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

Arthur Gonze Machado[1], Emanuel Antônio Parreiras[2], Gilson Antônio Giraldi[2], and Marcelo Bernardes Vieira[1(✉)]

[1] UFJF - Universidade Federal de Juiz de Fora, Departamento de Ciência da Computação, Juiz de Fora, MG, Brazil
arthur.gonze@ice.ufjf.br, marcelo.bernardes@ufjf.br

[2] LNCC - Laboratório Nacional de Computação Científica, Av. Getúlio Vargas, 333, Quitandinha, Petrópolis, RJ, Brazil
emanuel1@ice.ufjf.br, gilson@lncc.br

Abstract. We propose a mathematical and computational model to address an anisotropic incompressible fluid transport problem. Our method 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 an anisotropic projection step designed to accumulate and adjust pressure as the tensor field changes 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 varies through time. Also, we propose a novel tensor-based 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. Our method is unconditionally stable and presents suitable accuracy for designing incompressible fluid behavior. The experimental results show how anisotropic tensor fields modulate the pressure field, thereby influencing the behavior of the fluid flow.

Keywords: Anisotropic Pressure Segregation · Fluid Simulation · Incompressible Navier-Stokes · Time-Dependent Tensor Fields

1 Introduction

Even with significant advancements in Machine Learning in the last decade, simulating fluids still heavily relies on numerical methods. The computational costs involved in these methods are well known, and their use has numerical errors that need evaluation. Therefore, the Navier-Stokes equations are often simplified according to the objectives of each problem [17]. These assumptions and simplification may introduce several limitations and may not accurately capture all the

M. B. Vieira—Supported by UFJF, FAPEMIG grant APQ-00080-21, CAPES, and Grant Productivity Research Scholarship Level 2 CNPq (307645/2023-1).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
O. Gervasi et al. (Eds.): ICCSA 2024, LNCS 14813, pp. 65–82, 2024.
https://doi.org/10.1007/978-3-031-64605-8_5

details of a physically accurate fluid flow. 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 a version of the incompressible Navier-Stokes equations.

The incompressibility constraint is an important simplification for Computational Fluid Simulation. Physically accurate fluids indeed change their density, e.g. sound wave propagation. But in the macroscopic context, these perturbations in the fluid density are so small that they are irrelevant for several fluid flow applications. In this work, we propose a method that allows for compressible effects, plugged into an incompressible fluid computational method.

Assuming that the fluid flow is isotropic makes the Navier-Stokes equations less difficult to analytical or numerical solutions. The isotropic flow assumes that fluid properties, e.g. viscosity, are considered to be uniform in all orientations. 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).

The "Explicit Integration following the Velocity and Acceleration Streamline" (X-IVAS) method has proven to be a fitting solution for our challenges [7,9]. Unlike other methods, X-IVAS is a P2 Fractional-Step method [6] that incorporates pressure in the advection step by accumulating it throughout the simulation. This accumulated pressure is then transformed into kinetic energy, reducing the numerical dissipation of the momentum. It uses a pressure segregation scheme, which is the process of uncoupling the computation of the pressure from the velocity [2,3,14].

The purpose of this work is to develop a mathematical and computational model that can address anisotropic incompressible fluid transport based on symmetric second-order positive-definite tensor fields. Our model modulates and deviates the pressure and viscosity terms through a tensor field that can change temporally and spatially (Sect. 3.2). Also, we aim to adapt the X-IVAS method within its pressure segregation scheme to obtain anisotropic fluid flow (Sect. 3.3). 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. We provide examples of tensor fields varying in time to modulate fluid flow. One of them uses a fluid property to construct a constantly evolving tensor field. We build the experiments upon existing models presented in the works of [12,16,19] further advancing the understanding and application of our tensor-based X-IVAS method.

The main contributions of this paper 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.

2 Related Works

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 [10].

Recent contributions on anisotropic fluid flow for animations are presented by [12,16,19]. In [16], 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 transforms 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 to solve this novel numerical scheme was created over the [18] contributions. The work of [19] took a step further improving this approach to a 3D domain with custom numerical and computational methods. Finally, [12] proposed a numerical method 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 influence the fluid flow.

Advancements in machine learning and neural networks have also made a significant impact on CFD research. 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). Some notable works include the development of Physics-Informed Neural Networks (PINNs) [15]. These networks receive direct information about the underlying governing equations for training. There is also the application of CNNs for solving these PDEs on an image-based training approach without the need for labeled data [5]. 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.

3 Proposed Method

Our novel method enables pressure segregation while considering time-dependent tensor fields. This section addresses the challenge of incorporating the temporal variations of the tensor field into the pressure segregation process [2,3,14].

