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

Tristan Maquart, Romain Noël, Guy Courbebaisse, Laurent Navarro

applied sciences

MDPI

Article

Toward a Lattice Boltzmann Method for Solids—Application to Static Equilibrium of Isotropic Materials

Tristan Maquart [1,*], Romain Noël [2,*], Guy Courbebaisse [3] and Laurent Navarro [1]

[1] Mines Saint-Étienne, Univ. Lyon, Univ. Jean Monnet, INSERM U1059, SAINBIOSE, Centre Ingénierie Santé (CIS), 158 Cours Fauriel, F-42023 Saint-Étienne, France; navarro@emse.fr
[2] Chair of Building Physics, Department of Mechanical and Process Engineering, ETH Zürich (Swiss Federal Institute of Technology in Zürich), 8093 Zürich, Switzerland
[3] Univ. Lyon, INSA-Lyon, Université Claude Bernard Lyon 1, UJM Saint-Étienne, CREATIS, CNRS, Inserm, UMR 5220, U1294, 21 Avenue Jean Capelle, F-69621 Lyon, France; guy.courbebaisse@creatis.insa-lyon.fr
* Correspondence: tristan.maquart@hotmail.fr (T.M.); ronoel@ethz.ch or romainoel@free.fr (R.N.)

Abstract: This work 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. Remembering 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. This is indeed the case when combined with a suitable mean-field external force term to set 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 the 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 newly designed Lattice Boltzmann Method for Solids (LBMS) is exhibited through a 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. The obtained results reveal 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

Citation: Maquart, T.; Noël, R.; Courbebaisse, G.; Navarro, L. Toward a Lattice Boltzmann Method for Solids—Application to Static Equilibrium of Isotropic Materials. *Appl. Sci.* **2022**, *12*, 4627. https://doi.org/10.3390/app12094627

Academic Editor: José A.F.O. Correia

Received: 2 March 2022
Accepted: 27 April 2022
Published: 4 May 2022

1. Introduction

1.1. Background

The Lattice Boltzmann Method (LBM) has proven to be an efficient and reliable method for Computational Fluid Dynamics (CFD) for several decades now. Historically, it follows the seminal work of Frisch [1] on the use of Lattice Gas Cellular Automata (LGCA) for simulating the Navier-Stokes equations. This work was a remarkable breakthrough, as it proposed simulating fluids at the particle 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 the Boolean formulation of particle interactions. Decades before, Bhatnagar, Gross and Krook achieved a linearization of the collision operator of the Boltzmann Equation (BE) [2] that allows for the reduction in the complexity of the equation by avoiding the implementation of an integral collision model. Macnamara [3] and Succi [4] proposed using discretized versions of the Boltzmann equation to overcome the LGCA' limitations, and gave birth to the so-called Lattice Boltzmann Method. Following this line of thought, Karlin et al. proposed a Lattice Boltzmann-BGK Equation (LBGKE) with an optimized local equilibrium and proved the H-theorem for it [5].

A large amount of work was since done in the LBM field, and recent years saw some remarkable improvements for the numerical simulation of high Reynolds numbers

fluids, with for example, the lattice-kinetic theory [6]. One can also mention the Multiple Relaxation Times (MRT) LBM which also allows better stability of the method for high Reynolds numbers, using a multiple-relaxation time term [7]. Such matrix collision operator was first introduced by [8]. All these new insights are relevant for simulating high Reynolds number fluids, but also to 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-component 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 [9], has been developed for two-phase 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 with the Navier-Stokes Equations (NSE) in non-interface regions. The second category is based on the introduction of an inter-particle potential [10]. 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-phase flow [11]. 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 [12,13] for the simulation of incompressible two-phase flows. A review of multi-phase LBM is proposed in [14]. It gives a comprehensive overview of the methods and the associated algorithmic aspects.

The idea of combining specifically adapted collision operators and the 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. [15] applied the multi-phase method to the simulation of ligament break-up and gives insight into its potential in bio-mechanical soft tissue simulations.

1.2. The Need for a LBM for Solids

The efforts to construct top-down collision operators able to simulate many physical complex behaviors started when the LBM raised from the LGCA, with works from Succi et al. [16,17]. Some pioneering attempts, by Chopard et al. for example [18], in solid simulations using the LBM have been performed, with good application results on fractures and fragmentation for a solid body. These simulations used a distribution of forces rather than a distribution of particles as is usually the case with the LBM. Such distributions of forces bring two disadvantages. The mass conservation can not be recovered anymore. Secondly, since distributions are no more scalars but composed of vectors, it brings a large computational cost. Despite these limitations and the inability to fully recover the linear elastic equations, this work showed that the LBM is a good candidate when it comes to simulating the decohesion of the material.

Alternative approaches for solid simulations exist with the LBM. They can be classified into two main categories: those dealing with velocity distribution functions [19–21] and those working with displacement distribution functions [22] in order to recover the Sophie-Germain equation [23]. In these previous works, the LBM is more often used as a numerical solver and the underlying link with the particles' dynamic is lost.

Other works combined the lattice-spring method and the LBM for Fluid Structure Interaction (FSI) [24]. For instance, Ref. [25] is dedicated to the hemodynamic simulation in deformable blood vessels. However, the combination of two different methods with different coordinate systems complicates the simulation. In addition, pure viscoelastic wave propagation simulations using the LBM have been proposed [26–28]. 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 required for fluid simulations, but mandatory for solid simulations. Kamrin et al. [29] proposed the so-called "reference map" to compute finite-difference simulations 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, interest in the LBM for the simulation based on medical imaging data [30] and biological phenomena [31] has been highlighted. It appears that the use of the LBM framework allows for the simulation of various problems based on what constitutes raw data in many biomedical applications, i.e., medical images.

This work proposes an LBM for the simulation of solids, retaining the link with particles. It is mainly based on the use of the Boltzmann–Vlasov equation (see Section 5.2), which is a special case of the Boltzmann equation. To be more specific, only its restriction to static equilibrium, through dedicated force terms, is tested here. The aim is to introduce the newly proposed method in detail and to present some numerical implementations for its first validation.

The work is organized as follows. Section 2 describes our model of LBM for solid aiming static equilibrium simulations. Section 3 presents numerical implementations and key algorithms of the method, and the integration in the Parallel Lattice Boltzmann Solver (PaLaBoS) [32] framework. Section 4 is dedicated to benchmarks and comparisons with Finite Element Analysis (FEA) using COMSOL software [33], in order to validate the method. Section 5 discusses the results and their openings, and Section 6 concludes the paper.

2. Lattice Boltzmann Method for Solids

2.1. Lattice Boltzmann Method

The BE [34] 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.

Using the trapezoidal rule and a change of variable (f_i to $\widehat{f}_i$ see Equation (A2) in Appendix A), a second-order discretization of the BE is done. Then, the time evolution can be simulated by a succession of the collision step and streaming step, described by the following famous equations which forms the core of the LBM and read:

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

$$f_i{}^s(\boldsymbol{x} + \boldsymbol{\xi}_i \Delta t, t + \Delta t) = f_i{}^c(\boldsymbol{x}, t). \tag{3}$$

where the superscript c and s stands for collision and streaming, f_i and respectively g_i are the discretization over the velocity space of f and $\boldsymbol{g} \cdot \nabla_{\boldsymbol{\xi}}(f)$ respectively, through a projection on the Hermite basis. Δt is the increment of time chosen, $\boldsymbol{\xi}_i$ is the result of the Gauss–Hermite quadrature, $f_i^{(eq)}$ is the discretized equilibrium distribution function and ω is the invert of the relaxation time for the system to converge toward the equilibrium. One

might also note that in Equation (2) using a negative sign in the factor with the term g_i is not the most common convention, so readers should be careful to avoid sign mistakes.

The change of variable used in the Equation (A3) also has an impact on the macroscopic variable. With the use of Equation (A2), one can compute:

$$\sum_{i=0}^{q} f_i = \rho, \tag{4}$$

$$\sum_{i=0}^{q} \boldsymbol{\xi}_i f_i = \widehat{\rho \boldsymbol{v}} = \rho \boldsymbol{v} - \frac{\Delta t}{2} \boldsymbol{g}_i, \tag{5}$$

$$\sum_{i=0}^{q} (\boldsymbol{\xi}_i - \boldsymbol{v}) \otimes (\boldsymbol{\xi}_i - \boldsymbol{v}) f_i = \widehat{\Pi} = p^{(0)} - \left(1 + \frac{\Delta t \omega}{2}\right) \Pi^{(1)} - \frac{\Delta t}{2} (\boldsymbol{g} \otimes \boldsymbol{v} + \boldsymbol{v} \otimes \boldsymbol{g}). \tag{6}$$

In the last expression, $\Pi^{(1)}$ is given by Equation (A27). These previous sums justify the correctional terms in the macroscopic speed and the "lattice viscosity" that has to be compensated when fitting the physical cinematic viscosity with the collision frequency $\omega = \frac{2{c_s}^2}{{c_s}^2 + 2\nu}$. In the previous formula, c_s is the celerity of sound and is obtained by ${c_s}^2 = \frac{1}{3} R\theta$, where θ is the temperature.

Using Guo's approach [35], the projection of the forcing term on the Hermite basis at second order reads:

$$g_i = \frac{w_i \rho}{{c_s}^2} \boldsymbol{\xi}_i \cdot \boldsymbol{g} + \frac{w_i \rho}{2{c_s}^4} \left[(\boldsymbol{v} \cdot \boldsymbol{\xi}_i) \boldsymbol{\xi}_i - \frac{\boldsymbol{v}}{{c_s}^{-2}} \right] \cdot \boldsymbol{g}. \tag{7}$$

2.2. LBM for Solids

As Shan and Chen [10] specified, the non-local contribution of the collision-interaction operator should be integrated in a LBM scheme through its equivalent external force. Further argumentation about this assumption is detailed as an opening in the discussions of this paper (see Section 5.2). Therefore, we suggest using an external force related to the solid's Cauchy stress tensor, via its divergence:

$$\boldsymbol{g} = \frac{1}{\rho} \nabla_{\boldsymbol{x}} \cdot (\sigma). \tag{8}$$

It is worth noting that this expression is completely generic and might be applied theoretically to any constitutive law. More details about the constitutive law and material properties chosen for this study are detailed in Section 3. Moreover, the numerical evaluation of the divergence remains a critical part of the process.

Because the aim of the study is to define a framework for the use of Lattice Boltzmann Method for Solids (LBMS), it is necessary to validate it. Thus, as a first study, a validation of the static equilibrium of a solid is suggested. The static equilibrium of a solid is reached when there is no more relative movement inside the solid. Therefore, we naturally suggest introducing a zero-macroscopic speed equilibrium function, to ensure the convergence toward static equilibrium. Then, the equilibrium distribution function is simply given by:

$$f_i^{(eq)}(\rho, \boldsymbol{v} = 0) = w_i \rho. \tag{9}$$

This expression of the equilibrium distribution function is similar to those used in classical LBM for fluids when a null speed is imposed.

Using this model and through the numerical Chapman-Enskog expansion, we recover the undefined equation of static equilibrium:

$$\partial_t \rho = 0, \tag{10}$$

$$\nabla_{\boldsymbol{x}} \cdot (\sigma) = 0. \tag{11}$$

As a reminder, the Chapman–Enskog expansion studies the LBGKE by order of perturbation. Recombining equations obtained by order of perturbation, the macroscopic equations solved numerically by the LBM scheme are recovered.

3. Numerical Implementations and Algorithms

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.

3.1. Mechanical Introduction for Implementations

As a first study and implementation, solid cases that are as simple as possible are considered to build functions. 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 level, we consider the isotropic linear elastic constitutive relationship. In this framework, the Eulerian and Lagrangian descriptions can be considered as equivalent, yielding 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 other assumptions like the collision-less operator introduced in the discussion, which is more related to dynamics), we examine this proposition as a first study. Some of these requirements are further developed in Section 4 in order to set this static framework thoroughly.

3.2. 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 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 [32] 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 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 their ability to solve solid static equilibrium with the LBM. Library source code is available online [36], and main used numerical ingredients and scheme properties are thus freely accessible. Moreover, we refer the readers to Section 4 for more numerical details.

3.3. Additional Solid Routines

In order to not modify the classic LBM loop, additional independent solid routines are inserted into existing steps. This approach leads to appending new local operations across the lattice alongside the standard collide and stream stages, see Equations (2) and (3). 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 [37,38], discrete displacement values are the unknown variables of the system to solve. We use time integration to retrieve this required displacement from velocity, as detailed in Algorithm 1. Once the displacement tensor field $\mathbf{U}$ is recovered, the strain tensor field $\mathbf{E}$ is obtained from the displacement by numerical differentiation. One should remark that even if the equilibrium velocity is set to zero for our static state, it does

not imply that before reaching the "static convergence" the velocity is null in the domain. This emphasis the necessity of the Algorithm 1 in the numerical process. The stress tensor field $\boldsymbol{\Sigma}$ 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 $\nabla_x \cdot (\boldsymbol{\Sigma})$ 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.

Algorithm 1: Displacement Routine [Physical Units]

Input: Macroscopic velocity field v and time increment Δt
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
      $\mathbf{U}[G_x, G_y] \mathrel{+}= v^{cell} \times \Delta t$; // Time integration
    end
  end
end

Algorithm 2: Strain and Stress Routine [Physical Units]

Input: Macroscopic displacement tensor field $\mathbf{U}$, space discretization Δx and solid constitutive law C
Result: Macroscopic strain and stress tensor fields: $\mathbf{E}$ and $\boldsymbol{\Sigma}$
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($\mathbf{U}$, Δx, $\mathbf{P}$) ; // Strain computation
      $\boldsymbol{\Sigma}[G_x, G_y] =$ 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 $\boldsymbol{\Sigma}$ and space discretization Δx
Result: Macroscopic stress divergence tensor field $\nabla_x \cdot (\boldsymbol{\Sigma})$
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
      $\nabla_x \cdot (\boldsymbol{\Sigma})[G_x, G_y] =$ ComputeStressDivergenceFromStress$(\boldsymbol{\Sigma}, \Delta x, \mathbf{P})$; // $\nabla_x \cdot (\boldsymbol{\Sigma})[G_x, G_y]$ computation
      BlockLattice[b](x, y).ApplyLatticeBodyForce$(\nabla_x \cdot (\boldsymbol{\Sigma})[G_x, G_y])$; // Apply lattice body force to the current cell
    end
  end
end

These three routines allow the computation of required tensor fields over the lattice domain. They describe solid variables for each lattice cell at time t. In a two-dimensional space, the displacement tensor field has the form referenced in Equation (A36). In addition, considering mechanical applications with symmetric strain and stress tensors, strain and stress tensor fields are defined in Eqautions (A37) and (A38), 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:

$$\nabla_x \cdot (\boldsymbol{\Sigma}) = \begin{bmatrix} \nabla_x \cdot (\boldsymbol{\Sigma})[0,0] = \nabla_x \cdot (\sigma)_{0,0} = \begin{bmatrix} \frac{\partial \sigma_{xx}}{\partial \mathbf{X}} + \frac{\partial \sigma_{xy}}{\partial \mathbf{Y}} \\ \frac{\partial \sigma_{xy}}{\partial \mathbf{X}} + \frac{\partial \sigma_{yy}}{\partial \mathbf{Y}} \end{bmatrix}_{0,0} & \cdots & \nabla_x \cdot (\sigma)_{0,N_y-1} = \begin{bmatrix} \frac{\partial \sigma_{xx}}{\partial \mathbf{X}} + \frac{\partial \sigma_{xy}}{\partial \mathbf{Y}} \\ \frac{\partial \sigma_{xy}}{\partial \mathbf{X}} + \frac{\partial \sigma_{yy}}{\partial \mathbf{Y}} \end{bmatrix}_{0,N_y-1} \\ \vdots & \ddots & \vdots \\ \nabla_x \cdot (\sigma)_{N_x-1,0} = \begin{bmatrix} \frac{\partial \sigma_{xx}}{\partial \mathbf{X}} + \frac{\partial \sigma_{xy}}{\partial \mathbf{Y}} \\ \frac{\partial \sigma_{xy}}{\partial \mathbf{X}} + \frac{\partial \sigma_{yy}}{\partial \mathbf{Y}} \end{bmatrix}_{N_x-1,0} & \cdots & \nabla_x \cdot (\sigma)_{N_x-1,N_y-1} = \begin{bmatrix} \frac{\partial \sigma_{xx}}{\partial \mathbf{X}} + \frac{\partial \sigma_{xy}}{\partial \mathbf{Y}} \\ \frac{\partial \sigma_{xy}}{\partial \mathbf{X}} + \frac{\partial \sigma_{yy}}{\partial \mathbf{Y}} \end{bmatrix}_{N_x-1,N_y-1} \end{bmatrix}, \quad (12)$$

where $\nabla_x \cdot (\boldsymbol{\Sigma})$ 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 descriptors in order to modify distribution functions when colliding step is executed.

In addition to this general implementation, a particular attention should be paid the interfaces and boundary conditions. Indeed, the numerical method used should be able to capture the displacement and stress continuity or discontinuity, according to the given problem. The 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.

3.4. 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. Weighted Essentially Non-Oscillatory (WENO) [39] 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.

A programming task was performed, taking into account the successive use of different finite difference models as we move away from boundaries. Both forward and backward formulations are thus adopted; we refer the reader to library sources [36] for more details. For instance, around internal lattice nodes, a central difference scheme with a $\mathcal{O}(\Delta x^4)$ truncation error is built, see Equation (13). The first derivative along the first spatial direction $\mathbf{X}$ of any function Q with discrete values over the lattice can be expressed as:

$$\frac{\partial Q(G_x, G_y)}{\partial \mathbf{X}} = \frac{Q(G_x-2, G_y) - 8Q(G_x-1, G_y) + 8Q(G_x+1, G_y) - Q(G_x+2, G_y)}{12\Delta x} + \mathcal{O}\left(\Delta x^4\right), \tag{13}$$

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.

