UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
POST-GRADUATE PROGRAM IN COMPUTER SCIENCE

**Arthur Gonze Machado**

# Fluid Simulation with Anisotropic Pressure Segregation and Time-Dependent Tensor Fields

Juiz de Fora
2023

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Gonze Machado, Arthur.

Fluid Simulation with Anisotropic Pressure Segregation and Time-Dependent Tensor Fields / Arthur Gonze Machado. – 2023.

75 f. : il.

Advisor: Marcelo Bernardes Vieira

Dissertation (Master's) – Universidade Federal de Juiz de Fora, Instituto de Ciências Exatas. Post-Graduate Program in Computer Science, 2023.

1. Anisotropic Pressure Segregation. 2. Fluid Simulation. 3. incompressible Navier-Stokes. 4. Time-Dependent Tensor Fields. I. Surname, Marcelo Bernardes Vieira, advr. II. Title.

**Arthur Gonze Machado**

# Fluid Simulation with Anisotropic Pressure Segregation and Time-Dependent Tensor Fields

Dissertation presented to the Post-Graduate Program in Computer Science of the Universidade Federal de Juiz de Fora as a partial requirement to obtain the Master's degree in Computer Science. Concentration: Computer Science

Advisor: Ph.D. Marcelo Bernardes Vieira

Juiz de Fora

2023

**Arthur Gonze Machado**

**Fluid Simulation with Anisotropic Pressure Segregation and Time-Dependent Tensor Fields**

Dissertação apresentada ao Programa de Pós-graduação em Ciência da Computação da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Ciência da Computação. Área de concentração: Ciência da Computação.

Aprovada em 30 de junho de 2023.

BANCA EXAMINADORA

**Prof. Dr. Marcelo Bernardes Vieira** - Orientador
Universidade Federal de Juiz de Fora

**Prof. Dr. Heder Soares Bernardino**
Universidade Federal de Juiz de Fora

**Prof. Dr. Gilson Antonio Giraldi**
Laboratório Nacional de Computação Científica

Juiz de Fora, 05/06/2023.

Documento assinado eletronicamente por **Marcelo Bernardes Vieira**, **Professor(a)**, em 21/12/2023, às 12:47, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Heder Soares Bernardino**, **Professor(a)**, em 21/12/2023, às 22:51, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Gilson Antonio Giraldi**, **Usuário Externo**, em 23/12/2023, às 06:54, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Arthur Gonze Machado**, **Usuário Externo**, em 16/02/2024, às 11:58, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no Portal do SEI-Ufjf (www2.ufjf.br/SEI) através do ícone Conferência de Documentos, informando o código verificador **1314967** e o código CRC **EAA77CD6**.

# ACKNOWLEDGEMENTS

I extend my heartfelt gratitude to my parents Alpheu José Machado Júnior and Márcia Perensim Gonze, and my brothers for their unwavering support and motivation throughout my entire life, enabling me to pursue my academic goals. Without their invaluable assistance, none of this would have been possible.

I express my sincere appreciation to my advisor, Marcelo Bernardes Vieira, for his patience, friendship, and shared knowledge. His guidance has been instrumental in my academic journey.

I am grateful to my coworker and friend, Emanuel Antônio Parreiras, for his invaluable assistance in comprehending the underlying work carried out by our research group over the past years.

I would like to acknowledge the Group for Computer Graphics, Image and Vision research group for their contributions, which have greatly enriched my academic journey.

I would also like to thank my fellow lab researchers and friends, Karla Florentino and Aleksander Yacovenco, who embarked on this journey with me.

Lastly, I am deeply thankful to the FAPEMIG institution for their financial support, without which this research would not have been possible.

## RESUMO

Múltiplos fluxos complexos de fluido de sistemas naturais e de engenharia possuem uma anisotropia inerente cuja modelagem e a simulação precisa de seu comportamento ainda é um desafio. Nosso foco está em investigar a modulação anisotrópica da pressão e avaliar seu impacto no escoamento do fluido. Portanto, os campos tensoriais surgem como uma ferramenta matemática crucial que possibilita a simulação de efeitos anisotrópicos. O objetivo deste trabalho é desenvolver um modelo matemático e computacional capaz de resolver problemas de transporte de fluidos anisotrópicos. Nosso modelo proposto visa modular os termos de pressão e viscosidade usando um campo tensorial simétrico de segunda ordem positivo-definido que pode variar no espaço e no tempo. Apresentamos uma nova etapa de projeção anisotrópica projetada para acumular e ajustar a pressão à medida que o campo tensorial evolui ao longo do tempo. Esta etapa depende de um esquema de atualização de velocidade que satisfaça a condição de incompressibilidade, mesmo que o campo tensorial varie no tempo. Além disso, propomos um novo esquema de advecção anisotrópica de partículas utilizando tensores. O esquema proposto é baseado no método de "*Explicit Integration following the Velocity and Acceleration Streamline*" (X-IVAS), e implementado em uma grade estendida *Marker-and-Cell* (MAC). Apresentamos uma série de experimentos para avaliar o comportamento do escoamento de fluidos sob várias configurações de campos tensoriais. Esses experimentos são divididos em campos tensoriais constantes e campos tensoriais variantes no tempo. Além disso, mostramos um experimento que demonstra como uma força de corpo, baseada no divergente da configuração atual do campo tensorial, pode ser utilizada para influenciar o fluxo de fluido. Os resultados experimentais destacam como campos tensoriais anisotrópicos efetivamente modulam o campo de pressão, influenciando assim o comportamento do fluxo de fluido. Tais resultados demonstram a estabilidade numérica do nosso método e uma acurácia adequada para gerar efeitos visuais. Por fim, apresentamos uma discussão geral sobre as capacidades de nosso método, trabalhos futuros e possíveis aplicações em visualização.

Palavras-chave: 1. Segregação de pressão anisotrópica. 2. Simulação de fluidos incompressíveis. 3. Equações de Navier-Stokes. 4. Campos tensoriais variantes no tempo.

# ABSTRACT

The inherent anisotropy of multiple complex flows of natural and engineered systems is still a challenge when it comes to accurately modeling and simulating their behavior. Our focus lies in investigating the anisotropic modulation of pressure and evaluating its impact on fluid flow. To this end, tensor fields emerge as a crucial mathematical tool that enables the simulation of anisotropic effects. The objective of this work is to develop a mathematical and computational model capable of addressing anisotropic fluid transport problems. Our proposed model aims to modulate the pressure and viscosity terms using a symmetric positive-definite second-order tensor field that can vary both spatially and temporally. We introduce a novel anisotropic projection step designed to accumulate and adjust pressure as the tensor field evolves over time. This step relies on a velocity update scheme that ensures the divergence-free nature of the velocity field even as the tensor field undergoes temporal variations. Additionally, we propose a novel tensor-based anisotropic particle advection scheme based on the Explicit Integration following the Velocity and Acceleration Streamline (X-IVAS) method, implemented within an extended Marker-and-Cell (MAC) grid. We present a series of experiments to evaluate the behavior of fluid flow under various configurations of tensor fields. These experiments are divided into constant tensor fields and time-dependent tensor fields. Furthermore, we showcase an experiment that demonstrates how a body force, based on the divergence of the current tensor field configuration, can be utilized to influence fluid flow. The experimental results highlight how anisotropic tensor fields modulate the pressure field, thereby influencing the behavior of the fluid flow. These experimental results demonstrate our method's numerical stability and suitable accuracy for generating visual effects. Finally, we present an overall discussion of our method capabilities, future works, and possible applications over visualization.

Keywords: 1. Anisotropic Pressure Segregation. 2. Incompressible Fluid Simulation. 3. Navier-Stokes Equations. 4. Time-Dependent Tensor Fields.

# LIST OF FIGURES

## LIST OF TABLES

## LIST OF ABBREVIATIONS AND ACRONYMS

| | |
|---|---|
| CG | Computer Graphics |
| PDE | Partial Differential Equation |
| CFD | Computational Fluid Dynamics |
| 2D | Two Dimensions |
| 3D | Three Dimensions |
| SPH | Smoothed Particle Hydrodynamics |
| PIC | Particle-in-Cell |
| FLIP | Fluid Implicit Particle |
| X-IVAS | Explicit Integration following the Velocity and Acceleration Streamline |
| FSM | Fractional Step Method |
| PCM | Pressure Correction Method |
| CFL | Courant–Friedrichs–Lewy |
| MAC | Marker-and-Cell |
| VFX | Visual Effects |
| FEM | Finite Element Method |
| PFEM | Particle Finite Element Method |
| RHS | Right-hand Side |
| LHS | Left-hand Side |

## CONTENTS

# 1 INTRODUCTION

Navier-Stokes are non-linear partial differential equations that describe the flow behavior of a fluid in space [Fefferman, 2000]. It plays a crucial role in many areas of knowledge in understanding complex fluid behavior. However, the solution to these equations is still an open mathematical challenge, as the existence and smoothness of the solutions for 3 dimensions have not yet been fully achieved. Thus, most problems related to these equations require computational methods to find a solution that satisfies the conditions of the problem. The use of numerical methods combined with computational algorithms in fluid simulation problems gave rise to the Computational Fluid Dynamics (CFD) field.

Even with significant advancements in Machine Learning in the last decade, simulating fluids still heavily relies on numerical methods. The numerical methods deliver unquestionable physical fidelity to the simulation, but they are not without problems. The computational costs involved in these methods are well known, and their use has a numerical error that needs evaluation. Therefore, the Navier-Stokes equations are often simplified according to the objectives of each problem [Spurk and Aksel, 2007]. These assumptions and simplification may introduce several limitations and may not accurately capture all the details of a physically accurate fluid flow. However, these limitations are often acceptable on determined applications, i.e. generating visual effects or simulating an overly controlled specific scenario. In this work, we assume that the fluid density is constant. Therefore, we are interested in solving the incompressible Navier-Stokes equations.

As discussed by Bridson [2015], this incompressibility constraint is an important simplification for animation purposes. Physically accurate fluids indeed change their volume, e.g. sound wave propagation. However, in the macroscopic context, these perturbations in the fluid volume are so small that they are practically irrelevant for visualizing the fluid flow. Therefore, simulating the compressibility of a fluid has little to no contribution to the fluid visualization and adds a considerable amount of complexity and computational costs.

Although it is not a simplification, in this work we assume that the fluid flow is anisotropic. Assuming that the fluid flow is isotropic makes the Navier-Stokes equations more amenable to analytical or numerical solutions. The isotropic flow assumes that fluid properties, e.g. viscosity, are considered to be uniform in all orientations. However, in a considerable number of real-world scenarios, the isotropic flow assumption cannot capture the correct fluid behavior. Flow through porous media, fiber suspensions, biological systems, turbulent flows, and flow in magnetic or electric fields cannot be correctly captured by this assumption.

We are interested in the anisotropic modulation of pressure and evaluating its effects on fluid flow. In this sense, tensor fields are an important mathematical tool used to simulate anisotropic fluid flow effects. Also, it can emulate surface tension, porous walls, and fluid phase transitions. Applications of this approach can be used for scientific data visualization and Visual Effects (VFX).

## 1.1 MOTIVATION

Our primary motivation lies in the exploration of novel visual effects through the development of a fluid simulation model that incorporates an anisotropic medium property evolving with time, with a particular focus on pressure and its dynamic effects. Simulation methods such as Particle-in-Cell (PIC) and Fluid-Implicit-Particle (FLIP) have proven unsuitable for our objectives, especially as P1 Fractional-Step methods (FSM) (Sec. **2.3.1**), where pressure effects are not integrated into the advection step, resulting in an artificial loss of kinetic energy throughout the simulation.

In that sense, we use tensor fields as a mathematical tool to model anisotropic fluid behavior, seeking to couple them with the incompressible Navier-Stokes equations. This tensor field, capable of dynamic evolution, plays a crucial role in shaping the fluid flow behavior over time.

The “Explicit Integration following the Velocity and Acceleration Streamline” (X-IVAS) method has proven to be a fitting solution for our challenges. Unlike previous methods, X-IVAS, a P2 Fractional-Step method, incorporates pressure in the advection step by accumulating it throughout the simulation. This accumulated pressure is then transformed into kinetic energy, mitigating artificial numerical dissipation.

## 1.2 PROBLEM DEFINITION

This work focuses on two main problems. The first one is to address the primary challenge of constructing a mathematical model based on the incompressible Navier-Stokes equations. This model aims to modulate the viscous and pressure terms by utilizing a time-dependent tensor field (Sec. **3.1.1**).

The second problem is to adapt the X-IVAS method within its pressure segregation scheme to obtain anisotropic fluid flow (Sec. 3.2). The original X-IVAS method assumes an isotropic fluid flow, prompting us to propose an adaptation for anisotropic pressure segregation, thereby achieving anisotropic fluid flow.

## 1.3 OBJECTIVES

Our main objective is to develop a mathematical and computational model that can address anisotropic fluid transport based on symmetric second-order positive-definite tensor fields. Our model aims to modulate the pressure and viscosity terms through a tensor field that can change temporally and spatially.

Our secondary objective is to provide methods to design suitable tensor fields to modulate fluid flow. This is important to show how our propositions perform in contrast with isotropic fluid flow methods. We also aim to build upon existing models presented in the works of Renhe [2017], Vieira et al. [2021], and Parreiras [2022], further advancing the understanding and application of our tensor-based X-IVAS method.

## 1.4 CONTRIBUTIONS

Our core contributions are:

1. A novel tensor-based pressure correction method. We propose a new anisotropic projection step to accumulate and adjust pressure as the tensor field changes in time;
2. A velocity update scheme that guarantees that it remains divergence-free as the tensor field varies in time;
3. A tensor-based anisotropic particle advection scheme based on the X-IVAS method in Marker-and-Cell (MAC) grid;
4. Experimental results show how anisotropic tensor fields can be used to modulate pressure fields.

## 1.5 RELATED WORK

The development and improvement of numerical methods and models that can accurately capture the complexities of anisotropic fluid flow behavior are not new to the CFD research topic. Modeling turbulence is one of the main approaches to simulating anisotropic fluid flow. Turbulence is a fundamental problem of fluid dynamics that remains a formidable challenge for science. It appears in many physical phenomena and engineering problems. This topic has been a challenge for over 150 years as discussed by Marston and Tobias [2023].

A different approach to simulating anisotropic flow is through modeling the medium that the fluid is inserted. Fluid may be inserted into a porous medium or interact with permeable objects. Lenaerts et al. [2008] presented a smoothed-particle hydrodynamics (SPH) method capable of simulating porosity at a macroscopic scale for animation applications. This work presented significant advancements in computational efficiency over the previous works of Zhu et al. [1999], and Sawley et al. [1999]. Tran-Duc et al. [2016] showed how SPH can be used to anisotropically diffuse the fluid flow. They developed a novel diffusion operator that partially contours the smoothing done by the SPH isotropic kernel functions.

The Eulerian approach is also used to simulate porosity. A recent review of this approach was presented by Guibert et al. [2016]. This review goes way back to the foundation's work of Sanchez-Palencia [1982]. The latter shows the existence of continuous transitions of fluid flow through a porous medium using Darcy's law [Neuman, 1977], and Brinkman's model [Durlofsky and Brady, 1987]. All the aforementioned works are concerned with physical fidelity and accuracy. However, physical accuracy is not a main concern of this work as we mainly focus on effects generation and animation.

Recent contributions on anisotropic fluid flow for animations are presented by [Renhe et al., 2019, Vieira et al., 2021, Parreiras et al., 2022]. In Renhe et al. [2019], a symmetric positive-definite second-order tensor field is coupled to the incompressible Navier-Stokes equations. This tensor field rigidly conforms the fluid flow to behave according to its anisotropy. The authors proposed an anisotropic pressure correction scheme and a tensor-velocity term coupled to the advection step. This tensor field modulates the velocity and the pressure gradient over the projection step, enforcing the fluid flow to the tensor orientation. A centered grid was used to store the fluid quantities. The computation method used to solve this novel numerical scheme was created over the Stam [1999] contributions. Vieira et al. [2021] took a step further improving this approach to a 3D domain with custom numerical and computational methods. Finally, Parreiras et al. [2022]proposed several contributions to this approach. A novel numerical method was created to account for the multiple degrees of freedom of a 3D domain. A MAC grid with multiple points of reference was used to greatly improve the numerical accuracy. This numerical accuracy allows the usage of many different configurations of the tensor field to conform or anisotropically influence the fluid flow.

Advancements in machine learning and neural networks have also made a significant impact on CFD research. These advancements offered new horizons to solve fluid dynamics problems. Neural networks have been applied to fluid flow problems such as turbulence modeling, flow control, and flow prediction. This research front allows an efficient and suitable accurate solution to the governing Partial Differential Equations (PDEs).

Recent studies have utilized neural networks to solve PDEs directly or as a part of macroscopic CFD simulations. These approaches involve training these networks on available simulation data or directly learning the governing equations from data. Some examples of commonly used networks are the Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Generative Adversarial Networks (GANs). These approaches have shown promising results in terms of accuracy, efficiency, and generalization to unseen flow scenarios.

Some notable works include the development of Physics-Informed Neural Networks (PINNs)[Raissi et al., 2019]. This network has direct information about the underlying governing equations of training data. They are used for solving fluid dynamics problems related to the equations used in training. There is also the application of CNNs for solving these PDEs on an image-based training approach without the need for labeled data [Gao et al., 2021]. One final example is the usage of GANs for generating realistic and high-fidelity flow fields [Wu et al., 2020].

These recent advancements in both anisotropic fluid flow modeling and the application of neural networks to CFD hold great potential for advancing our understanding of complex fluid phenomena, improving simulation accuracy, and enabling efficient design optimization in a wide range of engineering applications.

## 1.6 OUTLINE

This dissertation is structured as follows: Chapter 2, we describe the theoretical foundations that contributed to the development of this work; Chapter 3 details the proposed method and the algorithm used to solve it; Chapter 4 we show the experimental results and discuss the proposed method capabilities; We conclude summarizing the overall characteristics and discussing the possible future work.

# 2 FUNDAMENTALS

This chapter presents a theoretical introduction to the main concepts that were used to construct the proposed method. Firstly, an introduction to orientation tensors and tensor fields. Secondly, an introduction to the Helmholtz Decomposition. Finally, an overview of the main concepts of numerical and computational methods used in CFD.

## 2.1 TENSORS

Tensors are mathematical objects that represent a multi-linear relationship between algebraic objects and mathematical, physical, or statistical quantities. The mathematical description that involves multiple dimensions and coordinate systems heavily relies on tensors. They are applied in multiple research areas in mathematics and physics, including differential geometry, electromagnetism, quantum mechanics, and general relativity.

Tensors are classified by their rank or order $k$ [Kolecki, 2002]. Also, the components of a tensor depend on the basis vector of the coordinate system that is being considered. The considered Euclidean space dimension will be denoted by $S$. The relation between the tensor order and space dimension is described as $S^k$. In this work, the $\mathbb{R}^3$ will be the reference Euclidean space. Therefore, the tensors will assume the form of $3^k$ and could represent the following mathematical objects:

Table 1 – Example of mathematical objects that a tensor can represent $\in \mathbb{R}^3$ varying $k$.

| k | Components | Mathematical Object | Characteristic |
|---|---|---|---|
| 0 | $3^0 = 1$ | Scalar | Describes a magnitude |
| 1 | $3^1 = 3$ | Vector | Describes a magnitude and a direction |
| 2 | $3^2 = 9$ | Matrix | Describes 3 magnitudes and directions |

Source: Created by the author.

### 2.1.1 Tensor fields

Tensor fields are a generalization of a scalar and a vector field. Therefore, a tensor of any given rank is assigned to every point of domain space. As they extend the tensor applications, they are commonly used to describe physical quantities throughout the space. Some examples are stress tensor fields in solid mechanics, electromagnetic tensor fields in electromagnetism, metric tensor fields in general relativity, and the deformation tensor field in continuum mechanics.

Our field is formed by positive definite second-order tensors, also known as orientation tensors [Westin, 1994]. Orientation tensors are commonly applied in the field of materials science. This is due to the concise way that these tensors describe and quantify the orientation of an object or system in 3D spaces. They are used to predict how a material will deform under stress, or fluid diffusion, or simulate how a wave propagates throughout an anisotropic medium [Liu et al., 2002, Basser and Pierpaoli, 2011, Spurk and Aksel, 2007]. At each point of the domain space $D \in \mathbb{R}^3$, these tensors are represented by $3 \times 3$ matrices. The components of these matrices can change from point to point and in time. An orientation tensor $\mathbf{T}$ can be mathematically represented as:

$$\mathbf{T} = \sum_{i=1}^{M} \lambda_i \mathbf{e}_i \mathbf{e}_i^{\mathrm{T}},$$

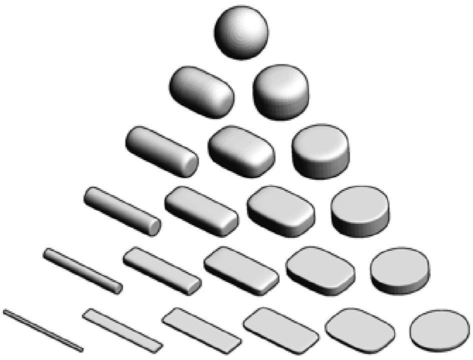
where $\mathbf{e}_i$ is a eigenvector related to the $i$th eigenvalue $\lambda$. The tensor $\mathbf{T}$ will be oriented by the directions denoted by each of its eigenvectors and scaled by its eigenvalues.

Our tensors will be geometrically represented as an ellipsoid following Kindlmann [2004] work. In this way, problems related to asymmetric representation and visual ambiguity are bypassed. As an ellipsoid, the tensor eigenvalues and eigenvectors will dictate its scale and orientation respectively. The eigenvalues variation will dictate the tensor anisotropy and shape as shown in Figure 1. Assuming eigenvalues $\lambda_1$, $\lambda_2$, and $\lambda_3$, the shape of the tensor will be as follows:

- $\lambda_1 \approx \lambda_2 \approx \lambda_3 \Rightarrow$ Spherical;
- $\lambda_1 \gg \lambda_2 \approx \lambda_3 \Rightarrow$ Linear;
- $\lambda_1 \approx \lambda_2 \gg \lambda_3 \Rightarrow$ Planar.

The spherical tensor is the identity matrix or it is scaled by a scalar. This shape also simplifies the analysis of the tensor field influence over the fluid diffusion or the medium anisotropy. While the linear and planar tensors have preferred directions, the spherical tensors have none.

Figure 1 – Superquadic tensor glyph shapes with different eigenvalues.



Source: Kindlmann [2004].

### 2.1.2 Strain-rate Tensor

The strain-rate tensor is a second-order tensor used to analyze the behavior of materials under deformation. This tensor denotes the local rates of deformation or strain experienced by a material. In a tensor field, it serves as a measure of how fast the material is stretching, compressing, shearing, or rotating at each point of the domain [Bridson, 2015]. In the context of fluid mechanics, the strain-rate tensor is commonly used to construct the viscous stress tensor $\tau$.

The strain-rate tensor is defined by the velocity field gradient $\nabla\mathbf{u}$ in a Cartesian coordinate system. As discussed by Landau and Lifshitz [2013], the velocity field gradient can be decomposed as the sum of a symmetric part $\mathbf{D}$, and an anti-symmetric part $\mathbf{W}$ as:

$$\nabla\mathbf{u} = \mathbf{D} + \mathbf{W}.$$

Where $\mathbf{D}$ and $\mathbf{W}$ are the strain-rate and the spin tensor respectively. They are denoted as:

$$\begin{cases} \mathbf{D} = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right), \\ \mathbf{W} = \frac{1}{2}\left(\nabla\mathbf{u} - \nabla\mathbf{u}^T\right). \end{cases}$$

While the strain-rate tensor is related to the rate of deformation, the spin tensor is related to the rate of rotation. Considering the partial terms, the $D$ tensor is denoted as:

$$\mathbf{D} = \begin{bmatrix} \frac{\partial\mathbf{u}_x}{\partial x} & \frac{1}{2}\left(\frac{\partial\mathbf{u}_x}{\partial y} + \frac{\partial\mathbf{u}_y}{\partial x}\right) & \frac{1}{2}\left(\frac{\partial\mathbf{u}_x}{\partial z} + \frac{\partial\mathbf{u}_z}{\partial x}\right) \\ \frac{1}{2}\left(\frac{\partial\mathbf{u}_x}{\partial y} + \frac{\partial\mathbf{u}_y}{\partial x}\right) & \frac{\partial\mathbf{u}_y}{\partial y} & \frac{1}{2}\left(\frac{\partial\mathbf{u}_y}{\partial z} + \frac{\partial\mathbf{u}_z}{\partial y}\right) \\ \frac{1}{2}\left(\frac{\partial\mathbf{u}_x}{\partial z} + \frac{\partial\mathbf{u}_z}{\partial x}\right) & \frac{1}{2}\left(\frac{\partial\mathbf{u}_y}{\partial z} + \frac{\partial\mathbf{u}_z}{\partial y}\right) & \frac{\partial\mathbf{u}_z}{\partial z} \end{bmatrix},$$

where $\mathbf{u}$ is the velocity vector. Any strain tensor can be decomposed into hydrostatic and deviatoric strains. Considering the strain-rate tensor, we will denote the hydrostatic and the deviatoric strains as $\mathbf{D}_{hyd}$ and $\mathbf{D}_{dev}$ respectively. The hydrostatic strain is a scalar quantity that is the strain tensor trace:

$$\mathbf{D}_{hyd} = \operatorname{tr}(\mathbf{D}) = \frac{\partial\mathbf{u}_x}{\partial x} + \frac{\partial\mathbf{u}_y}{\partial y} + \frac{\partial\mathbf{u}_z}{\partial z} = \nabla\cdot\mathbf{u},$$

where $\operatorname{tr}(\mathbf{D})$ is the trace of $\mathbf{D}$; This strain is closely related to the fluid volume change. It is a convenient approximation when strains are small. It is important to notice that $\mathbf{D}_{hyd} = \nabla\cdot\mathbf{u}$. Considering the rate of deformation of an incompressible fluid, the trace of $\mathbf{D}$ must be:

$$\operatorname{tr}(\mathbf{D}) = \nabla\cdot\mathbf{u} = 0.$$

However, if the strain is not small, the volume change is better measured by the deformation gradient determinant. Therefore, $\mathbf{D}_{dev}$ will be closely related to the fluid deformation in an incompressible flow, considering small strains. $\mathbf{D}_{dev}$ is obtained after subtracting $\mathbf{D}_{hyd}$ from $\mathbf{D}$. This strain measures all the deformation that changes the shape of the fluid but not its volume.