The main mathematical fact that supports this work is given by [4]. Any vector field $\mathbf{u}(\mathbf{x})$ can be uniquely decomposed from two arbitrary tensor fields

T and **S** that are symmetric positive-definite dyadics in $\mathbb{R}^3$. This decomposition can be performed by solving the **ST**-Poisson equation:

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

where $\mathbf{p}\left(\mathbf{x}\right)$ is a vector field with continuous second derivatives, $\nabla_{\mathbf{S}}$ and $\nabla_{\mathbf{T}}$ are the **S**-Gradient and **T**-gradient terms, respectively. We refer to [4,19] for more details. As proposed by [16,19], we set $\mathbf{S} = \mathbf{I}$ to obtain the **T**-Poisson equation:

$$\nabla . \nabla_{\mathbf{T}} \mathbf{p}\left(\mathbf{x}\right) = \nabla \cdot \mathbf{u}\left(\mathbf{x}\right), \tag{1}$$

3.1 Navier-Stokes Equations with Tensor Fields

Our approach 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. Our primary goal is to solve the model proposed by [16] that accounts for the anisotropy of a symmetric positive-definite second-order tensor field. The ensuing mathematical formulation is:

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} = \mu \nabla \cdot (\mathbf{T} \nabla \mathbf{u}) - \mathbf{T} \nabla \mathrm{p} + \mathbf{f}, \tag{2}$$

restricted to:

$$\nabla \cdot \mathbf{u} = 0. \tag{3}$$

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. Our interest lies in the modulation of the forces represented on the right-hand side (RHS) of Eq. 2 that act on the fluid.

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 perform both conversions through a second-order positive-definite tensor field. The conversions must be carefully performed to avoid the introduction of artifacts in the simulation and keep it stable.

As shown by [16], and [13], the tensor transformed 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 and deviate the fluid.

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

3.2 Anisotropic Helmholtz Projection and Varying Tensor Fields

Let us consider the anisotropic Helmholtz decomposition, denoted by Eq. 1. It shows how a vector field $\mathbf{u}(\mathbf{x})$ can be decomposed as a scalar $\phi(\mathbf{x})$ and a vector $\mathbf{u}'(\mathbf{x})$ field, subject to the tensor field $\mathbf{T}(\mathbf{x})$. Now suppose that the vector field smoothly varies in time as $\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_t$, and the tensor field smoothly varies as $\mathbf{T}(\mathbf{x}, t) = \mathbf{T}_t$. The anisotropic Helmholtz decomposition at any time t is defined as (Eq. 1 with $\mathbf{S} = \mathbf{I}$):

$$\begin{aligned} \mathbf{u}_t &= \nabla_{\mathbf{T}_t}\phi_t + \nabla \times \mathbf{a}_t, \\ \mathbf{u}_t &= \nabla_{\mathbf{T}_t}\phi_t + \mathbf{u}_t^{'}, \\ \mathbf{u}_t^{'} &= \mathbf{u}_t - \nabla_{\mathbf{T}_t}\phi_t, \end{aligned} \tag{4}$$

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. 1):

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

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:

$$\mathbf{u}_t = \mathbf{u}_{t-1}^{'} + \delta\mathbf{u}_t, \tag{5}$$

whose anisotropic Helmholtz decomposition is:

$$\begin{aligned} \nabla \cdot \nabla_{\mathbf{T}_t}\delta\phi_t &= \nabla \cdot \left(\mathbf{u}_{t-1}^{'} + \delta\mathbf{u}_t\right), \\ \nabla \cdot \nabla_{\mathbf{T}_t}\delta\phi_t &= \nabla \cdot \delta\mathbf{u}_t, \end{aligned} \tag{6}$$

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:

$$\mathbf{u}'_t = \mathbf{u}_{t-1}^{'} + \left[\delta\mathbf{u}_t - \nabla_{\mathbf{T}_t}\delta\phi_t\right], \tag{7}$$

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 Eqs. 6 and 7. In this work, the main problem is to find the accumulated scalar field:

$$\begin{cases} \phi_n = f_{\mathbf{T}_0}(\delta\phi_0) + f_{\mathbf{T}_1}(\delta\phi_1) + \cdots + f_{\mathbf{T}_{n-1}}(\delta\phi_{n-1}) + \delta\phi_n, \\ \nabla_{\mathbf{T}_n}\phi_n \text{ is } \mathbf{T}_n\text{-solenoinal}, \end{cases} \tag{8}$$

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 Eq. 8 is not trivial. Equation 7 shows that all elements in the sequence are coupled by the anisotropic Helmholtz decomposition (Eq. 6). 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. 8) results in a divergence-free vector field sequence that is different from that Eq. 7 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 Eq. 8. 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 Eq. 8 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:

$$\phi_t = g_{\mathbf{T}_{t-1}}(\phi_{t-1}) + \delta\phi_t, \tag{9}$$

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 ϕ_{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 ϕ_{t-1} into another one that is adjusted by the current tensor field $\mathbf{T}_t$. Equation 9 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 Eq. 8.

