\nu_{xx}) &= \begin{pmatrix} -\tan\langle\theta\rangle(\mathbf{b}_0) \\ 0 \end{pmatrix} + O(h_c),\\
\mathbf{W}^{\langle\theta\rangle(\mathbf{b}_0)}(\mathbf{S}^\nu_{xy}) &= \begin{pmatrix} 1 \\ -\tan\langle\theta\rangle(\mathbf{b}_0) \end{pmatrix} + O(h_c),\\
\mathbf{W}^{\langle\theta\rangle(\mathbf{b}_0)}(\mathbf{S}^\nu_{yy}) &= \begin{pmatrix} 0 \\ 1 \end{pmatrix} + O(h_c).
\end{aligned} \tag{B.59}$$

Note that, considering curved boundaries we have to use the averaged value on the h_c-cell instead of a constant inclination.

As a consequence, the relation with the continuous integral is stated in the following lemma.

Lemma B.6.

$$\sum_\zeta \mathbf{W}^{\langle\theta\rangle(\mathbf{b}_0)}(\zeta)\zeta(\mathbf{b}_0) = \frac{1}{\cos\theta(\mathbf{b}_0)}\mathbf{n}(\mathbf{b}_0)\cdot(-p(\mathbf{b}_0)\mathbf{I} + \mathbf{S}^\nu(\mathbf{b}_0)) + O(h_c). \tag{B.60}$$

Proof. From lemma B.2, using $\theta = \langle\theta\rangle(\mathbf{b}_0)$, it follows directly

$$\sum_\zeta \mathbf{W}^{\langle\theta\rangle(\mathbf{b}_0)}(\zeta)\zeta(\mathbf{b}_0) = \frac{1}{\cos\langle\theta\rangle(\mathbf{b}_0)}\mathbf{n}(\mathbf{b}_0)\cdot(-p(\mathbf{b}_0)\mathbf{I} + \mathbf{S}^\nu(\mathbf{b}_0)) + O(h_c). \tag{B.61}$$

Using a Taylor expansion for the function $g(x)$, we have also

$$\theta(\mathbf{b}_0) = \langle\theta\rangle(\mathbf{b}_0) + O(h_c),$$

which combined with (B.61) proves the statement. Note that the only difference with respect to the straight boundary case, is the additional error $O(h_c)$. □

Now, we have the ingredients to estimate the sum of momentum exchange.

$$\begin{aligned}
&\frac{1}{L}\sum_{(\mathbf{k},i)\in B(\mathbf{b}_0,h_c)}\phi_i^{(2)}(\mathbf{b}_i(\mathbf{k})) = \frac{1}{L}\sum_{k=1}^{8}\left(\sum_{\mathbf{k}\in X(\mathbf{b}_0,h_c)}\chi_{\mathbb{P}(\mathbf{k})}(i)\phi_i^{(2)}(\mathbf{b}_i(\mathbf{k}))\right) = \\
= &\frac{1}{L}\sum_{k=1}^{8}\left(\phi_i^{(2)}(\mathbf{b}_0)+O(h_c)\right)\sum_{\mathbf{k}\in X(\mathbf{b}_0,h_c)}\chi_{\mathbb{P}(\mathbf{k})}(i) = \\
= &\sum_{k=1}^{8}\phi_i^{(2)}(\mathbf{b}_0)\frac{1}{L}\sum_{\mathbf{k}\in X(\mathbf{b}_0,h_c)}\chi_{\mathbb{P}(\mathbf{k})}(i)+O\left(h_c\frac{|X(\mathbf{b}_0,h_c)|}{L}\right) = \\
= &\sum_{k=1}^{8}\phi_i^{(2)}(\mathbf{b}_0)m_{\langle\theta\rangle(\mathbf{b}_0),i}+O\left(\frac{1}{L}\right)+O\left(h_c\frac{|X(\mathbf{b}_0,h_c)|}{L}\right).
\end{aligned}$$

Regarding the last term, an estimate of the cardinality of $X(\mathbf{b}_0, h_c)$ follows from lemma B.5:

$$|X(\mathbf{b}_0, h_c)| \in O(L), \tag{B.62}$$

since it holds for the straight approximation of the boundary.

Condition (B.62) expresses indirectly a limit on the length of the interface in the considered coarse cell. If the interface has *finite length*, in each coarse cell the length goes to 0 with h_c, as well as the number of grid-boundary intersections. In practice, it might happen that the result is true only *eventually*, i.e. for h_c smaller than a threshold value $\overline{h}_c$ (figure B.7).

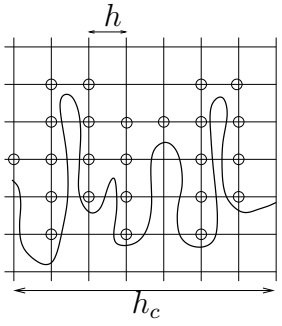


Figure B.7: In practice, to have a limit for the number of boundary nodes in a coarse interval, the interface should not be *too long* inside the considered coarse cell. If the interface is smooth and has finite length, decreasing with h_c, the number of intersection will be *eventually* bounded by the number L of nodes contained in a coarse grid step.

In conclusion, we can write

$$\begin{aligned}\frac{1}{L}\sum_{(\mathbf{k},i)\in B(\mathbf{b}_0,h_c)}\phi_i^{(2)}(\mathbf{b}_i(\mathbf{k})) &= \sum_{k=1}^{8}\phi_i^{(2)}(\mathbf{b}_0)m_{\langle\theta\rangle(\mathbf{b}_0),i} + O\left(\frac{1}{L}\right) + O\left(h_c\right) = \\ &= \frac{1}{\cos\langle\theta\rangle(\mathbf{b}_0)}\mathbf{Z}(\mathbf{b}_0) + O\left(\frac{1}{L}\right) + O(h_c),\end{aligned} \tag{B.63}$$

which combined with (B.33) yields again estimate (B.39), proving the statement.

