Towards a Lattice Boltzmann Method for Solids -Application to Static Equilibrium of Isotropic Materials

Tristan Maquart, Romain Noël, Guy Courbebaisse, Laurent Navarro

To cite this version:

Tristan Maquart, Romain Noël, Guy Courbebaisse, Laurent Navarro. Towards a Lattice Boltzmann Method for Solids -Application to Static Equilibrium of Isotropic Materials. 2021. hal-03225839

HAL Id: hal-03225839
https://hal.science/hal-03225839

Preprint submitted on 13 May 2021

Towards a Lattice BOLTZMANN Method for Solids - Application to Static Equilibrium of Isotropic Materials

Tristan MAQUART[a], Romain NOËL[b], Guy COURBEBAISSE[c], Laurent NAVARRO[a]

[a] *Mines Saint-Étienne, Univ. Lyon, Univ. Jean MONNET, INSERM U1059, SAINBIOSE, Centre Ingénierie Santé (CIS), 158 Cours Fauriel, 42023 Saint-Étienne, France*

[b] *Chair of Building Physics, Department of Mechanical and Process Engineering, ETH Zürich (Swiss Federal Institute of Technology in Zürich), Zürich 8093, Switzerland*

[c] *INSA-Lyon, Univ. Lyon, Université Claude BERNARD Lyon 1, UJM Saint-Étienne, CNRS, INSERM, UMR 5220, U1206, F69621, CREATIS, 7 Avenue Jean CAPELLE, 69621 Villeurbanne, France*

Abstract

This paper presents a novel method for simulating the behavior of solid objects with the Lattice BOLTZMANN Method (LBM). To introduce and validate our proposed framework, comparative studies are performed for computing the static equilibrium of isotropic materials. Reminding that the LBM has strong theoretical foundations in the BOLTZMANN equation; this latter is firstly adjusted to solid motions, through its BOLTZMANN-VLASOV special case. Indeed, the prior equation usually characterizes collisionless plasma and serves here, when combined with a suitable mean-field external force term, for setting a reliable solid framework. Secondly, a library is built and plugged on the top of the well-known Parallel Lattice Boltzmann Solver (PaLaBoS) library. Numerical implementations based on previous equation of motion for solids are led in a non-intrusive manner so as to present results with an easy and flawless reproducibility. A new designed Lattice BOLTZMANN Method for Solids (LBMS) is exhibited through few key algorithms showing the overall operation plus the major improvements. Efficiency, robustness and accuracy of the proposed approach are illustrated and contrasted with a commercial Finite Element Analysis (FEA) software. Obtained results reveal a considerable potential concerning static and further dynamic simulations involving solid constitutive laws within the LBM formalism.

Keywords: Lattice BOLTZMANN Method for Solids, Solid static equilibrium, VLASOV-MAXWELL equation, Mean-field external force term

---

1. Introduction

1.1. Background

The LBM[1] has proven to be an efficient and reliable method for CFD[2] since several decades now. Historically, it follows the seminal work of FRISCH[1] on the use of LGCA[3] for simulating the NAVIER-STOKES equations. This work was a remarkable breakthrough, as it proposed to simulate fluids at the particles level on an Eulerian grid using interaction rules. Unfortunately, this came at the cost of computation time to overcome the numerical statistical noise due to Boolean formulation of particle interactions. Decades before, BHATNAGAR, GROSS and KROOK achieved a linearization of the collision operator of the BE[4] [2] that allows to reduce the complexity of the equation by avoiding the implementation of an integral collision model. MACNAMARA[3] and SUCCI [4] proposed to use discretized versions of the BOLTZMANN equation to overcome the limitations of LGCA, and gave birth to the so-called Lattice BOLTZMANN Method. Following this idea, KARLIN *et al.* proposed a LBGKE[5] with an optimized local equilibrium and proved the H-theorem for it[5].

A high amount of work was since done in the LBM field, and recent years saw some remarkable improvements for the simulation of fluids with high REYNOLDS numbers, with for example the lattice-kinetic theory[6]. One can also mention the MRT[6] LBM which also allows better stability of the method

---

Email addresses: `tristan.maquart@hotmail.fr` (Tristan MAQUART), `romainoel@free.fr` (Romain NOËL)

[1] Lattice BOLTZMANN Method

[2] Computational Fluid Dynamics

[3] Lattice Gas Cellular Automata

[4] BOLTZMANN Equation

[5] Lattice BOLTZMANN-BGK Equation

[6] Multiple Relaxation Time

for high Reynolds numbers, using a multiple-relaxation time term. All these new insights are relevant for simulating fluids with high Reynolds numbers, but also open up on the idea of modifying the collision operator. Indeed, the collision operator is key in the method as it embeds the majority of the physics to be simulated.

The simulation of multi-components flows is another important research interest in the LBM field. This represents one of the major interests of the method: the LBM is conceptually relatively straightforward as it describes particle physics on a mesoscopic scale. However, the main challenge to overcome is the interface definition which can be sorted roughly into four categories. The first category, called color-gradient method[7], has been developed for two-phases flows. It consists in adding a perturbation to the linearized collision operator to make the pressure tensor locally anisotropic near a fluid-fluid interface. This results in surface tension at interfaces while retaining the compatibility to the NSE[7] in non-interfaces regions. The second category is based on the introduction of an inter-particle potential[8]. It allows the simulation of multi-phase and multi-component immiscible fluids with different masses at constant temperature, with a high efficiency. The third category can simulate hydrodynamics of phase separation and two-phases flow[9]. The principle is to use a non-ideal pressure tensor in the collision operator to ensure thermodynamic consistency. The goal of this approach is to improve physical consistency. The fourth category takes into account molecular interactions[10,11] for the simulation of incompressible two-phase flows. A review of multi-phase LBM is proposed in[12]. It gives a comprehensive overview of the methods and the associated algorithmic aspects.

The idea of combining specifically adapted collision operators and multi-phase approach leads to the question of the possibility of simulating solid matter with the LBM. In fact, the multi-phase LBM is of great interest, as it may simplify the notion of contact. As an example, the work of Chiappini *et al.* applied multi-phase method to the simulation of ligament break-up[13] gives an insight on its potential in bio-mechanical soft tissue simulations.

1.2. The Need for a LBM *for Solids*

Some attempts in solid simulations using the LBM have been done, for example in[14], with good application results on fractures and fragmentation for a solid body. Despite some limitations, such as the inability to fully recover the linear elastic equations, this work showed that the LBM is a good candidate when it comes to simulate the decohesion of the material. Other works combined the lattice-spring method and the LBM for FSI[8] [15]. For example,[16] is dedicated to the hemodynamic simulation in deformable blood vessels. However, the combination of two different methods with different coordinate systems complicates the simulation. Next to that, pure viscoelastic wave propagation simulations using the LBM have been proposed[17–19]. All these works highlight the potential of the LBM for simulating FSI in a lot of applications requiring the use of an Eulerian grid.

Indeed, simulating solids in an Eulerian framework, as in the LBM, is intricate and limited. The main limitation is certainly the loss of the initial configuration, which is not necessary for fluid simulations, but mandatory for solid simulations. Kamrin *et al.*[20] proposed the so-called "reference map" to compute finite-difference simulation of large solid-like deformations. They highlighted the fact that this reference map can simplify the simulation of fluid/solid interactions, as both materials have a similar Eulerian expression.

The benefits of being able to simulate solids are keys in solving FSI problems and simulation of specific constitutive laws in the LBM framework. The advantage of Eulerian framework is principally the absence of a deformable mesh. Moreover, the interest of the LBM for the simulation based on medical imaging data[21] and biological phenomena[22] have been highlighted. It appears that the use of the LBM framework allows the simulation of various problems based on what constitutes raw data in many biomedical applications, *i.e.* medical images.

This paper proposes a LBM for the simulation of solids. It is mainly based on the use of the Boltzmann-Vlasov equation, which is a special case of the Boltzmann equation. The aim is to introduce the new proposed method in detail and to present some numerical implementations for its validation.

The paper is organized as follows. Section 2 is dedicated to generalities about the LBM and BGK[9] approximation. Section 3 proposes a LBM for solid simulations, based on a Boltzmann-Vlasov equation. Section 4 presents numerical implementations and key algorithms of the method, and the integration

---

[7] Navier-Stokes Equations

[8] Fluid Structure Interaction

[9] Bhatnagar, Gross and Krook

in the PaLaBoS[10] [23] framework. Section 5 is dedicated to benchmarks and comparisons with FEA[11] using COMSOL software [24], in order to validate the method. Section 6 discusses the results and the work, and section 7 concludes the paper.

2. Lattice Boltzmann Method

2.1. *From* BE *to* LBM

The BE established in 1872 [25], describes the evolution of particles in the phase space and can be written as:

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} (f) + \boldsymbol{g} \cdot \nabla_{\boldsymbol{\xi}} (f) = \Omega (f, f), \tag{1}$$

where $\boldsymbol{g}$ is the force field felt by particles, like gravity for mass particles, and the distribution f corresponds to the statistical distribution of particles over the phase space, so $\boldsymbol{\xi}$ is the particles speed, and Ω is the collision-interaction operator representing the interactions between particles.

The latter operator is given, in its general case, by an integral. The Boltzmann equation is an integro-differential equation describing the evolution of an out-of-equilibrium thermodynamic system.

Then, macroscopic variables are found thanks to the distribution density as follows:

$$\text{mass density:} \qquad \rho = \int f(\boldsymbol{x}, \boldsymbol{\xi}, t)\, \mathrm{d}\boldsymbol{\xi}, \tag{2a}$$

$$\text{overall speed:} \qquad \boldsymbol{v} = \frac{1}{\rho} \int \boldsymbol{\xi} f(\boldsymbol{x}, \boldsymbol{\xi}, t)\, \mathrm{d}\boldsymbol{\xi}, \tag{2b}$$

$$\text{kinetic energy:} \qquad E_k = \frac{1}{2} \int \boldsymbol{v}^2 f(\boldsymbol{x}, \boldsymbol{\xi}, t)\, \mathrm{d}\boldsymbol{\xi}, \tag{2c}$$

$$\text{viscous pressure tensor:} \qquad \Pi = \int \boldsymbol{c} \otimes \boldsymbol{c} f(\boldsymbol{x}, \boldsymbol{\xi}, t)\, \mathrm{d}\boldsymbol{\xi}, \tag{2d}$$

where $\boldsymbol{c}$ is the microscopic velocity of the particles in the macroscopic average velocity frame, *i.e.* $\boldsymbol{c} = (\boldsymbol{\xi} - \boldsymbol{v})$.

To reduce the complexity of the collision operator, the BGK [2] linearization, also called simple relaxation time approximation, is useful. So the Boltzmann-BGK equation is expressed by:

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} (f) + \boldsymbol{g} \cdot \nabla_{\boldsymbol{\xi}} (f) = -\omega \left(f - f^{(0)} \right). \tag{3}$$

Where $f^{(0)}$ is the equilibrium distribution described by the following Maxwell-Boltzmann distribution:

$$f^{(0)}(\boldsymbol{x}, \boldsymbol{\xi}, t) = \frac{\rho(\boldsymbol{x}, t)}{(2\pi R\theta(\boldsymbol{x}, t))^{D/2}} \exp\left(-\frac{(\boldsymbol{\xi} - \boldsymbol{v}(\boldsymbol{x}, t))^2}{2R\theta(\boldsymbol{x}, t)} \right), \tag{4}$$

with θ the absolute temperature and R the universal gas constant.

It can be shown that in the zero Knudsen number limit, the BE with the BGK or a general collision kernel leads to the NSF[12] [26,27]. Then, the NSF equations can be recovered using the Chapman-Enskog expansion of the BE (for its simple development with a BGK kernel, see appendix Appendix A) and are expressed as:

$$\partial_t \rho + \nabla_{\boldsymbol{x}} \cdot (\rho \boldsymbol{v}) = 0, \tag{5a}$$

$$\partial_t (\rho \boldsymbol{v}) + \nabla_{\boldsymbol{x}} \cdot (\rho \boldsymbol{v} \otimes \boldsymbol{v} + p - \nu \nabla_{\boldsymbol{x}} \boldsymbol{v}) = 0, \tag{5b}$$

$$\partial_t (E_k + E_\theta) + \nabla_{\boldsymbol{x}} \cdot ((E_k + E_\theta) \boldsymbol{v} + p \cdot \boldsymbol{v} + \kappa \nabla_{\boldsymbol{x}} \theta) = 0. \tag{5c}$$

Here stand one conceptual advantage of the Boltzmann methods: solve several macroscopic coupled PDE[13] by solving one microscopic PDE in the phase space. To go solve numerical the BGK-Boltzmann

---

[10] Parallel Lattice Boltzmann Solver

[11] Finite Element Analysis

[12] Navier-Stokes-Fourier

[13] Partial Differential Equations

PDE, it is necessary to discretize the velocity space. Therefore, after a discretization through the GAUSS-HERMITE quadrature, the velocity space discretization of the BE reads:

$$\frac{\partial f_i}{\partial t}(\boldsymbol{x},t) + \boldsymbol{\xi}_i \cdot \nabla_{\boldsymbol{x}}(f_i) + g_i = \Omega(f_i, f_i), \tag{6}$$

where $\boldsymbol{\xi}_i$ are the discretized velocities; f_i and g_i are the projections of f and $\boldsymbol{g}$ on the HERMITE basis.

The previous BE discretized over the velocity space can be rewritten as:

$$\frac{\partial f_i}{\partial t}(\boldsymbol{x},t) + \boldsymbol{\xi}_i \cdot \nabla_{\boldsymbol{x}}(f_i) = \frac{D}{Dt} f_i(\boldsymbol{x},t) = \Omega_i(f_i, f_i) - g_i, \tag{7}$$

where $\frac{D}{Dt}$ is the particle derivative. Using the trapezoidal rule and a change of variable (f_i to $\widehat{f}_i$ see appendix Appendix B), the second-order discretization, called the LBGKE, reads:

$$\widehat{f}_i(\boldsymbol{x}+\boldsymbol{\xi}_i\Delta t, t+\Delta t) = \widehat{f}_i(\boldsymbol{x},t) - \Delta t\widehat{\omega}\left(\widehat{f}_i(\boldsymbol{x},t) - f_i^{(0)}(\boldsymbol{x},t)\right) - \Delta t\left(1-\frac{\widehat{\omega}}{2}\right) g_i(\boldsymbol{x},t) + \mathcal{O}(\Delta t^3). \tag{8}$$

In the sake of simplicity, the change of variable notation is dropped in the following of the paper. The time evolution can be simulated by a succession of collision step and streaming step described by the famous following equations which form the core of the LBM and reads:

$$f_i{}^c(\boldsymbol{x},t) = f_i{}^s(\boldsymbol{x},t) - \Delta t\omega\left(f_i{}^s(\boldsymbol{x},t) - f_i^{(0)}(\boldsymbol{x},t)\right) - \Delta t\left(1-\frac{\omega}{2}\right) g_i(\boldsymbol{x},t), \tag{9}$$

$$f_i{}^s(\boldsymbol{x}+\boldsymbol{\xi}_i\Delta t, t+\Delta t) = f_i{}^c(\boldsymbol{x},t). \tag{10}$$

2.2. Body force term

The change of variable used in the eq. (B.9) also has an impact on the macroscopic variable. With the use of the eq. (B.6) in appendix Appendix B, one can compute:

$$\sum_{i=0}^{q} f_i = \rho, \tag{11}$$

$$\sum_{i=0}^{q} \boldsymbol{\xi}_i f_i = \widehat{\rho\boldsymbol{v}} = \rho\boldsymbol{v} - \frac{\Delta t}{2} g_i, \tag{12}$$

$$\sum_{i=0}^{q} (\boldsymbol{\xi}_i - \boldsymbol{v}) \otimes (\boldsymbol{\xi}_i - \boldsymbol{v}) f_i = \widehat{\Pi} = p^{(0)} - \left(1+\frac{\Delta t\omega}{2}\right)\sigma^{(1)} - \frac{\Delta t}{2}(\boldsymbol{g}\otimes\boldsymbol{v} + \boldsymbol{v}\otimes\boldsymbol{g}). \tag{13}$$

In the last expression, $\sigma^{(1)}$ is given by eq. (A.10c). These previous sums justify the correctional terms in the macroscopic speed and the "lattice viscosity" that have to be compensated when fitting the physical cinematic viscosity with the collision frequency $\widehat{\omega} = \frac{2{c_s}^2}{{c_s}^2+2\nu}$. In the previous formula, c_s is the celerity of sound and obtained by ${c_s}^2 = \frac{1}{3}R\theta$.

And the equilibrium distribution in its usual form, at second order, is given by:

$$f_i^{(0)}(\rho,\boldsymbol{v}) = w_i\rho\left[1 + \frac{\boldsymbol{\xi}_i\cdot\boldsymbol{v}}{{c_s}^2} + \frac{(\boldsymbol{\xi}_i\cdot\boldsymbol{v})^2}{2{c_s}^4} - \frac{(\boldsymbol{v})^2}{2{c_s}^2}\right]. \tag{14}$$

Using GUO's approach[28], the projection of the forcing term on the HERMITE basis at third order reads:

$$\begin{aligned} g_i =& \frac{w_i\rho}{{c_s}^2}\boldsymbol{\xi}_i\cdot\boldsymbol{g} + \frac{w_i\rho}{2{c_s}^4}\left[(\boldsymbol{v}\cdot\boldsymbol{\xi}_i)\boldsymbol{\xi}_i - \frac{\boldsymbol{v}}{{c_s}^{-2}}\right]\cdot\boldsymbol{g} \\ &+ \frac{w_i\rho}{6{c_s}^6}\left[(\boldsymbol{v}\cdot\boldsymbol{\xi}_i)^2\boldsymbol{\xi}_i + (\sigma : \boldsymbol{\xi}_i\otimes\boldsymbol{\xi}_i)\boldsymbol{\xi}_i - \frac{{\boldsymbol{\xi}_i}^2\boldsymbol{\xi}_i}{{c_s}^{-2}} + \mathcal{P}_3\left(\frac{\boldsymbol{\xi}_i}{{c_s}^{-4}} - \frac{(\boldsymbol{v}\cdot\boldsymbol{\xi}_i)\boldsymbol{v}}{{c_s}^{-2}} - \frac{\sigma\cdot\boldsymbol{\xi}_i}{{c_s}^{-2}}\right)\right]\cdot\boldsymbol{g}. \end{aligned} \tag{15}$$

3. Lattice BOLTZMANN Method for Solids

3.1. Considerations about the VLASOV *equation*

In order to find a suitable framework for the use of the BOLTZMANN equation for solids, it is possible to note that at rest (at thermodynamic equilibrium) a particle distribution having a low temperature could be described by a DIRAC distribution (omitting all quantum effects). Then, the solid matter at rest condition can be seen as a combination of motionless point mass, and without large energy perturbation, the molecular interactions can be linearized. Starting from this state, a small vibration of one edge of the network will propagate freely. This situation can be described by the VLASOV equation[29], which is nothing but the BOLTZMANN's equation deprived of its collision term:

$$\partial_t (f)(\boldsymbol{x}, \boldsymbol{\xi}, t) + \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} (f) + \boldsymbol{g} \cdot \nabla_{\boldsymbol{\xi}} (f) = 0. \tag{16}$$

Then, the evolution of the particles described by this equation is a propagation of the particles without further modifications induced by the collisions.

