

UNIVERSIDADE FEDERAL DE JUIZ DE FORA  
INSTITUTO DE CIÊNCIAS EXATAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Emanuel Antônio Parreiras

Métodos *Particle-in-Cell* para Simulação Anisotrópica de Fluidos

Juiz de Fora

2022

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

Parreiras, Emanuel Antônio.

Métodos *Particle-in-Cell* para Simulação Anisotrópica de Fluidos /  
Emanuel Antônio Parreiras. – 2022.

69 f. : il.

Orientador: Marcelo Bernardes Vieira

Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Instituto  
de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computa-  
ção, 2022.

1. Simulação de Fluidos. 2. Decomposição de Helmholtz Anisotrópico.  
3. Particle-in-Cell. 4. Equação de Poisson Tensorial. I. Vieira, Marcelo  
Bernardes, orient. II. Título.

Emanuel Antônio Parreiras

Métodos *Particle-in-Cell* para Simulação Anisotrópica de Fluidos

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.

Orientador: Prof. Dr. Marcelo Bernardes Vieira

Juiz de Fora

2022

**Emanuel Antônio Parreiras**

**Métodos Particle-in-Cell para Simulação Anisotrópica de Fluidos**

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 21 de fevereiro de 2022.

**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, 08/02/2022.



Documento assinado eletronicamente por **Marcelo Bernardes Vieira, Professor(a)**, em 21/02/2022, às 16:32, 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 21/02/2022, às 16:32, 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 21/02/2022, às 16:32, 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 **Mario Antonio Ribeiro Dantas, Professor(a)**, em 23/02/2022, às 18:35, 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](http://www2.ufjf.br/SEI)) através do ícone Conferência de Documentos, informando o código verificador **0671789** e o código CRC **D9B466D5**.

---

## AGRADECIMENTOS

Durante o final de mais uma difícil etapa, gostaria de agradecer às pessoas que de alguma forma contribuíram ao longo do curso e principalmente neste trabalho. Primeiramente gostaria de agradecer aos meus pais, Regina Célia Pereira e Manoel Soares Parreiras (*in memoriam*), que desde a minha infância lutaram para eu ter uma educação de qualidade e que eu pudesse chegar onde cheguei. Agradeço também ao meu orientador, Marcelo Bernardes Vieira, pela paciência e suporte em todo o curso. Agradeço ao Arthur Gonze, Marcelo Caniato Renhe, Gilson Antônio Giraldi e ao meu orientador, que contribuíram para o desenvolvimento e escrita do artigo publicado. Gostaria também de agradecer a CAPES, já que a conclusão deste curso apenas foi possível com o seu suporte financeiro.

## RESUMO

Este trabalho apresenta dois métodos para redirecionar um fluxo de fluido utilizando campos tensoriais em simulações baseadas em métodos *particle-in-cell*. O nosso foco é usar o campo tensorial como um meio para defletir fluidos para geração de efeitos em Computação Gráfica. Este trabalho adapta a formulação clássica do método *Fluid-Implicit-Particle* (FLIP) incluindo a informação dos tensores durante os passos da projeção e advecção. São apresentadas duas novas abordagens numéricas para a resolução da projeção anisotrópica, uma baseada em faces e outra baseada em faces e arestas. O desenvolvimento desses métodos visa garantir uma simulação estável, suave e livre de divergente. Para tal, são empregadas duas estruturas em grade escalonada. A grade *Marker-and-Cell* (MAC) clássica é utilizada para armazenar o campo de velocidades e para encontrar o campo projetado pela projeção baseada em face. Entretanto, para comportar valores armazenados nas arestas, uma extensão para essa categoria de grade é proposta, sendo também utilizada na projeção baseada em face e aresta. Foi proposta uma nova advecção baseada no método Runge-Kutta de ordem três, onde incorporamos a deflexão tensorial como um campo de aceleração. Também é proposta uma força externa que visa balancear os efeitos do tensor na borda fluido-vazio. Tal força age para evoluir o fluxo na região, fazendo com que o campo tensorial exerça um efeito de meio permeável. Para mostrar as capacidades de cada abordagem, são realizados testes individuais e com diferentes categorias de campos tensoriais. Além disso, um experimento comparativo é apresentado para analisar os aspectos visuais e numéricos de cada método. Com esses experimentos é possível entender como diferentes tipos de tensores afetam o comportamento do fluido nas diferentes abordagens.

Palavras-chave: Simulação de Fluidos. Decomposição de Helmholtz Anisotrópico. Particle-in-Cell. Equação de Poisson Tensorial.

## ABSTRACT

This work presents two methods to redirect a fluid flow using tensor fields in *particle-in-cell* based simulations. Our focus is use tensor fields as medium to deflect fluids for computer graphics effects. This work adapts the classical formulation of the Fluid-Implicit-Particle (FLIP) method including the information of the tensors during the projection and advection steps. Two new numerical approaches for solving anisotropic projection will be presented, one based on faces and other based on faces and edges. The development of these methods aims to ensure a stable, smooth and divergence-free simulations. For this purpose, two staggered grid structures are employed. The classic Marker-and-Cell (MAC) grid is used to store the velocity field and to find the field projected by the face-based projection. However, to accommodate values stored on edges, an extension to this grid category is proposed, which is used in face and edge based projection. A new third order Runge-Kutta method is proposed, where we incorporate the tensor deflection as an acceleration field. An external force is also proposed to balance the tensor effects on the fluid-empty boundary. Such a force acts to evolve the flow in the region, causing the tensor field to exert an effect on a permeable medium. To show the capabilities of each approach, individual tests will be executed with different categories of tensor fields. In addition, a comparative experiment is presented to analyse the visual and numerical aspects of each method. With the experiments it is possible to understand how the different types of tensors affect the fluid with different approaches.

Keywords: Fluid Simulation. Anisotropic Helmholtz Decomposition. Particle-in-Cell.  $\mathbf{T}$ -Poisson Equation.

## LISTA DE ILUSTRAÇÕES

Região do volume de controle $\Omega$ e com bordo $\Gamma$ . . . . .	19
Figura 3 – Operadores $\mathbf{T}$ -Gradiente e $\mathbf{T}$ -Laplaciano face-célula. . . . .	30
Figura 4 – Ilustração de uma máscara $\mathbf{T}$ -Laplaciana face-célula 2D. . . . .	31
Figura 5 – Exemplos de máscaras 2D pré-computadas. . . . .	31
Figura 6 – Gradiente de $p$ na projeção. . . . .	34
Figura 7 – Faces, arestas e normais de uma célula. . . . .	38
Figura 8 – Operador $\mathbf{T}$ -Gradiente para faces e arestas . . . . .	39
Figura 9 – Piscina de fluido interagindo com tensores planares. . . . .	47
Figura 10 – Cubo de fluido caindo sob uma camada de tensores planares. . . . .	48
Figura 11 – Uma barragem permeável formada por tensores . . . . .	49
Figura 12 – Visão ortográfica da simulação da barragem. . . . .	50
Figura 13 – Tensores planares sob uma curvatura variável. . . . .	52
Figura 14 – Campo tensorial isotrópico com ganho variável. . . . .	53
Figura 15 – Simulação em um campo tensorial isotrópico com ganho variável. . . . .	53
Figura 16 – Campo tensorial planar utilizado no experimento. . . . .	55
Figura 17 – Simulação de um campo isotrópico e um planar sob efeito de um campo de força radial. . . . .	55
Figura 18 – Uma lâmina do campo tensorial linear circular . . . . .	56
Figura 19 – Simulação com advecção e condição de superfícies alternadas. . . . .	57
Figura 20 – Simulação com campo linear descontínuo . . . . .	58
Figura 21 – Foco na interação fluido-tensor . . . . .	59
Figura 22 – Método baseado em face com tensor linear alinhado . . . . .	60
Figura 23 – Borda fluido-vazio para o método baseado em faces . . . . .	60
Figura 24 – Método baseado em faces e arestas com tensor linear alinhado . . . . .	61
Figura 25 – Borda fluido-vazio para o método baseado em faces . . . . .	62
Figura 26 – Divergente absoluto máximo dos métodos . . . . .	62
Figura 27 – Soma dos divergentes absolutos dos métodos . . . . .	63

## LISTA DE TABELAS

Tabela 1 – Resumo dos parâmetros e tempos computacionais dos experimentos. . 63

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>11</b>
1.1	TRABALHOS RELACIONADOS . . . . .	11
1.2	PROPOSTA . . . . .	13
1.3	ORGANIZAÇÃO . . . . .	14
<b>2</b>	<b>FUNDAMENTOS . . . . .</b>	<b>15</b>
2.1	TENSORES . . . . .	15
<b>2.1.1</b>	<b>TENSORES DE ORIENTAÇÃO . . . . .</b>	<b>15</b>
2.2	DESCRIÇÃO DE UM FLUXO . . . . .	16
2.3	DERIVADA MATERIAL . . . . .	16
2.4	TENSOR TENSÃO DE CAUCHY . . . . .	17
2.5	CONSERVAÇÃO DO MOMENTO . . . . .	18
<b>2.5.1</b>	<b>EQUAÇÕES DE NAVIER-STOKES . . . . .</b>	<b>20</b>
2.6	EQUAÇÕES DE NAVIER-STOKES PARA MEIOS ANISOTRÓPICOS	23
2.7	SOLUÇÃO NUMÉRICA DAS EQUAÇÕES DE NAVIER STOKES . .	24
<b>3</b>	<b>MÉTODO BASEADO EM FACES . . . . .</b>	<b>28</b>
3.1	DISCRETIZAÇÃO DA PROJEÇÃO ANISOTRÓPICA . . . . .	29
<b>3.1.1</b>	<b>Operadores T-Gradiente e T-Laplaciano . . . . .</b>	<b>29</b>
<b>3.1.2</b>	<b>Máscara T-Laplaciana . . . . .</b>	<b>30</b>
<b>3.1.3</b>	<b>Sistema Linear para o Problema de T-Poisson . . . . .</b>	<b>32</b>
<b>3.1.4</b>	<b>Condições de Borda . . . . .</b>	<b>34</b>
3.2	EVOLUÇÃO DA SUPERFÍCIE DO FLUIDO . . . . .	34
3.3	ADVECCÃO ANISOTRÓPICA . . . . .	35
3.4	VISÃO GERAL DOS PASSOS DA SIMULAÇÃO . . . . .	36
<b>4</b>	<b>MÉTODO BASEADO EM FACES E ARESTAS . . . . .</b>	<b>37</b>
4.1	DISCRETIZAÇÃO DA PROJEÇÃO ANISOTRÓPICA . . . . .	37
<b>4.1.1</b>	<b>Operadores T-Gradiente e T-Laplaciano . . . . .</b>	<b>38</b>
<b>4.1.2</b>	<b>Máscara T-Laplaciana . . . . .</b>	<b>40</b>
<b>4.1.3</b>	<b>Condições de Borda . . . . .</b>	<b>41</b>
<b>4.1.4</b>	<b>Sistema Linear para o Problema de T-Poisson . . . . .</b>	<b>42</b>
4.2	EVOLUÇÃO DA SUPERFÍCIE DO FLUIDO . . . . .	44
4.3	VISÃO GERAL DOS PASSOS DA SIMULAÇÃO . . . . .	44
<b>5</b>	<b>EXPERIMENTOS E DISCUSSÕES . . . . .</b>	<b>46</b>
5.1	EXPERIMENTOS COM A PROJEÇÃO BASEADA EM FACES . . .	46
<b>5.1.1</b>	<b>Simulação com tensores planares . . . . .</b>	<b>46</b>
<b>5.1.2</b>	<b>Simulação com tensores lineares . . . . .</b>	<b>50</b>
<b>5.1.3</b>	<b>Tensores planares sob uma superfície de curvatura variável . .</b>	<b>51</b>
<b>5.1.4</b>	<b>Campo tensorial isotrópico . . . . .</b>	<b>53</b>

5.2	EXPERIMENTOS COM A PROJEÇÃO BASEADA EM FACES E ARESTAS . . . . .	54
5.2.1	Comparação entre campo isotrópico e planar . . . . .	54
5.2.2	Simulação com advecção e condição de superfícies alternadas .	56
5.2.3	Campo linear descontínuo . . . . .	58
5.3	COMPARAÇÃO ENTRE AMBOS OS MÉTODOS . . . . .	59
5.4	DISCUSSÃO . . . . .	62
6	CONCLUSÃO . . . . .	65
	REFERÊNCIAS . . . . .	66

## 1 INTRODUÇÃO

No cotidiano, diversos fenômenos físicos que presenciamos se apresentam como escoamento de fluidos, como o vento, a água, a fumaça, fogo e etc. Modelos matemáticos que tentam descrever o comportamento de fluidos são baseados nas leis de conservação de massa e momento, um desses modelos é conhecido como Equações de Navier-Stokes para escoamentos incompressíveis [Batchelor, 2000]. É comum utilizarmos a técnica de *splitting* para resolução de Equações Diferenciais Parciais (EDP's), o que implica resolver cada um dos fenômenos do modelo individualmente. Esse tipo de método tende a gerar um erro numérico maior nas soluções obtidas [Ferziger et al., 2002], entretanto, conseguimos simplificar o processo de resolução dos modelos. Para evitar esse tipo de dissipação numérica, métodos mais robustos tendem a resolver todos os fenômenos de forma conjunta, como por exemplo o algoritmo SIMPLE [Ferziger et al., 2002] para solução das Equações de Navier-Stokes. Se desejarmos reproduzir um ambiente real em um espaço virtual, será necessário incluir algum nível de simulação física para que possamos obter um nível adequado de foto-realismo. Simulações voltadas para a área de Computação Gráfica (CG) almejam construir animações visualmente convincentes, o que não necessariamente implica em uma simulação com precisão elevada. Portanto, a técnica de *splitting* se mostra vantajosa para a área.

Durante a produção de uma cena em CG, é comum que o artista digital utilize diversas categorias de efeitos, alguns desses sendo produzidos a partir de simulações de fluidos. Esse tipo de efeito pode ser observado no filme Moana [Frost et al., 2017] da Disney. Podemos ver diversas cenas onde o artista consegue controlar o escoamento de água a sua livre escolha. Esse tipo de ferramenta vem sendo desenvolvida na área há décadas para criar ferramentas cada vez mais elegantes, permitindo um controle mais fino do escoamento [Lu et al., 2019]. Recentemente houveram diversos trabalhos que introduziram [Renhe et al., 2019] e aprimoraram [Vieira et al., 2021] uma nova forma de controle. Esses trabalhos visam dominar o escoamento através de um campo tensorial que deforma o meio onde o fluido é transportado, fazendo com que seja possível criar novos tipos de efeitos que são difíceis, ou impossíveis, utilizando métodos desenvolvidos anteriormente a eles. Esse tipo de ferramenta pode ser utilizada para simular desde escoamentos em meios permeáveis a até em visualização científica, como para visualizar campos tensoriais. Campos esses que podem ser obtidos através de amostragem de propriedades físicas, desde a permeabilidade de um material, ou através de exames de ressonância magnética.

### 1.1 TRABALHOS RELACIONADOS

Diversos métodos foram propostos ao longo dos anos, desde um controlador incorporado ao método [Foster and Metaxas, 1997] ao uso de quadro-chave para controlar

simulações de fumaça [Treuille et al., 2003]. Entre esses trabalhos, os mais próximos com a proposta deste texto são de controlar o fluxo de fluido ou contê-lo em uma determinada região do espaço. Alguns exemplos são Foster and Fedkiw [2001], Kim et al. [2006], Nielsen and Christensen [2010] e McNamara et al. [2004], Fattal and Lischinski [2004], Shi and Yu [2005] respectivamente.

Pan and Manocha [2017] propuseram um algoritmo de otimização para computar sequencias de campos de força, baseados em quadros-chave, com o intuito de produzir formas esfumaçadas. Apesar dos autores afirmarem que o método deles é mais rápido que outros métodos baseados em otimização, simulações desse tipo ainda são computacionalmente custosas. Lu et al. [2019] entretanto, propuseram um esquema diferente para simulação de fluidos controlada. Eles utilizaram como inspiração a animação esquelética e implementaram dois passos: *fluid rigging*, baseado em uma nuvem de pontos que pode ser manipulada pelo usuário, e *fluid skinning*, que produz um fluxo plausível baseado em decomposição espectral. A ideia é reforçar as baixas frequências no campo de força. Os autores afirmam que o método deles melhora o controle em 2D e 3D, eliminando a necessidade de otimização dos parâmetros. Alguns trabalhos recentes com partículas usam o método de *Smoothed Particle Hydrodynamics* (SPH) [Lind et al., 2020], que visa controlar o fluido por meio da aplicação de restrições à deformação nas partículas [Feng and Liu, 2018], ou usando forças de mola para conduzir o fluido até um dado alvo de interesse [Zhang and Liu, 2018].

Inspirado por trabalhos que combinam técnicas tensoriais e as equações de Navier-Stokes para visualização de tensores de difusão obtidos através de ressonância magnética Hageman et al. [2006], um novo método para controle de fluidos através de campos tensoriais foi proposto em Renhe et al. [2019]. Os tensores são acoplados às equações de Navier-Stokes de fluxo, agindo como parte integrante do meio. Nesse trabalho, o fluido percorre caminhos que surgem naturalmente sob camadas do campo tensorial. Portanto, o fluido não é rigorosamente restringido a caminhos ou superfícies, nem mesmo sob influência de campos de força pré-definidos.

O fluido pode, entretanto, ser representado por uma abordagem híbrida, onde cada célula da grade contém um número específico de partículas. Métodos que trabalham com essa categoria de configuração são geralmente chamados Métodos *Particle-in-Cell* (PIC), referindo-se a abordagens híbridas que trabalham com partículas ao lado de células de grade. Os dados do fluido não são representados unicamente pela grade, passos adicionais são necessários para manipular os valores de entre células e partículas. O método PIC original [Evans et al., 1957] inicia a simulação com as quantidades armazenadas nas partículas. Essas quantidades são transportadas para a grade, onde todos os passos não-advectivos (difusão, projeção e forças externas) acontecem. Quando a advecção ocorre, as partículas têm suas posições e velocidades atualizadas com base na velocidade da grade. A ideia por trás de métodos PIC é obter as vantagens de ambas as abordagens, Euleriana

e Lagrangiana. Entretanto, como a cópia de todos os dados envolvem interpolação, é esperado um aumento no erro numérico do sistema, introduzindo como resultado uma viscosidade artificial nas simulações.

Brackbill et al. [1988] tentaram minimizar o problema da dissipação numérica dita acima, introduzindo uma variação do método original chamado FLIP. A interpolação usada durante a transferência grade-partícula consideram não apenas os novos valores da grade projetada, mas também a grade de antes dos passos não advectivos. Portanto, em cada iteração, logo após a transferência das velocidades da partícula para a grade, o método FLIP salva uma cópia dessas velocidades para uso futuro. Os próximos passos da simulação, exceto para a advecção são executadas na grade, tal como no método PIC. Quando ocorre a advecção, entretanto, as novas velocidades computadas na grade são subtraídas das velocidades salvas anteriormente e a diferença é interpolada e adicionada às partículas.

O método FLIP é, portanto, uma maneira de contornar a dissipação numérica adicional obtida com o PIC, entretanto, algumas instabilidades são adicionadas no processo. O fator  $\alpha$  é utilizado como balanceamento de quanto da solução original do PIC é preservada no FLIP, de modo a reduzir instabilidades e obter uma simulação menos viscosa. Jiang et al. [2015] tentaram resolver este problema prevenindo a perda de informação na representação da velocidade. A abordagem que eles utilizaram foi estendida em Fu et al. [2017]. O foco do trabalho é a integração do método FLIP com tensores.

Esse trabalho segue Renhe et al. [2019], onde as equações de Navier-Stokes são adotadas para restringir o fluxo de fluido de acordo com a geometria do campo tensorial. Como em Renhe et al. [2019], os fluxos de fluido são direcionados pelas propriedades do meio, representadas pelo campo tensorial, sem a necessidade de especificar campos de força ou caminhos para obter o mesmo efeito na simulação. Entanto, nesse trabalho, mecanismos de controle dependiam de modelos de simulação mais simples com velocidades centradas na célula e um passo de projeção com restrições de estabilidade a campos tensoriais mais desafiadores.

## 1.2 PROPOSTA

Mesmo com avanços na área de simulação anisotrópica de fluidos, os trabalhos desenvolvidos até hoje ainda não conseguem resolver adequadamente o campo de pressão para meios com anisotropia elevada. O objetivo deste trabalho é oferecer novas formas de se computar o *splitting* da pressão para meios anisotrópicos, que é fundamental para simulação, já que é a partir dele que conseguimos garantir a lei de conservação de massa, encontrando um campo de velocidades com divergente nulo. Será utilizado como base o trabalho de Vieira et al. [2021] com o intuito de prover formulações matemáticas mais consistentes. Um exemplo é a definição do limite teórico para o campo tensorial de acordo

com o teorema da decomposição de Helmholtz anisotrópica, onde podemos garantir a existência e unicidade da solução para o campo de pressão.

Neste trabalho, são propostas duas novas discretizações para o *splitting* da pressão, uma baseada em diferenças finitas e outra baseada em volumes finitos, além da introdução de novas condições de borda de Neumann tensorial para pressão. Discretizações essas mais simples de se implementar do que as anteriores e entregando resultados melhores ou similares.

Diferentemente dos trabalhos anteriores como Renhe et al. [2019] e Vieira et al. [2021], que utilizam da simulação de fumaça para demonstrar o método, este trabalho irá se limitar a simulação de líquidos utilizando o método FLIP como base. Outra diferença significativa neste trabalho é o uso de uma grade MAC para armazenamento do campo de velocidades ao invés de uma grade centrada na célula, além de utilizar uma extensão da grade MAC clássica, nos permitindo armazenar e tratar valores também nas arestas das células. Além disso, a grade MAC contribui para uma simulação menos dissipativa e mais estável. Por fim, o método Runge-Kutta é modificado para computar a influência do campo tensorial, evitando um passo numérico extra usado em Renhe et al. [2019]. Nos capítulos seguintes será discutido estas contribuições com mais detalhes.

### 1.3 ORGANIZAÇÃO

O restante do trabalho será dividido da seguinte forma: Capítulo 2 apresenta uma descrição de conceitos principais para o entendimento do trabalho; os capítulos 3 e 4 descrevem os métodos propostos testados e discutidos no capítulo 5. Durante o capítulo 6 será apresentado a conclusão e a possibilidade de trabalhos futuros.

## 2 FUNDAMENTOS

Neste capítulo será discutido os conceitos de conservação de massa e momento (seção 2.5), além de introduzir a noção de tensão através tensor de Tensões de Cauchy (seção 2.4). Apresentadas essas ideias, iremos discutir como as Equações de Navier-Stokes (seção 2.5.1) são formuladas e qual a modificação feita no modelo que é utilizado neste trabalho. Será discutido também o Teorema da Decomposição de Helmholtz Anisotrópica, o qual nos dá o limite teórico para que a equação de  $\mathbf{T}$ -Poisson 3.3 exista e seja única.

### 2.1 TENSORES

Tensores são objetos algébricos que descrevem uma relação multilinear entre conjuntos de objetos de um espaço vetorial [Itskov et al., 2007]. Tensores são utilizados em diversas áreas, como na física e engenharia para descrever grandezas como tensão e permeabilidade. Nesses casos, necessitamos de descrever a variação das quantidades em mais de uma direção. Entidades matemáticas mais simples, como escalares e vetores, não possuem capacidade de armazenamento de informação suficiente para que possamos utiliza-las nesses casos. Tensores conseguem descrever relações entre vetores, escalares, e outros tensores e também contêm um sistema de coordenadas independente.

Podemos tensores como extensões dos conceitos de vetores. De acordo com Kolecki [2002], podemos classifica-los de acordo com a sua ordem  $k$ . O número de componentes de um tensor depende do espaço euclidiano no qual ele está, assim, um tensor possui  $M^k$  componentes, onde  $M$  é a dimensão do espaço. Neste trabalho estamos interessados no espaço  $\mathbb{R}^3$ , logo, o número de componentes de um tensor é  $3^k$ . Um tensor de ordem 0, ou escalar, possui apenas uma componente e representa a magnitude de alguma grandeza. Um tensor de ordem 1, ou vetor, possui 3 componentes que representam uma direção e uma magnitude. Tensores de ordem 2, possuem 9 componentes que representam 3 direções e 3 magnitudes. Portanto, podemos observar os tensores como uma generalização para o conceito de vetores.

#### 2.1.1 TENSORES DE ORIENTAÇÃO

Um caso especial de tensores é o tensor de orientação [Westin, 1994], sendo uma matriz quadrada positiva definida. Um tensor de orientação  $\mathbf{T}$  pode ser representado matematicamente como:

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

onde  $\lambda_i$  é um autovalor de  $T$ ,  $\mathbf{e}_i$  é o autovetor associado à  $\lambda_i$  e  $\odot$  é o produto tensorial entre dois vetores. O tensor  $T$  resultante é uma combinação linear dos vetores  $e_i$  ponderados

pelos escalares  $\lambda_i$ . Cada um dos autovetores  $e_i$  define uma direção da base ortonormal armazenada em  $T$  escalada pelos autovalores  $\lambda_i$ .