3.5. Proposed Solid Equilibrium and Colliding Step

As we aim to study static equilibrium state, we suggest using Equation (9), in order to relax towards a solid equilibrium. Moreover, the use of the Bhatnagar, Gross and Krook (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 3.6. However, our BGK collision operator for solids differs by its equilibrium distribution from the usual one used in CFD.

We tried to reach the static equilibrium through the BGK operator considering an equilibrium distribution without macroscopic speed. A simple distribution that does not depend on temperature, and ideally where matter does not move at the microscopic scale level, is targeted. However, for arithmetic reasons, mass is not concentrated on the cell's first discretized distribution function. The following numerical tests proved the functionality of the proposed distribution functions at solid equilibrium. From there, for each cell composing the lattice, distribution at solid static equilibrium for a classic $D2q9$ scheme can be defined by (see Equation (9)):

$$f_i^{(eq),sld}(\rho(\boldsymbol{x},t),\boldsymbol{v}) = f_i^{(eq)}(\rho(\boldsymbol{x},t),0) = w_i\rho(\boldsymbol{x},t) = \begin{cases} \frac{4}{9}\rho(\boldsymbol{x},t) & \text{if } i=0, \\ \frac{1}{9}\rho(\boldsymbol{x},t) & \text{if } i\in[1,\ldots,4], \\ \frac{1}{36}\rho(\boldsymbol{x},t) & \text{if } i\in[5,\ldots,8], \end{cases} \tag{14}$$

where $f_i^{(eq),sld}$ represents a discrete value i of $f^{(eq),sld}$ for a cell node (G_x, G_y). The presented distribution is applied both for initial conditions and equilibrium during time iterations. Hence, the proposed equilibrium only varies in space and time through density, and does not depend on the solid constitutive law chosen. Thus, mass conservation is assured at solid equilibrium thanks to the summation of Gauss–Hermite weights w_i. One can also note that in Equation (14) the common choice of ${c_s}^2 = \frac{1}{3}$ is chosen. However, in this special case of solid static equilibrium, the equations are of diffusion type. In such a case, the sound celerity can be chosen freely since the BGK or Two Relaxation Times (TRT) operators are unconditionally stable [40].

The colliding step (see Equation (2)) is then modified to include the above suggested solid equilibrium and related force calculated from stress divergence. For instance, consid-

ering an internal node, not on boundaries, the discretized collision operator is expressed by:

$$\Omega\left(f_i(G_x, G_y, t), f_i^{(eq),sld}\right) = -\omega\left(f_i(G_x, G_y, t) - f_i^{(eq),sld}) + s_i(G_x, G_y, t\right), \forall i \in [0, \ldots, 8]. \quad (15)$$

Observe that $f_i(G_x, G_y, t)$ and $f_i^{(eq),sld}$ depends on space and time whereas $f_i^{(eq),sld}$ is also 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 because 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 [35], see Equations (2) and (7). This choice was made in the name of simplicity, even if it known to be related to some physical inconsistency, whereas the same remark is true independently for some collision operators [41].

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 for the addition of new solid objects without modifying initial PaLaBoS sources enabling the interoperability between different libraries.

3.6. Proposed Modifications for Solid Boundary Conditions

To deal with solids in an 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 follows. 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 distinguish the boundary behavior from the 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 a body force in internal nodes and shall alone govern inner behavior. This can be physically interpreted as if non-zero velocity boundary conditions being immediately converted into solid forces during iterations.

Although, all of these conditions are not perfect and subject to improvement, indeed works on highly accurate boundary conditions on the moving surface [42] could be exploited. Yet, the simple and perfectible boundary conditions used here lead to correct results, see Section 4. For more details concerning the implementation, see the available source code [36].

Algorithm 4: Boundary Conditions' Routine

Input: Boundary relaxation parameter ω_c and solid equilibrium $f^{(eq),sld}$
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(ω_c, $f^{(eq),sld}$) ; `// Relax current boundary cell`
end

3.7. 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 execut-

ing 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 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 4.

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 $\omega \in [0, \ldots, 1]$ 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 due to the truncated diffusive term which scales with the diffusion coefficient, here $\frac{1}{3}\left(\frac{1}{2} - \frac{1}{\omega}\right)$.

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, useful additional comprehension can be retrieved from this repository.

Algorithm 5: LBMS Loop

Input: Static equilibrium problem (isotropic materials)
Result: Solved static equilibrium state and related macroscopic variables

```
Initialization (see library sources [36]) ;          // Other necessary initializations
Lattice.SetPeriodicity(false) ;                      // Periodicity = false
Lattice.InitializeAtSolidEquilibrium(f^(eq),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 ← 0 to (i × Δt) < 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
```

3.8. Proposed Method and Algorithms in a Nutshell

To sum up, in order to extend the LBM to solid static equilibrium, the proposed method is to apply the LBM formalism through the Algorithm 5 (leading to the LBMS). Algorithm 5 itself relies on the previous Algorithms 1–4 and consequently on Equations (2)–(9). In addition to this generic method, a constitutive law needs to be added to close the system. In our following validation, a linear isotropic elastic behaviour is used which leads to the Equation (16), and which is further detailed in the following section.

4. Validations and Benchmarks

To validate the previous physical interpretations, LBMS developments and C++ algorithms, results are compared with an FEA software. The reference model must be accurate and faithful under mechanical framework assumptions. These validations are performed in a two-dimensional space for simplicity and clarity using a deformable solid body. Throughout this section, all numerical values are given in the International System of Units (SI) base units if units are not mentioned.

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

4.1.1. Mechanical Prerequisites

We resume some aspects partially developed in Section 3.1 concerning isotropic linear elastic materials, Eulerian and Lagrangian frameworks. Details are given to highlight with preciseness the type of matter used and useful assumptions to validate the LBMS.

Firstly, we point out that the 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 Equation (16) 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:

$$\boldsymbol{\Sigma}[G_x, G_y] = C \,:\, \mathbf{E}[G_x, G_y] = \lambda \,\mathrm{Tr}\big(\mathbf{E}[G_x, G_y]\big) I + 2\mu \mathbf{E}[G_x, G_y]. \tag{16}$$

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

Secondly, we want to superimpose the deformed configuration and the initial one. In order to do that, the involved displacement field must be small in terms of its norm. The material solid structure is then considered motionless during its transformation whereas the body remains deformable. In fact, all mechanical tensors such as displacement, stress, and strain are computed considering initial configuration, thus simplifying the problem. Furthermore, to be able to consider linearized strains, deformation must represent less than 1% of a solid characteristic dimension.

Thirdly, in 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. The Lagrange approach is widely preferred for solids where materials move with the coordinate system including elements. Basically, it can be explained by the fact that no matter passes from one element to another. On the other hand, in CFD, the mesh or lattice is fixed. Because the lattice structure is supposed to be stationary throughout time, this no-distinction assumption can be made freely. In substance, the considered trajectories of solid particles can be insignificant, thus allowing for the merging of 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 [43].

4.1.2. Mechanical Study

The comparison study setup is quite simple. All borders are affected with fixed boundary conditions except one where displacement is imposed; see Figure 1 for more details. This enforced displacement is formulated as follows:

$$\boldsymbol{u} = 0\mathbf{X} - 0.002\mathbf{Y}, \tag{17}$$

where 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.

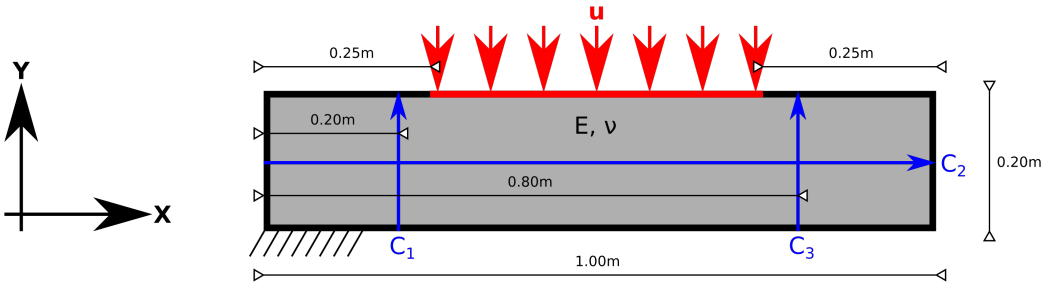


Figure 1. Mechanical study for LBMS approach validation. Black frontiers are fixed boundary conditions. A displacement is imposed across the vector $\boldsymbol{u}$ described with black, as is depicted on the left. Solid material properties are defined through Young's modulus E and Poisson's ratio ν.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

4.1.3. LBMS: Setup

Taking into account the previously targeted mechanical study, a lattice characterized by a $D2Q9$ scheme is sampled with a resolution of 200 nodes per meter, yielding a suitable discretization. In this spirit, a mesh convergence study was made and it shows an improvement of the solution quality when lattice becomes finer; see Figures A1 and A2 for more details. 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's nature.

To simulate an imposed displacement condition, a velocity boundary condition is applied during a certain amount of time and then reduced to zero. The previous process allows solid equilibrium to be reached which is established after a couple of supplementary iterations following the change in value of imposed macroscopic speed. Displacement 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 the sources of the LBMS library.

We aim to keep the initial configuration as close to the deformed one in order to compute fields onto the same lattice geometry regardless of the transformation. Furthermore, throughout the iterative LBMS process, error accumulation is prevented by using a linear behavior implicating linearized strains: 1% of deformation is, thus, the maximum amplitude that we allow ourselves for model validation.

4.1.4. 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 [33] as the FEA preprocessor and solver for its simplicity of use. However, some simulation options remain important to set even to resolve a standard static problem.

As mentioned above, linearized strains are assumed and in the way forced by default. Because displacement is considered non-existent, all stresses are supposed to be true. Hence, computed Cauchy stress can be directly compared with the tensor field $\boldsymbol{\Sigma}$ from the 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 when investigating results.

A standard extra fine triangular mesh is generated with fully integrated elements using cubic shape functions. This setup enables accurate computations of 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 [37,38,43] later presented in this work, 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., [44]. 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; however, an hourglass control then becomes necessary.

4.2. Results Comparison with 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., $\nu \in [-0.99, \ldots, 0.49]$ and $E \in [2 \times 10^3, \ldots, 2 \times 10^{11}]$. They show the undeniable capability of the LBM to solve solid mechanics cases.

4.2.1. Case E = 200 GPa, ν = 0.15

Firstly, we decide to illustrate a specific case, i.e., with material properties E = 200 GPa, ν = 0.15. This example shows typical results of our presented framework. Figure 2 contrasts the reference study with the proposed LBMS study. 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 the displacement field arising from the used finite difference scheme. As we mentioned 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 the deformed shape and displacement field magnitude of the lattice computed with our LBMS approach.

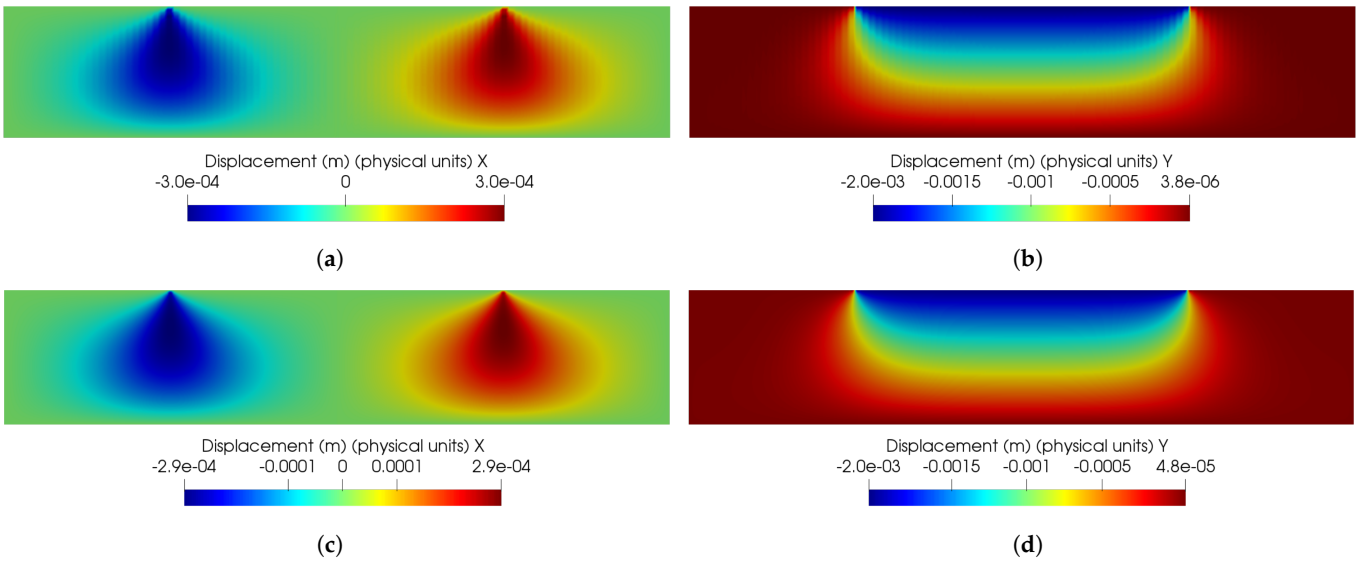
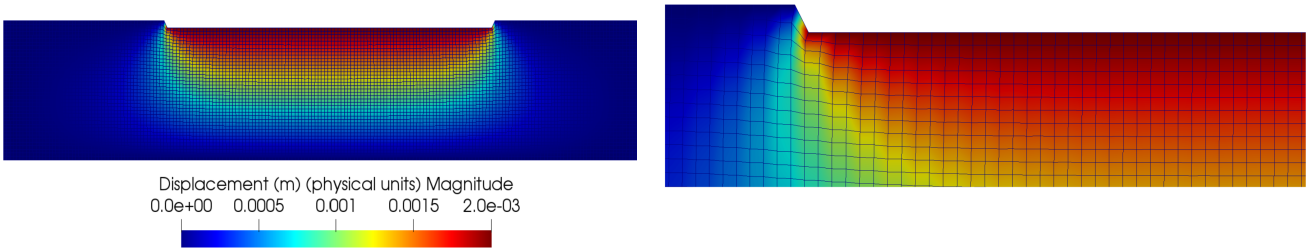


Figure 2. 2D results comparison of displacement field for case E = 200 GPa, ν = 0.15 per field component. (**a**) Displacement along X axis, with LBMS. (**b**) Displacement along Y axis, with LBMS. (**c**) Displacement along X axis, with COMSOL. (**d**) Displacement along Y axis, with COMSOL.



(**a**) (**b**)

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

Figures 4 and A3 depict the Von Mises stress distribution along cutting curves C_1 and C_2, respectively (see Figure 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 an FEA approach, stresses values are strongly dependent on the computed displacement field accuracy. Plotted curves show a similar global behavior and the same order of magnitude (no more than 15% of difference in Figure A3), although Figure 4 has a striking match with the reference study. These first results lead to the proposed method having an interesting potential.

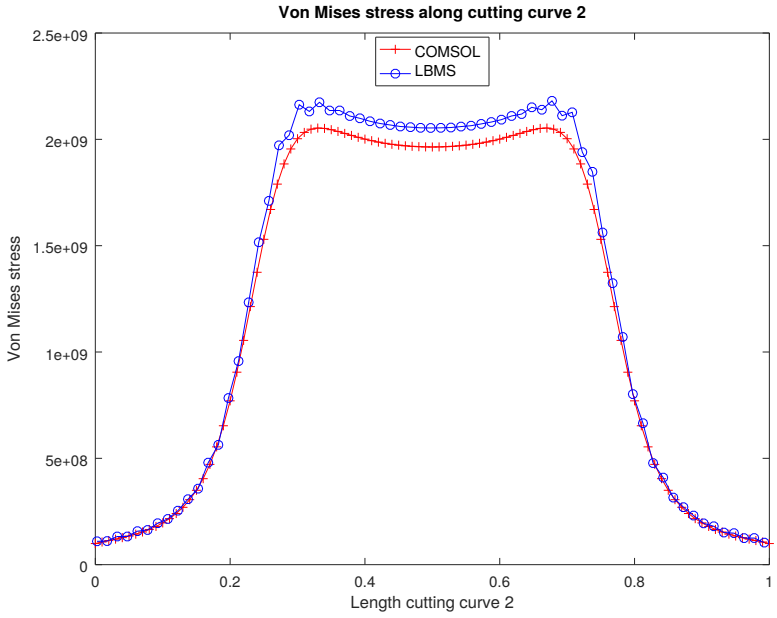


Figure 4. Von Mises stress along cutting curve C_2 for case E = 200 GPa, ν = 0.15.

4.2.2. 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 = 200 GPa. 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, A4, A6 and A11 exhibit results with negative ν whereas Figures A5, A7, A10 and A12 present positive ones. For $\nu \in [-0.50, \ldots, 0.35]$ 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 Figures A8 and A9 for the first displacement direction and in Figures A13 and A14 for the second one.