Usually this equation is used to describe collision-less plasma. In such cases, it is coupled with the POISSON equation to add the electrical interactions that drive the motion of this mixture of electrically charged particles. The VLASOV–POISSON equations are an approximation of the VLASOV–MAXWELL equations and read:

$$\partial_t (f)(\boldsymbol{x}, \boldsymbol{\xi}, t) + \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} (f) + \frac{q_P E_P}{m} \cdot \nabla_{\boldsymbol{\xi}} (f) = 0, \tag{17}$$

where q_P is the particle electric charge, m is the particle mass and E_P is the self-consistent electric field resulting from the POISSON equation. The external force $\frac{q_P E_P}{m}$ is the mean-field approximation of the interactions between the electrical charges.

The interesting idea behind this equation is that complex systems with few collisions and strong interactions can be modeled by a VLASOV equation. Plus, these strong interactions are incorporated through the mean-field external forces term.

3.2. BOLTZMANN-VLASOV *equation for solids*

Using the previous consideration ("low" temperature and few collisions), the BOLTZMANN-VLASOV equation can be used with a DIRAC distribution:

$$f(\boldsymbol{x}, \boldsymbol{\xi}, t) = \rho \delta_{\boldsymbol{v}} (\boldsymbol{\xi}), \tag{18}$$

where $\delta_{\boldsymbol{v}} (\boldsymbol{\xi})$ is the DIRAC distribution centered on $\boldsymbol{v}$. Thus, using the eq. (16) and eq. (18) for an isothermal system, one can obtain the following macroscopic equations:

$$\partial_t (\rho) + \nabla_x \cdot \rho \boldsymbol{v} = 0, \tag{19a}$$

$$\partial_t (\rho \boldsymbol{v}) + \nabla_{\boldsymbol{x}} \cdot (\rho \boldsymbol{v} \otimes \boldsymbol{v}) = \rho \boldsymbol{g}. \tag{19b}$$

As in the VLASOV-POISSON equation, it is then necessary to add to this model the interaction forces. To do so in a solid framework, we consider the CAUCHY stress tensor. It is possible to convert the CAUCHY stress tensor into an equivalent of force per mass unit. By noting that the divergence of the stress tensor acts as a mass force (this amounts to nothing less than using the GREEN-OSTROGRADSKY theorem), it is possible to write $\boldsymbol{g} = \frac{1}{\rho} \nabla_{\boldsymbol{x}} \cdot (\sigma)$. Therefore, by considering the following mesoscopic model:

$$\frac{\partial f}{\partial t}(\boldsymbol{x}, \boldsymbol{\xi}, t) + \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} (f) + \boldsymbol{g} \cdot \nabla_{\boldsymbol{\xi}} (f) = 0, \tag{20a}$$

$$f(\boldsymbol{x}, \boldsymbol{\xi}, t) = \rho \delta_{\boldsymbol{v}} (\boldsymbol{\xi}), \tag{20b}$$

$$\boldsymbol{g} = \frac{1}{\rho} \nabla_{\boldsymbol{x}} \cdot (\sigma), \tag{20c}$$

the aimed macroscopic system of undefined equation of motion for solids is obtained:

$$\partial_t (\rho) + \nabla_x \cdot \rho \boldsymbol{v} = 0, \tag{21a}$$

$$\partial_t (\rho \boldsymbol{v}) + \nabla_{\boldsymbol{x}} \cdot (\rho \boldsymbol{v} \otimes \boldsymbol{v}) = \nabla_{\boldsymbol{x}} \cdot (\sigma). \tag{21b}$$

Thus, the BOLTZMANN-VLASOV equation leading to the previous system of equations seems suitable for rigid matter simulations. To simulate solids with this method, two elements appears to be compulsory. Firstly, it is necessary to discretize the BOLTZMANN-VLASOV equation by projecting on the GAUSS-HERMITE basis (which is done without lost of generality in section 2). Secondly, the numerical evaluation the divergence of the stress tensor is required.

Several remarks arise from these previous equations. This approach remains general for any constitutive law. Since the particles distribution is closer to a DIRAC distribution centered on $\boldsymbol{v}$, the streamed particles (*i.e.* f_i for $i \neq 0$) are closer to zero if there is no collective speed. Which is closer to the conventional representations of solid lattices: particles are not moving freely through the solid. Moreover, it implies that the streaming step in the LBM algorithm could be removed and the results would remain. Indeed, the motion would be driven through the divergence of the stress tensor and not through the streaming step. This last ascertainment has been observed in preliminary tests.

3.3. Simplifications related to static equilibrium of solids

As a first study, solid cases as simple as possible, are considered to validate our model. So, the infinitesimal strain theory framework is used. Thus, it is possible to simply use linearized strains, *i.e.* the linearized strain tensor ε. Moreover, to keep a low complexity, we consider isotropic linear elastic constitutive relationship. In this framework, the Eulerian and Lagrangian descriptions can be considered as equivalent, yielding to true CAUCHY stresses σ.

The main idea of the proposed method is to incorporate the stress divergence tensor of a solid behavior as an external force. To test this proposal and only this one (not the collision-less operator which is more related to dynamics), we examine this proposition as a first study to compute only static state. Therefore, a modification of the collision operator is required for static cases. Some of these requirements and simplifications are further developed in section 5 in order to set this static framework with thoroughness.

4. Numerical implementations and algorithmic

In this section, the key algorithms are presented in order to understand the suggested method from a technical point of view. Details are given in such manner to catch on relevant arithmetic specific to the Lattice BOLTZMANN Method for Solids in a two-dimensional space.

4.1. General implementation approach

To carry out a program which is manageable, robust and easy to install, multiple considerations have to be taken into account. First, a user-friendly implementation seems to be mandatory and permits to the community to develop new functions or interfaces. Secondly, seeking for robustness is obvious and leads to correct simulation results, but code efficiency is not directly required. Thirdly, working with a light environment is greatly appreciated, *i.e.*, with few dependencies allowing a fast installation on different platforms. All of these requirements will enable reproducibility of following results.

These objectives are mainly reached using C++ language and the well-known PALABOS[23] library. Hence, we decided to build an additional library which is plugged on top of PALABOS, *i.e.*, which uses only PALABOS as dependence: the LBMS[14] library. The LBMS library is, thus, an easily accessible setup. This strategy leads to a non-intrusive coding task where solid classes and solid objects are called beside PALABOS without modifying its functions or methods. The LBMS library algorithms and the results presented here show its ability to solve solid static equilibrium with the LBM. Library source code is available online[30].

4.2. Additional solid routines

In order to not modify the classic LBM loop, additional independent solid routines are inserted in existing steps. This approach leads to append new local operations across the lattice alongside the standard collide and stream stages, see eqs. (9) and (10). They are executed at each time iteration. Algorithms are presented taking into account a lattice structure composed by several blocks, where each independent structure can be calculated on a specific core processor as PALABOS does. For that purpose, quantities are expressed in physical units.

In solid mechanics, discrete displacement values are the unknown variables of the system to solve. We use time integration to retrieve this needed displacement from velocity, as detailed in algorithm 1. Once displacement tensor field $\mathbf{U}$ is recovered, the strain tensor field $\mathbf{E}$ is obtained from displacement by numerical differentiation. The stress tensor field $\mathbf{\Sigma}$ is, thus, deducted with a given solid constitutive law, see algorithm 2. A body force is then added to each cell thanks to the stress divergence tensor field $\nabla_{\boldsymbol{x}} \cdot (\mathbf{\Sigma})$ determined in algorithm 3. Previous tensor fields are defined over the whole lattice, *i.e.* using lattice global coordinates G_x and G_y in a two-dimensional space.

---

[14] Lattice BOLTZMANN Method for Solids

Algorithm 1: Displacement routine [Physical units]

```
Input: Macroscopic velocity field $\boldsymbol{v}$ and time increment $\Delta t$
Result: Macroscopic displacement tensor field $\mathbf{U}$
for $b \leftarrow 0$ to BlockLattice.Count()-1 do
    ${G_x}^p \leftarrow$ BlockLattice[b].GetBlockLatticeGlobalPositionX() ;  // Get current lattice position X
    ${G_y}^p \leftarrow$ BlockLattice[b].GetBlockLatticeGlobalPositionY() ;  // Get current lattice position Y
    for $x \leftarrow 0$ to BlockLattice[b].GetNx()-1 do
        for $y \leftarrow 0$ to BlockLattice[b].GetNy()-1 do
            $G_x \leftarrow x + {G_x}^p$ ;                                   // Current cell global position X
            $G_y \leftarrow y + {G_y}^p$ ;                                   // Current cell global position Y
            $\mathbf{U}[G_x, G_y] \mathrel{+}= \boldsymbol{v}^{cell} \times \Delta t$ ;                      // Time integration
        end
    end
end
```

Algorithm 2: Strain and stress routine [Physical units]

```
Input: Macroscopic displacement tensor field $\mathbf{U}$, space discretization $\Delta\boldsymbol{x}$ and solid constitutive
       law $C$
Result: Macroscopic strain and stress tensor fields: $\mathbf{E}$ and $\mathbf{\Sigma}$
for $b \leftarrow 0$ to BlockLattice.Count()-1 do
    ${G_x}^p \leftarrow$ BlockLattice[b].GetBlockLatticeGlobalPositionX() ;  // Get current lattice position X
    ${G_y}^p \leftarrow$ BlockLattice[b].GetBlockLatticeGlobalPositionY() ;  // Get current lattice position Y
    for $x \leftarrow 0$ to BlockLattice[b].GetNx()-1 do
        for $y \leftarrow 0$ to BlockLattice[b].GetNy()-1 do
            $G_x \leftarrow x + {G_x}^p$ ;                                   // Current cell global position X
            $G_y \leftarrow y + {G_y}^p$ ;                                   // Current cell global position Y
            $\mathbf{P} \leftarrow$ BlockLattice[b].GetCellVicinityParameters($x$, $y$) ;       // Vicinity parameters
            $\mathbf{E}[G_x, G_y]$ = ComputeStrainFromDisplacement($\mathbf{U}$, $\Delta\boldsymbol{x}$, $\mathbf{P}$) ;      // Strain computation
            $\mathbf{\Sigma}[G_x, G_y]$ = ComputeStressFromStrain($\mathbf{E}[G_x, G_y]$, $C$) ;     // Stress computation with
             solid constitutive law $C$
        end
    end
end
```

Algorithm 3: Body force routine [Physical units]

```
Input: Macroscopic stress tensor field $\mathbf{\Sigma}$ and space discretization $\Delta\boldsymbol{x}$
Result: Macroscopic stress divergence tensor field $\nabla_{\boldsymbol{x}} \cdot (\mathbf{\Sigma})$
for $b \leftarrow 0$ to BlockLattice.Count()-1 do
    ${G_x}^p \leftarrow$ BlockLattice[b].GetBlockLatticeGlobalPositionX() ;  // Get current lattice position X
    ${G_y}^p \leftarrow$ BlockLattice[b].GetBlockLatticeGlobalPositionY() ;  // Get current lattice position Y
    for $x \leftarrow 0$ to BlockLattice[b].GetNx()-1 do
        for $y \leftarrow 0$ to BlockLattice[b].GetNy()-1 do
            $G_x \leftarrow x + {G_x}^p$ ;                                   // Current cell global position X
            $G_y \leftarrow y + {G_y}^p$ ;                                   // Current cell global position Y
            $\mathbf{P} \leftarrow$ BlockLattice[b].GetCellVicinityParameters($x$, $y$) ;       // Vicinity parameters
            $\nabla_{\boldsymbol{x}} \cdot (\mathbf{\Sigma})\,[G_x, G_y]$ = ComputeStressDivergenceFromStress($\mathbf{\Sigma}$, $\Delta\boldsymbol{x}$, $\mathbf{P}$) ;
             // $\nabla_{\boldsymbol{x}} \cdot (\mathbf{\Sigma})\,[G_x, G_y]$ computation
            BlockLattice[b]($x$, $y$).ApplyLatticeBodyForce($\nabla_{\boldsymbol{x}} \cdot (\mathbf{\Sigma})\,[G_x, G_y]$) ; // Apply lattice body
             force to the current cell
        end
    end
end
```

These three routines allow to compute needed tensor fields over the lattice domain. They describe solid variables for each lattice cell at time t. In a two-dimensional space, displacement tensor field has the form referenced in eq. (C.1). In addition, considering mechanical applications with symmetric strain and stress tensors, strain and stress tensor fields are defined in eq. (C.2) and eq. (C.3) respectively. Thus, body force applied to the whole lattice with number of nodes N_x and N_y at each time iteration can be expressed as:

$$\nabla_{\boldsymbol{x}} \cdot (\boldsymbol{\Sigma}) = \begin{bmatrix} \nabla_{\boldsymbol{x}} \cdot (\boldsymbol{\Sigma})\,[0,0] = \nabla_{\boldsymbol{x}} \cdot (\sigma)_{0,0} = \begin{bmatrix} \frac{\partial \sigma_{xx}}{\partial \mathbf{X}} + \frac{\partial \sigma_{xy}}{\partial \mathbf{Y}} \\ \frac{\partial \sigma_{xy}}{\partial \mathbf{X}} + \frac{\partial \sigma_{yy}}{\partial \mathbf{Y}} \end{bmatrix}_{0,0} & \cdots & \nabla_{\boldsymbol{x}} \cdot (\sigma)_{0,N_y-1} = \begin{bmatrix} \frac{\partial \sigma_{xx}}{\partial \mathbf{X}} + \frac{\partial \sigma_{xy}}{\partial \mathbf{Y}} \\ \frac{\partial \sigma_{xy}}{\partial \mathbf{X}} + \frac{\partial \sigma_{yy}}{\partial \mathbf{Y}} \end{bmatrix}_{0,N_y-1} \\ \vdots & \ddots & \vdots \\ \nabla_{\boldsymbol{x}} \cdot (\sigma)_{N_x-1,0} = \begin{bmatrix} \frac{\partial \sigma_{xx}}{\partial \mathbf{X}} + \frac{\partial \sigma_{xy}}{\partial \mathbf{Y}} \\ \frac{\partial \sigma_{xy}}{\partial \mathbf{X}} + \frac{\partial \sigma_{yy}}{\partial \mathbf{Y}} \end{bmatrix}_{N_x-1,0} & \cdots & \nabla_{\boldsymbol{x}} \cdot (\sigma)_{N_x-1,N_y-1} = \begin{bmatrix} \frac{\partial \sigma_{xx}}{\partial \mathbf{X}} + \frac{\partial \sigma_{xy}}{\partial \mathbf{Y}} \\ \frac{\partial \sigma_{xy}}{\partial \mathbf{X}} + \frac{\partial \sigma_{yy}}{\partial \mathbf{Y}} \end{bmatrix}_{N_x-1,N_y-1} \end{bmatrix}, \tag{22}$$

where $\nabla_{\boldsymbol{x}} \cdot (\boldsymbol{\Sigma})$ tensor field is space-time dependent and remains equivalent to a gravitational force in terms of dimensional units. A lattice node (G_x, G_y) is subject to a force estimated partly thanks to a solid constitutive law. This force is then processed depending on lattice units via the call of *ApplyLatticeBodyForce()* that incorporates forces into lattice descriptor in order to modify distribution functions when colliding step is executed.

Strain and stress divergence are calculated, in this paper, with high order finite difference schemes which are constructed on the cell's local vicinity. However, the presented methodology remains general and is true for others numerical schemes able to capture derivatives accurately. Presented numerical operations are local across lattice nodes and allow parallel computing thanks to independent arithmetic. With *GetCellVicinityParameters()* function, information of boundaries and lattice block frontiers are retrieved so as to adapt numerical derivative scheme topology to a specific cell.

4.3. Finite Difference Schemes

We seek for a robust numerical method to evaluate precisely first derivatives of displacement so that a correct strain state is obtained, and the same is requested for stress divergence. Firstly, evaluations that will not amplify non-desired oscillations like local scheme patterns are preferred. WENO[15] [31] schemes were especially developed with this idea. Secondly, we aim to design schemes with small truncation errors in order to minimize deviation during iterations so as to get an acceptable solid static equilibrium. Furthermore, versatile schemes able to capture behavior along boundaries are clearly better suited.

Programming task has been done taking into account successive use of different finite difference models as we move away from boundaries. Both forward and backward formulations are thus adopted, we refer the readers to library sources[30] for more details. For instance, around internal lattice nodes, a central difference scheme with a $\mathcal{O}(\Delta \boldsymbol{x}^4)$ truncation error is built, see eq. (23). First derivative along first spatial direction $\mathbf{X}$ of any function Q with discrete values over the lattice can be expressed as:

$$\frac{\partial Q(G_x, G_y)}{\partial \mathbf{X}} = \frac{Q(G_x-2, G_y) - 8Q(G_x-1, G_y) + 8Q(G_x+1, G_y) - Q(G_x+2, G_y)}{12\Delta \boldsymbol{x}} + \mathcal{O}(\Delta \boldsymbol{x}^4)\,, \tag{23}$$

where (G_x, G_y) denotes a specific lattice node in lattice global coordinates. Since this previous scheme uses a two-neighborhood environment, application near boundaries cannot be applied in this study.

4.4. Proposed solid equilibrium and colliding step

As we aim to study static equilibrium state, we suggest to keep to the BGK collision operator in the VLASOV equation, in order to relax towards a solid equilibrium. Moreover, the reintroduction of the BGK collision operator, enhances the numerical stability and avoid some unwanted mechanical waves or issues near boundaries. This last point is further developed in the next section 4.5. However, our BGK collision operator for solids differs by its equilibrium distribution from the usual one used in CFD.

Consequently, we strive to define an equilibrium that states the solid condition at the molecular scale like the MAXWELL-BOLTZMANN equilibrium distribution function (see eq. (4)) does for gases. Since the static steady state of a solid implies no macroscopic speed, a Maxwellian equilibrium distribution function is imposed with a zero collective velocity. In other words, we tried to reach the static equilibrium

---

[15] Weighted Essentially Non-Oscillatory

through the BGK operator considering an equilibrium distribution without macroscopic speed. A simple distribution that not depends on temperature is targeted and ideally where matter does not move at the microscopic scale level. But, for arithmetic reasons, mass is not concentrated on the cell's first discretized distribution function. Following numerical tests have proven functionality of proposed distribution functions at solid equilibrium. Therefrom, for each cell composing the lattice, distribution at solid static equilibrium for a classic $D2q9$ scheme can be defined by:

$$f_i^{(0),sld}(\rho,\boldsymbol{v}) = f_i^{(0)}(\rho,0) = \rho w_i = \begin{cases} \rho\frac{4}{9} & \text{if } i=0, \\ \rho\frac{1}{9} & \text{if } i\in[1,...,4], \\ \rho\frac{1}{36} & \text{if } i\in[5,...,8]. \end{cases} \tag{24}$$

Where $f_i^{(0),sld}$ represents a discrete value i of $f^{(0),sld}$ for a cell node (G_x, G_y). Presented distribution is applied both for initial conditions and equilibrium during time iterations. Hence, proposed equilibrium does not vary with time and space, and solid behavior has no influence on it. Thus, mass conservation is assured at solid equilibrium thanks to summation of GAUSS-HERMITE weights w_i.