Em  $\mathbb{R}^3$ , um tensor de orientação também pode ser decomposto em uma soma das contribuições de suas características lineares, planares e esféricas ( $\mathbf{T}_l$ ,  $\mathbf{T}_p$  e  $\mathbf{T}_s$ ), como:

$$\mathbf{T} = (\lambda_1 - \lambda_2)\mathbf{T}_l + (\lambda_2 - \lambda_3)\mathbf{T}_p + \lambda_3\mathbf{T}_s, \quad (2.1)$$

Esta decomposição tem um importante significado geométrico. Analisando a relação entre os três autovalores, nós podemos determinar aproximadamente o formato do tensor. Basicamente, temos três possibilidades:

1.  $\lambda_1 \gg \lambda_2 \approx \lambda_3$  indica aproximadamente um tensor linear,
2.  $\lambda_1 \approx \lambda_2 \gg \lambda_3$  indica aproximadamente um tensor planar e
3.  $\lambda_1 \approx \lambda_2 \approx \lambda_3$  indica aproximadamente um tensor isotrópico.

## 2.2 DESCRIÇÃO DE UM FLUXO

Imagine um fluxo de fluido onde cada partícula carrega suas próprias propriedades. A medida que a partícula avança, suas propriedades podem mudar com o tempo. O método de descrever todo o fluxo, registrando as histórias detalhadas de cada partícula de fluido, é a descrição **lagrangiana**, também denominada **descrição material**.

Por outro lado, se registrarmos a evolução das propriedades do fluxo em cada ponto do espaço à medida que o tempo varia, em vez de seguir cada partícula de fluido, estamos adotando a descrição **euleriana**, também denominada **descrição espacial**. Esse último método permite observar fenômenos em pontos de interesse, onde as localizações são descritas em sistemas de coordenadas.

## 2.3 DERIVADA MATERIAL

A derivada material é a derivada de tempo - taxa de variação - de uma propriedade seguindo uma partícula de fluido “p”. Denotada por:

$$\frac{D(\cdot)}{Dt}. \quad (2.2)$$

Essa derivada é um conceito lagrangiano que também pode ser expressado em termos de quantidades eulerianas.

Dado uma quantidade  $\theta$  associada a uma propriedade da partícula de fluido “p”, se usarmos a descrição lagrangiana para descrever essa taxa de variação,  $\theta$  é identificada

pelas coordenadas materiais  $X_1$ ,  $X_2$  e  $X_3$  e pelo tempo  $t$ , ou seja,  $\theta = \theta(X_1, X_2, X_3, t)$ . Dessa maneira, a derivada material pode ser escrita como:

$$\frac{D\theta}{Dt} = \frac{\partial\theta}{\partial t} \Big|_{X_i \text{ fixo}}, \quad (2.3)$$

Por outra perspectiva, ao usarmos a descrição euleriana, temos a derivada material dada por:

$$\frac{D\theta}{Dt} = \frac{\partial\theta}{\partial t} \Big|_{X_i \text{ fixo}} = \frac{\partial\theta}{\partial x_1} \frac{\partial x_1}{\partial t} + \frac{\partial\theta}{\partial x_2} \frac{\partial x_2}{\partial t} + \frac{\partial\theta}{\partial x_3} \frac{\partial x_3}{\partial t} + \frac{\partial\theta}{\partial t} \Big|_{x_i \text{ fixo}}, \quad (2.4)$$

onde a quantidade  $\theta$  é agora identificada pelas coordenadas espaciais  $x_1$ ,  $x_2$  e  $x_3$ , que indicam as posições em cada ponto do espaço considerado, e pelo tempo  $t$ , ou seja,  $\theta = \theta(x_1, x_2, x_3, t)$ . Nessa perspectiva, as coordenadas espaciais se relacionam com as coordenadas materiais pelo movimento  $x_i = x_i(X_1, X_2, X_3, t)$  e então  $\frac{\partial x_i}{\partial t}$  são obtidos com valores fixos dos  $X_i$ .

Mais ainda, se considerarmos o sistema retangular de coordenadas cartesianas, a derivada material pode ser expressa como:

$$\frac{D\theta}{Dt} = \frac{\partial\theta}{\partial t} + u \frac{\partial\theta}{\partial x_1} + v \frac{\partial\theta}{\partial x_2} + w \frac{\partial\theta}{\partial x_3} = \frac{\partial\theta}{\partial t} + \mathbf{v} \cdot \nabla\theta, \quad (2.5)$$

onde  $\left(\frac{\partial x_i}{\partial t}\right) \Big|_{X_i \text{ fixo}}$  fornece a  $i$ -ésima componente do vetor velocidade da partícula.

## 2.4 TENSOR TENSÃO DE CAUCHY

O tensor  $\boldsymbol{\sigma}$  que define completamente o estado de tensão em um ponto dentro de um material no estado deformado, é o tensor chamado **tensor tensão de Cauchy** ou tensor tensão verdadeiro. Esse tensor é medido quando um objeto está na configuração deformada e obtido a partir do vetor de tensões.

Considere, como exemplo, um corpo que está na configuração deformada e que é seccionado por um plano passando pelo ponto  $\mathbf{P}(\mathbf{x}, t)$ . Esse plano é definido pela sua normal  $\mathbf{n}$ .

Considere ainda uma área deformada  $\Delta a$  no entorno desse ponto  $\mathbf{P}(\mathbf{x}, t)$  cuja força interna seja  $\Delta \mathbf{f}$ .

Definimos no ponto  $\mathbf{P}(\mathbf{x}, t)$  o vetor de tensões que está associado a normal  $\mathbf{n}$ , como:

$$\mathbf{T}^{\mathbf{n}}(\mathbf{x}, t, \mathbf{n}) = \lim_{\Delta a \rightarrow 0} \left( \frac{\Delta \mathbf{f}}{\Delta a} \right), \quad (2.6)$$

cuja unidade de medida é N/m<sup>2</sup>, ou seja, Pa (Pascal). Esse vetor de tensões representa a força por área deformada.

O vetor de tensões  $\mathbf{T}^{\mathbf{n}}$  é função da posição  $\mathbf{x}$  e da normal  $\mathbf{n}$ , de qualquer plano arbitrário que passe por  $\mathbf{P}(\mathbf{x}, t)$ , pelo postulado fundamental de Cauchy [Chadwick, 2012].

Mais ainda, Cauchy provou que se definirmos o vetor de tensões em função de três normais de três planos perpendiculares entre si, que passam por  $\mathbf{P}(\mathbf{x}, t)$ , podemos descrever o estado de tensão do ponto. Considerando os vetores unitários da base  $e_1$ ,  $e_2$  e  $e_3$ , se adotarmos esses planos perpendiculares a eles teremos três vetores de tensão associados a essas direções para o ponto  $\mathbf{P}(\mathbf{x}, t)$ .

Decompondo cada vetor de tensão nas direções dos vetores  $e_1$ ,  $e_2$  e  $e_3$ , temos:

$$\begin{aligned}\mathbf{T}^{(e_1)} &= T_1^{(e_1)}e_1 + T_2^{(e_1)}e_2 + T_3^{(e_1)}e_3, \\ \mathbf{T}^{(e_2)} &= T_1^{(e_2)}e_1 + T_2^{(e_2)}e_2 + T_3^{(e_2)}e_3, \\ \mathbf{T}^{(e_3)} &= T_1^{(e_3)}e_1 + T_2^{(e_3)}e_2 + T_3^{(e_3)}e_3.\end{aligned}$$

O tensor tensão de Cauchy é obtido compondo um tensor de tensões para considerar os três vetores simultaneamente. Dessa forma, temos esse tensor tensão verdadeiro dado por:

$$\boldsymbol{\sigma} = \mathbf{T}^{e_1}e_1 + \mathbf{T}^{e_2}e_2 + \mathbf{T}^{e_3}e_3 = \sigma_{ij}e_i e_j, \quad (2.7)$$

onde  $\sigma_{ij}$  são as componentes do tensor tensão de Cauchy. Mais explicitamente, na forma matricial temos:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}. \quad (2.8)$$

Essa matriz representa nas direções da base do sistema de coordenadas cartesiano as componentes do vetor de tensão.

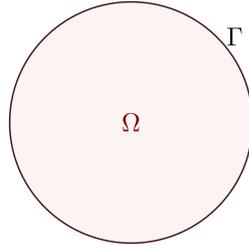
O tensor tensão de Cauchy relaciona um vetor de direção  $\mathbf{n}$  de comprimento unitário, ao vetor de tração  $T_{\mathbf{n}}$  através de uma superfície imaginária perpendicular a  $\mathbf{n}$  [Irgens, 2008]:

$$\begin{aligned}T_{\mathbf{n}} = \mathbf{n} \cdot \boldsymbol{\sigma} &\Leftrightarrow \mathbf{T}_{\mathbf{j}}^{\mathbf{n}} = \sigma_{ij}n_i \\ &\Leftrightarrow \begin{bmatrix} \mathbf{T}_1^{\mathbf{n}} & \mathbf{T}_2^{\mathbf{n}} & \mathbf{T}_3^{\mathbf{n}} \end{bmatrix} = \begin{bmatrix} n_1 & n_2 & n_3 \end{bmatrix} \cdot \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}\end{aligned}$$

## 2.5 CONSERVAÇÃO DO MOMENTO

Para realizar a dedução da conservação de momento iremos utilizar a lei de conservação de massa descrita em Batchelor [2000] como:

$$\int_{\Omega} \rho dV + \int_{\Gamma} \rho \mathbf{v} dA = f, \quad (2.9)$$



– Região do volume de controle  $\Omega$  e com bordo  $\Gamma$ .

onde  $\Omega$  é o volume de controle onde será feito o balanço de massa e  $\Gamma$  é o bordo de  $\Omega$  como visto na figura 2.5.

Dado uma quantidade macroscópica vetorial  $\Phi$  associada a um volume de controle  $\Omega$ . Podemos definir a grandeza infinitesimal  $\phi = \frac{\Phi}{m}$ , onde  $m$  é a massa de  $\Omega$ , associada a um corpo microscópico de matéria pertencente a  $\Omega$ . Tal corpo microscópico contém uma porção da propriedade  $\Phi$ , no qual é transportado dentro e através das paredes de  $\Gamma$ , que é a superfície fechada que engloba  $\Omega$ . A equação que representa esse transporte é dada por:

$$\frac{\partial \Phi}{\partial t} = \frac{\partial \Phi_V}{\partial t} + \Phi_f, \quad (2.10)$$

onde  $\Phi_V$  representa o quanto de  $\Phi$  que está dentro de  $\Omega$ ,  $\Phi_f$  representa o fluxo de  $\Phi$  sob  $\Omega$  e  $f$  representa uma fonte de  $\Phi$ , adicionando ou removendo a quantidade de  $\Omega$ . Se considerarmos  $\Phi$  representando a concentração de algum soluto, o termo de fonte pode representar por exemplo, a variação da concentração devido a alguma reação química envolvida. Os termos  $\Phi_V$  e  $\Phi_f$  representam o conceito clássico de conservação, onde não há geração de massa e nem da quantidade  $\Phi$ .

O termo  $\Phi_V$  pode ser escrito como:

$$\Phi_V = \int_{\Omega} \phi \rho dV, \quad (2.11)$$

onde  $\rho$  é a densidade de massa,  $dV$  representa do volume infinitesimal de integração e  $\rho dV$  representa a massa daquele corpo infinitesimal.

O termo  $\Phi_f$  pode ser escrito como:

$$\Phi_f = \int_{\Gamma} \phi \otimes \rho \mathbf{v} \cdot \mathbf{n} dA, \quad (2.12)$$

onde  $\mathbf{v}$  é a velocidade do infinitésimo de massa que contém  $\Phi$ ,  $\mathbf{n}$  é o vetor normal a  $\Gamma$  e  $\otimes$  é o produto tensorial entre dois vetores.

A equação 2.10 pode ser usada para representar a conservação de diferentes propriedades envolvidas a matéria, tal como energia, usando  $\Phi$  como escalar e momento utilizando  $\Phi$  como uma grandeza vetorial. Este trabalho é focado na conservação de momento para simulação de fluidos incompressíveis. Portanto, a quantidade  $\phi$  irá representar o momento

do corpo infinitesimal. Fazendo  $\Phi = \mathbf{p} = m\mathbf{v}$ , podemos substituir  $\phi = \frac{m\mathbf{v}}{m}$  na equação 2.10, então obtemos:

$$\frac{\partial \mathbf{p}}{\partial t} = \frac{\partial}{\partial t} \int_{\Omega} \mathbf{v} \rho dV + \int_{\Gamma} \mathbf{v} \otimes \rho \mathbf{v} \cdot \mathbf{n} dA. \quad (2.13)$$

Para prosseguirmos, iremos assumir a visão euleriana, onde  $\Omega$  é fixo no espaço, com isso, podemos colocar o termo diferencial  $\frac{\partial}{\partial t}$  dentro da integral no primeiro termo. Utilizando o teorema de Gauss, podemos reescrever a integral de superfície da equação 2.13 como uma integral de volume:

$$\frac{\partial \mathbf{p}}{\partial t} = \int_{\Omega} \frac{\partial(\rho \mathbf{v})}{\partial t} dV + \int_{\Omega} \nabla \cdot (\mathbf{v} \otimes \rho \mathbf{v}) dV. \quad (2.14)$$

Segundo Doran et al. [2003], o divergente de um produto tensorial entre dois vetores  $\mathbf{b}$  e  $\mathbf{c}$  quaisquer pode ser escrito como:

$$\nabla \cdot (\mathbf{b} \otimes \mathbf{c}) = \mathbf{b} \nabla \cdot \mathbf{c} + \nabla \mathbf{b} \mathbf{c} \quad (2.15)$$

Na primeira integral utilizamos a regra do produto para a derivada em relação ao tempo e conseguimos expandir o termo  $\frac{\partial(\rho \mathbf{v})}{\partial t}$ , além disso, utilizamos a identidade expressa na equação 2.15 para reescrever a equação 2.14 como:

$$\frac{\partial \mathbf{p}}{\partial t} = \int_{\Omega} \left( \rho \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \frac{\partial \rho}{\partial t} + \mathbf{v} \nabla \cdot (\rho \mathbf{v}) + \nabla \mathbf{v} (\rho \mathbf{v}) \right) dV. \quad (2.16)$$

Utilizando a lei de conservação de massa:

$$\int_{\Omega} \frac{\partial \rho}{\partial t} dV + \int_{\Omega} \rho \nabla \cdot \mathbf{v} dV = 0,$$

obtemos a equação 2.16 simplificada na forma:

$$\frac{\partial \mathbf{p}}{\partial t} = \int_{\Omega} \rho \left( \frac{\partial \mathbf{v}}{\partial t} + \nabla \mathbf{v} \mathbf{v} \right) dV. \quad (2.17)$$

Utilizando a derivada material da velocidade vista na seção 2.2, podemos simplificar a equação 2.17:

$$\frac{\partial \mathbf{p}}{\partial t} = \int_{\Omega} \rho \mathbf{a} dV. \quad (2.18)$$

No fim, para  $\mathbf{a}$  uniforme em  $\Omega$ , obtemos a famosa igualdade  $\mathbf{F} = m \mathbf{a}$ , que é uma consequência direta da lei de conservação do momento.

### 2.5.1 EQUAÇÕES DE NAVIER-STOKES

Partindo do resultado obtido na Seção anterior, podemos separar as forças em dois tipos, forças de campo e forças de superfície ( $\mathbf{F}_{\Gamma}$ ). Como força de campo, iremos considerar apenas a gravidade. Reescrevendo a equação 2.18, obtemos:

$$\mathbf{F}_{\Gamma} + \int_{\Omega} \rho \mathbf{g} dV = \int_{\Omega} \rho \mathbf{a} dV, \quad (2.19)$$

onde  $\mathbf{g}$  é a aceleração da gravidade na superfície terrestre.

A força de superfície ela é dada por:

$$\mathbf{F}_\Gamma = \int_\Gamma \mathbf{t} \, dA, \quad (2.20)$$

onde  $\mathbf{t}$  é o vetor tração, que representa a força de contato entre os corpos distribuída pela área de contato. Pelo teorema de Cauchy, podemos reescrever a equação 2.20 como:

$$\mathbf{F}_\Gamma = \int_\Gamma \boldsymbol{\sigma} \mathbf{n} \, dA. \quad (2.21)$$

Aplicando o teorema de Gauss na equação 2.21, obtemos:

$$\mathbf{F}_\Gamma = \int_\Omega \nabla \cdot \boldsymbol{\sigma} \, dV. \quad (2.22)$$

Para  $\Omega$  infinitesimal, podemos escrever a equação 2.19 como:

$$\nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g} = \rho \mathbf{a}. \quad (2.23)$$

Pela conservação de momento angular, sabemos que  $\boldsymbol{\sigma}$  é simétrico, portanto, precisamos determinar apenas 6 componentes que formam o tensor de Cauchy. Para tal, iremos definir o tensor de Cauchy a partir de equações constitutivas para fluidos incompressíveis.

Para um volume de controle, existem dois tipos de tensões sendo aplicadas à sua superfície, tensões normais e tensões de cisalhamento. A tensão normal que é aplicada a uma região é chamada de pressão. Segundo Batchelor [2000], a pressão e a tensão de cisalhamento podem ser expressas como:

$$\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\mathcal{T}}, \quad (2.24)$$

onde  $-p\mathbf{I}$  é a parte isotrópica do tensor de stress e  $p$  é a pressão. A parte cisalhante é definida a partir do tensor desviador, calculado utilizando o gradiente do campo de velocidades como:

$$\mathbf{D} = \frac{1}{2}(\nabla \mathbf{v} + \nabla \mathbf{v}^t). \quad (2.25)$$

Para fluidos newtonianos, a relação entre  $\boldsymbol{\mathcal{T}}$  e  $\mathbf{D}$  é linear, dada por:

$$\boldsymbol{\mathcal{T}} = 2\mu\mathbf{D}, \quad (2.26)$$

sendo  $\mu$  o coeficiente de viscosidade dinâmica constante. Substituindo o tensor de Cauchy na equação 2.23, temos:

$$\nabla \cdot (-p\mathbf{I} + \mu(\nabla \mathbf{v} + \nabla \mathbf{v}^t)) + \rho \mathbf{g} = \rho \mathbf{a}. \quad (2.27)$$

Segundo Doran et al. [2003], existe a seguinte identidade:

$$\nabla \cdot (\alpha \mathbf{T}) = \alpha \nabla \cdot \mathbf{T} + \mathbf{T} \nabla \alpha, \quad (2.28)$$

onde  $\mathbf{T}$  é um campo tensorial e  $\alpha$  é um campo escalar qualquer. Aplicando essa identidade à equação ao termo  $\nabla \cdot (-p\mathbf{I})$ , obtemos:

$$\nabla \cdot (-p\mathbf{I}) = -p \nabla \cdot \mathbf{I} - \mathbf{I} \nabla p, \quad (2.29)$$

como o campo de tensores identidade não varia espacialmente, podemos escrever a equação 2.27 como:

$$\mu \nabla \cdot (\nabla \mathbf{v} + \nabla \mathbf{v}^t) - \nabla p + \rho \mathbf{g} = \rho \mathbf{a}. \quad (2.30)$$

Assumindo a seguinte identidade encontrada em Doran et al. [2003]:

$$\nabla \cdot (\nabla \mathbf{b}^t) = \nabla (\nabla \cdot \mathbf{b}), \quad (2.31)$$

podemos reescrever a equação 2.30 como:

$$\mu \nabla \cdot \nabla \mathbf{v} + \mu \nabla (\nabla \cdot \mathbf{v}) - \nabla p + \rho \mathbf{g} = \rho \mathbf{a}. \quad (2.32)$$

Entretanto, pela lei de conservação de massa para fluidos incompressíveis, sabemos que  $\nabla \cdot \mathbf{v} = 0$ , portanto, o termo  $\mu \nabla (\nabla \cdot \mathbf{v}) = 0$ . Com isso, temos a equação simplificada:

$$\mu \nabla \cdot \nabla \mathbf{v} - \nabla p + \rho \mathbf{g} = \rho \mathbf{a}. \quad (2.33)$$

Utilizando a derivada material, conseguimos reescrever o termo de aceleração  $\mathbf{a}$  como:

$$\mu \nabla \cdot \nabla \mathbf{v} - \nabla p + \rho \mathbf{g} = \rho \frac{\partial \mathbf{v}}{\partial t} + \rho \mathbf{v} \cdot \nabla \mathbf{v}. \quad (2.34)$$

A equação 2.34 é conhecida como equação de Navier-Stokes para fluidos incompressíveis. Se dividirmos a equação 2.34 por  $\rho$  e adicionarmos a equação de conservação de massa, ficamos com a forma que geralmente é encontrada em livros-texto:

$$\begin{cases} \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \nabla \mathbf{v} + \frac{1}{\rho} \nabla p = \mathbf{g} + \nu \nabla \cdot \nabla \mathbf{v}, \\ \nabla \cdot \mathbf{v} = 0, \end{cases} \quad (2.35)$$

onde  $\nu$  é a viscosidade cinemática que é definida como  $\nu = \frac{\mu}{\rho}$ . Assumimos duas condições de contorno para o problema a de não deslização e a de não penetração. A condição de não deslização nos diz que quando o fluido está em contato direto com o sólido, no ponto de contato há a aderência do fluido ao sólido, impedindo o deslização. A condição de não penetração assume que o fluido não irá adentrar em um meio sólido.

## 2.6 EQUAÇÕES DE NAVIER-STOKES PARA MEIOS ANISOTRÓPICOS

A dinâmica de fluidos é geralmente expressa pelas equações de Navier-Stokes, que envolve a velocidade do fluido  $\mathbf{u}$ , pressão  $p$ , densidade do fluido  $\rho$  e forças externas  $\mathbf{f}$ , além de parâmetros como a viscosidade  $\nu$ . Em Renhe et al. [2019], o modelo clássico foi adaptado para incluir um campo tensorial. A ideia é que cada tensor haja como parte do meio, modificando a passagem do fluxo de fluido através de cada Célula. O tensor  $\mathbf{T}$  é um díado positivo definido em  $\mathbb{R}^3$ , onde  $\lambda_1$ ,  $\lambda_2$  e  $\lambda_3$  são seus autovalores, com os autovetores correspondentes  $\mathbf{e}_1$ ,  $\mathbf{e}_2$  e  $\mathbf{e}_3$ . As equações de Navier-Stokes modificadas, incluindo o elemento de campo tensorial, são descritas da seguinte forma:

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla) \mathbf{u} - \frac{1}{\rho} \nabla_{\mathbf{T}} p + \nu \nabla \cdot \nabla_{\mathbf{T}} \mathbf{u} + \mathbf{f}, \\ \nabla \cdot \mathbf{u} = 0, \end{cases} \quad (2.36)$$

onde  $\nabla_{\mathbf{T}}$  é o resultado do produto entre  $\mathbf{T}$  e o operador gradiente [Dassios and Lindell, 2002]:

$$\nabla_{\mathbf{T}} = \mathbf{T} \cdot \nabla = \begin{bmatrix} \lambda_1 \frac{\nabla}{\nabla_{\mathbf{e}_1}} \\ \lambda_2 \frac{\nabla}{\nabla_{\mathbf{e}_2}} \\ \lambda_3 \frac{\nabla}{\nabla_{\mathbf{e}_3}} \end{bmatrix}, \quad (2.37)$$

também conhecido como  $\mathbf{T}$ -Gradiente. Além da típica restrição do campo ser livre de divergente na Equação 2.36, uma adicional é incluída da seguinte forma:

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{T} \mathbf{u} - \beta \mathbf{u}, \quad (2.38)$$

para algum fator de escala  $\beta$ . Em Renhe et al. [2019], esta restrição é incorporada no passo da advecção, porém, neste trabalho, ao invés disso é adicionada como uma força de aceleração. Enfim, a Equação resultante é:

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla) \mathbf{u} - \frac{1}{\rho} \nabla_{\mathbf{T}} p + \nu \nabla \cdot \nabla_{\mathbf{T}} \mathbf{u} + \mathbf{T} \mathbf{u} - \beta \mathbf{u} + \mathbf{f}, \\ \nabla \cdot \mathbf{u} = 0. \end{cases} \quad (2.39)$$

Se o tensor  $\mathbf{T}$  é a identidade ( $\mathbf{T} = \mathbf{I}$ ), a equação 2.36 se reduz ao modelo clássico de Navier-Stokes.

A simulação de fluidos apresentado em Renhe et al. [2019] segue a mesma ideia de advecção-difusão-projeção encontrada em Stam [1999]. O fluido é discretizado em uma grade, e a simulação é composta de quatro passos. Primeiro, forças externas são aplicadas. Em seguida, é executada uma advecção semi-Lagrangiana, com a partícula de retrocesso sendo direcionada pelo tensor através da Equação 2.38. Por fim, a parte de difusão e projeção recorrem a um esquema de discretização complexa para computar o operador  $\mathbf{T}$ -Laplaciano  $\nabla \cdot \nabla_{\mathbf{T}}$  usado para adicionar anisotropia nesses casos.

## 2.7 SOLUÇÃO NUMÉRICA DAS EQUAÇÕES DE NAVIER STOKES

As equações de Navier-Stokes descrevem o movimento de fluidos viscosos e são fundamentais em diversos campos da ciência e engenharia, desde a aerodinâmica até a engenharia biomédica. Resolver numericamente essas equações é essencial para compreender e prever o comportamento de fluidos em diferentes cenários. Entre os diversos métodos numéricos disponíveis para essa tarefa, o método PIC (Particle-in-Cell) destaca-se por sua capacidade de lidar com fluidos de alta complexidade, como os turbulentos, de forma eficiente.

Proposto por Evans et al. [1957], o método PIC combina elementos de duas abordagens principais: a representação de partículas para descrever o movimento do fluido e uma malha Euleriana para resolver as equações de transporte de maneira eficiente. Nesse método, as partículas representam os elementos do fluido e movem-se de acordo com as equações de movimento, enquanto as propriedades do fluido são calculadas em pontos fixos da malha Euleriana.