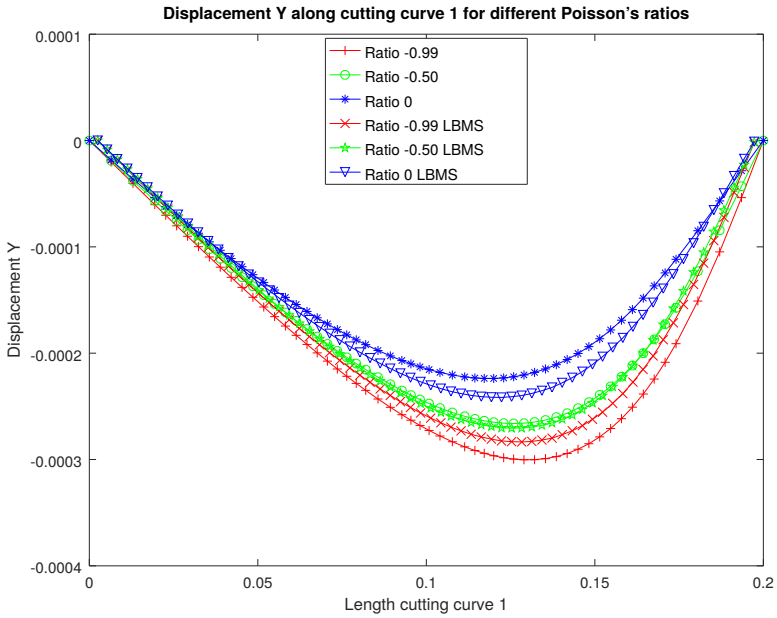


Figure 5. Second displacement components along cutting curve C_1 for negative Poisson's ratios.

The results shown are acceptable in terms of matching with the reference study. Differences between curves are more pronounced when poisson's ratio is moving away from zero. Further detailed explanations are presented in Section 5.

4.2.3. Young's Modulus Sensitivity

Like in the previous part, we perform a parametric study involving Young's modulus. Poisson's ratio is set to zero, leading to $\lambda = 0$. Thus, the isotropic law is modified and therefore the Poisson effect does not exist for this particular case. Figures A15 and A16 illustrate the first displacement coordinate along cutting curves C_1 and C_2 respectively. In the same way, Figures A17 and A18 depict the second displacement coordinate.

At boundaries, 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 Equation (16)), 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 the insensitivity and stability of LBMS with various modulus. To go further, we remark that the first component of displacement partly arises from intense shear stress at the frontiers of the velocity condition. A non-constant shear stress along second basis direction produces a non-zero stress divergence, i.e., contributing to its first component and hence yielding a displacement even if $\nu = 0$.

4.2.4. Triangular Loading: Case E = 200 GPa, 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 the previous case in Section 4.2.1, 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 Figure 1). Figure 6 presents the 2D results compared with the reference study. In general, the displacement field is consistent and replicates the expected solid behavior with fidelity. In detail, minor errors are detected in field values leading to the same conclusions as above. Additional curves are given in Figures A19 and A20 for Von Mises stress and in Figures 7, A21–A23 for displacement components along cutting

curves C_2 and C_3. Plots are in total concordance with the related FEA study despite slight deviations when displacement is important.

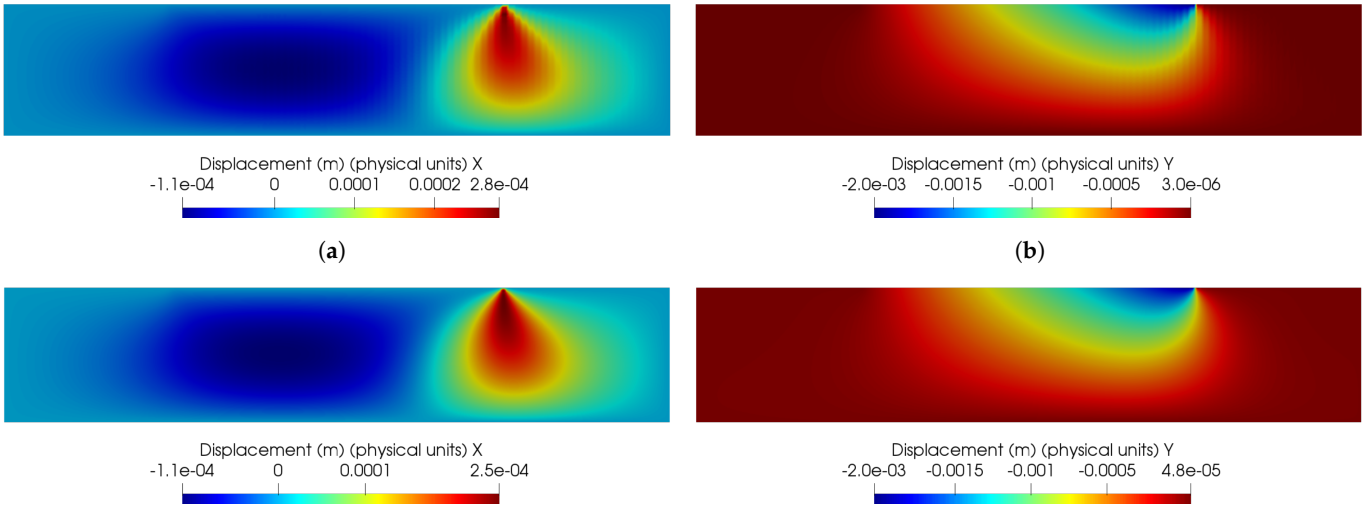


Figure 6. 2D results comparison of displacement field for case E = 200 GPa, ν = 0.15 per field component and with a triangular loading profile. (**a**) Displacement along X axis, with LBMS. (**b**) Displacement along Y axis, with LBMS. (**c**) Displacement along X axis, with COMSOL. (**d**) Displacement along Y axis, with COMSOL.

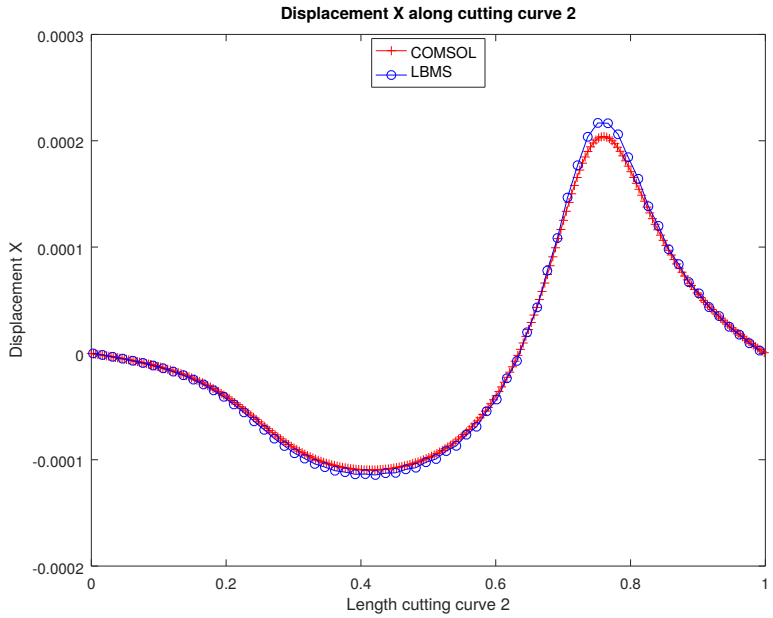


Figure 7. The first displacement component along cutting curve C_2 for case E = 200 GPa, ν = 0.15 with triangular loading.

5. Discussions and Outlooks

5.1. Obtained Results and Discussion

In addition to the global curves matching, which gives a qualitative validation, a quantitative analysis is provided. Thus, Table 1 shows the Root Mean Square Error (RMSE) and Normalized Root Mean Square Error (NRMSE) between the computed results with the LBMS and FEA, presented through Figures 4, 5 and 7. The results show a great precision of the LBMS compared to the FEA method, and this is 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 the computed fields, which yields only a few percents of error. These errors values tend to quantitatively validate 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 allows us to conclude 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 presented in this study, especially the incorporation of the stress divergence as a mean-field force.

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

Loading	Curve	Field	E (GPa)	ν	RMSE	NRMSE
Rectangular	C_2 (Figure 4)	Von Mises stress	200	0.15	7.37×10^7	3.77%
Rectangular	C_1 (Figure 5)	Displacement **Y**	200	−0.99	1.52×10^{-5}	5.01%
Rectangular	C_1 (Figure 5)	Displacement **Y**	200	−0.5	4.41×10^{-6}	1.66%
Rectangular	C_1 (Figure 5)	Displacement **Y**	200	0.0	1.23×10^{-5}	5.50%
Triangular	C_2 (Figure 7)	Displacement **X**	200	0.15	1.52×10^{-5}	2.01%

Concerning the introduced boundary conditions for solids, some work remains in order to enhance adaptation for solids. Currently, it has been pointed out that fixed conditions are far from perfect and suitable for 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, the LBMS method detains an iterative process so as to arrive at solid static equilibrium; thus, the finite difference scheme accuracy could play an important role concerning quality of the calculated results. Moreover, we point out some post-processing issues which may be at the origin of the errors. Indeed, along cutting curves, the FEA displacement field is mainly interpolated whereas the LBMS field hardly is. All of these considerations can cause additional deviations from the 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 to reach the equilibrium state (i.e., when stress divergence tends towards a null vector), yielding completely different pipelines between FEA and LBMS. Furthermore, as mentioned above in Section 4.1.4, the used plane stress assumption for two-dimensional reference studies might trigger some of the 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 better accuracy.

All these achievements seem promising in terms of the use of the Boltzmann–Vlasov approach for solid dynamics. It is 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.). In contrast to previous attempts to cope with solid behavior with the LBM, our approach does not require statistical distribution of forces [18,45] nor springs [24]. Both bring more complexity, heavier computations, are further away from the Boltzmann theory, and are less prone to FSI coupling.

For our validation case, we do not have an analytical solution due to boundary conditions issues. A simpler test case is not easily achievable in a two dimensional domain. Concerning numerical correlations, the standard FEA solution is completely approved by

the mechanical community and hence serves as a reference solution. We further point out that both methods (i.e., LBMS and FEA) are truly different and must be compared carefully, see Table 2 for more details. Our proposed LBMS method is not very competitive in terms of computational time, because it was developed to evolve to solve complex FSI dynamic problems. In these mechanical cases, FEA runs are time-consuming calculations and the previously explained trend could be drastically reversed. On the other hand, the LBM approach can trade its "locality" for an improved computational time in steady-state cases as it has been illustrated in [40]. However, because our focus is on FSI dynamic problems, the latter approach was not used here. However, it contributes to the explanation of the limited computational time competitiveness of the suggested method.

Table 2. Comparison between LBMS and standard FEA methods to compute a solid static equilibrium. Because LBMS method is intrinsically dynamic, results are not competitive for static studies.

Method	Computation Time (s)	Iterations	Dependencies	FSI Ready?	Is Dynamic?
LBMS	150	49,000	PaLaBoS	Yes	Yes
FEA	8	N/A	FEM solver	No	No

5.2. Considerations about the Vlasov Equation

Despite the previous attempts of LBM for solids mentioned in Section 1.1, to the best of the authors' knowledge, none of them use the physical representation of the particles of the Boltzmann equation. Indeed, the method previously used to derive an LBM–like method is considering the LBM equation as a pure mathematical framework to solve a set of Partial Differential Equations (PDE)s, without relying on the original fluid-based historical method. These previous mathematical methods found a set of advection–relaxation equations for new variables that are equivalent to a given macroscopic model. Such mathematical approaches contrast with the suggested method. We propose keeping this link between particles and the Boltzmann method. To do so, we find the similarities between plasma and solids. In both cases, the motion of particles is almost not subject to collisions: in the plasma case, the Knudsen number is large $Kn \gg 1$ and particles are collisionless in the free-flight regime; whereas in solids $Kn \to 0$ and particles are collisionless, because a particle does not move freely without interacting with the surrounding particles. Even if these two regimes should not be confused, they share some physical and mathematical concepts resulting in the fact that in both cases the collision term in the Boltzmann equation vanishes. Such a collisionless Boltzmann equation is called a Vlasov equation [46].

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 a 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 of Dirac distribution centered on the velocity of the the surrounding particles, $f(\boldsymbol{x}, \boldsymbol{\xi}, t) = \rho\delta_v(\boldsymbol{\xi})$. Then, the evolution of the particles described by this equation is a propagation of the particles without further modifications induced by the collisions.

The interesting idea behind the Vlasov equation for our purpose 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, which is the concept used in Equation (8). Therefore, considering the Vlasov equation of Dirac distribution with a mean-field divergence of Cauchy's stress, we obtained the following mesoscopic model:

$$\frac{\partial f}{\partial t}(\boldsymbol{x},\boldsymbol{\xi},t)+\boldsymbol{\xi}\cdot\nabla_x(f)+\boldsymbol{g}\cdot\nabla_{\boldsymbol{\xi}}(f)=0, \tag{18}$$

$$f(\boldsymbol{x},\boldsymbol{\xi},t)=\rho\delta_v(\boldsymbol{\xi}), \tag{19}$$

$$\boldsymbol{g}=\frac{1}{\rho}\nabla_x\cdot(\sigma). \tag{20}$$

Then, using the passage formula for the two firsts moments $(1,\boldsymbol{\xi})$, the desired macroscopic system of the undefined equation of motion for solids is obtained from a theoretical point of view:

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

$$\partial_t(\rho v)+\nabla_x\cdot(\rho v\otimes v)=\nabla_x\cdot(\sigma). \tag{22}$$

Thus, the Boltzmann–Vlasov equation leading to the previous system of equations seems suitable for rigid matter simulations.

Several remarks arise from these previous equations. This approach remains general for any constitutive law. Thus, the use of the Dirac distribution is somehow closer to the conventional representations of solid lattices: particles are not moving freely through the solid. In addition, the current discussion about the Vlasov equation for solids, is certainly presenting a working framework; however, the founding of this Vlasov equation requires a mathematical proof which is left for future work. Instead, we test here numerically some of the assumptions relative to this framework that are restricted to static equilibrium.

5.3. Global Outlook and Proposed Improvements

C++ programmed classes and methods integrated into LBMS library can be drastically enhanced, i.e., mostly for efficiency purposes. Although the library is robust, computation time might be substantially reduced if other guidelines are adopted.

For presented cases, tens or hundreds of seconds are necessary to reach the illustrated static equilibrium assuming sequential runs on an Intel(R) Xeon(R) CPU E5-1650 v3 @ 3.50 GHz, thus, involving a few tens of thousands of LBMS iterations. In contrast, a couple of seconds are enough to COMSOL even if the mesh is modeled very fine, because a standard static resolution is requested. Effectively, the involved computation times are hardly comparable because the methods are very different. In fact, LBMS is naturally dynamic due to its roots coming from the LBM and thus it requires an evaluation of the solid state at each time iteration. Concerning lattice discretization, additional experiments should be conducted in order to discover space convergence properties and effects.

Thanks to the pointed reproducibility of the depicted numerical results, complements to existing objects can be made freely in order to improve calculation efficiency. However, here stands 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, especially on Graphics Processing Unit (GPU) 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 part of the future work. For example, corners demand a special therapy because cell vicinity is restricted. Another possible way is to use robust mean-field forces as already implemented into the Shan-Chen approach [10]. Moreover, as mentioned above, WENO schemes could help us.

Another advantage of the LBM compared to FEA, is this ease of moving 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 a better comparison of the complex geometries with current methods usually dedicated to solving mechanical problems. The door is thus wide open in terms of further intensive analysis or significant

studies. In addition, the parallelized strategy by blocks remains unchanged for solids. However, depending on the used finite difference schemes, some complementary properties are needed for block envelopes.

Furthermore, for the broad range of Young's modulus and Poisson's ratios simulated, the presented method must 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 is more deeply characterized.

Only the static equilibrium has been compared to existing methods. Because 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 next step. The numerical stability for the relaxation parameter $\omega < 0.25$ is already unconventional LBM and supports the theoretical developments introduced here.

Validation of the dynamic behavior requires additional work, especially for the reference mechanical study. Indeed, the transitional state before reaching static equilibrium is hard to confirm taking into account the actual state of our research and LBM groundwork for solids.

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 Equation (15)):

$$\Omega\Big(f_i(G_x, G_y, t), f_i^{(eq),sld}, s_i(G_x, G_y, t)\Big) = -\omega f_i(G_x, G_y, t) + \omega\Big(f_i^{(eq),sld} + s_i(G_x, G_y, t)\Big), \forall i \in [0, \ldots, 8]. \quad (23)$$

This latter formulation yields a new form of solid equilibrium where solid body forces are integrated into the equilibrium function; thus, the stress divergence term may be integrated into the BGK operator if the relaxation parameter is equal to 1. Moreover, such an approach may looks like an artifact because with $\omega = 1$, the LBM is equivalent to a particular Finite Difference Method (FDM), with more complex boundary conditions. Thus, the previous equilibrium distribution deserves further dedicated investigations, beyond the work.

Of course, many applications are plausible from the next development of the LBMS, especially regarding 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, the solid–liquid interface leads to FSI in order to study the behavior of a solid product immersed in a liquid. A recent publication [47], illustrates the fact that a unified theory is needed to solve complex FSI phenomena, taking into consideration both fluid and solid characteristics with LBM.

6. Conclusions

We have introduced in this work a new method to solve solid static equilibriums from standard LBM. Our strategy depicts promising results concerning the static state of isotropic materials. The presented plots and figures illustrate well the efficiency of the presented method. Such a task is achieved thanks to detailed developments and a robust framework. In fact, the presented method is secured with a theory groundwork and then implemented using classical programming tools.

A first key contribution was introduced thanks to theoretical developments around the Boltzmann–Vlasov equation. Furthermore, the stress tensor divergence was introduced as a mean-field force term. These two considerations allowed for the building of a theoretical framework to deal with solids, with the LBM. This framework is unprecedented and is very unique compared to previous works on the subject.

Furthermore, we produced new programmed objects included in a user-friendly research library. Although this implementation is not optimal, the developed classes and methods are robust enough and trustworthy concerning the conclusions given on plots and figures. On the other hand, the adopted general implementation approach detains

several advantages. Non-invasive coding work permits the reproduction of results with a bewildering ease. Likewise, a 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; only the dynamic related to solids is adjusted. Previous abilities, among others, allowed for modeling of fluids and solids in the same framework.