## 2.2 HELMHOLTZ DECOMPOSITION

The Helmholtz Decomposition presented by Dassios and Lindell [2002], served as the main theoretic background for the custom anisotropic decomposition presented in this work. Throughout this decomposition, it is ensured that the pressure field can be reconstructed by solving the following Poisson equation:

$$\nabla \cdot \nabla \mathbf{p}\left(\mathbf{x}\right) = \nabla \cdot \mathbf{u}\left(\mathbf{x}\right),$$

where $\mathbf{p}$ and $\mathbf{u}$ represent vector fields with continuous second derivatives. In the projection step, the divergence-free velocity field is obtained by subtracting the pressure field from the current velocity field.

### 2.2.1 Isotropic Decomposition

Given a continuous vector field $\mathbf{u}\left(\mathbf{x}\right)$ in $\overline{\mathbf{D}} = \mathbf{D} \cup \partial\mathbf{D}$ with $\mathbf{D} \in \mathbb{R}^3$. Assuming this vector field has continuous first derivatives in $\mathbf{D}$. The Helmholtz decomposition states that the vector field $\mathbf{u}\left(\mathbf{x}\right)$ can be reconstructed from its scalar and vector invariants, such that:

$$\begin{cases} \mathbf{u}\left(\mathbf{x}\right) = \nabla\phi\left(\mathbf{x}\right) + \nabla \times \mathbf{a}\left(\mathbf{x}\right), \\ \nabla \cdot \mathbf{a}\left(\mathbf{x}\right) = 0, \end{cases} \tag{2.1}$$

where $\phi\left(\mathbf{x}\right)$ is a scalar field and $\mathbf{a}\left(\mathbf{x}\right)$ is a vector field. They are named "scalar potential" and "vector potential" respectively. The $\nabla\phi\left(\mathbf{x}\right)$ term is irrotational and $\nabla \times \mathbf{a}\left(\mathbf{x}\right)$ is solenoidal and therefore divergence-free. Dassios and Lindell [2002] also state that this decomposition is unique to the additive gradient of a harmonic function. Therefore, Equation 2.1 can be rewritten as:

$$\mathbf{u}\left(\mathbf{x}\right) = \left[\nabla\phi\left(\mathbf{x}\right) + \nabla\mathbf{v}\left(\mathbf{x}\right)\right] + \left[\nabla \times \mathbf{a}\left(\mathbf{x}\right) - \nabla\mathbf{v}\left(\mathbf{x}\right)\right],$$

where $\Delta\mathbf{v}\left(\mathbf{x}\right) = 0 \in \mathbf{D}$.

### 2.2.2 Anisotropic Decomposition

Dassios and Lindell [2002] also state that the uniqueness and reconstruction of the vector field $\mathbf{u}\left(\mathbf{x}\right)$ are achievable by the anisotropic Helmholtz decomposition. The anisotropic decomposition couple $\mathbf{T}$ and $\mathbf{S}$ is symmetric and positive definite dyadic in $\mathbb{R}^3$ to Equation 2.1. The decomposition is described as:

$$\begin{cases} \mathbf{u}\left(\mathbf{x}\right) = \nabla_{\mathbf{T}}\phi\left(\mathbf{x}\right) + \nabla_{\mathbf{S}} \times \mathbf{a}\left(\mathbf{x}\right), \\ \nabla_{\mathbf{T}} \cdot \mathbf{a}\left(\mathbf{x}\right) = \left(\mathbf{T} \cdot \nabla\right) \cdot \mathbf{a} = 0, \end{cases} \tag{2.2}$$

where $\nabla_{\mathbf{S}}$ and $\nabla_{\mathbf{T}}$ are the **S**-Gradient and **T**-gradient terms, respectively. The **T**-gradient of $\phi$ is a **T**-irrotational field, and the **S**-rotation of **a** is a **S**-solenoidal field. Considering the **T**-gradient term, it can be written as:

$$\nabla_{\mathbf{T}} = \mathbf{T} \cdot \nabla = \begin{bmatrix} \lambda_1 \frac{\nabla}{\nabla_{\mathbf{e}_1}} \\ \lambda_2 \frac{\nabla}{\nabla_{\mathbf{e}_2}} \\ \lambda_3 \frac{\nabla}{\nabla_{\mathbf{e}_3}} \end{bmatrix}, \tag{2.3}$$

where $\lambda_n$ with $1 \leq n \leq 3$ is a positive eigenvalue of **T** that corresponds to an orthogonal unit eigenvector $\mathbf{e}_n$. The **S**-gradient assumes the same form. Therefore, the gradient operator presented in Equation 2.1 is now medium-dependent. The gradient operator now depends on the directional characteristics of the anisotropic medium described by its respective tensor. It is easily seen that if the tensors are equal to the identity matrix, the $\nabla_{\mathbf{T}}$ or $\nabla_{\mathbf{S}}$ terms will be equivalent to the original $\nabla$ operator.

Consider the following **ST**-Poisson equation:

$$\nabla_{\mathbf{S}} \cdot \nabla_{\mathbf{T}} \mathbf{p}(\mathbf{x}) = \nabla_{\mathbf{S}} \cdot \mathbf{u}(\mathbf{x}), \tag{2.4}$$

where $\mathbf{p}(\mathbf{x})$ is a vector field with continuous second derivatives. Then, the full anisotropic Helmholtz decomposition of $\nabla_{\mathbf{S}} \cdot \mathbf{u}(\mathbf{x})$ will assume the form of:

$$\nabla_{\mathbf{S}} \cdot \nabla_{\mathbf{T}} \mathbf{p}(\mathbf{x}) = \nabla_{\mathbf{S}} \cdot \nabla_{\mathbf{T}} \phi(\mathbf{x}) + \nabla_{\mathbf{S}} \cdot \nabla_{\mathbf{S}} \times \mathbf{a}(\mathbf{x}).$$

Considering the gauge condition $\nabla_{\mathbf{S}} \cdot \nabla_{\mathbf{S}} \times \mathbf{a}(\mathbf{x}) = 0$, the **S**-solenoidal part of $\nabla_{\mathbf{S}} \cdot \mathbf{u}(\mathbf{x})$ is divergence-free, subject to the eigensystem of **S**. But we are interested in the special case of $\mathbf{S} = \mathbf{I}$, as proposed by Vieira et al. [2021]. In this work, the anisotropic Helmholtz decomposition refers to the solution of the **T**-Poisson problem:

$$\nabla \cdot \nabla_{\mathbf{T}} \mathbf{p}(\mathbf{x}) = \nabla \cdot \mathbf{u}(\mathbf{x}), \tag{2.5}$$

such that $\mathbf{u}(\mathbf{x}) = \nabla_{\mathbf{T}} \phi(\mathbf{x}) + \nabla \times \mathbf{a}(\mathbf{x})$, meaning that the resulting vector field $\nabla \times \mathbf{a}(\mathbf{x})$ is solenoidal in respect to the standard reference system, i.e. is **I**-solenoidal. The vector field $\nabla \times \mathbf{a}(\mathbf{x})$ should be divergence-free in the standard reference system to easily handle anisotropic fluid simulation.

## 2.3 FLUID SIMULATION

In this section, we present some important numerical and computational methods to solve the incompressible Navier-Stokes equations. Considering that, an isotropic acceleration integration scheme is presented. Finally, an anisotropic model of the incompressible Navier-Stokes and how it is numerically solved.

The Navier-Stokes equations consist of two separate parts: the continuity and the momentum equations. In this section, the basic conservation principles and laws that are utilized to derive the incompressible Navier-Stokes equations will be briefly presented. As discussed by Ferziger et al. [2002], more detailed derivations can be found in various standard texts on fluid mechanics, such as Bird [2006] and WHITE [2010].

The continuity equation is responsible for the mass conservation principle. Consider the rate of change in an arbitrary volume $\Omega$ of a fluid body and its boundary surface $\delta\Omega$. The conservation principle states that the volume rate of change of $\Omega$ is equal to the net flux of mass that crosses $\delta\Omega$. This rate of change can be computed as:

$$\int\int_{\delta\Omega} \mathbf{u} \cdot \hat{n} = \int\int\int_{\Omega} \nabla \cdot \mathbf{u},$$

where $\hat{n}$ is the normal component. When the control volume is allowed to become infinitesimally small, a differential coordinate-free form of the continuity equation arises. It is denoted on Ferziger et al. [2002] as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0.$$

As the density $\rho$ of an incompressible fluid is constant and uniform, the continuity equation is resumed to:

$$\nabla \cdot \mathbf{u} = 0. \tag{2.6}$$

For an incompressible fluid, its velocity field is divergence-free. The momentum equation describes the conservation of momentum. This conservation equation is derived from Newton's second law of motion:

$$\frac{d\,(m\mathbf{u})}{dt} = \sum \mathbf{f},$$

where $m$ stands for mass, $t$ is time, $\mathbf{u}$ is the velocity and $\mathbf{f}$ are the forces acting on the particle. In this work, the incompressible Navier-Stokes momentum equation follows the model presented by Ferziger et al. [2002] and it is written as:

$$\rho \left( \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\, \mathbf{u} \right) = -\nabla \mathrm{p} + \mu \nabla \cdot (\nabla \mathbf{u}) + \mathbf{f}, \tag{2.7}$$

where $\mathbf{u}$ is the fluid velocity in $m/s$, $t$ is time in seconds, $\rho$ is the specific mass $kg/m^3$, p is pressure $kg/(m \cdot s^2)$, $\mu$ is the viscosity coefficient $kg/(m \cdot s)$, $\mathbf{f}$ is the sum of all body and external forces $(kg \cdot m)/s^2$. The left-hand side (LHS) is composed of the fluid acceleration plus the velocity contribution to this acceleration, called convective acceleration. On the right-hand side (RHS), there is the gradient pressure term, the viscous forces, and the external body forces acting on the fluid.

### 2.3.1 Fractional-Step Method

The Navier-Stokes equations are highly non-linear and require numerical methods to approximate their solution. As they are difficult to solve as a whole, the Splitting Method is used. This method involves splitting the original PDE into simpler subcomponents that will be solved independently. This approach allows the advection and diffusion steps to be solved using explicit and implicit numerical methods, respectively. Following the work of Stam [1999], the incompressible Navier-Stokes equations can be split into the following steps:

1. Body Forces Step → $\frac{\partial \mathbf{U}}{\partial t} = \mathbf{F} - \frac{1}{\rho}\nabla \mathrm{P}$;
2. Advection Step → $\frac{\partial \mathbf{q}}{\partial t} = 0$;
3. Diffusion Step → $\frac{\partial \mathbf{U}}{\partial t} = \frac{1}{\rho}\nabla \cdot \left(\mu \left(\nabla \mathbf{U} + \nabla \mathbf{U}^T\right)\right)$;
4. Incompressibility Constraint Step → $\nabla \cdot \mathbf{U} = 0$.

The $\mathbf{q}$ term was used to illustrate that we can advect different fluid properties, i.e. density, velocity, and positions. The same is true for the $\mathbf{F}$ term used in body forces, as we may want to apply forces differently than just gravity. Chorin [1967] proposed a numerical method that uses operator splitting to solve the incompressible Navier-Stokes equations. This method is called the Pressure-Correction method but is also known as the Projection or the Fractional-Step method.

The Pressure-Correction method splits its computations into a predictor and a corrector step. The predictor step will integrate the convective and diffusive terms in time. In the predictor step, the velocity field is only an approximation and is not divergence-free. This approximation is obtained using only the current velocity field and the available information on body forces. The current information on the pressure can be used for this initial approximation, but it is not required [Ferziger et al., 2002]. Without considering the pressure effects, the approximated velocity field is obtained through:

$$\mathbf{U}^* = \mathbf{U}^n + \int_n^{n+t} \frac{1}{\rho} \left[\nabla \cdot \left(\mu \left(\nabla \mathbf{U}^n + \nabla \left(\mathbf{U}^n\right)^T\right)\right) + \mathbf{F}^n \left(\mathbf{x}^t\right)\right] dt,$$

where $\mathbf{U}^*$ is the velocity field approximation. Once discretized, this step can be time-integrated explicitly. At the corrector step, the pressure is computed and used to satisfy the velocity field incompressibility condition [Bridson, 2015]. As discussed in Ferziger et al. [2002], the continuity equation constraint implies that we need to take the divergence of Equation 2.7. In this sense, at step 4, we need to satisfy the continuity equation by computing the divergence of the current velocity field. Therefore, the pressure is obtained by solving the following Poisson equation:

$$\nabla \cdot \left(\frac{1}{\rho}\nabla \mathrm{P}^{n+1}\right) = \frac{1}{\Delta t}\nabla \cdot \mathbf{U}^*. \tag{2.8}$$

A numerical method for elliptic equations as a Finite Difference or Finite Volume Method may be used to solve this equation. The Helmholtz-Hodge Decomposition described in Section 2.2 guarantees the uniqueness and reconstruction of the gradient pressure field. The divergence-free velocity field is then obtained from the pressure correction on the estimated velocity field:

$$\frac{\mathbf{U}^{n+1} - \mathbf{U}^{*}}{\Delta t} = -\frac{1}{\rho}\nabla \mathrm{P}^{n+1}.$$

Table 2 – Fractional-Step Schemes Labels.

| Label | Description |
|---|---|
| P1 | The momentum equation does not account for pressure effects. The new pressure is computed at the corrector step. The new pressure is obtained from the pressure Poisson equation. |
| P2 | The momentum equation account for the pressure from the previous iteration. The pressure correction is obtained by solving the Poisson equation at the corrector step. |
| P3 | Same as P2 but the pressure is extrapolated from the two previous iterations. Therefore, this scheme is second-order accurate. |

Source: Gresho [1990]

[Gresho, 1990] presented a useful set of labels to classify different fractional-step schemes. Those labels are depicted in Table 2.

### 2.3.2 Particle-in-Cell Methods

After discussing the discretization of the transport equations terms, we will delve into the computational methods to solve them. Computational methods provide the means to simulate and analyze complex fluid phenomena that are difficult or impossible to study analytically or experimentally.

Considering this, the Particle-In-Cell (PIC) and Fluid-Implicit Particle (FLIP) methods were fundamental for the conception and theoretical development of this work [Harlow, 1962, Brackbill and Ruppel, 1986]. The fundamental characteristic of these methods that benefits our work is how the transport equation terms are solved. These methods introduce discrete particles within each grid cell to represent the physical particles in the system. The integration of the convective terms, or the “advection step” is solved in a particle context. The integration of all other terms of the transport equations is solved on the grid. Therefore, there is no filtering or information loss on the advection step regardless of the flow movement Bridson [2015].

Harlow [1962], proposed PIC be executed in three or four phases as follows:

- Phase 1: The Eulerian field functions are modified. Therefore, transport is neglected due to fluid motion;
- Phase 2: All particle quantities i.e. density, pressure, and velocities are advected. This advection follows a finite difference scheme of the transport term of Equation

2.7. This updated particle information is linear interpolated to the grid. This interpolation is weighted by the particle position to the center of its neighbor's grid cells;

- Phase 3: The transport information on the grid is corrected;
- Phase 4: This is an optional phase in which various diagnostic functions of motion are computed.

However, this method changed throughout the years. Advancements in pressure correction methods and advection schemes were fundamental to updating the PIC method. A recent approach is discussed in Bridson [2015]. Bridson [2015] uses a MAC grid instead of a cell-centered grid. As proposed by Harlow and Welch [1965], in a MAC grid, the vector quantities and scalar quantities are stored at the grid face's center and cell's center respectively. Therefore, the PIC method interpolation steps were changed to accommodate this new grid approach. Consider an $x$-aligned left cell face, the velocity interpolation is described as:

$$\mathbf{U}_{i-\frac{1}{2},j,k} = \frac{\sum_{p\in N} \mathbf{U}_p k\left(\mathbf{x}_p - \mathbf{x}_{i-\frac{1}{2},j,k}\right)}{\sum_{p\in N} k\left(\mathbf{x}_p - \mathbf{x}_{i-\frac{1}{2},j,k}\right)}, \tag{2.9}$$

where $N$ is the set of all the particles inside the cell and its neighbors, $\mathbf{U}$ is the grid velocity that is going to be interpolated, $\mathbf{x}_p$ is the particle position, and $\mathbf{x}_{i+\frac{1}{2},j,k}$ is the face center position. The $k$ term is a trilinear hat interpolation kernel function:

$$k\left(x, y, z\right) = h\left(\frac{x}{\Delta x}\right) * h\left(\frac{y}{\Delta x}\right) * h\left(\frac{z}{\Delta x}\right),$$

where

$$h\left(r\right) = \begin{cases} 1 - r, \text{ if } 0 \leq r \leq 1, \\ 1 + r, \text{ if } -1 \leq r \leq 0, \\ 0 \quad , \text{ if otherwise.} \end{cases}$$

The other 5 faces are interpolated analogously. Ultimately, the new PIC steps can be enumerated as:

1. Particle quantities are interpolated to the grid using Equation 2.9;
2. Integrate grid quantities and project the velocity field. At this projection step, the divergence-free velocity field is obtained through Equation 2.8;
3. Interpolate the grid information back to the particles using the reverse procedure of Equation 2.9;
4. Integrate the particle quantities in time.

The main problem of this method is numerical dissipation. This numerical dissipation is due to double interpolations in every time step. This operation adds considerable smoothing and piles up as the simulation evolves. Due to this problem, Brackbill and Ruppel [1986] proposed the FLIP method.

The FLIP method modified the way that the particle velocity is updated. In the second interpolation, the grid information does not overwrite the particle velocity. The particle velocities receive an increment computed on every iteration, considering the difference over the previous iteration grid. Considering the enumerated PIC steps, one new step is added to save the previous grid. This previous iteration grid is used to compute the increment that will be interpolated to the particles. In this way, the method is almost free from numerical diffusion. One main drawback of this approach is the noise added to the flow with every information increment through interpolation [Bridson, 2015].

#### *2.3.2.1 X-IVAS: eXplicit Integration following the Velocity and Acceleration Streamlines*

Another method to contour the PIC numerical dissipation was first proposed by Nigro et al. [2011] and further improved by Idelsohn et al. [2012, 2013, 2014], and Nadukandi et al. [2017]. They proposed a method that integrates not just velocity in time, but also an acceleration term. This method is called X-IVAS, and it is classified as a P2 fractional-step method. In the advection step, it performs explicit integration following the velocity and acceleration streamlines. It provides pressure estimation through integration in time of the Equation 2.8 solution. It allows for large time steps, i.e. convergence condition by Courant–Friedrichs–Lewy (CFL) $\geq 1$, with stability and suitable numerical accuracy.

The explicit integration scheme that follows the velocity and acceleration streamlines are:

$$\begin{cases} \mathbf{x}^{n+t} = \mathbf{x}^{n} + \int_{n}^{n+t} \mathbf{U}^{n}\left(\mathbf{x}^{\tau}\right) d\tau, \\ \mathbf{U}_{p}^{n+t} = \mathbf{U}_{p}^{n} + \int_{n}^{n+t} \mathbf{A}^{n}\left(\mathbf{x}^{\tau}\right) d\tau, \end{cases} \tag{2.10}$$

where $\mathbf{x}$ is the particle position, $\mathbf{U}^n$ is the velocity field, and $\mathbf{A}^n$ is the acceleration field. The $\mathbf{U}_p^n$ term denotes a notation simplification that will be used throughout this work. This notation describes that the grid and particle position considered for interpolation are in the same time step. The parenthesis notation that denotes interpolation at the particle position is replaced by the subscript $p$ in this case.

The acceleration term is denoted as:

$$\mathbf{A}^{n}\left(\mathbf{x}^{\tau}\right) = \frac{1}{\rho}\left[\nabla \cdot \sigma_{p}^{\tau} + \mathbf{F}_{p}^{\tau}\right], \tag{2.11}$$

where $\sigma^{\tau}$ is the Cauchy stress tensor, and $\mathbf{F}_p^{\tau}$ is the sum of all body and external forces field. As proposed by Batchelor and Batchelor [1967], the Cauchy stress tensor accounts for the effects of viscosity and has the following form:

$$\sigma_p^\tau = \tau_p^\tau - \mathrm{P}_p^\tau \mathbf{I}, \tag{2.12}$$

where $\tau_p^t$ is a symmetric tensor, and $\mathrm{P}_p^t$ is the pressure field.

The pressure must be separated from the stress tensor, as it will be used to handle the incompressibility constraint. The symmetric tensor $\tau$ is used to model multiple fluid behaviors, e.g. the effects of viscous resistance to changing bulk or Non-Newtonian fluids [Bridson, 2015]. This tensor is referred to as the “deviatoric tensor”, and is described as:

$$\tau_p^t = \mu \left\{ \nabla \mathbf{U}_p^t + \left[ \nabla \mathbf{U}_p^t \right]^T \right\}.$$

Expanding the Cauchy stress tensor in Equation 2.11, the acceleration term becomes:

$$\mathbf{A}^n \left( \mathbf{x}^t \right) = \frac{1}{\rho} \left[ \nabla \cdot \tau^n \left( \mathbf{x}^t \right) - \nabla \mathrm{P}_p^t + \mathbf{F}^n \left( \mathbf{x}^t \right) \right]. \tag{2.13}$$

For the method to remain unconditionally stable, the pressure needs to remain implicit, as discussed in Nigro et al. [2011]. So, the pressure gradient term of Equation 2.13 is expressed in terms of explicit and implicit terms:

$$\int_n^{n+1} \left[ \nabla \mathrm{P}_p^t \right] dt = \int_n^{n+1} \left\{ \nabla \mathrm{P}^n \left( \mathbf{x}^t \right) + \nabla \left[ \delta \mathrm{P} \left( \mathbf{x}^t \right) \right] \right\} dt.$$

The $\delta \mathrm{P}$ is the implicit pressure term and can be expressed as:

$$\delta \mathrm{P} \left( \mathbf{x}^t \right) = \mathrm{P}_p^t - \mathrm{P}^n \left( \mathbf{x}^t \right).$$

As the pressure is expressed by the sum of these two terms, the method becomes semi-explicit for the pressure-velocity time integration scheme. Therefore, the semi-explicit velocity integration scheme is divided into two equations that will be solved in different steps:

$$\begin{cases} \hat{\mathbf{U}}_p^{n+1} = \mathbf{U}_p^n + \int_n^{n+1} \hat{\mathbf{A}}^n \left( \mathbf{x}^t \right) dt, \\ \hat{\mathbf{A}}^n \left( \mathbf{x}^t \right) = \frac{1}{\rho} \left[ \nabla \cdot \tau^n \left( \mathbf{x}^t \right) - \nabla \mathrm{P}^n \left( \mathbf{x}^t \right) + \mathbf{F}^n \left( \mathbf{x}^t \right) \right], \end{cases} \tag{2.14}$$

and

$$\mathbf{U}_p^{n+1} = \hat{\mathbf{U}}_p^{n+1} - \frac{1}{\rho} \int_n^{n+1} \nabla \left[ \delta \mathrm{P} \left( \mathbf{x}^t \right) \right] dt, \tag{2.15}$$

where $\hat{\mathbf{U}}^{n+1}$ is the first velocity field approximation that will be referred to as the “advected velocity” field. The $\hat{\mathbf{A}}^n$ term is the explicit acceleration, and $\mathbf{U}^{n+1}$ is the divergence-free velocity field.

The first step will solve Equation 2.14. A first approximation of the divergence-free velocity field is obtained from the explicit information. The second step will solve the implicit pressure gradient term to correct the first velocity field approximation. Solving the second equation, the divergence-free velocity field is obtained.

Applying the divergence operator at both sides of Equation 2.15, and considering the velocity field divergence-free restriction. The following pressure Poisson equation is obtained:

$$\nabla \cdot \hat{\mathbf{U}}_p^{n+1} = \nabla \cdot \left\{ \frac{1}{\rho} \nabla \left[ \delta \mathrm{P} \left( \mathbf{x}^{n+1} \right) \right] \right\} \frac{\Delta t}{2}. \tag{2.16}$$

The divergence-free velocity field is then obtained by:

$$\mathbf{U}_p^{n+1} = \hat{\mathbf{U}}_p^{n+1} - \frac{1}{\rho} \nabla \left[ \delta \mathrm{P}_p \right] \frac{\Delta t}{2}. \tag{2.17}$$

Finally, the estimated pressure field is incremented as:

$$\mathrm{P}_p^{n+1} = \mathrm{P}^n \left( \mathbf{x}^{n+1} \right) + \delta \mathrm{P} \left( \mathbf{x}^{n+1} \right). \tag{2.18}$$

Although this velocity is interpolated directly to the particle, smoothing its information, the X-IVAS method accounts for the kinetic energy loss in the projection step on the next iteration advection step.

