

**Ericsson de Souza Leal**

**Traçado de linhas de fluxo em modelos de  
reservatórios naturais de petróleo baseado em  
métodos numéricos adaptativos**

**Dissertação de Mestrado**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática do Departamento de Informática da PUC-Rio.

Orientador: Prof. Waldemar Celes Filho

Rio de Janeiro  
Setembro de 2015



**Ericsson de Souza Leal**

**Traçado de linhas de fluxo em modelos de  
reservatórios naturais de petróleo baseado em  
métodos numéricos adaptativos**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática do Departamento de Informática do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

**Prof. Waldemar Celes Filho**

Orientador

Departamento de Informática — PUC-Rio

**Prof. Hélio Côrtes Vieira Lopes**

Departamento de Informática — PUC-Rio

**Prof. Luiz Henrique de Figueiredo**

IMPA

**Prof. Marcelo Gattass**

Departamento de Informática — PUC-Rio

**Prof. José Eugenio Leal**

Coordenador Setorial do Centro Técnico Científico — PUC-Rio

Rio de Janeiro, 01 de setembro de 2015

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

### **Ericsson de Souza Leal**

Graduou-se em Engenharia da Computação pela Universidade Federal de Itajubá-MG. Foi bolsista do CNPq no programa Brazil-IP para formação de projetistas de circuitos integrados. Participou do programa BRAFITEC para intercâmbio acadêmico na França, onde realizou estágio no Centre Européen de Réalité Virtuelle (CERV). Após a graduação, entrou no programa de Mestrado em Informática da Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio). Durante o mestrado, atuou em projetos voltados à indústria do petróleo no Instituto Tecgraf.

#### Ficha Catalográfica

De Souza Leal, Ericsson

Traçado de linhas de fluxo em modelos de reservatórios naturais de petróleo baseado em métodos numéricos adaptativos / Ericsson de Souza Leal; orientador: Waldemar Celes Filho. — Rio de Janeiro : PUC-Rio, Departamento de Informática, 2015.

51 f: il. (color.); 29,7 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2015.

Inclui bibliografia.

1. Informática – Teses. 2. Visualização. 3. Campo Vetorial. 4. Linhas de Fluxo. 5. Adaptativo. 6. Reservatório de Petróleo. I. Celes Filho, Waldemar. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

## Agradecimentos

Ao meu orientador Professor Waldemar Celes, que orientou o desenvolvimento deste trabalho com paciência e dedicação.

Ao Tecgraf e aos colegas do Tecgraf, em especial, ao Bernardo Franceschin, cuja ajuda foi fundamental em certas etapas do desenvolvimento desse trabalho.

À minha mãe Margarida e meu irmão Edmilson que me deram suporte sob diferentes formas e me incentivaram a realizar este trabalho.

Ao CAPES e à PUC-Rio pelo auxílio financeiro e outros auxílios sem os quais não poderia realizar este trabalho.

Aos meus amigos e familiares que me estimularam transmitindo confiança e demonstrando satisfação por mim.

## Resumo

De Souza Leal, Ericsson; Celes Filho, Waldemar. **Traçado de linhas de fluxo em modelos de reservatórios naturais de petróleo baseado em métodos numéricos adaptativos**. Rio de Janeiro, 2015. 51p. Dissertação de Mestrado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Tradicionalmente, para visualização de campos vetoriais em modelos discretos de reservatórios naturais de petróleo, traça-se linhas de fluxo resolvendo a sua equação diferencial ordinária célula-a-célula, seja através de soluções analíticas ou numéricas, considerando o campo de velocidade local de cada célula. Essa estratégia tem como desvantagem traçar a linha considerando um campo de velocidade discreto e portanto descontínuo. Além disso, para modelos massivos, resolver a equação célula-a-célula pode tornar o método ineficiente. Neste trabalho, exploramos uma estratégia diferente: ao traçar as linhas de fluxo considera-se um campo de velocidade contínuo, representado pelo modelo discreto do reservatório. Para tanto, propõe-se: (i) o uso de uma estrutura espacial para acelerar a localização de um ponto no modelo de reservatório; (ii) o uso de interpolação esférica para avaliação do campo de velocidade a partir do modelo discreto; (iii) o uso de um método numérico adaptativo para controlar o erro numérico da integração. Os resultados obtidos em modelos de reservatórios reais demonstram que o método proposto atende aos requisitos de precisão, mantendo um bom desempenho.

## Palavras-chave

Visualização; Campo Vetorial; Linhas de Fluxo; Adaptativo; Reservatório de Petróleo;

## Abstract

De Souza Leal, Ericsson; Celes Filho, Waldemar (Advisor). **Streamline tracing for oil natural reservoirs based on adaptive numerical methods**. Rio de Janeiro, 2015. 51p. MSc. Dissertation — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Traditionally, streamlines in discrete models of natural oil reservoirs are traced by solving an ordinary differential equation in a cell-by-cell way, using analytical or numerical solutions, considering the local velocity of each cell. This strategy has a disadvantage: the streamline is traced considering a discrete, and so discontinuous, vector field. Furthermore, for massive models, to solve the equation in a cell-by-cell way may be inefficient. In this work, we explore a different strategy: the streamline tracing considers a continuous vector field represented by the discrete model. Therefore, we propose: (i) to use a spatial structure to speed up the point location process inside the reservoir model; (ii) to use spherical interpolation to obtain the velocity field from the discrete model; (iii) to use an adaptive numerical method to control the numerical error from the integration process. The results obtained for actual reservoir models demonstrate that the proposed method fulfills the precision requirements, keeping a good performance.

## Keywords

Visualization; Vector Field; Streamline; Adaptive; Oil Reservoir;

## Sumário

1	Introdução	11
2	Conceitos Básicos	14
2.1	Linhas de Fluxo	14
2.2	Métodos Numéricos	15
2.3	O Modelo de Reservatório	16
3	Trabalhos relacionados	18
3.1	Métodos Analíticos	18
3.2	Métodos Numéricos	19
4	Método Proposto	21
4.1	Localização de um ponto no modelo de reservatório	21
4.2	Suavização e Interpolação das Velocidades	25
4.3	Método Numérico Adaptativo para Resolução da Equação Diferencial	30
5	Testes e Resultados	34
5.1	Lançamento de sementes	34
5.2	Modelos para Testes	35
5.3	Resultados e Avaliação da Corretude	35
5.4	Avaliação de Desempenho	42
6	Conclusão	49
	Referências Bibliográficas	50

