



Modelagem de Campos Tensoriais Utilizando Splines para Controle de Fluidos

Allan Carlos Amaral Ribeiro

JUIZ DE FORA
NOVEMBRO, 2019

Modelagem de Campos Tensoriais Utilizando Splines para Controle de Fluidos

ALLAN CARLOS AMARAL RIBEIRO

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento De Ciência Da Computação
Bacharelado em Ciência da Computação

Orientador: Marcelo Bernardes Vieira

JUIZ DE FORA
NOVEMBRO, 2019

MODELAGEM DE CAMPOS TENSORIAIS UTILIZANDO SPLINES PARA CONTROLE DE FLUIDOS

Allan Carlos Amaral Ribeiro

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Marcelo Bernardes Vieira
D.Sc. em Ciência da Computação

Marcelo Caniato Renhe
Doutor em Engenharia de Sistemas e Computação

Gilson Antonio Giraldi
Doutor em Engenharia de Sistemas e Computação

JUIZ DE FORA
14 DE NOVEMBRO, 2019

A minha mãe, meu pai,
meus familiares e professores

Resumo

Métodos para animação de fluidos são geralmente difíceis de se lidar dada a natureza inerentemente complexa da mecânica de fluidos. Nessas aplicações, a simulação da dinâmica de fluidos frequentemente recai na necessidade de conter ou controlar o fluido, artificialmente, de forma que o resultado ainda seja fisicamente plausível. O presente trabalho busca definir uma forma de construir campos tensoriais a partir de splines de Catmull-Rom que sejam capazes de definir a trajetória de um fluido quando utilizado nas equações anisotrópicas de Navier-Stokes em uma simulação anisotrópica do fluidos. Para isso, é definido um modelo sintético para simulação do escoamento de fluido que pode ser utilizado para complementar métodos já existentes e servir como base para trabalhos futuros.

Palavras-chave: Campos Tensoriais, Spline de Catmull-Rom, Simulação de Fluidos.

Abstract

Methods for fluid animation are challenging to deal with provided the inherent complexity of fluid mechanics. In such applications, the fluid dynamics often relies on the need to contain or control the fluid artificially so that the result is still physically plausible. This work aims to define a method to construct tensor fields from Catmull-Rom splines that are able to define the fluid trajectory when used in the anisotropic Navier-Stokes equations in an anisotropic fluid simulation. For this purpose, a synthetic model is defined for simulations of fluid flow over a trajectory that can be used as complement for previous works and serve as base for future methods.

Keywords: Tensor Field, Catmull-Rom Spline, Fluid Simulation.

Conteúdo

Lista de Figuras	5
Lista de Tabelas	7
Lista de Abreviações	8
1 Introdução	9
1.1 Problema	10
1.2 Justificativa	10
1.3 Objetivos	10
2 Trabalhos Relacionados	12
3 Fundamentação Teórica	14
3.1 Funções Polinomiais e Splines	14
3.1.1 Splines de Catmull-Rom	15
3.2 Tensores	17
3.2.1 Tensores de Orientação	18
3.3 Dinâmica de Fluidos	19
3.3.1 Equações de Navier-Stokes	20
3.3.2 Pontos de vista Lagrangeano e Euleriano	20
3.4 Simulação Anisotrópica de Fluidos	21
4 Método Proposto	22
4.1 Processamento da curva	22
4.1.1 Pré-processamento dos Pontos de Controle	23
4.1.2 Cálculo da Tangente	24
4.2 Particionamento do Domínio	25
4.2.1 Cálculo da Orientação	27
4.3 Criação do Campo Tensorial	28
5 Resultados	30
5.1 Curvas com Pontos de Controle Colineares	30
5.1.1 Campo Tensorial com Casca Planar	32
5.2 Curvas com Pontos de Controle Coplanares Sem Inflexão	33
5.2.1 Campo Tensorial com Casca Planar	35
5.3 Curva com Pontos de Controle não Coplanares	36
5.3.1 Campo Tensorial com Casca Planar	37
5.3.2 Campo Tensorial com Casca Planar e Região Interna Linear	39
5.3.3 Campo Tensorial com Casca Planar, Interior Linear e Exterior Dissipativo	40
6 Conclusão	43
Referências Bibliográficas	45

Lista de Figuras

3.1	Spline de Catmull-Rom utilizando os pontos $\mathbf{p}_{-2}, \mathbf{p}_{-1}, \mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$	15
3.2	Segmento gerado pelos pontos $\mathbf{p}_{-1}, \mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ em vermelho (a) e representação geométrica das restrições $\mathbf{f}(0), \mathbf{f}(1), \mathbf{f}'(0), \mathbf{f}'(1)$ (b)	16
3.3	Representação geométrica dos tensores de orientação variando os autovalores. (Renhe, 2017)	19
4.1	Exemplo de spline gerada utilizando apenas os pontos inseridos pelo usuário (a) e spline incluindo os pontos \mathbf{p}_{-1} e \mathbf{p}_{n+1} (b).	23
4.2	Segmento de spline formado pelo conjunto de pontos $\{\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\}$. Nela são mostrados os pontos rasterizados \mathbf{p}_l e suas respectivas tangentes \mathbf{t}_l (vermelho).	25
4.3	Visualização do processo de divisão do domínio em 2D	26
4.4	Exemplo de divisão de domínio em três regiões: Interna (Amarelo), Casca (Roxo) e Externo (Azul).	27
4.5	Exemplo de spline (a) e a representação geométrica do campo tensorial gerado (b) utilizando os conjuntos de autovalores $\mathbf{\Lambda}_I = \{1, 1, 1\}$, $\mathbf{\Lambda}_C = \{1, 0.05, 1\}$ e $\mathbf{\Lambda}_E = \{1, 1, 1\}$	29
5.1	Spline de uma reta horizontal definida por 2 pontos.	31
5.2	Frames 36, 78, 120, 162, 204, 246, 288 e 330 da simulação da curva com pontos de controle colineares utilizando campo tensorial isotrópico.	32
5.3	Campo tensorial gerado para curva com dois pontos colineares utilizando os conjuntos de autovalores $\mathbf{\Lambda}_I = \{1, 1, 1\}$, $\mathbf{\Lambda}_C = \{1, 0.05, 1\}$ e $\mathbf{\Lambda}_E = \{1, 1, 1\}$. Os tensores isotrópicos da região Externa foram omitidos pra facilitar a visualização.	32
5.4	Frames 36, 78, 120, 162, 204, 246, 288 e 330 da simulação da curva com pontos de controle colineares utilizando tensores planares na região da Casca.	33
5.5	Spline gerada utilizando 7 pontos de controle coplanares e sem inflexão.	34
5.6	Frames 36, 78, 120, 162, 204, 246, 288 e 330 da simulação da curva planar sem inflexão utilizando campo tensorial isotrópico.	34
5.7	Campo tensorial gerado para curva com sete pontos coplanares utilizando os conjuntos de autovalores $\mathbf{\Lambda}_I = \{1, 1, 1\}$, $\mathbf{\Lambda}_C = \{1, 0.05, 1\}$ e $\mathbf{\Lambda}_E = \{1, 1, 1\}$. Os tensores isotrópicos da região Externa foram omitidos pra facilitar a visualização.	35
5.8	Frames 36, 78, 120, 162, 204, 246, 288 e 330 da simulação da curva planar sem inflexão utilizando campo tensorial com planares na região da Casca.	35
5.9	Spline gerada a partir dos pontos definidos pela Equação (5.2)	36
5.10	Campo tensorial gerado para curva com treze pontos não coplanares utilizando os conjuntos de autovalores $\mathbf{\Lambda}_I = \{1, 1, 1\}$, $\mathbf{\Lambda}_C = \{1, 0.05, 1\}$ e $\mathbf{\Lambda}_E = \{1, 1, 1\}$. Os tensores isotrópicos foram omitidos pra facilitar a visualização.	37
5.11	Frames 25, 70, 115, 160, 205, 250, 295, 340, 385, 430, 475, 520, 565, 610, 655 e 700 da simulação da hélice utilizando tensores planares na região da Casca.	38

5.12	Campo tensorial gerado para curva com treze pontos não coplanares utilizando os conjuntos de autovalores $\mathbf{\Lambda}_I = \{1.2, 0.5, 0.5\}$, $\mathbf{\Lambda}_C = \{1, 0.05, 1\}$ e $\mathbf{\Lambda}_E = \{1, 1, 1\}$. Os tensores isotrópicos foram omitidos pra facilitar a visualização.	39
5.13	Frames 25, 70, 115, 160, 205, 250, 295, 340, 385, 430, 475, 520, 565, 610, 655 e 700 da simulação da hélice utilizando tensores planares na região da Casca.	40
5.14	Campo tensorial gerado para curva com treze pontos não coplanares utilizando os conjuntos de autovalores $\mathbf{\Lambda}_I = \{0.5, 1.2, 0.5\}$, $\mathbf{\Lambda}_C = \{0.5, 0.05, 0.5\}$ e $\mathbf{\Lambda}_E = \{0.05, 0.05, 0.05\}$. Os tensores isotrópicos foram omitidos pra facilitar a visualização.	41
5.15	Frames 25, 70, 115, 160, 205, 250, 295, 340, 385, 430, 475, 520, 565, 610, 655 e 700 da simulação da hélice utilizando tensores lineares na região Interna, planares na região da Casca, e isotrópicos com autovalores menores que 1 na região Externa.	42

Lista de Tabelas

5.1	Pontos de controle definindo uma reta horizontal.	30
5.2	Conjunto de pontos de controle definindo uma reta planar sem inflexão. . .	33
5.3	Conjunto de pontos de controle definindo uma curva não planar.	36

Lista de Abreviações

DCC	Departamento de Ciência da Computação
UFJF	Universidade Federal de Juiz de Fora
NURBS	<i>Non Uniform Rational Basis Spline</i>

1 Introdução

Métodos para animação de fluidos são geralmente difíceis de se lidar dada a natureza inerentemente complexa da mecânica de fluidos. Mesmo relaxando propriedades físicas, atingir resultados plausíveis se prova uma tarefa árdua. Nessas aplicações, a simulação da dinâmica de fluidos frequentemente recai na necessidade de conter ou controlar o fluido, artificialmente, de forma que o resultado ainda seja fisicamente plausível. Assim o fluido deve se comportar naturalmente, porém seu movimento ainda deve ser restringido de forma que consiga atender as necessidades da animação. Desta forma o animador pode definir um caminho pelo qual o fluido deva seguir ou definir uma região para concentrá-lo, produzindo por exemplo formas de objetos utilizando o fluido.

O presente trabalho busca definir uma forma de construir campos tensoriais a partir de splines que sejam capazes de definir a trajetória de um fluido quando utilizado nas equações anisotrópicas de Navier-Stokes em uma simulação anisotrópica de fluidos. Desta forma, o principal foco deste trabalho é a definição de um método sintético para a construção de campos tensoriais que serão utilizados para a simulação do escoamento do fluido por uma trajetória.

De forma alguma, este trabalho tem como objetivo fazer um controlador de fluidos ideal a partir do campo tensorial, mas propor um modelo que possa complementar métodos já existentes e servir como base para trabalhos futuros.

Levando em consideração a dificuldade de se controlar a simulação com campos tensoriais, os principais desafios a serem superados são o uso de um modelo de simulação anisotrópica de fluido definida para o contínuo, que deve ser discretizada; e utilizar um único campo tensorial que controle a trajetória do fluido sem que sejam definidas novas condições de contorno.

1.1 Problema

O uso de campos tensoriais para controle de fluidos na animação é relativamente recente. Trabalhos como o de Vieira (2020) aprimoram a simulação de fluidos para campos anisotrópicos e permitem que campos tensoriais possam ser usados em simulações de modo mais efetivo. No entanto, métodos para criação dos campos tensoriais para tais simulações têm sido pouco estudados.

Ao trabalhar com campos tensoriais, percebe-se grande dificuldade no controle do fluido, o comportamento do fluido para um dado campo tensorial não é trivial, o que torna difícil prever o resultado final da simulação.

Desta forma, o problema é definir um método para construção de campos tensoriais que sejam capazes de fazer com que o fluido a escoe por uma trajetória que passe por todos os pontos definidos.