#### *2.3.2.2 Anisotropic fluid simulation*

We are interested in the complexities of the anisotropic fluid flow. Our approach uses tensors to represent the anisotropic properties of the fluid. Considering that, an important method that used the same approach was proposed by Parreiras [2022]. The work of Parreiras [2022] proposed advancements in Renhe et al. [2019] method. These works modified the incompressible Navier-Stokes constitutive equations to describe the relationship between the tensor field and the fluid. The new proposed continuum equations are:

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} - \frac{1}{\rho} \nabla_{\mathbf{T}} p + \nu \nabla \cdot \nabla_{\mathbf{T}} \mathbf{u} + (\mathbf{T}\mathbf{u} - \beta \mathbf{u}) + \mathbf{f}, \tag{2.19}$$

restricted to:

$$\begin{cases} \nabla \cdot \mathbf{u} = 0, \\ \frac{\partial \mathbf{T}}{\partial t} = \mathbf{0}. \end{cases} \tag{2.20}$$

Where $\nu$ is the kinematic viscosity $(m^2/s)$. It is defined as the ratio of the dynamic viscosity $\mu$ over the specific mass of the fluid $\rho$:

$$\nu = \frac{\mu}{\rho}.$$

The **T**-gradient term $\nabla_{\mathbf{T}}$ is denoted by Equation 2.3. The term $\beta$ is a scalar variable. This modified model requires a specialized numerical method that can handle the new anisotropic properties. The proposed numerical method is a P1 fractional-step method. They used FLIP as the computational method used to solve this modified numerical method. Among all the proposed modifications, we will delve into the advection step, interpolation, and pressure correction steps.

The advection step was modified to accommodate the new tensor term $\mathbf{Tu} - \beta\mathbf{u}$. The explicit three-stage third-order Runge-Kutta time integration scheme denoted by Ralston [1962] was modified to:

$$\begin{aligned}
\mathbf{k}_1 &= \hat{\mathbf{u}}(\mathbf{x}^n) + \Delta t\left(\hat{\mathbf{T}}(\mathbf{x}^n)\hat{\mathbf{u}}(\mathbf{x}^n) - \hat{\mathbf{u}}(\mathbf{x}^n)\right), \quad \mathbf{x}' = \mathbf{x}^n + \frac{1}{2}\Delta t\mathbf{k}_1, \\
\mathbf{k}_2 &= \hat{\mathbf{u}}(\mathbf{x}') + \Delta t\left(\hat{\mathbf{T}}(\mathbf{x}')\hat{\mathbf{u}}(\mathbf{x}') - \hat{\mathbf{u}}(\mathbf{x}')\right), \quad \mathbf{x}'' = \mathbf{x}^n + \frac{3}{4}\Delta t\mathbf{k}_2, \\
\mathbf{k}_3 &= \hat{\mathbf{u}}(\mathbf{x}'') + \Delta t\left(\hat{\mathbf{T}}(\mathbf{x}'')\hat{\mathbf{u}}(\mathbf{x}'') - \hat{\mathbf{u}}(\mathbf{x}'')\right), \\
\mathbf{x}^{n+1} &= \mathbf{x}^n + \frac{2}{9}\Delta t\mathbf{k}_1 + \frac{3}{9}\Delta t\mathbf{k}_2 + \frac{4}{9}\Delta t\mathbf{k}_3,
\end{aligned} \tag{2.21}$$

where $\hat{\mathbf{u}}(\cdot)$ and $\hat{\mathbf{T}}(\cdot)$ are the velocity and tensor fields interpolated on the current particle position. This results in the particle velocity being forced to align with the tensor orientation. Therefore, the particles are transported considering the anisotropic properties of the tensor field.

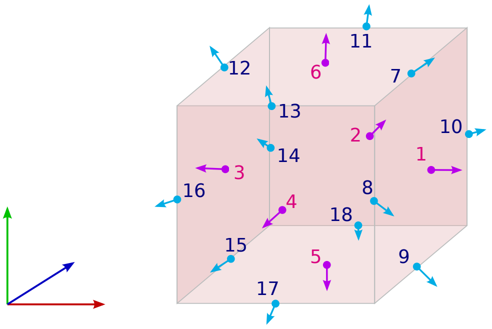
Renhe [2017] proposed a modified projection step that considers the anisotropic properties of the medium. The pressure is influenced by the proposed T-Laplacian operator. This modification implies that the pressure must account for the anisotropic properties of the medium. The new pressure Poisson equation is denoted by:

$$\nabla \cdot \nabla_{\mathbf{T}} \mathrm{P}\left(\mathbf{x}\right) = \nabla \cdot \mathbf{U}\left(\mathbf{x}\right). \tag{2.22}$$

Therefore, the pressure-correction step anisotropically updates the velocity field as:

$$\mathrm{P}_{\mathbf{T}}(\mathbf{U}\left(\mathbf{x}\right)) = \mathbf{U}\left(\mathbf{x}\right) - \nabla_{\mathbf{T}} \mathrm{P}\left(\mathbf{x}\right). \tag{2.23}$$

Figure 2 – Faces and edges normals of a cell.



Source: Parreiras [2022].

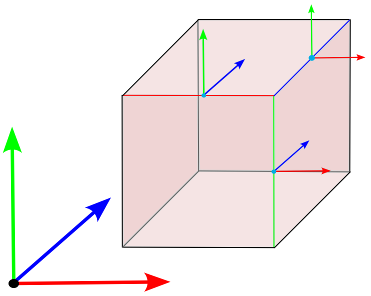
Lastly, Parreiras [2022] proposed an important modification to the MAC grid. The cell edges are also used to store vector and tensor quantities as shown in Figure 2. This modification results in three significant contributions:

- A suitable interpolation accuracy from the grid to the particles and vice-versa. The 12 cell edges are considered on all interpolations. Therefore, there are 19 points of reference considered for this operation: the cell center, six faces, and twelve edges;

- Non-aligned Tensors can be used without further numerical treatment. Tensors that have badly conditioned matrices or are not aligned with one of the three Cartesian coordinate system axes were difficult to solve without considering the cell edges.

- Computational stability due to more numerical precision on the pressure correction step. Both the velocity correction and the T-Laplacian operator benefit over the edges reference points.

Figure 3 – Data Structures Stored on Edges.



Source: Created by the author.

It is worth emphasizing that, in our approach, cell faces contain information in a single direction. However, cell edges possess the advantage of storing information in two directions simultaneously, as depicted in Figure 3. As a result, the interpolation of information from edges significantly enhances accuracy compared to relying solely on reference points situated on faces.

# 3 PROPOSED METHOD

In this chapter, we introduce our novel contribution, which enables pressure segregation while considering time-dependent tensor fields. This formal approach addresses the challenge of incorporating the temporal variations of the tensor field into the pressure segregation process, enhancing the fidelity of the simulations. Next, in Section 3.1, we present our mathematical model. In Section 3.2, we describe our numerical model and algorithm implementation. Finally, in Section *3.2.2.2* we discuss a discretization in a custom MAC grid.

## 3.1 NAVIER-STOKES EQUATIONS MODULATED BY TENSOR FIELDS

In this work, we introduce a novel approach that utilizes a prediction-correction anisotropic pressure segregation method. This method is coupled with a time-dependent positive-definite second-order tensor field to solve the incompressible Navier-Stokes equations. The formal mathematical model, which serves as the foundation of our work, is presented in this section.

Our primary goal is to solve the model proposed by Renhe et al. [2019] that accounts for the anisotropy of a symmetric positive-definite second-order tensor field. The ensuing mathematical formulation is:

$$\rho\frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u}\cdot\nabla)\mathbf{u} = \mu\nabla\cdot(\mathbf{T}\nabla\mathbf{u}) - \mathbf{T}\nabla \mathrm{p} + \mathbf{f}, \tag{3.1}$$

restricted to

$$\nabla\cdot\mathbf{u} = 0. \tag{3.2}$$

The tensor field plays a crucial role in our fluid simulation as it not only influences diffusive forces but also affects the pressure gradient force.

In this context, our interest lies in the modulation of the surface forces that act on the fluid, which are represented on the right-hand side of Equation 3.1. Modulating the advection terms on the left-hand side of Equation 3.1 requires a new model. This modification requires an alteration to the left-hand side of Newton's Second Law, which falls beyond the scope of this work. However, previous studies, such as Renhe et al. [2019] and Parreiras [2022], have utilized tensor fields to modulate these advective forces. The focus of this work is to modulate the forces originating from pressure and viscous forces that appear on the right-hand side of Equation 3.1. Additionally, it is possible to model external forces $\mathbf{f}$ based on the tensor field, as is demonstrated in Chapter 4.

The key point of this work is how to deal with pressure segregation under the influence of a tensor field. Local variations in pressure can drive fluid motion, causing the fluid to accelerate or decelerate. Consequently, this process is a conversion of kinetic pressure to kinetic energy (velocity). Conversely, in incompressible fluid dynamics, spots with a non-null divergence of the velocity can result in the conversion of kinetic energy into kinetic pressure. Numerically, the conversion of velocity into pressure occurs in the projection step. The reverse is performed in the advection step, by the pressure gradient term. Our main interest is how to modulate both conversions through a second-order positive-definite tensor field. Of course, a naive approach as simply applying the tensor field to the pressure term in the NS equations is not a suitable solution. The conversions must be carefully performed to avoid the introduction of artifacts in the simulation and keep it stable. As shown by Renhe et al. [2019], and Parreiras et al. [2022], the tensor-modulated pressure can be discarded throughout the simulation and still reflect the influence of the tensor field. However, as shown in this work, a properly accumulated anisotropic pressure estimation can open several possibilities to modulate the fluid and simulate local effects.

In summary, this work proposes to use the tensor field to modulate the exchange between kinetic pressure and kinetic energy, and vice versa, through a tensor pressure segregation based on the anisotropic Helmholtz decomposition. In this work, it is admitted that the tensor field varies in time, which makes the problem even more challenging. This circumstance is discussed in the next section.

### 3.1.1 Anisotropic Helmholtz Projection and Varying Tensor Fields

Let us consider the anisotropic Helmholtz decomposition denoted by Equation 2.2. It shows how a vector field $\mathbf{u}\left(\mathbf{x}\right)$ can be decomposed as a scalar $\phi\left(\mathbf{x}\right)$ and a vector $\mathbf{u}'\left(\mathbf{x}\right)$ field, subject to the tensor field $\mathbf{T}\left(\mathbf{x}\right)$. Now suppose that the vector field smoothly varies in time as $\mathbf{u}\left(\mathbf{x},t\right)=\mathbf{u}_t$, and the tensor field smoothly varies as $\mathbf{T}\left(\mathbf{x},t\right)=\mathbf{T}_t$. The anisotropic Helmholtz decomposition at any time $t$ is defined as (Eq. 2.2 with $\mathbf{S}$=$\mathbf{I}$):

$$\begin{aligned}\mathbf{u}_t &= \nabla_{\mathbf{T}_t}\phi_t + \nabla\times\mathbf{a}_t,\\ \mathbf{u}_t &= \nabla_{\mathbf{T}_t}\phi_t + \mathbf{u}_t^{'},\\ \mathbf{u}_t^{'} &= \mathbf{u}_t - \nabla_{\mathbf{T}_t}\phi_t,\end{aligned} \tag{3.3}$$

where $\mathbf{u}_t^{'}$ is a divergence-free vector field concerning the standard coordinate system and is subject to the tensor field $\mathbf{T}_t$. It can be obtained by solving the $\mathbf{T}$-Poisson equation (Eq. 2.5):

$$\nabla\cdot\nabla_{\mathbf{T}_t}\phi_t = \nabla\cdot\mathbf{u}_t.$$

Assume that $\{\mathbf{u}_0^{'},\mathbf{u}_1^{'},\cdots,\mathbf{u}_n^{'}\}$ represents a divergence-free vector field finite sequence until an arbitrary $n$-th vector field. Each arbitrary vector field $\mathbf{u}_t$ is the addition of the previous divergence-free vector field $\mathbf{u}_{t-1}'$ to an arbitrary vector field $\delta\mathbf{u}_t$ as:

$$\mathbf{u}_t = \mathbf{u}_{t-1}^{'} + \delta\mathbf{u}_t, \tag{3.4}$$

whose anisotropic Helmholtz decomposition is:

$$\begin{aligned}\nabla \cdot \nabla_{\mathbf{T}_t} \delta\phi_t &= \nabla \cdot \left( \mathbf{u}^{'}_{t-1} + \delta \mathbf{u}_t \right), \\ \nabla \cdot \nabla_{\mathbf{T}_t} \delta\phi_t &= \nabla \cdot \delta \mathbf{u}_t, \end{aligned} \tag{3.5}$$

where $\delta\phi_t$ refers to the decomposition of the incremental vector field $\delta\mathbf{u}_t$. This equation means that we only need to decompose the increment $\delta\mathbf{u}_t = \mathbf{u}_t - \mathbf{u}'_{t-1}$ to find the divergence-free vector field $\mathbf{u}'_t = \mathbf{u}_t - \nabla_{\mathbf{T}_t}\delta\phi_t$ in the sequence. Therefore, the equation:

$$\mathbf{u}'_t = \mathbf{u}^{'}_{t-1} + \left[\delta\mathbf{u}_t - \nabla_{\mathbf{T}_t}\delta\phi_t\right], \tag{3.6}$$

is sufficient to define the whole sequence, given a divergence-free initial vector field $\mathbf{u}'_0$ such that $\mathbf{u}_0 = \nabla_{\mathbf{T}_0}\delta\phi_0 + \mathbf{u}'_0$.

Now, consider the corresponding scalar field sequence $\{\delta\phi_0, \delta\phi_1, \cdots, \delta\phi_n\}$ as defined by Equations 3.5 and 3.6. In this work, the main problem is to find the accumulated scalar field:

$$\begin{cases} \phi_n = f_{\mathbf{T}_0}(\delta\phi_0) + f_{\mathbf{T}_1}(\delta\phi_1) + \cdots + f_{\mathbf{T}_{n-1}}(\delta\phi_{n-1}) + \delta\phi_n, \\ \nabla_{\mathbf{T}_n}\phi_n \text{ is } \mathbf{T}_n\text{-solenoinal}, \end{cases} \tag{3.7}$$

i.e., our problem is to find operators $f_{\mathbf{T}_t}(\cdot)$ that transform the intermediary scalar fields $\delta\phi_t$, that is the solution for the $t$-th element subject to $\mathbf{T}_t$, into scalar fields $f_{\mathbf{T}_t}(\delta\phi_t)$ that are decomposition solutions subject to the last tensor field $\mathbf{T}_n$.

For example, suppose the tensor fields are constant in $t$ with $\mathbf{T}_i = \mathbf{T}$. The solution, in this case, is straightforward with the identity operator $f_{\mathbf{T}_t}(\delta\phi_t) = \delta\phi_t$, resulting in the accumulation $\phi_n = \delta\phi_0 + \delta\phi_1 + \cdots + \delta\phi_n$ and the $\mathbf{T}_n$-irrotational total gradient $\nabla_{\mathbf{T}}\phi_n = \nabla_{\mathbf{T}}\delta\phi_0 + \nabla_{\mathbf{T}}\delta\phi_1 + \cdots + \nabla_{\mathbf{T}}\delta\phi_n$. Since all decompositions in the series are subject to the same tensor field $\mathbf{T}$, they admit the application of the linear operator $\mathbf{T}$ directly into a simple summation of all incremental scalar field solutions.

However, if the tensor fields vary along the sequence, the solution of Equation 3.7 is not trivial. Equation 3.6 shows that all elements in the sequence are coupled by the anisotropic Helmholtz decomposition (Eq. 3.5). Changing one vector field in the sequence impacts the subsequent vector fields because any distinct tensor field gives a unique divergence-free solution, up to the addition of harmonic scalar fields. Thus, the requirement that all scalar fields in the sequence have to be in accordance with a specific tensor field (Eq. 3.7) results in a divergence-free vector field sequence that is different from that Equation 3.6 provides.

In the context of this work, the problem can be incrementally solved for each term of the sequence, avoiding finding the direct operators $f_{\mathbf{T}_i}(\cdot)$ of Equation 3.7. The challenge is, for any element $t$, to conform all previous solutions $\{\delta\phi_0, \delta\phi_1, \cdots, \delta\phi_{t-1}\}$ and the $t$-th vector field $\mathbf{u}_t$ to the current tensor field $\mathbf{T}_t$. There is a reason to solve Equation 3.7 with an incremental approach. The series calculation focuses on finding divergence-free vector fields $\mathbf{u}'_t$ by subtracting a $\mathbf{T}_t$-irrotational vector field $\nabla_{\mathbf{T}_t}\delta\phi_t$ from the arbitrary vector field $\mathbf{u}_t$. Thus, we only have to guarantee that the incremental addition:

$$\phi_t = g_{\mathbf{T}_{t-1}}(\phi_{t-1}) + \delta\phi_t, \tag{3.8}$$

provides a $\mathbf{T}_t$-irrotational vector field in the form $\nabla_{\mathbf{T}_t}\phi_t$. Of course, this scheme implies that the previous scalar field summation $\phi_{t-1}$ gives a $\mathbf{T}_{t-1}$-irrotational vector field in the form $\nabla_{\mathbf{T}_{t-1}}\phi_{t-1}$. The operator $g_{\mathbf{T}_{t-1}}(\cdot)$ is responsible for converting the accumulated scalar field $\phi_{t-1}$ into another one that is adjusted by the current tensor field $\mathbf{T}_t$. Equation 3.8 is easier to handle since it takes into account only two consecutive vector fields of the sequence. Nevertheless, the last term of the sequence $\mathbf{u}'_n$ is divergence-free with respect to the last tensor field $\mathbf{T}_n$ and the vector field $\nabla_{\mathbf{T}_n}\phi_n$ is $\mathbf{T}_n$-irrotational, satisfying both requirements from Equation 3.7.

The solution is to modify the Poisson equation that projects the arbitrary vector field $\delta\mathbf{u}_t$ (Eq. 3.5). Our solution obtains the accumulated scalar field $\phi_t$ directly from the decomposition. Also, it conforms to the previous $\mathbf{T}_{t-1}$-irrotational vector field $\nabla_{\mathbf{T}_{t-1}}\phi_{t-1}$ to the new tensor field $\mathbf{T}_t$. One of the main contributions of this work is following anisotropic Helmholtz decomposition that directly deals with the accumulated scalar fields:

$$\nabla \cdot \nabla_{\mathbf{T}_t}\phi_t = \nabla \cdot \left[\delta\mathbf{u}_t + \nabla_{\mathbf{T}_{t-1}}\phi_{t-1}\right], \tag{3.9}$$

by adding the $\mathbf{T}_t$-irrotational part of the vector field $\nabla_{\mathbf{T}_{t-1}}\phi_{t-1}$ to the $\mathbf{T}_t$-irrotational decomposition of $\delta\mathbf{u}_t$. The resulting divergence-free vector field is then:

$$\mathbf{u}'_t = \mathbf{u}'_{t-1} + \left[\delta\mathbf{u}_t + \nabla_{\mathbf{T}_{t-1}}\phi_{t-1} - \nabla_{\mathbf{T}_t}\phi_t\right]. \tag{3.10}$$

Indeed, the $\mathbf{T}_{t-1}$-irrotational vector field $\nabla_{\mathbf{T}_{t-1}}\phi_{t-1}$ is not necessarily $\mathbf{T}_t$-irrotational. Equation 3.9 provides the $\mathbf{T}_t$-irrotational vector field $\nabla_{\mathbf{T}_t}\phi_t$ which is the sum of the $\mathbf{T}_t$-irrotational parts of $\delta\mathbf{u}_t$ and $\nabla_{\mathbf{T}_{t-1}}\phi_{t-1}$ vector fields. Also, the resulting accumulation scalar field $\phi_t$ is the sum of $\delta\phi_t$, which corresponds to the decomposition of $\delta\mathbf{u}_t$, to the previous accumulation $\phi_{t-1}$ conformed to $\mathbf{T}_t$. Analogously, Equation 3.10 provides the divergence-free vector field $\mathbf{u}'_t$ which is the sum of the solenoidal parts of both $\mathbf{u}_t$ and $\nabla_{\mathbf{T}_{t-1}}\phi_{t-1}$ vector fields. As a consequence, the effect of every arbitrary vector field $\delta\mathbf{u}_t$ added in the sequence is preserved either in the last divergence-free vector field $\mathbf{u}'_n$ or in the last $\mathbf{T}_n$-irrotational vector field $\nabla_{\mathbf{T}_n}\phi_n$.

Equations 3.9 and 3.10 can be adapted to solve the anisotropic pressure segregation step in the problem of fluid simulation, as shown in the next section.

## 3.2 TENSOR X-IVAS: X-IVAS WITH TENSOR-BASED PRESSURE SEGREGATION

In this section, we present our numerical model, beginning with our advection scheme. At this stage, the explicit information of particle position and velocity is integrated in time. The description of our semi-implicit advection model is as follows:

$$\begin{cases} \mathbf{x}^{n+t} = \mathbf{x}^{n} + \int_{n}^{n+t} \mathbf{U}^{n}\left(\mathbf{x}^{\tau}\right) \mathrm{d}\tau, \\ \mathbf{U}_{p}^{n+t} = \mathbf{U}_{p}^{n} + \frac{1}{\rho} \int_{n}^{n+1} \mathbf{A}^{n}\left(\mathbf{x}^{t}\right) + \mathbf{F}^{n}\left(\mathbf{x}^{t}\right) \mathrm{dt}. \end{cases} \tag{3.11}$$

Expanding the acceleration field term $\mathbf{A}$:

$$\mathbf{A}^{n}\left(\mathbf{x}^{t}\right) = \mu\nabla \cdot \left[\mathbf{T}^{n}\nabla\mathbf{U}^{n}\left(\mathbf{x}^{t}\right)\right] + \delta\sigma\left(\mathbf{x}^{t}\right) - \mathbf{T}^{n}\nabla\mathrm{P}^{n}\left(\mathbf{x}^{t}\right) - \delta\mathrm{P}\left(\mathbf{x}^{t}\right),$$

where $\delta\sigma$ and $\delta\mathrm{P}$ are the implicit viscosity and pressure terms, respectively. Both of these terms require careful treatment to ensure a fully explicit model. The equations governing these terms are as follows:

$$\delta\sigma\left(\mathbf{x}^{t}\right) = \mu\nabla \cdot \mathbf{T}^{t}\nabla\left(\mathbf{U}^{t}\left(\mathbf{x}^{t}\right) - \mathbf{U}^{n}\left(\mathbf{x}^{t}\right)\right),$$

and

$$\delta\mathrm{P}\left(\mathbf{x}^{t}\right) = \mathbf{T}^{t}\nabla\mathrm{P}^{t}\left(\mathbf{x}^{t}\right) - \mathbf{T}^{n}\nabla\mathrm{P}^{n}\left(\mathbf{x}^{t}\right).$$

We plug the implicit terms out of the integration scheme and solve them separately. Once discretized, Equation 3.11 is numerically integrated. A simple integration method like Forward Euler, which is a first-order accurate finite difference integrator, can be expressed as:

$$\mathbf{q}^{n+1} = \mathbf{q}^{n} + \Delta t f\left(\mathbf{q}^{n}\right),$$

where $\Delta t$ is the simulation time-step. Here, $f\left(\mathbf{q}\right)$ represents the derivative of $\mathbf{q}$ concerning time, and $\mathbf{q}$ denotes a vector term as position and velocity. However, in our work, we employ the explicit three-stage third-order Runge-Kutta time integration scheme, which is considered to best capture the complexities of anisotropic particle movement on both the velocity and acceleration streamline. This scheme is denoted as:

$$\begin{cases} \mathbf{k}_{1} = f\left(\mathbf{q}_{n}\right), \\ \mathbf{k}_{2} = f\left(\mathbf{q}_{n} + \frac{1}{2}\Delta t\mathbf{q}_{1}\right), \\ \mathbf{k}_{3} = f\left(\mathbf{q}_{n} + \frac{3}{4}\Delta t\mathbf{q}_{2}\right), \\ \mathbf{q}_{n+1} = \mathbf{q}_{n} + \frac{2}{9}\left(\Delta t\mathbf{k}_{1}\right) + \frac{3}{9}\left(\Delta t\mathbf{k}_{2}\right) + \frac{4}{9}\left(\Delta t\mathbf{k}_{3}\right). \end{cases} \tag{3.12}$$

The time integration of the implicit terms is described by the following equation:

$$\int_{n}^{n+1} \left[\delta\sigma\left(\mathbf{x}^{t}\right) - \delta\mathrm{P}\left(\mathbf{x}^{t}\right)\right] dt \approx \frac{\Delta t}{\theta}\left[\delta\sigma\left(\mathbf{x}^{n+1}\right) - \delta\mathrm{P}\left(\mathbf{x}^{n+1}\right)\right],$$

where $\theta$ is a scalar that defines the integration scheme that will be used. If $\theta = 1$, the integration scheme is Forward Euler. If $\theta = 2$, the integration is a Backward Euler. In this work, we will use a first-order approximation, $\theta = 1$. Therefore, the implicit term delta will consider only the ending point of the integration streamline [Bridson, 2015].