B.3.3 Smooth interfaces

In general, assume the interface to be given by

$$\Gamma = \{(x,y)\in\Omega \mid \mathcal{Z}(x,y) = 0\},$$

$\mathcal{Z}\in\mathcal{F}(\Omega,\mathbb{R})$ being a smooth function. Assume that the curve $\mathcal{Z}$ does not have singularity points, i.e. $\nabla\mathcal{Z}\neq 0$.
Once we have chosen a point $\mathbf{b}_0\in\Gamma$ and the grid size h_c, if the gradient of $\mathcal{Z}$ does not vanish we can always assume to solve the equation

$$\mathcal{Z}(x,y) = 0$$

with respect to x or to y (Dini's theorem). In practice it could be necessary to choose a parameter h_c small enough.
Therefore, we can reduce to the previous situation, *locally* around $\mathbf{b}_0$. Expressing x as a function of y, it is enough to perform a rotation of the reference system and repeat the proof of theorem 5.1 as in section B.3.1.

B.3.4 Statement for the original MEA

Concerning the original MEA (5.19), to generalize (B.39) we need a suitable *limit*, as the integral (5.2) is for the corrected algorithm. Recalling the expression of the ME

$$\begin{aligned}\phi_i(\mathbf{k}) = \phi_i^{(0)} + h^2\bar{\phi}_i^{(2)} = -2c_s^{-2}f_i^*\mathbf{c}_i + \\ h^2 2f_i^*c_s^{-2}\left(p + \frac{c_s^{-2}}{2}\left(|\mathbf{c}_i\cdot\mathbf{u}_B|^2 - c_s^2|\mathbf{u}_B|^2\right) - c_s^{-2}\nu\mathbf{c}_i\cdot\nabla\mathbf{u}_B\cdot\mathbf{c}_i\right)\mathbf{c}_i\end{aligned} \tag{B.64}$$

(the functions $\phi^{(k)}$, p, $\mathbf{u}_B$ are evaluated at the boundary point $\mathbf{b}_i(\mathbf{k})$), the extension of lemma B.6 is straightforward. Explicitely, using the same notation

$$\phi_i^{(2)} = \sum_{\zeta}\mathbf{a}^{\zeta,i}\zeta_i, \tag{B.65}$$

where $\zeta = p, \mathbf{S}^{\nu}_{\alpha,\beta}, u_\alpha u_\beta$ and $\mathbf{a}^{\zeta,i}$ are the corresponding weights (equation (B.6)), from (B.65) we obtain (compare with (B.59))

$$\mathbf{W}^{\langle\theta\rangle}(u^2) = \begin{pmatrix} \tan\langle\theta\rangle \\ 0 \end{pmatrix}, \ \mathbf{W}^{\langle\theta\rangle}(uv) = \begin{pmatrix} -1 \\ \tan\langle\theta\rangle \end{pmatrix}, \ \mathbf{W}^{\langle\theta\rangle}(v^2) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}. \tag{B.66}$$

Additionally:

$$\sum_i 2 f_i^* m_{\langle\theta\rangle,i} \mathbf{c}_i = -c_s^2 \frac{1}{\cos\langle\theta\rangle} \mathbf{n}. \tag{B.67}$$

Lemma B.6 becomes

$$\sum_{\zeta} \mathbf{W}^{\langle\theta\rangle(\mathbf{b}_0)}(\zeta)\zeta(\mathbf{b}_0) = \\ = \frac{1}{\cos\langle\theta\rangle(\mathbf{b}_0)} \mathbf{n}(\mathbf{b}_0) \cdot (-p(\mathbf{b}_0)\mathbf{I} + \mathbf{S}^{\nu}(\mathbf{b}_0) - \mathbf{u}_B(\mathbf{b}_0) \otimes \mathbf{u}_B(\mathbf{b}_0)) + O(h_c). \tag{B.68}$$

In view of (B.68), we define a *generalized integral*

$$\begin{aligned} \mathcal{J}(\mathbf{b}_0, h_c) &= c_s^2 \int_{I(\mathbf{b}_0,h_c)} -\mathbf{n} d\gamma + h^2 \int_{I(\mathbf{b}_0,h_c)} \mathbf{n} \cdot (-p\mathbf{I} + \mathbf{S}^{\nu} - \mathbf{u}_B \otimes \mathbf{u}_B)\, d\gamma = \\ &= \mathcal{J}_0(\mathbf{b}_0, h_c) + h^2 \mathcal{J}_2(\mathbf{b}_0, h_c). \end{aligned} \tag{B.69}$$

At the same time, we introduce an expansion for the sum of the different orders in the ME (B.64)

$$\Phi = \sum_{(\mathbf{k},i)\in B(\Gamma)} \phi_i(\mathbf{k}) = \Phi^{(0)} + h^2 \Phi^{(2)}.$$

Skipping the details of the derivation, which is completely analogous to the previous one, we find the following estimates, order by order:

$$\left| \frac{\mathcal{J}_0(\mathbf{b}_0, h_c)}{h_c} - \left(\frac{h}{h_c}\right)^d \Phi^{(0)}(\mathbf{b}_0, h_c) \right| \in O\left(h_c + \frac{h}{h_c}\right). \tag{B.70}$$

$$\left| \frac{\mathcal{J}_2(\mathbf{b}_0, h_c)}{h_c} - \left(\frac{h}{h_c}\right)^d \Phi^{(2)}(\mathbf{b}_0, h_c) \right| \in O\left(h_c + \frac{h}{h_c}\right). \tag{B.71}$$

Remarks. The additional term in the second order can be written as

$$\int_I \mathbf{u}_B (\mathbf{u}_B \cdot \mathbf{n}) d\gamma.$$

Hence, if $\mathbf{u}_B = 0$ (fixed boundary) or $\mathbf{u}_B \cdot \mathbf{n} = 0$ (for example, in the case of sliding flat walls) it does not give contribution.