The solution is to modify the Poisson equation that projects the arbitrary vector field $\delta\mathbf{u}_t$ (Eq. 6). Our solution obtains the accumulated scalar field ϕ_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:

$$\nabla \cdot \nabla_{\mathbf{T}_t}\phi_t = \nabla \cdot \left[\delta\mathbf{u}_t + \nabla_{\mathbf{T}_{t-1}}\phi_{t-1}\right], \tag{10}$$

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:

$$\mathbf{u}'_t = \mathbf{u}'_{t-1} + \left[\delta\mathbf{u}_t + \nabla_{\mathbf{T}_{t-1}}\phi_{t-1} - \nabla_{\mathbf{T}_t}\phi_t\right]. \tag{11}$$

Indeed, the vector field $\nabla_{\mathbf{T}_{t-1}}\phi_{t-1}$ is not necessarily $\mathbf{T}_t$-irrotational. Equation 10 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 ϕ_t is the sum of $\delta\phi_t$, which corresponds to the decomposition of $\delta\mathbf{u}_t$, to the previous accumulation ϕ_{t-1} conformed to $\mathbf{T}_t$. Analogously, Eq. 11 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 10 and 11 can be adapted to solve the anisotropic pressure segregation step in the problem of fluid simulation, as shown in the next section.

3.3 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 the semi-implicit advection model presented in [7,9] is as follows:

$$\begin{cases} \mathbf{x}^{n+t} = \mathbf{x}^n + \int_n^{n+t} \mathbf{U}^n\left(\mathbf{x}^\tau\right) \mathrm{d}\tau, \\ \mathbf{U}_p^{n+t} = \mathbf{U}_p^n + \frac{1}{\rho}\int_n^{n+1} \mathbf{A}^n\left(\mathbf{x}^t\right) + \mathbf{F}^n\left(\mathbf{x}^t\right) \mathrm{dt}. \end{cases} \tag{12}$$

We propose to expand the n-th acceleration field term $\mathbf{A}^n$ as:

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

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, Eq. 12 is numerically integrated. In our work, we employ the explicit three-stage third-order Runge-Kutta time integration scheme. This scheme is denoted as:

$$\begin{cases} \mathbf{k}_1 = f\left(\mathbf{q}_n\right), \\ \mathbf{k}_2 = f\left(\mathbf{q}_n + \frac{1}{2}\Delta t\mathbf{q}_1\right), \\ \mathbf{k}_3 = f\left(\mathbf{q}_n + \frac{3}{4}\Delta t\mathbf{q}_2\right), \\ \mathbf{q}_{n+1} = \mathbf{q}_n + \frac{2}{9}\left(\Delta t\mathbf{k}_1\right) + \frac{3}{9}\left(\Delta t\mathbf{k}_2\right) + \frac{4}{9}\left(\Delta t\mathbf{k}_3\right). \end{cases} \tag{13}$$

where Δt is the simulation time-step. Here, $f(\mathbf{q})$ represents the derivative of $\mathbf{q}$ concerning time, and $\mathbf{q}$ denotes a vector term as position and velocity. 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 θ 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 use a first-order approximation, $\theta = 1$. Therefore, the implicit term delta will consider only the ending point of the integration streamline [1].

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

$$\delta\mathrm{P}\left(\mathbf{x}^{n+1}\right) = \mathbf{T}^{n+1}\nabla\mathrm{P}_p^{n+1} - \mathbf{T}^n\nabla\mathrm{P}^n\left(\mathbf{x}^{n+1}\right). \tag{14}$$

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

$$\mathbf{U}_p^{n+1} = \mathbf{U}^n\left(\mathbf{x}^{n+1}\right) + \hat{\hat{\mathbf{A}}}^n\left(\mathbf{x}^{n+1}\right) + \frac{1}{\rho}\int_n^{n+1} \hat{\mathbf{A}}^n\left(\mathbf{x}^{n+1}\right) dt, \tag{15}$$