## Lista de figuras

Figura 1.1	Exemplo de uso das linhas de fluxo para visualização da distribuição do fluxo pelo reservatório de petróleo e análise da efetividade dos poços injetores. Esta imagem foi gerada pelo componente de Linhas de Fluxo do software GERESIM [1].	12
Figura 2.1	Modelo de reservatório organizado em uma grade tridimensional [2].	16
Figura 2.2	Corte de um modelo de reservatório de petróleo, onde se encontram uma região de falha à esquerda e uma região de células inativas à direita.	17
Figura 2.3	Corte de um modelo de reservatório natural de petróleo que apresenta células degeneradas.	17
Figura 4.1	Armazenamento de uma grade regular. O vetor C representa o vetor de células e L representa o vetor de listas.[3]	22
Figura 4.2	Transformação do espaço físico para o paramétrico [4].	23
Figura 4.3	Hexaedro e seus vértices numerados [5].	23
Figura 4.4	Transformação do fluxo na face para a velocidade no vértice	25
Figura 4.5	Mapeamentos Exponencial e Exponencial Inverso.	28
Figura 4.6	Interpolação Trilinear.	29
Figura 4.7	Comparação entre os métodos de interpolação <i>Lerp</i> e <i>Slerp</i> [6]: enquanto a interpolação do tipo <i>Lerp</i> divide a reta secante em partes iguais, com <i>Slerp</i> o grande arco é dividido em ângulos iguais.	30
Figura 4.8	Interpolação de vetores através de <i>Slerp</i> [6].	30
Figura 5.1	Modelo de Reservatório A.	35
Figura 5.2	Modelo de Reservatório B.	36
Figura 5.3	Modelo de Reservatório C.	36
Figura 5.4	Comparação entre as linhas geradas pelo método de referência, <i>Particle Tracking</i> , em (a) e pelo método Numérico Adaptativo, em (b)	37
Figura 5.5	Comparação entre os comprimentos das linhas geradas pelos dois métodos <i>Particle Tracking</i> e Método Proposto no modelo de reservatório A	38
Figura 5.6	Comparação entre as linhas geradas pelo Método Célula a Célula em (a) e pelo Método Proposto em (b) para o Modelo de Reservatório B	40
Figura 5.7	Comparação entre as linhas geradas pelo Método Célula a Célula em (a) e pelo Método Proposto em (b) enfatizando o efeito da suavização do campo de velocidades	41



Figura 5.8	Exemplo do efeito da suavização em duas células vizinhas. Se a célula A estiver na borda do modelo, as velocidades suavizadas em vermelho podem levar as linhas de fluxo que entrarem nessa célula a saírem do modelo	42
Figura 5.9	Comparação entre as linhas geradas pelo Método Célula a Célula em (a) e pelo Método Numérico Adaptativo em (b) para o Modelo de Reservatório C	43
Figura 5.10	Histogramas do número de chamadas da função derivada realizadas pelos Métodos: Célula a Célula em (a) e Adaptativo em (b) no modelo A	46
Figura 5.11	Histogramas do número de chamadas da função derivada realizadas pelos Métodos: Célula a Célula em (a) e Adaptativo em (b) no modelo B	47
Figura 5.12	Histogramas do número de chamadas da função derivada realizadas pelos Métodos: Célula a Célula em (a) e Adaptativo em (b) no modelo C	48

**Lista de tabelas**

Tabela 5.1	Resultados de tempo de execução para o Modelo A	44
Tabela 5.2	Resultados de tempo de execução para o Modelo B	44
Tabela 5.3	Resultados de tempo de execução para o Modelo C	45

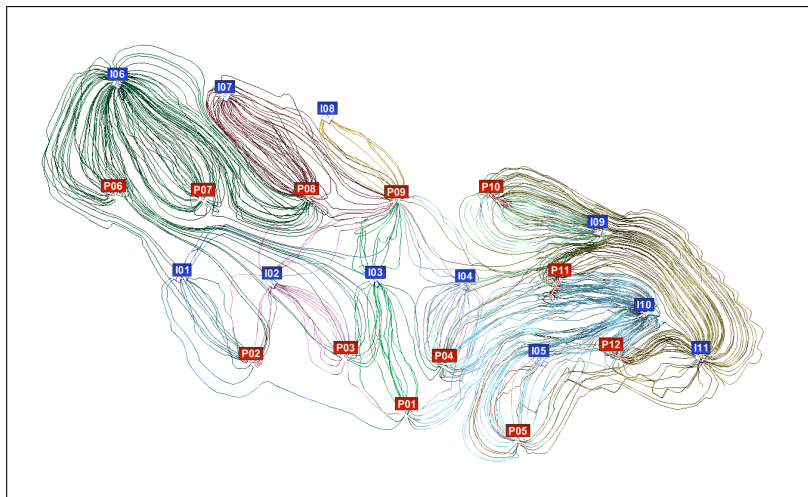
# 1

## Introdução

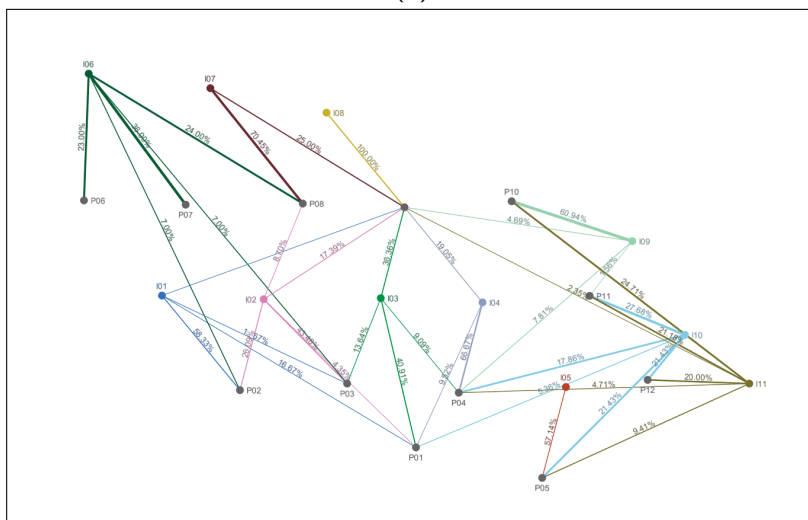
A técnica de traçado de linhas de fluxo é muito usada para a visualização de campos vetoriais em modelos discretos resultantes de simulações numéricas. Uma importante aplicação está na área de Visualização de Reservatórios Naturais de Petróleo, pois essa técnica, além de possibilitar visualizar o campo de velocidade do fluido resultante da simulação numérica do reservatório, permite avaliar a efetividade e contribuição de cada poço injetor. Um exemplo de tal aplicação está mostrado na Figura 1.1. Na Figura 1.1a estão as linhas de fluxo geradas como resultado do lançamento de 500 sementes distribuídas pelos poços injetores; na Figura 1.1b, o grafo que representa a distribuição do fluxo pelo reservatório, onde existe uma aresta a cada par injetor-produtor que possui pelo menos uma linha de fluxo. Associada a cada aresta desse grafo está a porcentagem de linhas que saíram do poço injetor e seguiram o caminho representado por tal aresta. Com base nesse grafo é possível analisar a efetividade de um poço injetor e entender sua contribuição na exploração do campo.