1.2 Justificativa

Embora métodos para criação de campos tensoriais não seja um tema novo, trabalhos como o de Zhang; Heys; Turk (2006) e Palacios; Zhang (2007) foram desenvolvidos para ser utilizados em conjunto com técnicas de renderização não realista Hays; Essa (2004), não sendo a melhor opção para simulação de fluidos.

Desta forma, o presente trabalho justifica-se por explorar um campo de estudo relativamente novo e em expansão, com a criação de campos tensoriais específicos para simulação de fluidos. Assim o trabalho contribui para que o uso de simulações de fluido anisotrópicos utilizando campos tensoriais possa complementar o sistema utilizado atualmente na indústria cinematográfica que se baseia exclusivamente na modelagem de campos de velocidades.

1.3 Objetivos

Usando como base o método proposto por Vieira (2020) para simulação de fluidos utilizando a projeção anisotrópica, o objetivo geral deste trabalho é investigar e definir um

método para construção de campos tensoriais que permitam que o fluido escoe por uma curva que passe por todos os pontos definidos. Para alcançar tal meta, alguns objetivos específicos são importantes:

- Desenvolver um método para criação de campos tensoriais a partir da representação da curva utilizando splines de Catmull-Rom.
- Variar os parâmetros do método proposto e observar os resultados correspondentes de forma qualitativa para cada uma das simulações.

2 Trabalhos Relacionados

Dentro da área de computação gráfica, tem-se grande interesse em controlar a simulação de fluidos e, por isso, vários trabalhos já foram publicados sobre o assunto. O trabalho de Treuille et al (2003) foi um dos primeiros a abordar o problema, o autor utiliza uma serie de *keyframes* definidos pelo usuário que estipulam o comportamento do campo de velocidade e a densidade no instante definido. Para cada iteração da simulação, o algoritmo calcula o quão próximo a simulação está do estado estipulado pelo *keyframe* e utiliza um processo de otimização para atualizar a velocidade e aproximar a simulação ao resultado esperado pelo animador.

Técnicas de *keyframes* também foram empregadas por Browning et al (2014) ao produzir animações de fluido a partir de desenhos feitos a mão. A partir dos desenhos definidos, o método utiliza *morphs* regenerativos para interpolar os quadros intermediários, sendo capaz de reproduzir o estilo artístico dos quadros definidos como entrada.

Enquanto os trabalhos anteriores focam no resultado final, outros trabalhos se preocupam em manter o controle do fluido durante toda a simulação. O trabalho de Kim et al (2006) utiliza uma curva NURBS definida pelo usuário para gerar um campo vetorial procedural que é utilizado para aproximar o campo de velocidades do fluido, de modo que ele convirja para o caminho determinado.

Há também métodos interativos como o de Manteaux (2016) que permitem que o usuário crie novas simulações editando e combinando outras já existentes, independente dos métodos utilizados para gerá-las. Para isso, o trabalho utiliza métodos de classificação e reconstrução topológica.

Outros métodos como o de Sato et al (2014), permitem que uma simulação possa ser interativamente transformada utilizando um processo de deformação do domínio. Neste trabalho, um campo escalar é gerado a partir da função de fluxo do campo de velocidade do fluido e pode ser então deformado pelo usuário. A simulação final, respeitando a deformação, é obtida utilizando o rotacional da função de fluxo.

Uma outra abordagem para controlar o escoamento do fluido durante toda a

simulação foi definida no trabalho de Renhe (2017), que propõe a utilização de campos tensoriais de orientação como parte do meio, de modo que, o tensor é capaz de defletir o fluido transformando o campo de velocidade. O trabalho de Vieira (2020) aprofunda a técnica ao introduzir um método de projeção anisotrópica utilizando o campo tensorial.

O presente trabalho optou por utilizar as equações modificadas de Navier-Stokes, definidas por Vieira (2020) como base para simulação, de forma que o campo tensorial gerado pelo método faça parte do meio e seja responsável por controlar o escoamento do fluido.

3 Fundamentação Teórica

Este trabalho se baseia em vários conceitos e a compreensão destes conceitos é importante para o entendimento do método proposto. Desta forma, nesse capítulo são apresentados os seguintes conceitos: Splines de Catmull-Rom, utilizadas para geração das curvas; Tensores de Orientação; Equações de Navier-Stokes e Simulação Anisotrópica de Fluido.

3.1 Funções Polinomiais e Splines

Uma função polinomial de n -ésimo grau em s é definida como:

$$\begin{aligned} y(s) &= \sum_{k=0}^n c_k s^k \\ &= c_0 + c_1 s + \cdots + c_{n-1} s^{n-1} + c_n s^n, \end{aligned} \quad (3.1)$$

onde n é um inteiro não negativo e c_k são constantes, com $c_n \neq 0$. De forma que quando $n = 3$ tem-se um polinômio cúbico, com $n = 2$, um polinômio quadrático e quando $n = 1$, tem-se uma simples reta.

Polinômios podem ser utilizados para definir caminhos e formas, nestes casos, um conjunto de pontos são definidos e um ajuste de curva é realizando, construindo curvas polinomiais que entre cada par de pontos. Cada segmento de curva pode ser definida de forma paramétrica como:

$$\begin{cases} x &= c_{x_0} + c_{x_1} s + \cdots + c_{x_{n-1}} s^{n-1} + c_{x_n} s^n, \\ y &= c_{y_0} + c_{y_1} s + \cdots + c_{y_{n-1}} s^{n-1} + c_{y_n} s^n, \\ z &= c_{z_0} + c_{z_1} s + \cdots + c_{z_{n-1}} s^{n-1} + c_{z_n} s^n, \end{cases} \quad (3.2)$$

ou na forma vetorial:

$$\mathbf{f}(s) = \mathbf{c}_0 + \mathbf{c}_1 s + \cdots + \mathbf{c}_{n-1} s^{n-1} + \mathbf{c}_n s^n, \quad (3.3)$$

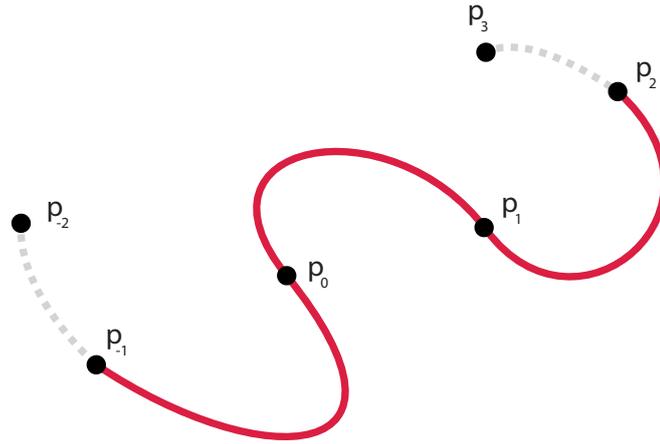


Figura 3.1: Spline de Catmull-Rom utilizando os pontos \mathbf{p}_{-2} , \mathbf{p}_{-1} , \mathbf{p}_0 , \mathbf{p}_1 , \mathbf{p}_2 , \mathbf{p}_3 .

onde o parâmetro s varia no intervalo $[0, 1]$. Os valores para os coeficientes de s são definidos nas condições de borda entre dois segmentos adjacentes. Uma das condições é de que as curvas tenham as mesmas coordenadas na região de fronteira e a segunda é que a derivada de ambas as curvas sejam iguais na região de fronteira, de forma que uma curva contínua e diferenciável seja obtida. Tais curvas polinomiais definidas por partes são chamadas de Splines (Hearn, 1996).

3.1.1 Splines de Catmull-Rom

Neste trabalho, tem-se interesse, em espacial, pelas splines de Catmull-Rom. Descritas pela primeira vez por Catmull; Rom (1974), são uma classe de splines cúbicas interpolativas locais que combinam interpolação Lagrangeana e B-splines (Curvas de Bézier). A Figura (3.1) mostra um exemplo de curva gerada através do método de Catmull-Rom utilizando 6 pontos de controle.

Considere um segmento da curva polinomial cúbica parametrizada pela variável s que varia no intervalo $[0, 1]$, definida da seguinte forma:

$$\mathbf{f}(s) = \mathbf{c}_0 + \mathbf{c}_1s + \mathbf{c}_2s^2 + \mathbf{c}_3s^3, \quad (3.4)$$

onde $\mathbf{f}(s) \in \mathbb{R}^3$ e os coeficientes de s , \mathbf{c}_0 , \mathbf{c}_1 , \mathbf{c}_2 e \mathbf{c}_3 , são pontos obtidos através do processo de ajuste da curva ao conjunto de pontos de controle $\{\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{n-1}\}$, com

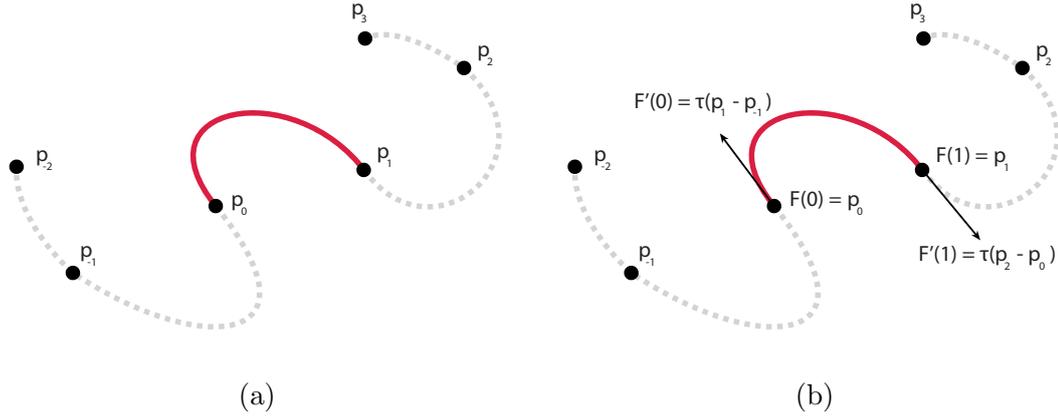


Figura 3.2: Segmento gerado pelos pontos \mathbf{p}_{-1} , \mathbf{p}_0 , \mathbf{p}_1 , \mathbf{p}_2 em vermelho (a) e representação geométrica das restrições $\mathbf{f}(0)$, $\mathbf{f}(1)$, $\mathbf{f}'(0)$, $\mathbf{f}'(1)$ (b)

$n \geq 4$, onde n é o número de pontos de controle.

Dessa forma, a spline de Catmull-Rom definida por um conjunto de $n - 2$ segmentos de curva polinomial cubica, onde cada segmento i é caracterizada pelas seguintes restrições nas condições de fronteira entre segmentos de curva adjacentes (Twigg, 2003):

$$\begin{cases} \mathbf{f}(0) &= \mathbf{p}_i, \\ \mathbf{f}(1) &= \mathbf{p}_{i+1}, \\ \mathbf{f}'(0) &= \tau(\mathbf{p}_{i+1} - \mathbf{p}_{i-1}), \\ \mathbf{f}'(1) &= \tau(\mathbf{p}_{i+2} - \mathbf{p}_i), \end{cases} \quad (3.5)$$

onde τ é o fator de "tensão". A Figura (3.2) mostra a representação geométrica do segmento $i = 0$ e as restrições $\mathbf{f}(0)$, $\mathbf{f}(1)$, $\mathbf{f}'(0)$ e $\mathbf{f}'(1)$.

Ao combinar as Equações (3.4) e (3.5) é possível obter os seguintes valores para \mathbf{c}_0 , \mathbf{c}_1 , \mathbf{c}_2 e \mathbf{c}_3 :

$$\begin{cases} \mathbf{c}_0 &= \mathbf{p}_i, \\ \mathbf{c}_1 &= -\tau\mathbf{p}_{i-1} + \tau\mathbf{p}_{i+1}, \\ \mathbf{c}_2 &= (2\tau)\mathbf{p}_{i-1} + (\tau - 3)\mathbf{p}_i + (3 - 2\tau)\mathbf{p}_{i+1} - \tau\mathbf{p}_{i+2}, \\ \mathbf{c}_3 &= -\tau\mathbf{p}_{i-1} + (2 - \tau)\mathbf{p}_i + (\tau - 2)\mathbf{p}_{i+1} + \mathbf{p}_{i+2}. \end{cases} \quad (3.6)$$

Desta forma, é possível representar $\mathbf{f}(s)$ como um produto de matriz:

$$\mathbf{f}(s) = \mathbf{s}^T \mathbf{M} \mathbf{p} = \begin{bmatrix} 1 & s & s^2 & s^3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\tau & 0 & \tau & 0 \\ 2\tau & \tau - 3 & 3 - 2\tau & -\tau \\ -\tau & 2 - \tau & \tau - 2 & \tau \end{bmatrix} \begin{bmatrix} \mathbf{p}_{i-1} \\ \mathbf{p}_i \\ \mathbf{p}_{i+1} \\ \mathbf{p}_{i+2} \end{bmatrix}, \quad (3.7)$$

onde o produto $\mathbf{M} \mathbf{p}$ é chamado matriz geométrica e é constante para todo o segmento de curva formado por \mathbf{p}_{i-1} , \mathbf{p}_i , \mathbf{p}_{i+1} , \mathbf{p}_{i+2} .