Colliding step (see eq. (9)) is then modified to include above suggested solid equilibrium and related force calculated from stress divergence. For instance, considering an internal node, not on boundaries, discretized collision operator is expressed by:

$$\Omega\left(f_i(G_x,G_y,t), f_i^{(0),sld}\right) = -\omega(f_i(G_x,G_y,t) - f_i^{(0),sld}) + s_i(G_x,G_y,t),\ \forall i\in[0,...,8]. \tag{25}$$

Observing that $f_i(G_x, G_y, t)$ depends on space and time whereas $f_i^{(0),sld}$ is only dependent on lattice scheme properties. Moreover, $s_i(G_x, G_y, t)$ represents the lattice source term related to a body force for a node (G_x, G_y) at time t applied on distribution, see algorithm 3. $s_i(G_x, G_y, t)$ is also scheme dependent since its computation is based, for instance, on GAUSS-HERMITE weights w_i. Different force implementations exist for LBM. All the following illustrated results are made using the GUO approach[28], see eqs. (9) and (15).

Related new objects and class methods are incorporated next to existing PALABOS dynamics and called when a solid study is requested. As we mentioned before, non-intrusive coding work allows to add new solid objects without modifying initial PALABOS sources enabling interoperation of different libraries.

4.5. Proposed modifications for solid boundary conditions

To deal with solids into a LBM framework, it is easier to work on existing boundary conditions. Indeed, solid behavior near domain frontiers must be identified. Our numerical approach can be explained as following. First, numerical runs have pointed out non-desired stress concentration issues in corners or along borders, yielding to important gradients skewing external forces values. Secondly, we aim to distinct boundary behavior with internal one.

Even though, this boundary behavior has not been clearly determined, we aim to suggest a patch treatment for solid matter. For all boundary nodes, we decide to relax them independently toward an equilibrium through algorithm 4, just before the colliding step. Hence, an additional relaxation parameter ω_c is introduced. In other words, along domain borders, a double colliding step is, thus, applied in a cascade manner including one with external force. In practice, we opt for $0.80 < \omega_c \leq 1$ to relax with efficiency.

Concerning non-zero velocity boundary conditions, ω_c is specially chosen to be equal to 1. Indeed, in a solid, no particles are streamed at constrained boundary interfaces. Stress divergence must act as body force in internal nodes and shall govern alone inner behavior. This can be physically interpreted as if non-zero velocity boundary conditions were immediately converted into solid forces during iterations.

Although, all of these conditions are not perfect and subject to improvement, it leads to correct results, see section 5. For more details concerning the implementation, see the available source code[30].

4.6. LBMS*: Loop to Solve Solid Static Equilibrium*

Lattice BOLTZMANN loop is slightly reorganized including above presented additional routines and modifications to perform solid simulations. Main steps are fulfilled by executing in sequence the displacement routine, strain and stress routine, body force routine and finally the special solid boundary task.

To demonstrate the capacity of the present LBM to solve solid static equilibrium, we decide to assign specific boundary conditions. The function *SetBoundaryConditions()* sets a zero-velocity condition along

Algorithm 4: Boundary conditions routine

```
Input: Boundary relaxation parameter $\omega_c$ and solid equilibrium $f^{(0),sld}$
Result: Modification of distribution functions $f_i(G_x, G_y, t)$ on boundaries
for $b \leftarrow 0$ to BoundaryCell.Count()-1 do
    BoundaryCell[b].RelaxBoundaryCell($\omega_c$, $f^{(0),sld}$) ;        // Relax current boundary cell
end
```

related frontiers. In solids, it is similar to fixed boundary conditions if zero velocity does not evolve throughout the simulation time. On the other hand, *SetBoundaryVelocity()* applies a classic velocity condition, which is reduced to zero when variable *MaxIterationBoundaryCondition* is reached. Expected solid static equilibrium is achieved after several supplementary loops. Previous setup exhibits good results for solids, therefore, this is carried out in section 5.

Solid relaxation parameter ω is defined to be very small, to simulate a non-collision condition. Practically, parametric numerical runs revealed that influence of the relaxation parameter can be neglected if $\omega \in]0, ..., 1]$ far from domain borders. Indeed, $\omega = 0.01$ and $\omega = 1$ give almost the same results. This can be explained by the fact there exists a balance between stress divergence and relaxing term.

Algorithm 5: LBMS loop

```
Input: Static equilibrium problem (isotropic materials)
Result: Solved static equilibrium state and related macroscopic variables
Initialization (see library sources[30]) ;                    // Other necessary initializations
Lattice.SetPeriodicity(false) ;                               // Periodicity = false
Lattice.InitializeAtSolidEquilibrium($f^{(0),sld}$) ;         // Solid equilibrium at $t = 0$
Lattice.SetBoundaryConditions() ;                             // Set fixed boundary conditions
Lattice.SetBoundaryVelocity() ;                               // Set velocity boundary condition
Lattice.Execute(algorithm 1) ;                                // Compute displacement field
Lattice.Execute(algorithm 2) ;                                // Compute strain field and stress field
for $i \leftarrow 0$ to $(i \times \Delta t) < MaxStudyTime$ do
    Lattice.Execute(algorithm 3) ;                            // Apply lattice body force
    Lattice.Execute(algorithm 4) ;                            // Relax boundaries
    Lattice.Collide() ;                                       // Standard PaLaBoS colliding step
    Lattice.Stream() ;                                        // Standard PaLaBoS streaming step
    if $i = MaxIterationBoundaryCondition$ then
        Lattice.UnsetBoundaryVelocity() ;                     // Set velocity boundary condition to zero
    end
    Lattice.Execute(algorithm 1) ;                            // Compute displacement field
    Lattice.Execute(algorithm 2) ;                            // Compute strain field and stress field
end
```

We refer the readers, in particular, to the C++ main file *Article.cpp* included in LBMS library which contains all the needed exhaustive details for algorithm 5. Since presented algorithms are based on library sources, thus, useful additional comprehension can be retrieved from this repository.

5. Validations and Benchmarks

To validate previous physical interpretations, LBMS developments and C++ algorithms, results are compared with a FEA software. Reference model has to be accurate and faithful under mechanical framework assumptions. These validations are performed in a two-dimensional space for simplicity and clarity. Throughout this section, all numerical values are given in SI[16] base units if units are not mentioned.

5.1. Mechanical Framework

A strict mechanical framework must be defined so as to identify with fidelity commonalities between the two different approaches in terms of results. To do that, a necessary mechanical background is fixed

---

[16] International System of Units

in this section. This background sets the mechanical assumptions and study properties for both methods enabling relevant comparison.

5.1.1. Mechanical Prerequisites

We resume some aspects partially developed in section 3.3 concerning isotropic linear elastic materials, Eulerian and Lagrangian frameworks. Details are given to highlight with preciseness used type of matter and useful assumptions to validate the LBMS.

First, we point out that stress-strain relation of such isotropic materials is straightforward and well known. This solid behavior is determined with respect to the classic HOOKE's law given in eq. (26) for a lattice node (G_x, G_y). Thus, stress can be determined from the strain with only two scalar values, and these two tensors are symmetric:

$$\mathbf{\Sigma}[G_x, G_y] = C : \mathbf{E}[G_x, G_y] = \lambda \operatorname{Tr}(\mathbf{E}[G_x, G_y])\, I + 2\mu \mathbf{E}[G_x, G_y]. \tag{26}$$

Where C is the corresponding fourth-order isotropic stiffness tensor. Moreover, λ and μ are the LAMÉ coefficients. As mentioned above, $\mathbf{E}$ and $\mathbf{\Sigma}$ are the strain and stress tensor fields defined over the whole lattice.

Secondly, we want to superimpose deformed configuration and initial one. To do this, involved displacement field must be small in terms of norm. Material solid structure is then considered motionless during its transformation. Furthermore, to be able to consider linearized strains, deformation must represent less than 1% of a solid characteristic dimension.

Thirdly, into our process, distinction between the two different descriptions, *i.e.*, Eulerian description (*a.k.a.* spatial coordinates) and Lagrangian description (*a.k.a.* material coordinates) does not apply. LAGRANGE approach is widely preferred for solids where materials are moving with the coordinate system including elements. Basically, it can be explained by the fact that no matter pass from one element to another. On the other hand, in CFD, the mesh or lattice is fixed. Because the lattice structure is supposed stationary throughout time, this no-distinction assumption can be made freely. In substance, considered trajectories of solid particles can be insignificant, thus allowing to merge these two descriptions. Furthermore, all stresses are assumed to be true, *i.e.*, no differences arise from the deformation gradient which is reduced to the identity matrix[32].

5.1.2. Mechanical Study

The comparison study setup is quite simple. All borders are affected with fixed boundary conditions except one that displacement is imposed, see fig. 1 for more details. This enforced displacement is formulated as following:

$$\boldsymbol{u} = 0\mathbf{X} - 0.002\mathbf{Y}. \tag{27}$$

Where we remember that numerical values are expressed in SI base units. $\mathbf{X}$ and $\mathbf{Y}$ represent first and second vectors of the space basis respectively. The LAMÉ coefficients are classically retrieved from material properties, mostly thanks to YOUNG's modulus E and POISSON's ratio ν. Then, we define three cut lines in order to post-process results with parametric curves so as to compare fields with thoroughness, *i.e.*, thanks to C_1, C_2 and C_3.

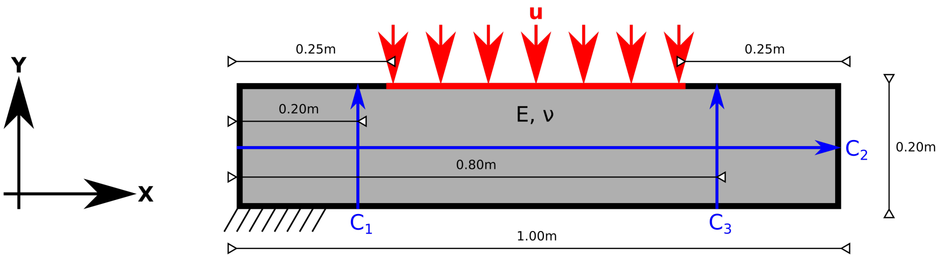

Figure 1: Mechanical study for LBMS approach validation. Black frontiers are fixed boundary conditions. A displacement is imposed across the vector $\boldsymbol{u}$ described with black basis depicted on the left. Solid material properties are defined through YOUNG's modulus E and POISSON's ratio ν. Blue oriented curves C_1, C_2 and C_3 are used for post-processing. All dimensions are given in meters.

5.1.3. LBMS*: setup*

Taking into account previous targeted mechanical study, a lattice characterized by a $D2q9$ scheme is sampled with a resolution of 200 nodes per meter, yielding a suitable discretization. A forced lattice descriptor is adopted in order to manage discrete force values. In all the following cases, a density of $\rho = 1000$ is affected to any cells regardless of the matter nature.

To simulate an imposed displacement condition, a velocity boundary condition is applied during a certain amount of time and then reduced to zero. Previous process allows to reach solid equilibrium which is established after a couple of supplementary iterations following the change in value of imposed macroscopic speed. Velocity field is then stable and no longer evolves, producing next presented results. These results can be all retrieved and re-simulated freely from the C++ main file *Article.cpp* included in sources of the LBMS library.

5.1.4. COMSOL: setup

A benchmark, *i.e.*, a mechanical study which is accurate under assumptions presented above, has to be carefully established. For that purpose, we choose COMSOL[24] as FEA preprocessor and solver for its simplicity of use. However, some simulation options remain important to set even for resolving a standard static problem.

As mentioned above, linearized strains are assumed and by the way forced by default. Because displacement is considered non-existent, all stresses are supposed true. Hence, computed CAUCHY stress can be directly compared with the tensor field $\mathbf{\Sigma}$ from proposed LBMS approach. A plane stress assumption is made so as to simulate a very thin plate. Although, stress-strain relation of this hypothesis is slightly different than the HOOKE's law presented previously, we keep this in mind to investigate results.

A standard extra fine triangular mesh is generated with fully integrated elements using cubic shape functions. This setup enables to compute with accuracy the solution at boundary interfaces or areas subject to stress concentration. More formally, we want a reliable solution that can be examined in detail, *e.g.* near discontinuities produced by the imposed displacement.

Concerning nearly incompressible materials[32–34] later presented in this paper, we use a hybrid element approach, *i.e.*, a mixed formulation that avoids volumetric locking for fully integrated elements. Other methods exist to solve this issue, see *e.g.*[35]. Indeed, due to the constant pressure inside triangles, this numerical phenomenon can arise and may distort the solution. In any cases, the mesh discretization and nature of shape functions further secure the simulation reliability avoiding other locking difficulties. Used triangular elements are rich enough to represent a correct result. Another solution would be to mesh with reduced integration elements where volumetric locking problem does not exist, but a hourglass control becomes necessary.

5.2. Results Comparison with a Commercial Software

Reference FEA studies and associated LBMS simulations are compared with 2D plots and curves. Parametric numerical analysis is led so as to show performance details among a wide range of isotropic materials, *i.e.*, $\nu \in [-0.99, ..., 0.49]$ and $E \in [2\times 10^3, ..., 2\times 10^{11}]$. They show the undeniable capability of the LBM to solve solid mechanics.

5.2.1. Case E=200GPa, ν=0.15

Firstly, we decide to illustrate a specific case, *i.e.*, with material properties E=200GPa, ν=0.15. This example shows typical results of our presented framework. Figure 2 contrasts reference study with proposed LBMS one. Strong similarities and likenesses are observed between the two different approaches. Displacement extremum values remain consistent across the lattice and are directly correlated with the related FEA study. More locally, field values are practically identical and same solid behavior is exhibited. However, we notice unwanted small oscillations of displacement field arising from used finite difference scheme. As we said above, WENO schemes may certainly improve the solution quality. In addition, these perturbations can be drastically reduced by setting a smoother loading profile. Figure 3 shows us deformed shape and displacement field magnitude of the lattice computed with our LBMS approach.

Figures 4 and D.8 depict VON-MISES stress distribution along cutting curves C_1 and C_2 respectively (see fig. 1 for more details concerning C_1 and C_2). Comparing stress values between different codes is often very sensitive. Because we solve for displacement in a FEA approach, stresses values are strongly dependent on computed displacement field accuracy. Plotted curves show a similar global behavior and same order of magnitude (not more than 15% of difference in fig. D.8), although fig. 4 has a striking match with reference study. These first results lead to say that the proposed method has an interesting potential.

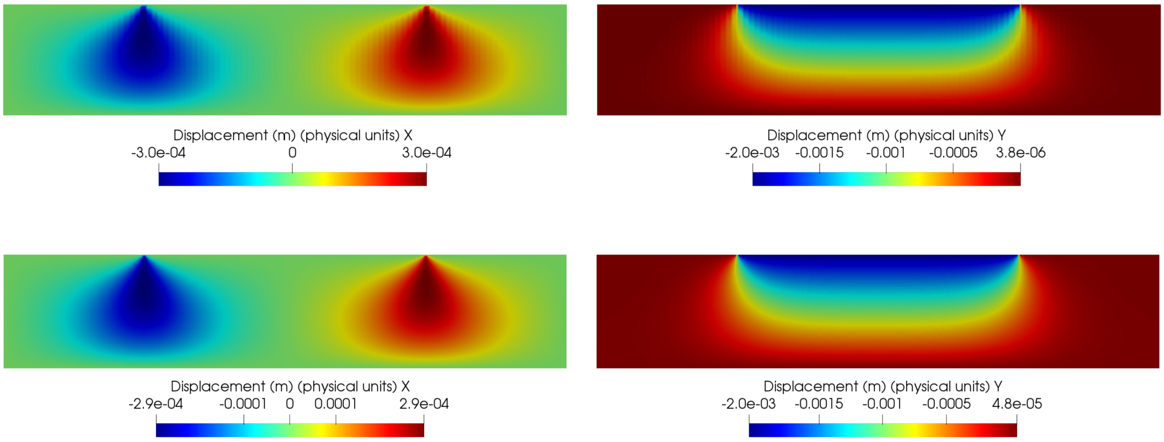

Figure 2: 2D results comparison of displacement field for case E=200GPa, ν=0.15 per field component. Displacement field obtained with our LBMS method (top). Displacement field of reference FEA study computed with COMSOL (bottom).

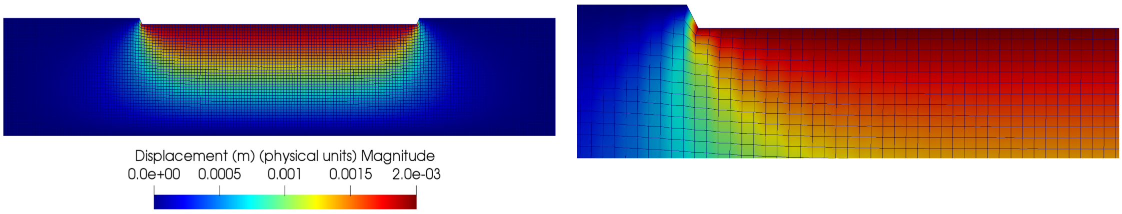

Figure 3: Deformed solid for case E=200GPa, ν=0.15 amplified 5 times with its distorted underlying lattice. Color mapping shows magnitude of displacement field. Overall view of the deformed solid (left). Detailed view along loading (right).

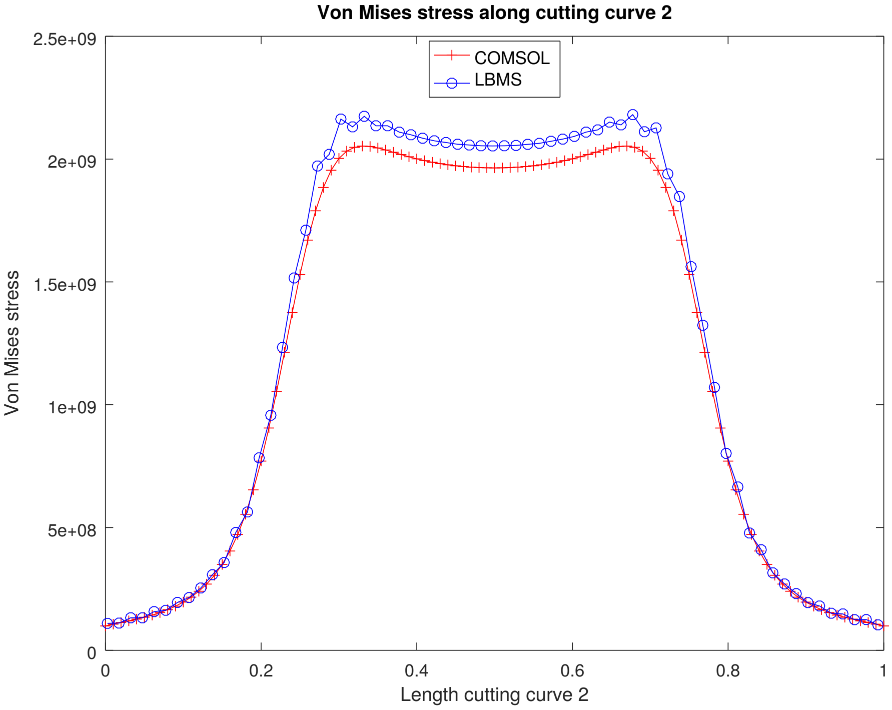

Figure 4: Von-Mises stress along cutting curve C_2 for case E=200GPa, ν=0.15.

5.2.2. POISSON*'s Ratio Sensitivity*

To evaluate the reliability of the proposed method, parametric studies are led with various values of ν with a fixed E=200GPa. For that purpose several curves are analyzed to illustrate the capability of our LBMS method to simulate a wide range of POISSON's ratios. Figures 5, D.9, D.11 and D.16 exhibit results with negative ν whereas figs. D.10, D.12, D.15 and D.17 present positive ones. For $\nu \in [-0.50, ..., 0.35]$ results are satisfying and replicate expected displacement field with a good accuracy and credibility. Concerning highly auxetic materials and nearly incompressible solids, results are mixed. Among all ratios, the general trend still remains analogous. Detailed views along cutting curve C_2 are provided in figs. D.13 and D.14 for first displacement direction and in figs. D.18 and D.19 for the second one.