The implicit terms with the first-order integration scheme will be expressed by the following equations:

$$\delta\sigma\left(\mathbf{x}^{n+1}\right) = \mu\nabla\cdot\left[\mathbf{T}^{n+1}\nabla\mathbf{U}_p^{n+1} - \mathbf{T}^{n}\nabla\mathbf{U}^{n}\left(\mathbf{x}^{n+1}\right)\right],$$

and

$$\delta\mathrm{P}\left(\mathbf{x}^{n+1}\right) = \mathbf{T}^{n+1}\nabla\mathrm{P}_p^{n+1} - \mathbf{T}^{n}\nabla\mathrm{P}^{n}\left(\mathbf{x}^{n+1}\right). \tag{3.13}$$

These terms are solved separately and will be referred to as separate terms. The new velocity integration model will be:

$$\mathbf{U}_p^{n+1} = \mathbf{U}^{n}\left(\mathbf{x}^{n+1}\right) + \hat{\hat{\mathbf{A}}}^{n}\left(\mathbf{x}^{n+1}\right) + \frac{1}{\rho}\int_{n}^{n+1}\hat{\mathbf{A}}^{n}\left(\mathbf{x}^{n+1}\right)dt, \tag{3.14}$$

where $\hat{\hat{\mathbf{A}}}$ and $\hat{\mathbf{A}}$ are the new terms for the implicit and explicit part of the acceleration, respectively. The acceleration implicit part is:

$$\hat{\hat{\mathbf{A}}}^{n}\left(\mathbf{x}^{n+1}\right) = \frac{\Delta t}{\rho}\left[\delta\sigma\left(\mathbf{x}^{n+1}\right) - \delta\mathrm{P}\left(\mathbf{x}^{n+1}\right)\right],$$

and the explicit part is:

$$\hat{\mathbf{A}}^{n}\left(\mathbf{x}^{n+1}\right) = \mu\nabla\cdot\left[\mathbf{T}^{n}\nabla\mathbf{U}^{n}\left(\mathbf{x}^{n+1}\right)\right] - \mathbf{T}^{n}\nabla\mathrm{P}^{n}\left(\mathbf{x}^{n+1}\right) + \mathbf{F}^{n}\left(\mathbf{x}^{n+1}\right). \tag{3.15}$$

The velocity integration of Equation 3.14 is divided into three steps. This division is required to solve the implicit viscosity and pressure terms independently. The first step predicts velocity by integrating the explicit acceleration term. In this step, the Runge-Kutta time integration scheme described by Equation 3.12 is used. This first velocity approximation is denoted as:

$$\hat{\mathbf{U}}_p^{n+1} = \mathbf{U}^{n}\left(\mathbf{x}^{n+1}\right) + \frac{1}{\rho}\int_{n}^{n+1}\hat{\mathbf{A}}^{n}\left(\mathbf{x}^{n+1}\right)dt, \tag{3.16}$$

where $\hat{\mathbf{U}}^{n+1}$ is the first velocity approximation term referred to as “advected velocity”. The second step will account for the effects of viscosity over the first velocity approximation. At this step, we obtain a second velocity approximation as:

$$\begin{cases}\left[\rho - \frac{\Delta t}{\theta}\left(\nabla\cdot\mu\mathbf{T}^{n+1}\nabla\right)\right]\delta\sigma_p^{n+1} = \rho\left[\hat{\mathbf{U}}_p^{n+1} - \mathbf{U}^{n}\left(\mathbf{x}^{n+1}\right)\right],\\ \hat{\hat{\mathbf{U}}}_p^{n+1} = \hat{\mathbf{U}}_p^{n+1} + \delta\sigma_p^{n+1},\end{cases} \tag{3.17}$$

where $\hat{\hat{\mathbf{U}}}$ is the viscosity-velocity prediction term.

Finally, the divergence-free velocity field will be obtained in the third step using the pressure correction method:

$$\mathbf{U}_p^{n+1} = \hat{\hat{\mathbf{U}}}_p^{n+1} - \frac{1}{\rho}\frac{\Delta t}{\theta}\delta\mathrm{P}\left(\mathbf{x}^{n+1}\right). \tag{3.18}$$

Applying the divergence operator at both sides of Equation 3.18, and considering the divergence-free restriction. The following **T**-Poisson equation is obtained:

$$\frac{\Delta t}{\theta}\frac{1}{\rho}\nabla\cdot\delta\mathrm{P}\left(\mathbf{x}^{n+1}\right)=\nabla\cdot\hat{\hat{\mathbf{U}}}_{p}^{n+1}. \tag{3.19}$$

Expanding this **T**-Poisson equation with Equation 3.13, we have:

$$\nabla\cdot\left[\mathbf{T}^{n+1}\nabla\mathrm{P}_{p}^{n+1}\right]=\nabla\cdot\left[\frac{\theta\rho}{\Delta t}\hat{\hat{\mathbf{U}}}_{p}^{n+1}+\mathbf{T}^{n}\nabla\mathrm{P}^{n}\left(\mathbf{x}^{n+1}\right)\right], \tag{3.20}$$

as proposed in Equation 3.9. We were only able to achieve this using the contribution presented in Section **3.1.1**. Finally, the divergence-free velocity field is obtained as:

$$\begin{cases}\mathbf{U}_{p}^{n+1}=\hat{\hat{\mathbf{U}}}_{p}^{n+1}+\frac{1}{\rho}\frac{\Delta t}{\theta}\left[\mathbf{T}^{n}\nabla\mathrm{P}^{n}\left(\mathbf{x}^{n+1}\right)-\mathbf{T}^{n+1}\nabla\mathrm{P}_{p}^{n+1}\right],\\ \nabla\cdot\mathbf{U}^{n+1}=0.\end{cases} \tag{3.21}$$

as proposed in Equation 3.10. The accumulation of pressure within the solution of the Poisson equation is an important aspect, as discussed in Section **3.1.1**. Indeed, Equations 3.20 and 3.21 convert kinetic energy into kinetic pressure and vice versa. After each iteration: 1) the pressure gradient field $\mathbf{T}^{n+1}\nabla\mathrm{P}^{n+1}$ is always $\mathbf{T}^{n+1}$-irrotational; 2) the divergence-free velocity field $\mathbf{U}_{p}^{n+1}$ is **I**-solenoidal with the influence of $\mathbf{T}^{n+1}$ that modulated both the incremental velocity vector field $\hat{\hat{\mathbf{U}}}_{p}^{n+1}$ and the previous accumulated pressure $\mathbf{T}^{n}\nabla\mathrm{P}^{n}\left(\mathbf{x}^{n+1}\right)$.

### 3.2.1 Boundary Conditions

In this work, special attention is given to two types of boundaries: the free-surface boundary and the solid wall boundary. The free-surface boundary represents the interface between the fluid and the surrounding empty space, while the solid wall boundary represents the interface between the fluid and a solid object.

The enforcement of boundary conditions is achieved within the framework of the MAC grid, similar to a fully Eulerian solver. The evaluation of the interface is based on the information contained within the neighboring cells. In our case, the boundaries occur at the faces and edges of each grid cell. The boundary conditions are enforced following the advection step and before the projection step. This adjustment is applied to the RHS of Equation 3.20. It directly affects the velocity divergence that is taken into account during the projection step, consequently modifying the linear system of equations derived from the **T**-Poisson equation.

#### *3.2.1.1 Free-surface boundary*

Since we are simulating a single-phase fluid, the treatment of the boundary is relatively straightforward. In this scenario, the fluid either interacts with a vacuum or with another fluid of much lower density, whose influence can be considered negligible. In this work, a Dirichlet boundary condition is employed at this interface, setting both the pressure and pressure gradient fields to zero.

#### *3.2.1.2 Solid Walls boundary*

By considering all reference points associated with the faces and edges of a cell, diverse fluid-solid interface configurations can be observed. Following the approach outlined in Parreiras [2022], the solid wall boundary condition is applied when a cell is classified as fluid and has at least two neighboring cells that are solid. In accordance with the no-slip condition, the fluid velocity on this interface assumes the velocity of the solid. In such cases, the quantities (e.g., velocity) associated with the face or edge of the grid cell intersecting the interface are set to solid velocity with respect to the interface's normal direction.

In addition to the right-hand side of Eq. (3.20), the no-slip condition must also be enforced on the pressure gradient and acceleration fields. This reinforcement takes place after the projection step and has a direct impact on the particle advection step.

### 3.2.2 Simulation Steps

In this section, we present the proposed method as an iterative algorithm within a fluid simulation environment. Except for the particle advection step, all other steps are solved within the grid context. The algorithm is summarized by the following steps:

1. Seed fluid particles to the environment;
2. Advect particles with third order Runge-Kutta (Eq. 3.12): $\mathbf{x}^{n+1}, \hat{\mathbf{u}}^{n+1}$;
3. Interpolate particle advected velocity to grid: $\hat{\mathbf{u}}^{n+1} \Rightarrow \hat{\mathbf{U}}^{n+1}$;
4. Compute viscosity-velocity (Eq. 3.17): $\hat{\hat{\mathbf{U}}}^{n+1}$;
5. Set no-slip condition to the viscosity-velocity grid: $\hat{\hat{\mathbf{U}}}^{n+1} = \hat{\hat{\mathbf{U}}}^{n+1}_{solid} = 0$;
6. Compute force field: $\mathbf{F}^{n+1}$;
7. Compute tensor field: $\mathbf{T}^{n+1}$;
8. Compute projection field $\mathbf{R}$ (RHS of Equation 3.20);
9. Set no-slip condition on projection field grid: $\mathbf{R} = \mathbf{R}_{solid} = 0$;
10. Compute divergence-free velocity field and accumulate pressure (Eq. 3.21): $\mathbf{U}^{n+1}, \mathrm{P}^{n+1}$;
11. Update pressure gradient field: $\mathbf{T}^{n+1}\nabla \mathrm{P}^{n+1}$;

12. Update acceleration field: $\mathbf{A}^{n+1}$;
13. Update particles velocity (Eq. 3.22): $\mathbf{U}^{n+1} \Rightarrow \mathbf{u}^{n+1}$;

During the first interaction, the initial velocity vector field is projected, after the seeding of particles and before the advection process. This ensures the utilization of a divergence-free field during the advection step. Also, the initial pressure is the scalar field resulting from the Anisotropic Helmholtz Decomposition. To achieve this, the following procedures are executed:

1. Initialize the force field;
2. Initialize the acceleration field with only the body forces information;
3. Set the no-slip condition at the initial velocity field;
4. Project the initial vector field.

#### *3.2.2.1 Particle Velocity Update*

In step 12, the grid velocity information is interpolated to the particles. We propose a new particle velocity update scheme. The new particle velocity update combines our Tensor X-IVAS method with the FLIP updating scheme, resulting in the following equation:

$$\mathbf{u}^{n+1} = \mathbf{u}_p^{n+1} + (1-\alpha)\left(\hat{\mathbf{u}}^{n+1} - \tilde{\hat{\mathbf{u}}}_p^{n+1}\right), \tag{3.22}$$

where $\mathbf{u}_p^{n+1}$ is the interpolated divergence-free velocity obtained from the projection in step 9. The $\hat{\mathbf{u}}^{n+1}$ term is the first particle velocity approximation obtained from step 2. $\tilde{\hat{\mathbf{u}}}_p^{n+1}$ is the particle velocity approximation before the projection step, without no-slip condition. The $\alpha$ term is a scalar parameter that can assume a value in the interval of $\alpha \in [0, 1]$. The particle velocity update will change as $\alpha$ changes. If $\alpha = 1$, the velocity update will be a PIC update scheme. If $\alpha < 1$, the new update term increments the particle velocity. This results in an advection with even less numerical diffusion.

#### *3.2.2.2 Implementation on an Extended MAC Grid*

Our Tensor X-IVAS method is discretized on an extended 3D staggered MAC grid, which was proposed by Parreiras [2022]. It provides a higher level of accuracy for solving the anisotropic $\mathbf{T}$-Poisson (Eq. 2.5) problem, coping well with tensors with eigensystem not aligned with the main axes. In this version, the vector quantities are stored in the faces and the edges of each voxel cell. The scalar and tensor quantities are stored at the cell centers. Tensor quantities are interpolated to the faces and edges in the extended gradient operator needed to solve the $\mathbf{T}$-Poisson (Eq. 2.5).

The use of the extended MAC grid was a choice that helped a faster development during our research. The GCG-Group for Computer Graphics, Image and Vision has worked on the problem of anisotropic fluid simulation in finite volumes since 2014. However, our method can be easily adapted to finite elements, which is the case for most works based on the X-IVAS method.

# 4 EXPERIMENTAL RESULTS AND DISCUSSION

In this chapter, we present qualitative experimental results obtained with the proposed method. The experiments were divided into two main categories: 1) experiments with a constant tensor field; 2) experiments with time-dependent tensor fields. The idea of this chapter is to present different configurations of tensor fields that show how they influence the fluid flow. In this way, we can analyze the result from each case, indicating how to obtain different flow effects with tensor fields.

The fluid density $\rho$, the time step $\Delta t$, and the cell width $\Delta x$ remained constant throughout all experiments. They were assigned values of $\rho = 1\,(kg/m^3)$, $\Delta t = 1/60\,(s)$, and $\Delta x = 1\,(m)$, respectively. Gravity exerted a force with a magnitude of $10\,(m/s^2)$ in the negative z-axis direction, except in Section **4.2.2**, where gravity was directed towards the center of the domain. Although our method is capable of modulating both viscosity and pressure terms, to evaluate the impact of pressure modulation on fluid flow, the viscosity was set to zero in Sections 4.1 and 4.2. This avoids the interference of non-null viscosity in the analysis of our method's ability to anisotropically modulate the fluid flow pressure terms.

Our domain space was discretized into a grid with a resolution of $60 \times 60 \times 60$ cells. An additional layer of cells, labeled as solids, surrounded the domain space. The no-slip condition was applied in all experiments, as discussed in Section **3.2.1**. Without loss of generalization, stationary solid walls were considered throughout all simulations. More detailed descriptions and specific setups of each experiment are provided in the corresponding sections.

To create the final rendered experiments, various third-party software tools were utilized. Firstly, the simulation data was acquired using our OpenGL/C++ fluid simulation environment. Then, we generated the fluid simulation meshes using Houdini [SideFX®, 2021], and Paraview [Sandia National Labs and Labs., 2021] software. Finally, we rendered each experiment using Blender [Foundation, 2021]. We have used Blender default render engine named “Cycles”, and a third-party open-source engine named “LuxCoreRender” [LuxCoreRender®, 2018] for rendering. Both these engines are physically-based path-tracing used for production rendering. Each experiment results in a video with 30 frames per second (FPS) during 20 seconds.

The experiments were performed on different hardware setups. Specifically, the simulation was executed on an Intel Xeon E5-4607 @ 2.20GHz processor with 32GB RAM DDR3. The mesh generation was performed on an AMD Ryzen 3 3100 @ 3.60GHz processor with 16GB RAM DDR4 and a NVIDIA GTX 1650S GPU. Finally, the simulations were rendered using one Intel i5 10400 CPU @ 2.90GHz with 16GB RAM DDR4 and NVIDIA RTX 3060 GPU, and one Intel i5 10400 CPU @ 2.90GHz with 8GB RAM DDR4 and

NVIDIA GTX 1660 GPU.

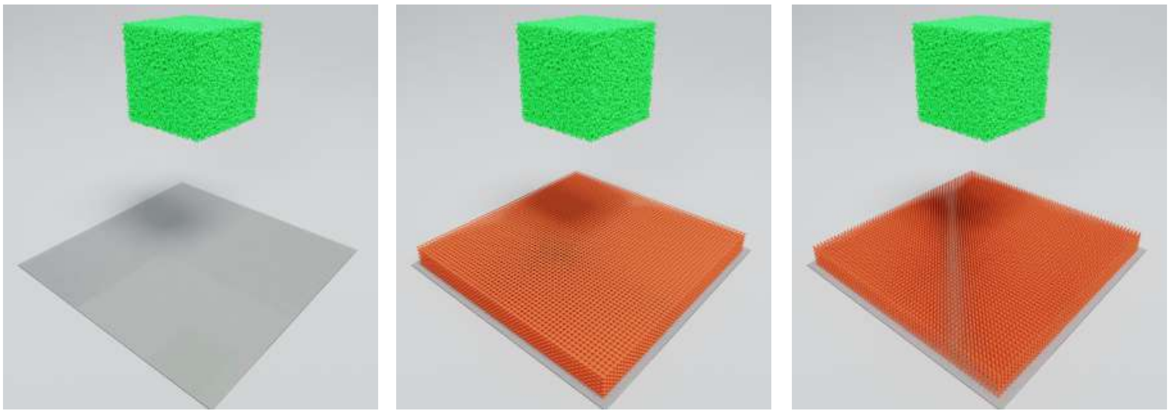
## 4.1 CONSTANT TENSOR FIELDS

All tensor fields considered for the following experiments are constant in time. Each section is structured to present the intrinsic characteristics of the tensor field, initial conditions, environment setup, and the fluid source and position.

### 4.1.1 Simulation with Axis-Aligned Planar Tensors

In this subsection, the three presented experiments involve different tensor fields. The first experiment utilizes an isotropic tensor field, i.e. the simulation is the same as the standard X-IVAS method. The second experiment employs a planar tensor field at the bottom, limiting the pressure exchange in a plane parallel to the $xy$-plane. Lastly, the third experiment incorporates a linear field at the bottom, enhancing the pressure exchange. These tensor fields modulate the pressure exchange mainly in the direction of the $z$-axis. The experiment with the isotropic tensor field is used as a reference to analyze the anisotropic effects of the other two different tensor fields.

Figure 4 – Initial setup of the falling cube simulations.



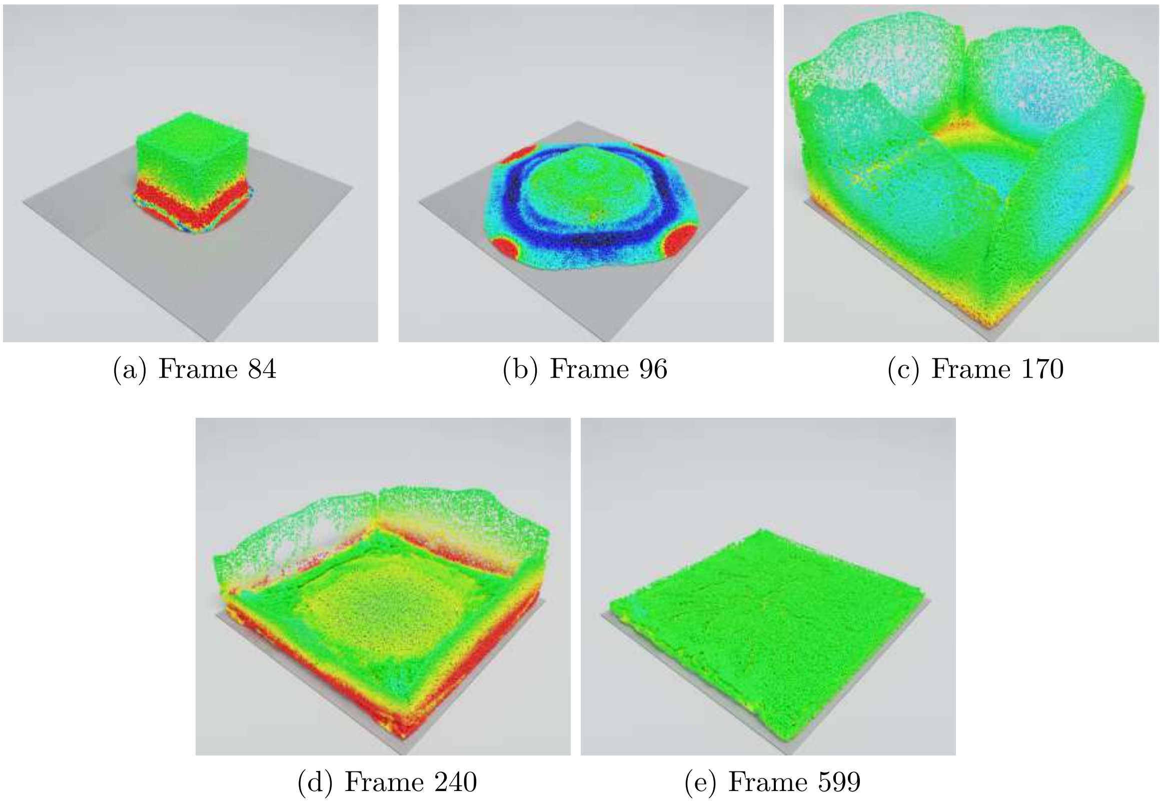
(a) Isotropic tensor field (b) Planar tensor field (c) Linear tensor field

Source: Created by the author.

The fluid source is a block of fluid with dimensions of $20 \times 20 \times 20$, resulting in a total of 8.000 voxels. Each voxel starts with 20 particles, leading to a total of 160.000 particles in the simulation. This block of fluid is initially at rest ($\mathbf{u}_0 = \mathbf{0}$) and is inserted at the top of the simulation domain. Its height $k$, goes from grid cell 38 (inclusive) to 58 (exclusive), where $k$ represent the grid $z$-axis coordinate. Also, this block of fluid is slightly displaced from the center, with $i, j \in [21, 40]$, where $i, j$ are the $x$ and $y$ grid coordinates respectively.

We apply the particle velocity update presented in Equation 3.22 with $\alpha = 1.0$, meaning that the simulation is less turbulent and more dissipative.

Figure 5 – Falling cube simulation with the isotropic tensor field.



(a) Frame 84

(b) Frame 96

(c) Frame 170

(d) Frame 240

(e) Frame 599

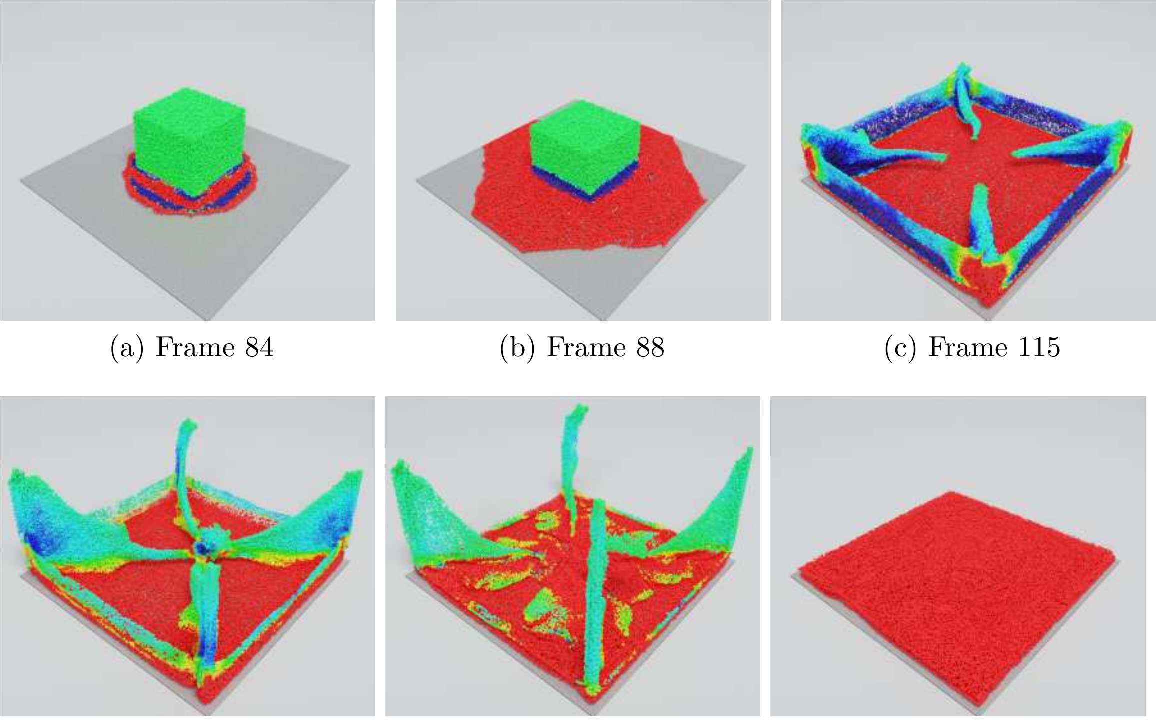
Source: Created by the author.

Except for the isotropic tensor field, the employed tensor field is composed of six layers and positioned at the bottom of the domain. All other tensors in the volume are isotropic. It means that the fluid is under the effect of anisotropy only in the bottom six layers. The main difference between the three simulations lies in the eigenvalues: $\lambda_1 = \lambda_2 = \lambda_3 = 1$ for the first experiment, $\lambda_1 = \lambda_2 = 1$, $\lambda_3 = 0.01$ for the second experiment (planar tensors), and $\lambda_1 = \lambda_2 = 1$, $\lambda_3 = 1.99$ for the third experiment (linear tensors). All eigenvectors are axis-aligned with: $\mathbf{e}_1$ and $\mathbf{e}_2$ aligned with the $xy$-axis (planar tensor), and $\mathbf{e}_3$ aligned with the $z$-axis (linear tensor).

The range of colors shows the pressure magnitude and goes from the lowest values being represented as dark blue colors and the highest ones being represented as bright red. These values range from $-100\,(\mathrm{Pa})$ to $100\,(\mathrm{Pa})$ The values outside this interval are clamped and, consequently, dark blue and bright red particles may have values arbitrarily beyond it.