3.2 Tensores

Tensores são utilizados classicamente na física para descrever grandezas como tensão superficial e permeabilidade magnética. Nesses problemas é necessário descrever variação de quantidades em mais de uma direção. Valores escalares e vetoriais simplesmente não possuem informação suficiente, deste modo, recorre-se ao uso de tensores.

Segundo Kolecki (2002), tensores são entidades matemáticas que pode ser classificada de acordo com sua ordem n , indicando o número de componentes e as propriedades que este representa. Além disso, o número de componentes também depende do espaço euclidiano no qual ele está inserido. Assim, o tensor possui k^n componentes, onde k é a dimensão do espaço. Como esse trabalho está restrito ao \mathbb{R}^3 , portanto, o número de componentes de um tensor é 3^n . Dessa forma, tensores podem ser vistos como generalização de escalares e vetores, sendo:

- **Escalares:** Tensores de Ordem 0, possuem 1 componente e representam uma magnitude.
 - Exemplos: temperatura, massa, intensidade de corrente elétrica.
- **Vetores:** Tensores de Ordem 1, possuem 3 componentes e representam uma direção e uma magnitude.
 - Exemplos: Velocidade, aceleração e força.

- **Díados:** Tensores de Ordem 2, possuem 9 componentes e representam duas direções e uma magnitude.

- Exemplos: Permeabilidade Magnética, tensão superficial e coeficientes de difusão

Neste trabalho, tem-se interesse, em específico, nos tensores de segunda ordem em \mathbb{R}^3 . Tais tensores podem ser representado como matrizes de transformação linear 3×3 :

$$\mathbf{T} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix}.$$

3.2.1 Tensores de Orientação

Um caso particular de tensores é introduzido em WESTIN (1994), chamados de Tensores de Orientação. Estes são um caso especial de tensores de segunda ordem não negativos e simétricos, capazes de estimar orientações em um campo tensorial, o que o tornam ideais para este trabalho.

Matematicamente, os tensores de orientação de segunda ordem são definidos como:

$$\mathbf{T} = \sum_{i=1}^n \lambda_i \cdot \mathbf{e}_i \cdot \mathbf{e}_i^T, \quad (3.8)$$

onde λ_i e \mathbf{e}_i são os i -ésimos autovalor e autovetor, respectivamente.

É possível, ainda, representar \mathbf{T} em relação às suas características lineares, planares e esféricas da seguinte forma:

$$\mathbf{T} = (\lambda_1 - \lambda_2)\mathbf{T}_l + (\lambda_2 - \lambda_3)\mathbf{T}_p + \lambda_3\mathbf{T}_s, \quad (3.9)$$

onde \mathbf{T}^l , \mathbf{T}^p e \mathbf{T}^s são as projeções linear, planar e esférica, respectivamente.

O benefício desta representação é que ela permite analisar as características geométricas e orientação dos tensores. Considerando os autovalores $\lambda_1 \geq \lambda_2 \geq \lambda_3$, é possível classificar a forma geométrica dos tensores da seguinte maneira:

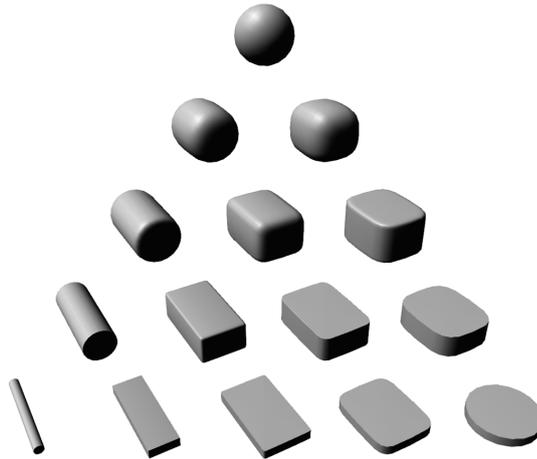


Figura 3.3: Representação geométrica dos tensores de orientação variando os autovalores. (Renhe, 2017)

- Linear: $\lambda_1 \gg \lambda_2 \approx \lambda_3$;
- Planar: $\lambda_1 \approx \lambda_2 \gg \lambda_3$;
- Isotrópico ou Esférico: $\lambda_1 \approx \lambda_2 \approx \lambda_3$.

A Figura 3.3 ilustra a interpretação geométrica dos tensores de orientação lineares, planares e isotrópicos, assim como os tensores intermediários.

3.3 Dinâmica de Fluidos

Segundo Fox et al (2011), "Fluido é uma substância que se deforma de forma continua sob a aplicação de uma tensão de cisalhamento não importa quão pequena seja a tensão". Diferente dos sólidos que tendem a manter sua forma ou sofrer pequenas deformações os fluidos tendem a se mover livremente e se deformar com a menor das interações. Assim, dado seu comportamento, substancias gasosas e líquidas podem ser considerados como fluidos.

Na literatura, há grande interesse na simulação de seu comportamento, seja na simulação de situações reais como na engenharia ao projetar veículos que dependem da aerodinâmica ou na computação gráfica, com uso crescente de fluidos tanto no uso do entretenimento como visualização científica.

3.3.1 Equações de Navier-Stokes

Segundo Stam (1999), há um consenso na literatura de que as equações de Navier-Stokes modelam bem o problema de simulação de fluidos, em especial as equações incompressíveis. Equação (3.10) são as de principal interesse na área de animação na computação gráfica (Bridson, 2015).

A Equação 3.10 mostra uma versão vetorial das *equações incompressíveis de Navier-Stokes*:

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \frac{1}{\rho} \nabla p = \mathbf{g} + \nu \nabla \cdot \nabla \mathbf{u} \\ \nabla \cdot \mathbf{u} = 0 \end{cases} \quad (3.10)$$

Onde \mathbf{u} é a velocidade do fluido, \mathbf{g} é o campo vetorial de força externa, p a pressão do fluido, ρ a densidade do fluido e ν o coeficiente de viscosidade.

3.3.2 Pontos de vista Lagrangeano e Euleriano

Segundo Bridson (2015) existem duas abordagens para simulação de fluidos: a forma Lagrangeana e a forma Euleriana.

A forma Lagrangeana trata a simulação como um sistema de partícula, onde cada partícula carrega as propriedades de velocidade, densidade, temperatura e pressão, além da própria posição. Dentre os algoritmos bem conhecidos que empregam a visão Lagrangeana estão métodos de vórtice (Yaeger et al, 1986) e *Smoothed-Particle Hydrodynamics* (Desbrun et al, 1996).

Já a forma Euleriana, ao invés simular as partículas do fluido, mede as propriedades em pontos fixos do domínio de simulação. Geralmente empregando grades para simular os sistemas, as propriedades de velocidade, densidade e temperatura são armazenadas nas células dessas estruturas.

Desta forma este trabalho, baseia-se na abordagem Euleriana, que segundo Bridson (2015) simplifica o tratamento de derivadas espaciais e cálculo de pressão do fluido.

3.4 Simulação Anisotrópica de Fluidos

O trabalho de Renhe (2017) estende as equações de Navier-Stokes Equação (3.10) integrando a influência dos tensores como parte do meio. O trabalho de Vieira (2020) estende o trabalho e define um método de projeção anisotrópica utilizando o mesmo campo tensorial, tendo como resultado:

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} &= -\mathbf{u} \cdot \nabla \mathbf{u} - \frac{1}{\rho} \nabla_{\mathbf{T}} p + \nu \nabla \cdot \nabla_{\mathbf{T}} \mathbf{u} + \mathbf{T} \mathbf{u} - \beta \mathbf{u} + \mathbf{g}, \\ \nabla \cdot \mathbf{u} &= 0, \\ \frac{\partial \rho}{\partial t} &= -\mathbf{u} \cdot \nabla \rho + \theta \nabla \cdot \nabla_{\mathbf{T}} \rho - \alpha \rho + s. \end{cases} \quad (3.11)$$

O campo tensorial utilizado altera localmente o momento do fluido, influenciando as etapas de advecção, difusão e projeção, permitindo maior controle sobre os passos da simulação. O que justifica a escolha deste método como base para a simulação de fluidos neste trabalho.

4 Método Proposto

O método recebe um conjunto de pontos que definem a curva por onde o usuário deseja que o fluido escoe, além da parametrização dos tensores que serão gerados em cada uma das três regiões do domínio. Para isso, ele é dividido em três etapas:

- **Processamento da curva:** Na primeira etapa do método, uma spline de Catmull-Rom é criada a partir dos pontos de controle definidos e um conjunto de pontos e suas respectivas tangentes são obtidos através da rasterização da spline.
- **Particionamento do domínio:** A partir dos pontos rasterizados da spline obtidos na etapa anterior, o domínio é particionado em três regiões distintas e um conjunto de vetores de orientação é calculado para cada um dos voxels.
- **Cálculo dos tensores:** Os tensores de cada voxel são calculados utilizando os vetores de orientação obtidos na etapa anterior e a parametrização dos autovalores definida pelo usuário.

4.1 Processamento da curva

A primeira etapa do método consiste em processar os pontos pelos quais o usuário deseja que o fluido escoe. Para isso, é gerada uma estrutura de dados intermediária que modela uma spline que passa por todos os pontos. A partir desta estrutura são gerados pontos uniformemente distribuídos pelo caminho, assim como sua orientação em relação à spline.

Para garantir que o caminho gerado passe por todos os pontos definidos pelo usuário, é ideal que seja utilizada uma classe de spline interpolativa, onde tem-se a garantia de que a curva passará por todos os pontos que a definem. Neste trabalho optou-se por utilizar a classe splines de Catmull-Rom, definida na Seção 3.1.

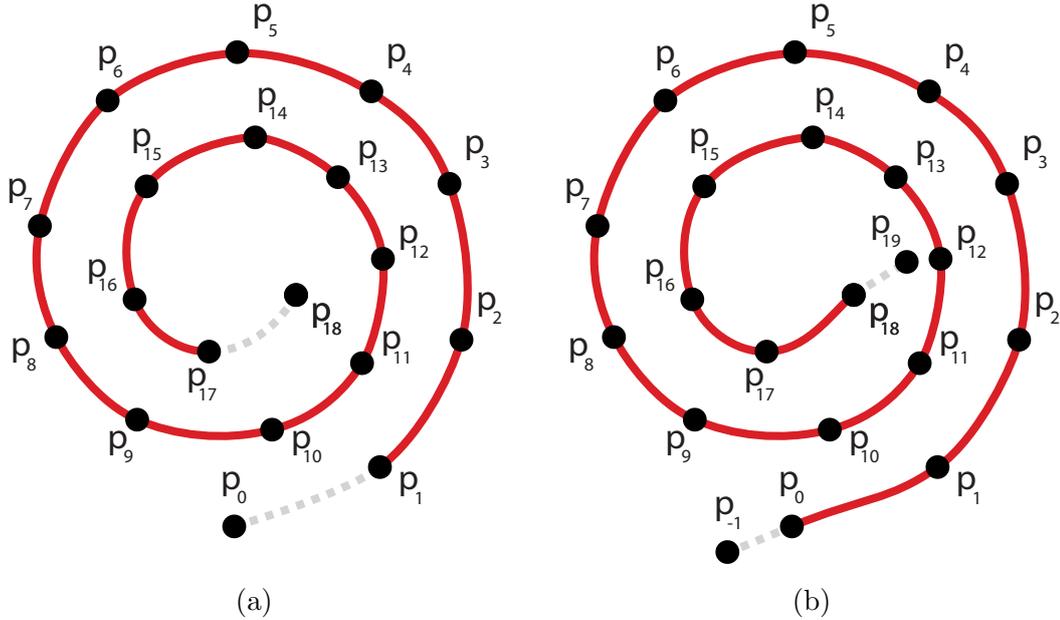


Figura 4.1: Exemplo de spline gerada utilizando apenas os pontos inseridos pelo usuário (a) e spline incluindo os pontos \mathbf{p}_{-1} e \mathbf{p}_{n+1} (b).