B.4 Global force evaluation

Theorem B.2 (5.2). *Let Γ be a smooth d-dimensional interface of finite length. Denoting with $\overline{\Phi}(\Gamma) = \sum_{(\mathbf{k},i)\in B(\Gamma)} \frac{\overline{\phi}_i(\mathbf{k})}{h^2}$ the total sum of corrected momentum exchange, it holds*

$$\left| \int_\Gamma (-p\mathbf{I} + \mathbf{S}) \cdot \mathbf{n}\, d\sigma - h^d\, \overline{\Phi}(\Gamma) \right| \in O(h). \tag{B.72}$$

As before, the proof is discussed for $d = 1$, considering a one-dimensional interface in a domain $\Omega \subset \mathbb{R}^2$. We will actually prove that estimate (B.72) is true *eventually*, i.e. for h small enough. The argument is based on the prediction of the second order coefficient

$$\phi_i^{(2)} = \left(p - c_s^{-2} \nu \mathbf{c}_i \cdot \nabla \mathbf{u}_B \cdot \mathbf{c}_i\right) \mathbf{c}_i. \tag{B.73}$$

Simplified interface geometries

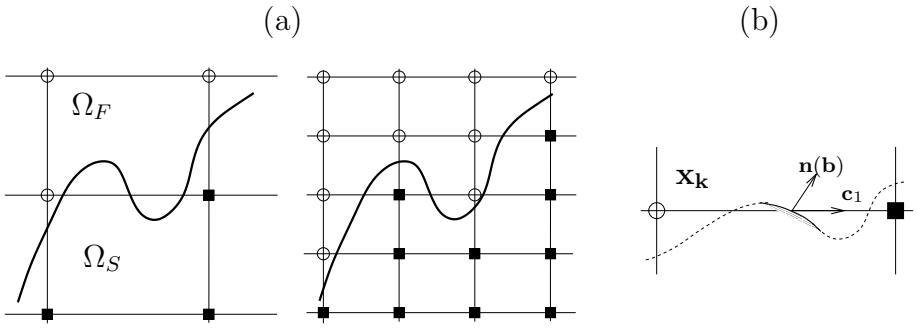


Figure B.8: **(a)** On the left, the interface (bold line) crosses the horizontal link three times in a single lattice cell. This case is excluded choosing a grid size h small enough (right). **(b)** The link $\mathbf{c}_1$ from the fluid node $\mathbf{k}$ intersects the boundary in a point $\mathbf{b}$, such that $\mathbf{c}_1 \cdot \mathbf{n}(\mathbf{b}) > 0$. In a neighborhood of $\mathbf{b}$ (solid line), the solid domain is on the same side of the fluid node. Therefore, there exist at least two other intersections on the link connecting $\mathbf{k}$ and $\mathbf{k} + \mathbf{c}_1$.

To avoid a nasty treatment of the different possible geometric configurations, we add some simplifying hypothesis. Let us consider the situations depicted in figure B.8a. We assume to have a grid of size h small enough, such that there is at most one intersection lattice-interface along a single node-node link.
In this case, if the direction $\mathbf{c}_i$ starting from the boundary node $\mathbf{k}$ is outgoing, i.e.

$$\mathbf{x_k} \in \Omega_F \cup \Gamma,\ \mathbf{x_k} + h\mathbf{c}_i \in \Omega_S,$$

and crosses the boundary at the point $\mathbf{b}_i(\mathbf{k})$, it has to hold

$$\mathbf{c}_i \cdot \mathbf{n}(\mathbf{b}) \leq 0. \tag{B.74}$$

In fact, if $\mathbf{c}_i \cdot \mathbf{n}(\mathbf{b}_i(\mathbf{k})) > 0$ it will be $\mathbf{c}_i \cdot \mathbf{n}(\mathbf{b}) > 0$ also in a neighborhood of $\mathbf{b}_i(\mathbf{k})$. Let us focus for simplicity on the case $i = 1$ (figure B.8b). This implies that at least a small part of solid domain will be on the left of the considered boundary point. However, if $\mathbf{x_k} \in \Omega_F$ and $\mathbf{x_k} + h\mathbf{c}_1 \in \Omega_S$, along the link $\mathbf{c}_1$ there have to be at least two intersections more.
Actually, regarding the intersection grid-boundary there are two possibilities. It could happen that a subset of the interface coincides with a part of the LB-lattice $\overline{\mathcal{G}(h)}$. In this case (see figure B.9a), the normal vector is orthogonal also to the lattice link, i.e. $\mathbf{c}_i \cdot \mathbf{n} = 0$.
In this situation we consider an *ϵ-perturbed interface*, where ϵ is an arbitrary small parameter. Without discussing the operation in detail, we assume to construct an interface Γ^ϵ which is a small deformation of Γ in a neighborhood of the points $\{\mathbf{b} \in \Gamma \mid \mathbf{c}_i \cdot \mathbf{n}(\mathbf{b}) = 0\}$, in such a way that the boundary nodes do not change, but the normal vector to Γ^ϵ is no longer orthogonal to $\mathbf{c}_i$ (figure B.9b).

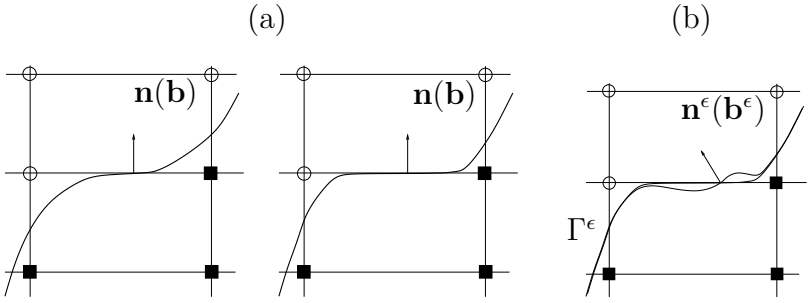


Figure B.9: **(a)** On the left, the link $\mathbf{c}_1$ crosses the interface in a point $\mathbf{b}$ such that $\mathbf{c}_1 \cdot \mathbf{n}(\mathbf{b}) = 0$. On the right, the outgoing direction coincides with part of the boundary. **(b)** The bold line is an *ϵ-perturbation* of the interface depicted in figure (a), right. Now there is a single intersection point $\mathbf{b}^\epsilon$, and $\mathbf{c}_1 \cdot \mathbf{n}^\epsilon(\mathbf{b}^\epsilon) < 0$.