O método FLIP (Fluid Implicit Particle), introduzido por Brackbill et al. [1988], é uma variante do método PIC que visa melhorar a precisão da simulação reduzindo a viscosidade numérica adicionada pela interpolação da velocidade feita pelo método PIC. O FLIP combina características do Método PIC com um esquema de interpolação semi-Lagrangiana, permitindo uma melhor captura de fenômenos de transporte e de superfícies de separação de fluidos.

Ambos os métodos, PIC e FLIP, são capazes de lidar com a complexidade dos fluxos turbulentos de maneira eficaz. Ao usar partículas para seguir o movimento do fluido, esses métodos conseguem capturar detalhes finos da turbulência e fenômenos de transporte complexos, como mistura de fluidos e interface de fase. Além disso, ambos os métodos são altamente paralelizáveis, facilitando a incorporação em sistemas de visualização em tempo real, como em *engines* de jogos.

O método FLIP, que é a base desta dissertação, utiliza-se o processo de *splitting* para resolução e EDP's, que consiste em separar os diferentes fenômenos representados pelo modelo matemático e solucioná-los separadamente. Realizando o processo de *splitting* nas equações de Navier-Stokes para fluidos incompressíveis (equação 2.34) obtemos as equações 2.40, 2.41, 2.42 e 2.43. A equação 2.40 representa a variação do momento do fluido devido a forças externas. A equação 2.41 descreve aceleração provocada pelas forças de cisalhamento do fluido. A equação 2.42 descreve a aceleração devida a média das forças normais, é representada pela pressão, onde também é incluso a conservação de massa pelo divergente da velocidade. A equação 2.43 descreve a integração da aceleração ao longo do percurso de uma partícula fluida.

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{g}, \quad (2.40)$$

$$\frac{\partial \mathbf{u}}{\partial t} = \nu \nabla \cdot \nabla \mathbf{u}, \quad (2.41)$$

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\rho} \nabla p, \nabla \mathbf{u} = 0, \quad (2.42)$$

$$\frac{D\mathbf{u}}{Dt} = 0. \quad (2.43)$$

O método FLIP utiliza de diversos passos para solucionar cada uma das equações que aparecem ao fazer o *splitting* da equação de Navier-Stokes. Seja  $P$  o conjunto de todas as partículas no domínio da simulação e  $G$  a grade MAC que armazena as velocidades interpoladas de  $P$ . Uma descrição rápida do algoritmo iterativo é:

1. Semeie um conjunto  $P$  de partículas de fluido;
2. Transferir as velocidades de  $P$  para as faces da grade  $G$ ;
3. Classifique todas as células como fluido, solido ou vazio;
4. FLIP: Salve uma cópia das velocidades nas faces;
5. Adicione as forças externas nas faces de  $G$ ;
6. Monte o sistema linear para a viscosidade e resolva;
7. Monte o sistema linear utilizando as condições de borda para a equação de Poisson;
8. Resolva o sistema linear e compute a nova grade com divergente nulo;
9. Compute as novas velocidades, armazenando o resultado nas faces de  $G$ ;
10. FLIP: Interpole e adicione a diferença entre as novas velocidades e as velocidades salvas a partir das faces de  $G$  para as velocidades das partículas em  $P$ ;
11. Aplique a advecção em  $P$ .

A Etapa 1 insere partículas no domínio quando necessário, operando como um termo de fonte de fluido, dependendo especificamente da simulação. As Etapas 2 e 10 são executadas interpolando as velocidades de  $P$  para  $G$ , considerando suas distâncias para cada centro da face da célula. Para uma face alinhada em no eixo  $x$ , a interpolação da velocidade é dada pela equação 2.44.

$$u_{i+1/2,j,k} = \frac{\sum_{a \in N} \theta(\mathbf{x}_a - \mathbf{x}_{i+1/2,j,k}) u_a^x}{\sum_{a \in N} h(\mathbf{x}_a - \mathbf{x}_{i+1/2,j,k})}, \quad (2.44)$$

onde  $N \subset P$  é um conjunto de partículas dentro das células vizinhas,  $\theta$  é uma função de interpolação,  $\mathbf{x}_a$  é a posição da partícula  $a$ ,  $u_a^x$  é o componente  $x$  da velocidade da partícula, e  $\mathbf{x}_{i+1/2,j,k}$  é a posição do centro da face. As outras cinco faces são interpoladas de forma análoga. A função de interpolação  $\theta$  utilizada neste trabalho é descrita pela equação 2.45 que também se encontra em Bridson [2015].

$$\theta(x, y, z) = K\left(\frac{x}{h}\right) K\left(\frac{y}{h}\right) K\left(\frac{z}{h}\right),$$

$$K(r) = \begin{cases} 1 - r & : 0 \leq r \leq 1, \\ 1 + r & : -1 \leq r \leq 0, \\ 0 & : \text{caso contrário.} \end{cases} \quad (2.45)$$

A Etapa 3 é feita analisando a distribuição de partículas dentro do domínio. Células que contêm pelo menos uma partícula são classificadas como células que contêm fluido. Células sem fluido são consideradas células vazias. Células que pertencem a borda do domínio são consideradas como células com propriedades de sólido. Dependendo da configuração da cena, células dentro do domínio também podem ter células sólidas, portanto não poderão ter partículas nelas.

A Etapa 4 é exclusiva do método FLIP, no qual armazena uma grade reserva com as informações de velocidade antes de serem alteradas pelo método.

A Etapa 5 resolve a equação 2.40. Uma forma simples de se resolver numericamente é utilizando o método de Euler explícito, com isso obtemos a equação 2.46.

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{g},$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = \mathbf{g}^n, \quad (2.46)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \mathbf{g}^n,$$

onde  $\mathbf{u}^{n+1}$  é a nova velocidade a ser calculada para a próxima iteração,  $\mathbf{u}^n$  é a velocidade atual e  $\mathbf{g}^n$  é o campo gravitacional da iteração atual.

A etapa 6 consiste em resolver a equação 2.41. Para isso podemos utilizar a versão apresentada por Stam [1999], onde é aplicado o método de Euler implícito, com isso obtemos a equação 2.47.

$$\frac{\partial \mathbf{u}}{\partial t} = \nu \nabla \cdot \nabla \mathbf{u},$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = \nu \nabla \cdot \nabla \mathbf{u}_{n+1}, \quad (2.47)$$

$$(\mathbf{I} - \nu \Delta t \nabla \cdot \nabla) \mathbf{u}^{n+1} = \mathbf{u}^n,$$

onde  $\mathbf{I}$  é o tensor identidade. Note que a velocidade  $\mathbf{u}^n$  utilizada nesta equação é equivalente ao campo  $\mathbf{u}^{n+1}$  que foi obtido pela equação 2.46, ou seja, o campo de velocidades obtido

na resolução de uma das equações é utilizado diretamente para a resolução da próxima. A discretização dos operadores divergente e gradiente irá depender especificamente do método escolhido para isso, podendo ser obtido utilizando diferenças finitas, ou algum mais robusto baseado em volumes finitos.

A Etapa 7 resolve a equação 2.42. Para tal, é utilizado o método de Euler explícito para discretizar a derivada temporal da velocidade e posteriormente é acoplado a equação de conservação de massa. Com isso, obtemos a equação 2.48.

$$\begin{aligned}\frac{\partial \mathbf{u}}{\partial t} &= -\frac{1}{\rho} \nabla p, \\ \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} &= -\frac{1}{\rho} \nabla p^n, \\ \nabla \cdot (\mathbf{v}^{n+1} - \mathbf{u}^n) &= -\nabla \cdot \left( \frac{\Delta t}{\rho} \nabla p^n \right), \\ \nabla \cdot \mathbf{u}^n &= \frac{\Delta t}{\rho} \nabla \cdot \nabla p^n,\end{aligned}\tag{2.48}$$

onde  $p^n$  é a pressão atual que será calculada pela equação a partir de algum método para discretização espacial da EDP. Note que, pela equação de conservação de massa,  $\nabla \cdot \mathbf{u}^{n+1} = 0$ . Note também que utilizando a equação 2.42 podemos encontrar  $\mathbf{u}^{n+1}$  utilizando a própria equação 2.48 para solucionar a Etapa 8 do algoritmo FLIP, como descrito na equação 2.49.

$$\begin{aligned}\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} &= -\frac{1}{\rho} \nabla p^n, \\ \mathbf{u}^{n+1} &= \mathbf{u}^n - \frac{\Delta t}{\rho} \nabla p^n.\end{aligned}\tag{2.49}$$

A combinação entre a solução encontrada pelo PIC e FLIP é interpolada e adicionada na Etapa 10 da seguinte forma:

$$\mathbf{u}_a = (1 - \alpha) \left( \hat{\mathbf{u}}_a^{novo} - \hat{\mathbf{u}}_a^{salvo} \right) + \alpha \hat{\mathbf{u}}_a^{novo},\tag{2.50}$$

onde  $\mathbf{u}_a$  é a velocidade atualizada da partícula  $a$ ,  $\hat{\mathbf{u}}_a^{salvo}$  é a velocidade anterior da grade (Etapa 4) interpolada na partícula  $a$ ,  $\hat{\mathbf{u}}_a^{novo}$  é a nova velocidade (Etapa 8) interpolada para a partícula  $a$ , e  $\alpha \in [0, 1]$  representa uma interpolação linear. O método é puramente FLIP se  $\alpha = 0$ , e puramente PIC se  $\alpha = 1$ . O método FLIP produz uma menor dissipação numérica já que as interpolações na grade não são acumuladas para as partículas. Geralmente  $\alpha = 0.05$ , significa que 5% da velocidade atualizada vem do PIC. A simulação resultante apresenta um movimento mais detalhado enquanto mantém algum nível de estabilidade.

### 3 MÉTODO BASEADO EM FACES

Neste capítulo é apresentado um método publicado em Parreiras et al. [2021] que é uma adaptação da abordagem presente em Renhe et al. [2019] para operar em conjunto com o método FLIP em uma grade escalonada MAC  $G$ , de dimensões  $N \times M \times W$ , representando o espaço de fluxo do fluido. A grade contém as velocidades posicionadas nas faces das células, enquanto a densidade do fluido, tensor  $\mathbf{T}$  da célula e todos os valores escalares usados para resolver as equações estão localizadas no centro da célula. Os voxels são referenciados como célula. Como em todos os métodos PIC, cada célula contém um conjunto de partículas. O meu papel no desenvolvimento de Parreiras et al. [2021] foi de elaborar as discretizações face-célula para os operadores  $\mathbf{T}$ -Laplaciano e  $\mathbf{T}$ -Gradiente, implementar o método e gerar os resultados.

O nosso objetivo é prover um método estável para simulação anisotrópica de fluidos governados por um campo tensorial  $\mathbf{T}(\mathbf{r})$ , onde  $\mathbf{r} \in \mathbb{R}^3$  é uma posição no espaço de fluxo. Como os métodos PIC tendem a ser dissipativos e o campo tensorial  $\mathbf{T}(\mathbf{r})$  não varia no tempo em nossa formulação, precisamos resolver a seguinte formulação contínua anisotrópica:

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla) \mathbf{u} - \frac{1}{\rho} \nabla_{\mathbf{T}} p + \mathbf{f}, \quad (3.1)$$

restrito a:

$$\begin{cases} \nabla \cdot \mathbf{u} = 0, \\ \frac{\partial \mathbf{T}}{\partial t} = \mathbf{0}, \end{cases} \quad (3.2)$$

onde todos os termos são da Equação 2.36, exceto a viscosidade que é zero. O termo difusivo é removido devido à elevada difusão numérica de métodos PIC tendem a apresentar. Como será utilizado a abordagem FLIP, o parâmetro  $\alpha$  pode ser utilizado para definir a influência método PIC na solução, entregando uma dissipação maior ou menor (Seção 3.4). Entretanto, se necessário, a difusão tensorial proposta em Renhe et al. [2019] pode ser facilmente incorporada. Em adição, nós usamos a deflexão dada pelo lado direito da Equação 2.38 com  $\beta = 1$ . Como mostrado na Seção 3.3, o lado direito da Equação 2.38 é aplicado como uma força externa no passo da advecção, incorporada diretamente pelo integrador Runge-Kutta.

A dinâmica abstrata representada pela Equação 3.1 é plausível de se interpretar a partir da perspectiva de simulação de fluidos em um meio poroso. Na prática, desenvolver um método numérico estável para a solução de um transporte anisotrópico em uma grade MAC é uma tarefa desafiadora. O texto será organizado da seguinte forma: Seção 3.1 mostra como os operadores  $\mathbf{T}$ -Gradiente e  $\mathbf{T}$ -Laplaciano face-célula e célula-célula são implementados para a resolução da projeção anisotrópica, Seção 3.3

discute como a Equação 2.38 é acoplada à advecção, e a Seção 3.4 mostra todos os passos para a simulação.

### 3.1 DISCRETIZAÇÃO DA PROJEÇÃO ANISOTRÓPICA

A projeção anisotrópica é computada através da solução da Equação de  $\mathbf{T}$ -Poisson Dassios and Lindell [2002], Vieira et al. [2021]:

$$\nabla \cdot \nabla_{\mathbf{T}} p(\mathbf{r}) = \nabla \cdot \mathbf{u}(\mathbf{r}), \quad (3.3)$$

onde  $\mathbf{r} \in G$  é qualquer ponto dentro da grade  $G$ ,  $\mathbf{u}(\mathbf{r})$  é o campo de velocidade e  $p(\mathbf{r})$  é a pseudo-pressão desconhecida. O objetivo é garantir a restrição  $\nabla \cdot \mathbf{u} = 0$  encontrando o campo vetorial projetado:

$$\mathbf{P}_{\mathbf{T}}(\mathbf{u}(\mathbf{r})) = \mathbf{u}(\mathbf{r}) - \nabla_{\mathbf{T}} p(\mathbf{r}), \quad (3.4)$$

que é livre de divergente em relação ao sistema de coordenadas padrão e sujeito à anisotropia do campo tensorial  $\mathbf{T}(\mathbf{r})$  Vieira et al. [2021].

Para ser possível resolver a Equação de  $\mathbf{T}$ -Poisson de forma discreta, dependemos de operadores adequados para o  $\mathbf{T}$ -Gradiente e o  $\mathbf{T}$ -Laplaciano, bem como condições de borda adequadas. Os métodos descritos neste texto procuram garantir a estabilidade da decomposição anisotrópica de Helmholtz e, conseqüentemente, garantir a estabilidade da evolução do fluxo de fluido.

Em uma grade MAC, é necessário apenas duas células incidentes para computar apenas um dos gradientes na face. Como é necessário aplicar o tensor em um gradiente estimado em um espaço 3D, mais do que duas células irão contribuir para a computação do gradiente em cada face. Portanto, encontrar o melhor arranjo para os cálculos se torna bastante difícil. Nesta capítulo iremos propor um novo esquema de discretização para a solução do problema de  $\mathbf{T}$ -Poisson (Equação 3.3) baseado em faces de uma grade MAC.

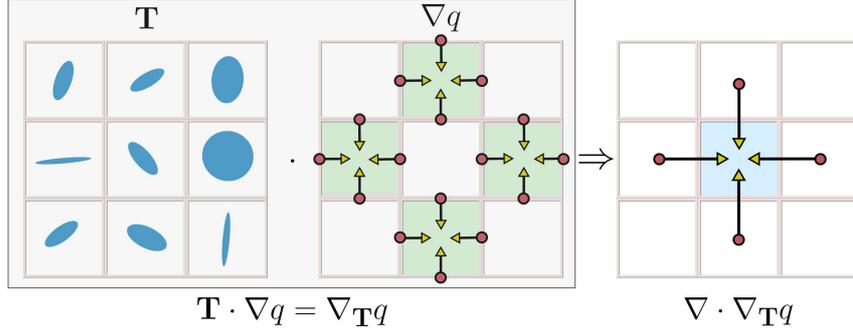
#### 3.1.1 Operadores $\mathbf{T}$ -Gradiente e $\mathbf{T}$ -Laplaciano

A computação de uma quantidade  $q$  nas faces é natural em uma grade MAC. O vetor resultante é posicionado ao centro da célula. Portanto, o  $\mathbf{T}$ -Gradiente na célula  $(i, j, k)$ ,  $i, j, k \in \mathbb{Z}$ , de uma quantidade  $q$  é:

$$\nabla_{\mathbf{T}} q_{i,j,k} = \frac{\mathbf{T}_{i,j,k}}{h} \begin{bmatrix} q_{i+1/2,j,k} - q_{i-1/2,j,k} \\ q_{i,j+1/2,k} - q_{i,j-1/2,k} \\ q_{i,j,k+1/2} - q_{i,j,k-1/2} \end{bmatrix}, \quad (3.5)$$

onde o tensor  $\mathbf{T}_{i,j,k}$  é também posicionado ao centro da célula (Figura 2),  $h$  é a distância entre duas células adjacentes.

- Figura 3 - Ilustração mostrando como os operadores  $\mathbf{T}$ -Gradiente (caixa cinza na esquerda) e o  $\mathbf{T}$ -Laplaciano (direita) são calculados para 2D.



Fonte: Extraído de Parreiras et al. [2021].

A partir da Equação 2.37, o operador  $\mathbf{T}$ -Laplaciano no contínuo é definido como  $\nabla \cdot \nabla_{\mathbf{T}} = \nabla \cdot (\mathbf{T} \cdot \nabla)$ , e pode ser numericamente computado por uma diferença finita entre às seis células MAC em torno da célula de referência. Mais especificamente, o  $\mathbf{T}$ -Laplaciano na célula  $(i, j, k)$  de uma quantidade  $q$  é:

$$\begin{aligned} \nabla \cdot \nabla_{\mathbf{T}} q_{i,j,k} = \frac{1}{2h} & (\nabla_{\mathbf{T}}^x q_{i+1,j,k} - \nabla_{\mathbf{T}}^x q_{i-1,j,k} + \\ & \nabla_{\mathbf{T}}^y q_{i,j+1,k} - \nabla_{\mathbf{T}}^y q_{i,j-1,k} + \\ & \nabla_{\mathbf{T}}^z q_{i,j,k+1} - \nabla_{\mathbf{T}}^z q_{i,j,k-1}), \end{aligned} \quad (3.6)$$

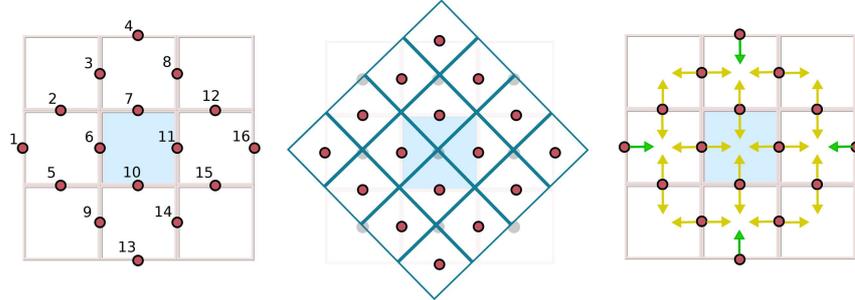
onde a notação  $\nabla_{\mathbf{T}}^x q_{i+1,j,k}$  significa o componente  $x$  do  $\mathbf{T}$ -Gradiente em  $(i + 1, j, k)$  de uma quantidade  $q$ . A mesma notação é válida para os componentes  $y$  e  $z$ . Note que o  $\mathbf{T}$ -Laplaciano depende dos 6 vizinhos da célula  $(i, j, k)$  em 3D, composta por 36 faces no total. A Figura 2 ilustra como esses operadores são computados em 2D.

### 3.1.2 Máscara $\mathbf{T}$ -Laplaciana

Computar a Equação 3.6 em cada iteração é custoso. Para evitar esses cálculos recorrentes, seguimos a abordagem adotada em [Renhe et al., 2019] e pré-computamos a máscara  $\mathbf{T}$ -Laplaciana fixa para cada célula. Isso é baseado ao observar que todas as operações acima são lineares. Como resultado, a computação do  $\mathbf{T}$ -Laplaciano é uma combinação linear de quantidades  $q$  a partir das 36 faces mais próximas.

A versão 2D do esquema acima é mostrado na Figura 3. As  $4 \times 4$  faces a partir dos quatro vizinhos mais próximos são representadas pelos pontos vermelhos enumerados. A célula azul é a célula de referência para a computação. Os pesos para cada face são representados pelos pontos vermelhos enumerados. Na esquerda, os pesos para a computação da máscara 2D centrada na célula: 0.5 para faces internas (amarelo) e 1 para as faces nas bordas (verde). A máscara  $\mathbf{T}$ -Laplaciana 2D é computada no centro da célula

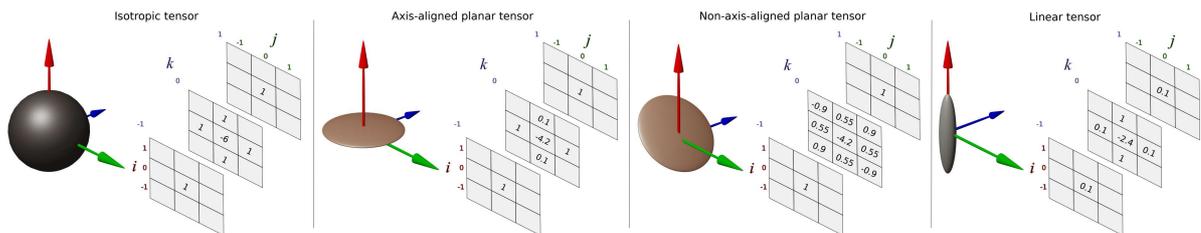
- Figura 4 - Ilustração de uma máscara **T**-Laplaciana para uma grade MAC 2D.



Fonte: Extraído de Parreiras et al. [2021]

de referência, sombreada de azul. A enumeração das faces é útil para a computação dos pesos da máscara corrigida para cada célula.

- Figura 5 - Exemplos de máscara **T**-Laplaciana pré-computada para diferentes tensores.



Fonte: Extraído de Parreiras et al. [2021]

Na Figura 4 temos exemplos de máscaras **T**-Laplaciana pré-computadas para diferentes tensores. Da esquerda para a direita, nos temos isotrópico, planar, e linear Kindlmann [2004]. Para os tensores planares, duas variações são apresentadas: a da esquerda é alinhada com os eixos, enquanto a outra é rotacionada  $45^\circ$  no eixo  $z$ . Em ambos os casos, nós temos  $\lambda_3 = 0.1$ . Para o tensor linear, definimos  $\lambda_2 = \lambda_3 = 0.1$ . Todos os outros autovalores foram definidos para 1. Note como a máscara é afetada pelo formato e orientação do tensor. Os elementos em branco da máscara são zero. Para os tensores alinhados com os eixos, a maioria dos elementos são zero. Todos os pesos da máscara podem diferir de zero para autovetores arbitrários.

Em 3D, a célula de referência é localizada em  $(i, j, k)$  e tem 6 células vizinhas mais próximas, cada uma com 6 faces. Considere que as 36 faces são enumeradas pelos índices  $f \in \{1, 2, \dots, 36\}$ . A geometria da grade MAC nesse caso é bastante complicada de se descrever. Agora, suponha que a notação  $g_{i,j,k}(f)$  significa uma quantidade escalar  $g$  na face  $f$  em torno de  $(i, j, k)$ . Os 36 pesos  $w_{i,j,k}(f)$  são computados aplicando o operador

**T**-Laplaciano em múltiplos campos escalares como:

$$w_{i,j,k}(f) = \nabla \cdot \nabla_{\mathbf{T}} S(f), \quad (3.7)$$

onde  $S(f) = g_{i,j,k}^f(a)$  é um conjunto de campos escalares com a forma:

$$g_{i,j,k}^f(a) = \begin{cases} 1, & \text{se } f = a, \\ 0, & \text{c.c.,} \end{cases} \quad (3.8)$$

e  $a \in \{1, 2, \dots, 36\}$ . Isto é,  $g_{i,j,k}^f(a)$  tem um pulso unitário na face  $a$  enquanto todos os outros são zero. Os pesos centrados na face resultantes  $w_{i,j,k}(f)$  são apenas dependentes de suas posições e dos respectivos auto-sistemas dos tensores utilizados na computação do operador **T**-Laplaciano (Equação 3.6).

Portanto, o **T**-Laplaciano no centro de uma célula  $(i, j, k)$  de qualquer quantidade  $q$  armazenada nas faces vizinhas  $f$  pode ser computada como:

$$\nabla \cdot \nabla_{\mathbf{T}} q_{i,j,k} = \sum_{f=1}^{36} w_{i,j,k}(f) \cdot q_{i,j,k}(f). \quad (3.9)$$

### 3.1.3 Sistema Linear para o Problema de **T**-Poisson

Nesta etapa, um sistema linear  $(\Delta t/\rho) \cdot \mathbf{A}\mathbf{p} = \mathbf{d}$  é resolvido para obter o campo escalar  $p(\mathbf{r})$  (Equação 3.3) a partir do vetor de incógnitas  $\mathbf{p}$ . O vetor  $\mathbf{d}$  é o divergente centrada na célula da velocidade, computado a partir de:

$$\begin{aligned} \nabla \cdot u_{i,j,k} = \frac{1}{h} & (u_{i+1/2,j,k} - u_{i-1/2,j,k} + \\ & u_{i,j+1/2,k} - u_{i,j-1/2,k} + \\ & u_{i,j,k+1/2} - u_{i,j,k-1/2}). \end{aligned} \quad (3.10)$$

A matriz de coeficientes  $A$  é construída a partir da Equação 3.9 e tem as suas dimensões  $n_c \times n_f$ , onde  $n_c$  é o número de células em que o fluido está presente, e  $n_f$  é o número de faces respectivo. Como tal, o método proposto utiliza as quantidades a partir das faces para obter um **T**-Laplaciano centrado na célula. Isso implica que  $\mathbf{p}$  tem  $n_f \times 1$  quantidade de incógnitas. Entretanto, como  $n_c < n_f$ , o sistema linear é sub-determinado. Uma solução possível é criar um mapeamento das faces de  $A$  nos respectivos centros de células como:

$$A' = \Phi(A), \quad (3.11)$$

onde  $A'$  é uma matriz de tamanho  $n_c \times n_c$ .