4.1.1 Pré-processamento dos Pontos de Controle

Uma das características do método de Catmull-Rom, é que segmentos de curva são definidos de 4 em 4 pontos $\{\mathbf{p}_i, \mathbf{p}_{i+1}, \mathbf{p}_{i+2}, \mathbf{p}_{i+3}\}$, para $0 \leq i \leq n - 3$, de forma que \mathbf{p}_i e \mathbf{p}_{i+3} são utilizados para calcular a tangente dos pontos \mathbf{p}_{i+1} e \mathbf{p}_{i+2} , sem que a curva passe sobre eles. Dessa forma, ao se utilizar apenas o conjunto de pontos $\{\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{n-1}, \mathbf{p}_n\}$ definido pelo usuário, obtém-se uma curva que passa apenas pelos pontos $\{\mathbf{p}_1, \dots, \mathbf{p}_{n-1}\}$. Como este trabalho propõe o uso de uma curva que passe por todos os pontos, é necessário que sejam criados dois pontos adicionais, da seguinte forma:

$$\mathbf{p}_{-1} = \mathbf{p}_0 + \tau(\mathbf{p}_0 - \mathbf{p}_1), \quad (4.1)$$

$$\mathbf{p}_{n+1} = \mathbf{p}_n + \tau(\mathbf{p}_n - \mathbf{p}_{n-1}), \quad (4.2)$$

onde cada segmento da spline passa ser calculado utilizando o conjunto de pontos $\{\mathbf{p}_{i-1}, \mathbf{p}_i, \mathbf{p}_{i+1}, \mathbf{p}_{i+2}\}$, para $0 \leq i \leq n - 2$. A Figura (4.1) ilustra a spline gerada sem e com os pontos \mathbf{p}_{-1} e \mathbf{p}_{n+1} .

4.1.2 Cálculo da Tangente

Ao se utilizar a Equação (3.8) para construir um tensor, se faz necessário definir três vetores unitários e ortogonais entre si, para que sejam utilizados como autovetores no cálculo do campo tensorial. Neste trabalho, optou-se por utilizar a tangente da curva como base para o cálculo dos três vetores.

Para calcular a orientação de cada ponto interno à spline, o primeiro passo é obter a tangente da curva. Partindo da Equação (3.4), é possível obter a tangente da curva com a derivada do polinômio $\mathbf{f}(s)$:

$$\mathbf{f}'(s) = \mathbf{c}_1 + 2\mathbf{c}_2s + 3\mathbf{c}_3s^2, \quad (4.3)$$

com os mesmos valores \mathbf{c}_1 , \mathbf{c}_2 , \mathbf{c}_3 calculados na Equação (3.6). Desta forma, é possível reutilizar a matriz geométrica $\mathbf{M}\mathbf{p}$ da Equação (3.7), calculada durante a geração da spline, e representar a derivada da curva da seguinte forma:

$$\mathbf{f}'(s) = \mathbf{s}^T \mathbf{M}\mathbf{p} = \begin{bmatrix} 0 & 1 & 2s & 3s^2 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\tau & 0 & \tau & 0 \\ 2\tau & \tau - 3 & 3 - 2\tau & -\tau \\ -\tau & 2 - \tau & \tau - 2 & \tau \end{bmatrix} \begin{bmatrix} \mathbf{p}_{i-1} \\ \mathbf{p}_i \\ \mathbf{p}_{i+1} \\ \mathbf{p}_{i+2} \end{bmatrix}. \quad (4.4)$$

Assim, o vetor tangente \mathbf{t} pode ser obtido através da normalização da derivada da curva calculada na Equação (4.4):

$$\mathbf{t} = \frac{\mathbf{f}'(s)}{\|\mathbf{f}'(s)\|}. \quad (4.5)$$

A partir da tangente \mathbf{t} é possível definir uma base ortonormal utilizando dois outros vetores unitários e ortogonais entre si e em relação à \mathbf{t} . Embora tais vetores possam ser derivados da própria curva, neste trabalho, optou-se por defini-los utilizando a distância do voxel à curva.

Ao final desta etapa, a spline é discretizada uniformemente em relação ao interpolador s e, como resultado, é gerado um conjunto de pontos \mathbf{p}'_i em conjunto com sua tangente \mathbf{t}_i . A Figura 4.2 ilustra os pontos e vetores de orientação gerados para um exemplo de curva em 2D.

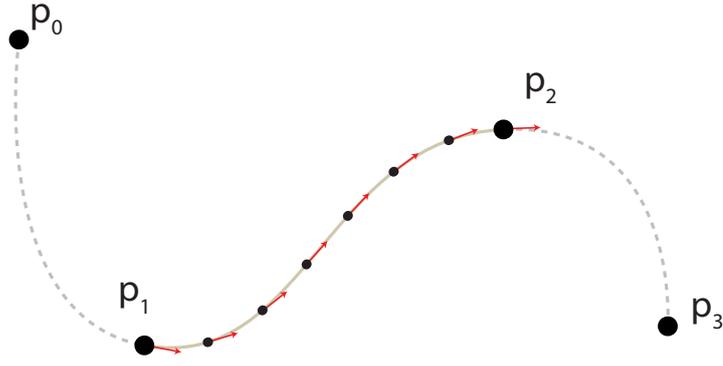


Figura 4.2: Segmento de spline formado pelo conjunto de pontos $\{\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\}$. Nela são mostrados os pontos rasterizados \mathbf{p}_l e suas respectivas tangentes \mathbf{t}_l (vermelho).

4.2 Particionamento do Domínio

A partir dos pontos calculados na etapa anterior, o passo seguinte é particionar o domínio em três regiões: Interna, região de interesse onde espera-se que o fluido seja confinado; Externa, região onde espera-se que o fluido não escoe e Casca, região responsável por isolar as duas regiões anteriores. Além disso, é necessário calcular a orientação de cada um dos voxels. Assim, além dos pontos gerados anteriormente, esta etapa também tem como entrada os raios $r_{interno}$ e $r_{externo}$ que delimitam o tamanho de cada uma das regiões.

Cada voxel na posição (i, j, k) do domínio tem um rótulo $rot_{i,j,k}$ que define à qual região ele pertence além dos vetores de orientação $\mathbf{t}_{i,j,k}$, $\mathbf{n}_{i,j,k}$, $\mathbf{b}_{i,j,k}$, tangente, normal e binormal, respectivamente. No início do método, esses valores são definidos como:

$$\left\{ \begin{array}{l} rot_{i,j,k} = EXTERNA, \\ \mathbf{t}_{i,j,k} = (1, 0, 0), \\ \mathbf{n}_{i,j,k} = (0, 1, 0), \\ \mathbf{b}_{i,j,k} = (0, 0, 1). \end{array} \right. \quad (4.6)$$

Em seguida, para cada um dos pontos p'_l , rasterizados na etapa anterior, são definidas duas regiões cilíndricas concêntricas com raio $r_{interno}$ e $r_{externo}$ e altura h na direção \mathbf{t}_l , onde:

$$h = \alpha \|\mathbf{p}'_{j+1} - \mathbf{p}'_{j-1}\|.$$

Para os testes realizados, o parâmetro $\alpha = 0.8$ se provou o suficiente para gerar campos

sem falhas de cobertura, mesmo em regiões e alta curvatura.

Para cada voxel do domínio, utiliza-se as coordenadas cilíndricas r e z do centro do voxel $\mathbf{v}_{i,j,k}$ para verificar se o mesmo está contido em uma das duas regiões definidas anteriormente. O cálculo de r e z é feito da seguinte forma:

$$\begin{cases} r = \|\mathbf{v}_{i,j,k} - \text{proj}_{\mathbf{t}}(\mathbf{v}_{i,j,k} - \mathbf{p}'_l)\|, \\ z = \mathbf{t} \cdot (\mathbf{v}_{i,j,k} - \mathbf{p}'_l), \end{cases} \quad (4.7)$$

onde $\text{proj}_{\mathbf{t}}(\mathbf{v}_{i,j,k} - \mathbf{p}'_l)$ é a projeção da distância do ponto \mathbf{p}'_l até o centro do voxel $\mathbf{v}_{i,j,k}$ sobre o vetor tangente \mathbf{t}_l . A Figura 4.3 ilustra as regiões utilizadas para a classificação dos voxels, assim como os valores r e z , projetado em 2D.

A partir das coordenadas r e z os voxels podem ser definidos como pertencentes à região Interna, delimitada pelo raio $r_{interno}$, à região da Casca, delimitada pelo raio $r_{externo}$ ou à região externa, caso contrário. Se o voxel for pertencente à região Interna ou à região da Casca, sua $rot_{i,j,k}$ é atualiza da seguinte forma:

$$rot_{i,j,k} = \begin{cases} INTERNA, & z \leq \frac{h}{2} \text{ e } r \leq r_{interno} \\ CASCA, & z \leq \frac{h}{2} \text{ e } r_{interno} < r \leq r_{externo} \end{cases} .$$

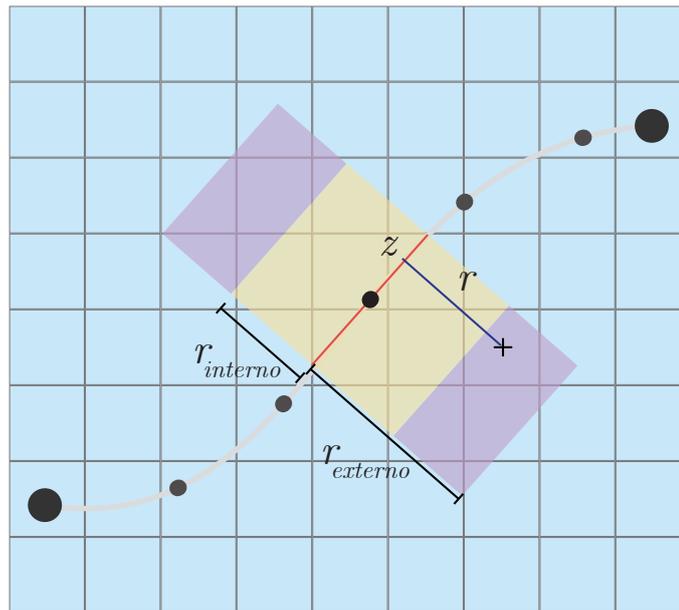


Figura 4.3: Visualização do processo de divisão do domínio em 2D

A Figura 4.4 ilustra o resultado da classificação dos voxels para um ponto e o resultado final da classificação para todos os pontos \mathbf{p}'_l .

4.2.1 Cálculo da Orientação

Partindo da tangente \mathbf{t}_l do ponto \mathbf{p}'_l , calculada na Equação (4.5), é possível definir uma base ortonormal utilizando dois outros vetores \mathbf{n} e \mathbf{b} , normal e binormal, respectivamente, de forma que \mathbf{t} , \mathbf{n} e \mathbf{b} sejam vetores unitários e ortogonais entre si.