Shown results are acceptable in terms of matching with reference study. Differences between curves are more pronounced when POISSON's ratio is moving away from zero. Further explanations are brought in detail in section 6.

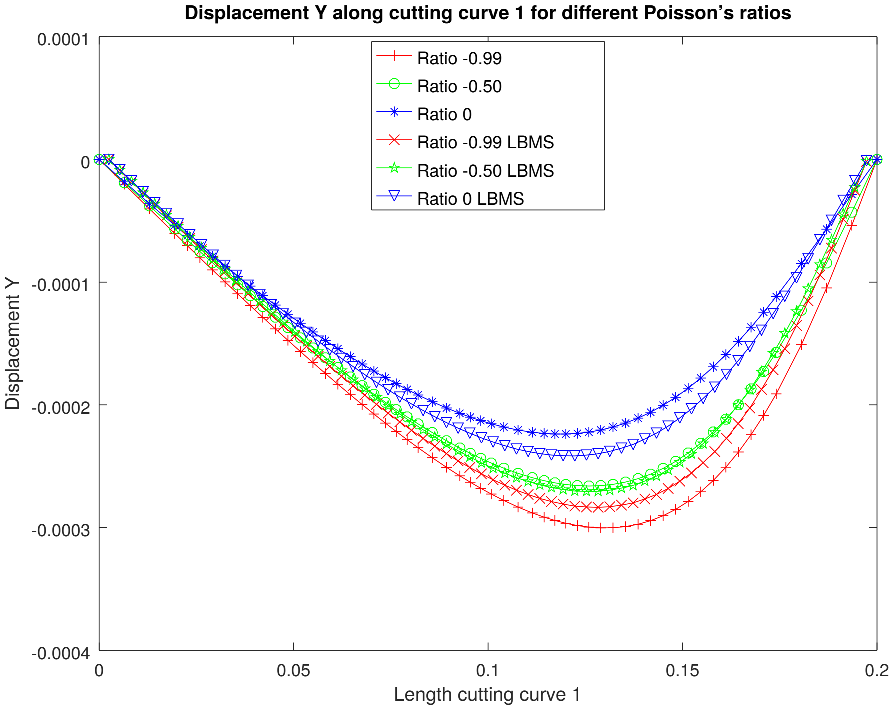

Figure 5: Second displacement components along cutting curve C_1 for negative POISSON's ratios.

5.2.3. YOUNG*'s Modulus Sensitivity*

Like previous part, we perform a parametric study involving YOUNG's modulus. POISSON's ratio is set to zero, leading to $\lambda = 0$. Thus, isotropic law is modified and therefore POISSON effect does not exist for this particular case. Figures D.20 and D.21 illustrate first displacement coordinate along cutting curves C_1 and C_2 respectively. In the same way, figs. D.22 and D.23 depict the second displacement coordinate.

At boundary, displacement is imposed, thus yielding to identical curves no matter is the material stiffness. Indeed, E acts as a stress factor into studied isotropic law (see eq. (26)), *i.e.* only stresses are subject to change. YOUNG's modulus can be viewed as a convergence coefficient during iterations. Investigations are made to demonstrate insensitivity and stability of LBMS with various modulus. To go further, we remark that first component of displacement partly arise from intense shear stress at the frontiers of the velocity condition. A non-constant shear stress along second basis direction produce a non-zero stress divergence, *i.e.* contributing to its first component and hence yield a displacement even if $\nu = 0$.

5.2.4. Triangular Loading: Case E=200GPa, ν=0.15

To exhibit robustness of the LBMS method and its ability to compute solid static equilibrium, a last example is illustrated with a different boundary condition. We resume previous case in section 5.2.1, but a right-angled triangular imposed displacement is applied (with the maximum displacement localized at the right of the considered problem) instead straight one (see fig. 1). Figure 6 presents 2D results compared with reference study. In general, the displacement field is consistent and replicates expected solid behavior with fidelity. In details, minor errors are detected in field values leading to same conclusions as above. Additional curves are given in figs. D.24 and D.25 for Von-Mises stress and in figs. 7 and D.26 to D.28 for displacement components along cutting curves C_2 and C_3. Plots are in total concordance with related FEA study despite slight deviations when displacement is important.

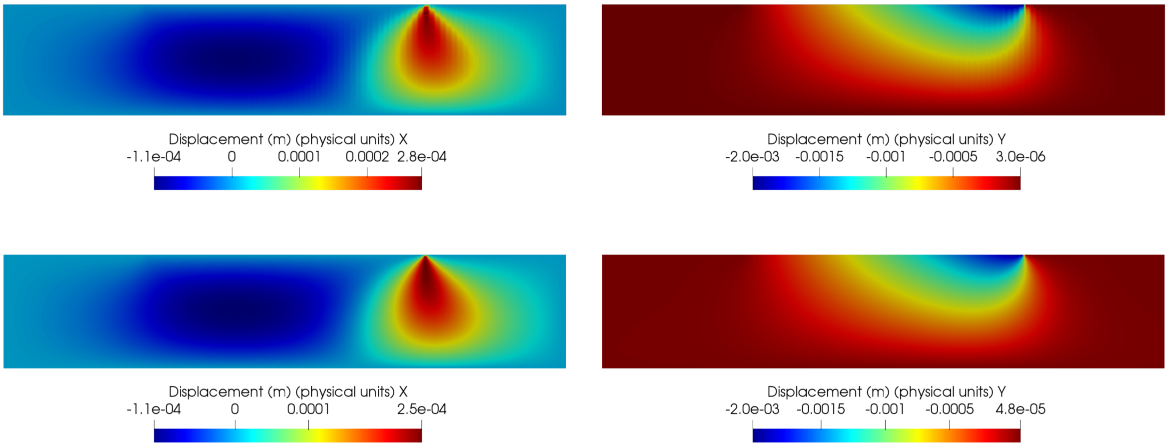

Figure 6: 2D results comparison of displacement field for case E=200GPa, ν=0.15 per field component and with a triangular loading profile. Displacement field obtained with our LBMS method (top). Displacement field of reference FEA study computed with COMSOL (bottom).

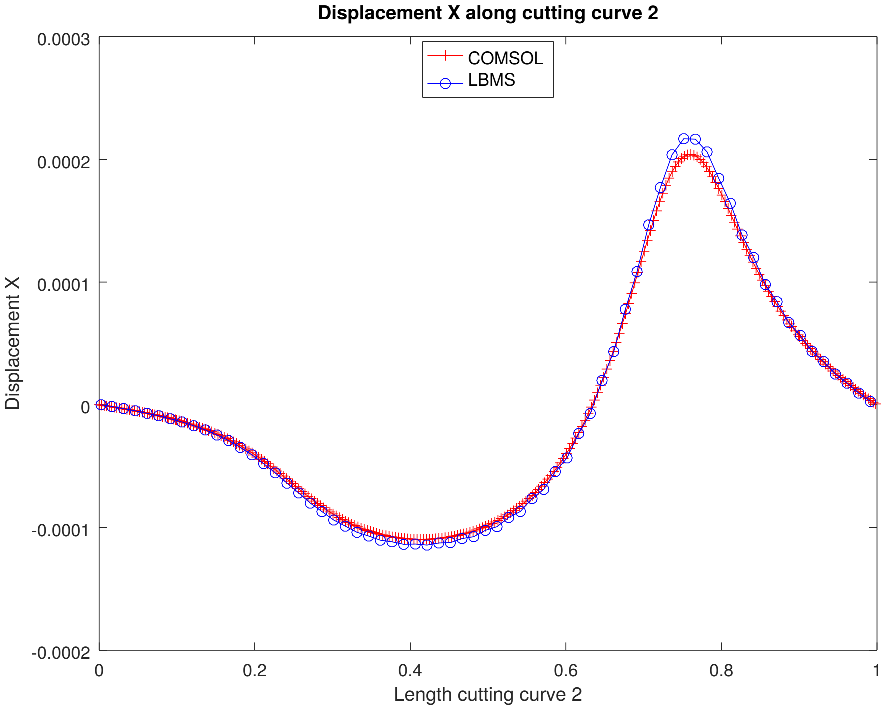

Figure 7: First displacement component along cutting curve C_2 for case E=200GPa, ν=0.15 with triangular loading.

6. Discussions and Outlooks

6.1. *Obtained Results Discussion*

In addition to the global curves matching which gives a qualitative validation, a quantitative analysis is provided. Thus, table 1 shows the RMSE[17] and NRMSE[18] between the computed results with the LBMS and FEA, presented through figs. 4, 5 and 7. The results show a great precision of the LBMS compared to FEA method, and this for a large number of study cases. This precision illustrates the fidelity of the LBMS. The numerical errors obtained are far lower than the amplitude of computed fields, which yields only few percents of error. These errors values tend to validate quantitatively the presented approach.

The numerical tests highlight the stability and accuracy of the suggested methods in a very broad range of mechanical parameters. This large range lets us think that the current method could be applied to numerous constitutive laws, and not only to the isotropic linear elasticity. This wide range tends to support the theoretical development brought in this paper, especially the incorporation of the stress divergence as a mean-field force.

Loading	Curve	Field	E (GPa)	ν	RMSE	NRMSE
Rectangular	C_2 (fig. 4)	Von-Mises stress	200	0.15	$7.37.10^{+7}$	3.77%
Rectangular	C_1 (fig. 5)	Displacement **Y**	200	-0.99	$1.52.10^{-5}$	5.01%
Rectangular	C_1 (fig. 5)	Displacement **Y**	200	-0.5	$4.41.10^{-6}$	1.66%
Rectangular	C_1 (fig. 5)	Displacement **Y**	200	0.0	$1.23.10^{-5}$	5.50%
Triangular	C_2 (fig. 7)	Displacement **X**	200	0.15	$1.52.10^{-5}$	2.01%

Table 1: RMSE and NRMSE of the displacement and Von-Mises stress obtained for different study cases.

Concerning introduced boundary conditions for solids, a certain work remains in order to enhance adaptation for solids. Currently, it has been pointed out that fixed conditions are far to be perfect and suitable to accurate computation near concerned areas: impreciseness on fields can arise due to this modeling issue.

Light differences with reference plots may have several explanations. First and foremost, LBMS method detains an iterative process so as to arrive to solid static equilibrium, thus, finite difference scheme accuracy could play an important role concerning quality of calculated results. Moreover, we point out some post-processing issues which may be at origin of errors. Indeed, along cutting curves, FEA displacement field is mainly interpolated while LBMS field hardly is. All of these considerations can cause additional deviations with reference study and may explain some of illustrated differences.

Other slight errors may be generated by the non-differentiation between Eulerian and Lagrangian frameworks. Several iterations are needed until equilibrium state (*i.e.* when stress divergence tends towards a null vector), yielding to completely different pipelines between FEA and LBMS. Furthermore, as mentioned above in section 5.1.4, used plane stress assumption for two-dimensional reference studies might trigger some of shown deviations in curve plots, especially for non-zero Poisson's ratios. Indeed, thin plates are usually modeled with a plane stress hypothesis, but this approach is softly inconsistent with the one used to produce illustrated LBMS runs, *i.e.* a plane strain assignment. For all of these reasons, extra investigations seem to be mandatory to fit calculated reference fields with a better accuracy.

All these achievements seem promising about the use of Boltzmann-Vlasov approach, presented here, for solid dynamics. It also worth noting that the mass of the system is always conserved. Indeed, the intrinsic advantages of the LBM are preserved by the LBMS (macroscopic equations recovered from the statistical moments, locality, simplicity *etc.*). By contrast with previous attempts to cope with solid behavior with the LBM, our approach does not require statistical distribution of forces[14,36] nor springs[15]. Both bring more complexity, heavier computations, are further away from the Boltzmann theory and are less prone to FSI coupling.

6.2. *Global outlook and proposed improvements*

C++ programmed classes and methods integrated in LBMS library can be drastically enhanced, *i.e.* mostly for efficiency purposes. Although library is robust, computation time might be substantially reduced if other guidelines are adopted. For presented cases, tens or hundreds of seconds are necessary

---

[17] Root Mean Square Error

[18] Normalized Root Mean Square Error

to reach illustrated static equilibriums assuming sequential runs on a Intel(R) Xeon(R) CPU E5-1650 v3 @ 3.50GHz. At contrary, couple of seconds are enough to COMSOL even if mesh is modeled very fine, because a standard static resolution is requested. Effectively, involved computation times are hardly comparable since methods are strongly different. In fact, LBMS is naturally dynamic due to its roots coming from the LBM and thus require an evaluation of solid state at each time iteration. Thanks to the pointed reproducibility of depicted numerical results, complements to existing objects can be made freely in order to improve calculation efficiency. However, here stand one of the great strengths of the LBM: parallel computing can be seen as one of the main advantages of LBM. The LBM is intrinsically efficient and adapted to massive parallel computing, and especially on GPU[19] because of its architecture. In fact, a GPU detains hundreds or thousands of cores capable of doing very fast executions despite management of few instructions and small caches.

Finite difference schemes are playing an important role, in fact they define external forces to be applied at each iteration for each cell. Upgrading abilities to treat mechanical singularities and enhancing accuracy under high field variation are a future work. For example, corners demand a special therapy since cell vicinity is restricted. A possible other way is to use robust mean-field forces as already implemented into the SHAN-CHEN approach[8]. Moreover, as mentioned above, WENO schemes could help us.

Another advantage of the LBM compared to FEA, is this ease of going from lower to higher dimensions. The transition from the 2D modeling to the 3D and 4D modeling is implied in the method. Then, LBMS extension to 3D should be straightforward and large-scale studies can be led from now on. Such computations permit to better compare complex geometries with current methods usually dedicated to solve mechanical problems. Door is thus wide open in terms of further intensive analysis or significant studies. In addition, parallelized strategy by blocks remains unchanged for solids. But, depending on used finite difference schemes, some complementary properties are needed for block envelopes.

Besides, the broad range of YOUNG's modulus and POISSON's ratios simulated, the presented method has to be compared with non-linear constitutive laws, to confirm its potentiality and generality. Based on the strength of the presented results, a very straightforward extension is to investigate quasi-static dynamic problems. Of course, this eventuality could only be viable if the computation time of the LBMS can be deeper characterized.

Only the static equilibrium has been compared to existing methods, in this first study. Since this static equilibrium has been reached for different relaxation time, in particular for $\omega = 0.01$; solid dynamics with the BOLTZMANN-VLASOV equation seems a natural following. The numerical stability for relaxation parameter $\omega < 0.25$ is already unconventional LBM and supports the theoretical developments brought here.

As we mentioned before, the goal of the solid behavior with the LBM, might also be investigated through the research of an appropriate equilibrium distribution function. To go further with that idea, if $\omega = 1$, we also observe that (see eq. (25)):

$$\Omega\left(f_i(G_x, G_y, t), f_i^{(0),sld}, s_i(G_x, G_y, t)\right) = -\omega f_i(G_x, G_y, t) + \omega(f_i^{(0),sld} + s_i(G_x, G_y, t)),\ \forall i \in [0, ..., 8]. \tag{28}$$

This latter formulation yields a new form of solid equilibrium where solid body forces are integrated into equilibrium function; thus stress divergence term may be integrated into the BGK operator if the relaxation parameter is equal to 1. This previous equilibrium distribution deserves further dedicated investigations, beyond the paper.

Of course, many applications are plausible from the next development of the LBMS, especially about the interface and boundary conditions. By solving the solid interface, the motion and deformation of solids under the action of forces and contact problems between two solids could be achieved. Then, solid-liquid interface leads to FSI for studying the behavior of a solid product immersed in a liquid. Recent publication[37] illustrates the fact that a unified theory is needed for solving complex FSI phenomena taking in consideration both fluids and solids characteristics with LBM.

7. Conclusions

We have introduced in this paper a new method to solve solid static equilibriums from standard LBM. Actually, our strategy depicts promising results concerning static state of isotropic materials. Shown plots and figures illustrate well efficiency of the presented method. Such task is achieved thanks

---

[19] Graphics Processing Unit

to detailed developments and robust framework. In fact, they are secured with a theory groundwork and then implemented using classical programming tools.

A first key contribution is introduced thanks to theoretical developments around the BOLTZMANN-VLASOV equation. Furthermore, the stress tensor divergence is introduced as a mean-field force term. These two considerations allow to build a theoretical framework to deal with solids with the LBM. This framework is unprecedented and far from the previous works on the subject.

Then, we gave new programmed objects wrapped into an user-friendly research library. Although this implementation is not optimal, developed classes and methods are robust enough and trustworthy concerning conclusions given on plots and figures. On the other hand, adopted general implementation approach detains several advantages. Non-invasive coding work permits to reproduce results with a bewildering ease. Likewise, light program environment enables easy-way additions for future improvements. Moreover, the standard LBM main loop is still not modified, *i.e.* only additional routines to treat solids are added. Basically, this means the colliding-streaming strategy is kept without compromise, just the dynamic related to solids is adjusted. Previous ability authorize, among others, to model fluids with solids in the same framework.

Shown results are validated through comparative plots including a large range of parametric studies. They clearly state credibility of proposed approach. Besides, for a significant number of plotted data, matching with reference analysis is impressive. In all cases, exhibited mapped fields are globally consistent and replicate a real solid matter's comportment. The great precision of the results, not only highlights the capacity of the method to capture isotropic elasticity; but also let us imagine a complete confirmation of the general theory developed. Moreover, the reliable and stable results obtained at very high relaxation time ($\omega \leq 0.01$) shows the strength of the theory and the perspectives for solid dynamics.

Despite promising illustrated results contrasted with reference studies, a lot of work still needs to be done to exploit presented potential of the LBMS. Several aspects still require further afterthought so as to simulate proper solid behaviors. Indeed, modeling solids with LBM is still a challenge among scientific communities even though advances are presented in this paper. This work also tries to answer increasing expectations in numerical simulation fields.

Beyond all these obtained results, LBMS seems to by very hopeful to simulate solids into an Eulerian framework. This work aims to bring new tools to push the limits of computational mechanics even further. Biomechanics is facing difficult problems such as large deformations, non-linear behaviors, complex geometries and FSI. Standard approaches detain several drawbacks in order to solve previous involved physics. LBM methods are, thus, naturally better suited to tackle these mentioned difficulties.

Solid-solid contact problems is another perspective of the LBMS. It seems that the formulation of contacts is less difficult on the geometrical aspect in an Eulerian framework but will be more complex regarding formulation of the interaction equations between different objects.

A large spectrum of perspectives is thus opened by the presented method. As discussed above, quasi-static studies are the next step for investigating computational solid dynamics. In contrast, a real dynamic case may still request supplementary efforts. Moreover, complex solid constitutive laws, such as non-linear ones have to be tested and contrasted with adapted reference results. In addition, extension to 3D should be natural and in consequence, could produce relevant results. As suggested in this paper, ideas related to solid distribution functions were given, but it undeniably needs further work so as to get deeper comprehension on solid equilibrium. For all of these reasons, we aim to share presented advances for the benefit of the community.

Acknowledgments

This work has been partially funded by the French National Research Agency (Agence Nationale de la Recherche) via the LBSMI project ANR-15-CE19-0002. This support is gratefully acknowledged.