O mapa  $\Phi : A \rightarrow A'$  pode ser baseado em uma combinação linear dos 36 pesos  $w_{i,j,k}(f)$  de uma máscara **T**-Laplaciana centrada na face. Iremos assumir que cada face contribui igualmente para todas as suas células incidentes. A face com duas células

incidentes contribuem com metade de seus pesos para cada uma. O peso de uma face na borda é totalmente dedicada a sua única célula incidente. Portanto, os 36 pesos das faces assumem a forma de 27 pesos centrados na célula. Na prática, o resultado dessa distribuição é uma máscara **T**-Laplaciana  $m_{i,j,k}(c)$  centrada na célula, onde  $c \in \{1, 2, \dots, 27\}$  é o índice de todas as células centradas ao redor de  $(i, j, k)$ , inclusive. A versão 2D desse processo é descrito na direita da Figura 3.

Por fim, o **T**-Laplaciano na célula  $(i, j, k)$  de qualquer quantidade  $q'_{i,j,k}$  armazenada no centro da célula é computada como:

$$\nabla \cdot \nabla_{\mathbf{T}} q'_{i,j,k} = \sum_{c=1}^{27} m_{i,j,k}(c) \cdot q'_{i,j,k}(c), \quad (3.12)$$

Para a equação de **T**-Poisson (Equação 3.3), nós procuramos encontrar a pseudo-pressão  $p(\mathbf{r})$  centrada na célula. A máscara  $m_{i,j,k}$  com dimensão  $3 \times 3 \times 3$  é pré-computada e armazenada na célula  $(i, j, k)$ , dado que o campo tensorial é estático dentro de toda a simulação. Essa máscara é ilustrada pela Figura 4, onde os glifos representam os tensores isotrópico, linear, e planar, como proposto por Kindlmann [2004].

A matriz  $A'$  no sistema linear final  $(\Delta t/\rho) \cdot A' \mathbf{p}' = \mathbf{d}$  é simétrica se o campo tensorial tem a forma  $\mathbf{T}(\mathbf{r}) = \gamma \mathbf{I}$ , para qualquer valor positivo  $\gamma$ . Caso  $\gamma = 1$ , a solução é a mesma encontrada pela projeção de Helmholtz isotrópica. No caso geral, a matriz é assimétrica para tensores anisotrópicos. Devido à natureza assimétrica do sistema linear, o método do Gradiente Biconjugado Estabilizado (BI-CGSTAB) Van der Vorst [1992], preconditionado com a fatoração LU incompleta, se apresenta mais adequado para solucioná-lo.

A solução  $\mathbf{p}'$  da equação de **T**-Poisson é um vetor de dimensão  $n_c \times 1$  que representa o campo escalar  $p(\mathbf{r})$  na grade  $G$ . O próximo passo é computar o gradiente de  $p$  nas faces. Para uma face direita em  $x$ , o gradiente é computado como:

$$\nabla p_{i+1/2,j,k} = \frac{1}{h} \begin{bmatrix} p_{i+1,j,k} - p_{i,j,k} \\ (p_{i+1,j+1,k} - p_{i+1,j-1,k} + p_{i,j+1,k} - p_{i,j-1,k})/2 \\ (p_{i+1,j,k+1} - p_{i+1,j,k-1} + p_{i,j,k+1} - p_{i,j,k-1})/2 \end{bmatrix}, \quad (3.13)$$

onde as componentes  $y$  e  $z$  são a média dos gradientes das duas células incidentes (Figura 5). Portanto, a partir da Equação 2.37, o **T**-Gradiente de  $p$  em uma face direita alinhada em  $x$  é:

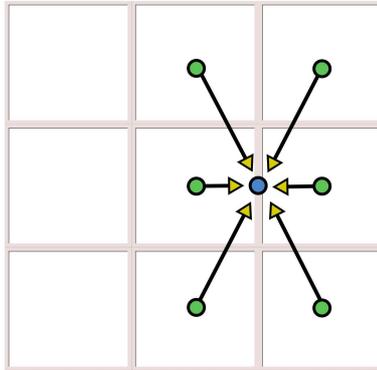
$$\nabla_{\mathbf{T}} p_{i+1/2,j,k} = \frac{\mathbf{T}_{i,j,k} + \mathbf{T}_{i+1,j,k}}{2} \nabla p_{i+1/2,j,k}. \quad (3.14)$$

Por fim, a velocidade escalar projetada é:

$$P_{\mathbf{T}}(u_{i+1/2,j,k}) = u_{i+1/2,j,k} - \nabla_{\mathbf{T}} p_{i+1/2,j,k} \cdot \mathbf{n}, \quad (3.15)$$

onde  $\mathbf{n}$  é a normal da face. A projeção é análoga para as 5 faces restantes (Equação 3.4). Entretanto, é apenas aplicada para as faces internas ao fluido ou em faces com borda fluido-vazio. A atualização da velocidade na borda com sólido deve considerar a condição de Neumann, que será explicada na Seção 3.1.4.

– Figura 6 - O gradiente da pressão  $p$ , definido na Equação 3.13, é calculado na face. Em 2D, seis células contribuem para o cálculo do gradiente em um único gradiente.



Fonte: Extraído de Parreiras et al. [2021]

### 3.1.4 Condições de Borda

A simulação de fluidos monofásicos precisa lidar com bordas fluido-vazio e fluido-sólido. A condição de borda de Dirichlet é usada para garantir a pseudo-pressão adequada para as células vazias. Basicamente, a pseudo-pressão  $p$  em qualquer célula vazia  $(i, j, k)$  é forçada a zero:

$$p_{i,j,k} = 0, \quad (3.16)$$

impactando diretamente a matriz de coeficientes do sistema linear de  $\mathbf{T}$ -Poisson (Seção 3.1.3).

A condição de borda de Neumann é usada para garantir o gradiente da pressão apropriado nas faces. Primeiro, iremos assumir que a borda esteja em uma face direita alinhada em  $x$  cuja célula de referência  $(i, j, k)$  contém fluido, enquanto a célula  $(i + 1, j, k)$  é sólida. A condição de borda de Neumann é:

$$\frac{\Delta t}{\rho} \cdot [\nabla_{\mathbf{T}} p_{i+1/2,j,k}] \cdot \mathbf{n}_s = [\mathbf{u}_{i+1/2,j,k} - \mathbf{u}_{i+1/2,j,k}^s] \cdot \mathbf{n}, \quad (3.17)$$

onde  $\nabla_{\mathbf{T}} p_{i+1/2,j,k}$  é o gradiente da pressão na borda,  $u_{i+1/2,j,k}$  é a velocidade armazenada na face, e  $\mathbf{u}_{i+1/2,j,k}^s$  é a velocidade do sólido na face. Note que para um tensor  $\mathbf{T} = \mathbf{I}$ , a Equação 3.17 se iguala a formulação clássica. Esse método é facilmente adaptado para as outras cinco faces.

## 3.2 EVOLUÇÃO DA SUPERFÍCIE DO FLUIDO

Suponha que a borda está em uma face alinhada em  $x$  cuja célula de referência  $(i, j, k)$  contém fluido, enquanto a célula  $(i + 1, j, k)$  é vazia. O método seguinte é facilmente adaptado para as outras 5 faces.

Esse procedimento é aplicado antes da construção do sistema linear de  $\mathbf{T}$ -Poisson e, consequentemente, modifica o divergente da velocidade considerada na projeção. Normalmente, velocidades na borda fluido-vazio não são modificados para permitir que o fluxo cruze as faces.

Neste caso específico, nós precisamos definir como o tensor externo  $\mathbf{T}_{ext} = \mathbf{T}_{i+1,j,k}$  se envolve com a superfície do fluido. Nesta seção, vamos propor que o tensor module a troca em bordas fluido-vazio, enquanto mantêm a formulação clássica para sólidos, resultando em:

$$u_{i+1/2,j,k}^{Vazio} = \eta \mathbf{u} \cdot \mathbf{n}, \quad (3.18)$$

onde  $\mathbf{u} = [u_{i+1/2,j,k} \ 0 \ 0]^T$  é a atual velocidade na face em forma vetorial,  $\mathbf{n}$  é a normal da face, e o fator escalar:

$$\eta = (2\mathbf{T}_{ext} - \mathbf{I}) \mathbf{n} \cdot \mathbf{n}, \quad (3.19)$$

é truncado para satisfazer  $\eta \in (-1, 1]$ . Sempre teremos  $\eta > -1$  já que  $\mathbf{T}_{ext}$  é positivo definido. Se o vetor  $\mathbf{T}_{ext}\mathbf{n}$  é pequeno na direção de  $\mathbf{n}$ , então  $\eta \approx -1$  e a face funciona como uma barreira. A medida que  $\mathbf{T}_{ext}\mathbf{n} \cdot \mathbf{n}$  aumenta, a face se torna permeável. Se  $\eta = 1$ , significa que  $\mathbf{T}_{ext}\mathbf{n} \cdot \mathbf{n} \geq 1$ , a face se torna uma superfície livre. Esta condição é aplicada se  $\mathbf{u} \cdot \mathbf{n} > 0$ , i.e. quando o fluido está se movendo em direção à uma célula vazia. Caso contrário,  $\eta$  é zero.

### 3.3 ADVECÇÃO ANISOTRÓPICA

A advecção em métodos *particle-in-cell* é tipicamente resolvida por um método de diferença finita avançada. A abordagem mais comum é o Runge-Kutta de terceira ordem. Nosso método resolve ambos os passos de advecção e de deflexão (Equação 2.38). Mais especificamente,  $\mathbf{T}\mathbf{u} - \mathbf{u}$  é adicionado como uma aceleração nos termos do Runge-Kutta. Considerando  $\mathbf{x}^n$  a posição inicial da partícula na iteração  $n$ , obtemos:

$$\begin{aligned} \mathbf{k}_1 &= \hat{\mathbf{u}}(\mathbf{x}^n) + \Delta t \left( \hat{\mathbf{T}}(\mathbf{x}^n) \hat{\mathbf{u}}(\mathbf{x}^n) - \hat{\mathbf{u}}(\mathbf{x}^n) \right), \quad \mathbf{x}' = \mathbf{x}^n + \frac{1}{2} \Delta t \mathbf{k}_1, \\ \mathbf{k}_2 &= \hat{\mathbf{u}}(\mathbf{x}') + \Delta t \left( \hat{\mathbf{T}}(\mathbf{x}') \hat{\mathbf{u}}(\mathbf{x}') - \hat{\mathbf{u}}(\mathbf{x}') \right), \quad \mathbf{x}'' = \mathbf{x}^n + \frac{3}{4} \Delta t \mathbf{k}_2, \\ \mathbf{k}_3 &= \hat{\mathbf{u}}(\mathbf{x}'') + \Delta t \left( \hat{\mathbf{T}}(\mathbf{x}'') \hat{\mathbf{u}}(\mathbf{x}'') - \hat{\mathbf{u}}(\mathbf{x}'') \right), \\ \mathbf{x}^{n+1} &= \mathbf{x}^n + \frac{2}{9} \Delta t \mathbf{k}_1 + \frac{3}{9} \Delta t \mathbf{k}_2 + \frac{4}{9} \Delta t \mathbf{k}_3, \end{aligned} \quad (3.20)$$

onde  $\hat{\mathbf{u}}(\cdot)$  é a velocidade interpolada, e  $\hat{\mathbf{T}}(\cdot)$  é o tensor interpolado na dada posição. O efeito é que a velocidade usada no cálculo é forçada a se alinhar com as principais direções do tensor. O resultado são partículas transportadas por caminhos definidos pelo campo tensorial, escalados por seus autovalores.

### 3.4 VISÃO GERAL DOS PASSOS DA SIMULAÇÃO

Esta seção resume como o método é executado. Seja  $P$  o conjunto de todas as partículas no domínio da simulação. Uma descrição rápida do algoritmo iterativo é:

1. Semear um conjunto  $P$  de partículas de fluido;
2. Transferir as velocidades de  $P$  para as faces da grade  $G$ ;
3. Classifique todas as células como fluido, sólido ou vazio;
4. FLIP: Salve uma cópia das velocidades nas faces;
5. Adicione as forças externas nas faces de  $G$ ;
6. Monte o sistema linear utilizando as condições de borda para  $\mathbf{T}$ -Poisson apenas para as células que contêm fluido;
7. Resolva a equação de  $\mathbf{T}$ -Poisson e compute a **projeção anisotrópica**;
8. Compute as novas velocidades, armazenando o resultado nas faces de  $G$ ;
9. FLIP: Interpole e adicione a diferença entre as novas velocidades e as velocidades salvas a partir das faces de  $G$  para as velocidades das partículas em  $P$ ;
10. Aplique a **advecção anisotrópica** em  $P$ .

A Etapa 1 insere partículas no domínio quando necessário, dependendo especificamente da simulação. As Etapas 2 e 9 são executadas interpolando as velocidades, considerando suas distâncias para cada centro da face da célula. Para uma face alinhada em  $x$ , a interpolação da velocidade é dada pela Equação 2.44. As outras cinco faces são interpoladas de forma análoga.

A combinação entre a solução encontrada pelo PIC e FLIP é interpolada e adicionada na Etapa 9 da seguinte Equação 2.50.

## 4 MÉTODO BASEADO EM FACES E ARESTAS

O método apresentado no Capítulo 3 é limitado em aliviar a pressão através das faces das células no passo da projeção. Além disso, a necessidade de se utilizar dois operadores  $\mathbf{T}$ -Gradientes diferentes, um para o operador  $\mathbf{T}$ -Laplaciano e outro na atualização da velocidade (Equações 3.5 e 3.14), pode gerar soluções pobres com divergente não nulo. Além de que a falta de simetria no sistema linear, mesmo para tensores alinhados com os eixos, pode gerar matrizes de coeficientes mal condicionadas e sistemas de difícil solução.

O problema da projeção anisotrópica é, portanto, primordial para a estabilidade da simulação de fluidos invíscidos modulados por campos tensoriais. Este capítulo se utiliza das mesmas definições para o problema de  $\mathbf{T}$ -Poisson que foi apresentado na Equações 3.3 e 3.4.

Este capítulo provê uma solução numérica para o problema de  $\mathbf{T}$ -Poisson apresentando um operador de divergente que utiliza mais do que seis pontos nas faces da célula para o cálculo, diferente do que foi apresentado no Capítulo 3. Mais especificamente, o propósito é usar adicionalmente as 12 arestas da célula de referência. Como resultado, o operador  $\mathbf{T}$ -Laplaciano requerido para resolver a Equação 3.3 é computado utilizando 19 pontos de referência: o centro da célula, as 6 faces e as 12 arestas. Como as arestas agora fazem parte do esquema numérico, a grade MAC é estendida para armazenar a velocidade nelas. O desafio é garantir a coerência das interpolações partícula-grade e grade-partícula requeridas pelo método FLIP em cada iteração. Nesse caso, cada aresta pode ser interpretada como uma interseção de duas faces da grade MAC. Portanto, diferente das faces, as arestas irão conter duas componentes.

Os operadores discretos, as condições de borda e serão apresentadas na Seção 4.1. Essa é a maior contribuição deste trabalho. Seção 4.3 apresenta as etapas de simulação além de apresentar as interpolações estendidas para grade MAC.

### 4.1 DISCRETIZAÇÃO DA PROJEÇÃO ANISOTRÓPICA

Para propósitos de simulação, dada qualquer célula  $C \subset G$  em uma grade 3D  $G$  é um subconjunto de  $\mathbb{R}^3$ . Considere que uma célula regular de fronteira  $\partial C$  é suave por partes. Suponha que  $\mathbf{v}(\mathbf{r})$  é um campo vetorial continuamente diferenciável definido em  $C$  e seus vizinhos. O teorema do divergente afirma que:

$$\iiint_C \nabla \cdot \mathbf{v} \, dC = \iint_{\partial C} \mathbf{v} \cdot \mathbf{n} \, d\partial C, \quad (4.1)$$

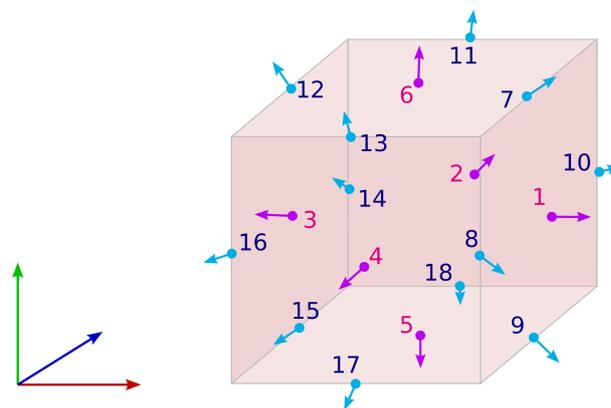
onde o lado esquerdo da equação é uma integral de volume sobre  $C$ , o lado direito da equação é uma integral de superfície sobre a borda  $\partial C$ , e  $\mathbf{n}$  são os vetores normais unitários apontando para fora em cada ponto da variedade fechada  $\partial C$ .

O lado esquerdo da Equação 4.1 é a base para resolver o problema de **T**-Poisson (Equação 3.3) com erros menores. Mais especificamente, ela provê um caminho para envolver mais pontos do que os clássicos centros das faces na solução numérica. Como será mostrado nas seções seguintes, o desafio é criar um modelo numérico que combine ambas Equações 3.3 e 3.4 de forma a compensar corretamente a pseudo-pressão. Isto envolve não apenas um novo operador **T**-Gradiente (Seção 4.1.1), mas também uma nova forma de aplicar a subtração definida na Equação 3.4 que considera múltiplas normais na borda da célula (Seção 4.1.4).

#### 4.1.1 Operadores **T**-Gradiente e **T**-Laplaciano

Considere as faces, arestas e normais de cada célula como ilustrado na Figura 6. A célula central é posicionada em  $(i, j, k)$ ,  $i, j, k \in \mathbb{Z}$ . Os pontos de referência são indexados de 1 até 18, posicionados em  $\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_{18}$  com as suas normais unitárias correspondentes  $\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_{18}$ . Índices de 1 até 6 representam os centros das faces (vermelho). Por exemplo, a face alinhada em  $x$  à direita tem o índice 1, as coordenadas do seu centro são  $\mathbf{r}_1 = (i + 1/2, j, k)$  e a normal  $\mathbf{n}_1 = [1 \ 0 \ 0]^T$ . Analogamente, arestas são representadas entre os índices de 7 até 18 (azul-escuro). Como um exemplo, a aresta acima da face à direita, i.e. aquele sobre o plano  $xy$  em direção ao eixo positivo  $y$ , tem o índice 7 sendo posicionada em  $\mathbf{r}_7 = (i + 1/2, j + 1/2, k)$  com a normal  $\mathbf{n}_7 = [\sqrt{2}/2 \ \sqrt{2}/2 \ 0]^T$ .

– Figura 7 - Pontos na superfície usados para computar o **T**-Laplaciano.



Fonte: Elaborado pelo autor.

Como mostrado na Figura 7, o gradiente para uma face requer 10 células para a

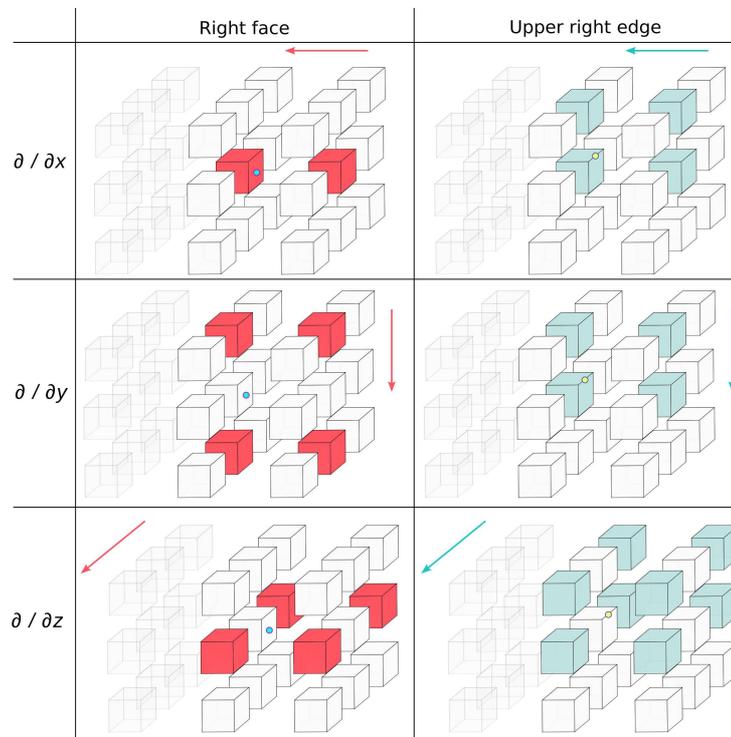
computação. O gradiente para uma face direita alinhada em  $x$  é:

$$\nabla q_1 = \frac{1}{h} \begin{bmatrix} q_{i+1,j,k} - q_{i,j,k} \\ \frac{q_{i+1,j+1,k} - q_{i+1,j-1,k} + q_{i,j+1,k} - q_{i,j-1,k}}{4} \\ \frac{q_{i+1,j,k+1} - q_{i+1,j,k-1} + q_{i,j,k+1} - q_{i,j,k-1}}{4} \end{bmatrix}, \quad (4.2)$$

onde os componentes  $y$  e  $z$  são a média de dois gradientes de suas duas células incidentes, e  $h$  é a distância entre quaisquer duas células adjacentes pela face. Esta notação implica que o índice representa ao centro da face  $(i + 1/2, j, k)$ , como descrito na Figura 6. A Figura 7 esclarece esse esquema. Portanto, a partir da Equação 2.37, o **T**-Gradiente de  $q$  em uma face alinhada em  $x$  é:

$$\nabla_{\mathbf{T}} q_1 = \frac{\mathbf{T}_{i,j,k} + \mathbf{T}_{i+1,j,k}}{2} \nabla q_1. \quad (4.3)$$

– Figura 8 - Computação do esquema para a face à direita (vermelho) e sua aresta superior (verde-claro). Os vetores gradientes se estão posicionados ao centro da face (ponto ciano) e centro da aresta (ponto amarelo). As setas indicam a ordem da diferença finita.



Fonte: Elaborado pelo autor.

O esquema de diferença finita para resolver a equação de Poisson 3D baseada nas seis faces mais a célula central é conhecida como estêncil 7-pontos. Como mostrado em Gupta and Kouatchou [1998], produz uma aproximação com erro de truncamento de ordem

$O(h^2)$ . O esquema de 19-pontos pode ser obtido adicionando mais 12 pontos posicionados nas arestas. O estêncil resultante aproxima a equação de Poisson padrão com um erro de truncamento na ordem de  $O(h^4)$ . Está relacionado ao esquema de Mehrstellen para a equação de Poisson 2D que tem a propriedade de ter erro de discretização isotrópica e tende a produzir evolução de simulação simétrica Ananthakrishnaiah et al. [1987]. Devido a essas propriedades desejáveis, nossa proposta é de adaptar o esquema de 19-pontos para resolver o problema de **T**-Poisson 3D com uma ordem de precisão maior.

Nós, portanto, precisamos estimar o gradiente para as 12 arestas da célula. O gradiente centrado na aresta de uma quantidade  $q$  armazenada nas células requer 12 células para a computação, como mostrado na figura 7. Por exemplo, o gradiente de uma aresta superior da face à direita posicionada em  $(i + 1/2, j + 1/2, k)$  é:

$$\nabla q_7 = \frac{1}{2h} \left[ \begin{array}{c} q_{i+1,j,k} - q_{i,j,k} + q_{i+1,j+1,k} - q_{i,j+1,k} \\ q_{i,j+1,k} - q_{i,j,k} + q_{i+1,j+1,k} - q_{i+1,j,k} \\ \frac{q_{i+1,j,k+1} - q_{i+1,j,k-1} + q_{i,j+1,k+1} - q_{i,j+1,k-1}}{4} + \\ \frac{q_{i,j,k+1} - q_{i,j,k-1} + q_{i+1,j+1,k+1} - q_{i+1,j+1,k-1}}{4} \end{array} \right], \quad (4.4)$$

onde o componente  $z$  é a média de quatro gradientes normais ao plano  $xy$ . O correspondente **T**-Gradiente na aresta é:

$$\nabla_{\mathbf{T}} q_7 = \frac{\mathbf{T}_{i,j,k} + \mathbf{T}_{i+1,j,k} + \mathbf{T}_{i,j+1,k} + \mathbf{T}_{i+1,j+1,k}}{4} \nabla q_7, \quad (4.5)$$

onde o tensor é a média das quatro células incidentes na aresta. As Equações 4.3 e 4.5 são facilmente estendidas para as outras cinco faces e onze arestas, respectivamente.