Neste trabalho, optou-se por calcular o vetor normal \mathbf{n} para cada voxel utilizando a distância \mathbf{d} do seu centro ao ponto \mathbf{p}'_l :

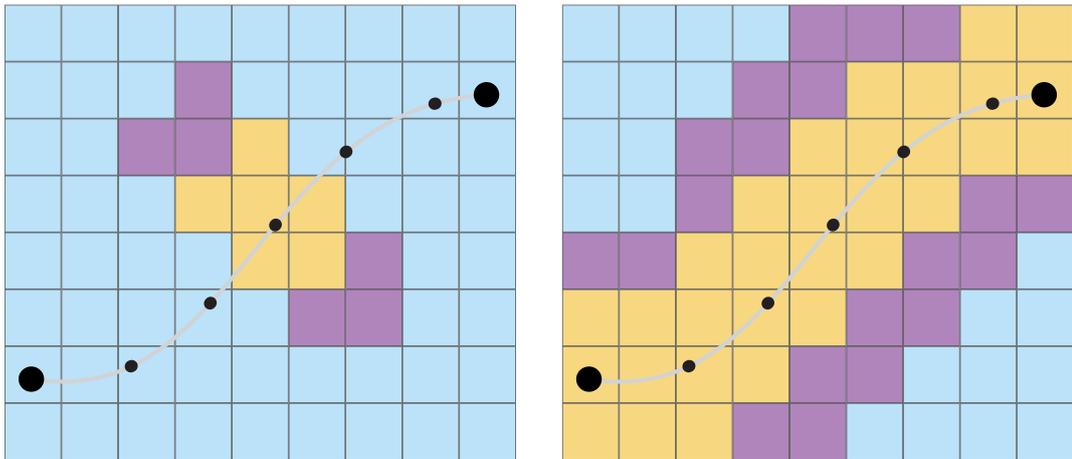
$$\mathbf{d} = \mathbf{v}_{i,j,k} - \mathbf{p}'_l.$$

Para garantir a ortogonalidade dos vetores, \mathbf{n} foi definido da seguinte forma:

$$\mathbf{n} = \frac{\mathbf{d} - (\mathbf{d} \cdot \mathbf{t}_l)\mathbf{t}_l}{\|\mathbf{d} - (\mathbf{d} \cdot \mathbf{t}_l)\mathbf{t}_l\|}. \quad (4.8)$$

Tendo definido \mathbf{t} e \mathbf{n} , o terceiro vetor \mathbf{b} pode ser facilmente obtido através de um produto vetorial simples:

$$\mathbf{b} = \mathbf{n} \times \mathbf{t}. \quad (4.9)$$



(a) Resultado da divisão para um ponto da curva. (b) Resultado da divisão para todos os pontos da curva.

Figura 4.4: Exemplo de divisão de domínio em três regiões: Interna (Amarelo), Casca (Roxo) e Externo (Azul).

Por fim, caso o voxel tenha sido classificado como pertencente à região Interna, ou à Casca, os vetores \mathbf{t}_l , \mathbf{n} e \mathbf{b} são acumulados aos respectivos vetores de orientação do voxel:

$$\begin{cases} \mathbf{t}_{i,j,k} = \mathbf{t}_{i,j,k} + \mathbf{t}_l, \\ \mathbf{n}_{i,j,k} = \mathbf{n}_{i,j,k} + \mathbf{n}, \\ \mathbf{b}_{i,j,k} = \mathbf{b}_{i,j,k} + \mathbf{b}. \end{cases}$$

4.3 Criação do Campo Tensorial

A última etapa do método consiste em utilizar a classificação e os vetores de orientação de cada voxel, obtidos anteriormente, para o cálculo do campo tensorial final. Para isso, o usuário deve definir três conjuntos de autovalores, $\mathbf{\Lambda}_I = \{\lambda_{Ix}, \lambda_{Iy}, \lambda_{Iz}\}$, $\mathbf{\Lambda}_C = \{\lambda_{Cx}, \lambda_{Cy}, \lambda_{Cz}\}$ e $\mathbf{\Lambda}_E = \{\lambda_{Ex}, \lambda_{Ey}, \lambda_{Ez}\}$, para os tensores das regiões Interna, Casca e Externa, respectivamente.

O primeiro passo dessa etapa é obter, para cada voxel, três vetores \mathbf{e}_x , \mathbf{e}_y e \mathbf{e}_z , formando uma base ortonormal em \mathbb{R}^3 , para que sejam utilizados como autovetores do tensor no voxel. Para isso, $\mathbf{t}_{i,j,k}$, $\mathbf{n}_{i,j,k}$, $\mathbf{b}_{i,j,k}$, são ortonormalizados através do processo de Gram-Schmidt (Björck, 1994):

$$\begin{cases} \mathbf{e}_x = \frac{\mathbf{t}_{i,j,k}}{\|\mathbf{t}_{i,j,k}\|}, \\ \mathbf{e}_y = \frac{\mathbf{n}_{i,j,k} - \text{proj}_{\mathbf{e}_x} \mathbf{n}_{i,j,k}}{\|\mathbf{n}_{i,j,k} - \text{proj}_{\mathbf{e}_x} \mathbf{n}_{i,j,k}\|}, \\ \mathbf{e}_z = \frac{\mathbf{b}_{i,j,k} - \text{proj}_{\mathbf{e}_x} \mathbf{b}_{i,j,k} - \text{proj}_{\mathbf{e}_y} \mathbf{b}_{i,j,k}}{\|\mathbf{b}_{i,j,k} - \text{proj}_{\mathbf{e}_x} \mathbf{b}_{i,j,k} - \text{proj}_{\mathbf{e}_y} \mathbf{b}_{i,j,k}\|}, \end{cases} \quad (4.10)$$

onde $\text{proj}_{\mathbf{e}_x}(\cdot)$ e $\text{proj}_{\mathbf{e}_y}(\cdot)$ são os operadores de projeção sobre os vetores \mathbf{e}_x e \mathbf{e}_y , respectivamente:

$$\text{proj}_{\mathbf{u}} \mathbf{w} = \frac{\mathbf{w} \cdot \mathbf{u}}{\mathbf{u} \cdot \mathbf{u}} \mathbf{u}.$$

Em seguida, um conjunto de autovalores é selecionado com base na classificação $\text{rot}_{i,j,k}$ e, em conjunto com os autovetores obtidos na Equação (4.10), o tensor $\mathbf{T}_{i,j,k}$ do voxel é calculado utilizando a Equação (3.8):

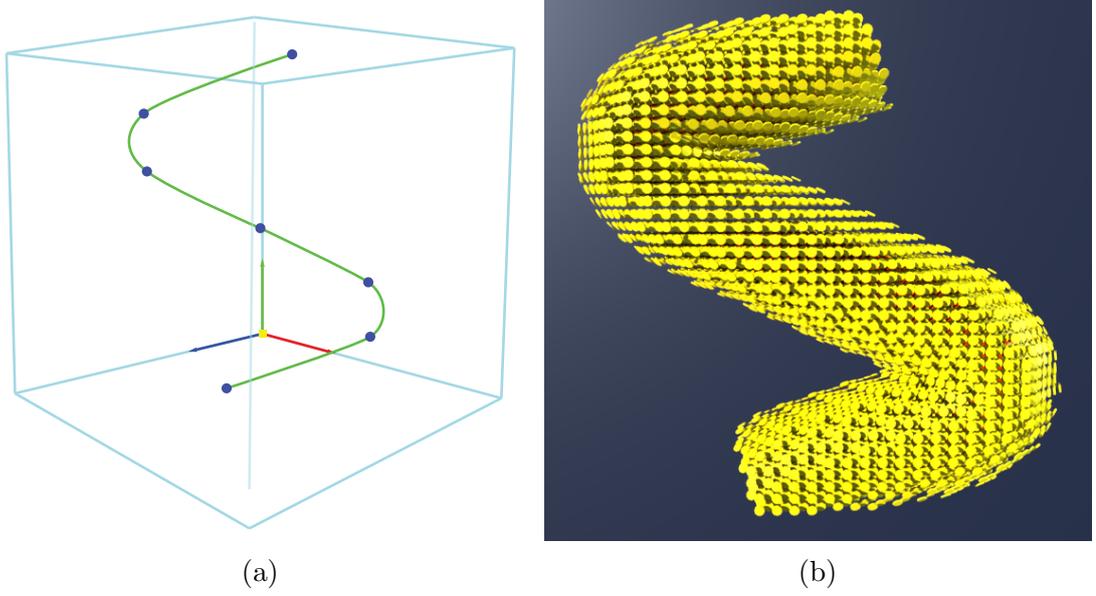


Figura 4.5: Exemplo de spline (a) e a representação geométrica do campo tensorial gerado (b) utilizando os conjuntos de autovalores $\mathbf{\Lambda}_I = \{1, 1, 1\}$, $\mathbf{\Lambda}_C = \{1, 0.05, 1\}$ e $\mathbf{\Lambda}_E = \{1, 1, 1\}$.

$$\mathbf{T}_{i,j,k} = \begin{cases} \lambda_{Ix} \mathbf{e}_x \mathbf{e}_x^T + \lambda_{Iy} \mathbf{e}_y \mathbf{e}_y^T + \lambda_{Iz} \mathbf{e}_z \mathbf{e}_z^T & \text{se } rot_{i,j,k} = \text{INTERNA}, \\ \lambda_{Cx} \mathbf{e}_x \mathbf{e}_x^T + \lambda_{Cy} \mathbf{e}_y \mathbf{e}_y^T + \lambda_{Cz} \mathbf{e}_z \mathbf{e}_z^T & \text{se } rot_{i,j,k} = \text{CASCA}, \\ \lambda_{Ex} \mathbf{e}_x \mathbf{e}_x^T + \lambda_{Ey} \mathbf{e}_y \mathbf{e}_y^T + \lambda_{Ez} \mathbf{e}_z \mathbf{e}_z^T & \text{se } rot_{i,j,k} = \text{EXTERNA}. \end{cases} \quad (4.11)$$

Como resultado, tem-se um campo tensorial \mathbf{T} , calculado a partir da curva definida pelos pontos passados como parâmetro, que pode ser utilizado diretamente na Equação (3.11) para simulação anisotrópica de fluidos. A Figura 4.5 ilustra a representação gráfica do campo tensorial gerado para um conjunto de pontos.

5 Resultados

Para avaliar o método proposto nesse trabalho, foram definidos três splines para testar diferentes aspectos da simulação. Para cada uma das splines, foram realizados experimentos com conjuntos distintos de autovalores Λ_I , Λ_C e Λ_E , tais valores foram definidos partindo do princípio utilizado por Vieira (2020) de utilizar tensores planares para dividir o domínio em regiões distintas.

As simulações foram feitas utilizando um simulador implementado com base na solução no modelo de equações de Navier-Stokes proposto por Vieira (2020), com o campo tensorial \mathbf{T} , resultante do processo de construção proposto neste trabalho diretamente nas Equações (3.11).

Todas as simulações foram feitas utilizando uma grade de $40 \times 40 \times 40$ voxels e em cada uma delas, foi inserida uma força externa na direção tangente à spline, além da força de empuxo definida pela seguinte equação:

$$\mathbf{g}_{emp} = \kappa(h - h_{amb})\mathbf{k} - v\rho\mathbf{k}, \quad (5.1)$$

onde \mathbf{k} é o vetor canônico $(0, 0, 1)$, h é a temperatura do fluido, h_{amb} é temperatura média do domínio, calculada toda iteração e ρ é a densidade do fluido. Os parâmetros κ e v variaram para simulação.

5.1 Curvas com Pontos de Controle Colineares

Com o propósito de testar a capacidade do campo tensorial de conter o fluido dentro da região de interesse, foi definida uma spline com 2 pontos (veja a Tabela 5.1), formando uma

Ponto	Coordenadas
\mathbf{p}_0	$(0, 0.35, 1)$
\mathbf{p}_1	$(1, 0.35, 0)$

Tabela 5.1: Pontos de controle definindo uma reta horizontal.

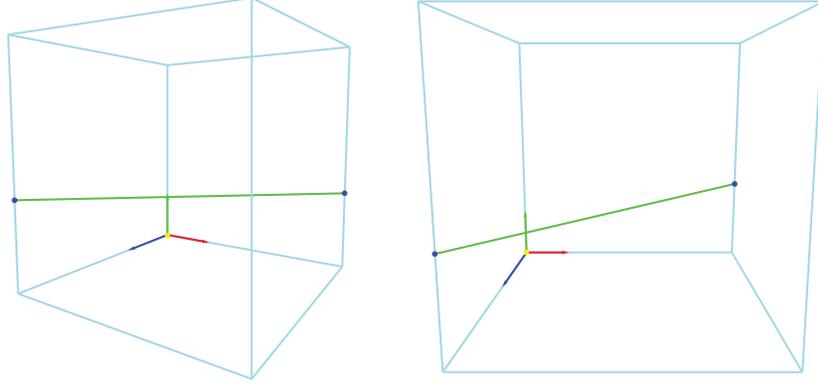


Figura 5.1: Spline de uma reta horizontal definida por 2 pontos.

reta horizontal. A Figura 5.1 ilustra o conjunto de pontos e a spline resultante utilizada para a simulação.