Appendix A. From CHAPMAN-ENSKOG Expansion to NAVIER-STOKES-FOURIER Equations

To go from EULER conservation equations to NSF equations with the BE, it is convenient to use the CHAPMAN-ENSKOG expansion[26] (but it is not the only possible path[38]). By introducing, an asymptotic development in order of small perturbations for the distribution density, it becomes:

$$f = f^{(0)} + \epsilon^1 f^{(1)} + ... = \sum_{r=0}^{\infty} \epsilon^r f^{(r)}, \tag{A.1}$$

with $f^{(1)}$ the first order of the out-of-equilibrium distribution. ϵ is a small parameter compared to 1 and it is related to the size of the disturbances and the KNUDSEN number. A similar development must be introduced for derivation operations, hence:

$$\partial_t = \epsilon^1\partial_t^{(1)} + \epsilon^2\partial_t^{(2)} + ... = \sum_{k=1}^{\infty}\epsilon^k\partial_t^{(k)} \qquad \partial_{\boldsymbol{x}} = \epsilon^1\partial_{\boldsymbol{x}}^{(1)} + ... = \sum_{k=1}^{\infty}\epsilon^k\partial_{\boldsymbol{x}}^{(k)}. \tag{A.2}$$

To reduce the complexity of the following development, the linearize BGK operator is used, however, the results remain true for the general collision operator. Indeed, the classical global solution to the incompressible NAVIER-STOKES-FOURIER equation with small initial data in the whole space is constructed through a zero KNUDSEN number limit with general collision kernels[27]. Using the linearized BGK operator, the BOLTZMANN-BGK equation (see eq. (3)), becomes:

$$\sum_{\substack{k=1,\\ r=0}}^{\infty}\epsilon^{k+r}\partial_t^{(k)}f^{(r)} + \sum_{\substack{k=1,\\ r=0}}^{\infty}\epsilon^{k+r}\boldsymbol{\xi}\cdot\nabla_{\boldsymbol{x}}^{(k)}f^{(r)} + \sum_{\substack{k=1,\\ r=0}}^{\infty}\epsilon^{k+r}\boldsymbol{g}\cdot\nabla_{\boldsymbol{\xi}}^{(k)}f^{(r)} = -\omega\left(\sum_{r=1}^{\infty}\epsilon^r f^{(r)}\right), \tag{A.3}$$

which enables a resolution in the order of ϵ.

At first order, the out-of-equilibrium distribution can be expressed through the following equation:

$$\partial_t^{(1)}f^{(0)} + \boldsymbol{\xi}\cdot\nabla_{\boldsymbol{x}}^{(1)}f^{(0)} + \boldsymbol{g}\cdot\nabla_{\boldsymbol{\xi}}^{(1)}f^{(0)} = -\omega f^{(1)}. \tag{A.4}$$

Since the equilibrium distribution function dependency in time and space is only through the variables $\rho(\boldsymbol{x},t)$, $\boldsymbol{v}(\boldsymbol{x},t)$ and $\theta(\boldsymbol{x},t)$; the use of the chain rule is needed. Thus, by forgetting the derivatives superscript for the sake of simplicity:

$$\partial_t f^{(0)} = \partial_\rho f^{(0)}\partial_t\rho + \partial_{\boldsymbol{v}}f^{(0)}\partial_t\boldsymbol{v} + \partial_\theta f^{(0)}\partial_t\theta, \tag{A.5a}$$

$$\partial_{\boldsymbol{x}} f^{(0)} = \partial_\rho f^{(0)}\partial_{\boldsymbol{x}}\rho + \partial_{\boldsymbol{v}}f^{(0)}\partial_{\boldsymbol{x}}\boldsymbol{v} + \partial_\theta f^{(0)}\partial_{\boldsymbol{x}}\theta. \tag{A.5b}$$

The temporal derivatives can be obtained thanks to the EULER equations, after some algebra, it leads to:

$$\partial_t\rho = -\nabla_{\boldsymbol{x}}\cdot(\rho\boldsymbol{v}), \tag{A.6a}$$

$$\partial_t\boldsymbol{v} = -\frac{1}{\rho}\nabla_{\boldsymbol{x}}\cdot(p) + \boldsymbol{g}, \tag{A.6b}$$

$$\partial_t\theta = -\boldsymbol{v}\cdot(\nabla_{\boldsymbol{x}}\theta) - \frac{2}{D}\theta(\nabla_{\boldsymbol{x}}\boldsymbol{v}). \tag{A.6c}$$

The derivatives of the equilibrium distribution function are easily obtained and give:

$$\partial_\rho f^{(0)} = \frac{f^{(0)}}{\rho}, \tag{A.7a}$$

$$\partial_{\boldsymbol{v}} f^{(0)} = \frac{f^{(0)}(\boldsymbol{\xi}-\boldsymbol{v})}{\theta}, \tag{A.7b}$$

$$\partial_{\boldsymbol{\xi}} f^{(0)} = \frac{-f^{(0)}(\boldsymbol{\xi}-\boldsymbol{v})}{\theta}, \tag{A.7c}$$

$$\partial_\theta f^{(0)} = f^{(0)}\left(\frac{(\boldsymbol{\xi}-\boldsymbol{v})^2}{2\theta^2} - \frac{D}{2\theta}\right). \tag{A.7d}$$

Once the previous systems eq. (A.6) and eq. (A.7) are injected and eq. (A.5) are injected in the expression of $f^{(1)}$ given by eq. (A.4), it becomes:

$$\begin{aligned}
&-\omega f^{(1)}\\
&=\left[\frac{f^{(0)}}{\rho}\right]\left[-\nabla_{\boldsymbol{x}}\cdot(\rho\boldsymbol{v}) + (\boldsymbol{\xi}\cdot\nabla_{\boldsymbol{x}})\rho\right] + \left[\frac{f^{(0)}(\boldsymbol{\xi}-\boldsymbol{v})}{\theta}\right]\cdot\left[-\frac{1}{\rho}\nabla_{\boldsymbol{x}}(p) + \boldsymbol{g} + (\boldsymbol{\xi}\cdot\nabla_{\boldsymbol{x}})\boldsymbol{v}\right]\\
&\quad+\left[f^{(0)}\left(\frac{(\boldsymbol{\xi}-\boldsymbol{v})^2}{2\theta^2} - \frac{D}{2\theta}\right)\right]\left[-\boldsymbol{v}\cdot(\nabla_{\boldsymbol{x}}\theta) - \frac{2}{D}\theta(\nabla_{\boldsymbol{x}}\boldsymbol{v}) + (\boldsymbol{\xi}\cdot\nabla_{\boldsymbol{x}})\theta\right] - \boldsymbol{g}\cdot\frac{f^{(0)}(\boldsymbol{\xi}-\boldsymbol{v})}{\theta}\\
&=\left[f^{(0)}\right]\left[\left(\boldsymbol{c}\otimes\boldsymbol{c} - \frac{\boldsymbol{c}^2}{D}I\right)\frac{\nabla_{\boldsymbol{x}}\boldsymbol{v}}{\theta} + \left(\frac{\boldsymbol{c}^2}{2\theta} - \frac{D+2}{2}\right)\frac{\boldsymbol{c}\cdot\nabla_{\boldsymbol{x}}\theta}{\theta}\right],
\end{aligned} \tag{A.8}$$

where $\boldsymbol{c} = (\boldsymbol{\xi} - \boldsymbol{v})$ is the microscopic variation of particles velocities around the mean value.

Therefore, it is possible to determine the moments of the none-equilibrium distribution function $f^{(1)}$, by reminding that consecutive centered moments of the equilibrium distribution function are:

$$\begin{aligned}
\int_{\mathbb{R}} f^{(0)}\,\mathrm{d}\boldsymbol{\xi} &= \rho, && \text{(A.9a)}\\
\int_{\mathbb{R}} \boldsymbol{\xi} f^{(0)}\,\mathrm{d}\boldsymbol{\xi} &= \rho\boldsymbol{v}, && \text{(A.9b)}\\
\int_{\mathbb{R}} \boldsymbol{c}\otimes\boldsymbol{c} f^{(0)}\,\mathrm{d}\boldsymbol{\xi} &= \rho R\theta I, && \text{(A.9c)}\\
\int_{\mathbb{R}} \boldsymbol{c}\otimes\boldsymbol{c}\otimes\boldsymbol{c} f^{(0)}\,\mathrm{d}\boldsymbol{\xi} &= 0, && \text{(A.9d)}\\
\int_{\mathbb{R}} \boldsymbol{c}\otimes\boldsymbol{c}\otimes\boldsymbol{c}\otimes\boldsymbol{c}\; f^{(0)}\,\mathrm{d}\boldsymbol{\xi} &= \rho R^2\theta^2\mathcal{P}_3(I\otimes I), && \text{(A.9e)}\\
\int_{\mathbb{R}} \boldsymbol{c}\otimes\boldsymbol{c}\otimes\boldsymbol{c}\otimes\boldsymbol{c}\otimes\boldsymbol{c}\; f^{(0)}\,\mathrm{d}\boldsymbol{\xi} &= 0, && \text{(A.9f)}\\
\int_{\mathbb{R}} \boldsymbol{c}\otimes\boldsymbol{c}\otimes\boldsymbol{c}\otimes\boldsymbol{c}\otimes\boldsymbol{c}\otimes\boldsymbol{c}\; f^{(0)}\,\mathrm{d}\boldsymbol{\xi} &= \rho R^3\theta^3\mathcal{P}_{15}(I\otimes I\otimes I), && \text{(A.9g)}
\end{aligned}$$

where $\mathcal{P}_n(\cdot)$ is an operator yielding the sum of the n cyclic permutations. Thus, these moments of $f^{(1)}$ are given by:

$$\begin{aligned}
\int_{\mathbb{R}} f^{(1)}\,\mathrm{d}\boldsymbol{\xi} &= 0, && \text{(A.10a)}\\
\int_{\mathbb{R}} \boldsymbol{\xi} f^{(1)}\,\mathrm{d}\boldsymbol{\xi} &= 0, && \text{(A.10b)}\\
\int_{\mathbb{R}} \boldsymbol{c}\otimes\boldsymbol{c} f^{(1)}\,\mathrm{d}\boldsymbol{\xi} &= -\frac{\rho\theta}{\omega}\left[\frac{1}{2}\left(\nabla_{\boldsymbol{x}}\boldsymbol{v} + \nabla_{\boldsymbol{x}}\boldsymbol{v}^T\right) - \frac{2}{D}(\nabla_{\boldsymbol{x}}\cdot\boldsymbol{v})I\right], && \text{(A.10c)}\\
\int_{\mathbb{R}} \boldsymbol{c}^2 f^{(1)}\,\mathrm{d}\boldsymbol{\xi} &= 0, && \text{(A.10d)}\\
\int_{\mathbb{R}} \boldsymbol{c}^2\boldsymbol{c} f^{(1)}\,\mathrm{d}\boldsymbol{\xi} &= -\frac{\rho\theta}{2\omega}\left[(D+2)\nabla_{\boldsymbol{x}}\theta\right], && \text{(A.10e)}
\end{aligned}$$

where I is the identity matrix. The three null moments are consistent with the invariant of the collision operator.

One can finally obtain the NSF equations by using the BE at first order:

$$\begin{aligned}
\partial_t\rho + \nabla_{\boldsymbol{x}}\cdot(\rho\boldsymbol{v}) &= 0, && \text{(A.11a)}\\
\partial_t(\rho\boldsymbol{v}) + \nabla_{\boldsymbol{x}}\cdot(\rho\boldsymbol{v}\otimes\boldsymbol{v} + p - \nu\nabla_{\boldsymbol{x}}\boldsymbol{v}) &= 0, && \text{(A.11b)}\\
\partial_t(E_k + E_\theta) + \nabla_{\boldsymbol{x}}\cdot((E_k + E_\theta)\,\boldsymbol{v} + p\cdot\boldsymbol{v} + \kappa\nabla_{\boldsymbol{x}}\theta) &= 0, && \text{(A.11c)}
\end{aligned}$$

where ν is the viscosity of the fluid that can be expressed by $\nu = \frac{\rho\theta}{2\omega}$ and $\kappa = \frac{\rho\theta(D+2)}{2\omega}$.

Appendix B. LBM at Second Order Precision

Before or after the velocity space discretization, one can start the discretization in space and time. These two dimensions (space and time) have to be discretized together because of the convective nature of the BE. Even if, the usual discretizations are accomplished on a regular grid, some works developed a LBM on an unstructured mesh[39,40].

A first idea to do this discretization could be a first-order finite difference. So, one could rewrite the BE discretized over the velocity space eq. (6) as:

$$\frac{\partial f_i}{\partial t}(\boldsymbol{x},t) + \boldsymbol{\xi}_i\cdot\nabla_{\boldsymbol{x}}(f_i) = \frac{D}{Dt}f_i(\boldsymbol{x},t) = \Omega_i(f_i,f_i) - g_i, \qquad \text{(B.1)}$$

where $\frac{D}{Dt}$ is the particle derivative. A second-order discretization can be easily obtained[41]. Indeed, using a well-chosen surrogate, one can use the LBE[20] at second order. To do so, a direct integration of the

---

[20] Lattice Boltzmann Equation

eq. (B.1) with the method of characteristics yields:

$$f_i(\boldsymbol{x}+\boldsymbol{\xi}_i\Delta t, t+\Delta t) - f_i(\boldsymbol{x},t) = \int_0^{\Delta t} \widehat{\Omega}_i(t+s)\,\mathrm{d}s, \tag{B.2}$$

where $\widehat{\Omega}_i(t) = \Omega\left(f_i, f_i\right) - g_i$. Then, using a trapezoidal rule on the right-hand side, one obtains:

$$f_i(\boldsymbol{x}+\boldsymbol{\xi}_i\Delta t, t+\Delta t) - f_i(\boldsymbol{x},t) = \frac{\Delta t}{2}\left(\widehat{\Omega}_i(t+\Delta t) + \widehat{\Omega}_i(t)\right) + \mathcal{O}\left(\Delta t^3\right). \tag{B.3}$$

This approximation is now at the second order but implicit. Nevertheless, thanks to the appropriate change of variable:

$$\widehat{f}_i(\boldsymbol{x},t) = f_i - \frac{\Delta t}{2}\widehat{\Omega}_i(t), \tag{B.4}$$

one can find the following explicit expression:

$$\widehat{f}_i(\boldsymbol{x}+\boldsymbol{\xi}_i\Delta t, t+\Delta t) - \widehat{f}_i(\boldsymbol{x},t) = \Delta t\widehat{\Omega}_i(t) + \mathcal{O}\left(\Delta t^3\right). \tag{B.5}$$

Thus, one can recognize the LBE for the new variable $\widehat{f}_i$. To dive deeper, the BGK operator needs to be exploited. For the BGK operator the previous equation can be written:

$$\widehat{f}_i(\boldsymbol{x},t) = f_i + \frac{\Delta t\omega}{2}\left(f_i(\boldsymbol{x},t) - f_i^{(0)}(\boldsymbol{x},t)\right) - \frac{\Delta t}{2}g_i(\boldsymbol{x},t), \tag{B.6}$$

$$\widehat{f}_i(\boldsymbol{x}+\boldsymbol{\xi}_i\Delta t, t+\Delta t) - \widehat{f}_i(\boldsymbol{x},t) = -\Delta t\omega\left(f_i(\boldsymbol{x},t) - \widehat{f_i^{(0)}}(\boldsymbol{x},t)\right), \tag{B.7}$$

where $\widehat{f_i^{(0)}}(\boldsymbol{x},t) = f_i^{(0)}(\boldsymbol{x},t) - \frac{g_i}{\omega}$. Therefore, to work only with the new variable, the change of variable can be inverted, which yield:

$$f_i(\boldsymbol{x},t) = \frac{\frac{2}{\omega}\widehat{f}_i(\boldsymbol{x},t) + \Delta t\widehat{f_i^{(0)}}(\boldsymbol{x},t)}{\frac{2}{\omega} + \Delta t}. \tag{B.8}$$

In the end, injecting eq. (B.8) into eq. (B.5) one can obtain the LBGKE:

$$\begin{aligned} \widehat{f}_i(\boldsymbol{x}+\boldsymbol{\xi}_i\Delta t, t+\Delta t) &= \widehat{f}_i(\boldsymbol{x},t) - \Delta t\widehat{\omega}\left(\widehat{f}_i(\boldsymbol{x},t) - \widehat{f_i^{(0)}}(\boldsymbol{x},t)\right) + \mathcal{O}\left(\Delta t^3\right), \\ &= \widehat{f}_i(\boldsymbol{x},t) - \Delta t\widehat{\omega}\left(\widehat{f}_i(\boldsymbol{x},t) - f_i^{(0)}(\boldsymbol{x},t)\right) - \Delta t\left(1 - \frac{\widehat{\omega}}{2}\right)g_i(\boldsymbol{x},t) + \mathcal{O}\left(\Delta t^3\right), \end{aligned} \tag{B.9}$$

where $\frac{1}{\widehat{\omega}} = \frac{1}{\omega} + \frac{\Delta t}{2}$. So, the simulation will be performed over $\widehat{f}_i$ rather than f_i. This change of variable is necessary to obtain a second order accuracy. For sake of simplicity, in this paper the following substitutions are performed: $\widehat{f}_i \to f_i$ and $\widehat{\omega} \to \omega$, which means that the reader should read $\widehat{f}_i$ rather than f_i and $\widehat{\omega}$ rather than ω.