O operador **T**-Laplaciano contínuo é definido como  $\nabla \cdot \nabla_{\mathbf{T}} = \nabla \cdot (\mathbf{T} \cdot \nabla)$ . Portanto, é necessário definir o operador divergente para o centro da célula. O fluxo total dentro de uma célula é dado pelo teorema do divergente. Nós aproximamos a integral do lado direito da Equação 4.1 utilizando os 18 **T**-Gradientes ao redor da célula  $(i, j, k)$  para obter o operador **T**-Laplaciano como:

$$\nabla \cdot \nabla_{\mathbf{T}} q_{i,j,k} = \frac{1}{3h} \sum_{m=1}^6 \nabla_{\mathbf{T}} q_m \cdot \mathbf{n}_m + \frac{1}{3h\sqrt{2}} \sum_{m=7}^{18} \nabla_{\mathbf{T}} q_m \cdot \mathbf{n}_m, \quad (4.6)$$

onde o primeiro somatório considera os **T**-Gradientes centrados nas faces, enquanto o segundo considera os **T**-Gradientes centrados nas arestas. O produto interno com as normais da superfície naturalmente produzem um esquema de diferença finita central para o divergente.

#### 4.1.2 Máscara **T**-Laplaciana

A Equação 3.6 é computacionalmente custosa. Como todos os operadores são lineares, portanto, cada célula da grade terá o seu próprio estêncil dependendo unicamente

dos tensores vizinhos. Assim como na Seção 3.1.2, iremos pré-computar a máscara  $\mathbf{T}$ -Laplaciana para cada célula.

Em 3D, a célula de referência é localizada em  $(i, j, k)$  e contém as 27 células vizinhas, incluindo ela mesma. Considere que as 27 células são enumeradas pelos índices  $c \in \{1, 2, \dots, 27\}$ . Agora, suponha que a notação  $g_{i,j,k}(c)$  significa uma quantidade escalar  $g$  na célula  $c$  em torno de  $(i, j, k)$ . Os 27 pesos  $w_{i,j,k}(c)$  que formam o estêncil para a célula em  $(i, j, k)$  são computadas aplicando o operador  $\mathbf{T}$ -Laplaciano em múltiplos campos escalares:

$$w_{i,j,k}(c) = \nabla \cdot \nabla_{\mathbf{T}} S(c), \quad (4.7)$$

onde  $S(c) = g_{i,j,k}^c(a)$  é um conjunto de  $3 \times 3 \times 3$  campos escalares com a forma:

$$g_{i,j,k}^c(a) = \begin{cases} 1, & \text{se } c = a, \\ 0, & \text{c.c.,} \end{cases} \quad (4.8)$$

e  $a \in \{1, 2, \dots, 27\}$ . Isto é,  $g_{i,j,k}^c(a)$  tem um pulso unitário na célula  $a$ , enquanto os outros valores são zero. Os  $w_{i,j,k}(c)$  centrados na célula resultante são apenas dependentes respectivamente das suas posições e tensores usados na computação do  $\mathbf{T}$ -Laplaciano (Equação 3.6).

Portanto, o  $\mathbf{T}$ -Laplaciano no centro da célula  $(i, j, k)$  de qualquer quantidade  $q$  armazenado no centro das células pode ser computado como:

$$\nabla \cdot \nabla_{\mathbf{T}} q_{i,j,k} = \sum_{c=1}^{27} w_{i,j,k}(c) \cdot q_{i,j,k}(c). \quad (4.9)$$

Esta máscara é ilustrada na Figura 4, onde os glifos Kindlmann [2004] representam tensores isotrópico, linear, e planar. Como assumimos que o tensor é constante no tempo, a máscara  $w_{i,j,k}(c)$  é computada uma vez para cada célula, armazenada e usada durante a simulação. Se todos os 27 tensores ao redor da célula de referência forem a matriz identidade, a máscara resultante é exatamente o estêncil de 19-pontos mostrado em Gupta and Kouatchou [1998], que propaga o erro isotropicamente e tem o erro de truncamento na ordem de  $O(h^4)$ .

### 4.1.3 Condições de Borda

A simulação de fluido monofásico deve lidar com bordas fluido-vazio e fluido-sólido. A condição de borda de Dirichlet apresentada na Seção 3.1.4 continua válida para a formulação deste capítulo. Entretanto, a condição de Neumann precisa ser complementada, já que a interação fluido-sólido apresentada anteriormente apenas considera borda com as faces (Equação 3.17).

Vamos supor que o limite da borda esteja na aresta superior da face direita posicionada em  $(i + 1/2, j + 1/2, k)$ . As células incidentes são  $\mathbf{c}_1 = (i, j, k)$ ,  $\mathbf{c}_2 = (i + 1, j, k)$ ,

$\mathbf{c}_3 = (i, j + 1, k)$  e  $\mathbf{c}_4 = (i + 1, j + 1, k)$ . Várias configurações de fluido-sólido e fluido-vazio são possíveis envolvendo essas quatro células. A condição de borda é aplicada se qualquer célula for fluida e pelo menos duas células forem sólidas. Deixamos os casos restantes para serem tratados pela condição de contorno de Neumann nas faces. Assim, o limite sólido é uma superfície 2D através das quatro células, passando pela aresta.

Para simplificar a discussão, desacoplamos os componentes  $x$  e  $y$  do gradiente no centro da aresta. Se  $(\mathbf{c}_1$  ou  $\mathbf{c}_2)$  e  $(\mathbf{c}_3$  ou  $\mathbf{c}_4)$  são sólidos, então existe uma superfície limite normal ao eixo  $x$  que passa pela aresta, e a seguinte condição de Neumann é aplicada ao componente  $x$ :

$$\frac{\Delta t}{\rho} \cdot \nabla_{\mathbf{T}} p_{i+1/2, j+1/2, k} \cdot \hat{x} = (u_{i+1/2, j+1/2, k} - u_s) \cdot \hat{x}, \quad (4.10)$$

onde  $u_{i+1/2, j+1/2, k}$  é a atual velocidade armazenada na borda,  $u_s$  é a velocidade do sólido na borda, e  $\hat{x}$  é o vetor unitário na direção de  $x$ . Da mesma forma, se  $(\mathbf{c}_1$  ou  $\mathbf{c}_3)$  e  $(\mathbf{c}_2$  ou  $\mathbf{c}_4)$  são sólidos, então existe um limite superfície normal ao eixo  $y$  que passa pela aresta, e a seguinte condição de Neumann é aplicada ao componente  $y$ :

$$\frac{\Delta t}{\rho} \cdot \nabla_{\mathbf{T}} p_{i+1/2, j+1/2, k} \cdot \hat{y} = (u_{i+1/2, j+1/2, k} - u_s) \cdot \hat{y}, \quad (4.11)$$

onde  $\hat{y}$  é o vetor unitário na direção  $y$ . As outras 11 arestas são tratadas de forma análoga.

Observe que os componentes  $x$  e  $y$  do gradiente de pressão (Equação 4.4), na borda superior da face direita, aparecem no lado esquerdo da Equação 4.10 e 4.11, respectivamente. Portanto, todas as condições de contorno fluido-sólido definem o comportamento do gradiente das faces e arestas correspondentes e devem ser encaixadas no problema  $\mathbf{T}$ -Poisson na Equação 3.3, através do  $\mathbf{T}$ -Laplaciano da Equação 3.6.

#### 4.1.4 Sistema Linear para o Problema de $\mathbf{T}$ -Poisson

Nesta etapa, um sistema linear  $(\Delta t/\rho) \cdot A^c \mathbf{p} = \mathbf{d}$  é resolvido para obter o campo escalar  $p(\mathbf{r})$  (Equação 3.3) do vetor de incógnitas resultante  $\mathbf{p}$ . O sistema linear é construído a partir da Equação 3.3 com as condições de borda impostas pelas Equações 3.16, 3.17, 4.10 e 4.11. O vetor  $\mathbf{d}$  é o divergente da velocidade centrado na célula.

Como este método depende de faces e arestas para calcular o  $\mathbf{T}$ -Laplaciano, precisamos armazenar a velocidade em todos os 18 pontos usados na Equação 3.6. Portanto, precisamos estender a grade MAC para incluir a velocidade no centro das arestas. A interpolação de partícula para grade estendida é discutida na Seção 3.4. O divergente da velocidade no centro da célula é calculada conforme proposto na Equação 3.6, considerando todas as faces e arestas:

$$\nabla \cdot u_{i,j,k} = \frac{1}{3h} \sum_{m=1}^6 \mathbf{u}_m \cdot \mathbf{n}_m + \frac{1}{3h\sqrt{2}} \sum_{m=7}^{18} \mathbf{u}_m \cdot \mathbf{n}_m. \quad (4.12)$$

Observe que a modulação baseada em tensores em limites vazios de fluido (Equação 4.14) é aplicada antes do cálculo do divergente da velocidade.

A matriz de coeficientes  $A^c$  é construída diretamente da Equação 3.9 e tem dimensões  $n_c \times n_c$ , onde  $n_c$  é o número de células em que o fluido está presente. Note que, na Seção 3.1.2, os pesos  $w_{i,j,k}(f)$  são computados sobre uma quantidade  $q$  centrada na face, enquanto na Seção 4.1.2  $w_{i,j,k}(c)$  é computada para uma quantidade  $q$  centrada na célula. Com isso, a matriz de coeficientes  $A^c$  tem as suas dimensões  $n_c \times n_c$ . Portanto, o sistema linear obtido com esse esquema é determinado, não sendo necessário realizar o mapeamento descrito pela Equação 3.11. Assim, o método proposto usa as quantidades das células para obter um  $\mathbf{T}$ -Laplaciana centrado célula. Isso significa que o vetor de incógnitas  $\mathbf{p}$  é um vetor de dimensão  $n_c \times 1$ .

A condição de contorno de Dirichlet afeta a respectiva pressão das células vazias na matriz  $A^c$ . As condições de contorno de Neumann, exemplificadas pelas Equações 3.17, 4.11 e 4.10, fornecem novas equações para o sistema linear onde as pressões aparecem no lado esquerdo da equação e as velocidades aparecem no lado direito. Cada linha na matriz  $A^c$  e um valor constante de  $\mathbf{d}$  é uma equação  $\mathbf{T}$ -Poisson referente a uma célula de fluido. Assim, cada condição de contorno de Neumann é somada às equações das células que estão adjacentes à sua face ou aresta. As condições de borda para face são somadas a exatamente uma equação na matriz  $A^c$ , enquanto as condições de borda para aresta podem ser somadas a no máximo duas equações.

Se o campo tensorial tem a forma  $\mathbf{T}(\mathbf{r}) = \mathbf{I}$ , a solução é a mesma da projeção isotrópica de Helmholtz com estêncil de 19-pontos. A matriz  $A^c$  é simétrica se o campo tensorial tiver apenas autovetores alinhados ao eixo. O sistema linear é mais fácil de resolver neste caso. Esta é uma contribuição do nosso método, visto que os trabalhos anteriores fornecem sistemas lineares assimétricos se tensores anisotrópicos estiverem presentes Renhe et al. [2019], Vieira et al. [2021], Parreiras et al. [2021]. Com autovetores arbitrários, o sistema linear é facilmente resolvido pelo método de Gradiente Biconjugado Estabilizado (BI-CGSTAB) [Van der Vorst, 1992], combinado com um pré-condicionador de fatoração LU incompleta esparsa.

A solução  $\mathbf{T}$ -Poisson  $\mathbf{p}$  é um vetor  $n_c \times 1$  que representa um campo escalar  $p(\mathbf{r})$  na grade  $G$ . O próximo passo é calcular o  $\mathbf{T}$ -Gradiente de  $p$  em todas as faces e arestas (Equações 4.3 e 4.5). Nossa proposta é aplicar a subtração necessária pela projeção (Equação 3.4) diretamente às velocidades armazenadas nas faces e arestas com:

$$P_{\mathbf{T}}(\mathbf{u}_m) = \mathbf{u}_m - (\nabla_{\mathbf{T}} p_m) \odot \tilde{\mathbf{n}}_m, \quad \text{for } m \in \{1, 2, \dots, 18\}, \quad (4.13)$$

onde  $\odot$  é o produto Hadamard,  $\nabla_{\mathbf{T}} p_i$  é o  $\mathbf{T}$ -Gradiente no ponto com índice  $m$  (Figura 6), calculado a partir da Equação 4.3 para  $m \leq 6$ , e da Equação 4.5 para  $m \geq 7$ . As coordenadas do vetor  $\tilde{\mathbf{n}}_m$  são compostas por 0 se a coordenada correspondente da normal de superfície  $\mathbf{n}_m$  for zero, ou 1 caso contrário. Ou seja, os eixos com componentes não

ulos têm 1 na componente do vetor  $\tilde{\mathbf{n}}_m$ . Assim, o produto de Hadamard descarta os componentes de  $\nabla_{\mathbf{T}} p_m$  que não estão presentes na normal correspondente. Apenas uma coordenada é preservada para faces e duas coordenadas são preservadas para arestas. A velocidade livre de divergência resultante é transportada para as partículas via interpolação.

## 4.2 EVOLUÇÃO DA SUPERFÍCIE DO FLUIDO

Nosso propósito é que a superfície do fluido deveria evoluir conforme a anisotropia do tensor. Isso é obtido aplicando os tensores diretamente nas velocidades da borda para modular a troca de fluido. A ideia é obter efeitos intermediários na borda, entre a obstrução de uma condição fluido-sólido e a permeabilidade total de uma borda fluido-vazio.

Considere a borda fluido-vazio em uma face alinhada em  $x$ . A célula de referência  $(i, j, k)$  contém fluido enquanto a célula  $(i + 1, j, k)$  é vazia. O tensor usado para a computação do  $\mathbf{T}$ -Gradiente é  $\mathbf{T}_{(i+1/2,j,k)} = (\mathbf{T}_{i,j,k} + \mathbf{T}_{i+1,j,k})/2$  (Equação 4.3).

Suponha que  $\mathbf{T}_{(i+1/2,j,k)} \cdot \mathbf{n}_1 \approx 0$ , i.e., o tensor aplicado na normal da face direita  $\mathbf{n}_1$  é um valor muito pequeno. Uma possível interpretação é que a face funcionaria como uma barreira para o fluido. Nesse caso, queremos que  $\nabla_{\mathbf{T}} p_1 \cdot \mathbf{n}_1$  seja muito pequena na computação do  $\mathbf{T}$ -Laplaciano (Equação 3.6). Nós também queremos que a velocidade  $\mathbf{u}_1 \cdot \mathbf{n}_1$  seja muito pequena. Nessa circunstância ira tender a funcionar como uma condição de borda fluido-sólido (Seção 4.1.3). Agora, suponha que  $|\mathbf{T}_{(i+1/2,j,k)} \cdot \mathbf{n}_1| = 1$ , i.e., a face é uma superfície livre. Nesse caso é análogo a ter  $\mathbf{T}_{(i+1/2,j,k)} = \mathbf{I}$  na computação do  $\mathbf{T}$ -Laplaciano, e a velocidade não deve ser modificada. Por fim, se  $|\mathbf{T}_{(i+1/2,j,k)} \cdot \mathbf{n}_1| > 1$ , queremos que  $\nabla_{\mathbf{T}} p_1 \cdot \mathbf{n}_1 > 1$  que rege a anisotropia local na borda, e um ganho de velocidade na mesma proporção.

Por definição do  $\mathbf{T}$ -Gradiente, o  $\mathbf{T}$ -Laplaciano considera todos esses casos. Apenas a velocidade na borda precisa ser modificada. Portanto, propomos que a velocidade de todas as faces e arestas na borda sejam transformadas como:

$$\mathbf{u}_m \mapsto \mathbf{T}_m \mathbf{u}_m, \quad \text{for } m \in \{1, 2, \dots, 18\}, \quad (4.14)$$

onde  $\mathbf{T}_m$  é o mesmo tensor usado pelos operadores  $\mathbf{T}$ -Gradiente (Equações 4.3 e 4.5). Isto modifica o divergente requerido no lado direito da equação do problema de  $\mathbf{T}$ -Poisson (Equação 3.3), discretizada pela Equação 4.12.

## 4.3 VISÃO GERAL DOS PASSOS DA SIMULAÇÃO

Esta seção resume como a simulação evolui no tempo. Seja  $P$  o conjunto de todas as partículas no domínio de simulação. O esboço do algoritmo iterativo é:

1. Semeie um conjunto  $P$  de partículas de fluido;

2. Transfira as velocidades de  $P$  para as faces e arestas da grade  $G$ ;
3. Classifique todas as células como fluido, sólido ou vazio;
4. FLIP: salve uma cópia das velocidades das faces e das arestas;
5. Adicione forças externas às faces e arestas de  $G$ . Aplique a modulação baseada em tensores nos limites fluido-vazios;
6. Construa o sistema linear **T**-Poisson a partir de células de fluido, aplicando as condições de contorno de Dirichlet e Neumann;
7. Resolva a equação **T**-Poisson;
8. Calcule a **projeção anisotrópica** e armazene as novas velocidades nas faces e arestas de  $G$ ;
9. FLIP: Interpole e some a diferença entre as velocidades novas e salvas das faces e arestas em  $G$  às velocidades das partículas em  $P$ ;
10. Aplicar advecção anisotrópica (Seção 3.3) em  $P$ .

A Etapa 1 insere partículas no domínio conforme necessário. As Etapas 2 e 9 são realizadas via interpolação de velocidades, considerando sua distância ao centro das faces e arestas. Para um ponto de referência com índice  $m \in \{1, 2, \dots, 18\}$  ao redor da célula  $(i, j, k)$ , a interpolação de velocidade é:

$$\mathbf{u}_m = \frac{\sum_{a \in N} \theta(\mathbf{r}_a - \mathbf{r}_m)(\mathbf{u}_a \odot \tilde{\mathbf{n}}_m)}{\sum_{a \in N} \theta(\mathbf{r}_a - \mathbf{r}_m)}, \quad (4.15)$$

onde  $N \subset P$  são as partículas dentro das células vizinhas,  $\theta$  é uma função de interpolação,  $\mathbf{r}_a$  é a posição da partícula  $a$ ,  $u_a$  é a velocidade da partícula,  $\mathbf{r}_m$  é a posição central da face ou aresta, e  $\tilde{\mathbf{n}}_m$  é o mesmo vetor direcional definido para a projeção na Equação 4.13.

Das Etapas descritas na Seção 3.4, apenas a Etapa 1 de semeadura e a Etapa 3 de classificação não foram alteradas, dado que, devido à incorporação das arestas na grade e a mudança do método de projeção, os outros passos precisaram sofrer alterações.

## 5 EXPERIMENTOS E DISCUSSÕES

Esta seção apresenta simulações de fluidos obtidas a partir de ambos os métodos. Todos os experimentos foram executados em um AMD Ryzen 5 3600 com 6 núcleos 3,6 GHz, 12 threads, 16 GB de RAM e GPU AMD RX5700. A densidade do fluido é fixa como  $\rho = 1$  em todas as simulações. O comprimento do lado da célula é  $h = 1$  m em todos os experimentos. Assumimos que as faces e arestas sólidas são estáticas e que as células sólidas têm a matriz identidade como tensor. A grade MAC com nossas condições de contorno propostas exige uma análise mais aprofundada para definir uma condição Courant–Friedrichs–Lewy (CFL) adequada. A condição CFL anisotrópica de limite inferior apresentada em Renhe et al. [2019] funciona para malhas regulares com soluções Eulerianas, porém pode ser usado como um passo de tempo de referência. Apenas células contendo fluido formam o sistema linear para resolver o **T**-Poisson (Seção 3.1.3). Devido ao BI-CGSTAB para resolver sistemas assimétricos, a complexidade de tempo é dominada pelo pré-condicionador de fatoração LU incompleta esparsa em cada iteração. Em cada um dos resultados a seguir, mantivemos  $\Delta t$  fixo em todas as iterações, baixo o suficiente para garantir a convergência numérica em todas as configurações. Geramos sequências de 30 quadros por segundo durante 20 segundos. O comprimento do lado da célula é  $h = 1$  m em todos os experimentos. Todas as imagens foram geradas usando Houdini [hou, 2021] e Paraview [par, 2021] para criação de malha, e Blender [Ble, 2021] para renderização.

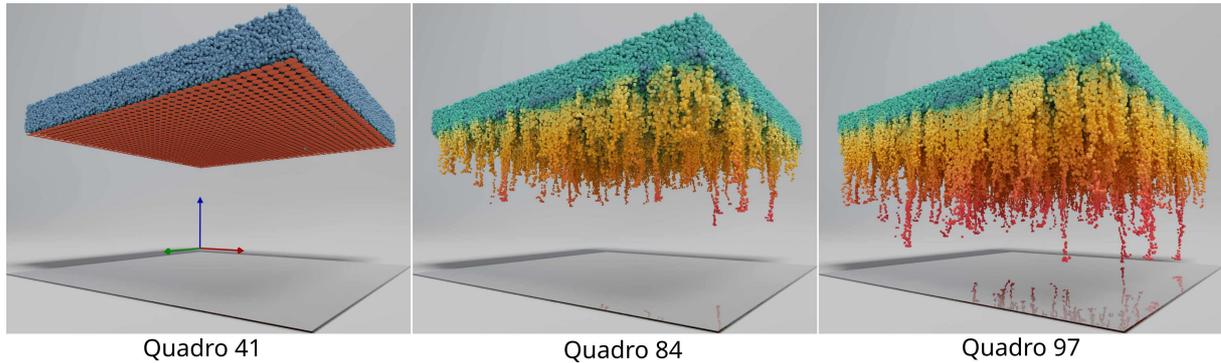
### 5.1 EXPERIMENTOS COM A PROJEÇÃO BASEADA EM FACES

Nos experimentos a seguir, utilizamos uma força de gravidade  $\mathbf{g} = [0, 0, -10]^T$  m/s<sup>2</sup>. A escala de cores para representar as magnitudes de velocidade variam entre azul (0 m/s) a vermelho (25 m/s). Os experimentos a seguir foram executados utilizando o reforço de Neumann proposto em Parreiras et al. [2021] que não foi apresentado nesse texto. Porém, devido ao pouco tempo para a execução dos experimentos, não houve a possibilidade de refazer os experimentos e obter as novas imagens. Entretanto, sabemos que haveria pouca modificação em relação aos resultados finais, mas nos comprometemos a obter novas renderizações para o texto final.

#### 5.1.1 Simulação com tensores planares

A Figura 8 apresenta um conjunto de 491.520 partículas interagindo com uma camada de tensores planares com autovalores  $\lambda_1 = \lambda_2 = 1, 0$  e  $\lambda_3 = 0, 1$  (glifos de disco laranja na Figura 8). Os tensores planares estão em  $k = 30$ , e os voxels restantes da grade têm tensores isotrópicos. A simulação começa com cinco camadas de fluido sobre a camada planar. A Figura 8 mostra três quadros resultantes. O quadro 41 mostra os

– Figura 9 - Piscina de fluido interagindo com tensores planares.



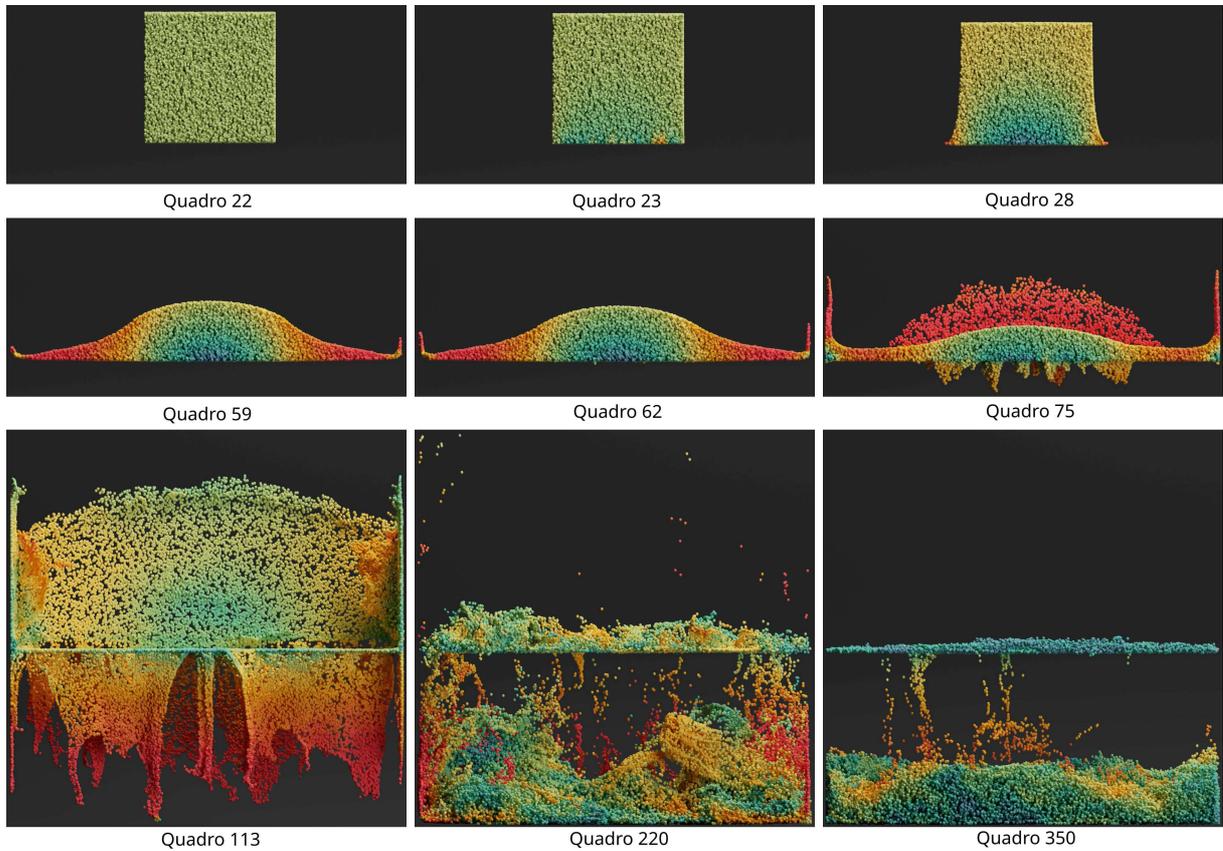
Fonte: Extraído de Parreiras et al. [2021].

tensores planares como discos laranja abaixo do reservatório de fluido. Este primeiro exemplo mostra como um campo com tensores planares pode ser usado para reproduzir efeitos de porosidade. O volume é representado por uma grade MAC  $64 \times 64 \times 64$  com paredes delimitadoras marcadas como limites sólidos. Uma única camada de tensores planares é posicionada em  $k = 32$ , com autovalores  $\lambda_1 = \lambda_2 = 1$  e  $\lambda_3 = 0,1$ . Todas as outras células da grade são tensores isotrópicos unitários (matrizes de identidade). Inicialmente, uma poça de fluido com 6 camadas é posicionada logo acima dos tensores planares ( $33 \leq k \leq 39$ ). Cada célula de fluido tem 20 partículas com velocidade zero. O passo de tempo  $\Delta t = 0,03$  s é fixo durante toda a simulação. Como  $\lambda_3 = 0,1$ , a velocidade do fluido se desenvolve muito lentamente na direção  $z$  sujeita à força da gravidade. Neste caso, as partículas são desaceleradas 10% por unidade de tempo dentro das células planares. Eventualmente, as primeiras partículas passam através da camada planar. Eles tendem a deixar pressões negativas que puxam outras partículas. O resultado é a formação de gotículas distribuídas uniformemente abaixo do plano conforme mostrado no quadro 84. O quadro 97 mostra gotículas atingindo a parte inferior do volume. Observe o efeito fluido parecendo chuva. As cores das partículas indicam a magnitude da velocidade do mais baixo (azul-claro) ao mais alto (vermelho claro). Este exemplo mostra o fluido gerando naturalmente cascatas uniformemente distribuídas sobre o plano. Assim, nossa formulação proporcionou uma simulação suave.