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:

$$\hat{\mathbf{A}}^n\left(\mathbf{x}^{n+1}\right) = \mu\nabla\cdot\left[\mathbf{T}^n\nabla\mathbf{U}^n\left(\mathbf{x}^{n+1}\right)\right] - \mathbf{T}^n\nabla\mathrm{P}^n\left(\mathbf{x}^{n+1}\right) + \mathbf{F}^n\left(\mathbf{x}^{n+1}\right). \tag{16}$$

The velocity integration of Eq. 15 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 Eq. 13 is used. This first velocity approximation is denoted as:

$$\hat{\mathbf{U}}_p^{n+1} = \mathbf{U}^n\left(\mathbf{x}^{n+1}\right) + \frac{1}{\rho}\int_n^{n+1} \hat{\mathbf{A}}^n\left(\mathbf{x}^{n+1}\right) dt, \tag{17}$$

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:

$$\begin{cases} \left[\rho - \frac{\Delta t}{\theta}\left(\nabla \cdot \mu \mathbf{T}^{n+1}\nabla\right)\right]\delta\sigma_p^{n+1} = \rho\left[\hat{\mathbf{U}}_p^{n+1} - \mathbf{U}^n\left(\mathbf{x}^{n+1}\right)\right], \\ \hat{\hat{\mathbf{U}}}_p^{n+1} = \hat{\mathbf{U}}_p^{n+1} + \delta\sigma_p^{n+1}, \end{cases} \tag{18}$$

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:

$$\mathbf{U}_p^{n+1} = \hat{\hat{\mathbf{U}}}_p^{n+1} - \frac{1}{\rho}\frac{\Delta t}{\theta}\delta\mathrm{P}\left(\mathbf{x}^{n+1}\right). \tag{19}$$

Applying the divergence operator at both sides of Eq. 19, and considering the divergence-free restriction. The following $\mathbf{T}$-Poisson equation is obtained:

$$\frac{\Delta t}{\theta}\frac{1}{\rho}\nabla \cdot \delta\mathrm{P}\left(\mathbf{x}^{n+1}\right) = \nabla \cdot \hat{\hat{\mathbf{U}}}_p^{n+1}. \tag{20}$$

Expanding this $\mathbf{T}$-Poisson equation with Eq. 14, we have:

$$\nabla \cdot \left[\mathbf{T}^{n+1}\nabla\mathrm{P}_p^{n+1}\right] = \nabla \cdot \left[\frac{\theta\rho}{\Delta t}\hat{\hat{\mathbf{U}}}_p^{n+1} + \mathbf{T}^n\nabla\mathrm{P}^n\left(\mathbf{x}^{n+1}\right)\right], \tag{21}$$

as proposed in Eq. 10. We were only able to achieve this using the contribution presented in Sect. 3.2. Finally, the divergence-free velocity field is obtained as:

$$\begin{cases} \mathbf{U}_p^{n+1} = \hat{\hat{\mathbf{U}}}_p^{n+1} + \frac{1}{\rho}\frac{\Delta t}{\theta}\left[\mathbf{T}^n\nabla\mathrm{P}^n\left(\mathbf{x}^{n+1}\right) - \mathbf{T}^{n+1}\nabla\mathrm{P}_p^{n+1}\right], \\ \nabla \cdot \mathbf{U}^{n+1} = 0. \end{cases} \tag{22}$$

as proposed in Eq. 11. The accumulation of pressure within the solution of the Poisson equation is an important aspect, as discussed in Sect. 3.2. Indeed, Eqs. 21 and 22 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 $\mathbf{I}$-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 $\mathbf{T}^n\nabla\mathrm{P}^n\left(\mathbf{x}^{n+1}\right)$.

3.4 Boundary Conditions

In this work, special attention is given to two types of boundaries: the free-surface boundary and the solid wall boundary. The enforcement of boundary conditions is achieved within the framework of the MAC grid, similar to a fully Eulerian solver. 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 Eq. 21. It affects the velocity divergence that is taken into account during the projection step, modifying the linear system to solve the $\mathbf{T}$-Poisson equation.

Free-Surface Boundary: Since we are simulating a single-phase fluid, the treatment of the boundary is relatively straightforward. In this work, a Dirichlet boundary condition is employed at this interface, setting both the pressure and pressure gradient fields to zero.

Solid Phase Boundary: Following the approach outlined in [12], the solid phase boundary condition is applied when a cell is classified as fluid and has at least two neighboring cells that are solid. We use the no-slip condition [1], where the fluid velocity on the face or edge of the grid cell forming an interface is set to the solid velocity in the interface's normal direction.

In addition to the RHS of Eq. (21), 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.