Os métodos tradicionais para traçado de linhas de fluxo (ou linhas de corrente), em modelos discretos, adotam a abordagem célula-a-célula para fazer a integração de cada linha. Nessa abordagem, o traçado da linha é feito localmente dentro de cada célula, considerando o campo de velocidade local da célula. Isso facilita a avaliação da velocidade ao longo da linha e minimiza os erros numéricos na integração de um segmento de linha devido à pequena variação do campo de velocidade existente no domínio de uma célula. Entretanto, esses métodos tradicionais têm o viés de fazer o traçado das linhas a partir de um campo de velocidade que é descontínuo. Além disso, o tratamento célula-a-célula acarreta a interrupção do traçado de diversas linhas quando passam por uma região de células de geometria complexa, com falhas (descontinuidade geométrica) e células inativas.

Os métodos que adotam a abordagem célula-a-célula também não apresentam bom desempenho em modelos massivos. É comum existir regiões que apresentam pequena variação do campo de velocidade em modelos massivos, e isto indica que o erro introduzido no processo de integração da linha deve



(a)



(b)

Figura 1.1 – Exemplo de uso das linhas de fluxo para visualização da distribuição do fluxo pelo reservatório de petróleo e análise da efetividade dos poços injetores. Esta imagem foi gerada pelo componente de Linhas de Fluxo do software GERESIM [1].

ser pequeno. Entretanto, o método célula-a-célula não otimiza o traçado das linhas nessas regiões. De fato, os métodos célula-a-célula não adotam nenhum mecanismo para controle do erro numérico da integração. Como consequência, esses métodos realizam um número de iterações maior do que seria necessário para assegurar uma precisão numérica adequada.

Este trabalho tem por objetivo oferecer um método para traçado de linhas de fluxo para visualização de campos de velocidades simulados em um reservatório discretizado por células hexaédricas irregulares (*corner-points*), e naturalmente também sendo válido para modelos do tipo *bloco centrado* (hexaedros regulares). A ideia principal do método proposto é interpretar o domínio discretizado como contínuo. Para tratar o domínio como contínuo, deve ser possível avaliar o campo de velocidade em qualquer localização do

modelo. Para alcançar tal objetivo, propomos:

1. Mapear as velocidades locais para o sistema global, ou seja, transformar o campo vetorial discreto por célula em um campo com representação por vértice. Para tanto, é necessário realizar uma etapa de suavização do campo de velocidade eliminando as descontinuidades. A velocidade suavizada é calculada através de *média esférica de vetores* [7].
2. Determinar, eficientemente, a localização de um ponto  $(x, y, z)$  qualquer em relação ao modelo discreto. Para isso, empregamos o uso de uma grade regular como estrutura espacial de aceleração [3].
3. Fazer a aquisição da velocidade em qualquer ponto do domínio através de uma interpolação linear esférica [6], apropriada para interpolação de vetores.

A partir desta ideia, o método realiza um traçado de linhas sem a restrição dos métodos célula-a-célula, que exigem que cada segmento esteja contido em uma célula. Para garantir o controle do erro numérico, propomos a utilização de um método numérico adaptativo.

O algoritmo proposto é testado considerando modelos reais de reservatório natural de petróleo. Os resultados demonstram que são geradas linhas de fluxo corretas e com bom desempenho. A vantagem do método proposto é expressiva quando são considerados modelos com regiões de falhas e descontinuidade (células inativas).

Esta dissertação está organizada da seguinte forma: o Capítulo 2 descreve alguns conceitos básicos relacionados com esta pesquisa; o Capítulo 3 descreve alguns trabalhos importantes na área de traçado de linhas de fluxo em reservatórios naturais de petróleo; o Capítulo 4 apresenta o método proposto; o Capítulo 5 apresenta os testes realizados e os resultados obtidos; finalmente, o Capítulo 6 conclui o trabalho salientando suas contribuições.

## 2

## Conceitos Básicos

Neste capítulo, são apresentados conceitos necessários para contextualizar o trabalho e facilitar a compreensão do mesmo.

### 2.1

### Linhas de Fluxo

#### Definição

Uma linha de fluxo pode ser entendida como a trajetória de uma partícula ao soltar-se em um campo de velocidade. Uma linha de fluxo é definida pela equação diferencial:

$$\frac{ds}{dt} = v \quad (2.1)$$

onde  $s$  representa o comprimento de arco,  $t$  o tempo e  $v$  a velocidade do fluido. Uma interpretação dessa equação é que a linha de fluxo em toda sua extensão é tangente ao campo de velocidade ali presente. No contexto de reservatório natural de petróleo,  $v$  é a soma das velocidades de Darcy ( $q$ ) das 3 fases do fluido (óleo, gás, água) dividida pela porosidade do meio ( $\phi$ ):

$$v = \frac{q}{\phi} = \frac{q_o + q_g + q_w}{\phi} \quad (2.2)$$

A equação diferencial deve ser resolvida para se encontrar a trajetória da linha, e existem diferentes abordagens para tal solução. A maior parte dos métodos propostos até hoje na área de reservatório de petróleo se baseia no *Particle Tracking* proposto por Pollock [8], que resolve a equação de forma analítica para células regulares. Entretanto, outros métodos optam pela solução numérica da equação diferencial, como veremos.

## Tempo de voo

O tempo de voo é definido como o tempo que uma partícula leva para “viajar” de um ponto até o outro, sendo expresso por:

$$\tau(s) = \int_{s_0}^s \frac{\phi}{|q|} ds \quad (2.3)$$

No contexto de traçado de linhas de fluxo em reservatórios de petróleo, o tempo de voo, em geral, é medido em dias devido às grandes dimensões dos reservatórios.

## 2.2

### Métodos Numéricos

Neste trabalho, empregamos o método de Runge–Kutta para resolução da equação diferencial ordinária. Os métodos de Runge–Kutta se caracterizam por fazer diferentes avaliações da função derivada ao longo do intervalo de integração, diferente do método de Euler que faz apenas uma avaliação da derivada no início do intervalo. A cada avaliação é feito o produto com o passo de integração que gera uma constante  $k_i$  que pode ser entendida como uma estimativa do novo ponto. No fim, um novo ponto da linha é encontrado através da média ponderada dessas estimativas  $k'_i$ s. Os métodos Runge–Kutta apresentam diferentes ordens que estão relacionadas diretamente com o número de avaliações no intervalo. Como exemplo, o Runge–Kutta de quarta ordem faz quatro avaliações da derivada dentro de cada intervalo de integração. O tradicional método de Runge–Kutta de quarta ordem é dado pelas expressões de  $k_i$  e a expressão do novo ponto que estão descritas abaixo :

$$\begin{aligned} k_1 &= f(t_n, y_n); \\ k_2 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right); \\ k_3 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right); \\ k_4 &= f(t_n + h, y_n + hk_3) \end{aligned}$$

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

onde  $h$  representa o passo de integração e  $y_n$  a aproximação da posição da

partícula na iteração  $n$ . Métodos numéricos adaptativos permitem avaliar o erro numérico em um passo de integração e então adaptá-lo para atender uma determinada tolerância pré-estabelecida.