The experiment with a planar tensor at the bottom with $\lambda_3 = 0.01$, shows how we can use the tensor field to induce pressure over the fluid as shown in frame 599 of Figure 6. This high pressure can be visualized by the red particle colors. This induced

Figure 6 – Falling cube simulation with the planar tensor field.



(a) Frame 84

(b) Frame 88

(c) Frame 115

(d) Frame 140

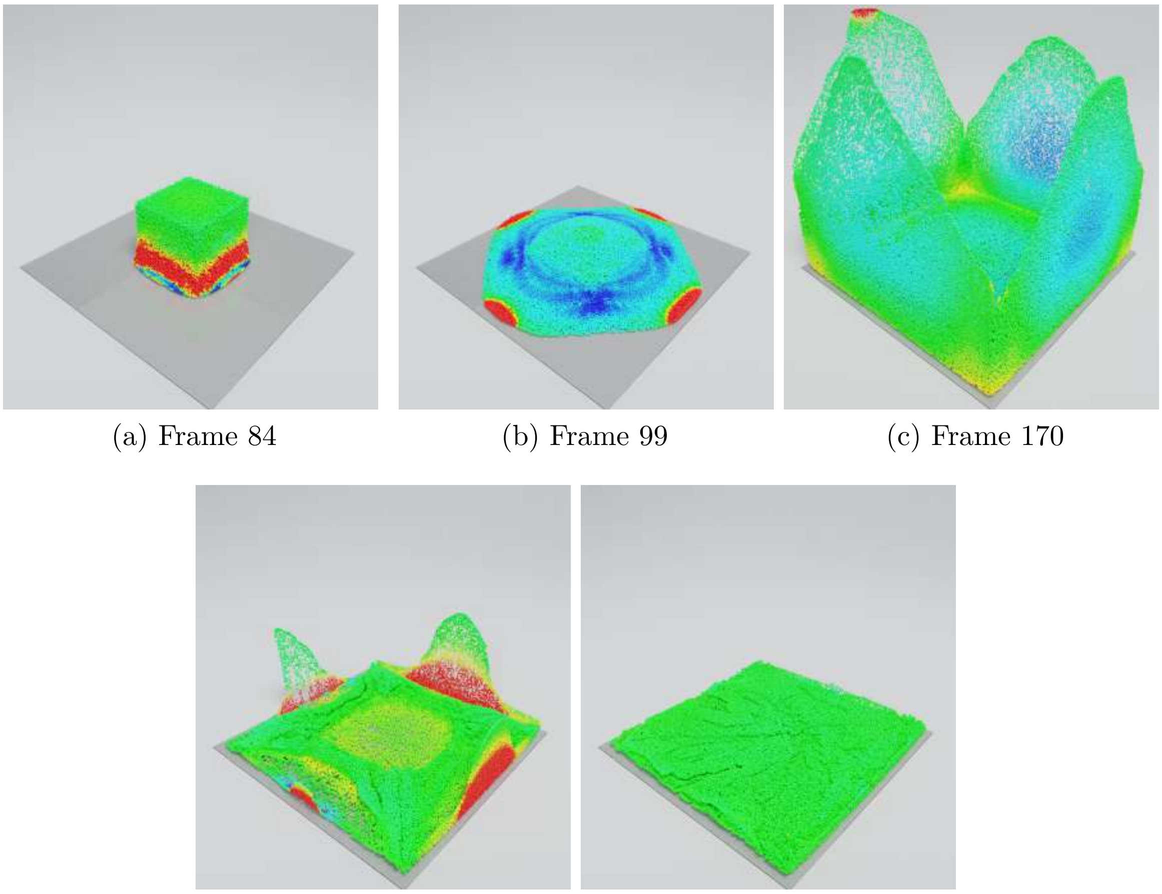
(e) Frame 170

(f) Frame 599

Source: Created by the author.

pressure can be used for containing the fluid in an arbitrary location. When the block of fluid comes into contact with the bottom of the domain, it rapidly propagates, attempting to alleviate the pressure in the $xy$ plane, as depicted in frame 92 of Figure 6 and the experiment video. The fluid accumulates at the border of the domain and eventually rises in the $z$-axis direction. However, due to insufficient velocity, it forms horizontal columns of fluid, as illustrated in frame 172 of Figure 6. These columns and the fluid that reach the side wall do not reach a high altitude as the isotropic experiment, due to the tensor field constraint. The resulting fluid columns are directed towards the center of the domain, as the tensor field limits pressure exchange in the $z$-axis direction and favor the $xy$ plane main axis directions.

Figure 7 – Falling cube simulation with the linear tensor field.



(a) Frame 84

(b) Frame 99

(c) Frame 170

(d) Frame 300

(e) Frame 599

Source: Created by the author.

On the other side, the experiment with the linear tensor field with $\lambda_3 = 1.99$, shows how we can use the tensor field to favor the pressure exchange, and consequently facilitate the fluid motion in an arbitrary direction. As depicted by the particle colors in Figure 7, the pressure of the fluid on the bottom of the domain does not achieve a high pressure as the isotropic or the planar tensor field experiments. That is due to the fact that the tensor field does not limit the pressure exchange in any direction and, more than that, enhances the exchange on the $z$-axis direction. This enhancement can be seen in frame 172 of Figure 7, as the fluid reached the ceiling of the domain. The experiment with the isotropic tensor field and planar tensor does not achieve such altitude.

The graphs depicted in the following figures display, for each variation of this experiment, the integration of the Bernoulli equation (Equation 4.1), the integration of the Bernoulli equation without pressure energy, and the divergence of velocity. The Bernoulli equation is a fundamental principle in fluid dynamics that describes the conservation of energy along a streamline in an inviscid, incompressible flow. It states that the sum of the kinetic energy, potential energy, and pressure energy per unit mass remains constant along a streamline. The Bernoulli equation can be expressed as:

$$\frac{\mathbf{u}^2}{2} + \frac{\mathrm{p}}{\rho} + \mathbf{g}\mathrm{h} = \text{constant} \tag{4.1}$$

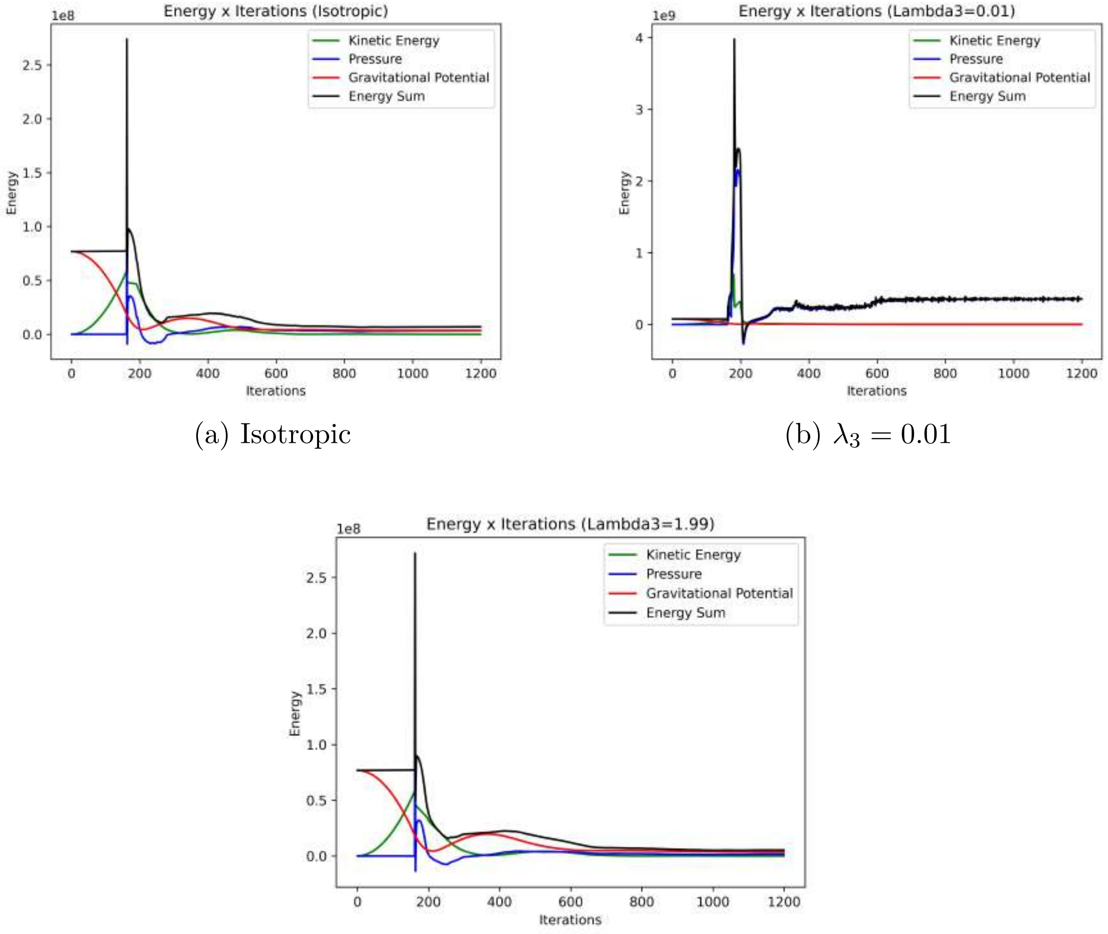
Where p is the pressure at a given point in the fluid, $\rho$ is the density of the fluid, **u** is the velocity of the fluid, **g** is the acceleration due to gravity, and h is the height above a reference point.

The conservation of energy principle underlying the Bernoulli equation implies that as fluid flows along a streamline, the total energy per unit mass remains constant. This means that if the velocity of the fluid increases (increasing kinetic energy), the pressure must decrease to maintain energy conservation, and vice versa. Similarly, changes in elevation lead to changes in potential energy, which are compensated for by corresponding changes in pressure and velocity to ensure energy conservation. Overall, the Bernoulli equation serves as a powerful tool for analyzing and predicting the behavior of fluid flow systems by quantifying the interplay between kinetic, potential, and pressure energies along streamlines.

It can be observed in Figure 8 that the impact of pressure energy on integration in the experiment with a planar field ($\lambda_3 = 0.01$) is of much greater magnitude than kinetic energy. Additionally, the magnitude increases by a factor of 10 comparing the experiment with the planar tensor with the other ones. It can also be observed that the sum of the system's energy with the planar tensors tends toward the value of pressure energy, given that it dominates the experiment energies. It's worth noting that the planar tensors generate an artificial pressure in the fluid volume as they limit the exchange with the z-axis.

The Bernoulli equation holds true, and energy conservation is upheld until the fluid volume collides with the bottom of the domain, which occurs just before iteration 200. From the moment the collision occurs, energy conservation no longer holds. The sum of kinetic energy, gravitational potential, and pressure remains constant before the collision, as we can see in Figure 8. After the inelastic collision with the bottom of the domain, this conservation does not hold.

Figure 8 – Constant Axis-Aligned Tensors Bernoulli Equation Integration



(a) Isotropic

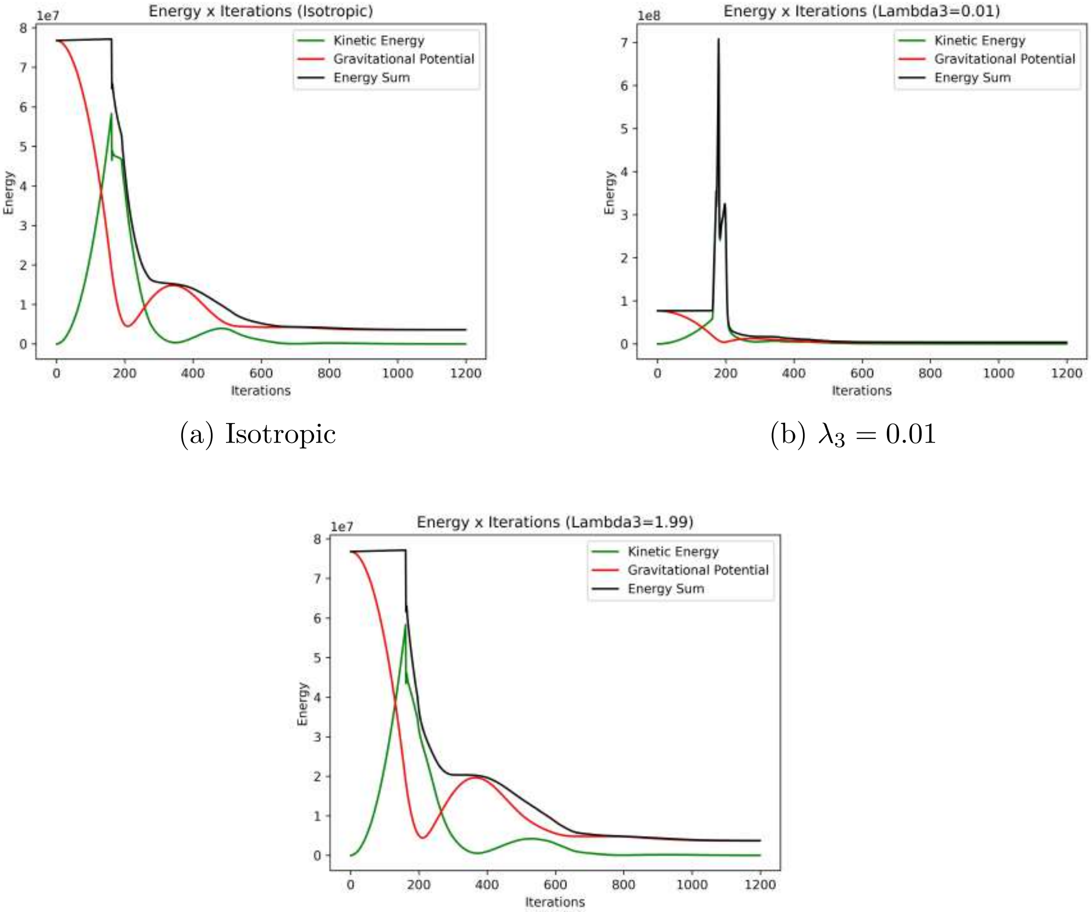
(b) $\lambda_3 = 0.01$

(c) $\lambda_3 = 1.99$

Source: Created by the author.

The integration of the Bernoulli equation without the pressure energy depicted in Figure 9, we can observe how velocity is affected in the experiment with planar tensors in the figure. After the collision, the energy increases as pressure relief is extremely limited. Kinetic energy peaks upon collision with the bottom of the domain and then again when the fluid strikes the domain walls, and then the energy decreases more rapidly with planar tensors than in other experiments.

Figure 9 – Axis-Aligned Planar Tensors Bernoulli Equation Integration without Pressure



(a) Isotropic

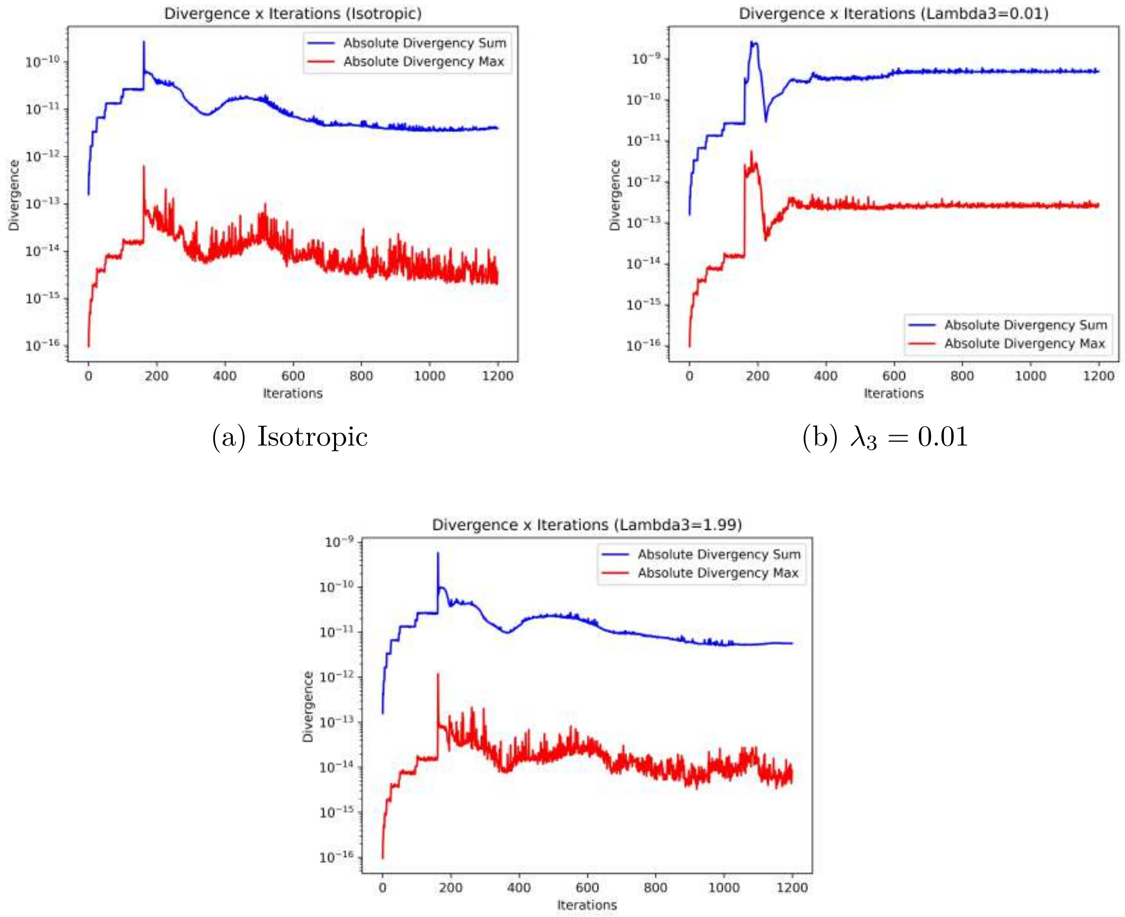
(b) $\lambda_3 = 0.01$

(c) $\lambda_3 = 1.99$

Source: Created by the author.

Finally, in Figure 10, we can see the impact on the divergence. Once again, the field with planar tensors exhibits a different behavior from the other experiments. It is noticeable that despite the increase in the magnitude of the divergence, the standard deviation of the mean is lower compared to the other two experiments. The experiment with linear and isotropic tensors exhibits a similar behavior regarding system energies in all analyzed cases.

Figure 10 – Axis-Aligned Planar Tensors Speed Divergence



(a) Isotropic

(b) $\lambda_3 = 0.01$

(c) $\lambda_3 = 1.99$

Source: Created by the author.

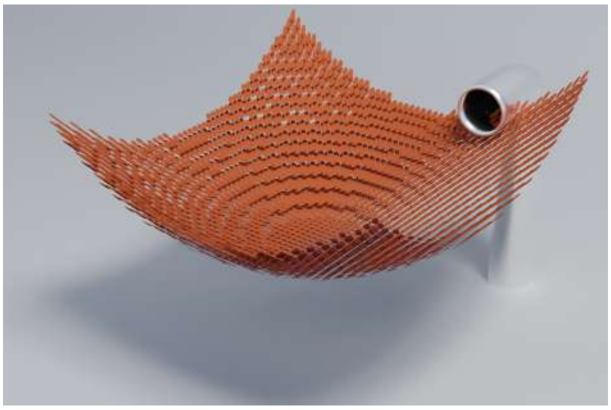
### 4.1.2 Simulation with non Axis-Aligned Planar Tensors

The previous experiments consisted of tensors whose eigenvectors are aligned to the grid coordinate system. In this section, the simulation exemplifies a tensor field with non-axis-aligned planar tensors representing a surface with variable curvature. The main goal is to have the effect of a permeable medium. The misaligned tensors allow the fluid to be exchanged through 26-nearest cell neighbors during the anisotropic projection. In contrast, axis-aligned tensor fields primarily exchange fluid through the six faces of a cell.

Our method is mainly focused on the pressure velocity adjustment along the simulation steps. It needs pressure changes to affect the fluid flow. Thus, it is not fully equipped to simulate porous transport. However, in Section 3.1, we presented the possibility of designing body forces according to the current attributes of the tensor field. This can be explored to get permeability effects.

We have devised a body force that is directly proportional to the magnitude of

Figure 11 – Initial setup of the permeable surface simulation.



(a)

Source: Created by the author.

the tensor field divergence and the velocity of the fluid flow. This force is integrated into Equation 3.15. The newly introduced term $F$, added to the grid in every step, is defined as follows:

$$\mathbf{F}^n\left(\mathbf{x}^{n+1}\right) = \mathbf{G}^n\left(\mathbf{x}^{n+1}\right) + \mathbf{R}^n\left(\mathbf{x}^{n+1}\right),$$

where $\mathbf{G}$ is the gravity force, and $\mathbf{R}$ is the novel proposed body force. The relation between the tensor field divergence and the fluid flow velocity is:

$$\frac{du}{dt} = -\psi\mathbf{T}^{-1}\left[\left(\mathbf{u}\cdot\mathbf{D}\right)\mathbf{D}\right], \tag{4.2}$$

where $\mathbf{D} = \nabla\cdot\mathbf{T}$ is a vector quantity stored at the grid. Notice that $\psi$ has $m^2/s$ unit, which is the same as the kinematic viscosity $\nu$. Expanding Equation 4.3 by the Euler method, we obtain:

$$\frac{\mathbf{u}^{n+t}-\mathbf{u}^n}{\Delta t} = -\psi\mathbf{T}^{-1}\left(\mathbf{u}^{n+t}\cdot\mathbf{D}\right)\mathbf{D},$$

where $\mathbf{u}^{n+t}$ and $\mathbf{u}^n$ are the next and previous velocities of the particles. As we need the velocity that is deviated by this tensor force, we rearrange this equation as:

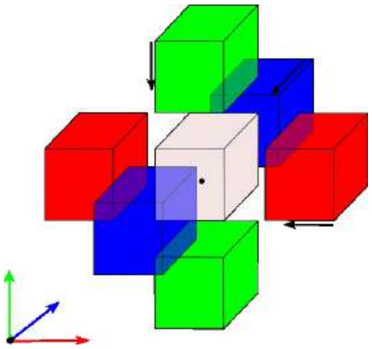
$$\mathbf{u}^{n+t} = \mathbf{u}^n\left[\mathbf{I}+\psi\mathbf{T}^{-1}\left(\mathbf{D}\mathbf{D}^T\right)\Delta t\right]^{-1}. \tag{4.3}$$

Notice that the force redirects the fluid flow in the direction of the vector field obtained from the tensor field divergence. Additionally, the resulting velocity $\mathbf{u}^{n+t}$ exhibits an inverse proportionality to the local tensor divergence magnitude. As such, the design of the tensor field is crucial to determining the influence of the force on the fluid flow. In the scenario where $\mathbf{D} = \mathbf{0}$, the resulting velocity aligns with the current velocity $\mathbf{u}^n$. Consequently, we can regulate the magnitude of this force by adjusting the magnitude of the tensor field divergence.

Numerically, the deviated velocity of Equation 4.3 is calculated immediately after the initial approximation of the velocity field. This force is integrated in time using a

Forward Euler method. Therefore, it is not coupled to the three-stage third-order Runge-Kutta used on the advection step. Regarding the discretization of the tensor divergence, it is stored at the center of the MAC grid cell and computed according to Figure 12. Only the two adjacent cell neighbors along each axis are taken into consideration. The calculation proceeds from the positive to the negative direction, as indicated by the black arrows in Figure 12.

Figure 12 – Tensor field divergence scheme.



(a)

Source: Created by the author.

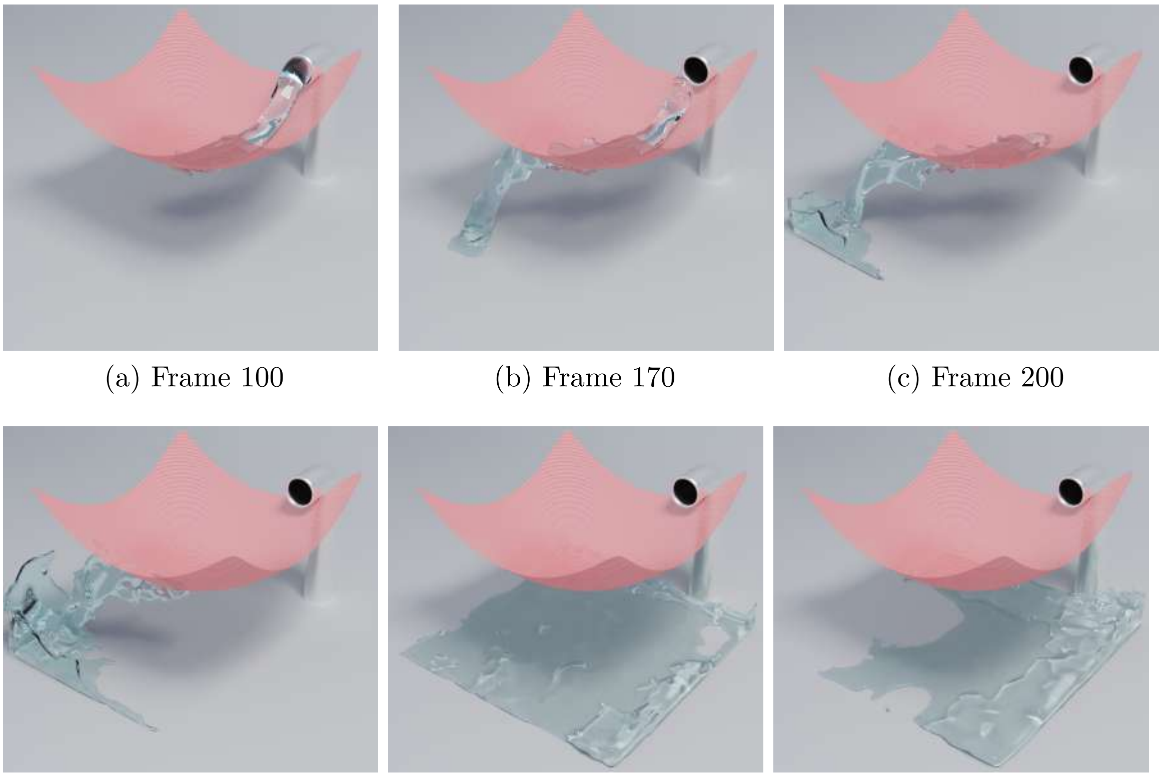
We present three experiments to demonstrate the use of the proposed tensor force. Each simulation varies the divergence of the tensor field. Specifically, we created three sets of eigenvalues for each simulation. All configurations employ two layers of planar tensors with eigenvalues of $\lambda_1 = \lambda_2 = 1$, while the values of $\lambda_3$ differ. The first simulation adopts $\lambda_3 = 0.01$, and its progression is illustrated in Figure 13. The second experiment, depicted in Figure 14, employs $\lambda_3 = 10^{-3}$. Finally, the third simulation incorporates $\lambda_3$ set to $10^{-4}$, and its depiction is presented in Figure 15.