The shown results are validated through comparative plots including a large range of parametric studies. They clearly prove the credibility of the proposed approach. Furthermore, for a significant number of plotted data, the 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 is also allows us to imagine a complete confirmation of the general theory developed. Moreover, the reliable and stable results obtained at very high relaxation time ($\omega \leq 0.01$) show the strength of the theory and the perspectives for solid dynamics.

Despite the promising illustrated results contrasted with reference studies, a lot of work still needs to be done to exploit the 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 work. This work also tries to answer to increasing expectations in numerical simulation fields.

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

The solid–solid contact problem 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 in investigating computational solid dynamics. In contrast, a real dynamic case may still require supplementary efforts. Moreover, complex solid constitutive laws, such as non-linear ones, must be tested and contrasted with the adapted reference results. In addition, the extension to 3D should be natural and in consequence could produce relevant results. In this work, ideas related to solid distribution functions were given, but they undeniably need further work so as to gain a deeper comprehension of solid equilibrium. For all of these reasons, we aim to share the presented advances for the benefit of the community.

Author Contributions: Conceptualization, R.N.; Funding acquisition, L.N.; Investigation, T.M.; Methodology, T.M. and R.N.; Project administration, L.N.; Software, T.M.; Validation, G.C. and L.N.; Writing—original draft, T.M., R.N. and L.N.; Writing—review and editing, T.M., R.N., G.C. and L.N. All authors have read and agreed to the published version of the manuscript.

Funding: 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.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are available from reasonable request to the authors.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BE	Boltzmann Equation
BGK	Bhatnagar, Gross and Krook
CFD	Computational Fluid Dynamics
FDM	Finite Difference Method
FEA	Finite Element Analysis
FEM	Finite Element Method
FPD	Fundamental Principe of the Dynamics
FSI	Fluid Structure Interaction
FVM	Finite Volume Method
GPU	Graphics Processing Unit
LBE	Lattice Boltzmann Equation
LBGKE	Lattice Boltzmann-BGK Equation
LBM	Lattice Boltzmann Method
LBMS	Lattice Boltzmann Method for Solids
LGCA	Lattice Gas Cellular Automata
MPM	Material Point Method
MRT	Multiple Relaxation Times
TRT	Two Relaxation Times
NRMSE	Normalized Root Mean Square Error
NSE	Navier-Stokes Equations
NSF	Navier-Stokes-Fourier
PaLaBoS	Parallel Lattice Boltzmann Solver
PDE	Partial Differential Equations
QE	Quasi-Equilibrium
REV	Representative Elementary Volume
RMSE	Root Mean Square Error
SPH	Smoothed Particle Hydrodynamics
SI	International System of Units
WENO	Weighted Essentially Non-Oscillatory
WSS	Wall Shear Stress

Nomenclature

ρ	mass density
D	physical space dimension
f	density distribution function over velocity space
$\boldsymbol{\xi}$	microscopic velocity of particles
$\boldsymbol{v}$	macroscopic speed field
$\boldsymbol{u}$	macroscopic displacement field
σ	Cauchy stress tensor
ε	linearised strain tensor
B	left Green-Cauchy strain tensor
H	Gibbs-Boltzmann entropy
s	entropy
$\boldsymbol{q}_\theta$	heat flux
r	radiation
$\boldsymbol{g}$	mass force field
$\Omega(.,.)$	collision operator
ω	relaxation frequency
p	pressure scalar
I	identity square matrix $D \times D$
Π	viscous stress tensor
$\boldsymbol{c}$	microscopic velocity in the $\boldsymbol{v}$ frame
θ	thermodynamical absolute temperature
$\boldsymbol{x}$	coordinate Eulerian vector field
t	time

E_k	kinematic energy
E_{int}	internal energy
E_θ	thermal energy
E_{tot}	total energy
E_{def}	deformation energy
ψ	collision invariant
w	Gauss measure
$\mathcal{H}_n$	n-th order Hermite polynomial
a_k	coordonate on the Hermite basis
$\mathcal{P}_n$	n cyclic permutation
ν	kinematic viscosity
λ	bulk viscosity
κ	thermal conductivity
$f^{\mathcal{N}}$	truncated density distribution
$\boldsymbol{\xi}_i$	discretized particles velocity
w_i	discrete weight of the Gauss quadrature
f_i	discretized density distribution
$\widehat{f}_i$	discretized velocity density distribution function for the numerical scheme
$\mathcal{N}$	troncature order over the Hilbert basis
m	algebraic precision of the quadrature
q	number of discretized speed used in the lattice
δ_x	Dirac distribution centred on x
m_k	math m
M_M	matrix transformation from distribution space to moments space
S_M	diagonal matrix containing the (multiple) relaxation time of each moments
Δt	increment of time
$\Delta \boldsymbol{x}$	increment of space
c_s	celerity of the sound
$I[\cdot]$	integral operator of a given quantity over the velocity space
$f^{(eq)}$	equilibrium distribution given by the Maxwell-Boltzmann distribution
$f_i^{(eq)}$	discretized equilibrium distribution given by the Maxwell-Boltzmann distribution
g_i	forcing term projected over the velocity space
s_G	variance of the Gaussian measure
ϵ	perturbation order
$\mathcal{A}$	any extensible physical quantity
R	ideal gas constant
E_P	electrical field
q_P	electrical charge of a particle
m	mass of a particle
C	fourth-order stiffness tensor
λ	Lamé's first parameter
μ	shear modulus
$\overline{I}_n$	symmetric part of the nth-rank identity tensor
E	Young's modulus
ν	Poisson's ratio
$\overline{\overline{C}}$	second-order stiffness tensor
x	first coordinate of a cell in a block lattice
y	second coordinate of a cell in a block lattice
G_x	global position x of a cell in the lattice
G_y	global position y of a cell in the lattice
$\mathbf{X}$	first direction of basis
$\mathbf{Y}$	second direction of basis
$\mathbf{U}$	displacement tensor field over the whole lattice
$\mathbf{E}$	strain tensor field over the whole lattice
$\mathbf{\Sigma}$	stress tensor field over the whole lattice
N_x	number of lattice nodes in the first direction
N_y	number of lattice nodes in the second direction
Q	any function with discrete values over the whole lattice
s_i	source forcing term projected over the velocity space
ω_c	boundary relaxation parameter

Appendix A. Chapman–Enskog Expansion for Static Solid Equilibrium

In order to keep the second order accuracy, the following substitution is performed [48]:

$$\frac{1}{\widehat{\omega}} = \frac{1}{\omega} + \frac{\Delta t}{2}, \tag{A1}$$

$$\begin{aligned}\widehat{f}_i(\boldsymbol{x},t) &= f_i - \frac{\Delta t}{2}\overline{\Omega_i}(t)\\ &= f_i + \frac{\Delta t\omega}{2}\left(f_i(\boldsymbol{x},t) - f_i^{(eq)}(\boldsymbol{x},t)\right) + \frac{\Delta t}{2}g_i(\boldsymbol{x},t)\\ &= f_i + \frac{\Delta t\widehat{\omega}}{2-\widehat{\omega}\Delta t}\left(f_i(\boldsymbol{x},t) - f_i^{(eq)}(\boldsymbol{x},t)\right) + \frac{\Delta t}{2}g_i(\boldsymbol{x},t),\end{aligned} \tag{A2}$$

which leads to the discretized version of the BE, i.e., the LBGKE expressed by:

$$\widehat{f}_i(\boldsymbol{x}+\boldsymbol{\xi}_i\Delta t, t+\Delta t) = \widehat{f}_i(\boldsymbol{x},t) - \Delta t\widehat{\omega}\left(\widehat{f}_i(\boldsymbol{x},t) - f_i^{(eq)}(\boldsymbol{x},t)\right) - \Delta t\left(1-\frac{\widehat{\omega}\Delta t}{2}\right)g_i(\boldsymbol{x},t) + \mathcal{O}\left(\Delta t^3\right). \tag{A3}$$

Then, using the Taylor expansion of $\widehat{f}_i$ in Equation (A3), we get:

$$\begin{aligned}\widehat{f}_i(\boldsymbol{x}+\boldsymbol{\xi}_i\Delta t, t+\Delta t) &= \widehat{f}_i(\boldsymbol{x},t) + \sum_{m=1}^{\infty}\frac{1}{m!}\left[\Delta t\partial_t + \boldsymbol{\xi}_i\Delta t\cdot\nabla_{\boldsymbol{x}}\right]^m\widehat{f}_i(\boldsymbol{x},t)\\ &= \widehat{f}_i(\boldsymbol{x},t) - \Delta t\widehat{\omega}\left(\widehat{f}_i(\boldsymbol{x},t) - f_i^{(eq)}(\boldsymbol{x},t)\right) - \Delta t\left(1-\frac{\widehat{\omega}\Delta t}{2}\right)g_i(\boldsymbol{x},t) + \mathcal{O}\left(\Delta t^3\right).\end{aligned} \tag{A4}$$

Therefore, the LBGKE becomes:

$$\sum_{m=1}^{\infty}\frac{1}{m!}\left[\Delta t\partial_t + \boldsymbol{\xi}_i\Delta t\cdot\nabla_{\boldsymbol{x}}\right]^m\widehat{f}_i(\boldsymbol{x},t) = -\Delta t\widehat{\omega}\left(\widehat{f}_i(\boldsymbol{x},t) - f_i^{(eq)}(\boldsymbol{x},t)\right) - \Delta t\left(1-\frac{\widehat{\omega}\Delta t}{2}\right)g_i(\boldsymbol{x},t). \tag{A5}$$

The modal decomposition is introduced by the time-scales (ϵ) power series decomposition such as:

$$\Delta t\partial_t = \sum_{k=1}^{\infty}\epsilon^k\partial_t^{(k)}, \tag{A6}$$

$$\Delta t\nabla_{\boldsymbol{x}} = \epsilon^1\nabla_{\boldsymbol{x}}{}^{(1)}, \tag{A7}$$

$$\widehat{f}_i = \sum_{k=0}^{\infty}\epsilon^k\widehat{f}_i{}^{(k)}, \tag{A8}$$

$$g_i = \sum_{k=1}^{\infty}\epsilon^k g_i{}^{(k)} = \epsilon^1 g_i{}^{(1)}. \tag{A9}$$

The source term is local, so its perturbation of the distribution function will be brought through time. furthermore, the source term g_i is limited to the first order of perturbation here.

The previous equation is true for every order ϵ^k. So, the three first orders lead to the following equations:

$$\epsilon^0: \qquad -\Delta t\widehat{\omega}f_i^{(eq)} = -\Delta t\widehat{\omega}\widehat{f}_i{}^{(0)}, \tag{A10}$$

$$\epsilon^1: \qquad \left[\partial_t^{(1)} + \boldsymbol{\xi}_i\cdot\nabla_{\boldsymbol{x}}{}^{(1)}\right]\widehat{f}_i{}^{(0)} + \Delta t\left(1-\frac{\widehat{\omega}\Delta t}{2}\right)g_i{}^{(1)} = -\Delta t\widehat{\omega}\widehat{f}_i{}^{(1)}, \tag{A11}$$

$$\epsilon^2: \qquad \frac{1}{2}\left[\partial_t^{(1)} + \boldsymbol{\xi}_i\cdot\nabla_{\boldsymbol{x}}{}^{(1)}\right]^2\widehat{f}_i{}^{(0)} + \partial_t^{(2)}\widehat{f}_i{}^{(0)} + \left[\partial_t^{(1)} + \boldsymbol{\xi}_i\cdot\nabla_{\boldsymbol{x}}{}^{(1)}\right]^1\widehat{f}_i{}^{(1)} = -\Delta t\widehat{\omega}\widehat{f}_i{}^{(2)}. \tag{A12}$$

After simplification and reinjection of previous order, these expressions can be simplified:

$$f_i^{(eq)} = \widehat{f}_i^{(0)}, \tag{A13}$$

$$\left[\partial_t^{(1)} + \boldsymbol{\xi}_i \cdot \nabla_{\boldsymbol{x}}{}^{(1)}\right] \widehat{f}_i^{(0)} = -\Delta t \widehat{\omega} \widehat{f}_i^{(1)} - \Delta t \left(1 - \frac{\widehat{\omega}\Delta t}{2}\right) g_i^{(1)}, \tag{A14}$$

$$\partial_t^{(2)} \widehat{f}_i^{(0)} + \left[\partial_t^{(1)} + \boldsymbol{\xi}_i \cdot \nabla_{\boldsymbol{x}}{}^{(1)}\right]^1 \left(\left(1 - \frac{\Delta t \widehat{\omega}}{2}\right) \widehat{f}_i^{(1)} - \frac{\Delta t}{2}\left(1 - \frac{\widehat{\omega}\Delta t}{2}\right) g_i^{(1)}\right) = -\Delta t \widehat{\omega} \widehat{f}_i^{(2)}. \tag{A15}$$

Since the collision operator conserves mass and momentum, the *solvability conditions* are assumed to hold at each order and using the definition of $\widehat{f}_i$ (see Equation (A2)), it leads to:

$$\sum_i \widehat{f}_i^{(k)} = +\frac{\Delta t}{2} \sum_i g_i^{(k)} \qquad \text{and} \qquad \sum_i \boldsymbol{\xi}_i \widehat{f}_i^{(k)} = +\frac{\Delta t}{2} \sum_i \boldsymbol{\xi}_i g_i^{(k)} \qquad \forall k \geq 1. \tag{A16}$$

Therefore, using Equation (A14) and the passage formula, the moments associated to 1, $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_i \otimes \boldsymbol{\xi}_i$ are:

$$\partial_t^{(1)}(\rho) + \partial_{\boldsymbol{x}}^{(1)} \cdot \left(\rho \boldsymbol{v}^{(eq)}\right) = -\frac{\Delta t^2 \widehat{\omega}}{2} \sum_i g_i^{(1)} - \left(1 - \frac{\widehat{\omega}\Delta t}{2}\right) \Delta t \sum_i g_i^{(1)}, \tag{A17}$$

$$\partial_t^{(1)}\left(\rho \boldsymbol{v}^{(eq)}\right) + \partial_{\boldsymbol{x}}^{(1)} \cdot \left(\rho \boldsymbol{v}^{(eq)} \otimes \boldsymbol{v}^{(eq)} + p\right) = -\frac{\Delta t}{2} \sum_i \boldsymbol{\xi}_i g_i^{(1)}, \tag{A18}$$

$$\partial_t^{(1)}\left(\rho \boldsymbol{v}^{(eq)} \otimes \boldsymbol{v}^{(eq)} + p\right) + \partial_{\boldsymbol{x}}^{(1)} \cdot \left(\rho \boldsymbol{v}^{(eq)} \otimes \boldsymbol{v}^{(eq)} \otimes \boldsymbol{v}^{(eq)} + \mathcal{P}_3\left(\boldsymbol{v}^{(eq)} \otimes p\right)\right) = -\frac{2\Delta t \widehat{\omega}}{2 - \widehat{\omega}\Delta t} \Pi^{(1)} - \frac{\Delta t}{2} \sum_i \boldsymbol{\xi}_i \otimes \boldsymbol{\xi}_i g_i^{(1)}, \tag{A19}$$

where $\boldsymbol{v}^{(eq)}$ is the equilibrium speed, i.e., the speed parameter in the equilibrium distribution function. Similarly, using Equation (A15) the moments yield by the second perturbation order is:

$$\partial_t^{(2)}(\rho) + 0 = 0, \tag{A20}$$

$$\partial_t^{(2)}\left(\rho \boldsymbol{v}^{(eq)}\right) + \partial_{\boldsymbol{x}}^{(1)} \cdot \left(\Pi^{(1)} + 0\right) = 0. \tag{A21}$$

In order to determine $\Pi^{(1)}$, let us compute all the terms in Equation (A19). The first term, using the general relation

$$\partial_*(abc) = \partial_*(bc)a + \partial_*(ac)b - \partial_*(c)ab, \tag{A22}$$

and $p = \rho {c_s}^2 I$, becomes:

$$\partial_t^{(1)}\left(\rho \boldsymbol{v}^{(eq)} \otimes \boldsymbol{v}^{(eq)} + p\right) = \partial_t^{(1)}\left(\rho \boldsymbol{v}^{(eq)}\right) \otimes \boldsymbol{v}^{(eq)} + \boldsymbol{v}^{(eq)} \otimes \partial_t^{(1)}\left(\rho \boldsymbol{v}^{(eq)}\right) - \boldsymbol{v}^{(eq)} \otimes \boldsymbol{v}^{(eq)} \partial_t^{(1)}(\rho) + {c_s}^2 I \partial_t^{(1)}(\rho). \tag{A23}$$