Para mesma spline foram realizados dois experimentos, um utilizando tensores isotrópicos em todo o domínio e outro utilizando tensores planares na região da Casca. Ambas as simulações utilizaram o mesmo conjunto de parâmetros, sendo o coeficiente de difusão $\theta = 0.001$, o coeficiente de viscosidade $\phi = 0.075$, a fonte de densidade externa $s = 0.001$, fonte de calor $h_0 = 0.1$ e força externa \mathbf{g} posicionada na base da spline com a direção tangente à curva e módulo $\|\mathbf{g}\| = 0.11$ e fatores de empuxo $\kappa = 1.0$ e gravidade $v = 0.025$.

É importante notar que a Equação (3.11) pode ser vista como uma generalização das equações de Navier-Stokes para casos anisotrópicos e, quando o campo tensorial \mathbf{T} é a matriz identidade \mathbf{I}_3 , as equações são revertidas à forma isotrópica como a Equação (3.10).

Por esse motivo, para ser utilizado como base de comparação, os conjuntos de autovalores foram definidos como $\mathbf{\Lambda}_I = \{1, 1, 1\}$, $\mathbf{\Lambda}_C = \{1, 1, 1\}$ e $\mathbf{\Lambda}_E = \{1, 1, 1\}$, tendo como resultado um campo tensorial isotrópico no qual todos os tensores são a matriz identidade \mathbf{I}_3 .

A Figura 5.2 mostra a simulação para o campo tensorial isotrópico. Note que, apesar da força externa empurrando o fluido na direção da curva, o empuxo faz com que a densidade flutue, saindo da trajetória definida.



Figura 5.2: Frames 36, 78, 120, 162, 204, 246, 288 e 330 da simulação da curva com pontos de controle colineares utilizando campo tensorial isotrópico.

5.1.1 Campo Tensorial com Casca Planar

Com o intuito de apenas separar a região Interna da região Externa, neste experimento os conjuntos de autovalores foram definidos como $\Lambda_I = \{1, 1, 1\}$, $\Lambda_C = \{1, 0.05, 1\}$ e $\Lambda_E = \{1, 1, 1\}$. Dessa forma, tanto os tensores da região Interna, quanto a Externa tem seus tensores definidos como a matriz identidade e os tensores da Casca, definidos como tensores planares com o autovetor com menor autovalor associado alinhado com a normal do cilindro. A Figura 5.3 ilustra a representação geométrica do campo tensorial gerado.

A Figura 5.4 ilustra o resultado da simulação utilizando o campo tensorial com a Casca de tensores planares. Observe que mesmo com o empuxo, o fluido permanece

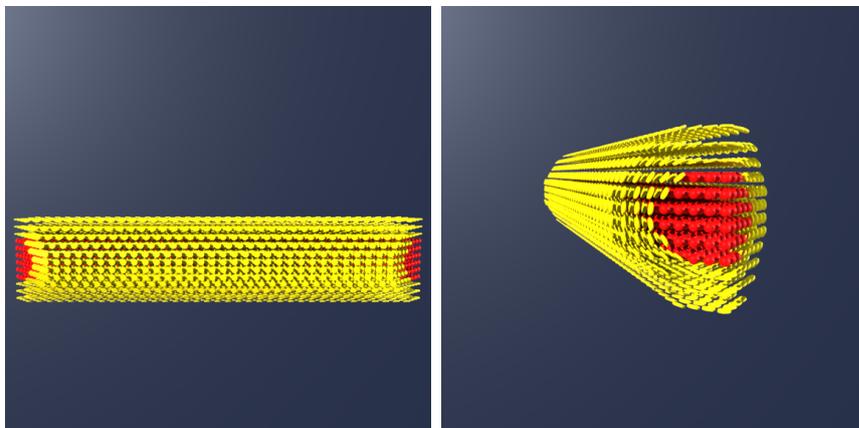


Figura 5.3: Campo tensorial gerado para curva com dois pontos colineares utilizando os conjuntos de autovalores $\Lambda_I = \{1, 1, 1\}$, $\Lambda_C = \{1, 0.05, 1\}$ e $\Lambda_E = \{1, 1, 1\}$. Os tensores isotrópicos da região Externa foram omitidos pra facilitar a visualização.

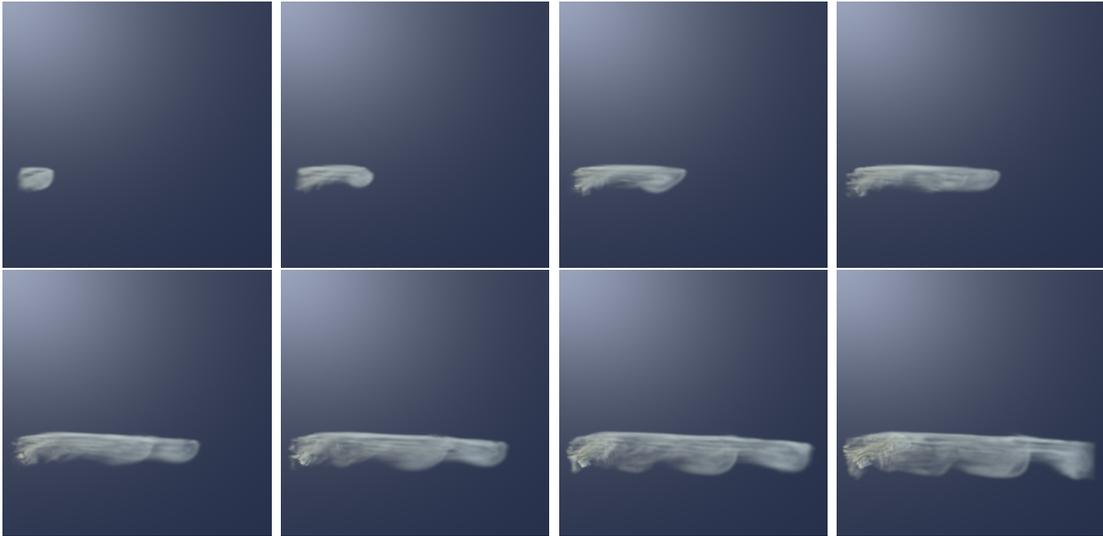


Figura 5.4: Frames 36, 78, 120, 162, 204, 246, 288 e 330 da simulação da curva com pontos de controle colineares utilizando tensores planares na região da Casca.

confinado entre a região Interna e a Casca.

5.2 Curvas com Pontos de Controle Coplanares Sem Inflexão

Com o propósito de testar a capacidade do campo tensorial manter o fluido dentro da região de interesse em regiões de curvatura não nula, foi definida uma spline com 7 pontos coplanares (veja a Tabela 5.2), formando uma curva planar sem inflexão. A Figura 5.5 ilustra o conjunto de pontos e a spline resultante utilizada para a simulação.

Assim como no conjunto de testes anterior, para mesma spline foram realizados dois experimentos, um utilizando tensores isotrópicos em todo o domínio e outro utilizando tensores planares na região da Casca. Ambas as simulações utilizaram o mesmo conjunto de parâmetros, sendo o coeficiente de difusão $\theta = 0.001$, o coeficiente de viscosidade $\phi = 0.075$, a fonte de densidade externa $s = 0.001$, fonte de calor $h_0 = 0$ e força externa

Ponto	Coordenadas	Ponto	Coordenadas
\mathbf{p}_0	(0.05, 0.05, 0.95)	\mathbf{p}_4	(0.65, 0.55, 0.35)
\mathbf{p}_1	(0.2, 0.2, 0.8)	\mathbf{p}_5	(0.8, 0.55, 0.2)
\mathbf{p}_2	(0.35, 0.35, 0.65)	\mathbf{p}_6	(0.95, 0.49, 0.05)
\mathbf{p}_3	(0.5, 0.49, 0.5)		

Tabela 5.2: Conjunto de pontos de controle definindo uma reta planar sem inflexão.

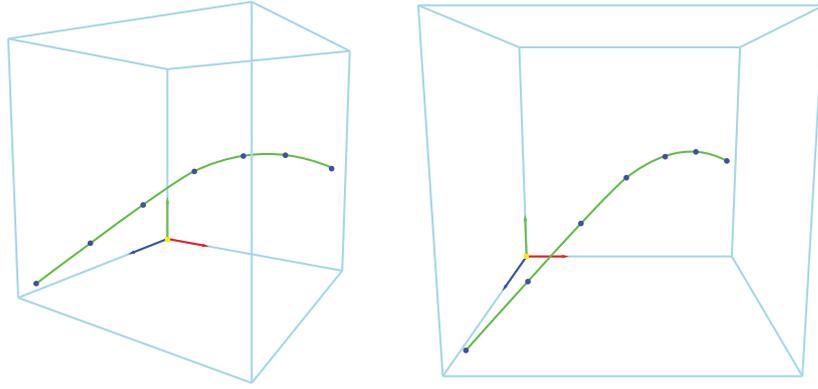


Figura 5.5: Spline gerada utilizando 7 pontos de controle coplanares e sem inflexão.

\mathbf{g} posicionada na base da spline com a direção tangente à curva e módulo $\|\mathbf{g}\| = 0.11$. Nestes experimentos tanto a gravidade quanto o empuxo foram removidos utilizando os parâmetros $\kappa = 0$ e $\nu = 0$.

Para definir a base de comparação entre o uso de campos tensoriais e a simulação utilizando as equações isotrópicas de Navier-Stokes, assim como na Seção 5.1, foi construído o campo tensorial isotrópico definido os conjuntos de autovalores como $\mathbf{\Lambda}_I = \{1, 1, 1\}$, $\mathbf{\Lambda}_C = \{1, 1, 1\}$ e $\mathbf{\Lambda}_E = \{1, 1, 1\}$.

A Figura 5.6 ilustra o resultado da simulação utilizando o campo tensorial isotrópico. Note que apesar do fluido passar pela trajetória definida durante a maior parte da simulação, o fluido deixa de acompanhar a trajetória assim que sua curvatura aumenta.

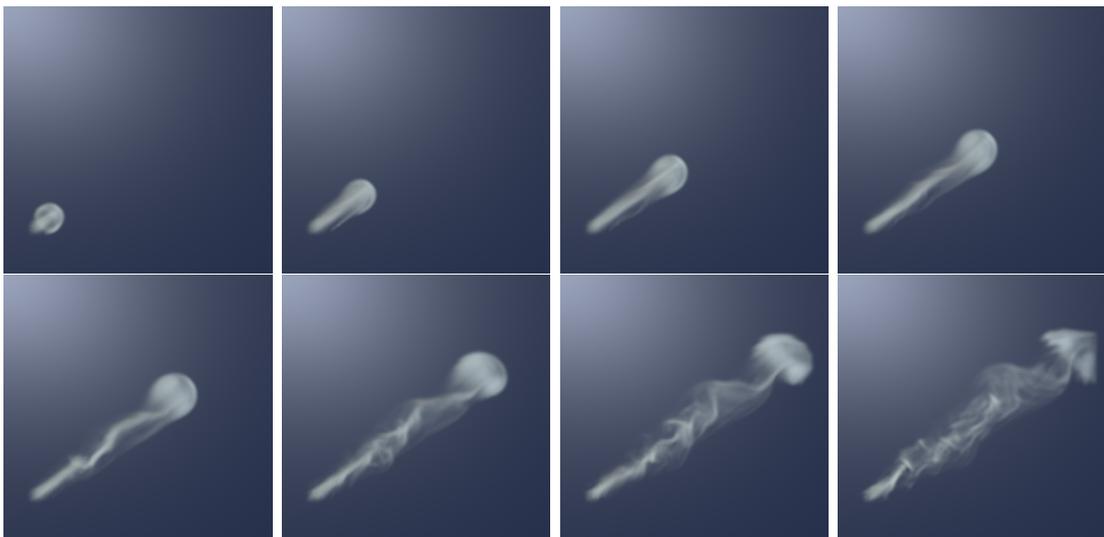


Figura 5.6: Frames 36, 78, 120, 162, 204, 246, 288 e 330 da simulação da curva planar sem inflexão utilizando campo tensorial isotrópico.