Starting assumptions Summarizing,

- up to considering an arbitrary small perturbation of the interface,
- taking a grid size h small enough

we assume to deal with an interface Γ *smooth one-dimensional* such that

$$\begin{aligned} &(\mathbf{a}) \text{ a node-node link intersects } \Gamma \text{ at most once,} \\ &(\mathbf{b}) \ \forall \mathbf{b}_i(\mathbf{k}) \in \mathcal{B}(\Gamma) : \ \mathbf{c}_i \cdot \mathbf{n}(\mathbf{b}_i(\mathbf{k})) < 0. \end{aligned} \tag{B.75}$$

i-dependent partitions

Holding condition (B.75), we have

$$\forall \mathbf{b}_i(\mathbf{k}) \in \mathcal{B}(\Gamma) : \frac{\mathbf{c}_i \cdot \mathbf{n}(\mathbf{b}_i(\mathbf{k}))}{|\mathbf{c}_i \cdot \mathbf{n}(\mathbf{b}_i(\mathbf{k}))|} = -1.$$

We write explicitely the expressions (B.73) for different i (suppressing the argument $\mathbf{b}_i(\mathbf{k})$) as

$$\begin{aligned}
\phi_1^{(2)} &= -\frac{\mathbf{c}_1 \cdot \mathbf{n}}{|\mathbf{c}_1 \cdot \mathbf{n}|}\left(-\frac{2}{3}p + \mathbf{S}^\nu_{xx}, 0\right)\\
\phi_2^{(2)} &= -\frac{\mathbf{c}_2 \cdot \mathbf{n}}{|\mathbf{c}_2 \cdot \mathbf{n}|}\left(-\frac{1}{6}p + \frac{1}{2}\mathbf{S}^\nu_{xy}, -\frac{1}{6}p + \frac{1}{2}\mathbf{S}^\nu_{xy}\right)\\
\phi_3^{(2)} &= -\frac{\mathbf{c}_3 \cdot \mathbf{n}}{|\mathbf{c}_3 \cdot \mathbf{n}|}\left(0, -\frac{2}{3}p + \mathbf{S}^\nu_{yy}\right)\\
\phi_4^{(2)} &= -\frac{\mathbf{c}_4 \cdot \mathbf{n}}{|\mathbf{c}_4 \cdot \mathbf{n}|}\left(\frac{1}{6}p + \frac{1}{2}\mathbf{S}^\nu_{xy}, -\frac{1}{6}p - \frac{1}{2}\mathbf{S}^\nu_{xy}\right)\\
\phi_5^{(2)} &= \frac{\mathbf{c}_1 \cdot \mathbf{n}}{|\mathbf{c}_5 \cdot \mathbf{n}|}\left(-\frac{2}{3}p + \mathbf{S}^\nu_{xx}, 0\right)\\
\phi_6^{(2)} &= \frac{\mathbf{c}_2 \cdot \mathbf{n}}{|\mathbf{c}_6 \cdot \mathbf{n}|}\left(-\frac{1}{6}p + \frac{1}{2}\mathbf{S}^\nu_{xy}, -\frac{1}{6}p + \frac{1}{2}\mathbf{S}^\nu_{xy}\right)\\
\phi_7^{(2)} &= \frac{\mathbf{c}_3 \cdot \mathbf{n}}{|\mathbf{c}_7 \cdot \mathbf{n}|}\left(0, -\frac{2}{3}p + \mathbf{S}^\nu_{yy}\right)\\
\phi_8^{(2)} &= \frac{\mathbf{c}_4 \cdot \mathbf{n}}{|\mathbf{c}_8 \cdot \mathbf{n}|}\left(\frac{1}{6}p + \frac{1}{2}\mathbf{S}^\nu_{xy}, -\frac{1}{6}p - \frac{1}{2}\mathbf{S}^\nu_{xy}\right).
\end{aligned} \tag{B.76}$$

The use of $\dfrac{\mathbf{c}_i \cdot \mathbf{n}}{|\mathbf{c}_i \cdot \mathbf{n}|}$ we made in equations (B.76) is an expedient to have the expressions in brackets in a form analogous as the one that appears in integral (5.2) (where p has a minus sign and $\mathbf{S}^\nu$ has a plus, for $i = 1, \ldots, 8$).
The intersections grid-boundary are irregularly distributed along Γ. Collecting separately the intersections with links of different orientation, we obtain more structured partitions. Let us divide the lattice in sub-grids (four, for the D2Q9)

$$\begin{aligned}
\overline{\mathcal{G}^{(1,5)}} &:= \{(x,y) \in \Omega \mid y = hj_y,\ \mathbf{j} \in \mathcal{G}(h)\},\\
\overline{\mathcal{G}^{(3,7)}} &:= \{(x,y) \in \Omega \mid x = hj_x,\ \mathbf{j} \in \mathcal{G}(h)\},\\
\overline{\mathcal{G}^{(2,6)}} &:= \{(x,y) \in \Omega \mid y = (x - hj_x) + hj_y,\ \mathbf{j} \in \mathcal{G}(h)\},\\
\overline{\mathcal{G}^{(4,8)}} &:= \{(x,y) \in \Omega \mid y = -(x - hj_x) + hj_y,\ \mathbf{j} \in \mathcal{G}(h)\}.
\end{aligned} \tag{B.77}$$

In words, $\mathcal{G}^{(i,i^*)}$, with $\mathbf{c}_{i^*} = -\mathbf{c}_i$, is the subset of Ω containing the lines of the lattice parallel to $\mathbf{c}_i$ and $\mathbf{c}_{i^*}$.