Then, reinjecting the temporal derivatives from Equations (A17) and (A18)

$$\begin{aligned}
&\partial_t^{(1)}\left(\rho \boldsymbol{v}^{(eq)} \otimes \boldsymbol{v}^{(eq)} + p\right) = \\
&\boldsymbol{v}^{(eq)} \otimes \left[-\partial_{\boldsymbol{x}}^{(1)} \cdot \left(\underline{\rho \boldsymbol{v}^{(eq)} \otimes \boldsymbol{v}^{(eq)}} + p\right) - \frac{\Delta t}{2} \sum_i \boldsymbol{\xi}_i g_i^{(1)}\right] + \left[-\partial_{\boldsymbol{x}}^{(1)} \cdot \left(\underline{\rho \boldsymbol{v}^{(eq)} \otimes \boldsymbol{v}^{(eq)}} + p\right) - \frac{\Delta t}{2} \sum_i \boldsymbol{\xi}_i g_i^{(1)}\right] \otimes \boldsymbol{v}^{(eq)} \\
&- \boldsymbol{v}^{(eq)} \otimes \boldsymbol{v}^{(eq)} \left[-\partial_{\boldsymbol{x}}^{(1)} \cdot \left(\underline{\rho \boldsymbol{v}^{(eq)}}\right) - \frac{\Delta t}{2} \sum_i g_i^{(1)}\right] + {c_s}^2 I \left[-\partial_{\boldsymbol{x}}^{(1)} \cdot \left(\rho \boldsymbol{v}^{(eq)}\right) - \frac{\Delta t}{2} \sum_i g_i^{(1)}\right],
\end{aligned} \tag{A24}$$

and using backward the Equation (A22) for the underlined terms, we get:

$$\begin{aligned}\partial_t^{(1)}\left(\rho v^{(eq)}\otimes v^{(eq)}+p\right)=&-\partial_x^{(1)}\cdot\left(\rho v^{(eq)}\otimes v^{(eq)}\otimes v^{(eq)}\right)-{c_s}^2\left(v^{(eq)}\otimes\partial_x^{(1)}(\rho)+\partial_x^{(1)}(\rho)\otimes v^{(eq)}+I\partial_x^{(1)}\cdot\left(\rho v^{(eq)}\right)\right)\\&-v^{(eq)}\otimes\left(\frac{\Delta t}{2}\sum_i\xi_i{g_i}^{(1)}\right)-\left(\frac{\Delta t}{2}\sum_i\xi_i{g_i}^{(1)}\right)\otimes v^{(eq)}-\frac{\Delta t}{2}\sum_i{g_i}^{(1)}\left({c_s}^2I-v^{(eq)}\otimes v^{(eq)}\right).\end{aligned}\tag{A25}$$

The second term in Equation (A19) is straight-forward and yields:

$$\begin{aligned}\partial_x^{(1)}\cdot\left(\rho v^{(eq)}\otimes v^{(eq)}\otimes v^{(eq)}+\mathcal{P}_3\left(v^{(eq)}\otimes p\right)\right)=&\,\partial_x^{(1)}\cdot\left(\rho v^{(eq)}\otimes v^{(eq)}\otimes v^{(eq)}\right)+{c_s}^2I\partial_x^{(1)}\cdot\left(\rho v^{(eq)}\right)\\&+{c_s}^2\left[\rho\partial_x^{(1)}\left(v^{(eq)}\right)+\rho\partial_x^{(1)}\left(v^{(eq)}\right)^T+v^{(eq)}\otimes\partial_x^{(1)}(\rho)+\partial_x^{(1)}(\rho)\otimes v^{(eq)}\right].\end{aligned}\tag{A26}$$

Then, by summing Equations (A25) and (A26) into Equation (A19), the expression of the first order non-equilibrium stress tensor is given by:

$$\begin{aligned}\Pi^{(1)}=&-\frac{(2-\widehat{\omega}\Delta t)}{4\widehat{\omega}}\sum_i\xi_i\otimes\xi_i{g_i}^{(1)}+\frac{(2-\widehat{\omega}\Delta t)}{4\widehat{\omega}}\left[v^{(eq)}\otimes\left(\sum_i\xi_i{g_i}^{(1)}\right)+\left(\sum_i\xi_i{g_i}^{(1)}\right)\otimes v^{(eq)}\right]\\&+\frac{(2-\widehat{\omega}\Delta t)}{4\widehat{\omega}}\sum_i{g_i}^{(1)}\left(v^{(eq)}\otimes v^{(eq)}-{c_s}^2I\right)-\frac{(2-\widehat{\omega}\Delta t)}{2\widehat{\omega}\Delta t}{c_s}^2\rho\left[\partial_x^{(1)}\left(v^{(eq)}\right)+\partial_x^{(1)}\left(v^{(eq)}\right)^T\right].\end{aligned}\tag{A27}$$

Finally, the global discrete macroscopic equations solved numerically by the LBM are obtained by recombining the perturbation orders. Thus, considering Equations (A17) and (A20)

$$\left(\epsilon^1\partial_t^{(1)}+\epsilon^2\partial_t^{(2)}\right)\rho+\epsilon^1\partial_x^{(1)}\left(\rho v^{(eq)}\right)=-\frac{\Delta t}{2}\sum_i{g_i}^{(1)},\tag{A28}$$

and by summing Equations (A18) and (A21)

$$\begin{aligned}\left(\epsilon^1\partial_t^{(1)}+\epsilon^2\partial_t^{(2)}\right)\left(\rho v^{(eq)}\right)+&\epsilon^1\partial_x^{(1)}\left(\rho v^{(eq)}\otimes v^{(eq)}+p+\Pi^{(1)}\right)\\&=-\frac{\Delta t}{2}\sum_i\xi_i{g_i}^{(1)}.\end{aligned}\tag{A29}$$

Taking the macroscopic moment equations (Equations (A28) and (A29)) and reversing the perturbation expansion of the derivatives Equations (A6) and (A7), we obtain the following macroscopic equation solved:

$$\partial_t\rho+\partial_x\left(\rho v^{(eq)}\right)=-\frac{\Delta t}{2}\sum_i{g_i}^{(1)},\tag{A30}$$

$$\partial_t\rho v^{(eq)}+\partial_x\left(\rho v^{(eq)}\otimes v^{(eq)}+p+\Pi^{(1)}\right)=-\frac{\Delta t}{2}\sum_i\xi_i{g_i}^{(1)}.\tag{A31}$$

We keep the numerical evaluation of the external force is here as stress divergence given by Equation (13). Then, considering this special case for LBMS ($\sum{g_i}^{(1)}=0$, $\sum\xi_i{g_i}^{(1)}=\nabla_x\cdot(\sigma)$, $\sum\xi_i\otimes\xi_i{g_i}^{(1)}=\left(v^{(eq)}\otimes\nabla_x\cdot(\sigma)+\nabla_x\cdot(\sigma)\otimes v^{(eq)}\right)$), the macroscopic equation solved becomes:

$$\partial_t\rho+\partial_x\left(\rho v^{(eq)}\right)=0,\tag{A32}$$

$$\begin{aligned}\partial_t\rho v^{(eq)}+&\partial_x\left(\rho v^{(eq)}\otimes v^{(eq)}+p-\frac{(2-\widehat{\omega}\Delta t)}{2\widehat{\omega}\Delta t}{c_s}^2\rho\left[\partial_x^{(1)}\left(v^{(eq)}\right)+\partial_x^{(1)}\left(v^{(eq)}\right)^T\right]\right)\\&=-\nabla_x\cdot(\sigma).\end{aligned}\tag{A33}$$

The last equation is nothing else than the undefined equation of the motion. Furthermore, using the static equilibrium ($\boldsymbol{v}^{(eq)} = 0$) it simplifies to:

$$\partial_t \rho = 0, \tag{A34}$$

$$\nabla_{\boldsymbol{x}} \cdot (\sigma) = 0. \tag{A35}$$

Appendix B. Tensors Details

The 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. The displacement tensor field has the following form:

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}[0,0] = \boldsymbol{u}_{0,0} = \begin{bmatrix} \boldsymbol{u}_x \\ \boldsymbol{u}_y \end{bmatrix}_{0,0} & \cdots & \boldsymbol{u}_{0,N_y-1} = \begin{bmatrix} \boldsymbol{u}_x \\ \boldsymbol{u}_y \end{bmatrix}_{0,N_y-1} \\ \vdots & \ddots & \vdots \\ \boldsymbol{u}_{N_x-1,0} = \begin{bmatrix} \boldsymbol{u}_x \\ \boldsymbol{u}_y \end{bmatrix}_{N_x-1,0} & \cdots & \boldsymbol{u}_{N_x-1,N_y-1} = \begin{bmatrix} \boldsymbol{u}_x \\ \boldsymbol{u}_y \end{bmatrix}_{N_x-1,N_y-1} \end{bmatrix}. \tag{A36}$$

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

$$\mathbf{E} = \begin{bmatrix} \mathbf{E}[0,0] = \varepsilon_{0,0} = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix}_{0,0} & \cdots & \varepsilon_{0,N_y-1} = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix}_{0,N_y-1} \\ \vdots & \ddots & \vdots \\ \varepsilon_{N_x-1,0} = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix}_{N_x-1,0} & \cdots & \varepsilon_{N_x-1,N_y-1} = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix}_{N_x-1,N_y-1} \end{bmatrix}. \tag{A37}$$

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

$$\mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}[0,0] = \sigma_{0,0} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix}_{0,0} & \cdots & \sigma_{0,N_y-1} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix}_{0,N_y-1} \\ \vdots & \ddots & \vdots \\ \sigma_{N_x-1,0} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix}_{N_x-1,0} & \cdots & \sigma_{N_x-1,N_y-1} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix}_{N_x-1,N_y-1} \end{bmatrix}. \tag{A38}$$

Appendix C. Curve Plots

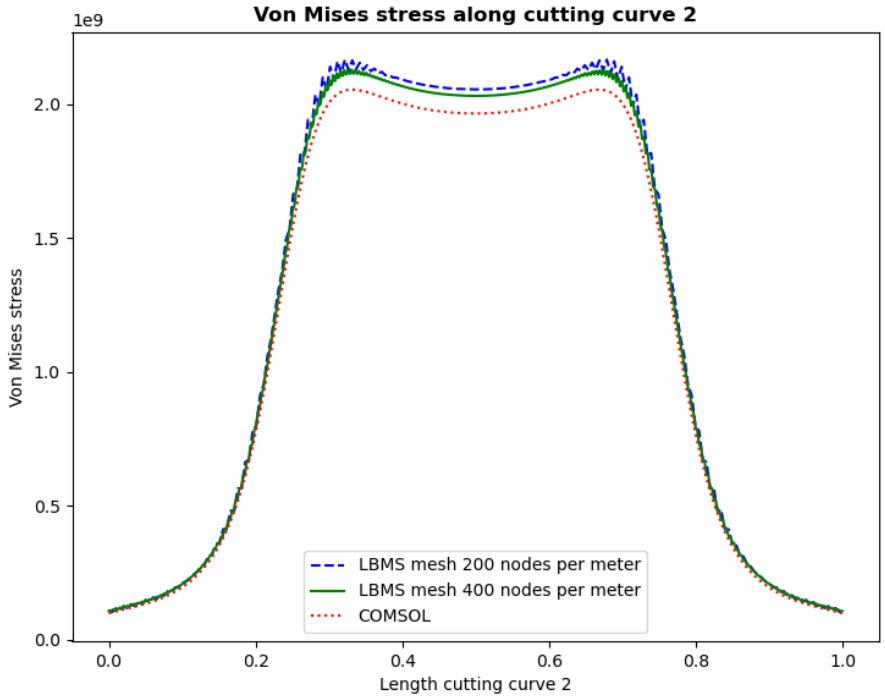


Figure A1. Von Mises stress along cutting curve C_2 for case $E = 200$ GPa, $\nu = 0.15$.

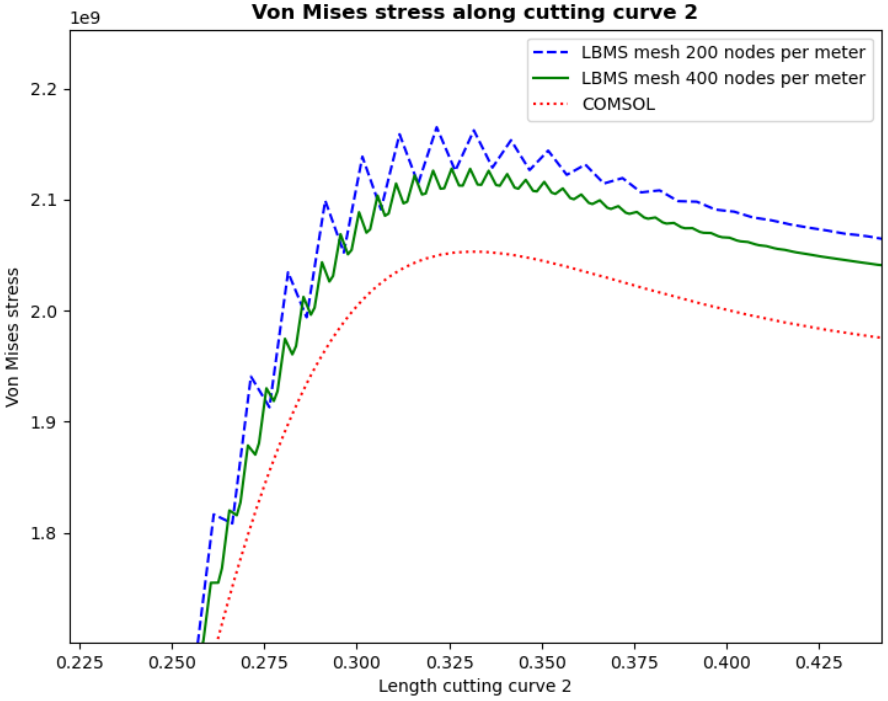


Figure A2. Detailed view of Von Mises stress along cutting curve C_2 for case $E = 200$ GPa, $\nu = 0.15$.

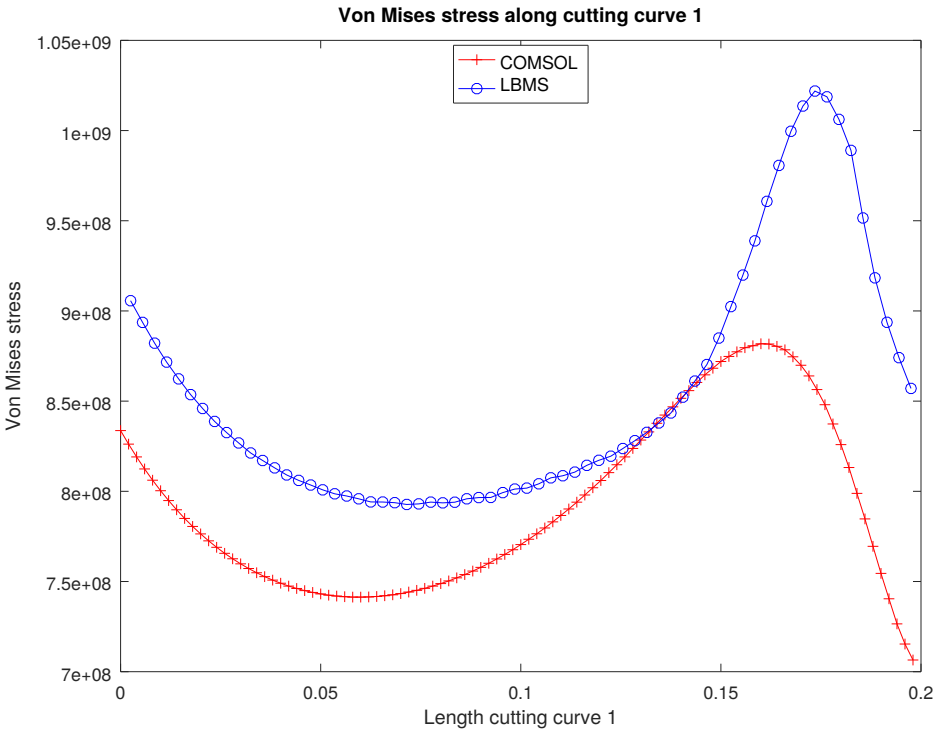


Figure A3. Von Mises stress along cutting curve C_1 for case E = 200 GPa, ν = 0.15.

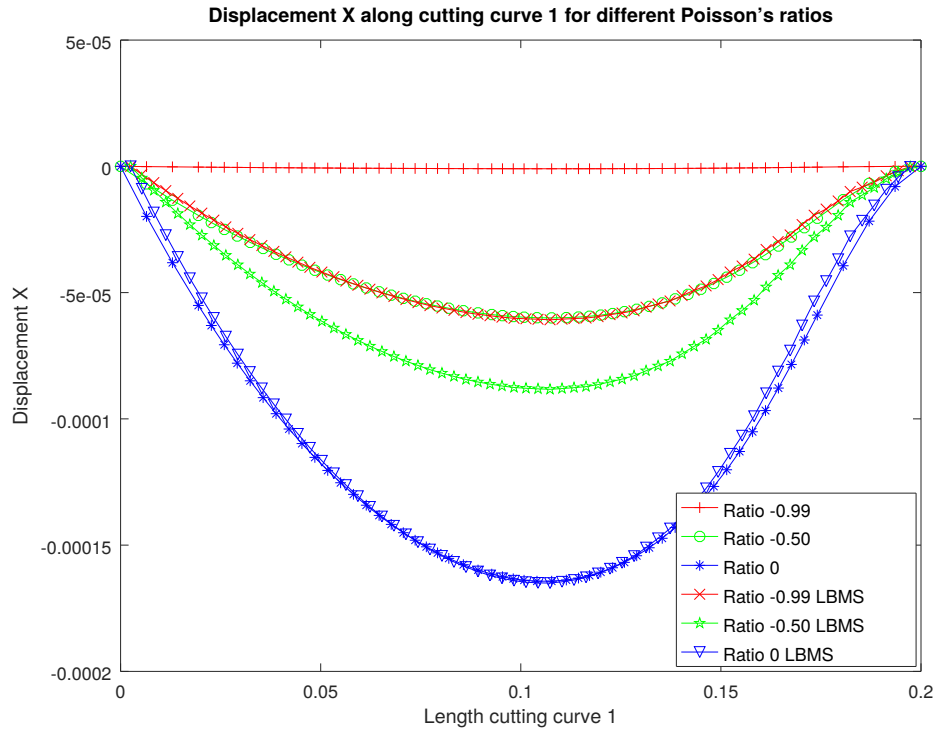


Figure A4. First displacement components along cutting curve C_1 for negative Poisson's ratios.