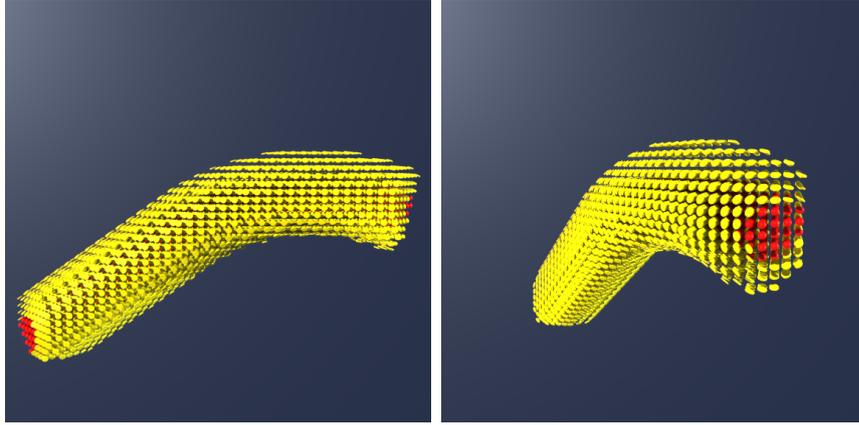


Figura 5.7: Campo tensorial gerado para curva com sete pontos coplanares utilizando os conjuntos de autovalores $\Lambda_I = \{1, 1, 1\}$, $\Lambda_C = \{1, 0.05, 1\}$ e $\Lambda_E = \{1, 1, 1\}$. Os tensores isotrópicos da região Externa foram omitidos pra facilitar a visualização.

5.2.1 Campo Tensorial com Casca Planar

Da mesma forma como no exemplo da curva formada por dois pontos, as regiões Interna e Externa foram separadas utilizando tensores planares na região da Casca. O campo tensorial foi gerado definindo o conjunto de autovalores $\Lambda_I = \{1, 1, 1\}$, $\Lambda_C = \{1, 0.05, 1\}$ e $\Lambda_E = \{1, 1, 1\}$. A Figura 5.7 ilustra a representação geométrica do campo tensorial gerado.

O resultado da simulação é exibido na Figura 5.8. Observe que, durante toda a simulação, o fluido permanece confinado na região Interna, se acumulando na fronteira entre as regiões Interna e Casca conforme se aproxima da região de maior curvatura.



Figura 5.8: Frames 36, 78, 120, 162, 204, 246, 288 e 330 da simulação da curva planar sem inflexão utilizando campo tensorial com planares na região da Casca.

Ponto	Coordenadas	Ponto	Coordenadas
\mathbf{p}_0	(0.95, -0.01, 0.72)	\mathbf{p}_7	(0.05, 0.59, 0.28)
\mathbf{p}_1	(0.81, 0.07, 0.89)	\mathbf{p}_8	(0.19, 0.67, 0.11)
\mathbf{p}_2	(0.61, 0.16, 0.99)	\mathbf{p}_9	(0.39, 0.76, 0.01)
\mathbf{p}_3	(0.39, 0.24, 0.99)	\mathbf{p}_{10}	(0.61, 0.84, 0.01)
\mathbf{p}_4	(0.19, 0.33, 0.89)	\mathbf{p}_{11}	(0.81, 0.93, 0.11)
\mathbf{p}_5	(0.05, 0.41, 0.72)	\mathbf{p}_{12}	(0.95, 1.01, 0.28)
\mathbf{p}_6	(0, 0.5, 0.5)		

Tabela 5.3: Conjunto de pontos de controle definindo uma curva não planar.

Neste exemplo o campo tensorial consegue impedir que o fluido escoe pela região Externa durante toda a simulação. É importante notar, também, que a utilização do campo tensorial reduz a vorticidade do fluido.

5.3 Curva com Pontos de Controle não Coplanares

Para avaliar o comportamento do fluido em casos mais complexos, com curvas não planares, foi definida uma spline com 13 pontos (veja a Tabela 5.3). Os pontos de controle \mathbf{p}_i , tal que $0 \leq i < 13$ distribuídos de forma que suas coordenadas (x_i, y_i, z_i) são calculadas da seguinte forma:

$$\begin{cases} x_i = \cos\left(\frac{2\pi i}{12}\right), \\ y_i = \frac{i}{6} - 1, \\ z_i = \sin\left(\frac{2\pi i}{12}\right). \end{cases} \quad (5.2)$$

A Figura 5.9 ilustra o conjunto de pontos e a spline resultante utilizada para a simulação.

Para mesma spline foram realizados três experimentos, o primeiro utilizando

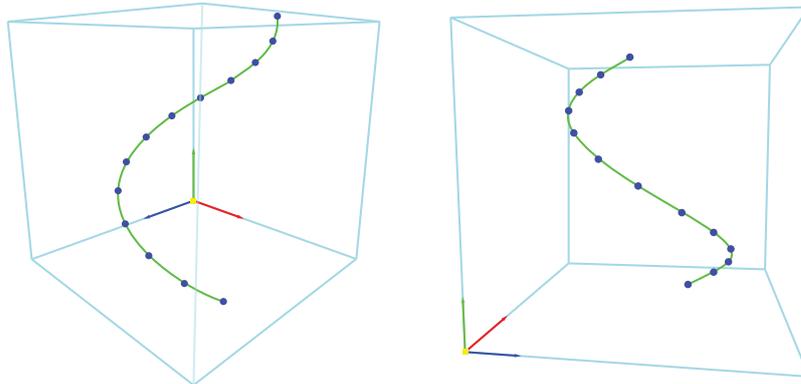


Figura 5.9: Spline gerada a partir dos pontos definidos pela Equação (5.2)

tensores planares na região da Casca, o segundo utilizando tensores planares na região da Casca e lineares na região Interna e, por fim, um experimento utilizando tensores isotrópicos com autovalores menores que 1.0 na região Externa, tensores planares na região da Casca e lineares na região Interna.

Todas as três simulações utilizaram o mesmo conjunto de parâmetros, sendo o coeficiente de difusão $\theta = 0.001$, o coeficiente de viscosidade $\phi = 0.075$, a fonte de densidade externa $s = 0.001$, fonte de calor $h_0 = 0.1$ e força externa \mathbf{g} posicionada na base da spline com a direção tangente à curva e módulo $\|\mathbf{g}\| = 0.055$ e fatores de empuxo $\kappa = 1.0$ e gravidade $\nu = 0.025$. Nestes exemplos, a força externa, a fonte de densidade e a fonte de calor foram inseridas somente até o frame 150.

5.3.1 Campo Tensorial com Casca Planar

Utilizando o mesmo princípio dos exemplos anteriores de isolar a região Interna da região Externa utilizando tensores planares, neste exemplo os conjuntos de autovalores foram

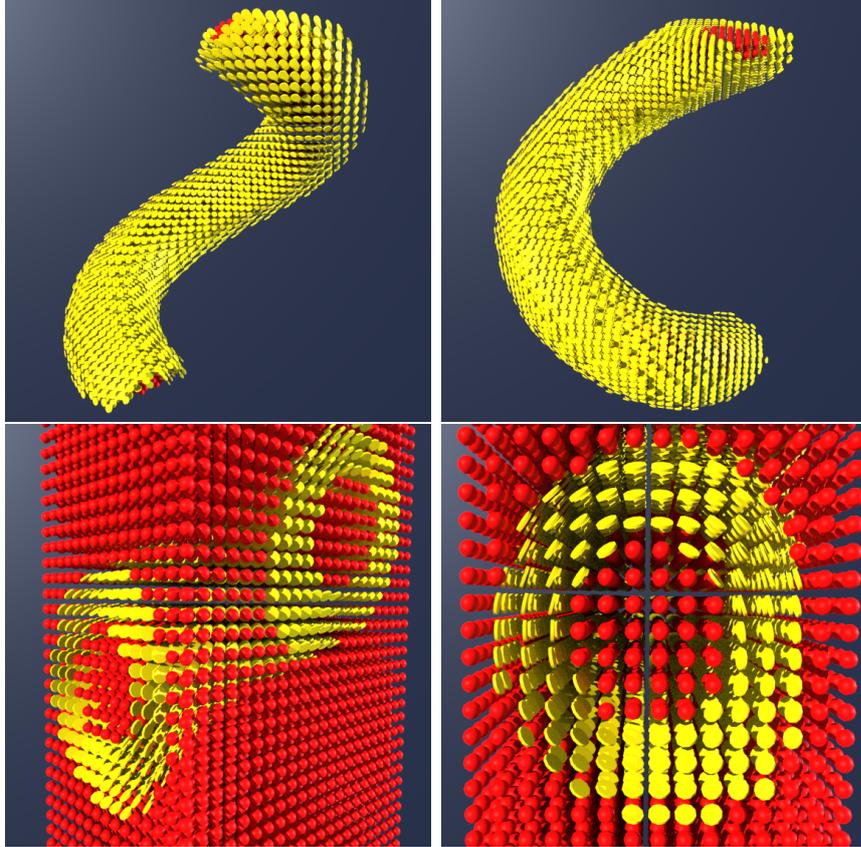


Figura 5.10: Campo tensorial gerado para curva com treze pontos não coplanares utilizando os conjuntos de autovalores $\mathbf{\Lambda}_I = \{1, 1, 1\}$, $\mathbf{\Lambda}_C = \{1, 0.05, 1\}$ e $\mathbf{\Lambda}_E = \{1, 1, 1\}$. Os tensores isotrópicos foram omitidos pra facilitar a visualização.

definidos como $\Lambda_I = \{1, 1, 1\}$, $\Lambda_C = \{1, 0.05, 1\}$ e $\Lambda_E = \{1, 1, 1\}$, gerando tensores isotrópicos igual a matriz identidade $\mathbf{T} = \mathbf{I}_3$ nas regiões Interna e Externa do domínio e tensores planares como o autovetor com o menor autovalor associado alinhado com a normal da spline. A Figura 5.10 mostra a representação geométrica do campo tensorial gerado.

O resultado da simulação é ilustrado pela Figura 5.11. Note que, mesmo utilizando uma força externa com modulo menor que os exemplos anteriores, o campo tensorial não é capaz de conter o fluido dentro da trajetória, ou fazer com que o fluido escoe até o final da curva. No frame 250 já é possível observar o fluido escoando para a região Externa.



Figura 5.11: Frames 25, 70, 115, 160, 205, 250, 295, 340, 385, 430, 475, 520, 565, 610, 655 e 700 da simulação da hélice utilizando tensores planares na região da Casca.

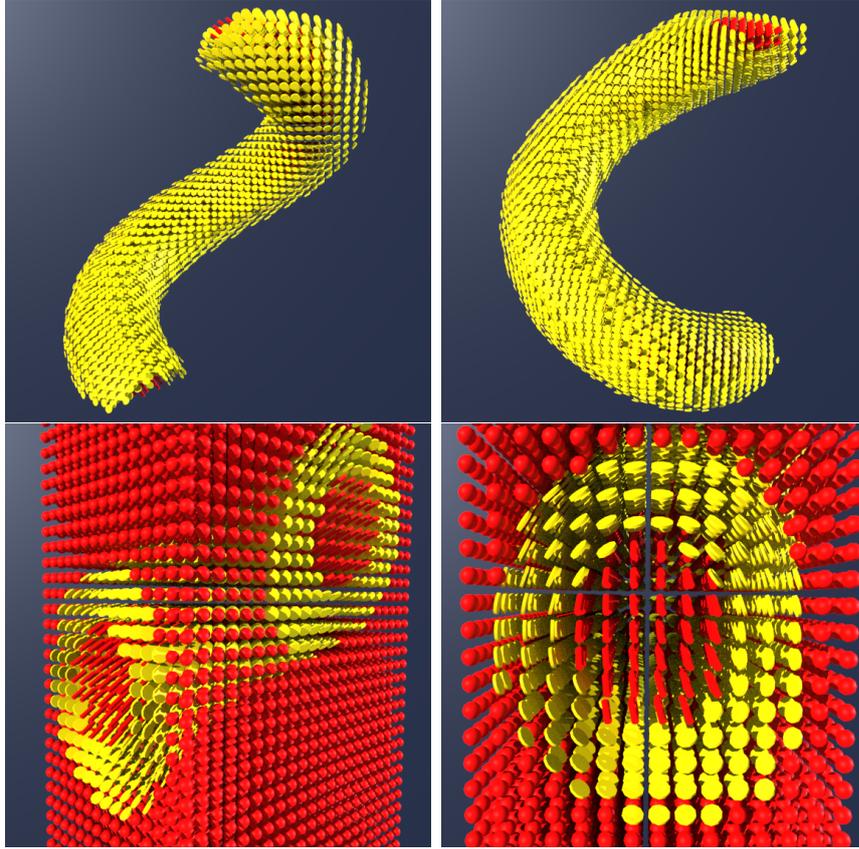


Figura 5.12: Campo tensorial gerado para curva com treze pontos não coplanares utilizando os conjuntos de autovalores $\Lambda_I = \{1.2, 0.5, 0.5\}$, $\Lambda_C = \{1, 0.05, 1\}$ e $\Lambda_E = \{1, 1, 1\}$. Os tensores isotrópicos foram omitidos pra facilitar a visualização.