3.5 Implementation

In this section, we present the two core aspects to implement the solution for our proposed method: the particle velocity update scheme and the extended MAC Grid which is used to represent the domain.

Implementation on an Extended MAC Grid: Our Tensor X-IVAS method is discretized on an extended 3D staggered MAC grid, which was proposed by [12]. It provides a higher level of accuracy for solving the anisotropic **T**-Poisson (Eq. 1) 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 **T**-Poisson problem (Eq. 1).

Particle Velocity Update: The particle velocity update scheme combines our Tensor X-IVAS method with the FLIP updating scheme as follows:

$$\mathbf{u}^{n+1} = \mathbf{u}_p^{n+1} + (1-\alpha)\left(\hat{\mathbf{u}}^{n+1} - \tilde{\hat{\mathbf{u}}}_p^{n+1}\right), \tag{23}$$

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 α 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 α 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 numerical diffusion.

4 Experiments with Time-Dependent Tensor Fields

In this section, we present qualitative experimental results generated by the proposed method. All tensor fields considered for the following experiments vary in time.

The fluid density ρ, the time step Δt, and the cell width Δx 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 Sect. 4.2, 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. This avoids the interference of non-null viscosity in the analysis of our method's ability to anisotropically modulate and deviate 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 Sect. 3.4. 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.

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

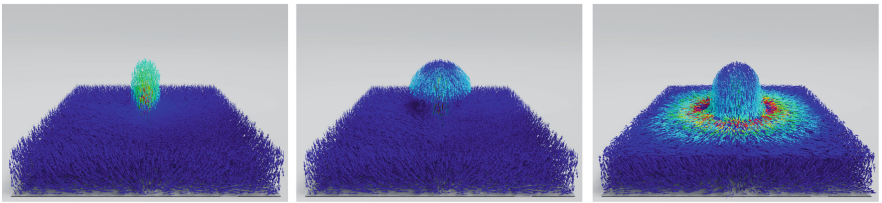
The experiments were performed on an Intel Xeon E5-4607 @ 2.20 GHz processor with 32 GB RAM DDR3.

4.1 Simulation with Axis-Aligned Planar Tensors

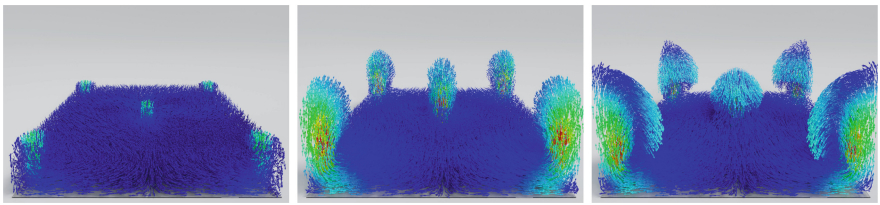
The purpose of this experiment is to illustrate the impact of abruptly modifying the tensor field configuration on pressure relief. This modification allows us to redirect the fluid flow in specific directions given by the tensor fields. All tensors eigenvectors are axis-aligned with $\mathbf{e}_1$ and $\mathbf{e}_2$ aligned with the xy-axis (planar tensor).

We provide two simulations with distinct values for the α term in Eq. 23. The first simulation sets the α term to 1, whereas the second simulation sets it to 0.15. We intend to highlight the influence of the proposed update velocity equation on the simulation of fluid flow and its numerical dissipation.

In each simulation, 12 planar tensor layers are positioned at the bottom of our domain ($k \in [1, 12]$). These layers restrict pressure exchange to the xy-plane. We set a λ_3 eigenvalue of 10^{-3} to the planar tensors, decreasing pressure exchange in the z direction by a factor of 10. Also, 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, the planar layers have a single central hole, allowing the fluid to exchange pressure