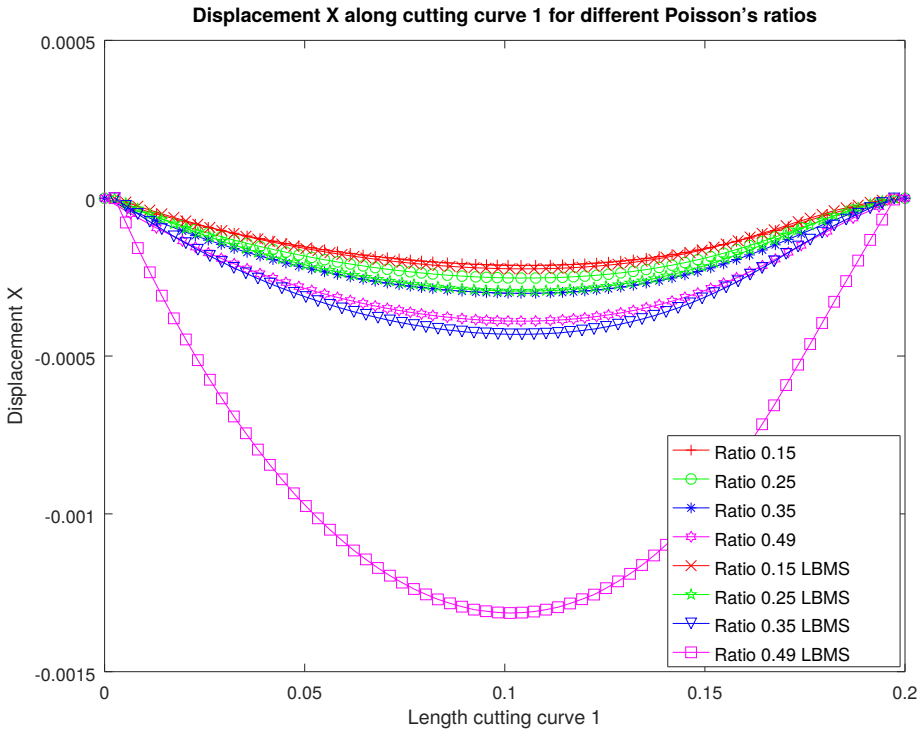


Figure A5. First displacement components along cutting curve C_1 for positive Poisson's ratios.

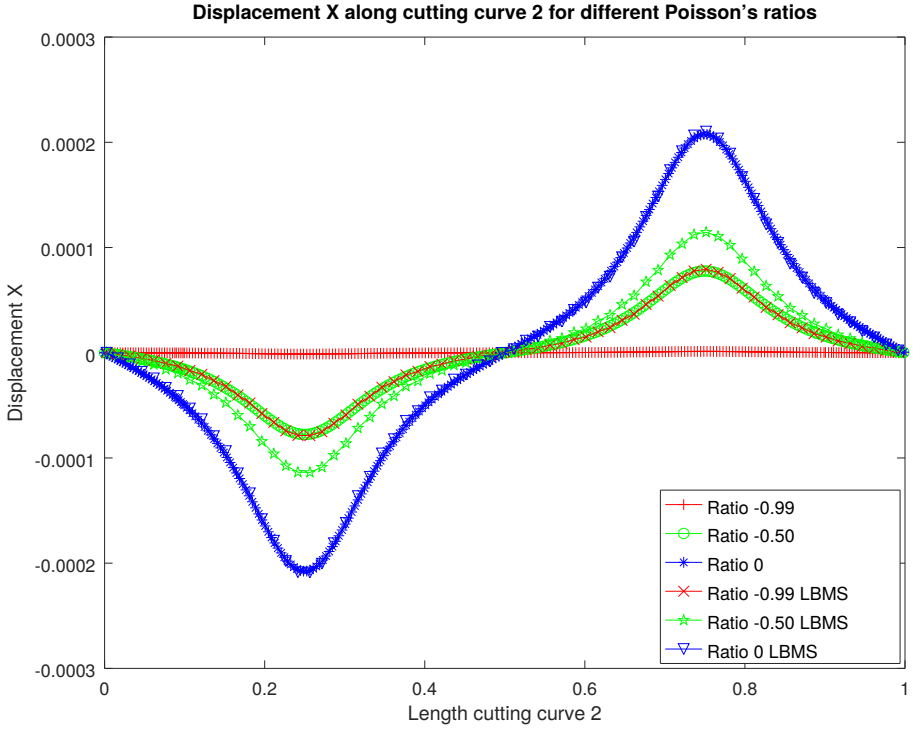


Figure A6. First displacement components along cutting curve C_2 for negative Poisson's ratios.

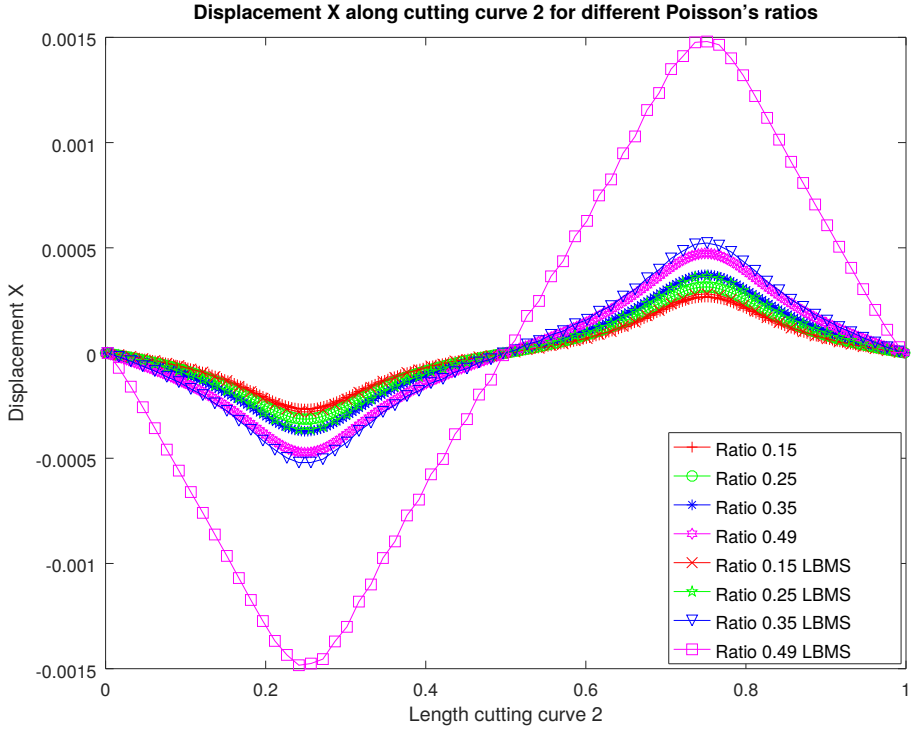


Figure A7. First displacement components along cutting curve C_2 for positive Poisson's ratios.

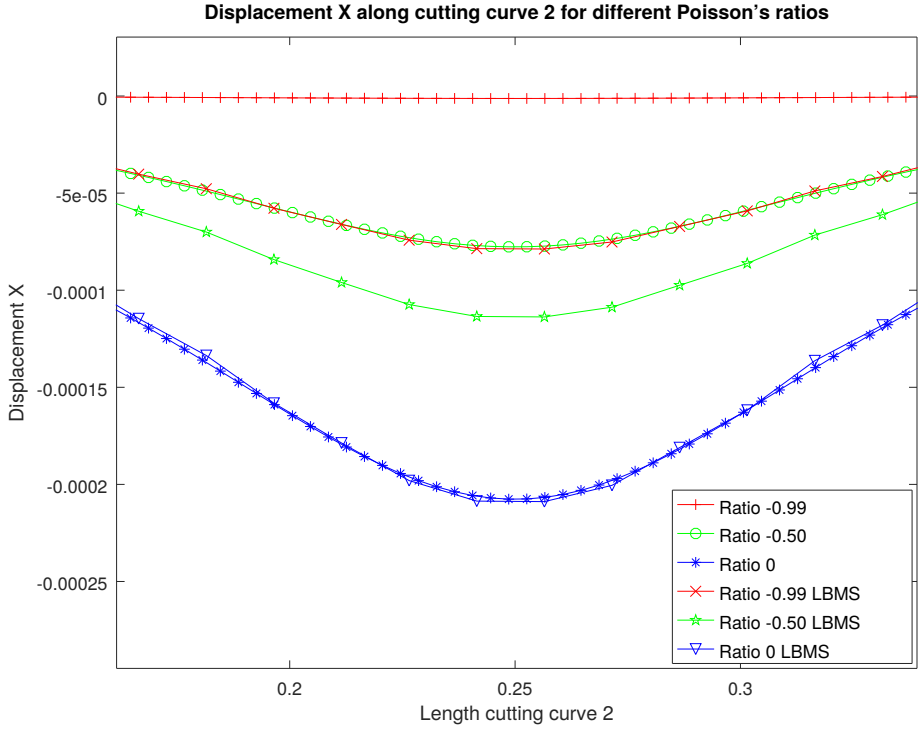


Figure A8. Detailed view of first displacement components along cutting curve C_2 for negative Poisson's ratios.

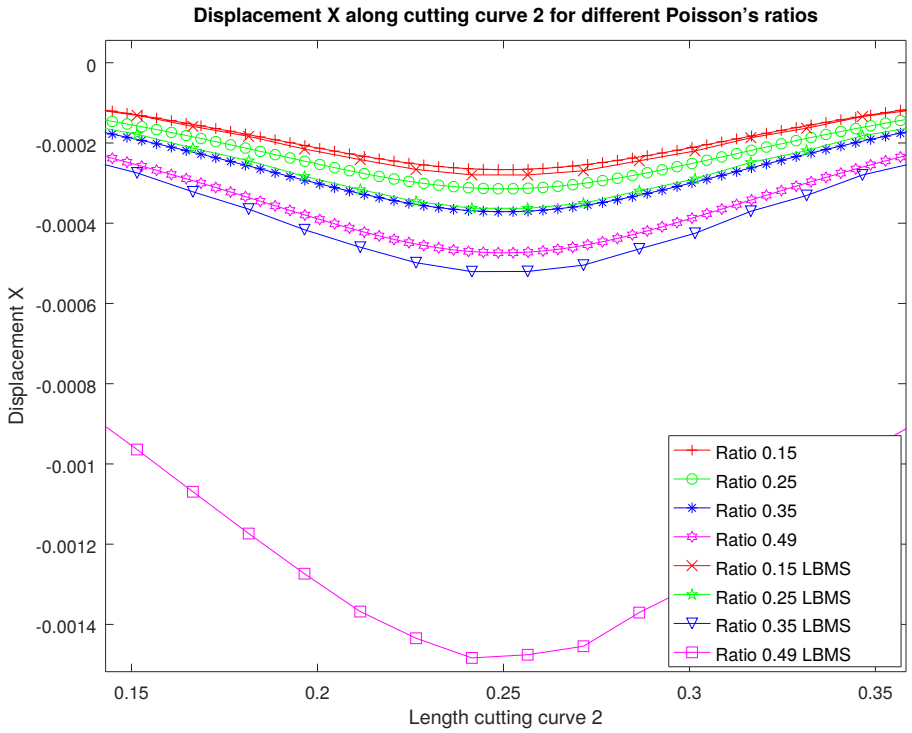


Figure A9. Detailed view of first displacement components along cutting curve C_2 for positive Poisson's ratios.

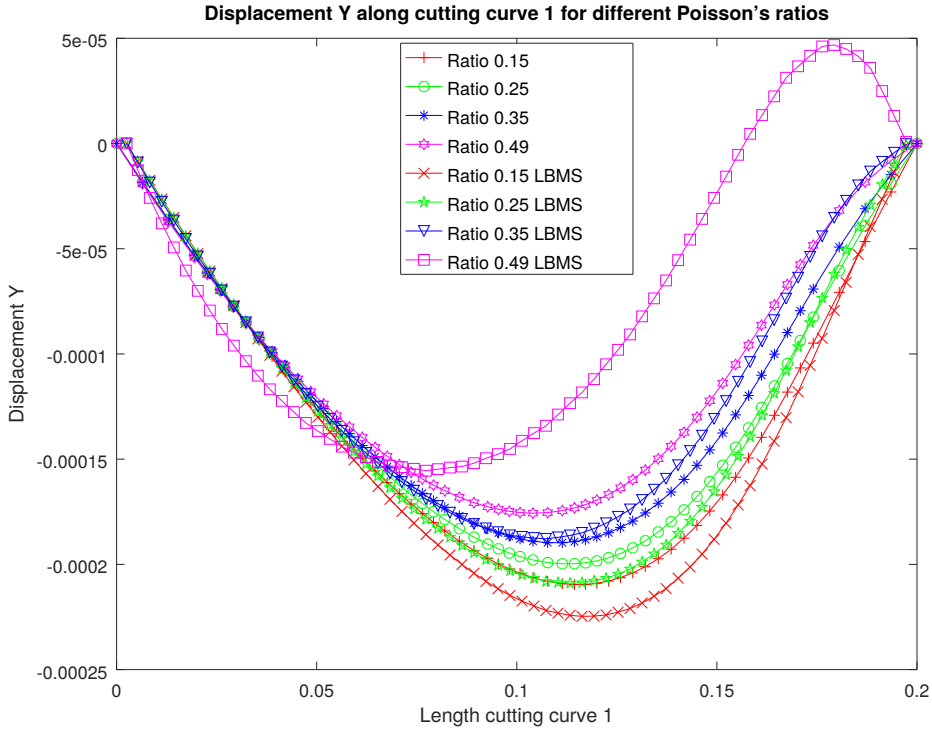


Figure A10. Second displacement component along cutting curve C_1 for positive Poisson's ratios.

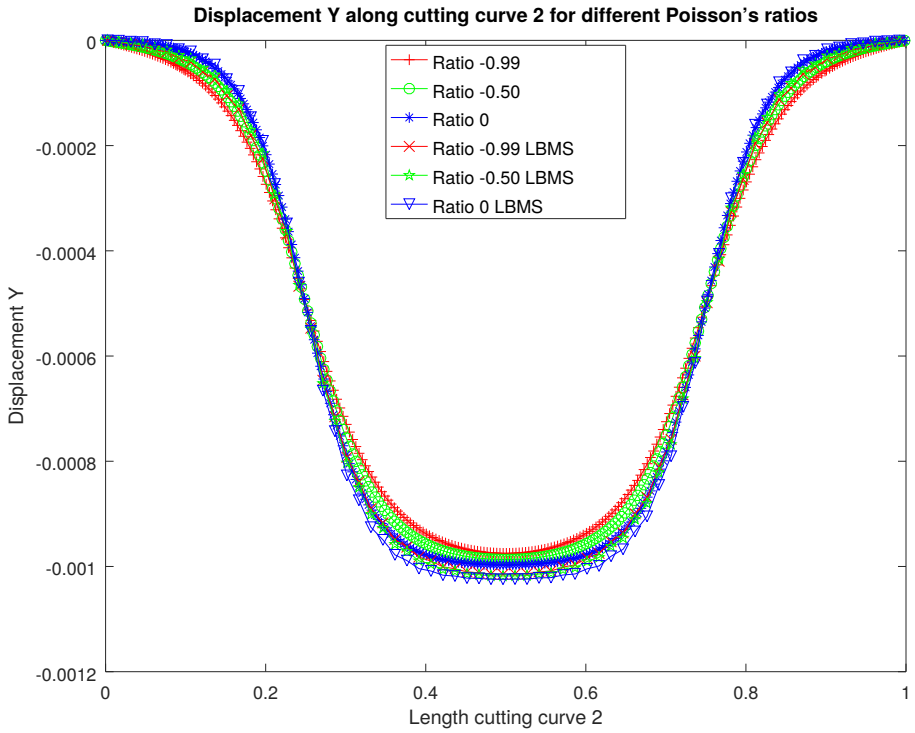


Figure A11. Second displacement components along cutting curve C_2 for negative Poisson's ratios.

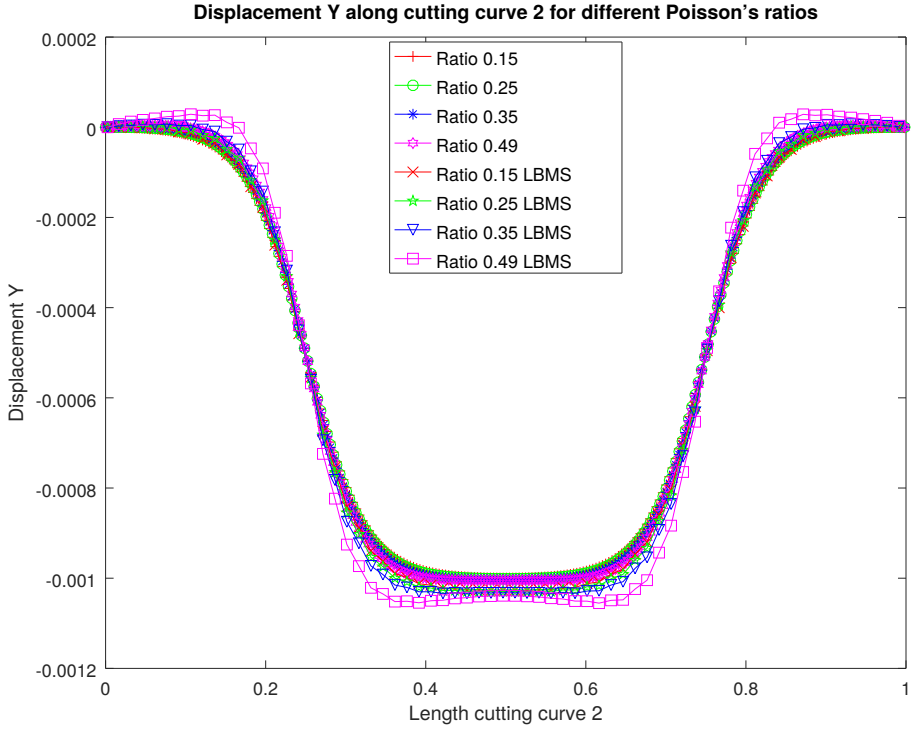


Figure A12. Second displacement components along cutting curve C_2 for positive Poisson's ratios.