To construct the tensor field, we utilize non-aligned planar tensors generated through the following parametric parabolic equation:

$$\begin{cases} x = i,\ i \in [6, 54]\,, \\ y = j,\ j \in [6, 54]\,, \\ z = c\left(\frac{i^2}{a^2} + \frac{j^2}{b^2}\right) + k,\ k \in [19, 20]\,. \end{cases}$$

The terms $i$, $j$, and $k$ represent grid coordinates corresponding to the $x$, $y$, and $z$ axis, respectively. The parameters $a$ and $b$ have a value of 20, while $c$ is set to 7 to give the tensor field the cloth shape that is shown in Figure 11. The tensor field is visually represented by a pinkish color "cloth" texture in the simulation figures.

Figure 13 – Permeable surface simulation with $\lambda_3 = 10^{-2}$.



(a) Frame 100

(b) Frame 170

(c) Frame 200

(d) Frame 240

(e) Frame 480

(f) Frame 599

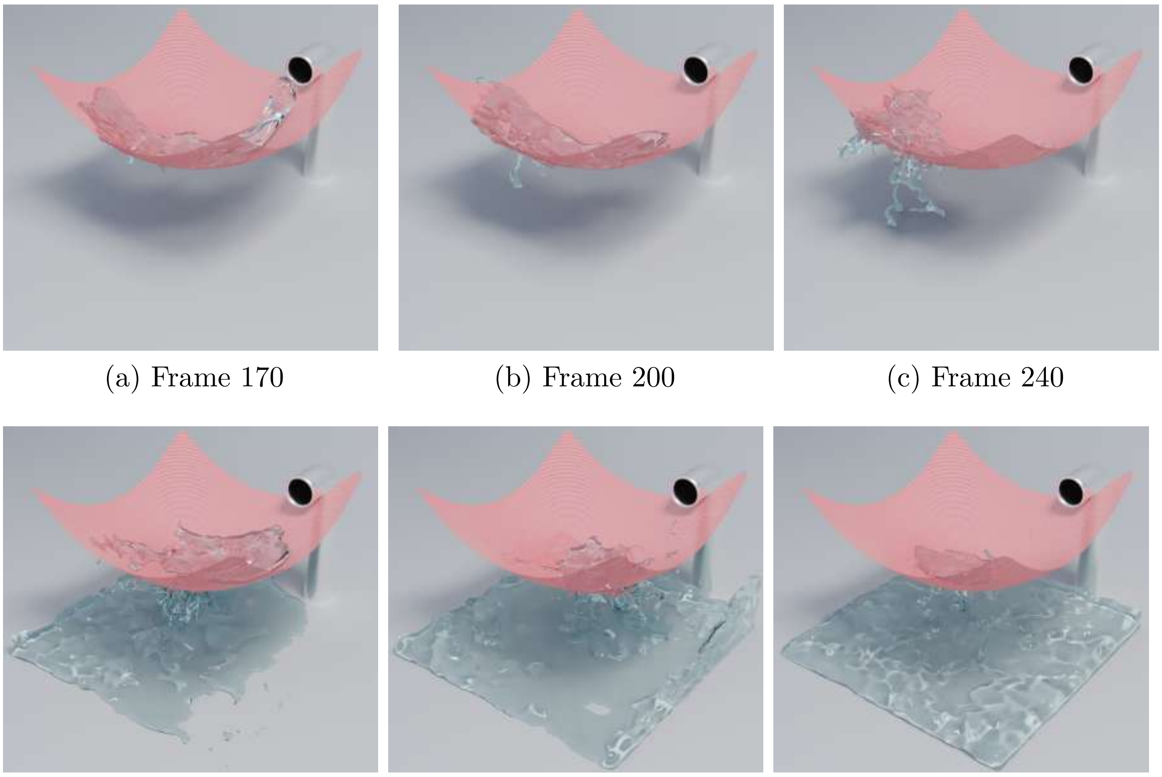
Source: Created by the author.

The fluid is inserted by a circular shape source with radius 3 of voxel and positioned at the right side of our domain. It is visually depicted as a metallic pipe in the simulation figures. This font emits a constant stream of particles, with a rate of $50,000$ particles per second for five seconds. Consequently, a total of $250,000$ particles are inserted into the simulation. They are initially imparted with an impulse force of $12kN$ in the $y$-direction. Figure 11 illustrates the visual representation of the aforementioned initial setup. The purpose of this font is to showcase the swirling motion of the fluid as it traverses the surface of the tensor field with different $\lambda_3$.

In Figures 13, 14, and 15 we can see how the variation of the $\lambda_3$ magnitude affects the tensor field permeability. In frame 240 of Figure 13, the fluid already has transposed completely the tensor field. This full transposition does not occur in the other two experiments. However, we can compare the final frame of Figures 14, and 15 to visualize how lower eigenvalues retained a greater volume of fluid above the tensor field.

By comparing Figures 13, 14, and 15, we observe the impact of varying the magnitude of $\lambda_3$. This is the eigenvalue assigned to the normal of the planar tensor and, consequently, it defines the effect of permeability. In Figure 13, at frame 240, the fluid has completely passed through the tensor field. Conversely, this full transposition does not occur in the other two experiments. However, we can examine the final frames of Figures

Figure 14 – Permeable surface simulation with $\lambda_3 = 10^{-3}$.



(a) Frame 170

(b) Frame 200

(c) Frame 240

(d) Frame 380

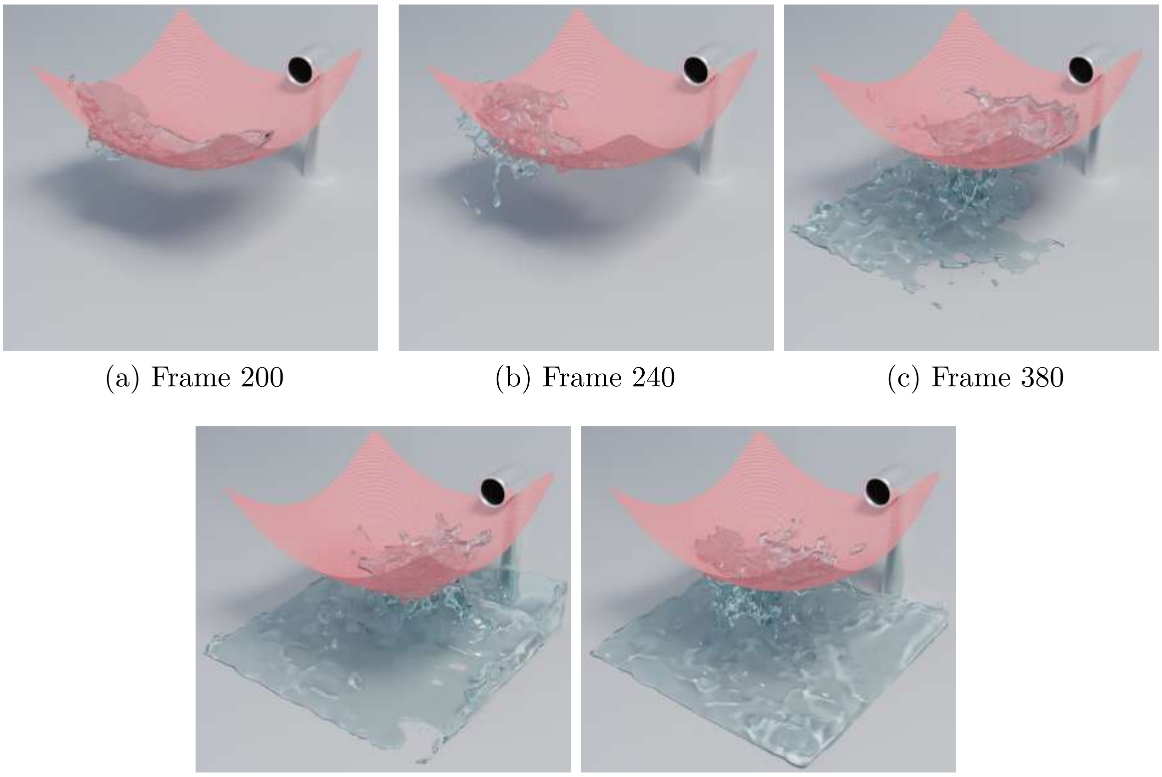
(e) Frame 480

(f) Frame 599

Source: Created by the author.

14 and 15, where a lower eigenvalue resulted in a larger volume of fluid retained above the tensor field.

Figure 15 – Permeable wall simulation with $\lambda_3 = 10^{-4}$.



(a) Frame 200

(b) Frame 240

(c) Frame 380

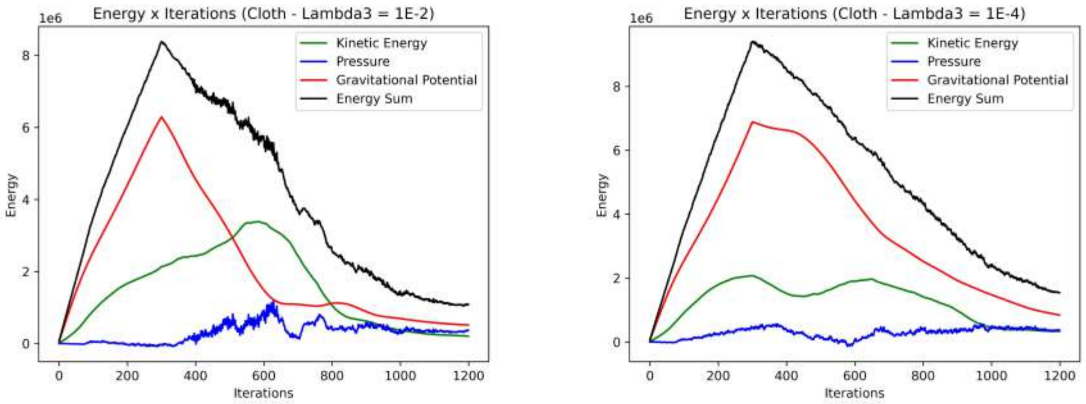
(d) Frame 480

(e) Frame 599

Source: Created by the author.

In the following figures, we observe the impact of varying $\lambda_3$ on the system's energy and the velocity divergence for the field with non-axis-aligned planar tensors. It can be observed that the field with lower $\lambda_3$ retains more fluid over the field. At the peak of gravitational potential energy, since the fluid does not penetrate the field with the lower $\lambda_3$, both energies take longer to decrease than in the field with the higher $\lambda_3$. This can be seen in Figure 16. The potential energy does not reach such large peaks, given that the fluid penetrates the field and collides with the domain bottom in a more controlled manner, in the experiment with $\lambda_3 = 1^{-4}$. In the first graph, in Figure 16, kinetic energy reaches its maximum near iteration 600, and by iteration 800, it approaches a value close to the minimum and remains close to this value throughout the simulation. Meanwhile, in the simulation with lower $\lambda_3$, kinetic energy remains more constant, not reaching a peak like in the first graph, and only approaches its minimum value near iteration 1000.

Figure 16 – Non Axis-Aligned Planar Tensors Bernoulli Equation Integration

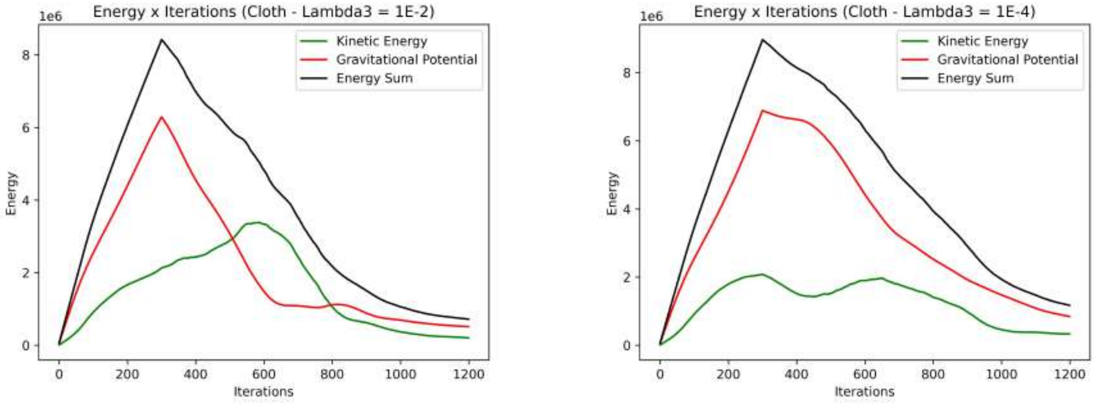


(a) $\lambda_3 = 1^{-2}$ (b) $\lambda_3 = 1^{-4}$

Source: Created by the author.

Figure 17 shows how pressure does not interfere as much in this experiment compared to other energies. This is visible if we compare Figure 17, and Figure 16 as the behavior of the energy sum remains almost the same.

Figure 17 – Non Axis-Aligned Planar Tensors Bernoulli Equation Integration Without Pressure Energy

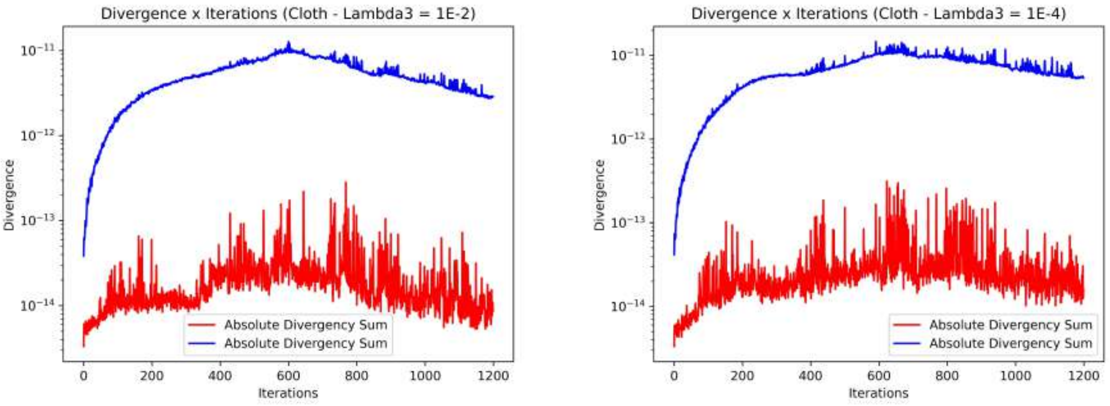


(a) $\lambda_3 = 1^{-2}$ (b) $\lambda_3 = 1^{-4}$

Source: Created by the author.

The same can be said for the velocity divergence in Figure 18; we can perceive a slight difference between both experiments, but this difference is so subtle that it is difficult to determine the real impact of the field or pressure on the divergence.

Figure 18 – Non Axis-Aligned Planar Tensors Divergence



(a) $\lambda_3 = 1^{-2}$

(b) $\lambda_3 = 1^{-4}$

Source: Created by the author.

## 4.2 TIME-DEPENDENT TENSOR FIELDS

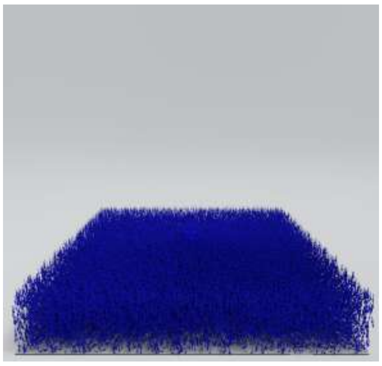
All tensor fields considered for the following experiments vary in time. In the previous section, we describe our experiments to present the intrinsic characteristics of the tensor field, initial conditions, environment setup, and the fluid source and position. Also, we discuss the overall behavior of each experiment and their variations, to point out the main characteristics of each one. The first following experiment denotes an abrupt change of tensor field and how it impacts the fluid flow. The last one, the tensor field, varies smoothly in time over every time step.

### 4.2.1 Simulation with Axis-Aligned Planar & Linear Tensors

The purpose of this experiment is to illustrate the impact of abruptly modifying the current tensor field configuration on pressure relief. This modification allows us to redirect the fluid flow in an arbitrary direction. In this subsection, we provide two simulations with distinct values for the $\alpha$ term in Equation 3.22. The first simulation sets the $\alpha$ term to 1, whereas the second simulation sets it to 0.15. These variations are intended to highlight the influence of the proposed update velocity equation on the simulation of fluid flow and its numerical dissipation.

In each simulation, multiple planar tensor layers are positioned at the bottom of our domain ($k \in [1, 12]$). These layers restrict pressure exchange to the $xy$-plane, akin to the approach used in Section **4.1.1**. However, there are two distinctions regarding the previous tensor field. Firstly, we now set a $\lambda_3$ eigenvalue of $10^{-3}$ to the planar tensors, decreasing pressure exchange in the $z$ direction by a factor of 10. Secondly, the planar tensor layers have “holes” filled with isotropic tensors. Throughout the simulation, we present three variations of the tensor field configuration, each occurring every 200 iterations. Initially,

Figure 19 – Initial setup of the geyser simulation.

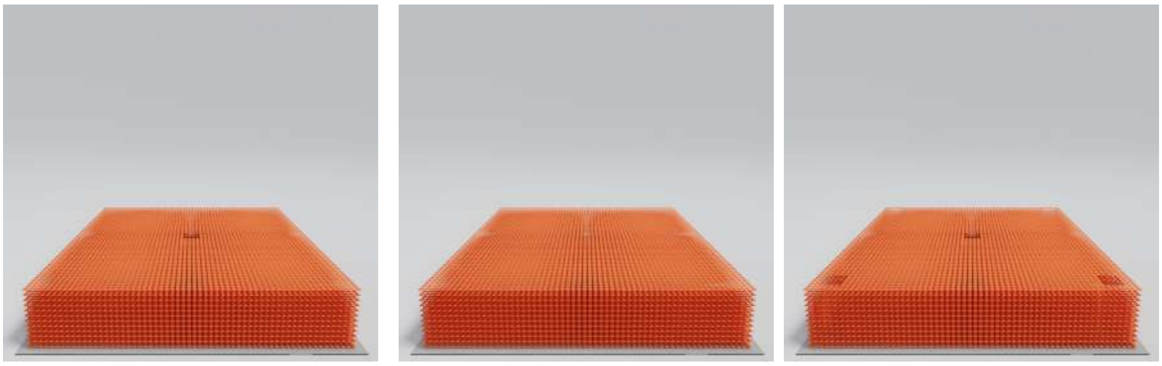


(a)

Source: Created by the author.

the planar layers have a single central hole, allowing the fluid to exchange pressure in the $z$-axis through it. After 200 iterations, the isotropic tensors on the central hole are changed to planar tensors, limiting again the pressure exchange to the $xy$-plane. Finally, in the last 200 iterations, the planar layers contain five holes: one at its center and four at each corner. This variation of the tensor field is illustrated in Figure 20.

Figure 20 – Tensor fields employed on the geyser simulation.



(a) Central Hole

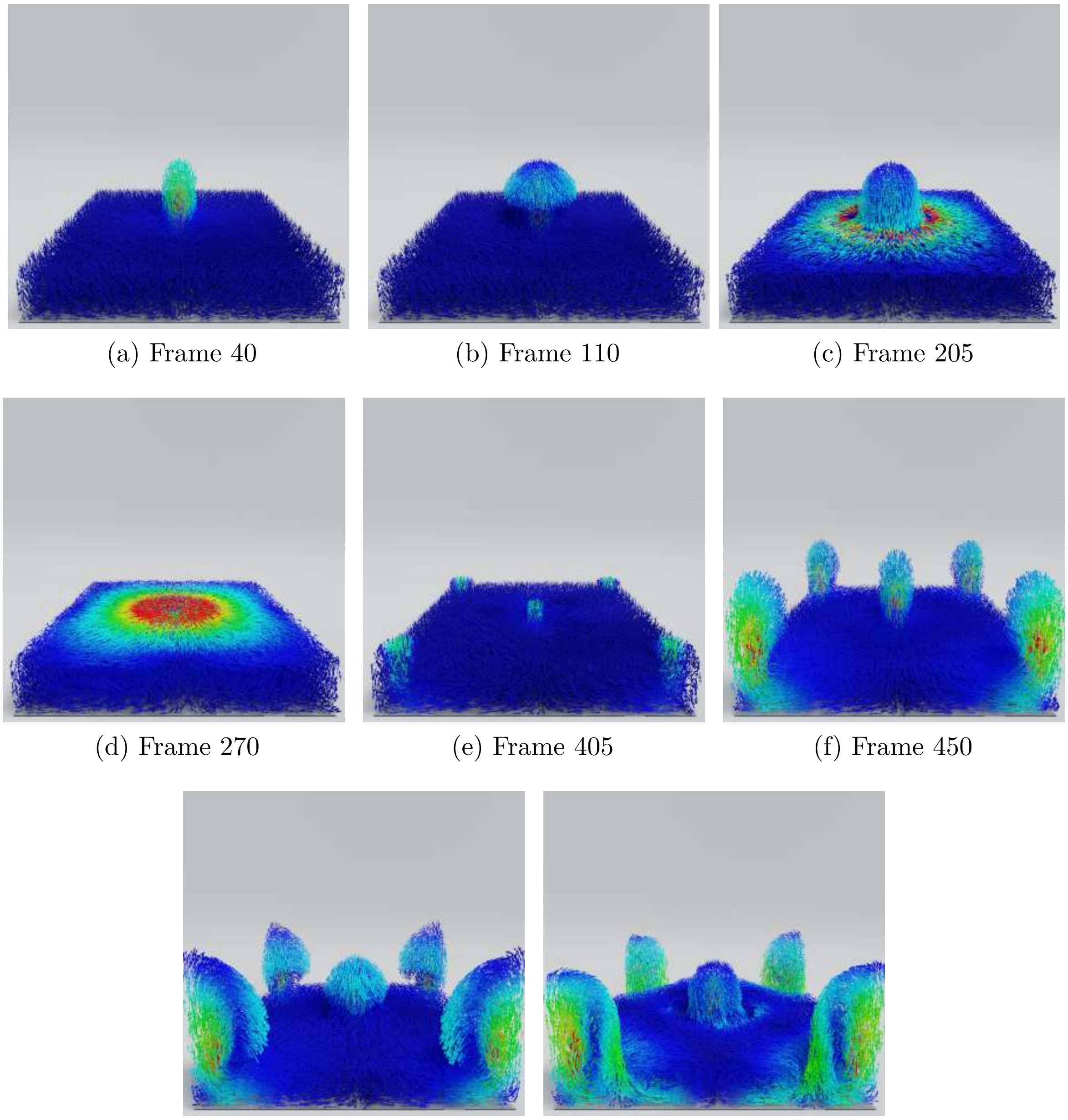
(b) No Holes

(c) Five Holes

Source: Created by the author.

The experiment utilizes a quiescent pool of fluid located at the bottom of the domain ($k \in [1, 11]$). The entire bottom of the domain is occupied by this fluid pool, with an initial quantity of 20 particles per voxel. With the pool having dimensions of $60 \times 60 \times 11$ voxels, a total of $792,000$ particles is present in each simulation. Refer to Figure 19 for a visual representation of the initial setup employed in this experiment.

Figure 21 – Geyser simulation with $\alpha = 1$.