We can also write

$$\overline{\mathcal{G}^{(i,i^*)}} = \{h\mathbf{j} \pm [0,1]h\mathbf{c}_i \mid \mathbf{j} \in \mathcal{G}(h)\}.$$

As next, we construct the subsets of $\mathcal{B}$ containing the intersection between interface and sub-grids

$$\mathcal{B}^s(\Gamma) = \Gamma \cap \overline{\mathcal{G}^s}, \tag{B.78}$$

$s = (i, i^*)$ being a couple of indexes such that $\mathbf{c}_i = -\mathbf{c}_{i^*}$.
Each set $\mathcal{B}^s$ defines a different partition of Γ. The total number of points in each partition depends on the geometry of the grid and the interface. With the usual parameterization of Γ through the coordinate $\gamma \in [0, 2\pi)$

$$\begin{array}{ccc} [0,2\pi) & \rightarrow & \Gamma \\ \gamma & \mapsto & \mathbf{b}(\gamma) = (x(\gamma), y(\gamma)), \end{array} \tag{B.79}$$

we can order the elements of $\mathcal{B}^s$ according to their coordinates:

$$\mathcal{B}^s = \{\mathbf{b}_r^s = \mathbf{b}(\gamma_r^s) \mid r = 1, \ldots, P^s\}.$$

Remark. Because of the condition $\mathbf{c}_i \cdot \mathbf{n} < 0$, each point in $\mathcal{B}^s$ is related either to the link $\mathbf{c}_i$ or to the opposite $\mathbf{c}_{i^*}$, being

$$\mathbf{c}_i \cdot \mathbf{n} = -\mathbf{c}_{i^*} \cdot \mathbf{n}.$$

Equivalently, within the four sets $\mathcal{B}^s(\Gamma)$, we are counting *exactly once* all the intersections between Γ and the lattice $\overline{\mathcal{G}(h)}$.

Reference systems

Let us focus on a particular direction $\mathbf{c}_i$ which intersects the interface in a certain point of coordinate γ_r. We introduce the orthogonal reference systems $\mathcal{S}_i$, defined, for a given $\mathbf{c}_i$, by (figure B.10)

$$\begin{array}{l} \xi = \mathbf{c}_{iy}x - \mathbf{c}_{ix}y \\ \eta = \mathbf{c}_{ix}x + \mathbf{c}_{iy}y. \end{array} \tag{B.80}$$

The coordinate $\mathcal{S}_i = (\xi, \eta)$ is obtained rotating the original system, so to orient the ordinates as the vector $\mathbf{c}_i$. For example, for $\mathbf{c}_i = \mathbf{c}_1 = (1, 0)$, the coordinates change from (x, y) to $(-y, x)$.
Expressing a point on the interface as $\mathbf{b} = (x(\gamma), y(\gamma))$, for the tangential and the normal vector the following geometrical relations hold:

$$\mathbf{t} = \left(\frac{\partial x}{\partial \gamma}, \frac{\partial y}{\partial \gamma}\right)$$

$$\mathbf{n} = \left(\frac{\partial y}{\partial \gamma}, -\frac{\partial x}{\partial \gamma}\right) = (n_x, n_y).$$

In terms of the coordinate system (ξ, η), we have

$$\frac{\partial \xi}{\partial \gamma} = \frac{\partial \xi}{\partial x}\frac{\partial x}{\partial \gamma} + \frac{\partial \xi}{\partial y}\frac{\partial y}{\partial \gamma} = \mathbf{c}_{iy}(-n_y) + (-\mathbf{c}_{ix})n_x = -\mathbf{c}_i \cdot \mathbf{n}. \tag{B.81}$$

In view of (B.75), if $\gamma_r \in \mathcal{B}$ then $\mathbf{c}_i \cdot \mathbf{n}(\mathbf{b}(\gamma)) \neq 0$ in a neighborhood of γ_r. Therefore, we can locally define an inverse

$$\gamma = \gamma(\xi),$$

and express a point $\mathbf{b} \in \Gamma$ as (Dini's Theorem)

$$\mathbf{b} = (\xi, \eta(\gamma(\xi))). \tag{B.82}$$

It holds, in view of (B.81),

$$\frac{\partial \gamma}{\partial \xi}(\mathbf{b}) = -\frac{1}{\mathbf{c}_i \cdot \mathbf{n}(\mathbf{b})}.$$

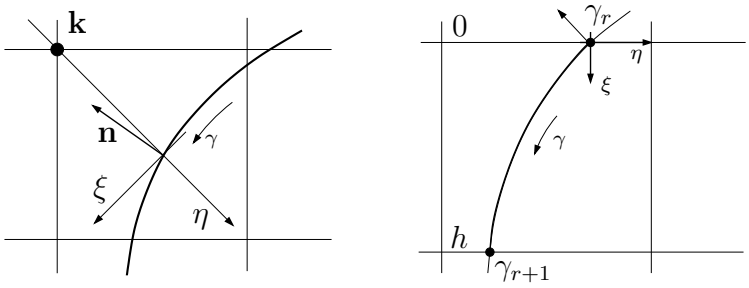


Figure B.10: **Left:** Rotation of the axis which defines the reference system $\mathcal{S}_i$, according to a boundary couple $(\mathbf{k}, i)$. The axes η has the same versus and direction as the outgoing direction $\mathbf{c}_i$. **Right:** Interval for the integration (B.84), between $\gamma_r = \gamma(0)$ (coordinate of the intersection $\mathbf{b} = (0, \eta(0))$, using $\xi = 0$ in the representation (B.82) for a point on the interface) and $\gamma_{r+1} = \gamma(h)$ (for $\xi = h$).

Generalized i-quadrature

Assume now to deal with two consecutive intersections, between Γ and two parallel links of the lattice. We indicate these points with

$$\mathbf{b}(\gamma_r), \mathbf{b}(\gamma_{r+1}) \in \mathcal{B}^{(i,i^*)}.$$

Technical problems might arise if the scalar product $\mathbf{c}_i \cdot \mathbf{n}$ approaches zero between γ_r and γ_{r+1}. Hence, we consider the set

$$\mathcal{B}_0^{(i,i^*)} = \left\{ \mathbf{b}(\gamma_r) \in \mathcal{B}^{(i,i^*)} \mid \forall \gamma \in (\gamma_r, \gamma_{r+1}) : \ \mathbf{c}_i \cdot \mathbf{n}(\mathbf{b}(\gamma)) < 0 \right\} \tag{B.83}$$

of all the nodes γ_r in the i-partition for which the scalar product does not experience any change of sign, remaining below zero (as it is in $\mathbf{b}(\gamma_r)$).
In the coordinate system $\mathcal{S}_i$, fixing the zero of the coordinate ξ in correspondence with the considered intersection point, we have (figure B.10)

$$\gamma_r = \gamma(0), \quad \gamma_{r+1} = \gamma(h).$$

The terms involved in the MEA-sum are of the form (equation (B.6))

$$\phi_i(\mathbf{b}_r) = \sum_{\zeta} \mathbf{a}^{\zeta,i} \zeta(\mathbf{b}_r).$$