A Figura 9 apresenta um cubo de fluido com 185.220 partículas caindo sobre uma camada de tensores planares com autovalores  $\lambda_1 = \lambda_2 = 1,0$  e  $\lambda_3 = 0,1$ . A visão ortográfica enfatiza a interação das partículas com a camada de tensores. Além disso, as imagens são cortes transversais no plano  $i = 32$  para ser possível visualizar o interior do cubo (metade do volume é omitido). A magnitude da velocidade é ilustrada do azul (mais baixo) ao vermelho (mais alto). Durante os Quadros 22, 23, 28 e 59 o fluido é mantido acima dos tensores planares devido à condição de evolução da superfície descrita na Seção

## 3.2.

– Figura 10 - Cubo de fluido caindo sob uma camada de tensores planares.

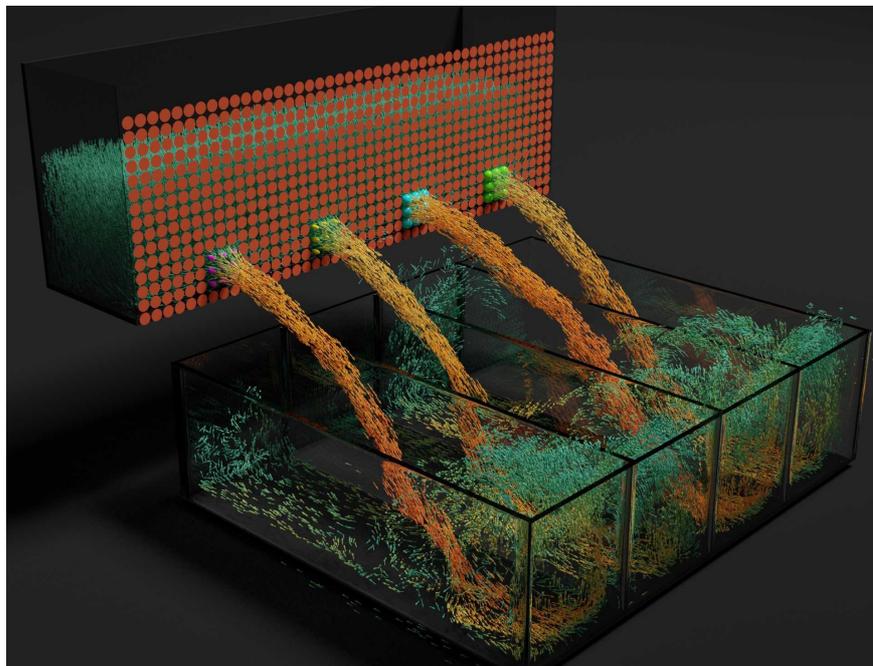


Fonte: Extraído de Parreiras et al. [2021].

Os tensores planares são úteis para mostrar como o fluido se comporta com a evolução da superfície proposta. O próximo exemplo também usa uma grade MAC de  $64 \times 64 \times 64$  com uma camada de tensores planares na posição  $k = 32$ . Como no exemplo anterior, seus autovalores são  $\lambda_1 = \lambda_2 = 1$  e  $\lambda_3 = 0, 1$ . Todos os outros tensores são a matriz identidade. Um cubo de fluido, cujo lado mede 21 células, está inicialmente centrado na posição  $(32, 32, 44)$ . Cada célula tem 20 partículas com velocidade zero. O passo de tempo fixo é  $\Delta t = 0,01$  s. A gravidade lentamente adiciona impulso ao cubo. A Figura 9 mostra os quadros-chave da simulação. Na Figura 9, o volume é reduzido pela metade em  $i = 32$ . As partículas mostradas na seção transversal estão dentro do fluido. Os quadros foram renderizadas a partir de uma visão ortográfica na direção  $x$ . Esta configuração captura o comportamento do fluido com um destaque na camada de tensores planares. Na imagem superior esquerda, o cubo está caindo e está prestes a tocar a camada planar (quadro 22). O quadro 23 na imagem central superior mostra o primeiro toque e o primeiro efeito das condições de evolução da superfície. As cores azuladas mostram uma desaceleração repentina das partículas fazendo com que a pressão aumente. Algumas

partículas reagiram à alta pressão aumentando suas velocidades (laranja). O quadro 28 no canto superior direito mostra as partículas começando a se espalhar pelo plano em direção às paredes da grade (laranja-avermelhado). As partículas no meio do fluido têm velocidades mais baixas (azul-escuro). A central esquerda mostra o fluido já subindo pelas paredes da grade (quadro 59). Até agora, a condições de evolução da superfície, a projeção anisotrópica e a advecção anisotrópica impediram que o fluido entrasse nos voxels da camada planar. No entanto, com  $\lambda_3 = 0.1$ , os tensores planares não impõem uma barreira firme. O quadro 62 mostra algumas das primeiras partículas que entraram na camada de tensores planares. Eles iniciam fluxos de fluido sugando as partículas superiores com pressões negativas, conforme mostrado na imagem central direita (quadro 75). Observe como as partículas se espalham para cima sobre a parede da grade traseira com altas velocidades (vermelho). As três imagens inferiores mostram os últimos instantes da simulação compreendendo toda a grade. A imagem inferior esquerda mostra o fluido antes de tocar a parte inferior da grade (quadro 113). Comparando com o quadro 75, percebe-se que os fluxos formavam as principais cachoeiras. Na imagem do meio (quadro 220), a maior parte do fluido passou pela camada planar. Ainda há muita turbulência. No quadro 350 à direita, o fluido na parte inferior da grade se acomoda e uma pequena quantidade de fluido ainda fica retida pela camada planar. No entanto, todas as partículas acabarão caindo pela ação da gravidade.

– Figura 11 - Uma barragem permeável formada por tensores

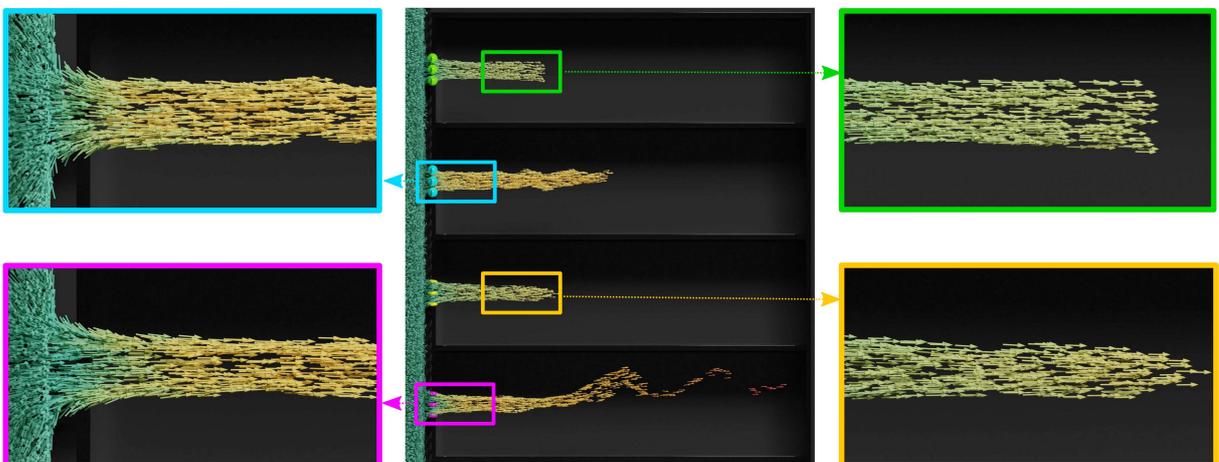


Fonte: Extraído de Parreiras et al. [2021].

### 5.1.2 Simulação com tensores lineares

A Figura 10 mostra uma barragem permeável formada por tensores com autovalores  $\lambda_1 = \lambda_2 = 1$ ,  $\lambda_3 = 10^{-5}$  e  $e_3$  apontando para a direção  $y$ . Os vetores de velocidade são representados por setas cujas magnitudes variam do menor (azul) ao maior (vermelho). Para mostrar o efeito de tensores lineares, a parede de tensores planares é perfurada em quatro lugares próximos à sua base. Os quatro furos são compostos por um tipo diferente de tensor organizado em células  $3 \times 3$ . Os glifos verdes são tensores unitários isotrópicos com  $\lambda_1 = \lambda_2 = \lambda_3 = 1$ . Um total de 126.000 partículas são simuladas com  $\Delta t = 5 \cdot 10^{-3}$  s. Os glifos ciano representam tensores isotrópicos com autovalores  $\lambda_1 = \lambda_2 = \lambda_3 = 6$  cujo efeito é produzir um ganho de velocidade nas partículas que passam sem alterar suas direções. O resultado é um jato reforçado de partículas com direções de velocidade preservadas. Os glifos amarelos representam tensores lineares com autovalores  $\lambda_1 = 1$  e  $\lambda_2 = \lambda_3 = 0.01$ , e cujo autovetor  $e_1$  aponta para a direção  $y$ . Partículas com velocidades desalinhadas com a direção principal são desaceleradas e forçadas a alinhar com  $e_1$  pela advecção anisotrópica (Equação 3.20). O resultado é um fluxo muito mais alinhado na direção  $y$ . Os glifos magenta representam tensores lineares com autovalores  $\lambda_1 = 6$  e  $\lambda_2 = \lambda_3 = 0.01$ , e cujo autovetor  $e_1$  aponta para a direção  $y$ . O efeito é semelhante aos glifos amarelos, exceto que as velocidades das partículas alinhadas a  $e_1$  são aumentadas por um fator de 6 por unidade de tempo. O resultado é um jato de partículas impulsionadas cujas velocidades estão muito alinhadas com  $e_1$ . Na Figura 10, quatro recipientes são colocados abaixo dos orifícios para coletar o fluido vazado. Os tensores impulsionados geram fluxos com velocidades mais altas (mais setas laranja). Eles drenam mais fluido como resultado. Após 175 quadros, observe que os versos de seus contêineres são atingidos pelo fluido antes dos demais.

– Figura 12 - Visão ortográfica da simulação da barragem.



Fonte: Extraído de Parreiras et al. [2021].

A Figura 11 mostra o efeito de cada categoria de tensor representado como glifos na imagem do meio. Os tensores lineares (magenta e amarelo) tendem a concentrar as partículas na direção  $e_1$ . Por outro lado, os tensores isotrópicos (ciano e verde) geram fluxos mais amplos. Tensores com ganho (ciano e magenta) produzem velocidades mais altas (mais setas laranja). Seus fluxos são mais distantes que os tensores com ganhos unitários. O tensor linear com ganho (magenta) impulsionou fortemente as primeiras partículas como mostrado pelas setas vermelhas no canto inferior direito. O fluxo seguinte é bastante turbulento até que haja partículas suficientes para estabilizar a saída. Mas seu fluxo se estabiliza após algumas iterações, conforme o retângulo magenta se destaca.

### 5.1.3 Tensores planares sob uma superfície de curvatura variável

Até agora, os experimentos são baseados em uma única camada de células coplanares. Além disso, os autovetores de todos os tensores são colineares com o sistema de coordenadas da grade. Isso significa que todas as trocas de fluidos acontecem entre os 6 vizinhos mais próximos dos voxels. Tensores com autovetores não alinhados com os eixos principais podem trocar fluido com todos os 26 vizinhos mais próximos (Equação 3.12).

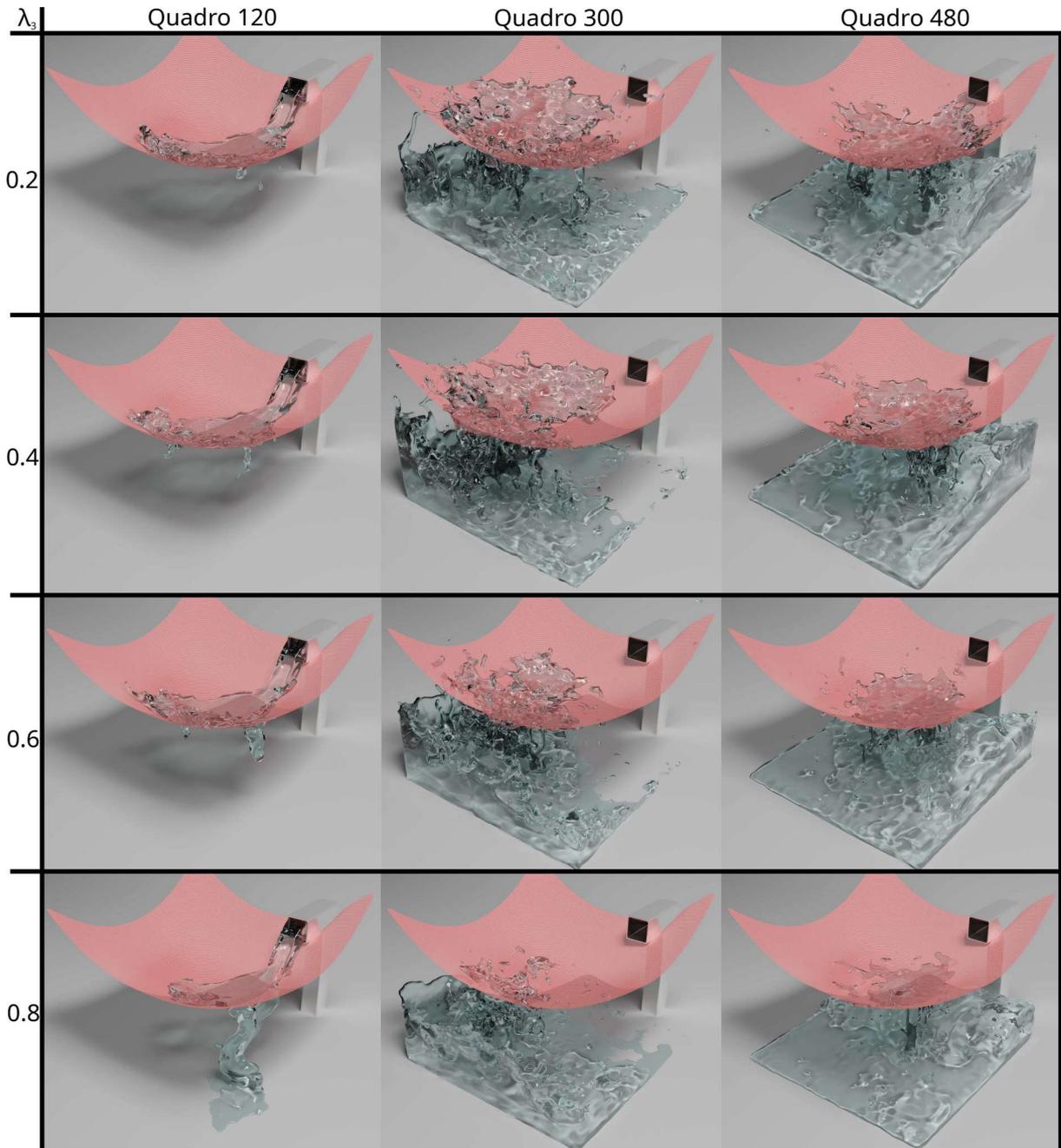
A Figura 12 exhibe uma camada de tensores planares sobre um parabolóide (superfície vermelha) com  $\lambda_1 = \lambda_2 = 1$  e  $\lambda_3 \in \{0.2, 0.4, 0.6, 0.8\}$ . O autovetor  $e_3$  é a normal do parabolóide no respectivo voxel. Alguns autovetores de tensores planares não estão alinhados com o eixo da grade principal.

Uma camada de tensores planares é usada para representar a superfície de um parabolóide cuja equação é:

$$k = k_0 + c \cdot \left[ \frac{(i - N/2)^2}{a^2} + \frac{(j - M/2)^2}{b^2} \right], \quad (5.1)$$

onde  $k \in \mathbb{Z}$  é a altura desconhecida do voxel na grade de dimensões  $N \times M \times W$ ,  $i \in \mathbb{Z}$  varia no intervalo  $[1, N]$ ,  $j \in \mathbb{Z}$  varia no intervalo  $[1, M]$ ,  $k_0 = \lfloor N/3 \rfloor$  é o mínimo do parabolóide, e os coeficientes reais são  $a = N/3$ ,  $b = M/3$  e  $c = W/10$ . O autovetor  $e_3$  é a superfície normal. Devido à curvatura variável, existem tensores com autovetores não alinhados com os eixos da grade. Os autovalores são  $\lambda_1 = \lambda_2 = 1$  e  $\lambda_3 \in \{0.2, 0.4, 0.6, 0.8\}$ . A variação de  $\lambda_3$  serve para mostrar como o fluido interage com a superfície. A figura 12 mostra a superfície do parabolóide como uma textura avermelhada representando um pano. O tamanho da grade é  $64 \times 64 \times 64$  e  $\Delta t = 10^{-3}$  s. No lado direito, uma fonte de líquido insere 250.000 partículas por segundo durante 5 segundos. Ou seja, em torno de 1.250.000 partículas são inseridas no sistema. Eles vêm de  $6 \times 6$  voxels com 8 kN de força na direção  $y$ . Esta fonte é colocada de tal forma que a corrente de fluido tende a girar sobre o parabolóide devido à condição de evolução da superfície descrita na Seção 3.2. Eventualmente, o fluido passa pelo campo tensorial formando gotículas e cachoeiras. Pode-se comparar o mesmo quadro renderizado das quatro simulações. À medida que  $\lambda_3$

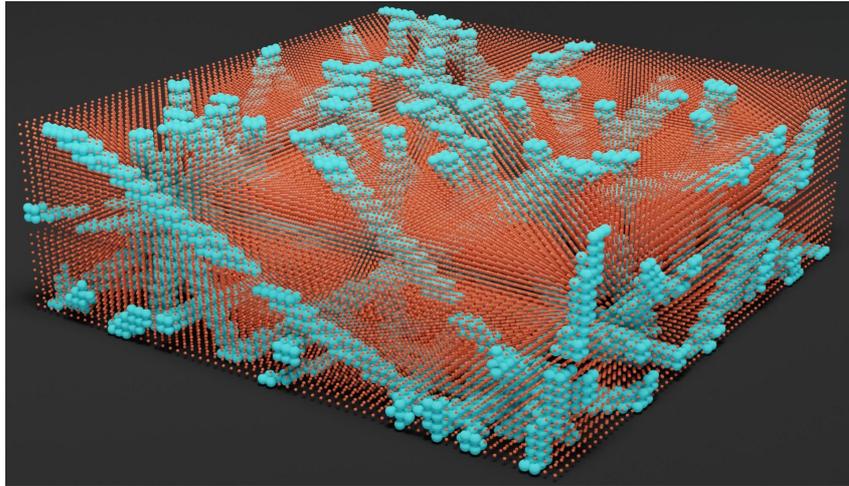
– Figura 13 - Tensores planares sob uma curvatura variável.



Fonte: Extraído de Parreiras et al. [2021].

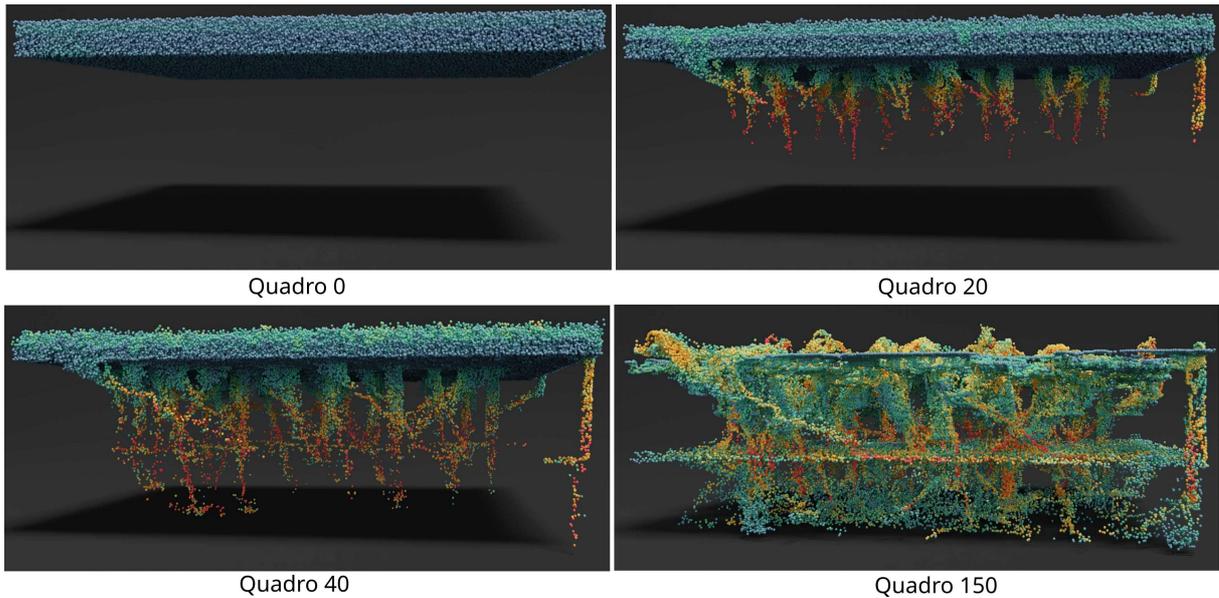
aumenta, o fluido interage como se a superfície fosse mais permeável. Com  $\lambda_3 = 0.2$ , o fluido gira por mais tempo sobre o parabolóide antes de finalmente escoar. Com  $\lambda_3 = 0.8$ , o fluido escoar mais rápido do que nos outros casos, e apenas um pequeno número de partículas pode ser visto ainda retido próximo ao mínimo parabolóide em quadro 480.

– Figura 14 - Campo tensorial isotrópico com ganho variável.



Fonte: Extraído de Parreiras et al. [2021].

– Figura 15 - Simulação em um campo tensorial isotrópico com ganho variável.



Fonte: Extraído de Parreiras et al. [2021].

#### 5.1.4 Campo tensorial isotrópico

Considere o campo tensorial na Figura 13 composto apenas de tensores isotrópicos. Neste exemplo, os tensores isotrópicos são usados para destacar estruturas colineares e coplanares. Os glifos laranja têm  $\lambda_1 = \lambda_2 = \lambda_3 = 0.1$  e os glifos ciano são  $\lambda_1 = \lambda_2 = \lambda_3 = 1$  unitários. O tamanho da grade é  $64 \times 64 \times 64$  e o campo tensorial tem 21 células de altura. Os glifos ciano formam trajetórias retas com posições e ângulos uniformemente distribuídos para o eixo  $z$  no intervalo  $[45^\circ, 60^\circ]$ . Os raios das trajetórias são 1, 2 voxels

em média. Uma camada adicional de tensores isotrópicos unitários é colocada em  $k = 10$ , que não é mostrada para maior uma visualização mais clara.

Os quadros de simulação resultantes são mostrados na Figura 14 com  $\Delta t = 0,01$  s. Inicialmente, quatro camadas de líquido são colocadas sobre o campo tensor com partículas de 20 por célula. Uma pressão de 100 N/face é imposta nas células superiores para empurrar o fluido para baixo. Tensores isotrópicos com  $\lambda_1 = \lambda_2 = \lambda_3 < 1$  diminuem o fluxo de fluido, mas não o interrompem. Como resultado, o fluido entra mais facilmente nas trajetórias retas (quadro 20). Sob a ação da pressão, o fluido vai mais fundo e atinge a camada isotrópica (quadro 40). As partículas se espalham pela camada isotrópica e continuam descendo nas trajetórias retas. Eventualmente, todas as partículas atingirão o fundo da grade, pois nenhuma barreira sólida é imposta. Observe que algumas das estruturas colineares e coplanares são visíveis no quadro 150.