(a) Frame 40

(b) Frame 110

(c) Frame 205

(d) Frame 270

(e) Frame 405

(f) Frame 450

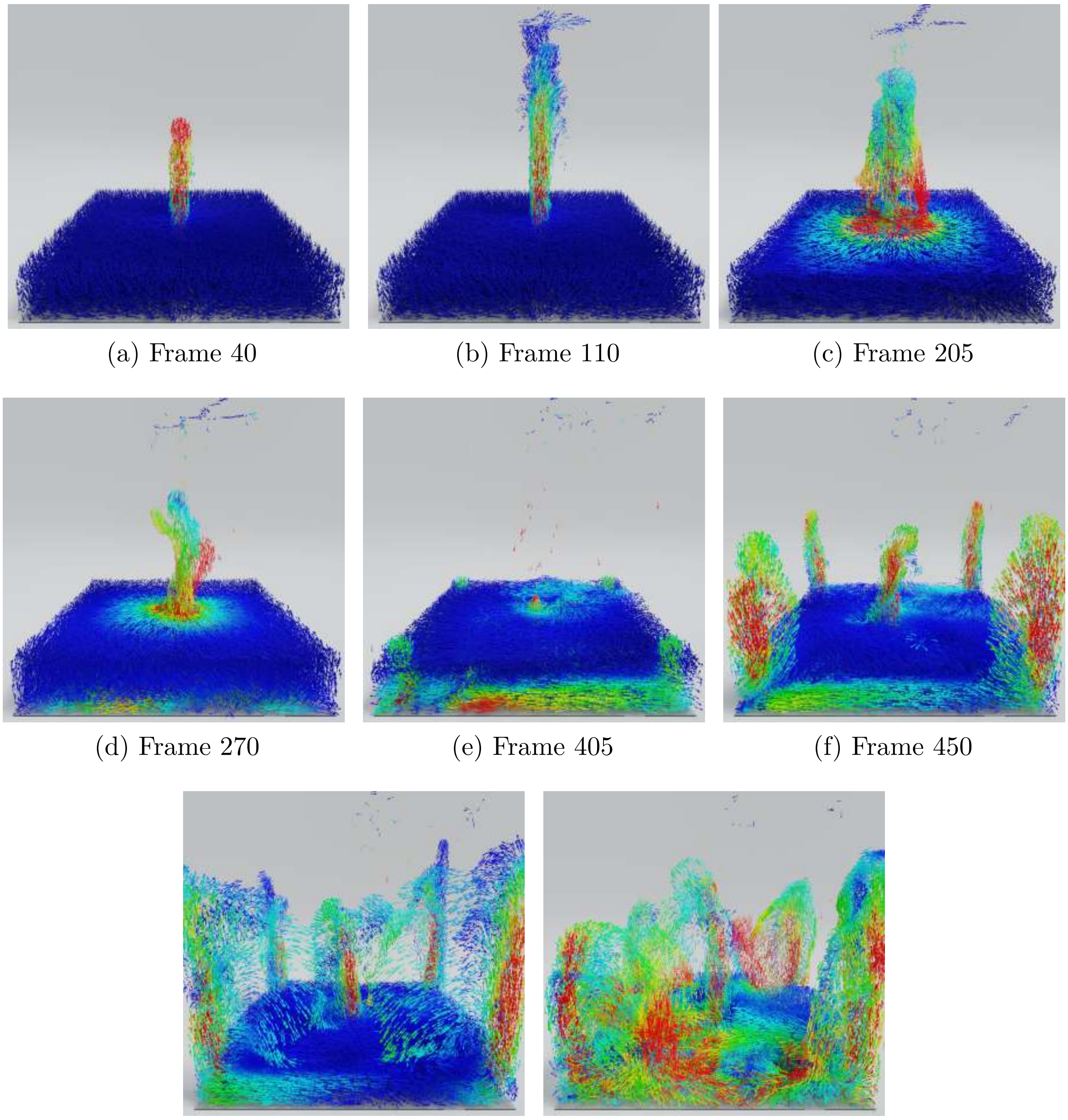
(g) Frame 500

(h) Frame 599

Source: Created by the author.

In this experiment, we aim to observe the impact of the pressure relief scheme and particle velocity update variations on the velocity of fluid flow. Figures 21 and 22 depict arrows indicating the particle velocity vectors. These arrows are color-coded based on the magnitude of the velocity, ranging from $0(m/s)$ to $30(m/s)$. Dark blue colors represent lower values, while bright reddish colors represent higher values.

Figure 22 – Geyser simulation with $\alpha = 0.15$.



(a) Frame 40

(b) Frame 110

(c) Frame 205

(d) Frame 270

(e) Frame 405

(f) Frame 450

(g) Frame 500

(h) Frame 599

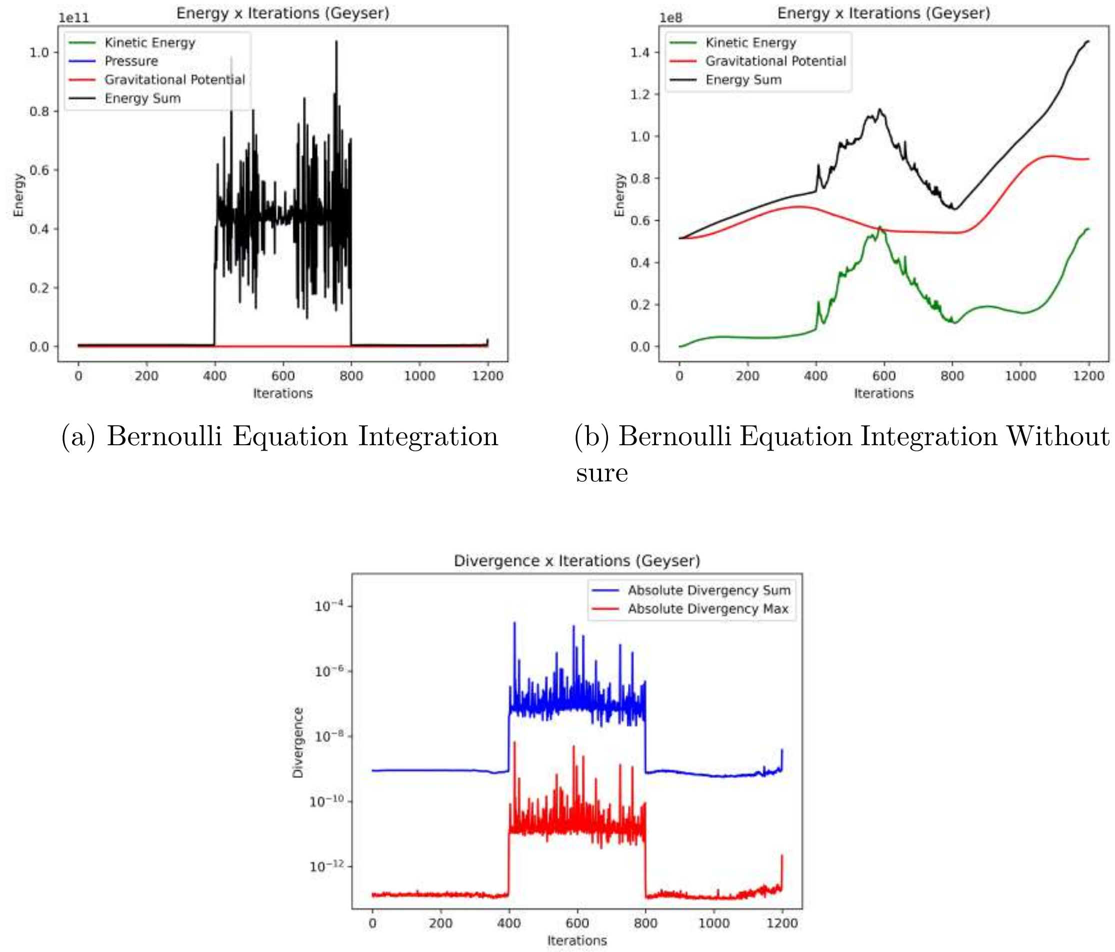
Source: Created by the author.

When comparing Figures 21 and 22, the geyser altitudes and arrow colors demonstrate the clear impact of the velocity update and the tensor field variation. The arrow colors of the fluid in the geysers (Figure 22) appear more reddish, indicating that the pressure relief in this experiment yields a higher acceleration compared to the $\alpha = 1$ simulation. Consequently, the simulation with $\alpha = 1$ (Figure 21) achieves a significantly lower altitude compared to the $\alpha = 0.15$ simulation. This is evident comparing both simulations on frame 110 or frame 500. Frames 205 and 405 demonstrate how the fluid behaves slightly after the tensor field variation. These frames showcase the rapid flow redirection towards the pressure relief points. An intriguing aspect of the $\alpha = 0.15$

simulation is that a single-hole geyser reaches a higher altitude than the one with five holes. With multiple holes, the fluid's kinetic energy is distributed, limiting the height it can reach when compared to a geyser with a single hole. This is not achievable with $\alpha = 1$ independently of the number of holes, the simulation does not have enough energy to attain a higher altitude.

In line with previous experiments, in Figure 23, we observe the impact of varying the tensor field on the integration of the Bernoulli equation, the Bernoulli equation without pressure energy, and the velocity divergence. In the following descriptions, whenever a hole in the domain is mentioned, we are referring to the sections on the tensor field that we filled with isotropic tensors between the planar tensors. We can see in Figure 23 that closing the central hole in the domain from iteration 400 to 800 significantly affects the behavior of system energies and divergence. Filling the central cells of the domain with planar tensors results in an increase in divergence from a scale of $10^{-}9$ to $10^{-}7$ with a notably high standard deviation of the mean. The divergence graph closely mirrors that of the complete integration of the Bernoulli equation. However, in the graph depicting integration without pressure, we notice a different behavior on gravitational and kinetic energies between the phase with five holes in the domain compared to the initial phase with a single hole at the center. The kinetic energy of the system exhibits a more substantial initial increase after introducing additional holes in the domain, followed by a subsequent decline and then an exponential rise, contrasting with an almost linear progression in the phase with only one hole in the domain. Furthermore, both gravitational potential energy and kinetic energy experience an uptick with an increased number of holes in the domain. This arises from the simultaneous movement of a greater volume of fluid and the cumulative energy that occurs when no holes are present in the system. Consequently, despite achieving greater heights in the initial phase with only one hole, the third phase with five holes moves a larger volume of fluid and nearly reaches the same height as the initial phase.

Figure 23 – Time Varying Axis-Aligned Planar & Linear Tensors Bernoulli Equation Integrations and Velocity Divergence



(a) Bernoulli Equation Integration

(b) Bernoulli Equation Integration Without Pressure

(c) Velocity Divergence

Source: Created by the author.

### 4.2.2 Simulation with the Strain-Rate Tensors

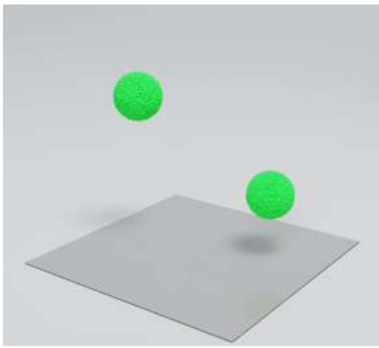
This subsection proposes a smooth adjustment to the tensor field across the entire fluid simulation. The primary objective of this experiment is to demonstrate the utilization of the tensor field to influence the pressure of fluid flow throughout each step of the simulation. Our focus lies in ensuring that the fluid alleviates its pressure in the direction of deformation, while limiting pressure exchange in other directions. To accomplish this, we incorporate the strain-rate tensor, as discussed in Section **2.1.2**, as a crucial element in designing the tensor field for this experiment.

However, a challenge arises when attempting to use the strain-rate tensor, as it possesses the issue of potential semi-definiteness. Our method, outlined in Section **3.1.1**, mandates that our tensor must be positive-definite. However, this limitation can be resolved by integrating the strain-rate tensor. Consequently, the tensors employed in this experiment are described as follows:

$$\mathbf{T} = \sum_{i=1}^{M} \mathrm{e}^{\gamma(\Delta t \lambda_i)} \mathbf{e}_i \mathbf{e}_i^{\mathrm{T}},$$

where $\lambda_i$ and $\mathbf{e}_i$ are the $i$-th eigenvalues and eigenvectors of the strain-rate tensor, respectively. The scalar factor parameter, denoted as $\gamma$, is incorporated into the equation for us to control the tensor influence on each simulation. Analyzing the behavior of the aforementioned equation, a positive-definite tensor is consistently generated. If $\lambda_i < 0$, the resulting tensor indicates fluid compression along the corresponding axis. If $\lambda_i = 0$, the tensor will have no impact on the pressure exchange along the referenced axis. Conversely, if the eigenvalue surpasses 1, the tensor will enhance pressure relief along the $i$-th axis.

Figure 24 – Initial setup of the colliding spheres simulation.

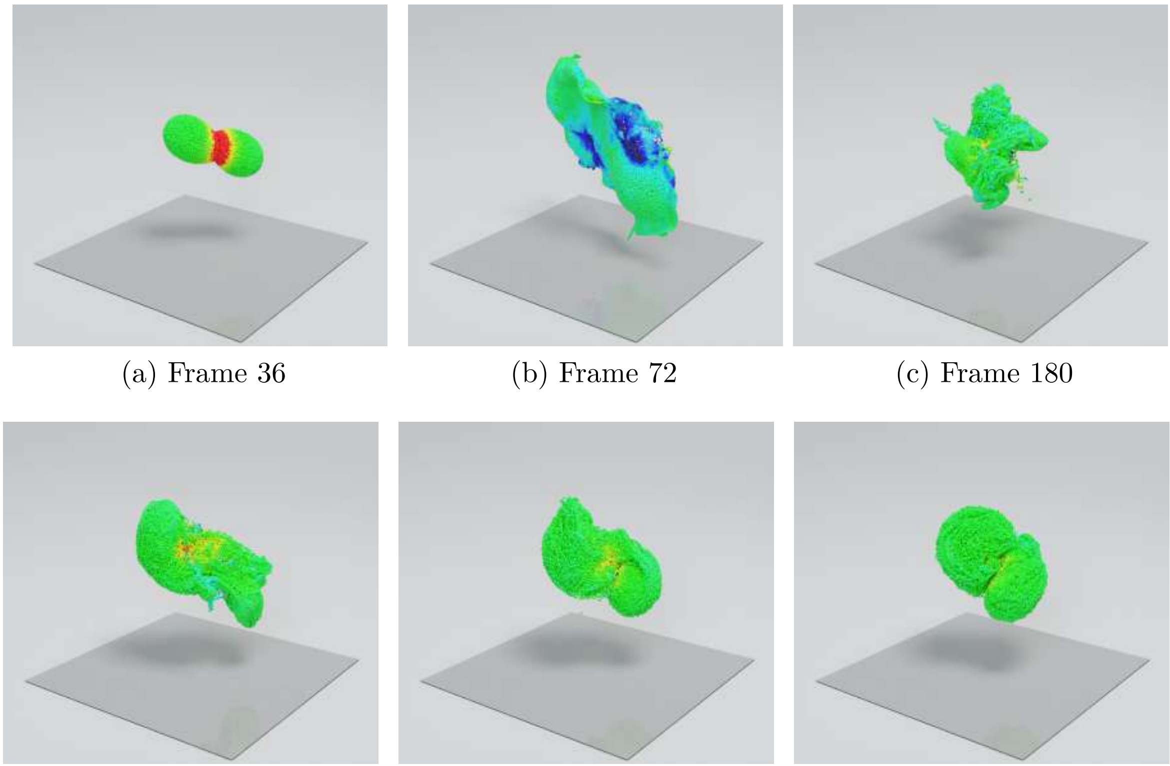


(a)

Source: Created by the author.

In this section, we introduce three variations of the proposed experiment. The first simulation serves as a reference, where the standard X-IVAS method is employed. This first simulation is denoted over Figure 25. Subsequently, two additional simulations are conducted with different $\gamma$ values. The first simulation utilizes $\gamma = 1$, while the second simulation utilizes $\gamma = 5$ as we want to emphasize the tensor's impact on the fluid flow. These two simulations are described in Figure 26, and Figure 27 respectively.

Figure 25 – Colliding spheres using the isotropic tensor field.



(a) Frame 36

(b) Frame 72

(c) Frame 180

(d) Frame 252

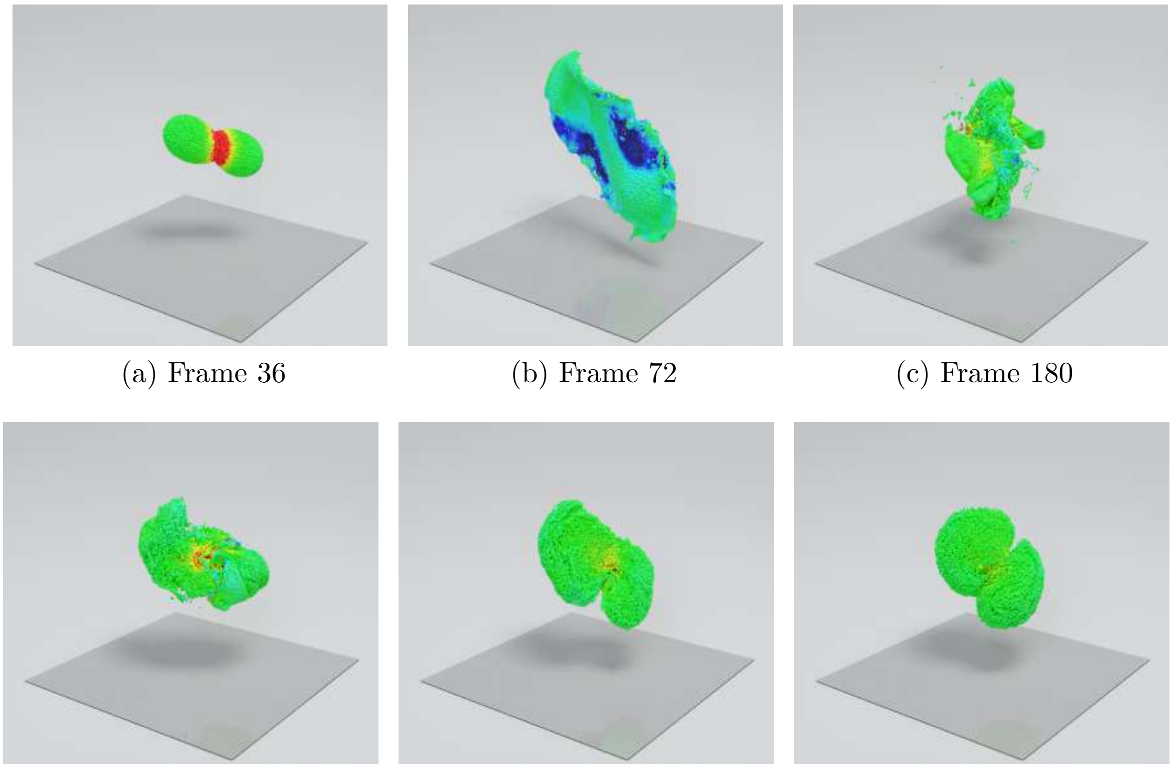
(e) Frame 350

(f) Frame 550

Source: Created by the author.

The velocity update term, denoted as $\alpha$, has a fixed value of 0.15. Additionally, it is important to note that the gravity force is directed towards the center of the simulation domain, with a magnitude of $10(m/s^2)$. The color scheme of the particles depicted in the figures within this subsection follows the same scheme as the previously presented falling block experiment discussed in Section **4.1.1**.

Figure 26 – Colliding spheres using the strain tensor field with $\gamma = 1$.



(a) Frame 36

(b) Frame 72

(c) Frame 180

(d) Frame 252

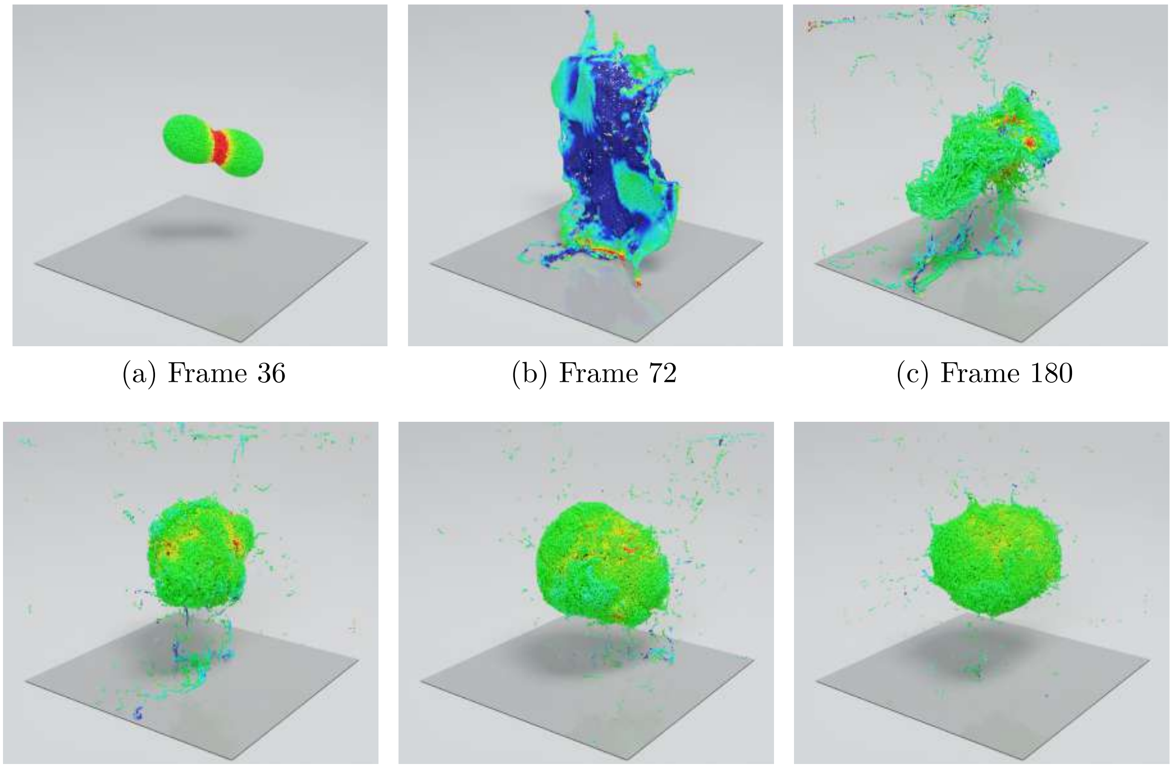
(e) Frame 350

(f) Frame 550

Source: Created by the author.

The experiment consists of two diagonally opposed spheres of fluid, as illustrated in Figure 24. Each sphere has a radius of 6 voxels, and an initial particle density of 100 particles per voxel. Thus, the total number of particles in this experiment is calculated as $2 \times (4\pi/3) \times 6^3 \times 100 = 180,956$ particles. Both spheres possess identical initial velocity vectors but in opposing directions, represented as $\mathbf{u}_0 = [5, 0, 5]$ and $\mathbf{u}_1 = [-5, 0, -5]$.

Figure 27 – Colliding spheres using the strain tensor field with $\gamma = 5$.



(a) Frame 36

(b) Frame 72

(c) Frame 180

(d) Frame 252

(e) Frame 350

(f) Frame 550

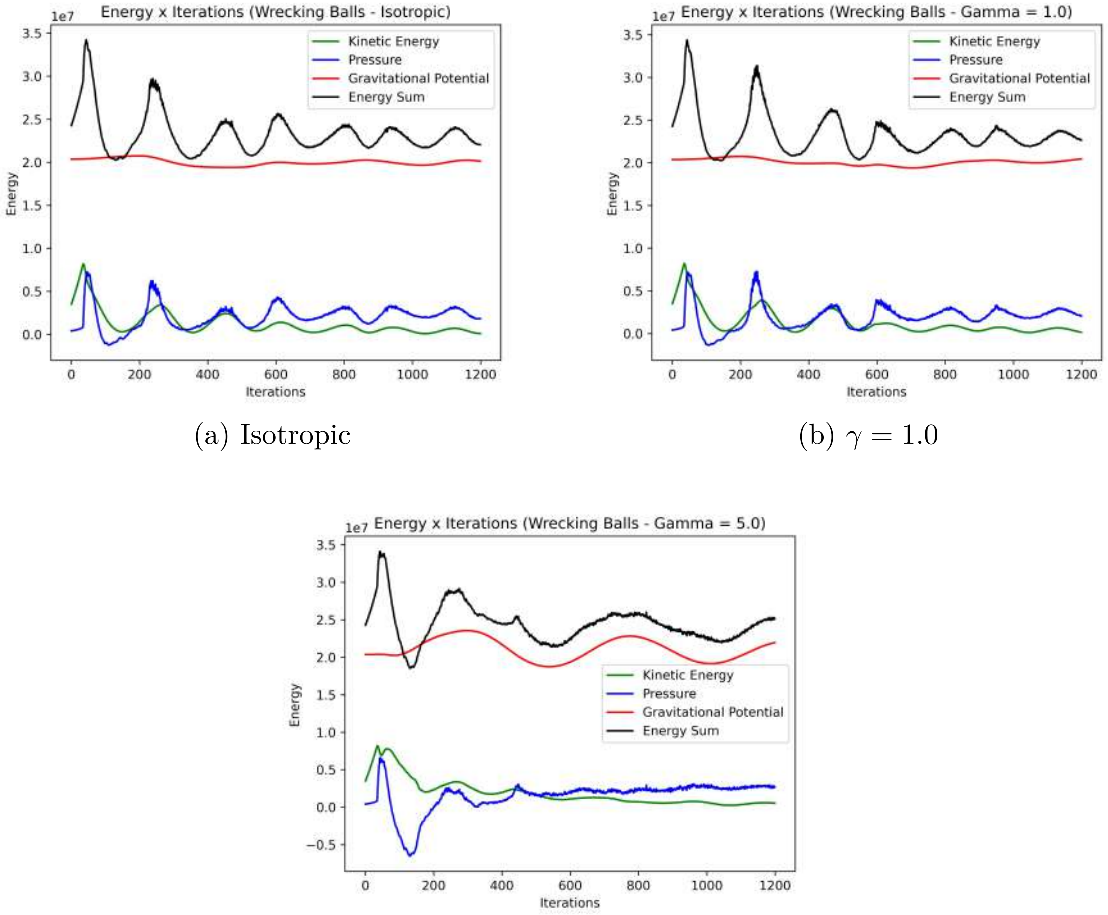
Source: Created by the author.