Considering the integration of the pressure, for a fixed direction i (an analogous argument can be used also for the stress tensor). Using the abbreviations $p(\gamma) = p(\mathbf{b}(\gamma))$ and $\mathbf{n}(\gamma) = \mathbf{n}(\mathbf{b}(\gamma))$, we have

$$hp(\gamma_r) = \int_0^h p(\gamma(\xi))d\xi + O(h^2) = \int_0^h p(\gamma(\xi))\frac{\mathbf{c}_i \cdot \mathbf{n}(\gamma(\xi))}{\mathbf{c}_i \cdot \mathbf{n}(\gamma(\xi))}d\xi + O(h^2) = \\ = \int_{\gamma_r}^{\gamma_{r+1}} -p(\gamma)\mathbf{c}_i \cdot \mathbf{n}(\gamma)d\gamma + O(h^2). \quad \text{(B.84)}$$

Extending the sum to the set $\mathcal{B}_0^s$ defined in (B.83), we conclude

$$\Phi_x^s = \sum_{\substack{(\mathbf{k},m)\in B(\Gamma):\\ m\in\{i,i^*\},\, \mathbf{b}_i(\mathbf{k})\in\mathcal{B}_0^s}} h\phi_m(\mathbf{k}) = \mathbf{a}^{p,i} \int_{\Gamma_0^s} -p(\gamma)\mathbf{c}_i \cdot \mathbf{n}(\gamma)d\gamma + \\ \sum_{\alpha,\beta} \mathbf{a}^{\mathbf{S}^\nu_{\alpha,\beta},i} \int_{\Gamma_0^s} \mathbf{S}^\nu_{\alpha,\beta}(\gamma)(\mathbf{c}_i \cdot \mathbf{n}(\gamma))d\gamma + O(h), \quad \text{(B.85)}$$

where $s = (i, i^*)$ and the subset $\Gamma_0^s \subset \Gamma$ collects all the interval between the consecutive nodes in $\mathcal{B}_0^s$.
The above formula takes into account also the intersections between the interface and the links $\mathbf{c}_{i^*}$. In this case, as in equation (B.76), the inversion of sign

$$\mathbf{a}^{p,i} = -\mathbf{a}^{p,i^*}$$

is reabsorbed by the defined pre-factor $\dfrac{\mathbf{c}_i \cdot \mathbf{n}}{\mathbf{c}_{i^*} \cdot \mathbf{n}}$.

Estimates for the neglected parts Together with equation (B.85), we need to estimate the contribution to the integral due to the part of the interface out of Γ_0^s. By definition, $\gamma_r \notin \mathcal{B}_0^s$ if there is a point γ_* between γ_r and γ_{r+1} such that $\mathbf{c}_i \cdot \mathbf{n}(\gamma_*) = 0$. If the grid size is small enough (i.e. eventually), we can restrict to two situations (figure B.11):

(a) the function $\mathbf{c}_i \cdot \mathbf{n}(\gamma)$ has a simple zero between γ_r and γ_{r+1},
(b) there exists an interval $I_r \subset (\gamma_r, \gamma_{r+1})$ such that $\mathbf{c}_i \cdot \mathbf{n}(\gamma) = 0$, for all $\gamma \in I_r$.

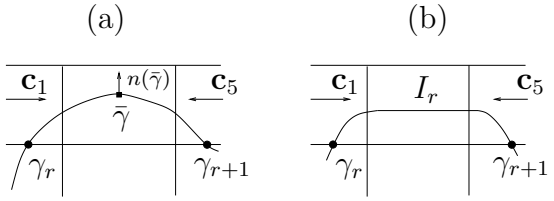


Figure B.11: **(a)** The function $\mathbf{c}_i \cdot \mathbf{n}(\gamma)$ has a zero between γ_r and γ_{r+1}. **(b)** The function $\mathbf{c}_i \cdot \mathbf{n}(\gamma)$ vanishes in a subset $I_r \subset (\gamma_r, \gamma_{r+1})$. The integral for the force does not receive contributions from this part of the boundary.

If the interface is *smooth* and has *finite length*, the function $\mathbf{c}_i \cdot \mathbf{n}$ has only a *finite number* of simple zeros, i.e. there will be only a finite number of points belonging to the cases **(a)** and, consequently, a finite number (independent on h) of neglected intervals of length $O(h)$. Concerning the case **(b)**, we can simply include the subset I_r of the interface into the integral of equation (B.85):

$$\Phi_x^s = \mathbf{a}^{p,i} \int_{\Gamma_0^s \cup I_r} -p(\gamma)\mathbf{c}_i \cdot \mathbf{n}(\gamma) d\gamma + \sum_{\alpha,\beta} \mathbf{a}^{\mathbf{S}^\nu_{\alpha,\beta},i} \int_{\Gamma_0^s \cup I_r} \mathbf{S}^\nu_{\alpha,\beta}(\gamma)(\mathbf{c}_i \cdot \mathbf{n}(\gamma)) d\gamma + O(h), \quad \text{(B.86)}$$

since $\mathbf{c}_i \cdot \mathbf{n}(\gamma) = 0$, for all $\gamma \in I_r$. Therefore, including in Γ_0^s also the subsets of the interface belonging to the case **(b)**, the set excluded from the computation only consists of a finite number of intervals of length $O(h)$. We can then estimate

$$\int_{\Gamma/\Gamma_0^s} \mathbf{n} \cdot (-p\mathbf{I} + \mathbf{S}^\nu)\, d\gamma \in O(h), \text{ eventually.}$$