(a) Frame 40 (b) Frame 110 (c) Frame 205



(d) Frame 405 (e) Frame 450 (f) Frame 500

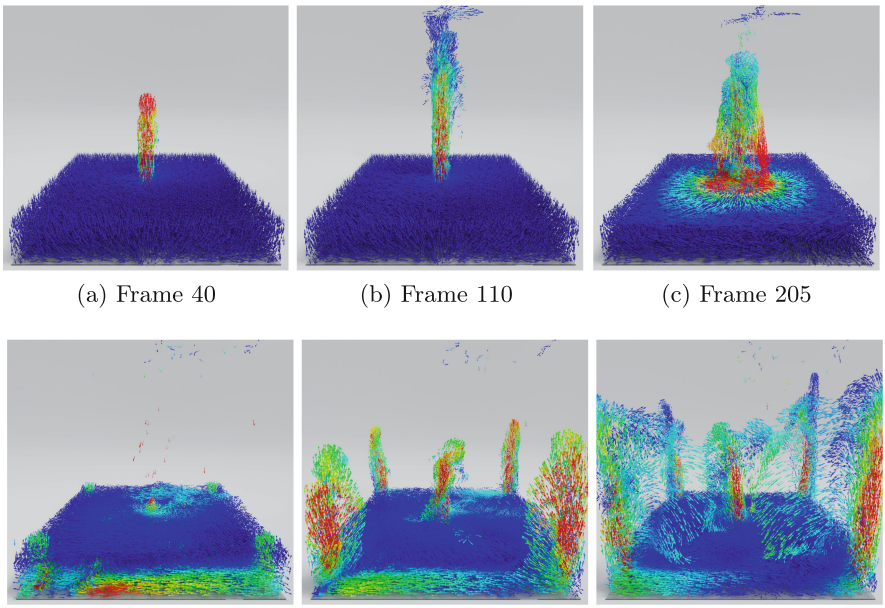
Fig. 1. Geyser simulation with $\alpha = 1$.

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.

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, there is a total of $792,000$ particles.

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 1 and 2 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.

When comparing Figs. 1 and 2, the geyser altitudes and arrow colors demonstrate the impact of the velocity update and the tensor field variation in time. The arrow colors of the fluid in Fig. 2 appear more reddish, indicating that the pressure relief in the experiment with 0.15 yields a higher acceleration compared to the $\alpha = 1$ simulation. Consequently, the simulation with $\alpha = 1$ (Fig. 1) reaches a significantly lower altitude compared to the $\alpha = 0.15$ simulation. This is evident by comparing both simulations on frames 110 and 500. Frames 205 and 405 show 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



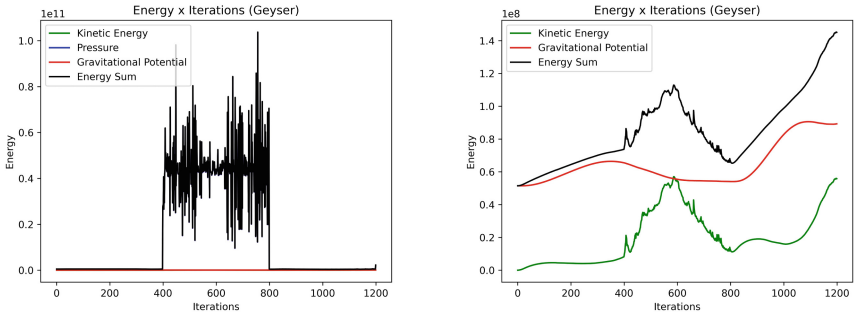
(a) Frame 40 (b) Frame 110 (c) Frame 205

(d) Frame 405 (e) Frame 450 (f) Frame 500

Fig. 2. Geyser simulation with $\alpha = 0.15$.

kinetic energy is distributed, limiting the height it can reach when compared to a geyser with a single hole. This is less evident with $\alpha = 1$ (Fig. 1).

The graphs in Fig. 3 display the integration of the Bernoulli equation throughout the simulation. They are only for visualization purposes since our method does not conserve mass and momentum. The Bernoulli equation states that the sum of the kinetic, potential, and pressure energy per unit mass must remain constant along a streamline for the system to conserve energy. In Fig. 3a, the pressure energy is three orders of magnitude higher than the other energies from iteration 400 when the central hole is closed. This is due to the $\lambda_3 = 10^{-3}$ eigenvalue which compresses the fluid in z direction. As expected, the kinetic energy (green curve in Fig. 3b) exhibits a substantial increase after adding holes in the domain (steps 0 and 800) due to pressure relief. It is followed by a decline and then a rise, when the ejected fluid gets back to the bottom. Furthermore, both gravitational potential and kinetic energy experience an uptick when five holes appear in the domain (step 800) because a greater volume of fluid is put in motion.



(a) Bernoulli equation integration

(b) Integration without pressure

Fig. 3. Bernoulli Equation Integration for the geyser simulation in Fig. 2. (Color figure online)

4.2 Simulation with the Strain-Rate Tensors

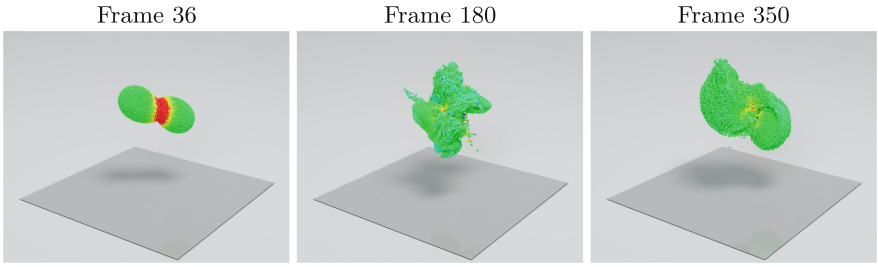
This subsection proposes a continuous adjustment to the tensor field across the entire fluid simulation. The primary objective is to demonstrate the utilization of the tensor field to influence the pressure of fluid flow throughout each step of the simulation. Our focus lies in ensuring that the fluid alleviates its pressure in the direction of deformation while limiting pressure exchange in other directions.