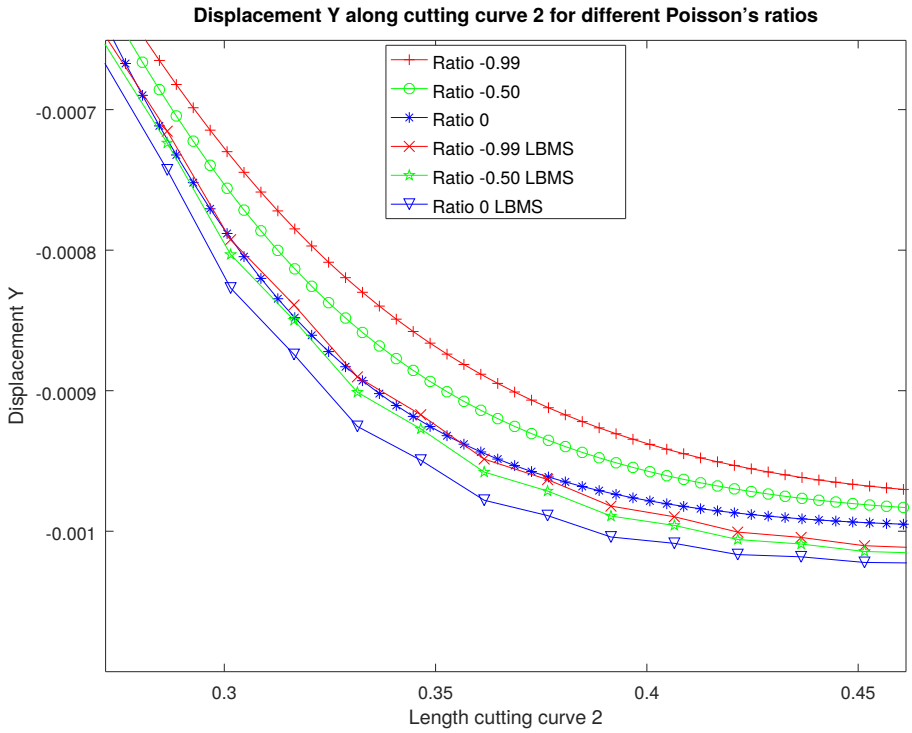


Figure A13. Detailed view of second displacement components along cutting curve C_2 for negative Poisson's ratios.

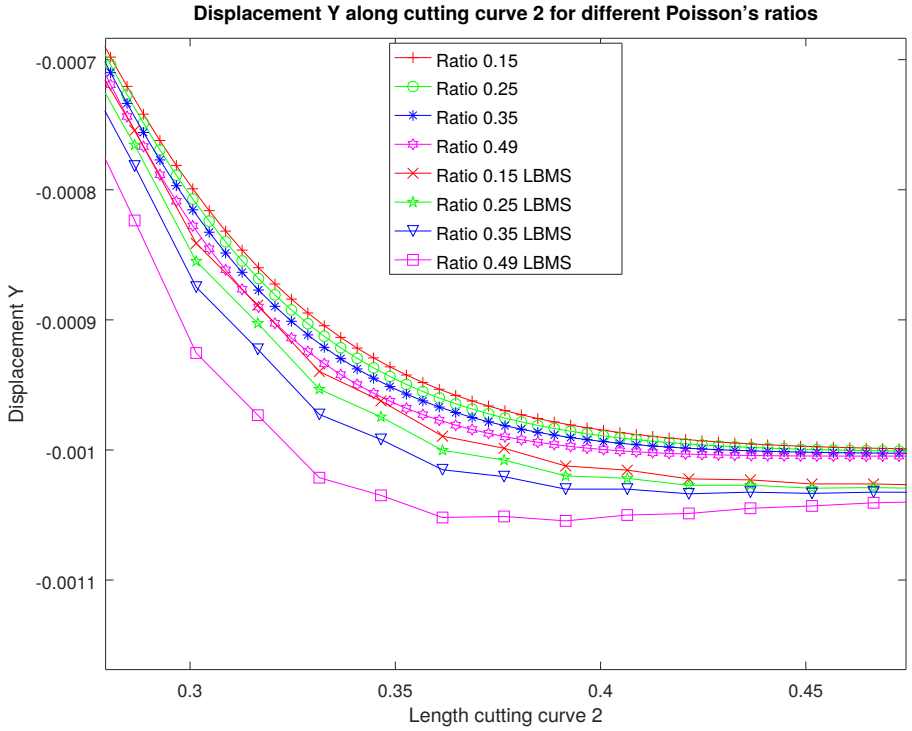


Figure A14. Detailed view of second displacement components along cutting curve C_2 for positive Poisson's ratios.

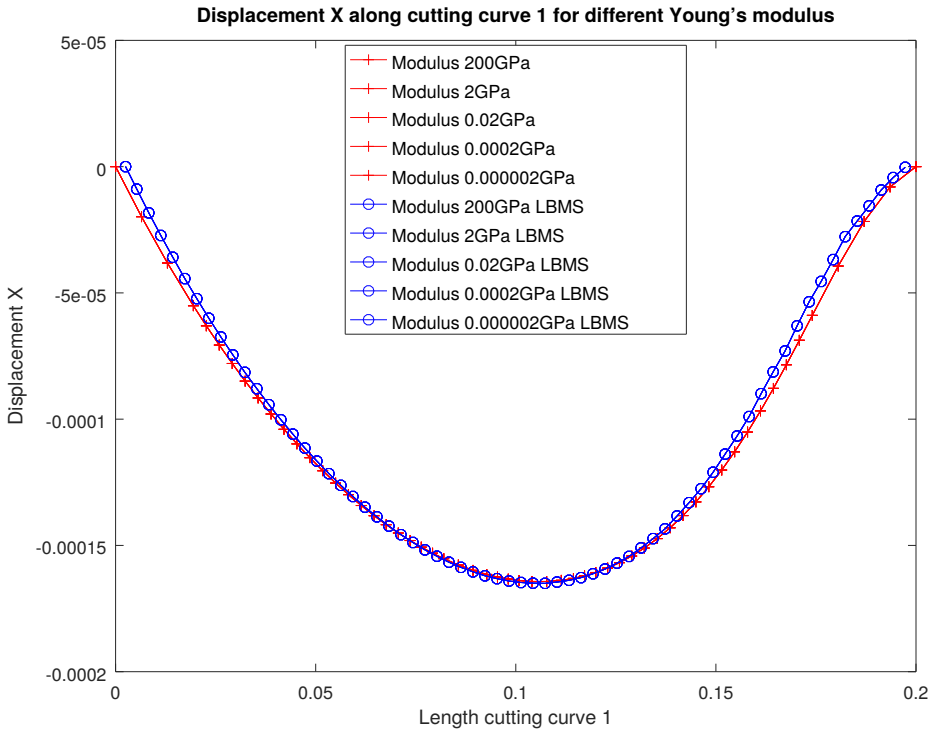


Figure A15. First displacement components along cutting curve C_1 for different Young's modulus.

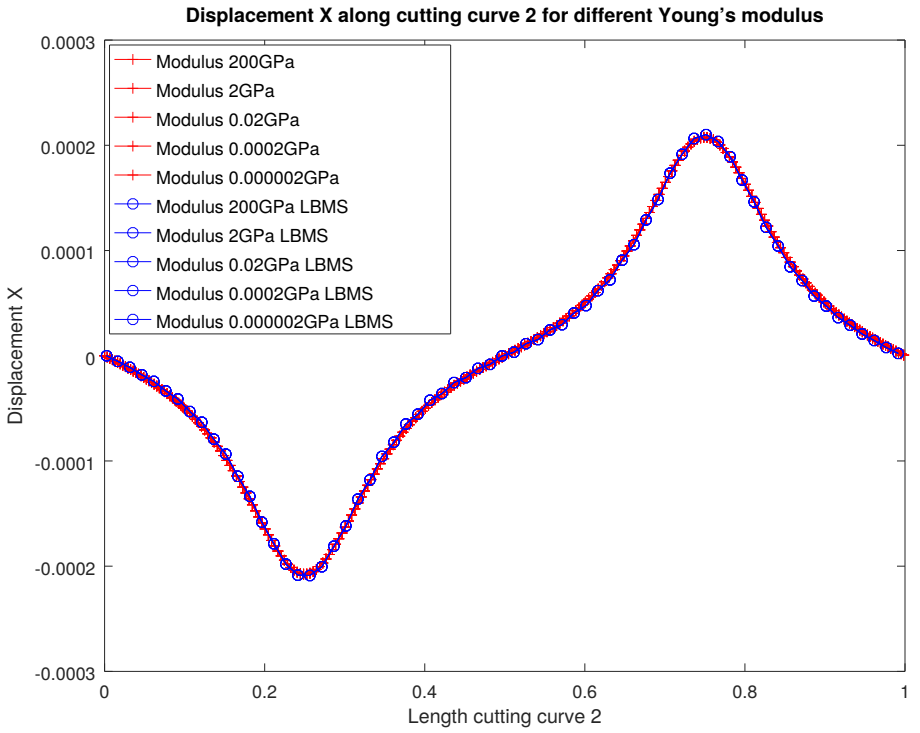


Figure A16. First displacement components along cutting curve C_2 for different Young's modulus.

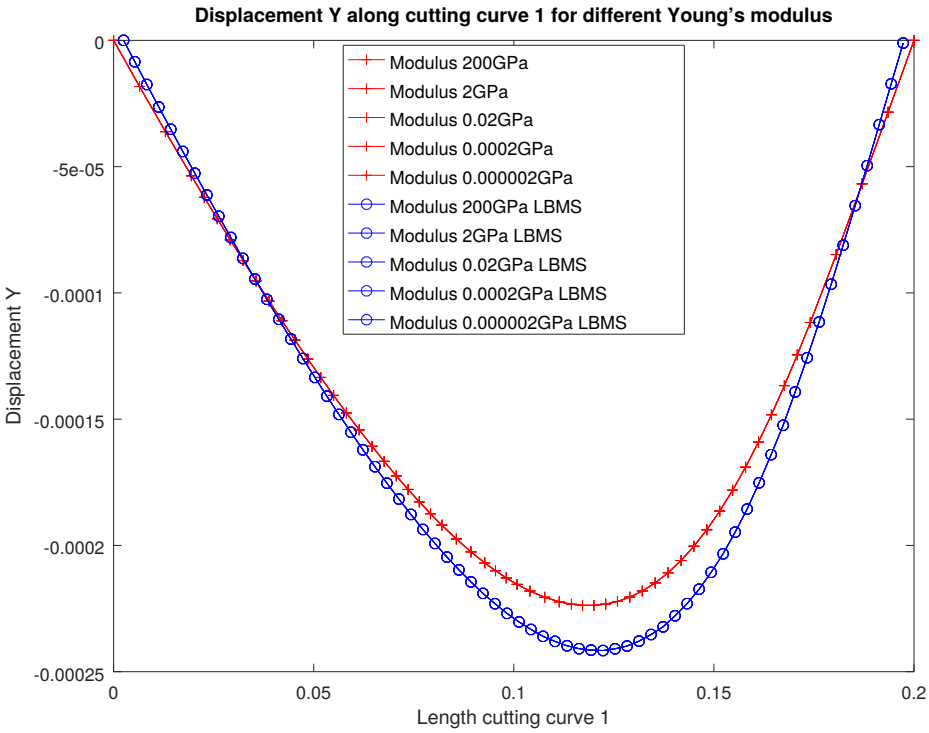


Figure A17. Second displacement components along cutting curve C_1 for different Young's modulus.

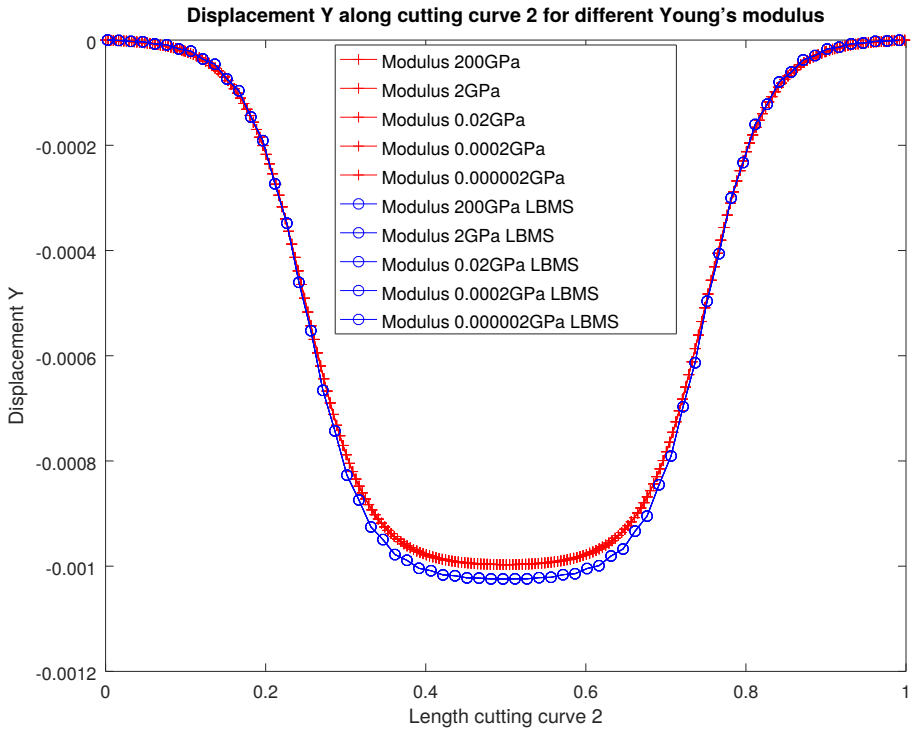


Figure A18. Second displacement components along cutting curve C_2 for different Young's modulus.

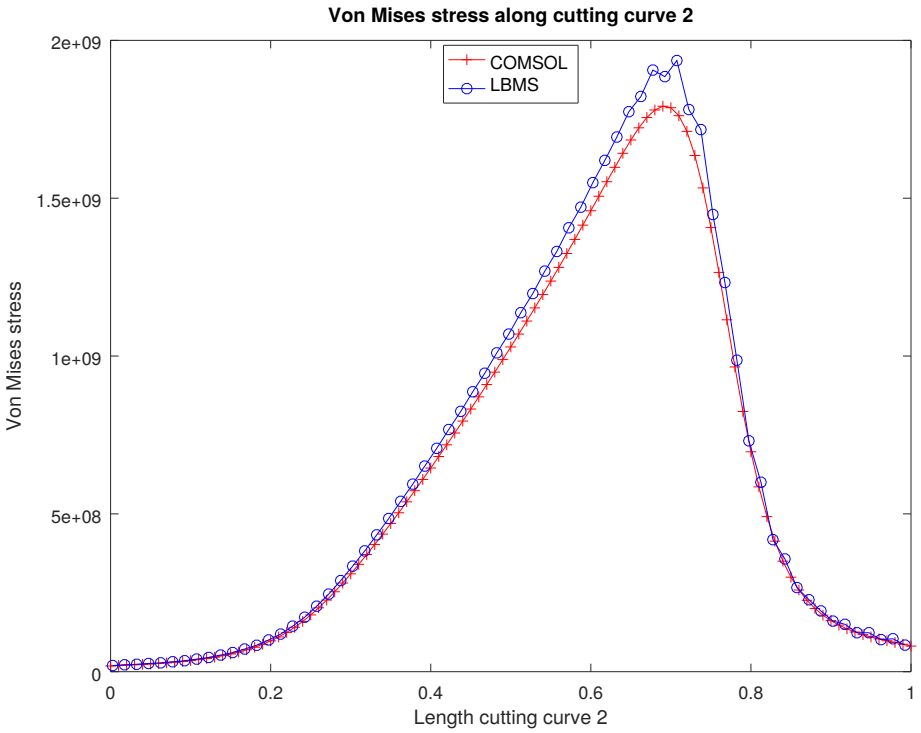


Figure A19. Von Mises stress along cutting curve C_2 for case E = 200 GPa, ν = 0.15 with triangular loading.

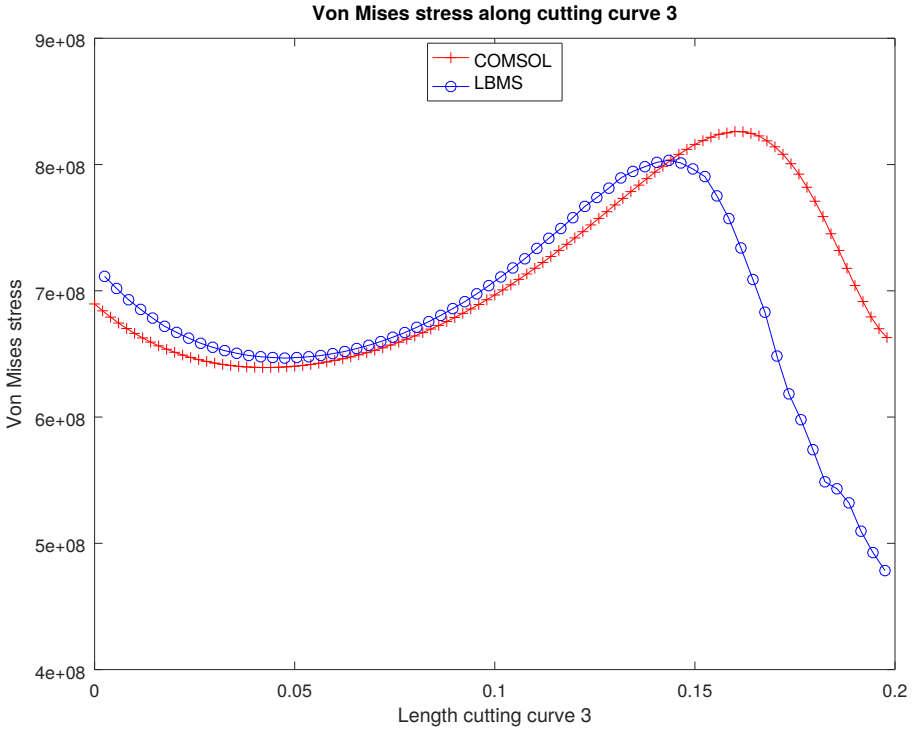


Figure A20. Von Mises stress along cutting curve C_3 for case E = 200 GPa, ν = 0.15 with triangular loading.