## 2.3

### O Modelo de Reservatório

Os modelos de reservatórios naturais de petróleo aos quais se direcionam este trabalho se caracterizam como modelos discretizados em malhas de células hexaédricas irregulares (modelos *corner points*). As células dessa malha estão organizadas topologicamente em uma grade tridimensional, onde cada célula é identificada por uma tripla  $[i, j, k]$ . A Figura 2.1 mostra a grade topológica utilizada em tais modelos. A geometria associada a essa grade pode ser bastante irregular como mostrado na Figura 2.2 devido a descontinuidades na elevação, caracterizando falhas geológicas, ou seja, algumas células vizinhas na topologia da grade não compartilham faces [2].

Além disso, algumas células podem ser inativas e outras degeneradas. Células inativas são aquelas que não contribuem com volumes de fluido ou fluxo no reservatório. Células degeneradas (*pinch-out*) são aquelas que possuem vértices coincidentes. A Figura 2.3 exemplifica um modelo de reservatório com células degeneradas em destaque.

Esses modelos podem chegar a dezenas de milhões de células, o que os caracterizam como modelos massivos.

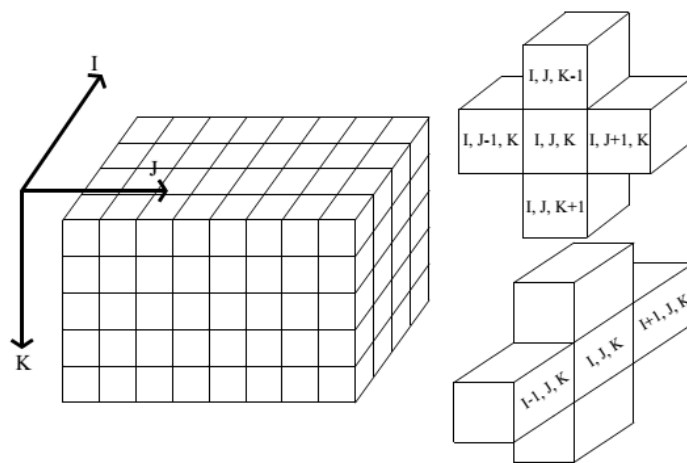


Figura 2.1 – Modelo de reservatório organizado em uma grade tridimensional [2].

O campo de velocidade resultante da simulação numérica de reservatório de petróleo é representado pelo valor do fluxo nas faces de cada célula. Este valor de fluxo ( $F$ ), dividido pela área da face associada, representa a velocidade de Darcy:  $q = \frac{F}{A}$ .



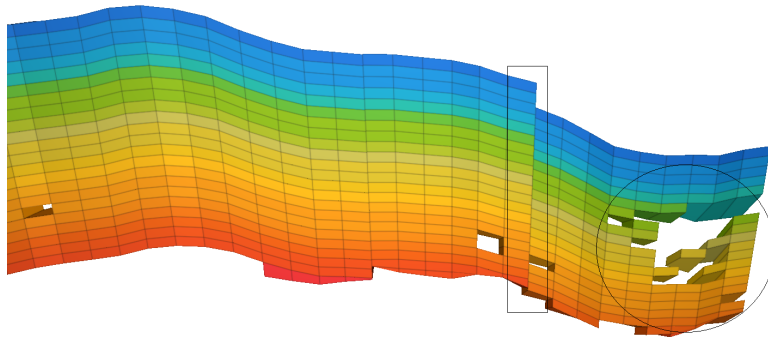


Figura 2.2 – Corte de um modelo de reservatório de petróleo, onde se encontram uma região de falha à esquerda e uma região de células inativas à direita.

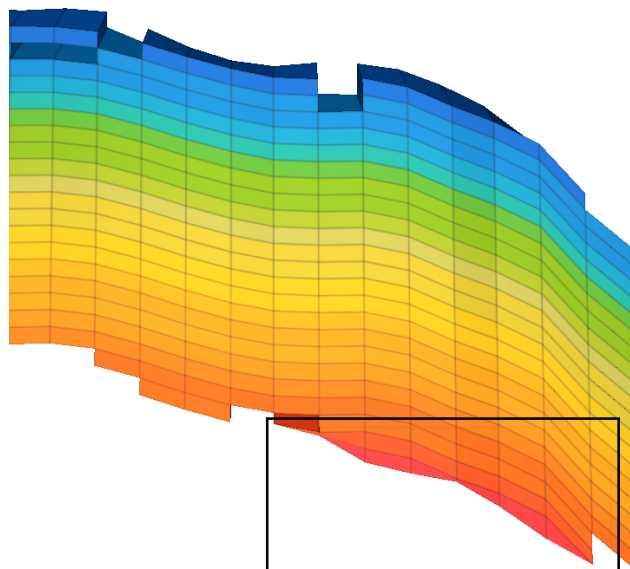


Figura 2.3 – Corte de um modelo de reservatório natural de petróleo que apresenta células degeneradas.

## 3

### Trabalhos relacionados

#### 3.1

##### Métodos Analíticos

Os métodos para traçado de linhas de fluxo em modelos discretos de reservatório de petróleo têm se baseado numa abordagem célula-a-célula, ou seja, traçar um segmento de linha de fluxo a cada célula do domínio que a linha entrar. Muitos métodos foram baseados no método *Particle Tracking* proposto por Pollock [8].

Pollock [8] propôs o método semi-analítico para traçado de linhas de fluxo. No método *Particle Tracking*, cada célula tem as componentes de velocidade do fluido definidas em suas faces, e a partir dessas velocidades o método faz uma interpolação linear para obter a velocidade em qualquer ponto. Essa interpolação linear permite que dentro de cada célula um segmento de linha seja integrado de forma analítica. O uso desta expressão analítica evita a necessidade de sucessivos cálculos como nos métodos que fazem uso de métodos numéricos. Assim, a partir do ponto de entrada na célula é possível encontrar o ponto por onde a linha sai da célula diretamente através das expressões analíticas. Esse método é restrito a modelos discretizados em bloco centrados (células hexaédricas regulares), não sendo diretamente aplicado a domínios de células irregulares (*corner points*).

Prevost et al. [9] propuseram o método *Standard Flux Mapping* que estende o *Particle Tracking* para domínio de células irregulares. Para isso, cada célula irregular é mapeada em uma célula regular e unitária através de uma transformação isoparamétrica. Nesse método, as velocidades são obtidas a partir do fluxo do fluido definido nas faces. Apesar do avanço em relação ao tipo de domínio suportado, ele apresenta problema quando o fluxo é uniforme em uma célula, ou seja, quando o fluxo que entra por uma face é igual ao fluxo que sai da face diretamente oposta. Devido ao mapeamento isoparamétrico, o fluxo resultante no espaço físico perde uniformidade.

Haegland et al. [10] propõem o método *Corner Velocity Interpolation* que traça linhas de fluxo em domínios de células irregulares de hexaedros.

A característica fundamental desse método é que ele, ao invés de interpolar a velocidade diretamente a partir do fluxo nas faces como no *Standard Flux Mapping*, interpola diretamente a partir das velocidades nos vértices da célula. Essa característica permitiu a reprodução de fluxos uniformes que nos métodos mencionados anteriormente não podiam ser reproduzidos [10].