In this experiment, the strain-rate tensor is used to design the tensor field for this experiment. Our method, outlined in Sect. 3.2, mandates that our tensor must be positive-definite. This limitation can be resolved by integrating the strain-rate tensor in time, generating:

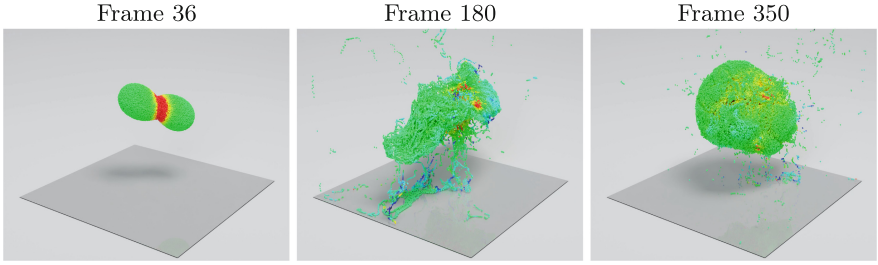
$$\mathbf{T} = \sum_{i=1}^{M} \mathrm{e}^{\gamma(\Delta t \lambda_i)} \mathbf{e}_i \mathbf{e}_i^{\mathrm{T}}, \tag{24}$$

where λ_i and $\mathbf{e}_i$ are the i-th eigenvalues and eigenvectors of the strain-rate tensor, respectively. The scalar factor parameter, denoted as γ, is incorporated into the equation 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 i-th eigenvalue surpasses 1, the tensor will enhance pressure relief along the i-th axis.

We present two 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 Fig. 4a. Subsequently, another simulation is conducted with a strain rate tensor with $\gamma = 5$ as we want to emphasize the tensor's impact on the fluid flow. This simulation is presented in Fig. 4b.



(a) Simulation with isotropic tensor field ($\mathbf{T} = \mathbf{I}$)



(b) Using anisotropic tensor fields from strain rate tensors with $\gamma = 5$ (Eq. 24)

Fig. 4. Simulating two colliding spheres with radial gravitational force.

The velocity update term, denoted as α, has a fixed value of 0.15. Additionally, the gravity force is directed towards the center of the simulation domain, with a magnitude of 10 $\mathrm{m/s^2}$.

The experiment consists of two diagonally opposed spheres of fluid. 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 $\mathbf{u}_1 = [-5, 0, -5]$.

Upon comparing the frames of the reference simulation depicted in Fig. 4a with the experiment with $\gamma = 5$ depicted in Fig. 4b the influence of the strain-rate tensors is evident. 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, radial 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.

Table 1. Summary of experiments setup and simulation times

Experiment	Parameter			Simulation	
	# particles	Tensor	α	Sec./frame	Total time
Geysers 1 (Fig. 1)	792,000	Varying aligned planar	1.0	20.40	06:47:55
Geysers 2 (Fig. 2)	792,000	Varying aligned planar	0.15	20.69	06:53:52
Colliding Spheres (Fig. 4a)	180,956	Strain	0.15	1.89	00:37:52
Colliding Spheres $\gamma = 5$ (Fig. 4b)	180,956	Strain	0.15	6.70	02:13:58

4.3 Discussion

The core parameters of our experiments are presented in Table 1. It includes the computational time cost for each experiment. It increases significantly when dealing with time-varying tensor fields. The main factor is the high-cost computation of the Laplacian mask proposed by [12]. Nevertheless, a constant tensor field throughout the simulation keeps the time cost manageable.

Throughout all experiments, and despite introducing a whole new computational complexity on the computation of the tensor field, our method yields very low average magnitude error around 10^{-11} for all divergence-free vector fields. This is possible by the discretization proposed by [12] on an extended MAC grid. For the sake of repeatability, our source code and all videos of experiments are available in www.gcg.ufjf.br/files/tensor-xivas.