Appendix C. Tensors details

Used tensors in algorithms are detailed here for symmetric strain and stress tensors related to 2D applications. They are defined across the whole lattice and thus contain information for all cells. Displacement tensor field has the following form:

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}[0,0] = \boldsymbol{u}_{0,0} = \begin{bmatrix}\boldsymbol{u}_x\\ \boldsymbol{u}_y\end{bmatrix}_{0,0} & \cdots & \boldsymbol{u}_{0,N_y-1} = \begin{bmatrix}\boldsymbol{u}_x\\ \boldsymbol{u}_y\end{bmatrix}_{0,N_y-1} \\ \vdots & \ddots & \vdots \\ \boldsymbol{u}_{N_x-1,0} = \begin{bmatrix}\boldsymbol{u}_x\\ \boldsymbol{u}_y\end{bmatrix}_{N_x-1,0} & \cdots & \boldsymbol{u}_{N_x-1,N_y-1} = \begin{bmatrix}\boldsymbol{u}_x\\ \boldsymbol{u}_y\end{bmatrix}_{N_x-1,N_y-1} \end{bmatrix}. \tag{C.1}$$

Then, the strain tensor field is calculated thanks to displacement one:

$$\mathbf{E} = \begin{bmatrix} \mathbf{E}[0,0] = \varepsilon_{0,0} = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix}_{0,0} & \cdots & \varepsilon_{0,N_y-1} = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix}_{0,N_y-1} \\ \vdots & \ddots & \vdots \\ \varepsilon_{N_x-1,0} = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix}_{N_x-1,0} & \cdots & \varepsilon_{N_x-1,N_y-1} = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix}_{N_x-1,N_y-1} \end{bmatrix}. \tag{C.2}$$

The stress tensor field is obtained through a constitutive law and reads:

$$\mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}[0,0] = \sigma_{0,0} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix}_{0,0} & \cdots & \sigma_{0,N_y-1} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix}_{0,N_y-1} \\ \vdots & \ddots & \vdots \\ \sigma_{N_x-1,0} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix}_{N_x-1,0} & \cdots & \sigma_{N_x-1,N_y-1} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix}_{N_x-1,N_y-1} \end{bmatrix}. \tag{C.3}$$

Appendix D. Curve plots

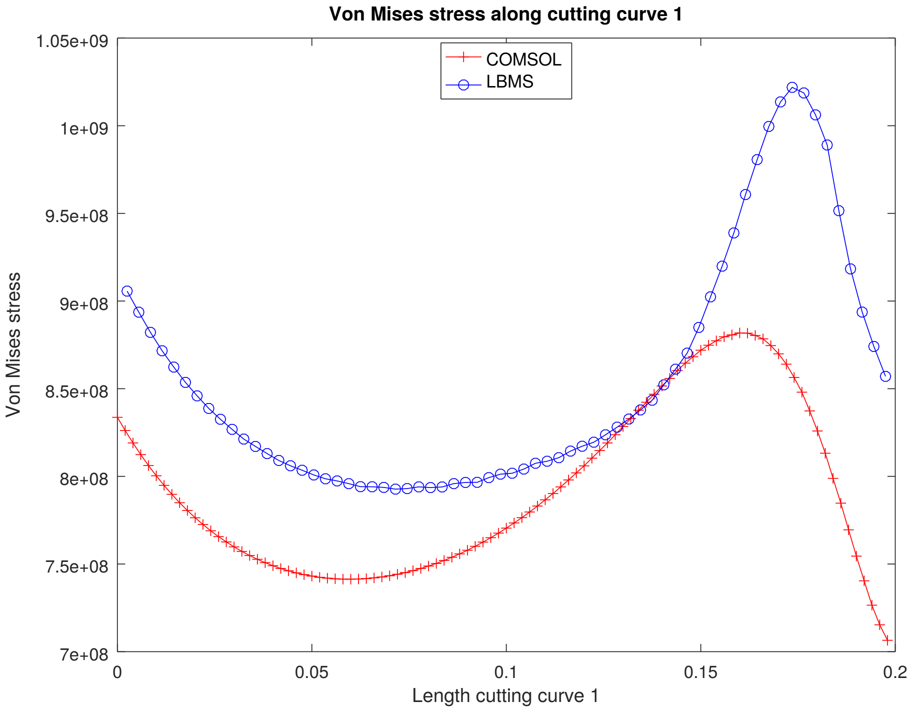

Figure D.8: Von-Mises stress along cutting curve C_1 for case E=200GPa, ν=0.15.

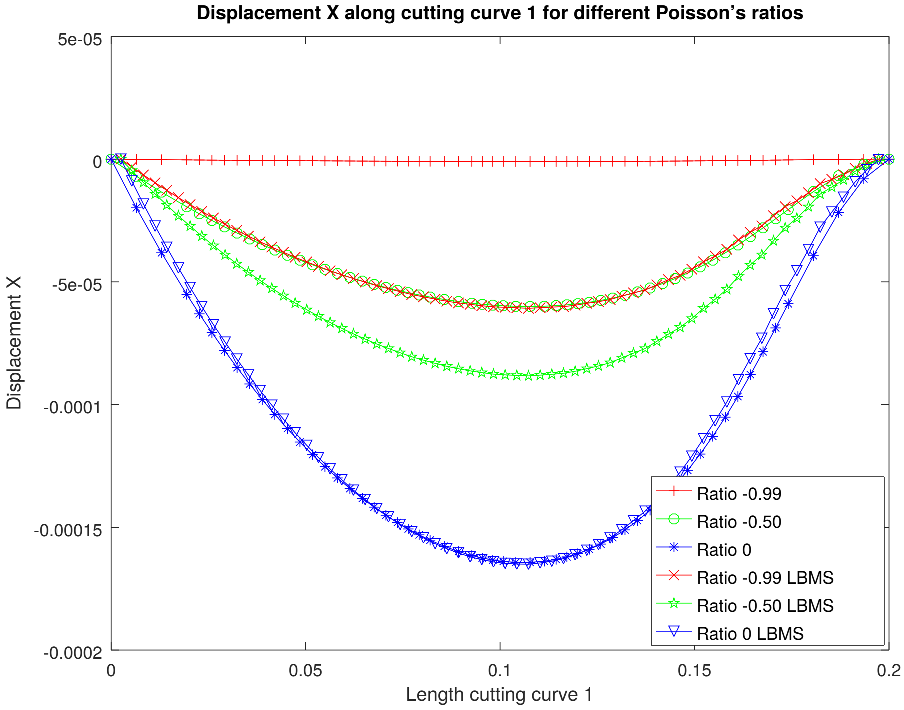

Figure D.9: First displacement components along cutting curve C_1 for negative Poisson's ratios.

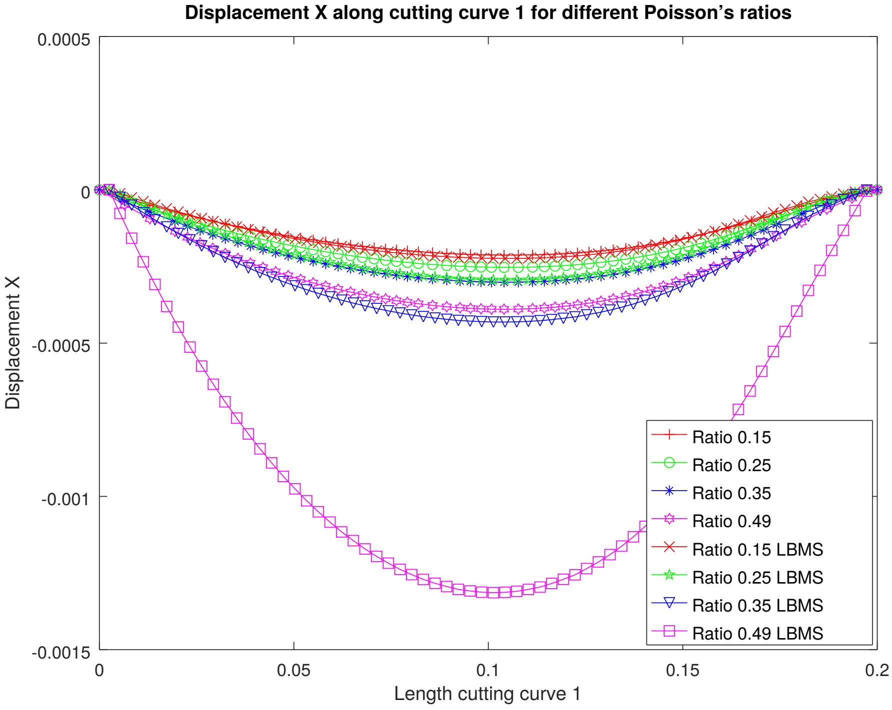

Figure D.10: First displacement components along cutting curve C_1 for positive Poisson's ratios.

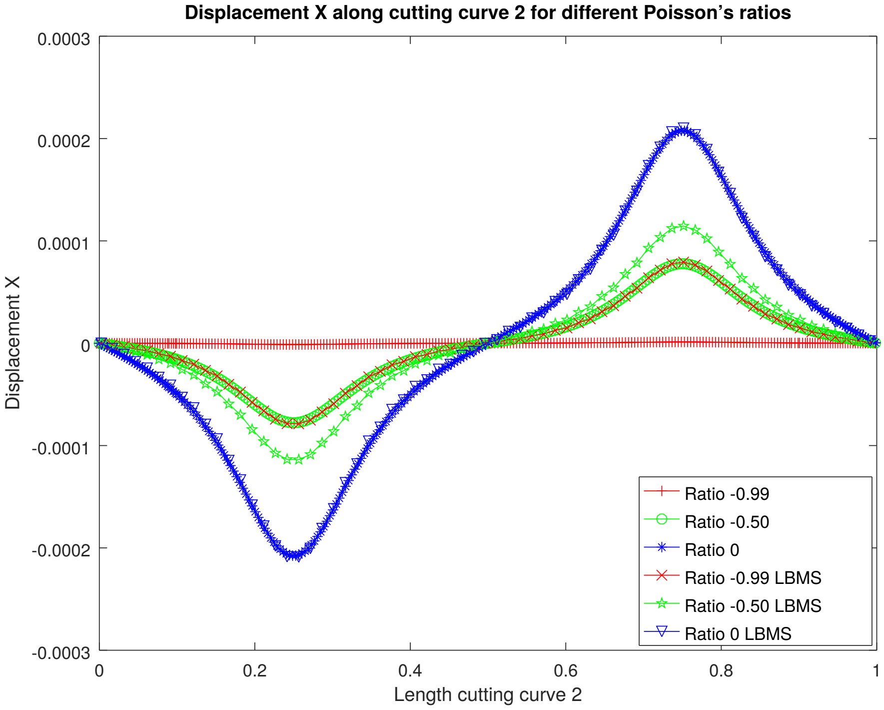

Figure D.11: First displacement components along cutting curve C_2 for negative Poisson's ratios.

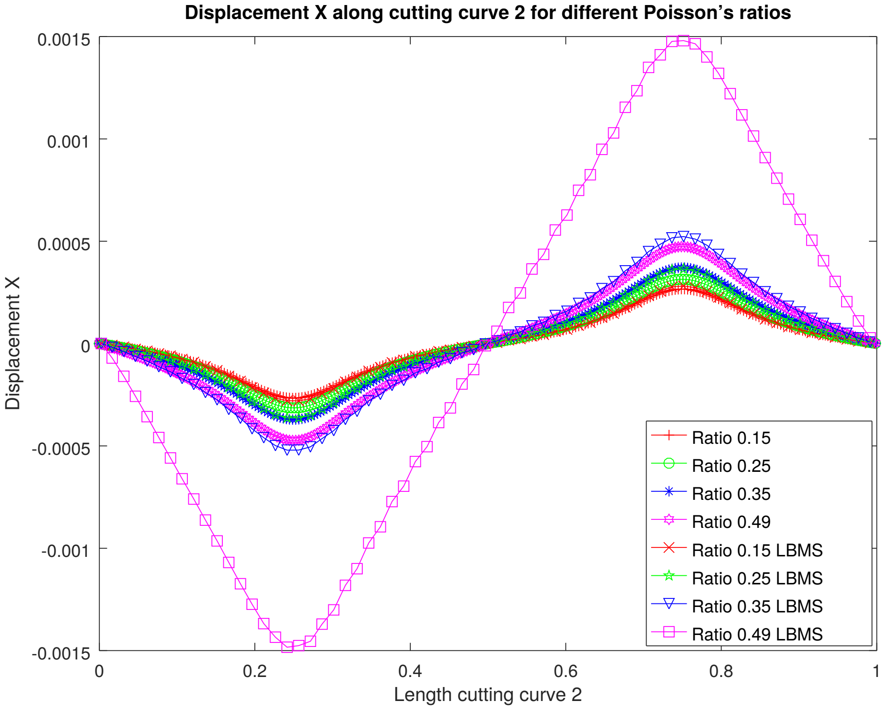

Figure D.12: First displacement components along cutting curve C_2 for positive Poisson's ratios.

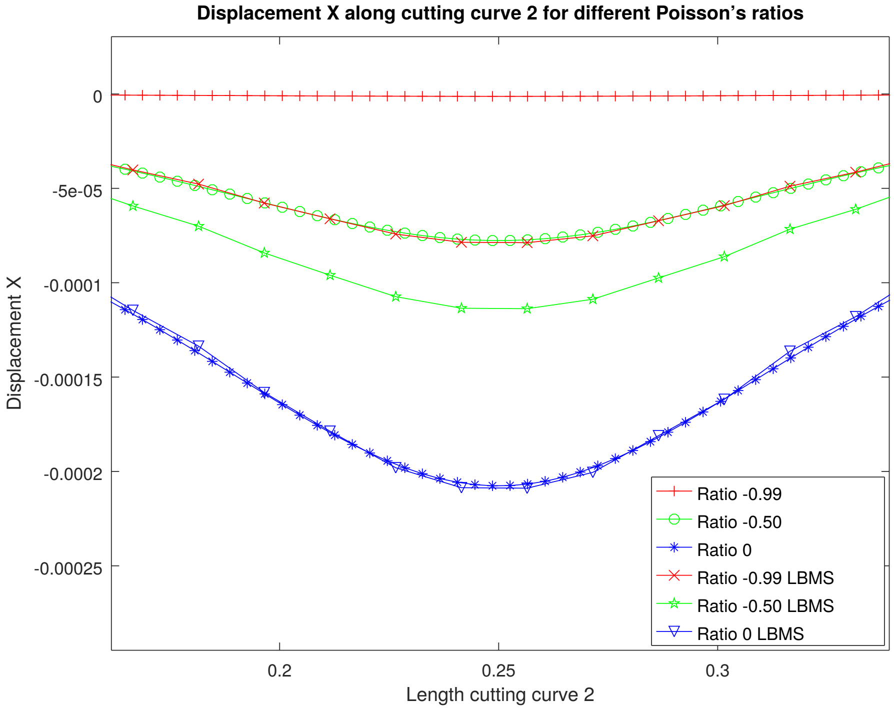

Figure D.13: Detailed view of first displacement components along cutting curve C_2 for negative Poisson's ratios.

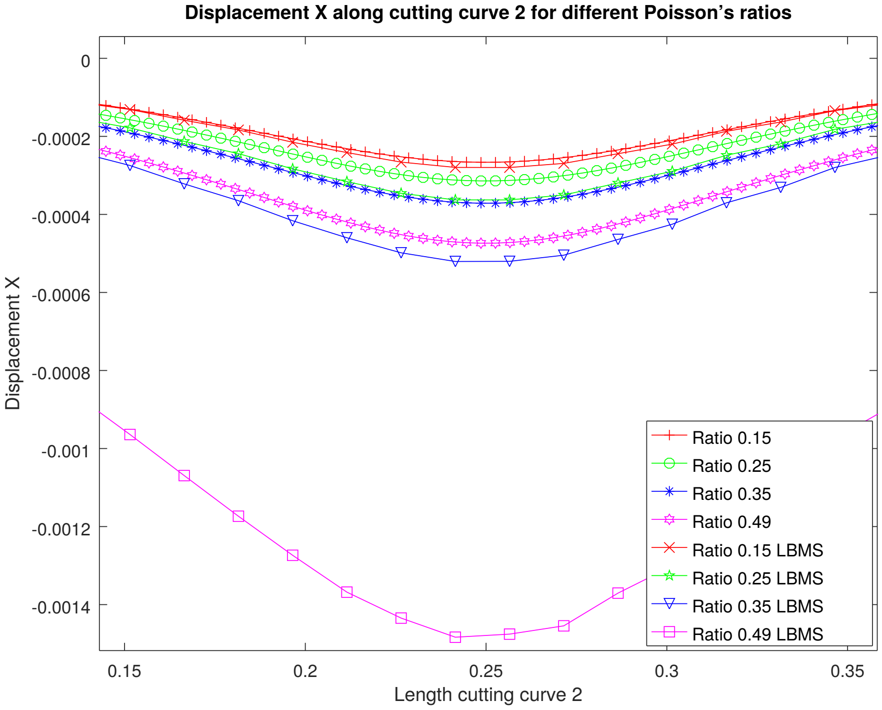

Figure D.14: Detailed view of first displacement components along cutting curve C_2 for positive Poisson's ratios.

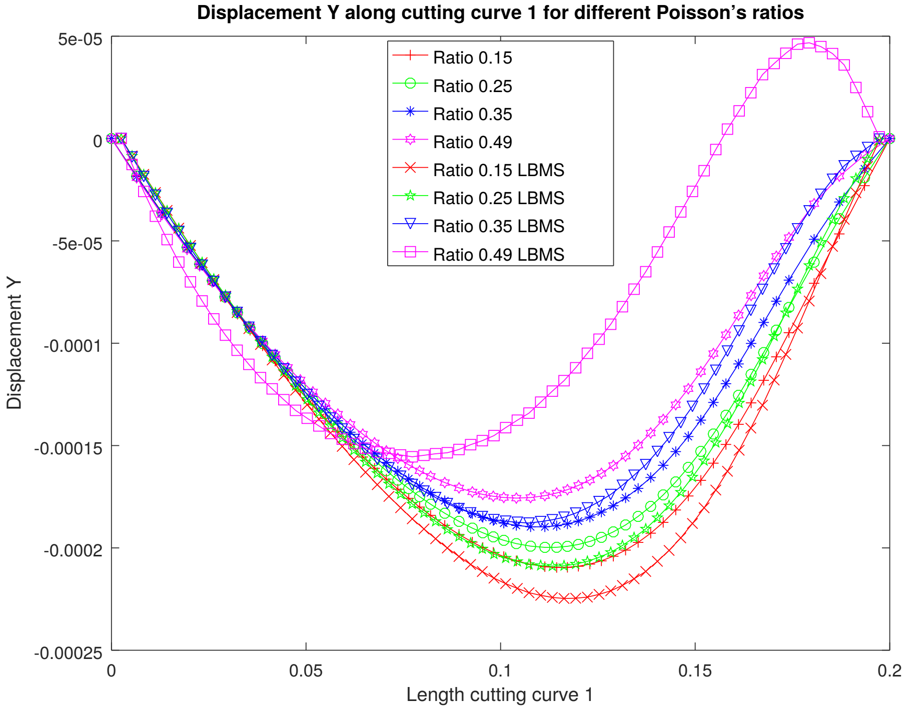

Figure D.15: Second displacement component along cutting curve C_1 for positive Poisson's ratios.

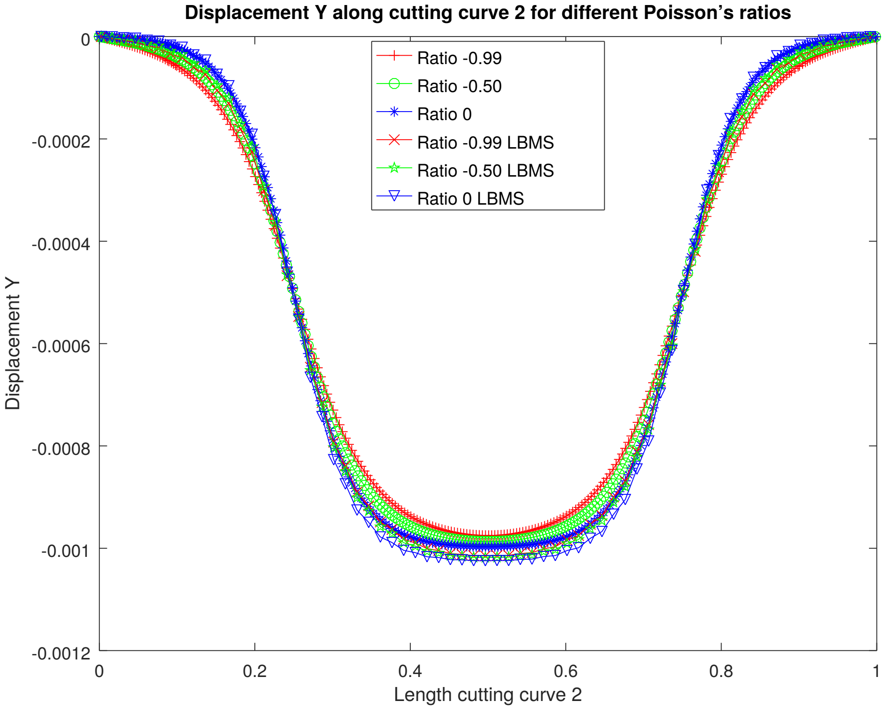

Figure D.16: Second displacement components along cutting curve C_2 for negative Poisson's ratios.