Uma preocupação constante nessa área é minimizar os erros que aparecem ao longo do processo desde a simulação até a visualização de um campo de velocidade por linhas de fluxo. Jimenez et al. [11] realizaram um estudo sobre os diferentes tipos de erros que existem nesse processo. Primeiramente é examinado o que Jimenez et al. [11] chamam de erros espaciais transversais, por exemplo o erro resultante da etapa de lançamento das sementes das linhas de fluxo. Outro erro espacial é causado pela ocorrência de regiões de falha onde a falta de continuidade do fluxo nas faces leva a erros na trajetória da linha. É detectado, também, um erro espacial longitudinal devido ao cálculo impreciso do tempo de voo das linhas obtidas nos algoritmos em modelos com células irregulares.

### 3.2

#### Métodos Numéricos

Ueng et al. [12] propôs um método numérico para resolução da equação diferencial da linha de fluxo, ao invés de fazer a integração analítica da linha. Ele utilizou o método numérico de Runge-Kutta de quarta ordem sem adaptação do passo de integração. O método aplica-se a domínios de células tetraédricas regulares ou irregulares, mas também pode ser usado com células hexaédricas. Os cálculos são feitos em um sistema de coordenadas canônico ao invés do sistema físico para aumentar o desempenho, fazendo a devida transformação de coordenadas. Cada segmento de linha fica restrito ao domínio de uma célula, portanto a integração é do tipo célula-a-célula. Desse modo, observa-se que é simples a localização de um ponto no modelo, já que um novo vértice calculado estará localizado dentro da mesma célula ou em uma célula adjacente. Nesse mesmo trabalho de Ueng et al. [12] encontra-se a construção de fitas e tubos que são as extensões em duas e três dimensões, respectivamente, das linhas de fluxo. Elas fornecem as informações adicionais do campo vetorial: taxa de rotação e divergência.

O trabalho de Ueng et al. [12] foi uma das fontes de inspiração para a realização deste trabalho pelo fato de usar um método numérico não-adaptativo para integração da linha de fluxo. Portanto, uma versão do método proposto de Ueng et al. [12] adaptada a modelos discretizados em células hexaédricas será utilizado como parâmetro de comparação na seção de testes e resultados.

O método proposto neste trabalho também usa um método numérico para a resolução da equação diferencial, mas o método usado é do tipo adaptativo e trata o modelo discreto como um campo contínuo de velocidades.

## 4

### Método Proposto

O método desenvolvido tem como ideia principal propôr que as linhas sejam traçadas a partir de um campo de velocidade contínuo, representado por um modelo discreto de reservatório. Para viabilizar essa ideia, propõe-se dividir o método em três tarefas principais: (i) a localização de um ponto no modelo de reservatório; (ii) interpolação do tipo esférica para avaliação do campo de velocidade a partir do modelo discreto; (iii) utilização de um método numérico adaptativo para controlar o erro numérico da integração.

#### 4.1

##### Localização de um ponto no modelo de reservatório

Para tornar rápida a localização de um ponto no modelo do reservatório, utiliza-se nessa etapa uma estrutura espacial de localização: a grade regular uniforme. Essa estrutura de grade para modelos de reservatórios de petróleo foi proposta por Franceschin [3] e utilizada neste trabalho.

##### 4.1.1

##### Grade Regular Uniforme

A grade regular uniforme é uma estrutura de subdivisão espacial. A ideia básica da grade consiste na subdivisão da caixa envolvente alinhada aos eixos (AABB) do modelo em *voxels* de dimensões iguais, mas a subdivisão pode ser feita em tamanhos diferentes para cada eixo. Para cada *voxel* da grade, é construída uma lista de células do modelo de reservatório. As células da lista de um determinado *voxel* da grade consistem nas células do modelo que se sobrepõem a esse *voxel*. Dado um ponto, esta estrutura possibilita o acesso direto à lista de células do *voxel* da grade que contém o ponto. Em seguida, para descobrir em qual célula o ponto está contido, basta realizar um teste de intersecção; no pior caso, para cada uma das células da lista [3]. As listas de células são guardadas em um vetor de índices das células, como ilustrado na Figura 4.1.

Uma questão importante tanto para o desempenho quanto para o gasto de memória da grade regular é a escolha de uma resolução adequada. Como os

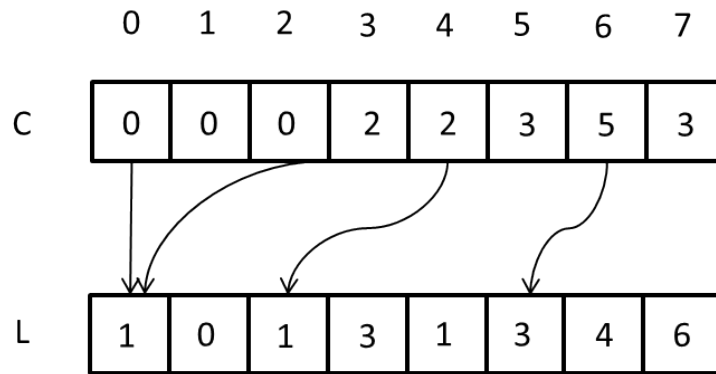


Figura 4.1 – Armazenamento de uma grade regular. O vetor C representa o vetor de células e L representa o vetor de listas.[3]

modelos de reservatórios de petróleo têm suas células organizadas topologicamente em uma grade tridimensional, tiramos proveito das dimensões da grade topológica do modelo para o cálculo da resolução. As dimensões da grade  $n_x$ ,  $n_y$  e  $n_z$  são calculadas como múltiplos das dimensões da grade topológica  $n_i$ ,  $n_j$  e  $n_k$ .

$$n_x = kn_i, \quad n_y = kn_j, \quad n_z = kn_k \quad (4.1)$$

onde  $k$  representa o fator que determina a densidade da grade.

#### 4.1.2

##### Coordenadas paramétricas

Um grande desafio é o cálculo para determinar se uma dada célula contém um ponto. Como as células são irregulares, com faces não planares e podendo ser degeneradas, não existe um procedimento analítico para a localização de um ponto no espaço físico  $(x, y, z)$  no interior da célula. Além disso, após a localização do ponto, como veremos, será necessário avaliar o campo de velocidade neste ponto.

Para a realização destes procedimento, converte-se as células do espaço físico  $(x, y, z)$  para o espaço paramétrico  $(\xi, \eta, \mu)$ . O resultado desta transformação de espaços é mostrado na Figura 4.2.

As coordenadas do ponto no espaço paramétrico  $(\xi, \eta, \mu)$ , denominadas coordenadas naturais, variam de  $-1$  a  $+1$  de uma face a outra, tendo valor zero na face “mediana” [5]. Os oito vértices do hexaedro são numerados como mostrado na Figura 4.3.