The proposed method yielded appealing results across our experiments. While we maintained fixed grid resolution and time step, 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, 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 Sect. 4.1. The utilization of linear tensor fields is valuable when the pressure exchange needs to occur in specific directions, as shown in Sect. 4.2. 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 [7]. 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 Sect. 4.1. 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.

5 Conclusion

We present a mathematical and computational model that deals with anisotropic incompressible fluid transport. The model is based on second-order positive-definite tensor fields that can change over time. We propose a new numerical method 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.

We also propose a novel computational method for tensor-based anisotropic decomposition (Eq. 21). It is designed to accumulate and adjust pressure as the tensor field evolves in time (Sect. 3.2). Unlike previous literature works [7–9,11], our model was discretized using the extended MAC grid proposed by [12]. 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. 22).

In Sect. 4, we explored different parameter and tensor field variations for comparison purposes. Our results show that our method succeeds in anisotropic modulation of the pressure term. In future works, we aim to incorporate tensor influence into our advection scheme, without compromising numerical accuracy and stability. Designing tensor fields to modulate and deviate 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, mathematical models are needed to design meaningful tensor fields, application-based tensor fields, and perhaps families of tensor fields with valuable differential properties.

As discussed in Sect. 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, flow simulations with isotropic methods will be included in future works for comparison purposes.

References

1. Bridson, R.: Fluid simulation for computer graphics. CRC press (2015)
2. Codina, R., Badia, S.: On some pressure segregation methods of fractional-step type for the finite element approximation of incompressible flow problems. Comput. Methods Appl. Mech. Eng. **195**(23), 2900–2918 (2006)
3. Codina, R., Soto, O.: Approximation of the incompressible Navier-stokes equations using orthogonal subscale stabilization and pressure segregation on anisotropic finite element meshes. Comput. Methods Appl. Mech. Eng. **193**(15), 1403–1419 (2004)
4. Dassios, G., Lindell, I.V.: Uniqueness and reconstruction for the anisotropic Helmholtz decomposition. J. Phys. A: Math. Gen. **35**(24), 5139 (2002)

5. Gao, H., Sun, L., Wang, J.X.: Phygeonet: physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state pdes on irregular domain. J. Comput. Phys. **428**, 110079 (2021)
6. Gresho, P.M.: 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. Int. J. Num. Methods Fluids **11**(5), 587–620 (1990)
7. Idelsohn, S., Nigro, N., Limache, A., Oñate, E.: Large time-step explicit integration method for solving problems with dominant convection. Comput. Methods Appl. Mech. Eng. **217**, 168–185 (2012)
8. Idelsohn, S.R., Marti, J., Becker, P., Oñate, E.: Analysis of multifluid flows with large time steps using the particle finite element method. Int. J. Numer. Meth. Fluids **75**(9), 621–644 (2014)
9. Idelsohn, S.R., Nigro, N.M., Gimenez, J.M., Rossi, R., Marti, J.M.: A fast and accurate method to solve the incompressible Navier-stokes equations. Engineering Computations (2013)
10. Marston, J., Tobias, S.: Recent developments in theories of inhomogeneous and anisotropic turbulence. Annu. Rev. Fluid Mech. **55**, 351–375 (2023)
11. Nadukandi, P., Servan-Camas, B., Becker, P.A., García-Espinosa, J.: Seakeeping with the semi-Lagrangian particle finite element method. Comput. Particle Mech. **4**, 321–329 (2017)
12. Parreiras, E.A.: Métodos Particle-in-Cell para Simulação de Fluidos Anisotrópico. Master's thesis, Universidade Federal de Juiz de Fora, Juiz de Fora, MG (fev 2022)
13. Parreiras, E.A., Vieira, M.B., Machado, A.G., Renhe, M.C., Giraldi, G.A.: A particle-in-cell method for anisotropic fluid simulation. Comput. Graph. **102**, 220–232 (2022)
14. Perot, J.B.: An analysis of the fractional step method. J. Comput. Phys. **108**(1), 51–58 (1993)
15. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. **378**, 686–707 (2019)
16. Renhe, M.C., Vieira, M.B., Esperança, C.: A stable tensor-based method for controlled fluid simulations. Appl. Math. Comput. **343**, 195–213 (2019)
17. Spurk, J., Aksel, N.: Fluid mechanics. Springer Science and Business Media (2007)
18. Stam, J.: Stable fluids. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, pp. 121–128 (1999)
19. Vieira, M.B., Giraldi, G.A., Ribeiro, A.C.A., Renhe, M.C., Esperança, C.: Anisotropic Hdelmholtz decomposition for controlled fluid simulation. Appl. Math. Comput. **411**, 126501 (2021)