Finally, writing explicitely the relations (B.86) for all the couples (i, i^*):

$$\begin{aligned}
\Phi_x^{(1,5)} &= \int_\Gamma n_x(\gamma) \left(-\frac{2}{3} p(\gamma) + \mathbf{S}^\nu_{xx}(\gamma) \right) d\gamma + O(h), \quad \Phi_y^{(1,5)} \in O(h), \\
\Phi_x^{(3,7)} &\in O(h), \quad \Phi_y^{(3,7)} = \int_\Gamma n_y(\gamma) \left(-\frac{2}{3} p(\gamma) + \mathbf{S}^\nu_{yy}(\gamma) \right) d\gamma + O(h), \\
\Phi_x^{(2,6)} &= \int_\Gamma (n_x + n_y) \left(-\frac{1}{6} p + \frac{1}{2} \mathbf{S}^\nu_{xy} \right) d\gamma + O(h), \\
\Phi_y^{(2,6)} &= \int_\Gamma (n_x + n_y) \left(-\frac{1}{6} p + \frac{1}{2} \mathbf{S}^\nu_{xy} \right) d\gamma + O(h), \\
\Phi_x^{(4,8)} &= \int_\Gamma (n_x - n_y) \left(-\frac{1}{6} p - \frac{1}{2} \mathbf{S}^\nu_{xy} \right) d\gamma + O(h), \\
\Phi_y^{(4,8)} &= \int_\Gamma (-n_x + n_y) \left(-\frac{1}{6} p - \frac{1}{2} \mathbf{S}^\nu_{xy} \right) d\gamma + O(h).
\end{aligned} \quad \text{(B.87)}$$

So that, being the numerical result the total amount of *all* the contributions of (B.87), we obtain the estimate which proves lemma 5.2:

$$\left(\Phi^{(1,5)} + \Phi^{(3,7)} + \Phi^{(2,6)} + \Phi^{(4,8)}\right) = \begin{pmatrix} \int_\Gamma n_x(-p + \mathbf{S}^\nu_{xx}) + n_y \mathbf{S}^\nu_{xy} \\ \int_\Gamma n_y(-p + \mathbf{S}^\nu_{yy}) + n_x \mathbf{S}^\nu_{xy} \end{pmatrix} + O(h). \tag{B.88}$$

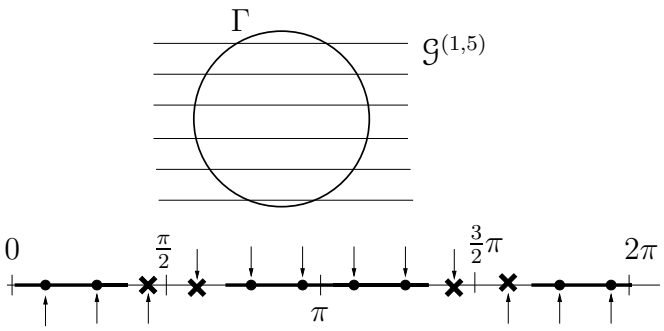


Figure B.12: Along the interface of a disk, parameterized in $[0, 2\pi)$, the partition $\mathcal{B}^{(1,5)}$ is drawn. The arrows on the top represent the intersection with the links $\mathbf{c}_1$, for $\frac{\pi}{2} < \gamma < \frac{3}{2}\pi$, the ones on the bottom, the intersection with $\mathbf{c}_5$. The nodes belonging to $\mathcal{B}_0^{(1,5)}$ are drawn with circles ($\bullet$), and the related intervals are highlighted with a bold line. The crosses ($\times$) are the nodes where the product $\mathbf{c}_i \cdot \mathbf{n}$ vanishes.

The uncorrected momentum exchange Using a similar argument (see also section B.3.4), it can be proven that the uncorrected momentum exchange leads to an approximation of

$$\mathcal{J} = c_s^2 \int_\Gamma -\mathbf{n} d\gamma + h^2 \int_\Gamma (-p + \mathbf{S}^\nu - \mathbf{u}_B \otimes \mathbf{u}_B) \cdot \mathbf{n}\, d\gamma. \tag{B.89}$$

Observe that

$$\int_\Gamma -\mathbf{n} d\gamma = 0$$

if Γ is a closed curve. Moreover,

$$\int_\Gamma (\mathbf{u}_B \otimes \mathbf{u}_B) \cdot \mathbf{n}\, d\gamma = 0$$

if the boundary is fixed, or moving orthogonal to a constant normal (plane walls). If the structure is a *rigid body*, it holds

$$\forall \mathbf{b} \in \Gamma : \ \mathbf{u}_B(\mathbf{b}) = \mathbf{u}_B(\mathbf{b}_0) + \boldsymbol{\Omega} \times \overline{\mathbf{b}_0 \mathbf{b}},$$

being $\mathbf{\Omega}$ the angular velocity and $\mathbf{b}_0$ a reference point of the body. For some particular configuration, like a circular interface, choosing $\mathbf{b}_0$ as the center we have

$$\int_\Gamma (\mathbf{u}_B \otimes \mathbf{u}_B) \cdot \mathbf{n}\, d\gamma = 0,$$

since $\mathbf{n}$ is parallel to $\overline{\mathbf{b}_0\mathbf{b}}$.

Conclusive remark Even if *separately* the sums of momentum exchange along the different lattice directions lead to different integrals, the symmetry properties of the discrete velocities set $\mathbb{V}$ balance the *global sum*.

List of Figures

Bibliography

[1] P. Asinari, *Semi-implicit-linearized Multiple-relaxation-time formulation of Lattice Boltzmann Schemes for Mixtures Modeling*, submitted to Phys. Rew. E (2006).

[2] R. Benzi, S. Succi, and M. Vergassola, *The lattice Boltzmann equation: Theory and applications*, Phys. Rep. **222** (1992), 147–197.

[3] P. L. Bhatnagar, E-P. Gross, and M. Krook, *A model for collision processes in gases I: Small amplitude processes in charged and neutral one-component systems*, Phys. Rew. E **94** (1954), 511–525.

[4] M. Bouzidi, M. Firdaouss, and P. Lallemand, *Momentum transfer on a Boltzmann-lattice fluid with boundaries*, Physics of Fluids **13** (2001), 3452–3459.

[5] A. Caiazzo, *Analysis of lattice Boltzmann initialization routines*, J. Stat. Phys. **121** (2005), 37–48.