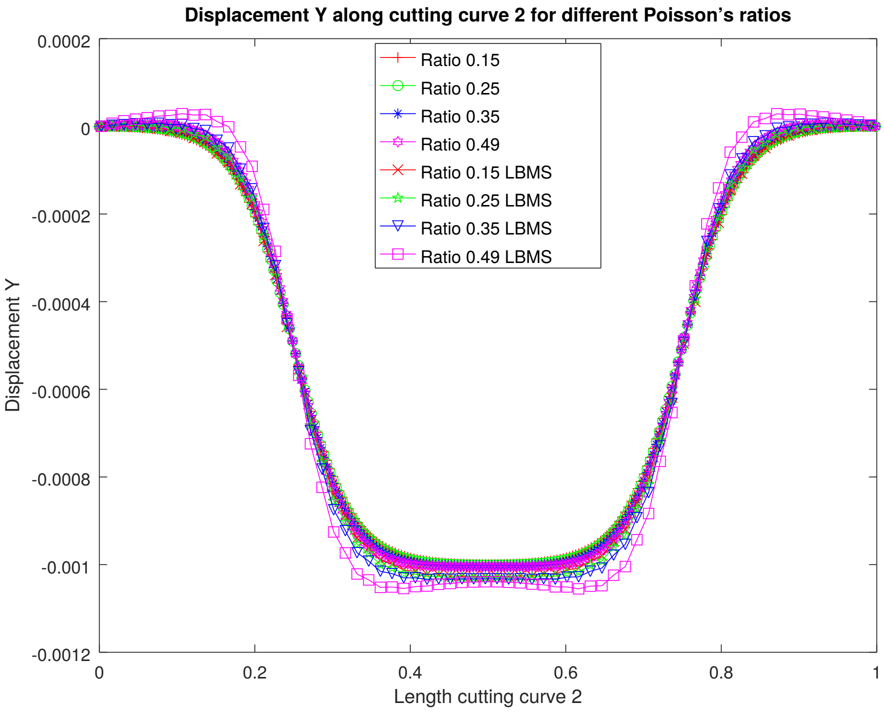

Figure D.17: Second displacement components along cutting curve C_2 for positive POISSON's ratios.

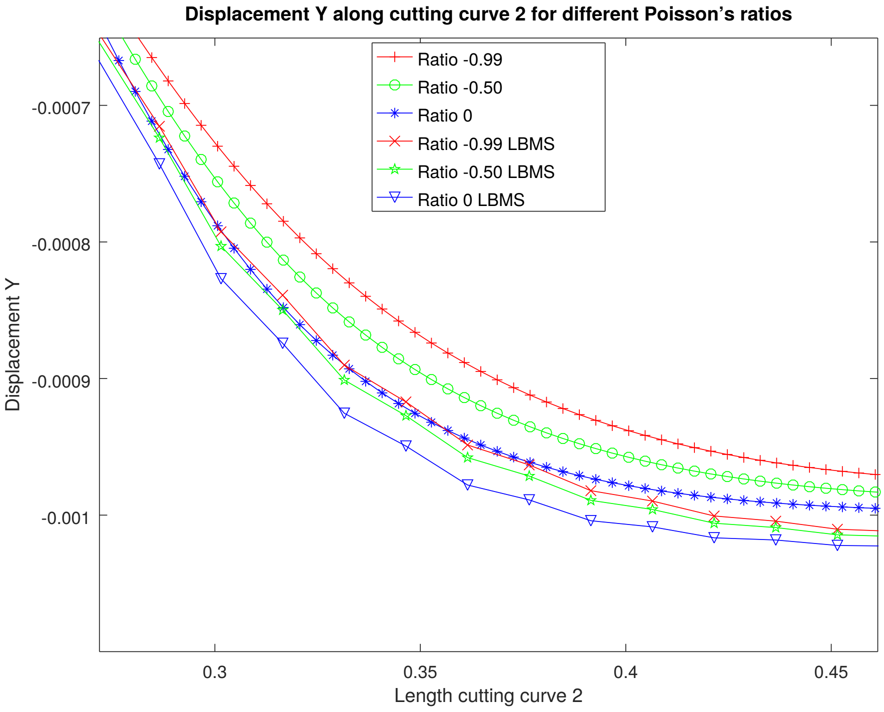

Figure D.18: Detailed view of second displacement components along cutting curve C_2 for negative POISSON's ratios.

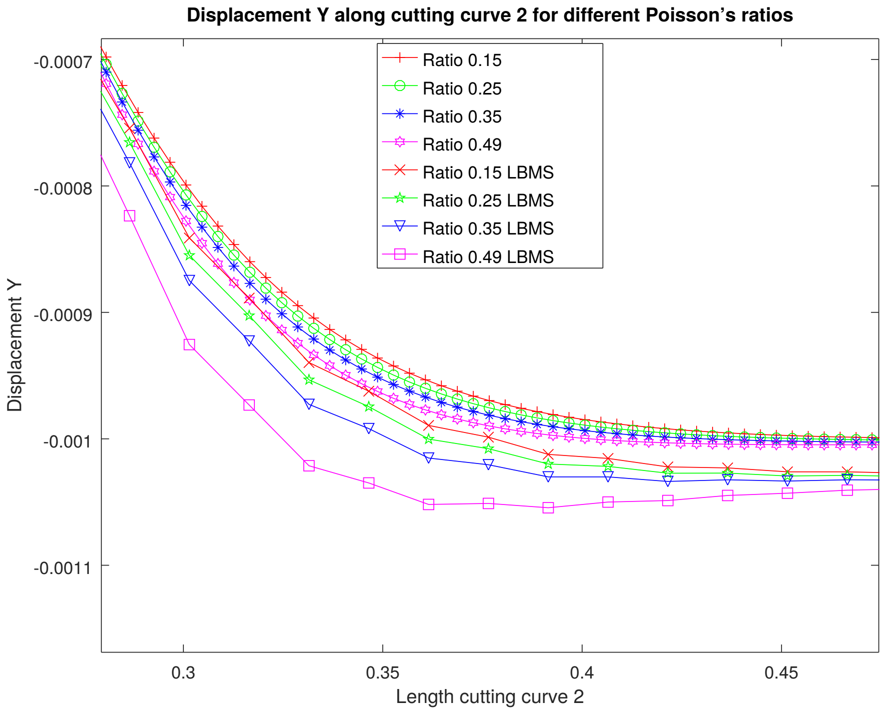

Figure D.19: Detailed view of second displacement components along cutting curve C_2 for positive POISSON's ratios.

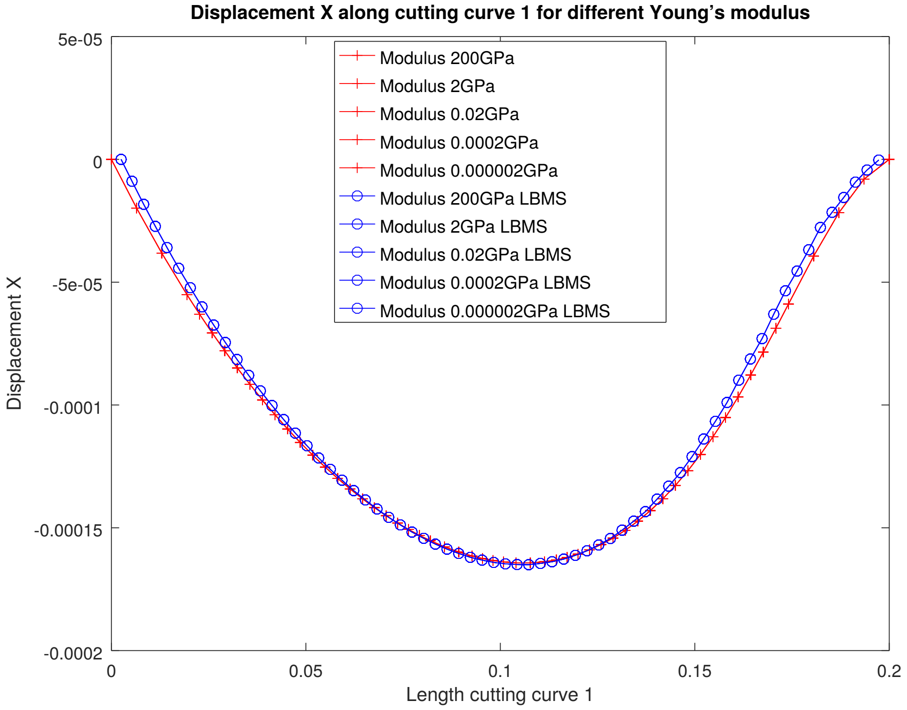

Figure D.20: First displacement components along cutting curve C_1 for different YOUNG's modulus.

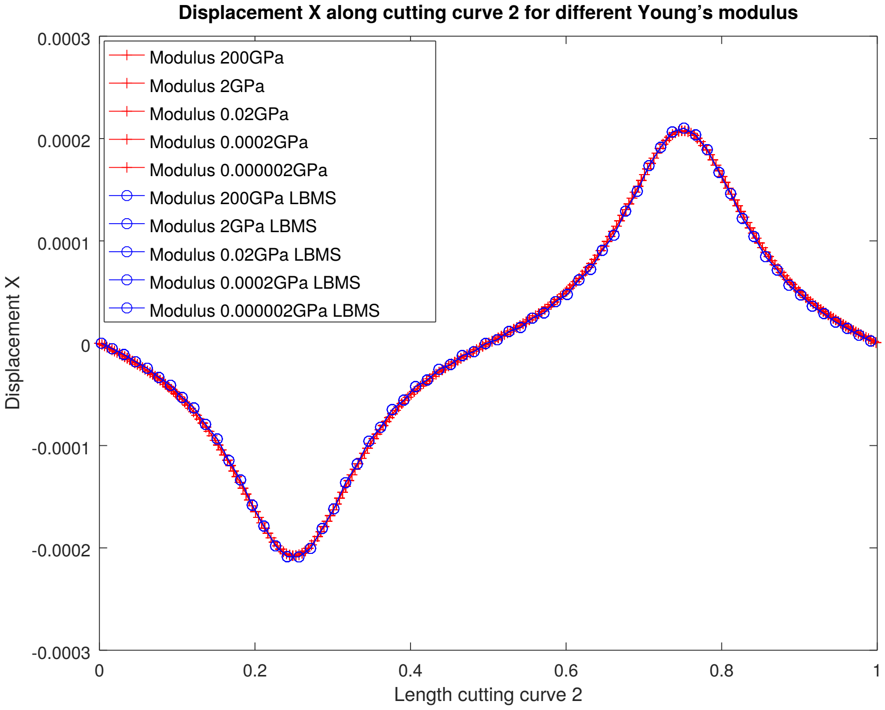

Figure D.21: First displacement components along cutting curve C_2 for different YOUNG's modulus.

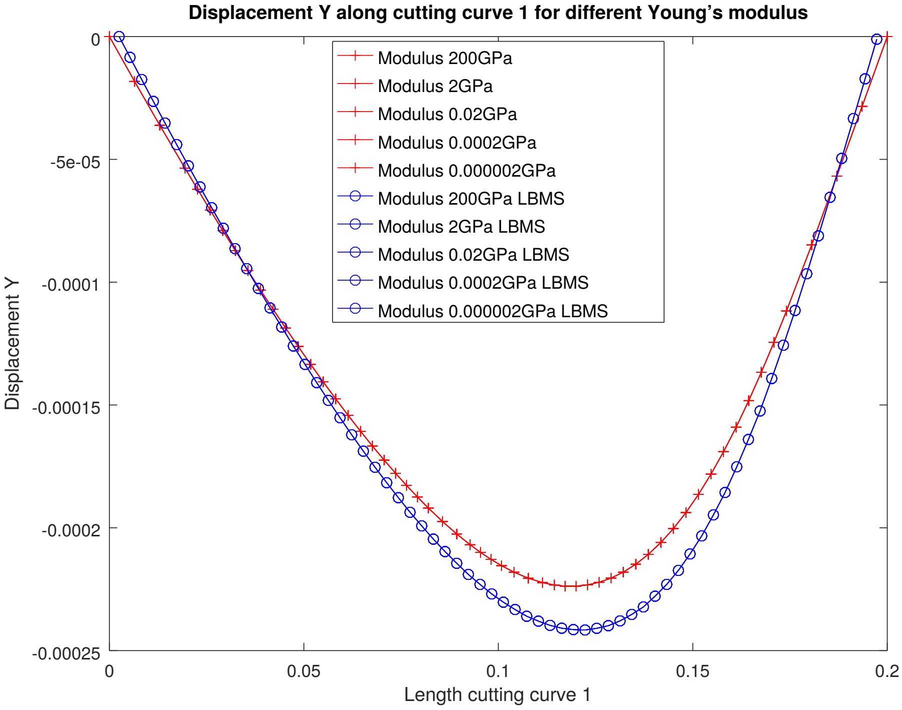

Figure D.22: Second displacement components along cutting curve C_1 for different YOUNG's modulus.

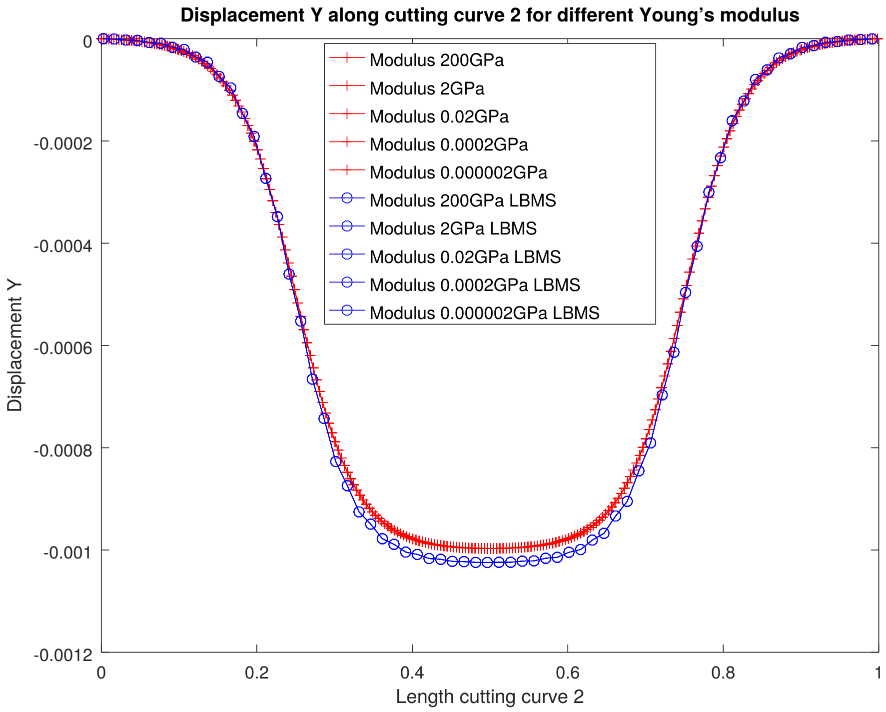

Figure D.23: Second displacement components along cutting curve C_2 for different YOUNG's modulus.

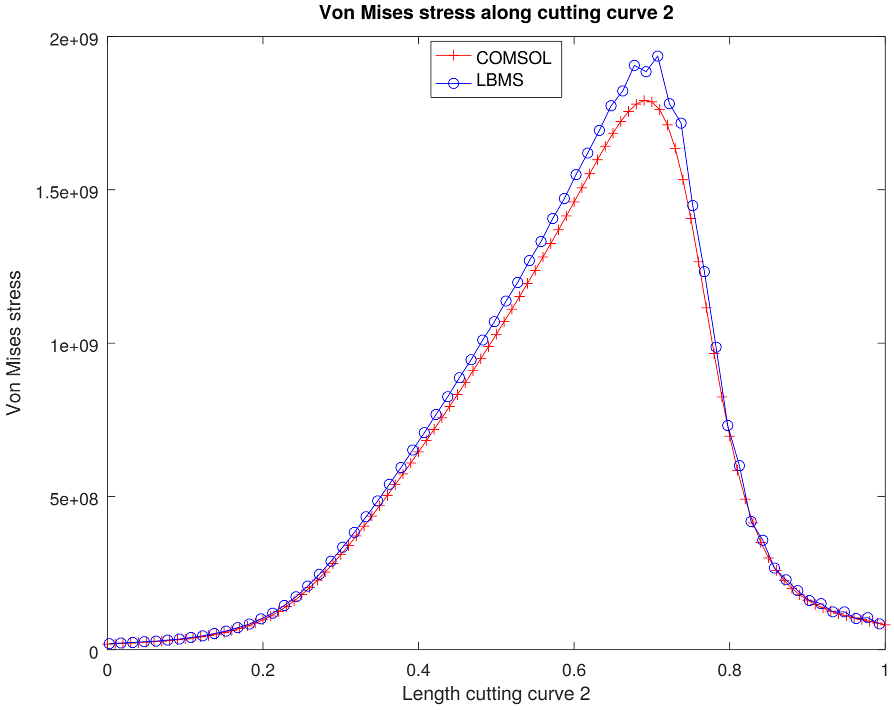

Figure D.24: VON-MISES stress along cutting curve C_2 for case E=200GPa, ν=0.15 with triangular loading.

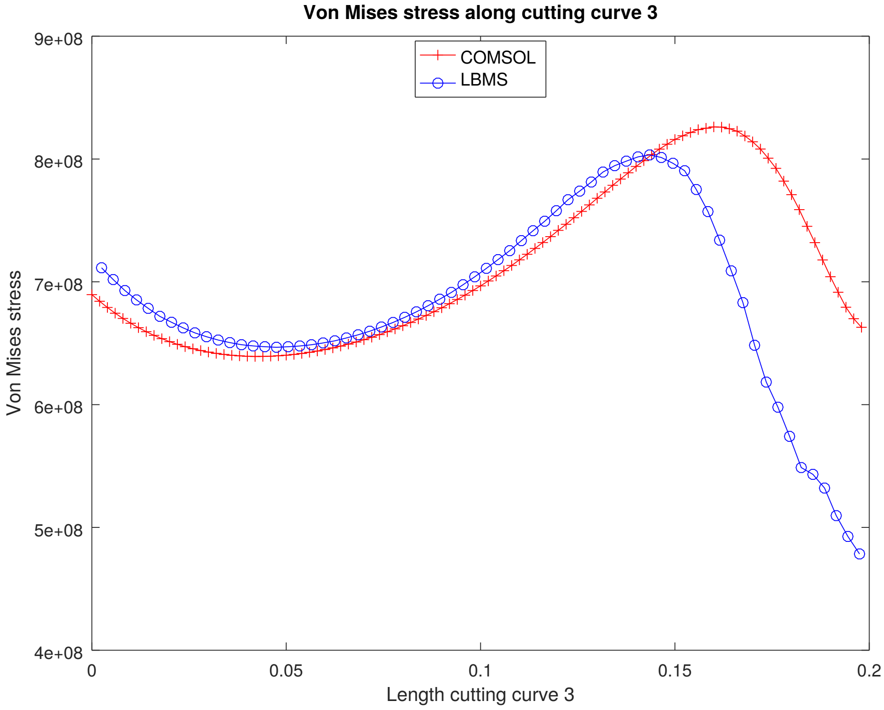

Figure D.25: Von-Mises stress along cutting curve C_3 for case E=200GPa, ν=0.15 with triangular loading.