O mapeamento das coordenadas paramétricas de um ponto qualquer do hexaedro para as coordenadas físicas é dado pelas funções de forma (*shape functions*),  $N_i$ , para hexaedros:

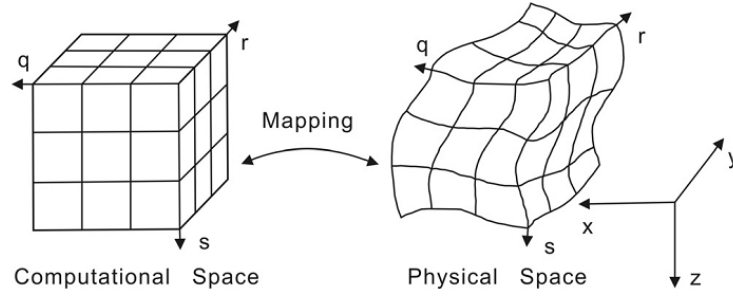


Figura 4.2 – Transformação do espaço físico para o paramétrico [4].

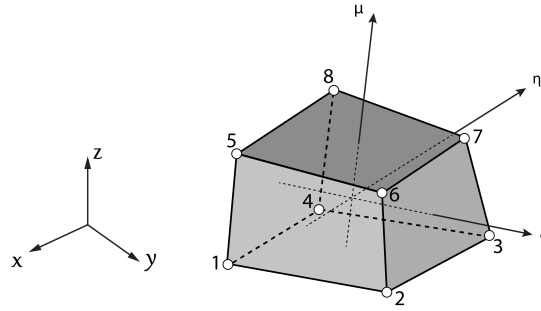


Figura 4.3 – Hexaedro e seus vértices numerados [5].

$$\begin{aligned}
 N_1 &= \frac{1}{8}(1 - \xi)(1 - \eta)(1 - \mu) \\
 N_2 &= \frac{1}{8}(1 - \xi)(1 + \eta)(1 - \mu) \\
 N_3 &= \frac{1}{8}(1 + \xi)(1 + \eta)(1 - \mu) \\
 N_4 &= \frac{1}{8}(1 + \xi)(1 - \eta)(1 - \mu) \\
 N_5 &= \frac{1}{8}(1 - \xi)(1 - \eta)(1 + \mu) \\
 N_6 &= \frac{1}{8}(1 - \xi)(1 + \eta)(1 + \mu) \\
 N_7 &= \frac{1}{8}(1 + \xi)(1 + \eta)(1 + \mu) \\
 N_8 &= \frac{1}{8}(1 + \xi)(1 - \eta)(1 + \mu)
 \end{aligned}$$

As coordenadas físicas (ou qualquer outro atributo escalar) no interior da célula podem ser obtidas a partir dos valores associados aos vértices ( $v_i$ ). Em particular, as coordenadas físicas no interior da célula são expressas pelas funções:

$$\begin{aligned}
 x &= f(\xi, \eta, \mu) = \sum_{i=1}^8 N_i x_i \\
 y &= g(\xi, \eta, \mu) = \sum_{i=1}^8 N_i y_i \\
 z &= h(\xi, \eta, \mu) = \sum_{i=1}^8 N_i z_i
 \end{aligned}$$

O sistema de equações resultante do mapeamento de coordenadas é do tipo não-linear e para resolver este tipo de sistema utiliza-se aqui um método numérico e neste caso é o Método de Newton-Raphson.

#### 4.1.3

##### Método de Newton-Raphson

O problema existe quando se deseja o mapeamento inverso: dado um ponto no espaço físico  $(x, y, z)$ , qual a coordenada paramétrica  $(\xi, \eta, \mu)$  correspondente? Recai-se num sistema não linear. Uma forma de resolver este problema é transformá-lo num problema de determinação de raízes simultâneas, e daí usar o método de Newton-Raphson. As funções cujas raízes  $(\xi, \eta, \mu)$  queremos determinar são:

$$\begin{aligned}
 u(\xi, \eta, \mu) &= x - f(\xi, \eta, \mu) = 0 \\
 v(\xi, \eta, \mu) &= y - g(\xi, \eta, \mu) = 0 \\
 w(\xi, \eta, \mu) &= z - h(\xi, \eta, \mu) = 0
 \end{aligned}$$

O método de Newton-Raphson para determinação de raízes simples é:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

No caso do elemento hexaédrico, trabalhamos com funções de três variáveis e, portanto, a derivada da função é expressa pela matriz Jacobiana:

$$J = \begin{bmatrix} \frac{\partial u}{\partial \xi} & \frac{\partial u}{\partial \eta} & \frac{\partial u}{\partial \mu} \\ \frac{\partial v}{\partial \xi} & \frac{\partial v}{\partial \eta} & \frac{\partial v}{\partial \mu} \\ \frac{\partial w}{\partial \xi} & \frac{\partial w}{\partial \eta} & \frac{\partial w}{\partial \mu} \end{bmatrix}$$



E a iteração de Newton-Raphson, para o nosso problema, é expressa por:

$$\begin{bmatrix} \xi_{i+1} \\ \eta_{i+1} \\ \mu_{i+1} \end{bmatrix} = \begin{bmatrix} \xi_i \\ \eta_i \\ \mu_i \end{bmatrix} - J^{-1} \begin{bmatrix} u(\xi_i, \eta_i, \mu_i) \\ v(\xi_i, \eta_i, \mu_i) \\ w(\xi_i, \eta_i, \mu_i) \end{bmatrix}$$

## 4.2

### Suavização e Interpolação das Velocidades

Os modelos de reservatório natural de petróleo utilizados neste trabalho apresentam o campo de velocidade discretizado como fluxo na face de cada célula hexaédrica. Para fazer a suavização e interpolação das velocidades é necessário transformar os fluxos nas faces para velocidades nos vértices. Assim como é feito no método CVI (*corner velocity interpolation*) proposto por Haegland et al. [10], para cada célula obtemos as velocidades nos vértices resolvendo os sistemas de equações lineares  $3 \times 3$  :

$$\begin{cases} \mathbf{v}_i \cdot \mathbf{n}_{xi} = F_{xi} \\ \mathbf{v}_i \cdot \mathbf{n}_{yi} = F_{yi}, \\ \mathbf{v}_i \cdot \mathbf{n}_{zi} = F_{zi} \end{cases} \quad i = 1, 2, \dots, 8 \quad (4.2)$$

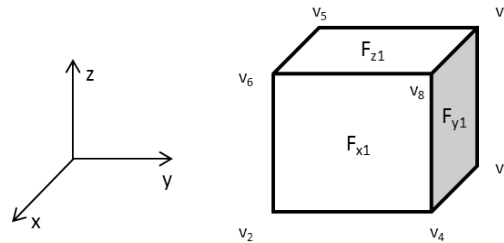


Figura 4.4 – Transformação do fluxo na face para a velocidade no vértice

### 4.2.1

#### Suavização do campo de velocidade

O método desenvolvido tem como ideia principal propôr que as linhas de fluxo sejam traçadas a partir de um campo de velocidade contínuo, representado por um modelo discreto do reservatório. Para obter esse contínuo é necessário eliminar as descontinuidades do campo de velocidade, e isso é feito através da suavização das velocidades.

A suavização do campo de velocidade pode ser feita através de uma simples média das contribuições das células vizinhas que compartilham o vértice. Essa abordagem não é a mais recomendada já que o dado em si é

do tipo vetorial. Deste modo, faz-se uso, aqui, da média esférica[7] de vetores que é descrita a seguir.