[6] A. Caiazzo and M. Junk, *Boundary forces in lattice Boltzmann: analysis of Momentum Exchange algorithm*, Proc. of ICMMES 2005, accepted for publication on Computers and Mathematics with Applications (2006).

[7] S. Chen and G.D. Doolen, *Lattice Boltzmann method for fluid flows*, Ann. Rev. Fluid Mech. **30** (1992), 329–364.

[8] P. J. Dellar, *Incompressible limits of lattice Boltzmann equations using multiple relaxation times*, J. Comput. Phys. **190** (2003), 351–370.

[9] Irwan A. Dharmawan, private communication.

[10] D. d'Humières, M. Bouzidi, and P. Lallemand, *Thirteen-velocity three-dimensional lattice Boltzmann model*, Phys. Rew. E **63** (2001), 1–7.

[11] D. d'Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, and L.-S. Luo, *Multiple-relaxation-time lattice Boltzmann models in three dimensions*, Philos. Trans. R. Soc. London A **360** (2002), 437–451.

[12] O. Filippova and D. Hänel, *Grid Refinement for lattice-BGK Models*, J. Comp. Phys. **147** (1998), 219–224.

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

[14] U. Frisch, B. Hasslacher, and Y. Pomeau., *Lattice-gas automata for the Navier-Stokes equation*, Phys. Rew. Lett. **56** (1986), 1505–1508.

[15] I. Ginzbourg and D. d'Humiéres, *The multireflection boundary conditions for lattice Boltzmann models*, Phys. Rew. E **68** (2003), 066614.

[16] Z. Guo, C. Zheng, and B. Shi, *An extrapolation method for boundary conditions in lattice Boltzmann method*, Physics of Fluid **14** (2002), 2007–2010.

[17] X. He and L.-S. Luo, *Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation*, Phys. Rew. E **56** (1997), 6811–6817.

[18] M. Junk, *A finite difference interpretation of the lattice Boltzmann method*, Numerical methods for Partial Differential Equations **17** (2001), 383–402.

[19] M. Junk, A. Klar, and L.-S. Luo, *Asymptotic analysis of the lattice Boltzmann Equation*, Journal Comp. Phys. **210** (2005), 676–704.

[20] M. Junk and Z. Yang, *Analysis of lattice Boltzmann Boundary Conditions*, Proc. Appl. Math. Mech. **3** (2003), 76–79.

[21] ———, *Asymptotic analysis of finite difference method*, Appl. Math. Comput. **158** (2004), 267–301.

[22] ———, *A one-point boundary condition for the lattice Boltzmann method*, Phys. Rew. E **72** (2005), 066701.

[23] ———, *Convergence of Lattice Boltzmann Methods for Stokes Flows in Periodic and Bounded Domains*, Proc. of ICMMES 2005, submitted (2005).

[24] ———, *Outflow boundary conditions for the lattice boltzmann method*, Proc. of ICMMES 2006, submitted (2006).

[25] M. Junk and W. A. Yong, *Rigorous Navier-Stokes limit of the lattice Boltzmann equation*, Asymptotic Analysis **35** (2003), 165–185.

[26] ———, *Weighted L^2-stability of the lattice Boltzmann method*, Preprint (2005).

[27] J. Kevorkian and J.D. Cole, *Perturbation Methods in Applied Mathematics*, Springer-Verlag, 1981.

[28] A. J. C. Ladd, *Numerical simulations of particular suspensions via a discretized Boltzmann equation. Part 1*, J. Fluid Mech. **271** (1994), 285–310.

[29] ———, *Numerical simulations of particular suspensions via a discretized Boltzmann equation. Part 2 (Numerical results)*, J. Fluid Mech. **271** (1994), 311–339.

[30] P. Lallemand and L.-S. Luo, *Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galileian invariance and stability*, Phys. Rew. E **61** (2000), 6546–6562.

[31] ———, *Lattice Boltzmann method for moving boundaries*, J. Comp. Phys. **184** (2003), 406–421.

[32] H. Li, X. Lu, H. Fang, and Y. Qian, *Force evaluations in lattice Boltzmann simulations with moving boundaries in two dimensions*, Phys. Rew. E **70** (2004), 026701.

[33] G.R. McNamara and G. Zanetti, *Use of the Boltzmann equation to simulate lattice-gas automata*, Phys. Rew. Lett. **61** (1988), 2332–2335.

[34] R. Mei, L.-S. Luo, P. Lallemand, and D. d'Humières, *Consistent initial conditions for lattice Boltzmann simulations*, Computers and Fluids **35 (8/9)** (2006), 855–862.

[35] R. Mei, D. Yu, W. Shyy, and L.-S. Luo, *An accurate curved boundary treatment in the lattice Boltzmann method*, J. Comput. Phys. **155** (1999), 307–330.

[36] ———, *Force evaluation in the lattice Boltzmann method involving curved geometry*, Phys. Rew. E **65** (2002), 041203.

[37] R.R. Nourgaliev, T.N. Dinh, T.G. Theofanous, and D. Joseph, *The lattice Boltzmann equation: Theoretical Interpretation, Numerics and Implications*, Int. J. Multiphase Flow **29** (2003), 117–169.

[38] M. Rheinländer, *A consistent grid-coupling method for lattice-Boltzmann schemes*, J. Stat. Phys. **121** (2005), 49–74.

[39] ———, *Analysis of lattice Boltzmann method: Numeric and Asymptotic investigations of singularly perturbed systems*, Ph.D. thesis, Universität Konstanz (Germany), to appear.

[40] P.A. Skordos, *Initial and boundary conditions for the Lattice Boltzmann method*, Phys. Rew. E **48** (1993), 4823–4842.

[41] S. Succi, *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*, Oxford University Press, 2001.

[42] D. A. Wolf-Gladrow, *Lattice Gas Cellular Automata and Lattice Boltzmann Models*, Springer, Berlin, 2000.

[43] D. Yu, R. Mei, L.-S. Luo, and W. Shyy, *Viscous flow computation with the method of the lattice Boltzmann equation*, Progress in Aerospace Sciences **39** (2003), 329–367.