## 5.2 EXPERIMENTOS COM A PROJEÇÃO BASEADA EM FACES E ARESTAS

Nos experimentos a seguir, será utilizado o método descrito no Capítulo 4, com uma escala de cores representando as magnitudes da velocidade das partículas, que variam entre azul (0 m/s) a vermelho (10 m/s). Em todos os experimentos o tamanho da grade é de  $30 \times 30 \times 30$  e o passo de tempo é  $\Delta t = 0,01$  s. No início de cada simulação, um bloco de fluido contendo 32000 partículas é adicionado centrado na posição (15, 15, 15). A grade reduzida e o baixo número de partículas se deve ao pouco tempo disponível para execução e renderização dos experimentos. Na versão final do texto esses experimentos serão refeitos com a mesma qualidade vista na Seção 5.1.

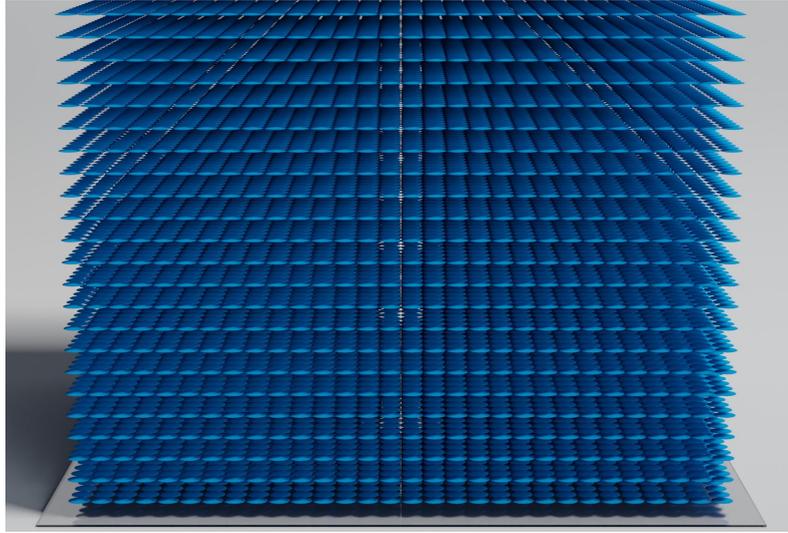
### 5.2.1 Comparação entre campo isotrópico e planar

Nesse experimento, foi adicionado um campo de forças radiais direcionadas ao centro da grade de posição (15, 15, 15), fazendo com que o fluido se aglomere ao redor do centro. Tal campo tem magnitude de  $10 \text{ m/s}^2$  em todas as posições. Para este experimento, não foram utilizados a condição de evolução da superfície (Seção 4.2) e a advecção tensorial (Seção 3.3).

A Figura 15 apresenta um campo tensorial com autovetores alinhados com os eixos, onde  $e_3$  está alinhado com o eixo  $z$  e  $\lambda_3 = 0.1$ .

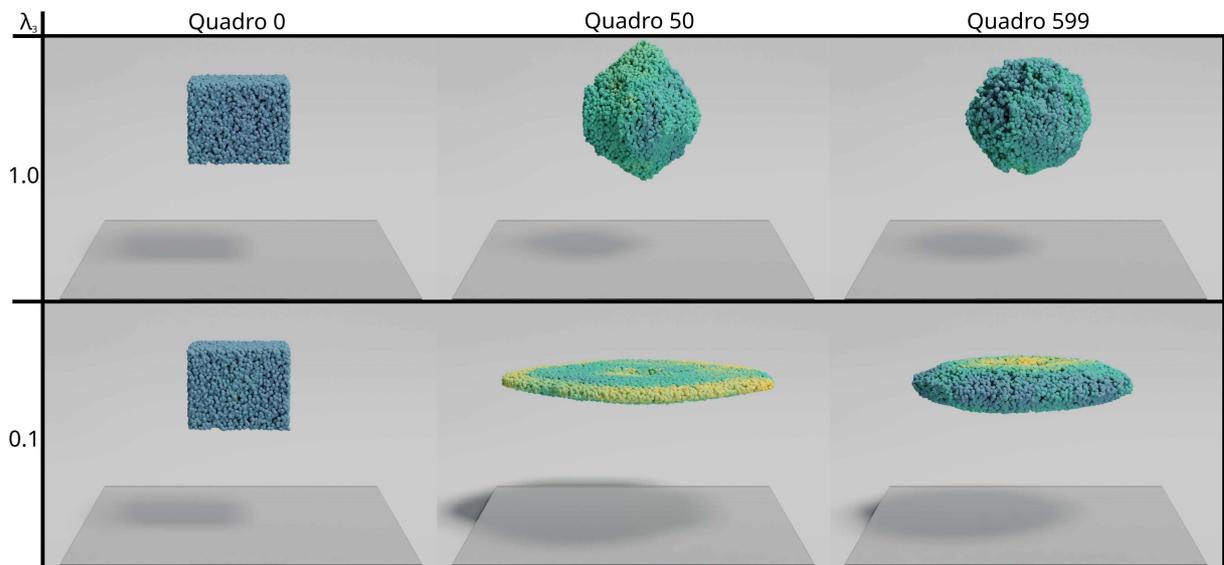
A Figura 16 mostra a simulação em um campo isotrópico unitário e um planar com  $\lambda_3 = 0.1$  (Figura 15). O objetivo é mostrar como a projeção tensorial atua na deformação do fluido. Inicialmente o bloco de fluido se localiza centrado na grade, e ao longo da simulação, o formato do fluido se desenvolve, tendendo a se tornar esférico, como visto no último quadro (Quadro 599). Perceba que para esse campo isotrópico o fluido se comporta de maneira intuitiva, tendo um comportamento próximo do que experienciamos no cotidiano. É notável a diferença entre às duas simulações. Com a adição da anisotropia,

– Figura 16 - Campo tensorial planar utilizado no experimento.



Fonte: Elaborado pelo autor.

– Figura 17 - Comparação entre um campo tensorial isotrópico e um planar sob efeito de um campo de força radial.



Fonte: Elaborado pelo autor.

a massa de líquido mantém o seu comportamento fluídico, entretanto o alívio de pressão é feito prioritariamente nas direções menos restritivas do campo. Com isso, durante a evolução da simulação, o fluido assume um formato achatado. Com o aumento de  $\lambda_3$ , a tendência é que o fluido se torne mais esférico, enquanto se diminuirmos  $\lambda_3$ , a tendência é que o fluido se torne cada vez mais próximo de um plano no domínio.

### 5.2.2 Simulação com advecção e condição de superfícies alternadas

Neste experimento será utilizado um campo tensorial linear formando circunferências centradas na posição  $(15, 15, k)$ , onde  $1 \leq k \leq 4$ . Os autovetores  $e_1, e_2$  e  $e_3$  são definidos pela seguinte equação:

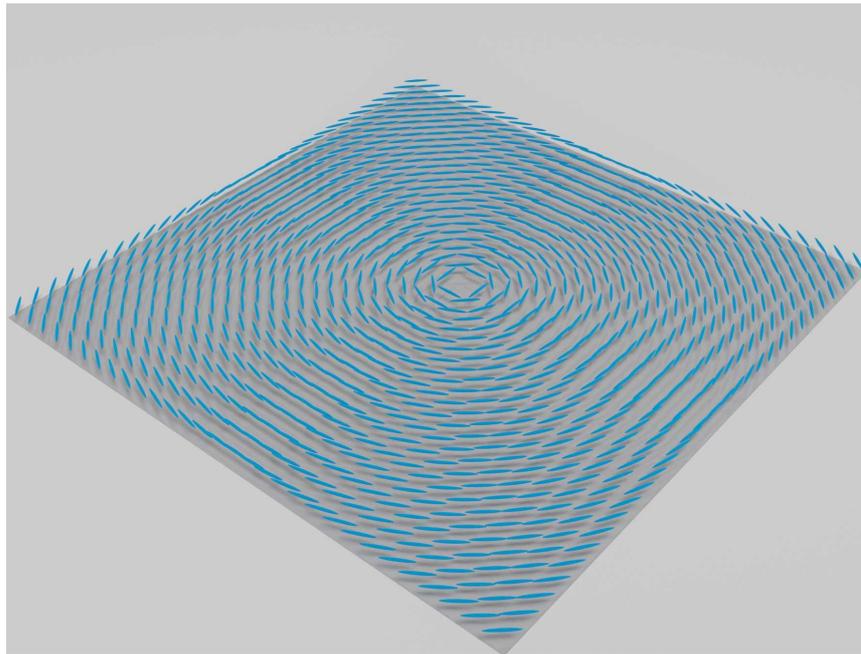
$$e_1 = \frac{e_2 \times e_3}{|e_2 \times e_3|}, \quad (5.2)$$

$$e_2 = \frac{(i - 15, j - 15, 0)}{|(i - 15, j - 15, 0)|}, \quad (5.3)$$

$$e_3 = (0, 0, 1), \quad (5.4)$$

onde  $e_1$  é o vetor tangente à circunferência na célula  $(i, j, k)$  e  $e_2$  é o vetor radial. Os autovalores são  $\lambda_1 = 1, \lambda_2 = \lambda_3 = 0, 1$ . A força de gravidade  $\mathbf{g} = [0 \ 0 \ -10]^T$  foi utilizada.

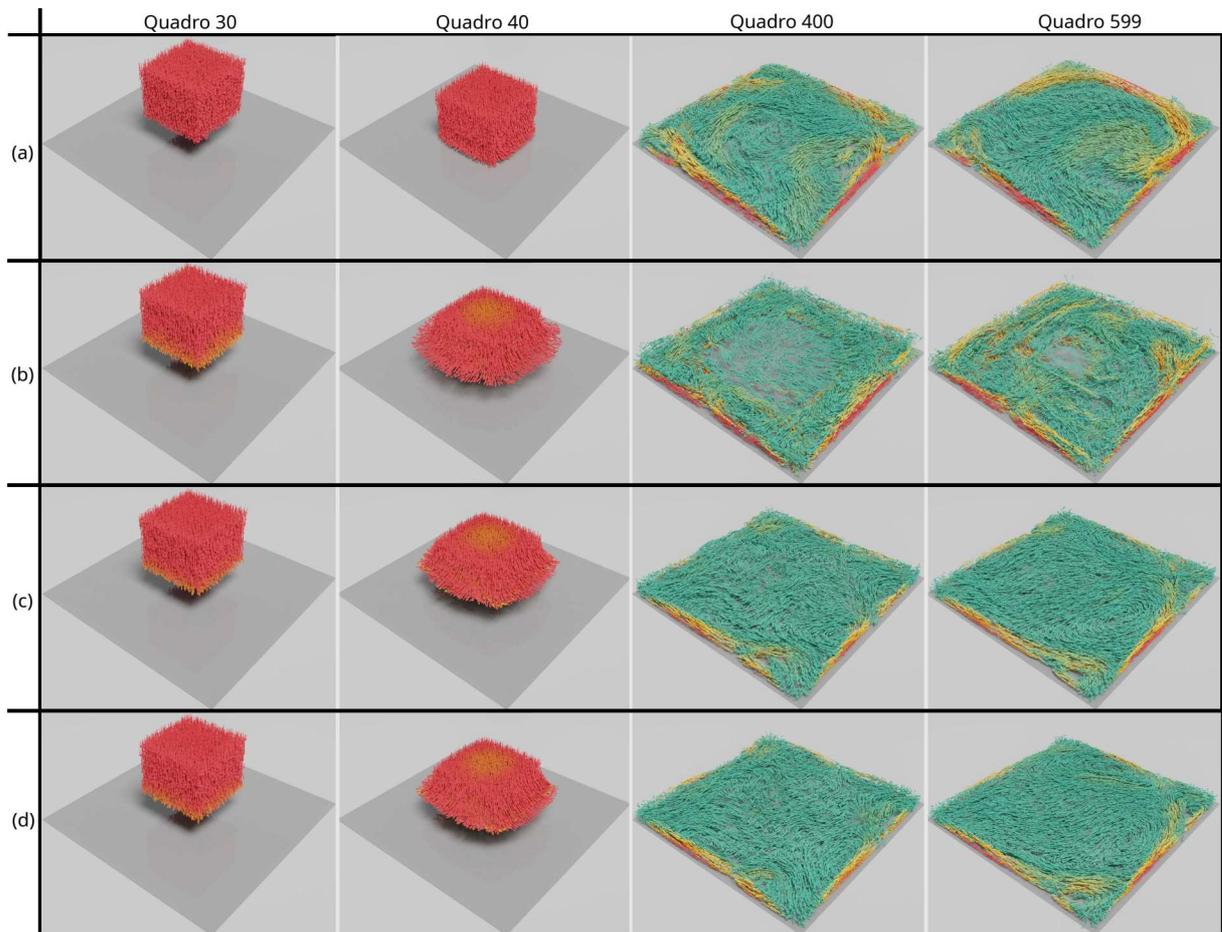
– Figura 18 - Uma lamina do campo tensorial linear circular.



Fonte: Elaborado pelo autor.

A Figura 17 mostra uma camada do campo tensorial, para esse experimento foram adicionadas 4 dessas camadas de tensores lineares no fundo do domínio. O objetivo principal é analisarmos como o método responde à adição da condição de evolução da superfície e a adição de uma advecção tensorial. Um objetivo secundário é mostrar a atuação de campos anisotrópicos atuando em conjunto com a condição de borda de Neumann para uma mesma célula. A Figura 18 está subdividida em 4 sequências de imagens, (a), (b), (c) e (d). A sequência de imagens (a) mostra o método sem a advecção tensorial e sem a condição de

– Figura 19 - Simulação com advecção e condição de superfícies alternadas.



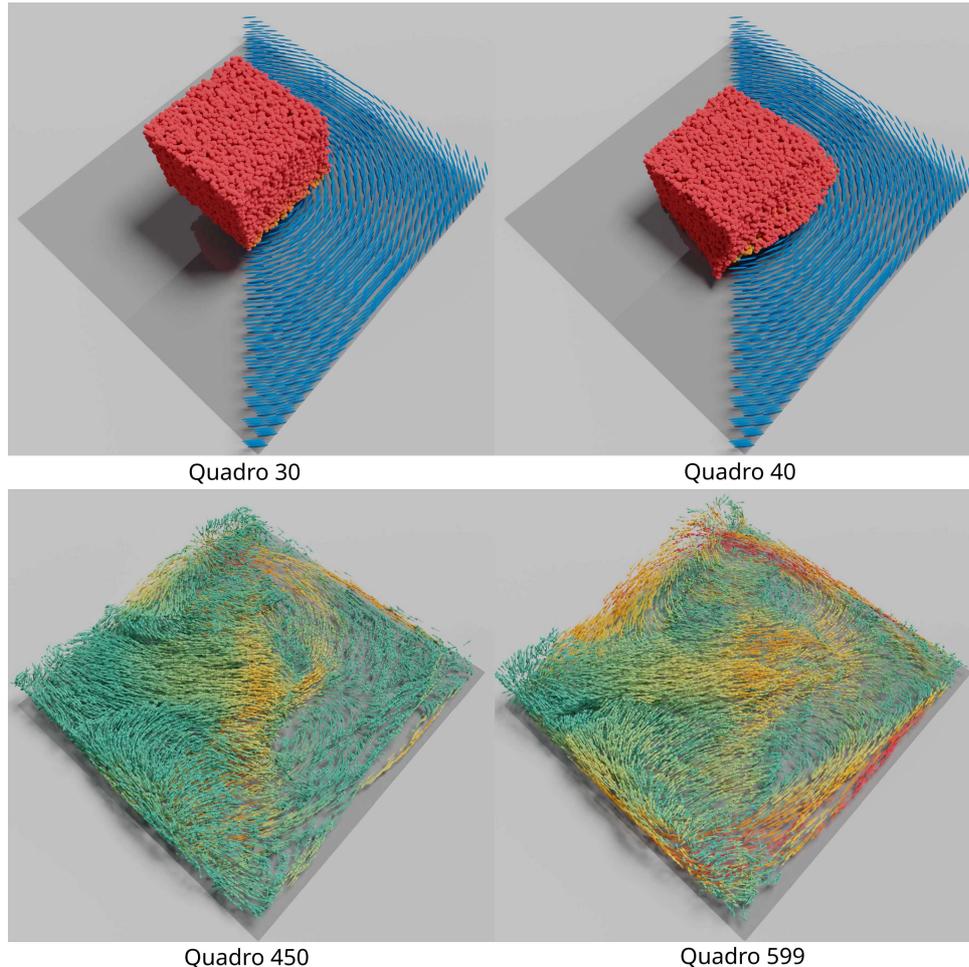
Fonte: Elaborado pelo autor.

evolução da superfície (Seções 3.3 e 4.2). Podemos observar que, o bloco de fluido ao tocar no campo tensorial no Quadro 30 não sofre nenhuma desaceleração, visto que a condição de evolução da superfície não está atuante, diferentemente das sequências (b),(c) e (d), que tem esta condição ativada e que apresentam uma variação de velocidade na parte inferior do cubo (em amarelo). A sequência de imagens (b) foi obtida utilizando o método 4, porém usando apenas os pontos pertencentes as faces da superfície do voxel (Figura 6). Note que durante a evolução da simulação apareceram alguns padrões alinhados com os eixos do sistema de coordenadas, o que não acontece nas sequências (c) e (d). A sequência (c) se mostrou mais suave e com maior geração de vórtices do que a sequência (b), isso devido à inclusão de todos os pontos das faces e arestas, e com a condição de superfície ativada. A simulação nesse caso evoluiu para gerar dois vórtices predominantes perto das bordas inferiores do cubo. Vórtices esses minimizados na sequência (d) devido à adição da advecção tensorial, que atua para alinhar os fluxos de fluido na direção menos restritiva do campo, tangente às circunferências, portanto, o fluxo de fluido tende a perder energia na direção radial.

### 5.2.3 Campo linear descontínuo

Este experimento utiliza uma formulação para o campo tensorial parecida com o experimento anterior (Equação 5.4), entretanto, foi adicionado a restrição  $1 \leq j \leq i$ , fazendo com que a região linear do campo fique na parte triangular inferior da grade. Os autovalores são  $\lambda_1 = 1.0, \lambda_2 = \lambda_3 = 0.1$ . A gravidade utilizada foi  $\mathbf{g} = [0 \ 0 \ -10]^T$ .

– Figura 20 - Simulação com variações de elementos do método.

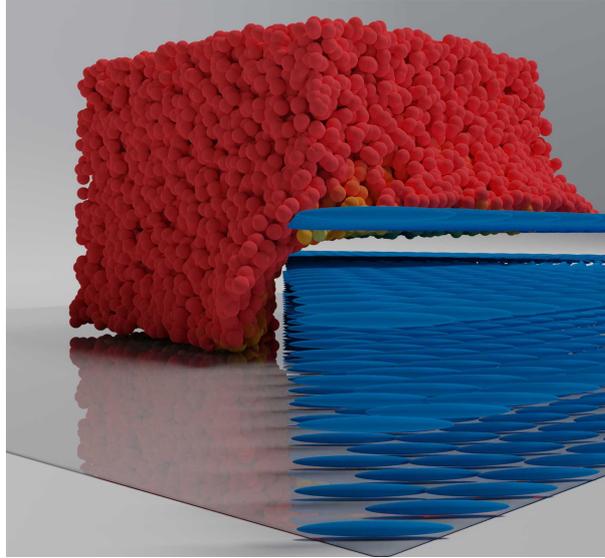


Fonte: Elaborado pelo autor.

A Figura 19 mostra como o método lida com campos descontínuos. O Quadro 30 apresenta o momento em que o fluido toca o campo linear. O Quadro 40 é quando o fluido toca o chão do domínio enquanto ainda está sob influência do campo linear. Durante a evolução da simulação a parte que está em contato desce lentamente por dentro do campo, simulando um meio permeável. Como é descrito no Capítulo 4, os tensores na face e na aresta são uma média dos tensores centrados nas células vizinhas. Como consequência, podemos constatar na Figura 20 que, na borda entre o campo linear e o isotrópico, o fluido desenvolve uma superfície arredondada, provocada por essa suavização. A Figura 20 é um render do Quadro 40 com um ângulo de câmera diferente, com foco na interação com o

campo tensorial. Durante os Quadros 450 e 599, podemos ver que o fluido no campo linear forma fluxos circulares conforme o desenho do campo. Devido à anisotropia, a pressão aumenta drasticamente a medida que um fluxo contínuo de fluido adentra pelo centro da grade, fazendo com que a projeção alivie a pressão de forma brusca pelas laterais formando correntes de fluido conformadas pelo campo.

– Figura 21 - Foco na interação fluido-tensor.



Fonte: Elaborado pelo autor.

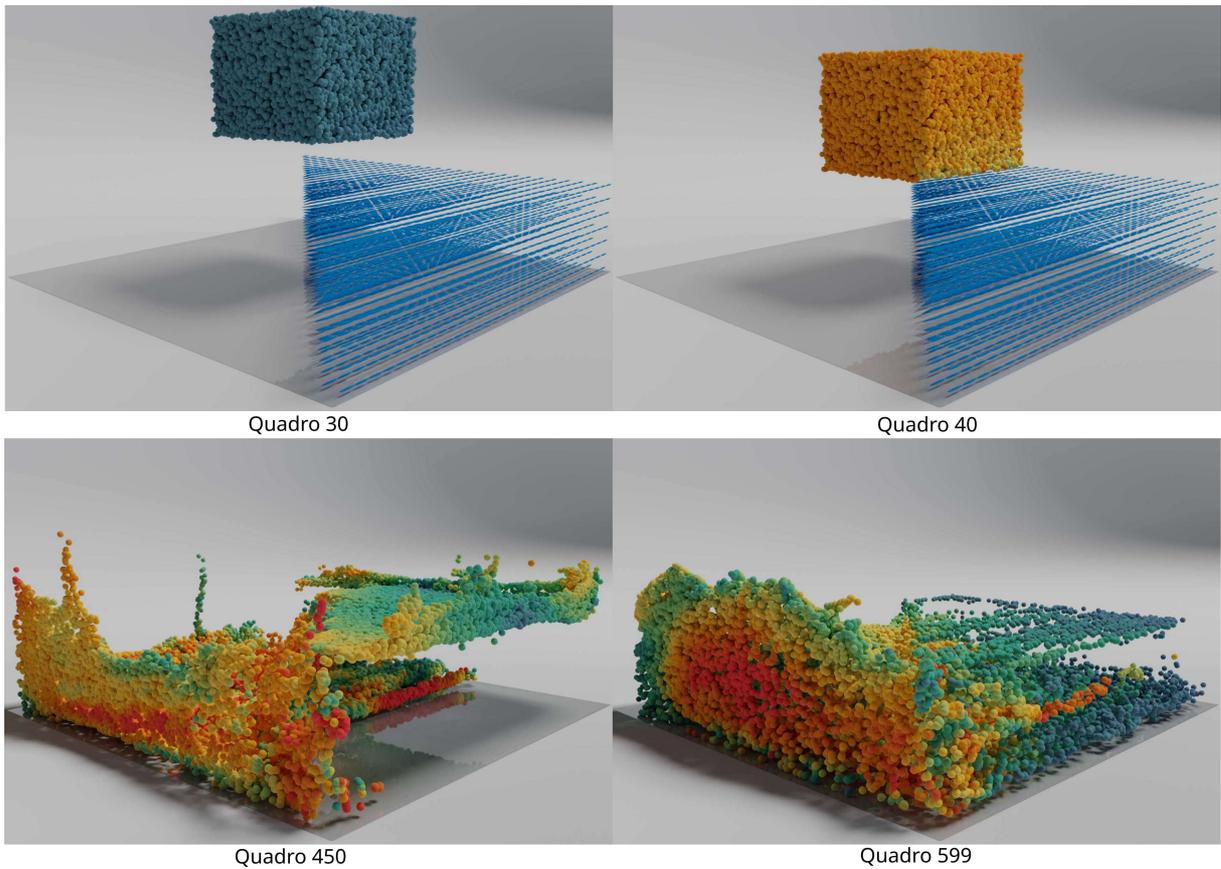
### 5.3 COMPARAÇÃO ENTRE AMBOS OS MÉTODOS

O último experimento tem o objetivo de mostrar a diferença entre os resultados obtidos por ambos os métodos apresentados nesse texto. Nesta seção estamos utilizando uma grade de tamanho  $30 \times 30 \times 30$ , com a gravidade  $\mathbf{g} = [0 \ 0 \ -10]^T$ . O campo linear contém tensores com os autovalores alinhados com os eixos, de forma que  $e_1$  esteja orientado em direção ao eixo  $y$ , e os autovalores  $\lambda_1 = 1, \lambda_2 = \lambda_3 = 0, 1$ . O campo foi construído de forma que  $j \leq i$ , formando uma descontinuidade em diagonal.

As Figuras 21 e 21 mostram a simulação feita com o método apresentado nos Capítulos 3 e 4 respectivamente. Durante a evolução da simulação, o fluido interage com o campo a partir do Quadro 40 e de forma que ele atue de um lado como um meio anisotrópico e do outro como um meio isotrópico. Quando o fluido toca no chão (Figuras 23 e 22) é possível perceber como as duas formulações para a evolução da superfície do fluido funcionam.

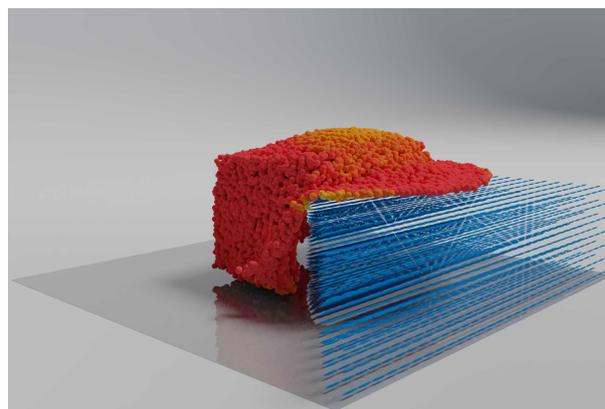
É perceptível que o método descrito na Seção 3.2 (Figura 22) conforma o fluido de maneira mais rígida do que o método com face e aresta descrita na Seção 4.2 (Figura 24). Isso se deve ao fato do método anterior utilizar o tensor vizinho diretamente no

– Figura 22 - Método baseado em face com tensor linear alinhado.