### Média Esférica Ponderada

A média esférica foi elaborada originalmente para obtenção da média de pontos que estão sobre a superfície de uma esfera. Neste trabalho, a média esférica é utilizada para obter a média de vetores, considerando que a média de pontos sobre uma esfera de raio unitário é equivalente à média dos vetores normalizados que tem a origem da esfera como um ponto em comum. A magnitude dos vetores será usada como ponderação no cálculo da média.

No Espaço Euclidiano, a média ponderada  $\sum_i w_i \hat{p}_i$  é igual ao vetor  $\hat{q}$  onde a função

$$f(\hat{q}) = \sum w_i ||\hat{q} - \hat{p}_i||^2 \quad (4.3)$$

é minimizada, onde  $w_i$  representa a ponderação do vetor  $\hat{p}_i$

Analogamente, define-se a média esférica ponderada pela Equação 4.4, onde  $dist_S(\hat{q}, \hat{p}_i)$  é a distância geodésica entre o vetor  $\hat{p}$  e o ponto  $\hat{q}$  na esfera  $S$ .

$$f(\hat{q}) = \frac{1}{2} \sum_i w_i dist_S(\hat{q}, \hat{p}_i) \quad (4.4)$$

A média esférica é igual ao vetor  $\hat{q}$  na esfera  $S$  que minimiza  $f(\hat{q})$  [7].

### Algoritmo da Média Esférica Ponderada

O algoritmo da média esférica ponderada é um processo iterativo onde em cada iteração é calculada uma nova estimativa da média. Primeiro, a estimativa inicial é a média aritmética dos vetores. A partir daí o algoritmo realiza, para cada vetor  $\hat{p}_i$ , um mapeamento para o plano tangente do ponto da esfera onde se encontra a estimativa  $\hat{q}$ . Esse mapeamento recebe o nome de mapeamento exponencial inverso (*InverseMap*). A função que representa este mapeamento inverso está mostrada no Algoritmo 1.

**Algoritmo 1** Mapeamento Exponencial Inverso**Entrada:** Vetores  $\hat{q}$  e  $\hat{p}_i$ Pesos positivos  $w_1, \dots, w_n$ **Saída:** O vetor  $\hat{p}_i$  no plano tangente**início**

$$\theta = \cos^{-1}(\hat{q}_0 \cdot \hat{p}_i)$$

$$\hat{t} = \hat{p}_i \times \hat{q}_0$$

$$\hat{d} = \hat{q}_0 \times \hat{t}$$

$$p_i^* = \hat{q}_0 + \theta \hat{d}$$

**fim****retorna**  $p_i^*$ 

Após o cálculo da nova estimativa no plano tangente, o algoritmo realiza o mapeamento da estimativa de volta à esfera; este recebe o nome de mapeamento exponencial (*Map*). A função que representa este mapeamento está mostrada no Algoritmo 2.

**Algoritmo 2** Mapeamento Exponencial**Entrada:** Vetores  $\hat{q}_0$  e  $\hat{q}$ **Saída:** O vetor  $\hat{p}_i$  no plano tangente**início**

$$\theta = \hat{q}_0 \cdot \hat{q}$$

$$\hat{t} = \hat{q}_0 \times \hat{q}$$

$$q^* = \cos(\theta) \cdot \hat{q} + \sin(\theta) \cdot (\hat{t} \times \hat{q}) + (1 - \cos(\theta))(\hat{t} \cdot \hat{q})\hat{t}$$

**fim****retorna**  $q^*$ 

Os mapeamentos exponencial e exponencial inverso são ilustrados na Figura 4.5. O processo termina quando a diferença entre a estimativa atual e anterior for próxima de zero.

O Algoritmo 3 apresenta o procedimento que calcula a média esférica ponderada aplicada no cálculo das velocidades suavizadas nos vértices das células.

**4.2.2****Interpolação da Velocidade**

A abordagem utilizada neste trabalho é apropriada para se fazer interpolação de grandezas vetoriais é a interpolação trilinear esférica.

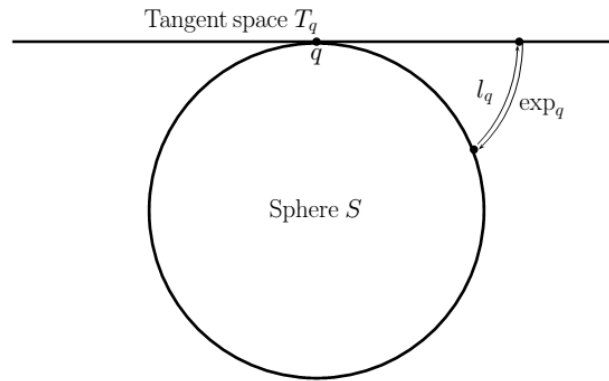


Figura 4.5 – Mapeamentos Exponencial e Exponencial Inverso.

**Algoritmo 3** Média Esférica Ponderada**Entrada:** Pontos  $p_1, \dots, p_n$  on  $S^d$ Pesos positivos:  $w_1, \dots, w_n$ **Saída:** A média esférica ponderada das entradas**início**Inicialização:  $q := \sum_{i=1}^n w_i p_i / \|\sum_{i=1}^n w_i p_i\|$ **enquanto**  $q_{new} \neq q$  **faça**  **para** cada  $i \in n$  **faça**     $p_i^* := InverseMap(p_i)$  u  $:= \sum_{i=1}^n w_i (p_i^* - q)$      $q_{new} := Map(q + u)$   **fim****fim****fim****Interpolação Trilinear**

Fazer a interpolação linear entre dois pontos em uma dimensão é simples. Se temos dois valores  $v_0$  e  $v_1$ , a interpolação linear entre os dois pontos é feita através da Equação 4.5.

$$v_x = (1 - f_x)v_0 + f_x v_1 \quad (4.5)$$

Em três dimensões, a interpolação é feita entre oito pontos  $v_{000}$ ,  $v_{001}$ ,  $v_{010}$ ,  $v_{011}$ ,  $v_{100}$ ,  $v_{101}$  e  $v_{111}$ . Uma abordagem eficiente é dividir o processo em três etapas. Essa abordagem requer sete interpolações do tipo linear usando interpolações em cascata [13]. As interpolações são feitas na sequência mostrada na Figura 4.6.

Primeiro interpola-se linearmente os quatro pares de vértices que compartilham aresta na direção  $x$  e assim obtém-se quatro pontos “intermediários” que se situam ao longo da aresta:  $v_{00}$ ,  $v_{01}$ ,  $v_{10}$ ,  $v_{11}$ . Na segunda etapa,

interpola-se linearmente os 2 pares de vértices que estão opostos na direção  $y$ , e obtém-se 2 pontos:  $v_0$  e  $v_1$ . Por último, interpola-se linearmente os dois pontos e encontra-se o valor  $v$  resultante.