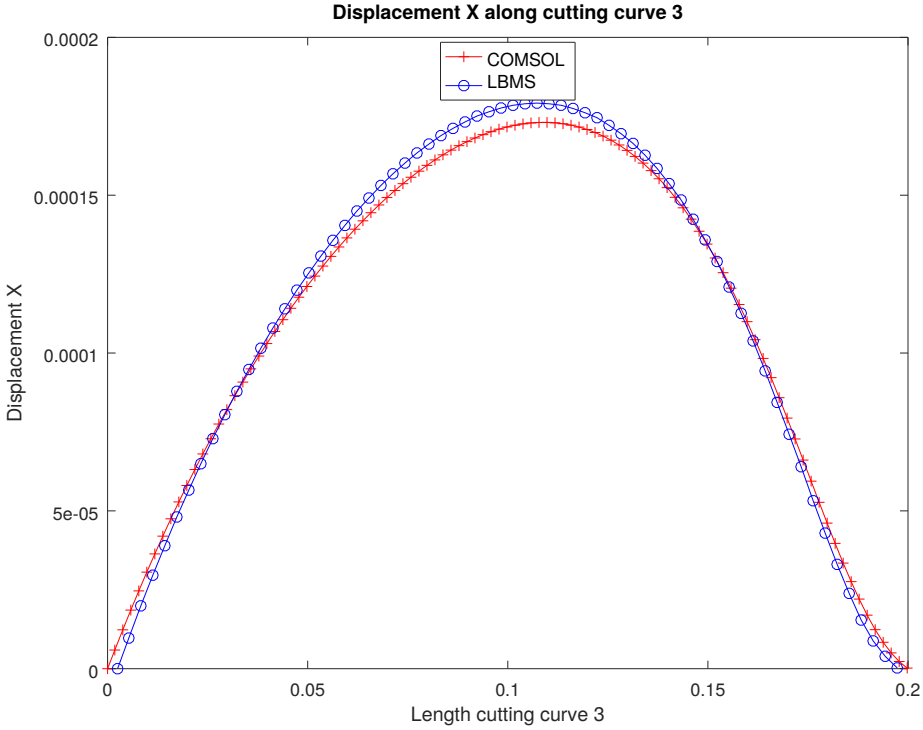


Figure A21. First displacement components along cutting curve C_3 for case E = 200 GPa, ν = 0.15 with triangular loading.

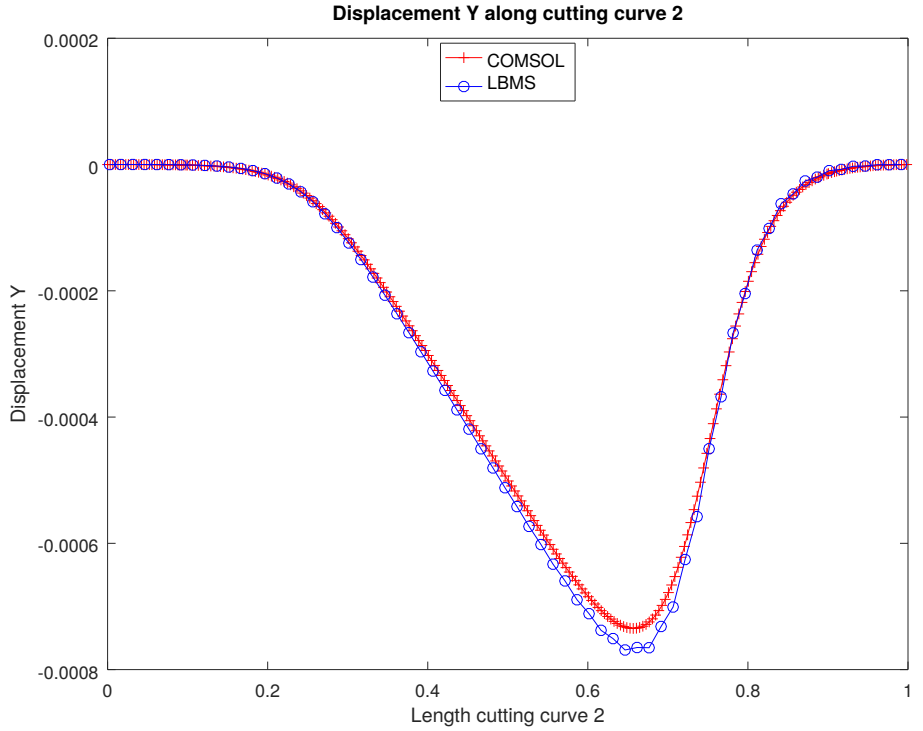


Figure A22. Second displacement components along cutting curve C_2 for case E = 200 GPa, ν = 0.15 with triangular loading.

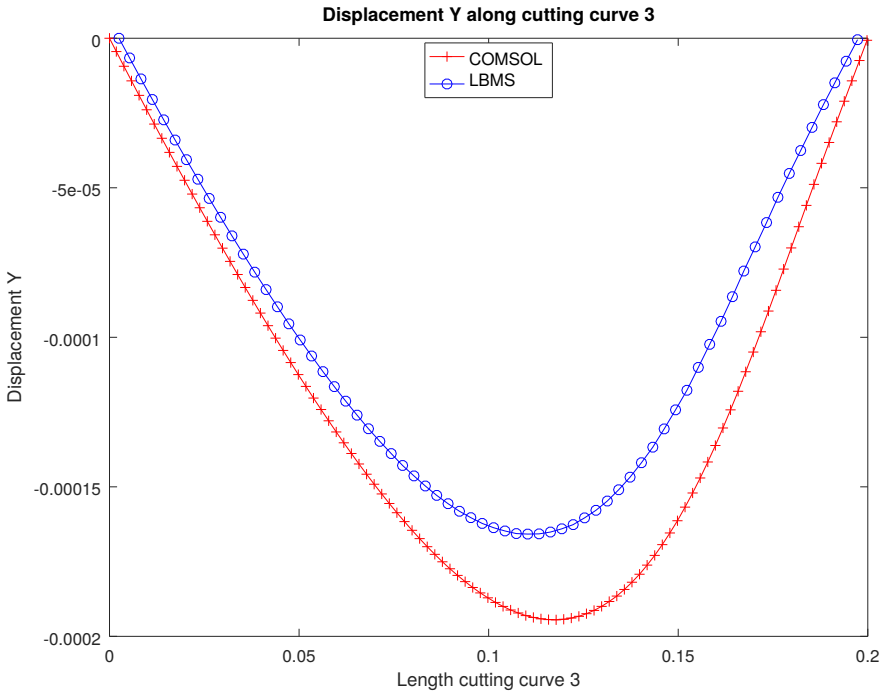


Figure A23. Second displacement components along cutting curve C_3 for case E = 200 GPa, ν = 0.15 with triangular loading.

References

1. Frisch, U.; Hasslacher, B.; Pomeau, Y. Lattice-gas automata for the Navier-Stokes equation. *Phys. Rev. Lett.* **1986**, *56*, 1505. [CrossRef] [PubMed]
2. Bhatnagar, P.L.; Gross, E.P.; Krook, M. A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems. *Phys. Rev.* **1954**, *94*, 511–525. [CrossRef]
3. McNamara, G.R.; Zanetti, G. Use of the Boltzmann Equation to Simulate Lattice-Gas Automata. *Phys. Rev. Lett.* **1988**, *61*, 2332–2335. [CrossRef] [PubMed]
4. Succi, S.; Benzi, R.; Higuera, F. The Lattice Boltzmann Equation: A New Tool for Computational Fluid-Dynamics. *Phys. D Nonlinear Phenom.* **1991**, *47*, 219–230. [CrossRef]
5. Karlin, I.V.; Gorban, A.N.; Succi, S.; Boffi, V. Maximum Entropy Principle for Lattice Kinetic Equations. *Phys. Rev. Lett.* **1998**, *81*, 6–9. [CrossRef]
6. Karlin, I.V.; Bösch, F.; Chikatamarla, S.S. Gibbs' Principle for the Lattice-Kinetic Theory of Fluid Dynamics. *Phys. Rev. E* **2014**, *90*, 031302. [CrossRef]
7. Dellar, P.J. Incompressible limits of lattice Boltzmann equations using multiple relaxation times. *J. Comput. Phys.* **2003**, *190*, 351–370. [CrossRef]
8. d'Humières, D. Generalized Lattice-Boltzmann Equations. In *Rarefied Rarefied Gas Dynamics: Theory and Simulations*; Progress in Astronautics and Aeronautics; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 1992; Volume 159, pp. 450–458. [CrossRef]
9. Gunstensen, A.K.; Rothman, D.H.; Zaleski, S.; Zanetti, G. Lattice Boltzmann Model of Immiscible Fluids. *Phys. Rev. A* **1991**, *43*, 4320–4327. [CrossRef]
10. Shan, X.; Chen, H. Lattice Boltzmann Model for Simulating Flows with Multiple Phases and Components. *Phys. Rev. E* **1993**, *47*, 1815–1819. [CrossRef]
11. Swift, M.R.; Osborn, W.R.; Yeomans, J.M. Lattice Boltzmann Simulation of Nonideal Fluids. *Phys. Rev. Lett.* **1995**, *75*, 830. [CrossRef]
12. He, X.; Chen, S.; Zhang, R. A lattice Boltzmann scheme for incompressible multiphase flow and its application in simulation of Rayleigh–Taylor instability. *J. Comput. Phys.* **1999**, *152*, 642–663. [CrossRef]
13. Lee, T.; Lin, C.L. A stable discretization of the lattice Boltzmann equation for simulation of incompressible two-phase flows at high density ratio. *J. Comput. Phys.* **2005**, *206*, 16–47. [CrossRef]
14. Huang, H.; Sukop, M.; Lu, X. *Multiphase Lattice Boltzmann Methods: Theory and Application*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
15. Chiappini, D.; Xue, X.; Falcucci, G.; Sbragaglia, M. Ligament Break-up Simulation through Pseudo-Potential Lattice Boltzmann Method. In Proceedings of the International Conference of Numerical Analysis and Applied Mathematics (ICNAAM 2017), Thessaloniki, Greece, 25–30 September 2017; p. 420003. [CrossRef]

16. Higuera, F.; Succi, S.; Benzi, R. Lattice gas dynamics with enhanced collisions. *Europhys. Lett.* **1989**, *9*, 345. [CrossRef]
17. Benzi, R.; Succi, S.; Vergassola, M. The lattice Boltzmann equation: Theory and applications. *Phys. Rep.* **1992**, *222*, 145–197. [CrossRef]
18. Marconi, S.; Chopard, B. A Lattice Boltzmann Model for a Solid Body. *Int. J. Mod. Phys. B* **2003**, *17*, 153–156. [CrossRef]
19. Xiao, S. A lattice Boltzmann method for shock wave propagation in solids. *Commun. Numer. Methods Eng.* **2007**, *23*, 71–84. [CrossRef]
20. O'Brien, G.S.; Nissen-Meyer, T.; Bean, C. A lattice Boltzmann method for elastic wave propagation in a poisson solid. *Bull. Seismol. Soc. Am.* **2012**, *102*, 1224–1234. [CrossRef]
21. Escande, M.; Kolluru, P.K.; Cléon, L.M.; Sagaut, P. Lattice Boltzmann Method for wave propagation in elastic solids with a regular lattice: Theoretical analysis and validation. *arXiv* **2020**, arXiv:2009.06404.
22. Yin, X.; Yan, G.; Li, T. Direct simulations of the linear elastic displacements field based on a lattice Boltzmann model. *Int. J. Numer. Methods Eng.* **2016**, *107*, 234–251. [CrossRef]
23. Yan, G.; Li, T.; Yin, X. Lattice Boltzmann model for elastic thin plate with small deflection. *Comput. Math. Appl.* **2012**, *63*, 1305–1318. [CrossRef]
24. Buxton, G.A.; Verberg, R.; Jasnow, D.; Balazs, A.C. Newtonian Fluid Meets an Elastic Solid: Coupling Lattice Boltzmann and Lattice-Spring Models. *Phys. Rev. E* **2005**, *71*, 056707 . [CrossRef] [PubMed]
25. Wu, T.H.; Qi, D. Lattice-Boltzmann Lattice-Spring Simulations of Influence of Deformable Blockages on Blood Fluids in an Elastic Vessel. *Comput. Fluids* **2017**, *155*, 103–111. [CrossRef]
26. Guangwu, Y. A Lattice Boltzmann Equation for Waves. *J. Comput. Phys.* **2000**, *161*, 61–69. [CrossRef]
27. Frantziskonis, G.N. Lattice Boltzmann Method for Multimode Wave Propagation in Viscoelastic Media and in Elastic Solids. *Phys. Rev. E* **2011**, *83*, 066703. [CrossRef] [PubMed]
28. Murthy, J.S.N.; Kolluru, P.K.; Kumaran, V.; Ansumali, S. Lattice Boltzmann Method for Wave Propagation in Elastic Solids. *Commun. Comput. Phys.* **2018**, *18*. [CrossRef]
29. Kamrin, K.; Nave, J.C. An Eulerian approach to the simulation of deformable solids: Application to finite-strain elasticity. *arXiv* **2009**, arXiv:0901.3799.
30. Noël, R.; Ge, F.; Zhang, Y.; Navarro, L.; Courbebaisse, G. Lattice Boltzmann Method for Modelling of Biological Phenomena. In Proceedings of the 2017 25th European Signal Processing Conference (EUSIPCO), Kos, Greece, 28 August–2 September 2017; pp. 2654–2658. [CrossRef]
31. Noël, R.; Navarro, L.; Courbebaisse, G. Lattice Boltzmann Method & Mathematical Morphology. In Proceedings of the GRETSI 2019 XXVIIe Colloque, Lille, France, 26–29 August 2019.
32. PALABOS v2.0r0—Parallel Lattice Boltzmann Solver. 2017. Available online: https://palabos.unige.ch/ (accessed on 1 March 2022).
33. COMSOL Multiphysics 5.5. 2019. Available online: www.comsol.com (accessed on 1 March 2022).
34. Boltzmann, L. Further Studies on the Thermal Equilibrium of Gas Molecules. In *History of Modern Physical Sciences*; World Scientific Publishing: Singapore, 2003; Volume 1, pp. 262–349. [CrossRef]
35. Guo, Z.; Zheng, C.; Shi, B. Discrete Lattice Effects on the Forcing Term in the Lattice Boltzmann Method. *Phys. Rev. E* **2002**, *65*, 046308. [CrossRef]
36. Maquart, T.; Noël, R.; Navarro, L. Lattice Boltzmann Method for Solids (LBMS)—Library Source Code. 2020. Available online: https://github.com/tmaquart/LBMS.git (accessed on 1 March 2022).
37. Lemaitre, J.; Chaboche, J.L. *Mechanics of Solid Materials*; Cambridge University Press: Cambridge, UK, 1994.
38. Bower, A.F. *Applied Mechanics of Solids*; CRC Press: Boca Raton, FL, USA, 2009.
39. Liu, X.D.; Osher, S.; Chan, T. Weighted essentially non-oscillatory schemes. *J. Comput. Phys.* **1994**, *115*, 200–212. [CrossRef]
40. Ginzburg, I. Steady-State Two-Relaxation-Time Lattice Boltzmann Formulation for Transport and Flow, Closed with the Compact Multi-Reflection Boundary and Interface-Conjugate Schemes. *J. Comput. Sci.* **2021**, *54*, 101215. [CrossRef]
41. Postma, B.; Silva, G. Force Methods for the Two-Relaxation-Times Lattice Boltzmann. *Phys. Rev. E* **2020**, *102*, 063307. [CrossRef]
42. Ginzburg, I.; d'Humières, D. Multireflection Boundary Conditions for Lattice Boltzmann Models. *Phys. Rev. E* **2003**, *68*, 066614. [CrossRef] [PubMed]
43. Simo, J.C.; Hughes, T.J. *Computational Inelasticity*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 7.
44. Elguedj, T.; Bazilevs, Y.; Calo, V.M.; Hughes, T.J. B and F projection methods for nearly incompressible linear and non-linear elasticity and plasticity using higher-order NURBS elements. *Comput. Methods Appl. Mech. Eng.* **2008**, *197*, 2732–2762. [CrossRef]
45. Chopard, B.; Luthi, P.O. Lattice Boltzmann Computations and Applications to Physics. *Theor. Comput. Sci.* **1999**, *217*, 115–130. [CrossRef]
46. Vlasov, A.A. The Vibration Properties of Electron Gas. *Sov. Phys. Uspekhi* **1938**, *10*, 721–733. [CrossRef]
47. Liu, F.; Liu, G.; Shu, C. Fluid-Structure Interaction Simulation Based on Immersed Boundary-Lattice Boltzmann Flux Solver and Absolute Nodal Coordinate Formula. *Phys. Fluids* **2020**, *32*, 047109. [CrossRef]
48. Krüger, T.; Kusumaatmaja, H.; Kuzmin, A.; Shardt, O.; Silva, G.; Viggen, E.M. *The Lattice Boltzmann Method*; Springer: Berlin/Heidelberg, Germany, 2017; Volume 10, pp. 4–15.