Fonte: Elaborado pelo autor.

– Figura 23 - Interação na borda fluido-vazio para o método baseado em faces.

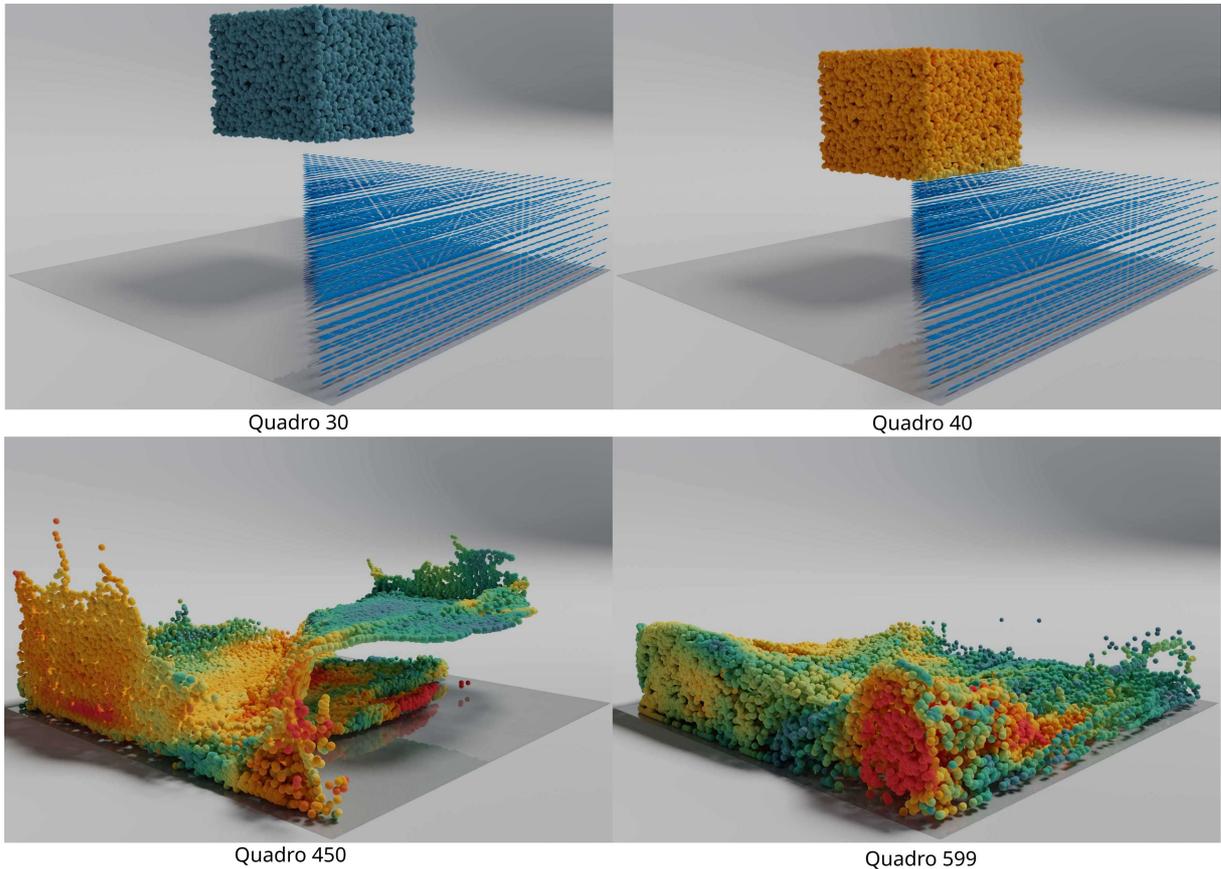


Fonte: Elaborado pelo autor.

cálculo, enquanto o método novo utiliza uma média dos tensores vizinhos para estimar o tensor nas faces e arestas, gerando uma suavização do campo tensorial. Note também que no Quadro 599 de ambos os métodos, há fluido suspenso no campo linear, entretanto o

método anterior retém muito mais fluido, isso se deve também às condições da evolução da superfície. Comparando às duas simulações, é visível que o método novo tem uma simulação mais suave em relação ao anterior, e isso se dá pela evolução do divergente durante a simulação.

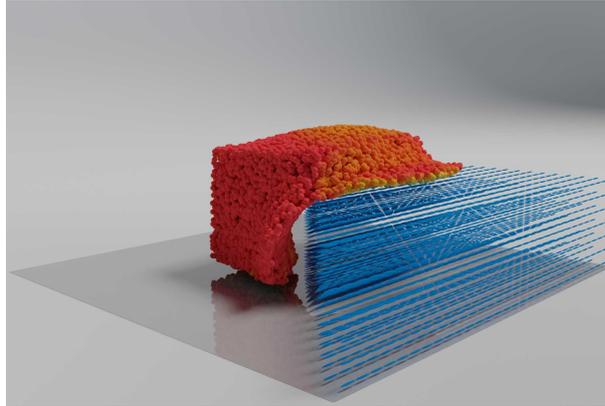
– Figura 24 - Interação na borda fluido-vazio para o método baseado em faces.



Fonte: Elaborado pelo autor.

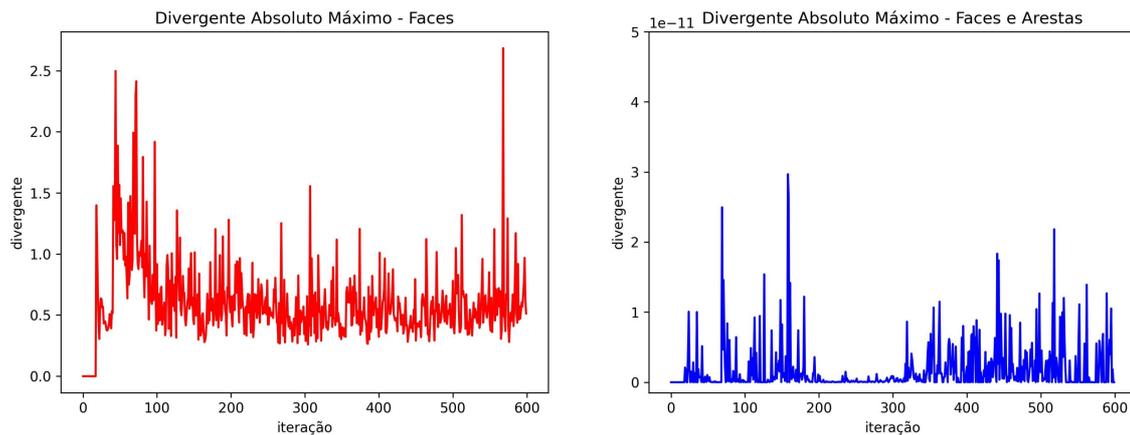
Enquanto o bloco de fluido não toca o campo linear, ambos os métodos evoluem com um divergente baixo. Entretanto, após o toque, é nítido que a evolução do divergente da velocidade no método novo (Figura 25) é de fato muito melhor do que foi obtido com método anterior, tornando o novo método consideravelmente mais eficiente para a solução do problema de  $\mathbf{T}$ -Poisson nesse caso. Um dos fatores que contribuíram para a discrepância entre os métodos foi a possibilidade de que o método novo tem de realizar o alívio de pressão pelas diagonais de forma mais eficiente. Além disso, outro fator foi o uso de operadores  $\nabla_{\mathbf{T}}p$  diferentes para o cálculo do  $\mathbf{T}$ -Laplaciano e para a atualização do campo de velocidades no método antigo, o que naturalmente dificulta o alívio de pressão corretamente. É possível notar que o fluido está consideravelmente mais volumoso no método anterior (Figura 21) no Quadro 599. Isso se deve ao divergente não nulo apresentado pelo método (Figura 25).

- Figura 25 - Interação na borda fluido-vazio para o método baseado em faces.



Fonte: Elaborado pelo autor.

- Figura 26 - Divergente absoluto máximo dos métodos. Para o cálculo foi utilizado o módulo do campo divergente da velocidade e tomado o maior entre todos.



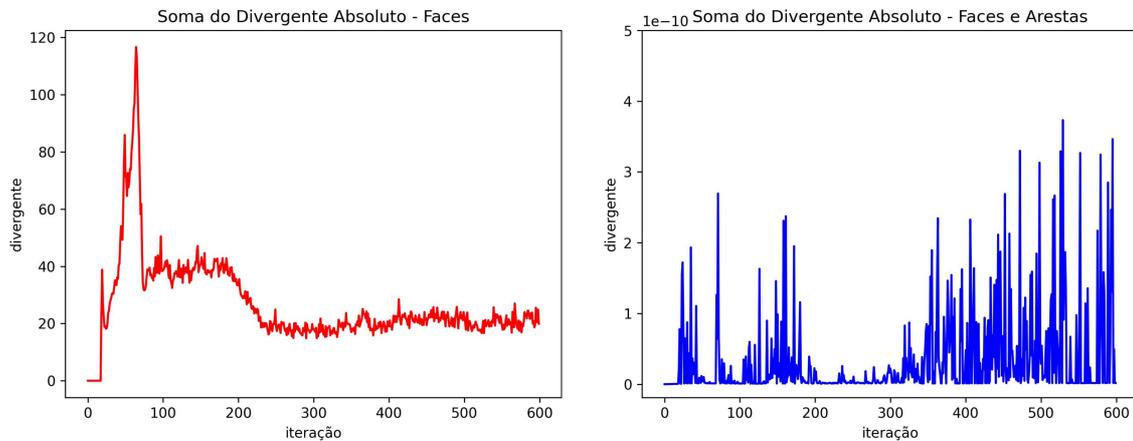
Fonte: Elaborado pelo autor.

## 5.4 DISCUSSÃO

Os principais parâmetros utilizados nos experimentos, bem como seus tempos computacionais, estão resumidos na Tabela 1. Os tempos de simulação não são proibitivos, com o solucionador do sistema linear assimétrico dominando o custo de tempo computacional.

Os métodos propostos fornecem simulações atraentes mesmo com uma grade de resolução bastante baixa e um pequeno número de partículas. Os tensores planares são interessantes para simulações de limites suaves estáticos. Eles também são úteis para simulações de superfícies permeáveis. A simulação de tecido usando um paraboloide (Figura 12) é um exemplo desse recurso. Os efeitos são convincentes mesmo com uma

- Figura 27 - Soma dos divergentes absolutos dos métodos. Para o cálculo foi utilizado o módulo do campo divergente da velocidade somado todos os valores.



Fonte: Elaborado pelo autor.

Tabela 1 – Resumo dos parâmetros e tempos computacionais dos experimentos.

Experimento	Parâmetro				Simulação	
	Grade	# partículas	Tensor	$\Delta t$ (s)	Seg./Quadro	Tempo Total
Piscina (Figura 8)	$64^3$	491,520	planar	0.03	21.43	3:34:16
Cubo (Figura 9)	$64^3$	85.220	planar	0.01	9.94	1:39:24
Barragem (Figura 10)	$45^3$	126.000	linear / planar	$5 \cdot 10^{-3}$	3.67	0:36:43
Tecido (Figura 12)	$64^3$	1.249.750	planar	$10^{-3}$	99.31	16:33:07
Isotrópico (Figura 13)	$64^3$	327.680	isotrópico	0.01	9.22	1:32:12
Projeção (Figura 16)	$30^3$	32.000	planar	0.01	0,675	0:6:45
Parâmetros (Figura 18)	$30^3$	32.000	linear	0.01	0,753	0:7:32
Descontínuo (Figura 19)	$30^3$	32.000	linear	0.01	1,108	0:11:05
Anterior (Figura 21)	$30^3$	32.000	linear	0.01	1,156	0:11:34
Novo (Figura 23)	$30^3$	32.000	linear	0.01	0,805	0:8:03

única camada de tensores planares.

Os tensores lineares podem ser usados para produzir fluxos retos de fluxo de fluido (Figura 11). No entanto, observamos que usar tensores lineares para fazer o fluido se adequar ao campo é mais difícil na formulação do Capítulo 3. Um problema nesse caso é que tensores lineares tendem a ter máscaras  $\mathbf{T}$ -Laplacianas (Equação 3.9) com poucas trocas em direções diagonais. Entretanto, o método apresentado em 4 teve um desempenho notável, gerando campos projetados com divergentes absolutos baixíssimos, mesmo com tensores lineares. Isso é possível devido a trocas em diagonal permitidas pelas arestas e que não são feitas em máscaras baseadas apenas em faces.

Com o acréscimo das arestas no cálculo do  $\mathbf{T}$ -Laplaciano, o método ficou computacionalmente mais custoso. Entretanto, a Tabela 1 nos mostra que o novo método executou

consideravelmente mais rápido do que o anterior. É plausível supor que a melhora do condicionamento da matriz de coeficientes levou a esse desempenho melhor.

Ao escolher etapas de tempo suficientemente pequenas, não observamos problemas de convergência e artefatos de fluido. A condição CFL proposta em Renhe et al. [2019] é útil como limite superior para o passo de tempo. Claro, é importante avaliar as condições específicas de convergência dos nossos métodos. Esta tarefa precisa de mais investigação e o principal desafio vem da projeção anisotrópica e advecção. De fato, as trajetórias deformadas das partículas exigem uma análise cuidadosa para encontrar uma condição CFL.

Em relação à simulação de meios porosos, as metodologias Euleriana e Lagrangiana costumam utilizar um parâmetro de permeabilidade  $\tau$  para configurar o modelo matemático. Eles são definidos a partir de formulações baseadas na lei de Darcy, como a utilizada por Ye et al. [2019]. Como essas formulações dependem do gradiente de pressão  $\nabla p$ , podemos adaptá-las para usar campos tensoriais substituindo este termo pelo  $\mathbf{T}$ -Gradiente  $\nabla_{\mathbf{T}} p$ . Como exemplo, pode-se alterar o modelo definido em Lenaerts et al. [2008] para elaborar uma formulação SPH que inclua efeitos anisotrópicos. Esta pesquisa está fora do escopo do nosso trabalho porque os métodos Lagrangianos requerem formulações muito específicas para encapsular a influência do campo tensorial. Essa possibilidade será objeto de trabalhos futuros.

O projeto de campos tensoriais é um assunto desafiador. Existem várias aplicações possíveis, como texturização sólida, animação de fluidos, simulação de meio poroso e rastreamento de partículas. Em todos os casos, o usuário precisa especificar valores de tensor e padrões locais nos domínios do fluido para obter o fluxo desejado. Em primeiro lugar, devemos notar que o problema de projeto de campo tensorial é desafiador devido aos graus de liberdade e requisitos matemáticos que um campo tensorial deve cumprir para ser adequado. Áreas como otimização de topologia Nomura et al. [2019] e ciência de materiais Irwin et al. [2019] abordam esse problema impondo restrições ao tensor para gerar relacionamentos envolvendo elementos tensoriais. Para aplicações de computação gráfica, é proposto em Palacios et al. [2017] um sistema interativo para projeto de campos tensores simétricos 3D em domínios representados por uma malha tetraédrica. O sistema também inclui um conjunto de operações para especificar a topologia do tensor e uma etapa de propagação baseada na suavização Laplaciana para calcular os valores dos tensores em locais não especificados pelo usuário durante o processo de edição. Tal abordagem poderia ser adaptada ao nosso quadro numérico. Além disso, o trabalho de Palacios et al. [2017] pode ser ampliado com desenvolvimentos recentes em subespaços de campos tensoriais tangentes Nasikun et al. [2020] para melhorar a capacidade de projetar tensores em variedades bidimensionais dentro de um volume para controle de fluidos. Trabalhos futuros incluem o estudo de projeto de campo tensorial para aplicações específicas do nosso método.

## 6 CONCLUSÃO

Este trabalho apresenta dois métodos para simulação de fluidos anisotrópicos baseados no método FLIP. Propomos uma solução para a advecção tensorial e duas soluções numéricas para a projeção tensorial. Diferente dos trabalhos anteriores na literatura, Renhe et al. [2019], este trabalho foi desenvolvido para operar em grades escalonadas do tipo MAC. Entretanto, as propostas das máscaras  $\mathbf{T}$ -Laplacianas também são úteis para abordagens Eulerianas anisotrópicas não baseadas no PIC. Propomos duas formas para controlar a evolução da superfície do fluido sob regiões anisotrópicas. Experimentos foram conduzidos com diferentes parâmetros para evidenciar a capacidade dos métodos de conformarem o fluido sob influência de um campo tensorial, além de um experimento comparativo onde observamos o comportamento do divergente da velocidade em cada método.

Como discutido no Capítulo 5, utilizar tensores lineares para o controle do fluxo de fluido pode ser desafiador. Neste problema, apresentam-se evidências de superioridade do método que utiliza face e aresta se comparado com o método proposto em Parreiras et al. [2021]. As simulações se mostraram mais estáveis e com erros de divergente nulo significativamente menores. O último experimento mostra a dificuldade do método anterior de resolver um exemplo de campo com tensores lineares, o método com faces e arestas conseguiu solucionar o problema de  $\mathbf{T}$ -Poisson com um divergente absoluto máximo na ordem de  $10^{-10}$ , contra  $10^1$  do método apresentado em Parreiras et al. [2021].

Em trabalhos futuros, podem ser explorados mecanismos para o projeto eficaz de campos tensoriais adequados para animação controlada de fluidos, bem como extensões para cenários como fluidos multifásicos ou a já mencionada simulação de meios porosos. O projeto de campos tensoriais específicos para uma dessas aplicações é de grande interesse para desenvolver cenários para comparações imparciais de escoamento de fluidos com outros métodos sob as mesmas condições iniciais e de contorno. Uma análise minuciosa deve ser realizada para determinar uma condição CFL adequada às especificidades das abordagens propostas neste trabalho. Além da possibilidade de extensão do método do Capítulo 4 para a utilização de um estêncil de 27-pontos Gupta and Kouatchou [1998]. Por fim, com essa ampliação, espera-se que haja uma melhora em casos de tensores não alinhados com os eixos, principalmente tensores lineares que impõem uma maior restrição à passagem de fluido.

## REFERÊNCIAS

- G. K. Batchelor. *An Introduction to Fluid Dynamics*, DOI=10.1017/CBO9780511800955. Cambridge Mathematical Library. Cambridge University Press, 2000.
- Joel H Ferziger, Milovan Perić, and Robert L Street. *Computational methods for fluid dynamics*, volume 3. Springer, 2002.
- Ben Frost, Alexey Stomakhin, and Hiroaki Narita. **Moana: Performing Water**. In *ACM SIGGRAPH Talks*, pages 1–2, New York, NY, USA, 2017.
- Jia-Ming Lu, Xiao-Song Chen, Xiao Yan, Chen-Feng Li, Ming Lin, and Shi-Min Hu. **A Rigging-Skinning Scheme to Control Fluid Simulation**. *Computer Graphics Forum*, 38(7):501–512, 2019.
- 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.
- Marcelo Bernardes Vieira, Gilson Antonio Giraldo, 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.
- N. Foster and D. Metaxas. **Controlling fluid animation**. In *Proceedings Computer Graphics International*, pages 178–188, 1997.
- Adrien Treuille, Antoine McNamara, Zoran Popović, and Jos Stam. **Keyframe control of smoke simulations**. In *ACM SIGGRAPH*, pages 716–723. ACM SIGGRAPH, 2003.
- Nick Foster and Ronald Fedkiw. **Practical animation of liquids**. *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pages 23–30, 2001.
- Yootai Kim, Raghu Machiraju, and David Thompson. **Path-based control of smoke simulations**. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 33–42. Eurographics Association, 2006.
- Michael B Nielsen and Brian B Christensen. **Improved variational guiding of smoke animations**. *Computer Graphics Forum*, 29(2):705–712, 2010.
- Antoine McNamara, Adrien Treuille, Zoran Popović, and Jos Stam. **Fluid control using the adjoint method**. *ACM Transactions On Graphics*, 23(3):449–456, 2004.
- Raanan Fattal and Dani Lischinski. **Target-driven smoke animation**. In *ACM SIGGRAPH*, pages 441–448. ACM SIGGRAPH, 2004.
- Lin Shi and Yizhou Yu. **Taming liquids for rapidly changing targets**. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 229–236, 2005.

Zherong Pan and Dinesh Manocha. **Efficient solver for spacetime control of smoke.** *ACM Transactions on Graphics*, 36(4):1, 2017.

Steven J. Lind, Benedict D. Rogers, and Peter K. Stansby. **Review of smoothed particle hydrodynamics: towards converged Lagrangian flow modelling.** *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 476(2241):20190801, 2020.

Gang Feng and Shiguang Liu. **Detail-preserving SPH fluid control with deformation constraints.** *Computer Animation and Virtual Worlds*, 29(1):e1781, 2018.

Xiaoyong Zhang and Shiguang Liu. **Parallel SPH fluid control with dynamic details.** *Computer Animation and Virtual Worlds*, 29(2):e1801, 2018.

N. Hageman, A. Toga, K. Narr, and D. Shattuck. **A Diffusion Tensor Imaging Tractography Algorithm Based on Navier–Stokes Fluid Mechanics.** *IEEE Transactions on Medical Imaging*, 28:348–360, 2006.

Martha W Evans, Francis H Harlow, and Eleazer Bromberg. **The particle-in-cell method for hydrodynamic calculations.** Technical report, Los Alamos National Lab., 1957.

Jeremiah U Brackbill, Douglas B Kothe, and Hans M Ruppel. **FLIP: a low-dissipation, particle-in-cell method for fluid flow.** *Computer Physics Communications*, 48(1):25–38, 1988.

Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. **The affine particle-in-cell method.** *ACM Transactions on Graphics*, 34(4):1–10, 2015.

Chuyuan Fu, Qi Guo, Theodore Gast, Chenfanfu Jiang, and Joseph Teran. **A polynomial particle-in-cell method.** *ACM Transactions on Graphics*, 36(6):1–12, 2017.

Mikhail Itskov et al. *Tensor algebra and tensor analysis for engineers.* Springer, 2007.

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.

Peter Chadwick. *Continuum mechanics: concise theory and problems.* Courier Corporation, 2012.

Fridtjov Irgens. **Continuum Mechanics in Curvilinear Coordinates.** In *Continuum Mechanics*, pages 599–624. Springer, 2008.

Chris Doran, Steven R Gullans, Anthony Lasenby, Joan Lasenby, and William Fitzgerald. *Geometric algebra for physicists.* Cambridge University Press, 2003.

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.

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

Robert Bridson. ***Fluid Simulation for Computer Graphics***. CRC Press, 2015.

Emanuel Antônio Parreiras, Marcelo Bernardes Vieira, Arthur Gonze Machado, Marcelo Caniato Renhe, and Gilson Antônio Giraldo. **A particle-in-cell method for anisotropic fluid simulation**. *Computers & Graphics*, 2021. doi: <https://doi.org/10.1016/j.cag.2021.08.010>.

G. Kindlmann. **Superquadric Tensor Glyphs**. In *Proceedings of IEEE TVCG/EG Symposium on Visualization*, pages 147–154, May 2004.

H. A. Van der Vorst. **BI-CGSTAB: A Fast and Smoothly Converging Variant of BI-CG for the Solution of Nonsymmetric Linear Systems**. *SIAM Journal on Scientific & Statistical Computing*, 13(2):631–644, 1992.

Murli M. Gupta and Jules Kouatchou. Symbolic derivation of finite difference approximations for the three-dimensional poisson equation. *Numerical Methods for Partial Differential Equations*, 14(5):593–606, 1998.

U. Ananthakrishnaiah, R. Manohar, and J. W. Stephenson. **Fourth-order finite difference methods for three-dimensional general linear elliptic problems with variable coefficients**. *Numerical Methods for Partial Differential Equations*, 3(3): 229–240, 1987.

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

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

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

Fei Ye, Junjing Duan, W. Fu, and Yuan Xingyu. **Permeability Properties of Jointed Rock with Periodic Partially Filled Fractures**. *Geofluids*, 2019(4039024):1–14, 2019.

Toon Lenaerts, Bart Adams, and Philip Dutré. **Porous Flow in Particle-Based Fluid Simulations**. *ACM Transactions on Graphics*, 27(3):1–8, 2008.

T. Nomura, A. Kawamoto, Tsuguo Kondoh, E. Dede, Jaewook Lee, Yuyang Song, and N. Kikuchi. **Inverse design of structure and fiber orientation by means of topology optimization with tensor field variables**. *Composites Part B-engineering*, 176:107187, 2019.

J. Irwin, S. Lindemann, W. Maeng, J. Wang, V. Vaithyanathan, J. M. Hu, L. Q. Chen, D. Schlom, C. Eom, and M. Rzechowski. **Magnetolectric Coupling by Piezoelectric Tensor Design**. *Scientific Reports*, 9, 2019.

Jonathan Palacios, Lawrence Roy, Prashant Kumar, Chen-Yuan Hsu, Weikai Chen, Chongyang Ma, Li-Yi Wei, and Eugene Zhang. **Tensor Field Design in Volumes**. *ACM Transactions on Graphics*, 36(6):188:1–188:15, 2017.

Ahmad Nasikun, C. Brandt, and K. Hildebrandt. **Locally supported tangential vector,  $n$ -vector, and tensor fields.** *Computer Graphics Forum*, 39, 2020.