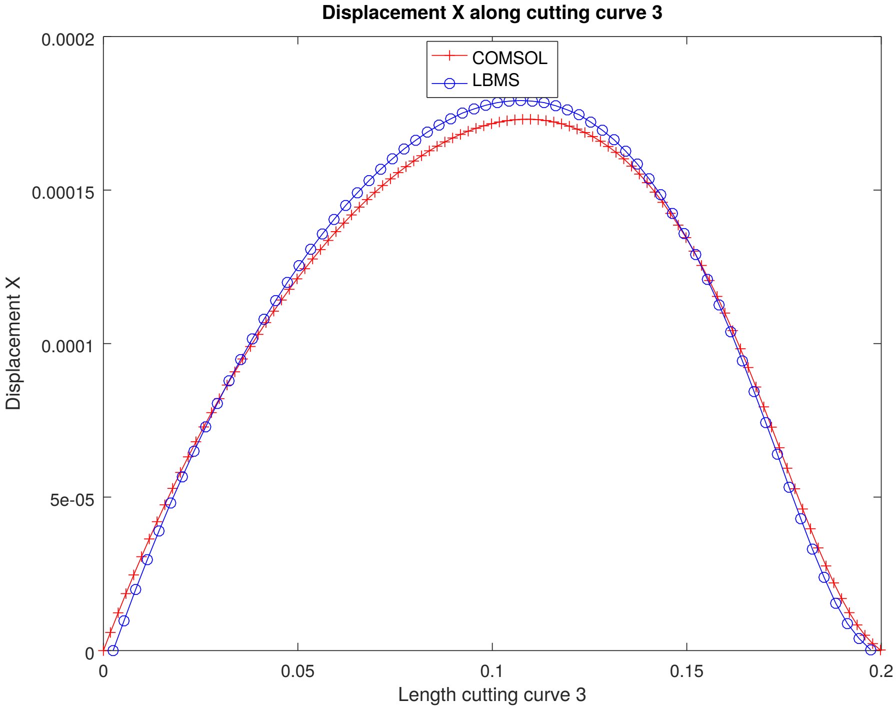

Figure D.26: First displacement components along cutting curve C_3 for case E=200GPa, ν=0.15 with triangular loading.

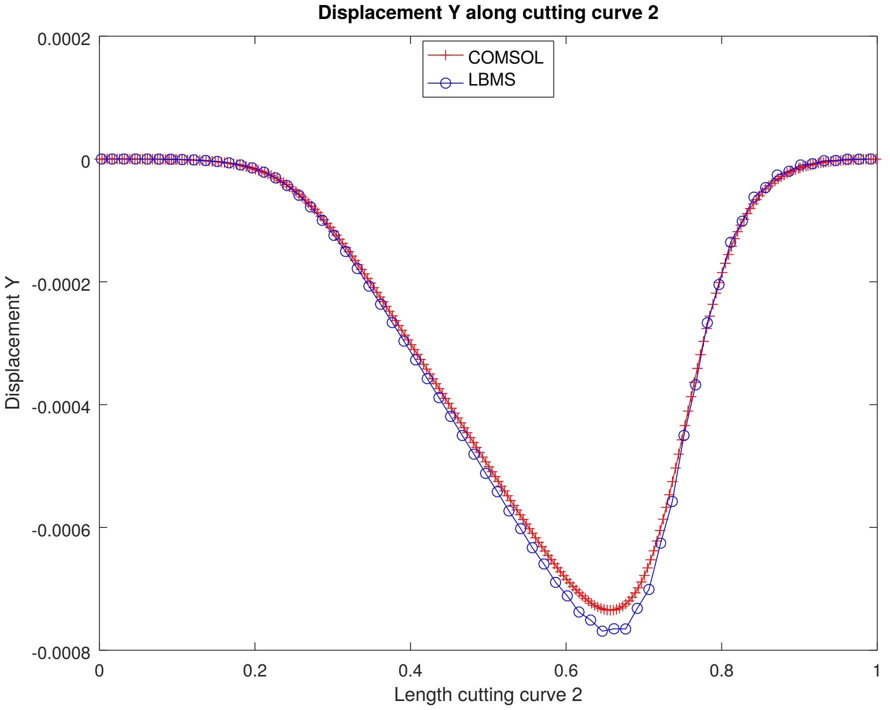

Figure D.27: Second displacement components along cutting curve C_2 for case E=200GPa, ν=0.15 with triangular loading.

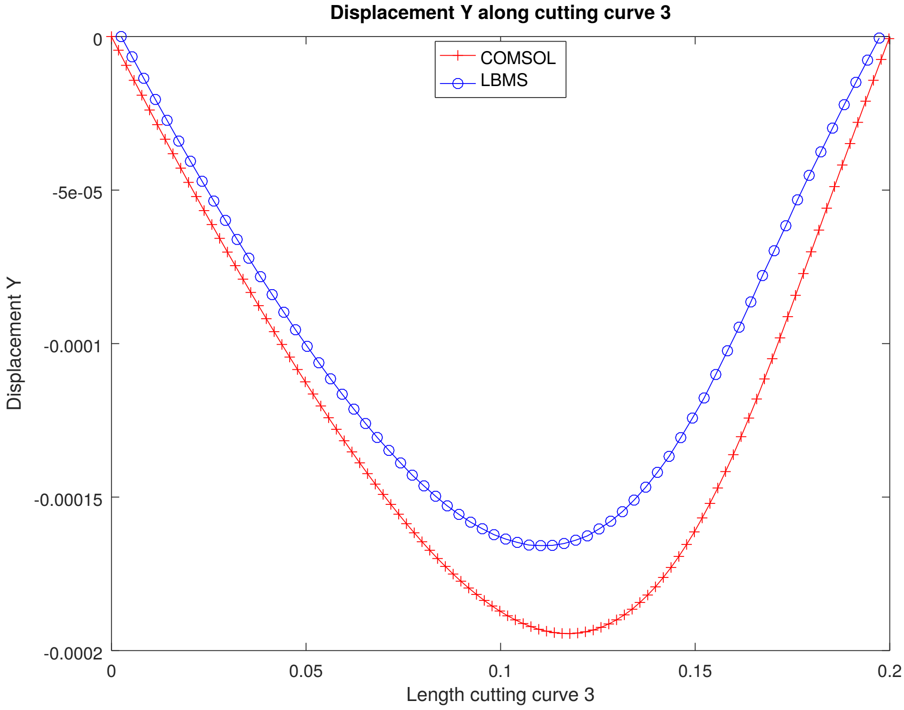

Figure D.28: Second displacement components along cutting curve C_3 for case E=200GPa, ν=0.15 with triangular loading.

References

[1] Frisch U, Hasslacher B, Pomeau Y. Lattice-gas automata for the Navier-Stokes equation. *Physical review letters* 1986; 56(14): 1505.

[2] Bhatnagar PL, Gross EP, Krook M. A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems. *Physical Review* 1954; 94(3): 511–525. doi: 10.1103/PhysRev.94.511

[3] McNamara GR, Zanetti G. Use of the Boltzmann Equation to Simulate Lattice-Gas Automata. *Physical Review Letters* 1988; 61(20): 2332–2335. doi: 10.1103/PhysRevLett.61.2332

[4] Succi S, Benzi R, Higuera F. The Lattice Boltzmann Equation: A New Tool for Computational Fluid-Dynamics. *Physica D: Nonlinear Phenomena* 1991; 47(1-2): 219–230.

[5] Karlin IV, Gorban AN, Succi S, Boffi V. Maximum Entropy Principle for Lattice Kinetic Equations. *Physical Review Letters* 1998; 81(1): 6–9. doi: 10.1103/PhysRevLett.81.6

[6] Karlin IV, Bösch F, Chikatamarla SS. Gibbs' Principle for the Lattice-Kinetic Theory of Fluid Dynamics. *Physical Review E* 2014; 90(3): 031302. doi: 10.1103/PhysRevE.90.031302

[7] Gunstensen AK, Rothman DH, Zaleski S, Zanetti G. Lattice Boltzmann Model of Immiscible Fluids. *Physical Review A* 1991; 43(8): 4320–4327. doi: 10.1103/PhysRevA.43.4320

[8] Shan X, Chen H. Lattice Boltzmann Model for Simulating Flows with Multiple Phases and Components. *Physical Review E* 1993; 47(3): 1815–1819. doi: 10.1103/PhysRevE.47.1815

[9] Swift MR, Osborn WR, Yeomans JM. Lattice Boltzmann Simulation of Nonideal Fluids. *Physical review letters* 1995; 75(5): 830.

[10] He X, Chen S, Zhang R. A lattice Boltzmann scheme for incompressible multiphase flow and its application in simulation of Rayleigh–Taylor instability. *Journal of computational physics* 1999; 152(2): 642–663.

[11] Lee T, Lin CL. A stable discretization of the lattice Boltzmann equation for simulation of incompressible two-phase flows at high density ratio. *Journal of Computational Physics* 2005; 206(1): 16–47.

[12] Huang H, Sukop M, Lu X. *Multiphase lattice Boltzmann methods: Theory and application.* John Wiley & Sons . 2015.

[13] Chiappini D, Xue X, Falcucci G, Sbragaglia M. Ligament Break-up Simulation through Pseudo-Potential Lattice Boltzmann Method. In: ICNAAM. ; 2018; Thessaloniki, Greece: 420003

[14] Marconi S, Chopard B. A Lattice Boltzmann Model for a Solid Body. *International Journal of Modern Physics B* 2003; 17(01n02): 153–156. doi: 10.1142/S0217979203017254

[15] Buxton GA, Verberg R, Jasnow D, Balazs AC. Newtonian Fluid Meets an Elastic Solid: Coupling Lattice Boltzmann and Lattice-Spring Models. *Physical Review E* 2005; 71(5). doi: 10.1103/PhysRevE.71.056707

[16] Wu TH, Qi D. Lattice-Boltzmann Lattice-Spring Simulations of Influence of Deformable Blockages on Blood Fluids in an Elastic Vessel. *Computers & Fluids* 2017; 155: 103–111. doi: 10.1016/j.compfluid.2017.03.029

[17] Guangwu Y. A Lattice Boltzmann Equation for Waves. *Journal of Computational Physics* 2000; 161(1): 61–69. doi: 10.1006/jcph.2000.6486

[18] Frantziskonis GN. Lattice Boltzmann Method for Multimode Wave Propagation in Viscoelastic Media and in Elastic Solids. *Physical Review E* 2011; 83(6). doi: 10.1103/PhysRevE.83.066703

[19] Murthy JSN, Kolluru PK, Kumaran V, Ansumali S. Lattice Boltzmann Method for Wave Propagation in Elastic Solids. *Communications in Computational Physics* 2018: 18. doi: 10.4208/cicp.OA-2016-0259

[20] Kamrin K, Nave JC. An Eulerian approach to the simulation of deformable solids: Application to finite-strain elasticity. *arXiv preprint arXiv:0901.3799* 2009.

[21] Noël R, Ge F, Zhang Y, Navarro L, Courbebaisse G. Lattice Boltzmann Method for Modelling of Biological Phenomena. In: EUSIPCO. IEEE; 2017; Kos: 2654–2658

[22] Noël R, Navarro L, Courbebaisse G. Lattice Boltzmann Method & Mathematical Morphology. In: GRETSI. GRETSI; 2019; Lille, France.

[23] PALABOS v2.0r0 - Parallel Lattice Boltzmann Solver. `https://palabos.unige.ch/`; 2017.

[24] COMSOL Multiphysics 5.5. `www.comsol.com`; 2019.

[25] Boltzmann L. *Further Studies on the Thermal Equilibrium of Gas Molecules.* 1. : 262–349; Published by Imperial College Press and distributed by World Scientific Publishing Co. in 2003 . 1872

[26] Chapman S, Cowling TG. *The Mathematical Theory of Non-Uniform Gases: An Account of the Kinetic Theory of Viscosity, Thermal Conduction, and Diffusion in Gases.* Cambridge Mathematical LibraryCambridge ; New York: Cambridge University Press. 3rd ed ed. 1970.

[27] Jiang N, Xu CJ, Zhao HJ. Incompressible Navier-Stokes-Fourier Limit from the Boltzmann Equation: Classical Solutions. *Indiana University Mathematics Journal* 2018; 67(5): 1817–1855.

[28] Guo Z, Zheng C, Shi B. Discrete Lattice Effects on the Forcing Term in the Lattice Boltzmann Method. *Physical Review E* 2002; 65(4). doi: 10.1103/PhysRevE.65.046308

[29] Vlasov AA. The Vibration Properties of Electron Gas. *Soviet Physics Uspekhi* 1938; 10(6): 721–733. doi: 10.1070/PU1968v010n06ABEH003709

[30] Maquart T, Noël R, Navarro L. Lattice Boltzmann Method for Solids (LBMS) - Library source code. `https://github.com/tmaquart/LBMS.git`; 2020.

[31] Liu XD, Osher S, Chan T, others . Weighted essentially non-oscillatory schemes. *Journal of computational physics* 1994; 115(1): 200–212.

[32] Simo JC, Hughes TJ. *Computational inelasticity.* 7. Springer Science & Business Media . 2006.

[33] Bower AF. *Applied mechanics of solids.* CRC press . 2009.

[34] Lemaitre J, Chaboche JL. *Mechanics of solid materials.* Cambridge university press . 1994.

[35] Elguedj T, Bazilevs Y, Calo VM, Hughes TJ. B and F projection methods for nearly incompressible linear and non-linear elasticity and plasticity using higher-order NURBS elements. *Computer methods in applied mechanics and engineering* 2008; 197(33-40): 2732–2762.

[36] Chopard B, Luthi PO. Lattice Boltzmann Computations and Applications to Physics. *Theoretical Computer Science* 1999; 217(1): 115–130. doi: 10.1016/S0304-3975(98)00153-4

[37] Liu F, Liu G, Shu C. Fluid-Structure Interaction Simulation Based on Immersed Boundary-Lattice Boltzmann Flux Solver and Absolute Nodal Coordinate Formula. *Physics of Fluids* 2020; 32(4): 047109. doi: 10.1063/1.5144752

[38] Cushman-Roisin B, Epps BP. From Boltzmann Kinetics to the Navier-Stokes Equations without a Chapman-Enskog Expansion. 2018.

[39] Succi S. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond.* Numerical Mathematics and Scientific ComputationOxford : New York: Clarendon Press ; Oxford University Press . 2001.

[40] van der Sman RGM, Ernst MH. Convection-Diffusion Lattice Boltzmann Scheme for Irregular Lattices. *Journal of Computational Physics* 2000; 160(2): 766–782. doi: 10.1006/jcph.2000.6491

[41] He X, Shan X, Doolen GD. Discrete Boltzmann Equation Model for Nonideal Gases. *Physical Review E* 1998; 57(1): R13-R16. doi: 10.1103/PhysRevE.57.R13

Abbreviations

BE Boltzmann Equation 1, 3, 4, 18, 20

BGK Bhatnagar, Gross and Krook 2, 3, 8, 9, 17, 19, 21

CFD Computational Fluid Dynamics 1, 8, 11

FEA Finite Element Analysis 3, 10, 12, 13, 15–17

FSI Fluid Structure Interaction 2, 16–18

GPU Graphics Processing Unit 17

LBE Lattice Boltzmann Equation 20, 21

LBGKE Lattice Boltzmann-BGK Equation 1, 4, 21

LBM Lattice Boltzmann Method 1, 2, 4–6, 9, 12, 16–18, 20

LBMS Lattice Boltzmann Method for Solids 6, 10–18

LGCA Lattice Gas Cellular Automata 1

MRT Multiple Relaxation Time 1

NRMSE Normalized Root Mean Square Error 16

NSE Navier-Stokes Equations 2

NSF Navier-Stokes-Fourier 3, 18, 20

PaLaBoS Parallel Lattice Boltzmann Solver 3, 6, 9, 10

PDE Partial Differential Equations 3, 4

RMSE Root Mean Square Error 16

SI International System of Units 10, 11

WENO Weighted Essentially Non-Oscillatory 8, 12, 17

Nomenclature

Q any function with discrete values over the whole lattice 8

C fourth-order stiffness tensor 7, 11
c_s celerity of the sound 4
$\Omega(.,.)$ collision operator 3, 4, 9, 17, 20, 21
κ thermal conductivity 3, 20

ρ mass density 3–5, 9, 12, 19, 20
D physical space dimension 3, 9, 12, 19, 20
δ_x Dirac distribution centred on x 5
$\boldsymbol{u}$ macroscopic displacement field 11, 21
$\mathbf{U}$ displacement tensor field over the whole lattice 6, 7, 21
$\boldsymbol{c}$ microscopic velocity in the $\boldsymbol{v}$ frame 3, 19, 20

E Young's modulus 11–16, 22, 30–32
E_P electrical field 5
E_k kinematic energy 3, 20
E_θ thermal energy 3, 20

f density distribution function over velocity space 3–5, 18–20
$f^{(0)}$ equilibrium distribution given by the Maxwell-Boltzmann distribution 3, 9, 10
$f_i^{(0)}$ discretized equilibrium distribution given by the Maxwell-Boltzmann distribution 4, 9, 17, 21
f_i discretized density distribution 4, 5, 9, 10, 17, 20, 21
$\widehat{f}_i$ discretized velocity density distribution function for the numerical scheme 4, 21
$\boldsymbol{g}$ mass force field 3–5, 19
g_i forcing term projected over the velocity space 4, 20, 21
ω relaxation frequency 3, 4, 9, 10, 17–21
ω_c boundary relaxation parameter 9, 10

G_x global position x of a cell in the lattice 6–11, 17
G_y global position y of a cell in the lattice 6–11, 17

I identity square matrix $D \times D$ 11, 19, 20
$\Delta\boldsymbol{x}$ increment of space 7, 8
Δt increment of time 4, 7, 10, 21

λ Lamé's first parameter 11, 14

m mass of a particle 5
μ shear modulus 11

N_x number of lattice nodes in the first direction 8, 21, 22

N_y number of lattice nodes in the second direction 8, 21, 22

ν POISSON's ratio 11–16, 22, 30–32

$\mathcal{P}_n$ math P 4, 20

ϵ perturbation order 18, 19

$\boldsymbol{x}$ coordinate Eulerian vector field 3–8, 19–21

p pressure scalar 3, 4, 19, 20

Π viscous stress tensor 3, 4

q_P electrical charge of a particle 5

q number of discretized speed used in the lattice 4, 9, 12

R ideal gas constant 3, 4, 20

s_i source forcing term projected over the velocity space 9, 17

$\boldsymbol{v}$ macroscopic speed field 3–5, 7, 9, 19, 20

ε linearised strain tensor 6, 22

$\mathbf{E}$ strain tensor field over the whole lattice 6, 7, 11, 22

σ CAUCHY stress tensor 4–6, 8, 22

$\mathbf{\Sigma}$ stress tensor field over the whole lattice 6–8, 11, 12, 22

θ thermodynamical absolute temperature 3, 4, 19, 20

t time 3–5, 8–10, 17, 19–21

ν kinematic viscosity 3, 4, 20

w_i discrete weight of the GAUSS quadrature 4, 9

$\mathbf{X}$ first direction of basis 7, 8, 11, 16

x first coordinate of a cell in a block lattice 7

$\boldsymbol{\xi}$ microscopic velocity of particles 3, 5, 19, 20

$\boldsymbol{\xi}_i$ discretized particles velocity 4, 20, 21

$\mathbf{Y}$ second direction of basis 7, 8, 11, 16

y second coordinate of a cell in a block lattice 7