5.3.2 Campo Tensorial com Casca Planar e Região Interna Linear

Com o intuito de aumentar o controle sobre o fluido e restringir a troca de densidade e velocidade entre a região Interna e a região da Casca, além dos tensores planares na Casca, neste exemplo, também foram utilizados tensores lineares na região Interna. Para isso, os conjuntos de autovalores foram definidos como $\Lambda_I = \{1.2, 0.5, 0.5\}$, $\Lambda_C = \{1, 0.05, 1\}$ e $\Lambda_E = \{1, 1, 1\}$. A Figura 5.12 ilustra o campo tensorial gerado.

Note que os tensores da região Interna possuem autovalores maiores que 1.0 no valor associado ao autovetor com direção tangente à curva e valor menor que 1.0 nos demais, desta forma, velocidades alinhadas à tangente da curva são amplificadas enquanto as demais são reduzidas.

A Figura 5.13 ilustra o resultado da simulação utilizando o campo tensorial construído. Nesta simulação é possível observar que os tensores lineares da região interna em



Figura 5.13: Frames 25, 70, 115, 160, 205, 250, 295, 340, 385, 430, 475, 520, 565, 610, 655 e 700 da simulação da hélice utilizando tensores planares na região da Casca.

conjunto com os tensores planares na região da Casca são capazes de levar o fluido até o final da trajetória, no entanto, nem toda a densidade permanece confinada entre a região Interna e a Casca, parte dela escapa para a região Externa, onde pode se movimentar sem nenhuma restrição.

5.3.3 Campo Tensorial com Casca Planar, Interior Linear e Exterior Dissipativo

Para reduzir a troca de densidade entre a região da Casca e a Externa, neste exemplo foram utilizados os seguintes conjuntos de autovalores: $\Lambda_I = \{0.5, 1.2, 0.5\}$, $\Lambda_C = \{0.5, 0.05, 0.5\}$ e $\Lambda_E = \{0.05, 0.05, 0.05\}$. Desta forma, a região Externa foi preenchida com tensores

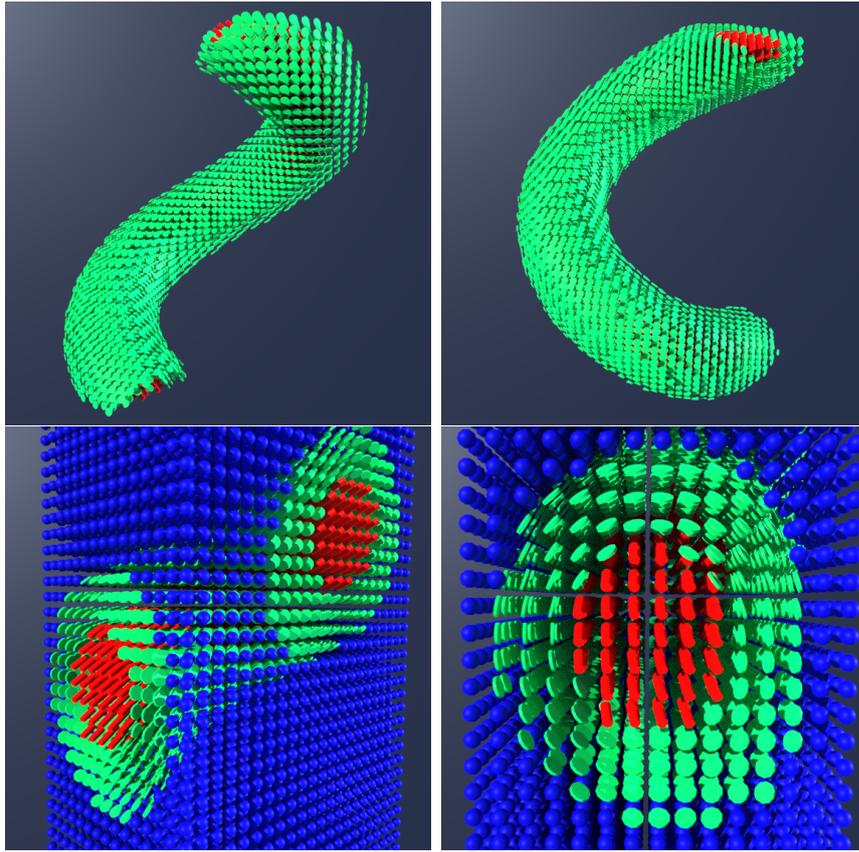


Figura 5.14: Campo tensorial gerado para curva com treze pontos não coplanares utilizando os conjuntos de autovalores $\Lambda_I = \{0.5, 1.2, 0.5\}$, $\Lambda_C = \{0.5, 0.05, 0.5\}$ e $\Lambda_E = \{0.05, 0.05, 0.05\}$. Os tensores isotrópicos foram omitidos pra facilitar a visualização.

isotrópicos com autovalores menores que 1.0, agindo como dissipadores de momento e dificultando as trocas de densidade e velocidade. Além disso, os tensores planares da região da Casca tiveram seus autovalores associados à tangente e binormal da curva reduzidos. A Figura 5.14 ilustra a representação geométrica do campo tensorial.

A Figura 5.15 mostra o resultado da simulação utilizando o campo tensorial. Note que além de ser mais efetivo que os exemplos anteriores em levar o fluido até o final da curva, o campo tensorial foi capaz de manter toda a densidade confinada entre a região Interna e a Casca durante toda a simulação.



Figura 5.15: Frames 25, 70, 115, 160, 205, 250, 295, 340, 385, 430, 475, 520, 565, 610, 655 e 700 da simulação da hélice utilizando tensores lineares na região Interna, planares na região da Casca, e isotrópicos com autovalores menores que 1 na região Externa.

6 Conclusão

Neste trabalho foi apresentado um método de construção de campos tensoriais utilizando splines de Catmull-Rom geradas a partir de pontos por onde deseja-se que o fluido escoe. Durante todo o processo de desenvolvimento do método foram testados diferentes conjuntos de pontos e para cada conjunto, diferentes configurações de campos tensoriais.

A partir dos resultados foi possível observar que para curvas mais simples como curvas definidas por pontos coplanares, que não apresentam pontos de inflexão, a utilização de campos tensoriais com tensores planares na região da Casca foi o suficiente para manter o fluido confinado na região de interesse de forma que ele escoasse pela trajetória. Para curvas mais elaboradas, maior cuidado é necessário durante a escolha de parâmetros para a geração do campo tensoriais. Os resultados para o caso apresentado na Seção 5.3, mostram que tensores isotrópicos dissipativos, com norma menor que 1.0, na região Externa e tensores lineares na região Interna são configurações válidas e podem servir como ponto de partida para a definição de parâmetros para outras curvas.

É importante notar que, ainda que o método tenha sido capaz fazer o fluido escoar pela trajetória e de mantê-lo confinado na região de interesse, ele pode não ser ideal para utilização direta em sistemas de animação. Nos resultados, foi possível observar a redução de detalhes e vorticidade durante a simulação do fluido. A utilização de esquemas de confinamento de vórtices como o proposto no trabalho de Fedkiw (2001) pode ser útil para destacar a vorticidade reduzida pela utilização de campos tensoriais.

Uma das principais limitações do método está relacionada ao elevado custo computacional da solução das equações modificadas de Navier-Stokes propostas por Vieira (2020). O problema requer um resolvidor de sistemas assimétricos de equações, o que pode ser consideravelmente mais custoso que métodos para solução de sistemas simétricos. Desta forma, a utilização de resoluções elevadas pode se tornar proibitiva. Como alternativa, para uso em sistemas de animação, a simulação pode ser realizada em conjunto com técnicas de refinamento de detalhes como a proposta no trabalho de Kim (2008) que, a partir do campo de velocidade de uma simulação em baixa resolução, realiza um

processo de *upscaling* e utiliza *wavelets* para reintroduzir turbulências no campo já em alta resolução.

Diante dos resultados obtidos, é possível concluir que o método desenvolvido é capaz de gerar um campo tensorial que controle o fluido de forma que este escoe pela trajetória definida, no entanto um estudo mais aprofundado é necessário para que o método possa ser empregado em soluções para animação de fluido de forma prática.

Bibliografia

- Björck, Å. Numerics of gram-schmidt orthogonalization. **Linear Algebra and Its Applications**, v.197, p. 297–316, 1994.
- Bridson, R. **Fluid simulation for computer graphics**. AK Peters/CRC Press, 2015.
- Browning, M.; Barnes, C.; Ritter, S. ; Finkelstein, A. **Stylized keyframe animation of fluid simulations**. In: Proceedings of the Workshop on Non-Photorealistic Animation and Rendering, p. 63–70. ACM, 2014.
- Catmull, E.; Rom, R. **A class of local interpolating splines**. In: Computer aided geometric design, p. 317–326. Elsevier, 1974.
- Desbrun, M.; Gascuel, M.-P. **Smoothed particles: a new paradigm for animating highly deformable bodies**. In: Computer Animation and Simulation, p. 61–76. Springer, 1996.
- Fedkiw, R.; Stam, J. ; Jensen, H. W. **Visual simulation of smoke**. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, p. 15–22. ACM, 2001.
- Fox, R. W. **Fox and mcdonald’s introduction to fluid mechanics**, 2011.
- Hays, J.; Essa, I. **Image and video based painterly animation**. In: Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering, p. 113–120. ACM, 2004.
- Hearn, D.; Baker, M. P. **Computer Graphics, C Version**. Prentice Hall, 1996.
- Kim, Y.; Machiraju, R. ; Thompson, D. **Path-based control of smoke simulations**. In: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, p. 33–42. Eurographics Association, 2006.
- Kim, T.; Thürey, N.; James, D. ; Gross, M. **Wavelet turbulence for fluid simulation**. In: ACM Transactions on Graphics (TOG), volume 27, p. 50. ACM, 2008.
- Kolecki, J. C. An introduction to tensors for students of physics and engineering. **Relatorio tecnco**, 2002.
- Manteaux, P.-L.; Vimont, U.; Wojtan, C.; Rohmer, D. ; Cani, M.-P. **Space-time sculpting of liquid animation**. In: Proceedings of the 9th International Conference on Motion in Games, p. 61–71. ACM, 2016.
- Palacios, J.; Zhang, E. **Rotational symmetry field design on surfaces**. In: ACM Transactions on Graphics (TOG), volume 26, p. 55. ACM, 2007.
- Renhe, M. C. **A stable tensor-based deflection model for controlled fluid simulations**. 2017. Tese de Doutorado - COPPE - Universidade Federal do Rio de Janeiro.

- Sato, S.; Dobashi, Y.; Iwasaki, K.; Yamamoto, T. ; Nishita, T. **Deformation of 2D flow fields using stream functions**. In: SIGGRAPH Asia 2014 Technical Briefs, p. 4. ACM, 2014.
- Stam, J. **Stable fluids**. In: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, p. 121–128. ACM Press/Addison-Wesley Publishing Co., 1999.
- Treuille, A.; McNamara, A.; Popovic, Z. ; Stam, J. Keyframe control of smoke simulations. **ACM Trans. Graph.**, v.22, n.3, p. 716–723, Jul 2003.
- Twigg, C. Catmull-rom splines. **Computer**, v.41, n.6, p. 4–6, 2003.
- Vieira, M. B.; Giraldo, G. A.; Amaral, A. ; Renhe, M. C. Anisotropic helmholtz decomposition for controlled fluid simulation. **Submetido**, 2020.
- Westin, C.-F. **A tensor framework for multidimensional signal processing**. 1994. Tese de Doutorado - Linköping University Electronic Press.
- Yaeger, L.; Upson, C. ; Myers, R. **Combining physical and visual simulation creation of the planet Jupiter for the film 2010**. In: Acm Siggraph Computer Graphics, volume 20, p. 85–93. ACM, 1986.
- Zhang, E.; Hays, J. ; Turk, G. Interactive tensor field design and visualization on surfaces. **IEEE Transactions on Visualization and Computer Graphics**, v.13, n.1, p. 94–107, 2006.