Upon comparing the frames presented in Figure 26 with the reference simulation depicted in Figure 25, we observe subtle differences. The smooth variation of the tensor field has a relatively minimal impact on pressure relief. However, the influence of the tensor becomes evident in Figure 27. Notable distinctions can be observed, such as the collision portrayed in frame 72 and the formation of a fluid sphere at the center of the domain. Additionally, the color of the fluid following the collision illustrates a lower-pressure configuration. As the fluid splashes, it generates a sparser layer of fluid, leading to a broader area of low pressure. This sparsity arises from the generation of high pressure during the sphere collision. The motion toward the opposing collision direction is accentuated by the strong influence of the tensor field ($\gamma = 5$), propelling the fluid flow in the direction of deformation. Furthermore, gravitational forces pull the fluid towards the domain's center, which is intensified by the tensor field that enhances the fluid pressure exchange in this deforming direction. Consequently, a dense fluid sphere forms, distinguishing it from the other two simulations.

In the following figures, we observe the influence of the strain-rate tensor on the system's energies and the velocity divergence. When using the dimensionless tensor, with $\gamma = 1$, the system's energy remains quite similar to that of the isotropic field; the difference is too subtle to highlight a significant impact on the fluid flow energies, as we can see in

Figure 28. However, with $\gamma = 5$, we can discern a smoothing effect on the energies, as if the field better preserves the system's energy even with numerous collisions occurring. In this experiment, gravitational potential energy dictates the graph behavior over the others. However, due to the numerous collisions, we can also observe the impact that pressure has on the system as the energy sum tends to have the same behavior as the pressure energy.

Figure 28 – Strain-Rate Tensors Bernoulli Equation Integration Speed Divergence



(a) Isotropic

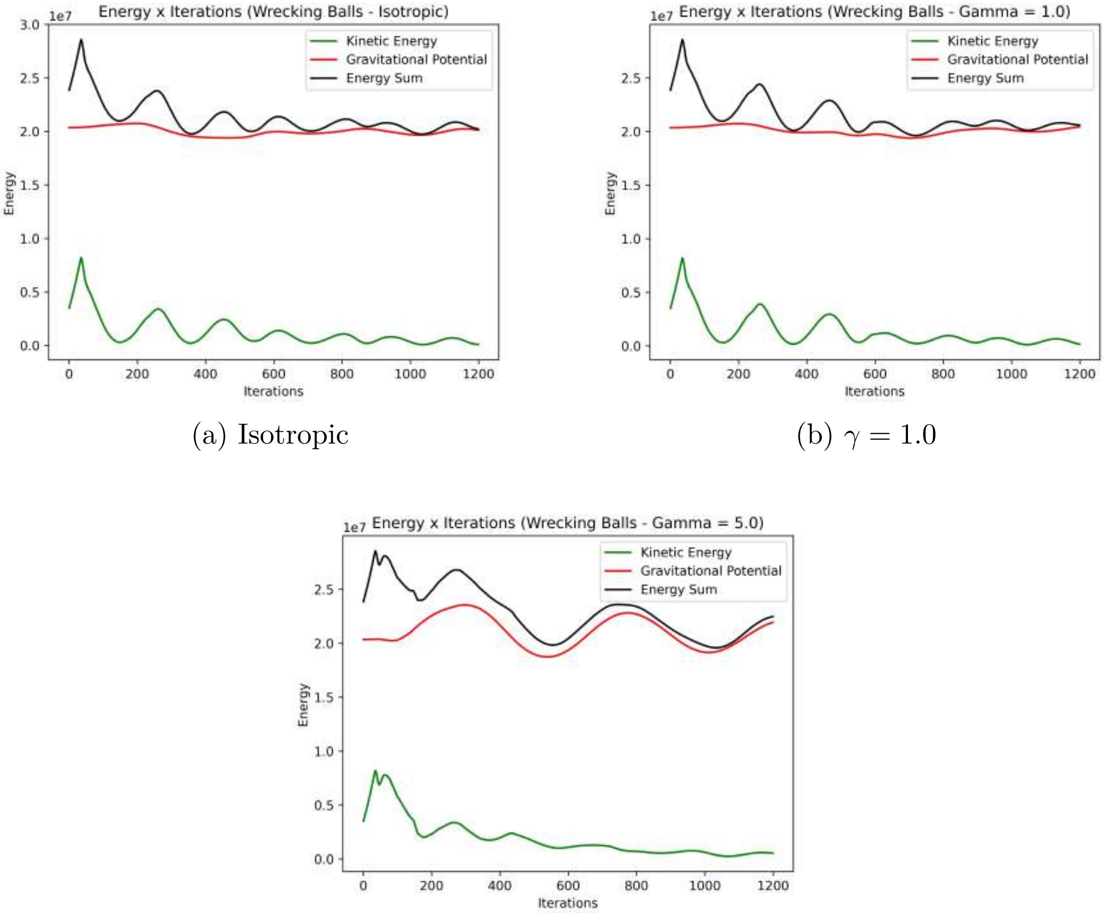
(b) $\gamma = 1.0$

(c) $\gamma = 5.0$

Source: Created by the author.

In Figure 29, we observe how gravitational and kinetic energy are significantly smoother compared to Figure 28, showing how pressure inserts some noise in the system. Additionally, we can observe how the field with $\gamma = 5$ truly influences the experiment and energy conservation, given that fluid flow tends to favor energy exchange in the direction where deformation occurs.

Figure 29 – Strain-Rate Tensors Bernoulli Equation Integration without Pressure



(a) Isotropic

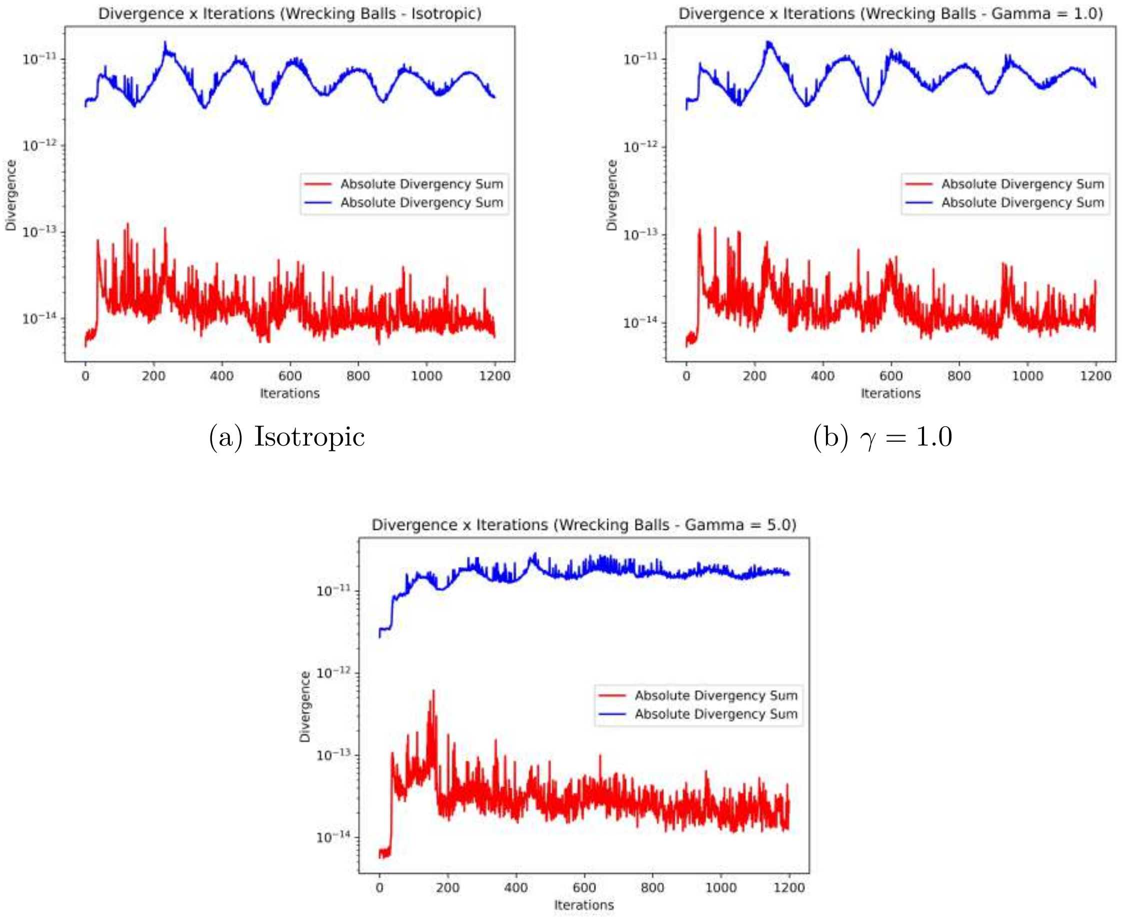
(b) $\gamma = 1.0$

(c) $\gamma = 5.0$

Source: Created by the author.

Interestingly, the same can be said about the fluid divergence in Figure 30; while the isotropic field and $\gamma = 1$ result in numerous collisions and a greater variation in divergence with each collision, the field with $\gamma = 5$ tends to have a divergence with less standard deviation of the mean, while also tending to possess a lower magnitude than the other two graphs.

Figure 30 – Strain-Rate Tensors Bernoulli Equation Integration



(a) Isotropic

(b) $\gamma = 1.0$

(c) $\gamma = 5.0$

Source: Created by the author.

## 4.3 DISCUSSION

The core parameters of our experiments are presented in Table 3. It includes the computational time cost for each experiment. The computational cost increases significantly when dealing with time-varying tensor fields. The main factor contributing to this computational cost is the recomputation of the Laplacian mask proposed by Parreiras [2022]. Such recomputation poses a prohibitive cost for real-time applications. Nevertheless, if the tensor field remains static throughout the simulation, the computational time cost becomes manageable.

Throughout all experiments, and despite introducing a whole new computational complexity on the computation of the tensor field, our method yields very low magnitude errors around $10^{-12}$ (maximum) for all divergence-free vector fields. A key contribution that enabled us to achieve this level of tolerance and low divergence is the discretization proposed by Parreiras [2022] on an extended MAC grid. For the sake of reproducibility,

Table 3 – Summary of experiments setup and simulation times

| Experiment | Parameter | | | | | Simulation | |
|---|---|---|---|---|---|---|---|
| | Grid | # particles | Tensor | $\Delta t$ $(s)$ | $\alpha$ | Sec./frame | Total time |
| Falling Block 1 (Fig. 5) | $60^3$ | $160,000$ | Isotropic | 1/60 | 1.0 | 1.91 | 00:38:10 |
| Falling Block 2 (Fig. 6) | $60^3$ | $160,000$ | Aligned planar | 1/60 | 1.0 | 1.79 | 00:35:50 |
| Falling Block 3 (Fig. 7) | $60^3$ | $160,000$ | Aligned linear | 1/60 | 1.0 | 2.16 | 00:43:08 |
| Cloth $\lambda_3 = 10^{-2}$ (Fig. 13) | $60^3$ | $250,000$ | Non-Aligned planar | 1/60 | 0.15 | 3.27 | 01:05:19 |
| Cloth $\lambda_3 = 10^{-3}$ (Fig. 13) | $60^3$ | $250,000$ | Non-Aligned planar | 1/60 | 0.15 | 3.24 | 01:04:51 |
| Cloth $\lambda_3 = 10^{-4}$ (Fig. 13) | $60^3$ | $250,000$ | Non-Aligned planar | 1/60 | 0.15 | 3.06 | 01:01:17 |
| Geysers 1 (Fig. 21) | $60^3$ | $792,000$ | Varying aligned planar | 1/60 | 1.0 | 20.40 | 06:47:55 |
| Geysers 2 (Fig. 22) | $60^3$ | $792,000$ | Varying aligned planar | 1/60 | 0.15 | 20.69 | 06:53:52 |
| Colliding Spheres (Fig. 25) | $60^3$ | $180,956$ | Strain | 1/60 | 0.15 | 1.89 | 00:37:52 |
| Colliding Spheres $\gamma = 1$ (Fig. 26) | $60^3$ | $180,956$ | Strain | 1/60 | 0.15 | 6.45 | 02:09:04 |
| Colliding Spheres $\gamma = 5$ (Fig. 27) | $60^3$ | $180,956$ | Strain | 1/60 | 0.15 | 6.70 | 02:13:58 |

our source code is available in `www.gcg.ufjf.br`.

The proposed method yielded many appealing results across our experiments. While we maintained fixed grid resolution and time step for comparison purposes, we observed that plausible results can be generated even with higher time steps and lower resolutions. Although we presented specific tensor field configurations and showcased their impact on fluid flow, it is important to note that this work does not encompass an exhaustive enumeration of all potential applications and effects of the proposed method.

The utilization of planar tensors made a significant contribution to artificially increasing the pressure over the fluid volume, as depicted in Sections **4.1.1**, and **4.2.1**. Also, through planar tensors, we were able to generate permeability-like effects, as exemplified in Section **4.1.2**. This demonstrates the possibility of either exerting subtle control over the fluid trajectory or retaining the fluid to a specific location. However, achieving this effect required the development of a novel body force (Eq. 4.3). It should be noted that this force application necessitates a tensor field comprising at least two surface layers (two cells in the target direction) to be captured by our divergence computation scheme. Fields with only one layer would result in a considerably smoother tensor-based force, with possibly low impact on the fluid flow. Another important limitation of this force lies in its sensitivity to the time step. Higher time steps tend to reduce the force influence on the local fluid dynamics. A possible future improvement could involve integrating this force with the three-step third-order Runge-Kutta scheme.

The utilization of linear tensor fields is valuable when the pressure exchange needs to occur in specific directions, as shown in Section **4.1.1**. They proved quite difficult to design since they present fewer paths to the fluid flow to relieve the pressure in the projection step. However, they are useful when combined with planar tensors to divert fluid in a specific location on a surface with a specific induced pressure. Moreover, linear tensors appear frequently in the strain tensor simulation with $\gamma = 5$, during the fluid collision. As a consequence, the deformations appeared in specific directions thereafter.

Our method maintains stability over higher time steps, as suggested by the default X-IVAS method presented in the work of Idelsohn et al. [2012]. As expected, when the FLIP update is fully used with $\alpha \approx 0$, the simulation can present instabilities. This is a well-known behavior, that happens when the velocity field does not necessarily have divergence-free.

The tensor field modulation over the pressure and viscous terms in our model does not preserve momentum, as is seen in the geyser experiment. Thus, tensor fields can increase and decrease local velocities and consequently can represent dynamic medium attributes. A deeper analysis of the moment and mass conservation is a future work to improve our propositions.

By analyzing the integration of energies for the geyser (Figure 23), and strain rate experiment (Figure 28), we observe that in the presence of a strong impact, pressure exhibits a peak resembling a Dirac delta function. Subsequently, it may generate a zone with negative pressure, after which the system tends to normalize. This renders the integration values somewhat counterintuitive in these cases. Therefore, we believe a more in-depth analysis is necessary to determine whether this phenomenon is physically feasible or if such an impact could be propagated less immediately within the system.

# 5 CONCLUSION

We present a mathematical and computational model that deals with anisotropic fluid transport. The model is based on second-order positive-definite tensor fields that can change over time. We propose a new numerical solution for anisotropic advection based on the X-IVAS method. This solution takes into account the influence of tensors on viscous forces and accumulated pressure gradient.

Additionally, we propose a numerical solution for our novel tensor-based anisotropic decomposition (Eq. 3.20). This decomposition is designed to accumulate and adjust pressure as the tensor field evolves over time as discussed in Section **3.1.1**. Unlike previous literature works [Nigro et al., 2011, Idelsohn et al., 2012, 2013, 2014, Nadukandi et al., 2017], our contributions were discretized using the extended MAC grid proposed by Parreiras [2022]. However, our method can be implemented on various discretization schemes, such as PFEM. Furthermore, we devised a velocity update scheme that guarantees that it remains divergence-free as the tensor field varies in time (Eq. 3.21).

In Chapter 4, we conducted various experiments using static and time-dependent tensor fields. We explored different parameter and tensor field variations for comparison purposes. Our results demonstrate that the proposed method succeeds in anisotropic modulating pressure and viscosity terms. We also present in Section **4.1.2** an experiment showing the usage of a body force based on the tensor field configuration. This possibility enables our method to simulate a wider range of effects. However, in this case, we did not couple the body force with the advection step as in previous works Renhe et al. [2019], Parreiras et al. [2022]. In future works, we aim to incorporate tensor influence into our advection scheme, with compromise to numerical accuracy and stability. Designing tensor fields to modulate fluid dynamics is still a challenge. Its inherent multivariate effect locally in space is not intuitive. Furthermore, in our work, we introduce one more degree of freedom by allowing the tensor field to change in time. Consequently, it is necessary to create mathematical models to design meaningful tensor fields, application-based tensor fields, and perhaps families of tensor fields with valuable differential properties.

As discussed in Section 4.3, our anisotropic pressure and viscous term modulation do not preserve momentum. Therefore, future works will delve deeper into the analysis of momentum and mass conservation. Also, we seek to find out what properties a tensor field needs to enforce conservation.

# REFERENCES

Charles L Fefferman. **Existence and smoothness of the Navier-Stokes equation**. *The millennium prize problems*, 57:67, 2000.

Joseph Spurk and Nuri Aksel. ***Fluid mechanics***. Springer Science & Business Media, 2007.

Robert Bridson. ***Fluid simulation for computer graphics***. CRC press, 2015.

Marcelo Caniato Renhe. **A stable tensor-based deflection model for controlled fluid simulations**. 2017.

Marcelo Bernardes Vieira, Gilson Antonio Giraldi, Allan Carlos Amaral Ribeiro, Marcelo Caniato Renhe, and Claudio Esperança. **Anisotropic Helmholtz decomposition for controlled fluid simulation**. *Applied Mathematics and Computation*, 411:126501, 2021. ISSN 0096-3003.

Emanuel Antônio Parreiras. **Métodos Particle-in-Cell para Simulação de Fluidos Anisotrópico**. Master's thesis, Universidade Federal de Juiz de Fora, Juiz de Fora, MG, fev 2022.

JB Marston and SM Tobias. **Recent developments in theories of inhomogeneous and anisotropic turbulence**. *Annual Review of Fluid Mechanics*, 55:351–375, 2023.

Toon Lenaerts, Bart Adams, and Philip Dutré. **Porous flow in particle-based fluid simulations**. *ACM Transactions on Graphics (TOG)*, 27(3):1–8, 2008.

YI Zhu, Patrick J Fox, and Joseph P Morris. **A pore-scale numerical model for flow through porous media**. *International journal for numerical and analytical methods in geomechanics*, 23(9):881–904, 1999.

M Sawley, P Cleary, and Joseph Ha. **Modelling of flow in porous media and resin transfer moulding using smoothed particle hydrodynamics**. 1999.

Thien Tran-Duc, Erwan Bertevas, Nhan Phan-Thien, and Boo Cheong Khoo. **Simulation of anisotropic diffusion processes in fluids with smoothed particle hydrodynamics**. *International Journal for Numerical Methods in Fluids*, 82(11): 730–747, 2016.

Romain Guibert, Pierre Horgue, Gérald Debenest, and Michel Quintard. **A comparison of various methods for the numerical evaluation of porous media permeability tensors from pore-scale geometry**. *Mathematical Geosciences*, 48:329–347, 2016.

E Sanchez-Palencia. **On the asymptotics of the fluid flow past an array of fixed obstacles**. *International Journal of Engineering Science*, 20(12):1291–1301, 1982.

Shlomo P Neuman. **Theoretical derivation of Darcy's law**. *Acta mechanica*, 25(3-4): 153–170, 1977.

L Durlofsky and JF Brady. **Analysis of the Brinkman equation as a model for flow in porous media**. *The Physics of fluids*, 30(11):3329–3341, 1987.

Marcelo Caniato Renhe, Marcelo Bernardes Vieira, and Claudio Esperança. **A stable tensor-based method for controlled fluid simulations**. *Applied Mathematics and Computation*, 343:195–213, 2019.

Emanuel Antônio Parreiras, Marcelo Bernardes Vieira, Arthur Gonze Machado, Marcelo Caniato Renhe, and Gilson Antônio Giraldi. **A particle-in-cell method for anisotropic fluid simulation**. *Computers & Graphics*, 102:220–232, 2022.

Jos Stam. **Stable fluids**. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 121–128, 1999.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. **Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations**. *Journal of Computational Physics*, 378:686–707, 2019.

Han Gao, Luning Sun, and Jian-Xun Wang. **PhyGeoNet: Physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain**. *Journal of Computational Physics*, 428: 110079, 2021.

Jin-Long Wu, Karthik Kashinath, Adrian Albert, Dragos Chirila, Heng Xiao, et al. **Enforcing statistical constraints in generative adversarial networks for modeling chaotic dynamical systems**. *Journal of Computational Physics*, 406:109209, 2020.

Joseph C Kolecki. **An introduction to tensors for students of physics and engineering**. Technical report, Glenn Research Center, 2002.

Carl-Fredrik Westin. ***A tensor framework for multidimensional signal processing***. PhD thesis, Linköping University Electronic Press, 1994.

I-Shih Liu et al. ***Continuum mechanics***, volume 5. Springer, 2002.

Peter J Basser and Carlo Pierpaoli. **Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor MRI**. *Journal of magnetic resonance*, 213(2):560–570, 2011.

Gordon Kindlmann. **Superquadric tensor glyphs**. In *Proceedings of the Sixth Joint Eurographics-IEEE TCVG conference on Visualization*, pages 147–154, 2004.

Lev Davidovich Landau and Evgenii Mikhailovich Lifshitz. ***Fluid Mechanics: Landau and Lifshitz: Course of Theoretical Physics, Volume 6***, volume 6. Elsevier, 2013.

George Dassios and Ismo V Lindell. **Uniqueness and reconstruction for the anisotropic Helmholtz decomposition**. *Journal of Physics A: Mathematical and General*, 35(24):5139, 2002.

Joel H Ferziger, Milovan Perić, and Robert L Street. ***Computational methods for fluid dynamics***, volume 3. Springer, 2002.

RD Bird. **Transport Phenomena Wiley**. *Transport phenomena Wiley. 2nd edition*, 2006.

Frank M WHITE. **Fluid Mechanics, Nueva York**, 2010.

Alexandre Joel Chorin. **The numerical solution of the Navier-Stokes equations for an incompressible fluid**. *Bulletin of the American Mathematical Society*, 73(6): 928–931, 1967.

Philip M Gresho. **On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 1: Theory**. *International Journal for Numerical Methods in Fluids*, 11(5):587–620, 1990.

Francis H Harlow. **The particle-in-cell method for numerical solution of problems in fluid dynamics**. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 1962.

Jeremiah U Brackbill and Hans M Ruppel. **FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions**. *Journal of Computational physics*, 65(2):314–343, 1986.

Francis H Harlow and J Eddie Welch. **Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface**. *The physics of fluids*, 8(12): 2182–2189, 1965.

Norberto Marcelo Nigro, Juan Marcelo Gimenez, Alejandro Cesar Limache, Sergio Rodolfo Idelsohn, Eugenio Oñate, Nestor Alberto Calvo, Pablo José Novara, and Pedro Morin. **A New Approach to Solve Incompressible Navier Stokes Equations Using a Particle Method**. 2011.

Sergio Idelsohn, Norberto Nigro, Alejandro Limache, and Eugenio Oñate. **Large time-step explicit integration method for solving problems with dominant convection**. *Computer Methods in Applied Mechanics and Engineering*, 217:168–185, 2012.

Sergio Rodolfo Idelsohn, Norberto Marcelo Nigro, Juan Marcelo Gimenez, Riccardo Rossi, and Julio Marcelo Marti. **A fast and accurate method to solve the incompressible Navier-Stokes equations**. *Engineering Computations*, 2013.

Sergio R. Idelsohn, Julio Marti, Pablo Becker, and Eugenio Onate. **Analysis of multifluid flows with large time steps using the particle finite element method**. *International Journal for Numerical Methods in Fluids*, 75(9):621–644, 2014.

Prashanth Nadukandi, Borja Servan-Camas, Pablo Agustín Becker, and Julio García-Espinosa. **Seakeeping with the semi-Lagrangian particle finite element method**. *Computational Particle Mechanics*, 4:321–329, 2017.

Cx K Batchelor and George Keith Batchelor. ***An introduction to fluid dynamics***. Cambridge university press, 1967.

Anthony Ralston. **Runge-Kutta methods with minimum error bounds**. *Mathematics of computation*, 16(80):431–437, 1962.

SideFX®. ***Houdini***, 2021. URL `http://www.sidefx.com`.

Kitware Inc Sandia National Labs and Los Alamos National Labs. ***Paraview: Parallel visualization application***, 2021. URL `http://paraview.org`.

Blender Foundation. ***Blender Online Community, Blender - a 3D modelling and rendering package***. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2021. URL `http://www.blender.org`.

LuxCoreRender®. ***LuxCoreRender***, 2018. URL `https://luxcorerender.org`.