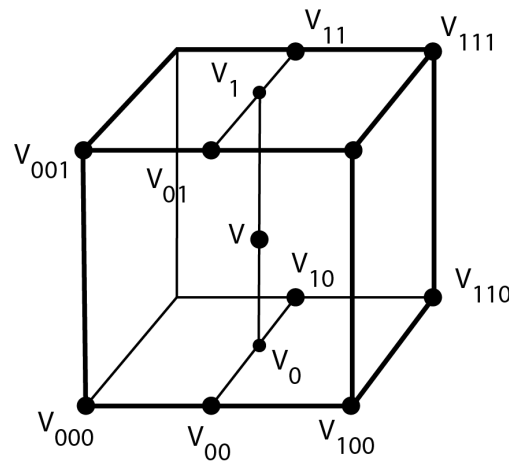


Figura 4.6 – Interpolação Trilinear.

$$v_{00} = v_{000} + f_x(v_{100} - v_{000})$$

$$v_{01} = v_{001} + f_x(v_{101} - v_{001})$$

$$v_{10} = v_{010} + f_x(v_{110} - v_{010})$$

$$v_{11} = v_{011} + f_x(v_{111} - v_{011})$$

$$v_0 = v_{00} + f_x(v_{10} - v_{00})$$

$$v_1 = v_{01} + f_x(v_{11} - v_{01})$$

$$v = v_0 + f_x(v_1 - v_0)$$

(4.6)

### Interpolação Esférica

Para obter uma interpolação mais precisa dos vetores, fazemos aqui uma interpolação do tipo esférica, operação também conhecida como *Slerp*, ao invés de uma simples interpolação linear componente a componente. Esse tipo de interpolação é introduzida para interpolar quatérnios, mas nesse contexto será utilizada para interpolar vetores.

A simples interpolação linear, também conhecida como *Lerp*, determina uma reta secante entre os dois quatérnios como mostrado na Figura 4.7. Portanto, a função de interpolação tem uma velocidade angular maior no meio

da curva. Uma ideia óbvia é definir um método de interpolação que produza a mesma curva de interpolação mas onde os quatérnios interpolados sejam unitários. Ao invés de fazer uma simples interpolação linear, a curva deve seguir um grande arco na esfera de quatérnio unitário, saindo de um quatérnio até o outro. Isto é chamado de *Great Arc of Interpolation* ou *Spherical Linear Interpolation - Slerp* [6].

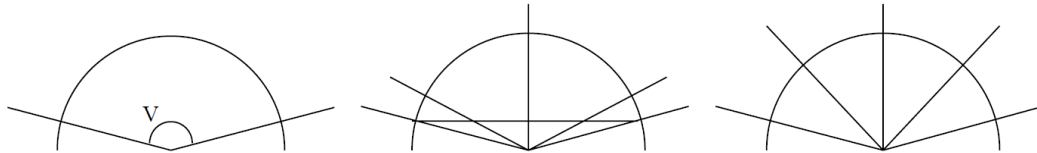


Figura 4.7 – Comparação entre os métodos de interpolação *Lerp* e *Slerp* [6]: enquanto a interpolação do tipo *Lerp* divide a reta secante em partes iguais, com *Slerp* o grande arco é dividido em ângulos iguais.

Nos termos da geometria diferencial, esse grande arco é uma curva geodésica, correspondente a uma linha estreita. O *Slerp* não apenas segue esse arco como também segue o menor arco. Portanto, *Slerp* permite o menor caminho de interpolação entre dois quatérnios na esfera unitária. Todas características do *Slerp* otimizam a curva de interpolação entre dois quatérnios. A Figura 4.8 mostra um exemplo simples de interpolação de vetores usando *Slerp* para se obter o vetor que representa a bissetriz do arco formado por dois vetores.

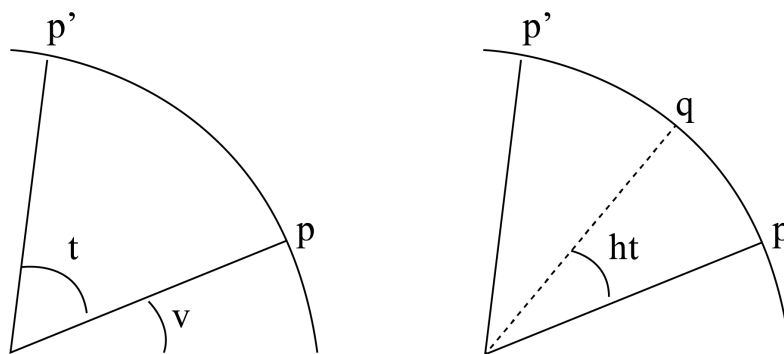


Figura 4.8 – Interpolação de vetores através de *Slerp* [6].

### 4.3

#### Método Numérico Adaptativo para Resolução da Equação Diferencial

O método numérico utilizado para resolução da equação diferencial da linha de fluxo é do tipo Runge-Kutta Adaptativo.

### 4.3.1

#### Método Adaptativo

Os métodos adaptativos têm como característica a atualização do tamanho do passo de integração a cada iteração. Deste modo, os métodos de Runge–Kutta podem ser modificados para incorporar esse passo adaptativo no seu algoritmo. O passo vai ser modificado de acordo com o erro local estimado a cada iteração. O método Runge–Kutta Adaptativo utilizado neste trabalho é o de Dormand–Prince [14]. Ele é um método que produz dois resultados a cada iteração,  $y_{n+1}$  e  $y_{n+1}^*$ , sendo que o primeiro é de ordem quatro e o segundo é de ordem cinco. Isto significa dizer que o erro cometido pelo primeiro resultado é da ordem  $\mathcal{O}(h^5)$  e o segundo resultado de ordem  $\mathcal{O}(h^6)$ , onde  $h$  é o passo de integração. Esse segundo resultado de ordem cinco é utilizado para estimar o erro do primeiro resultado, fazendo a simples diferença dos dois resultados. Portanto, tem-se no método de Dormand–Prince um resultado de quarta ordem com estimativa de erro de ordem cinco. As equações das constantes  $k_i$  estão descritas abaixo :

$$\begin{aligned} k_1 &= hf(x_n, y_n); \\ k_2 &= hf(x_n + \frac{1}{5}h, y_n + \frac{1}{5}k_1); \\ k_3 &= hf(x_n + \frac{3}{10}h, y_n + \frac{3}{40}k_1 + \frac{9}{40}k_2); \\ k_4 &= hf(x_n + \frac{4}{5}h, y_n + \frac{44}{45}k_1 - \frac{56}{15}k_2 + \frac{32}{9}k_3); \\ k_5 &= hf(x_n + \frac{8}{9}h, y_n + \frac{19372}{6561}k_1 - \frac{25360}{2187}k_2 + \frac{64448}{6561}k_3 - \frac{212}{729}k_4); \\ k_6 &= hf(x_n, y_n + \frac{9017}{3168}k_1 - \frac{355}{33}k_2 + \frac{46732}{5247}k_3 + \frac{49}{176}k_4 + \frac{-5103}{18656}k_5) \end{aligned}$$

$$y_{n+1} = y_n + \frac{35}{384}k_1 + \frac{500}{1113}k_3 + \frac{125}{192}k_4 - \frac{2187}{6784}k_5 + \frac{11}{84}k_6$$

$$y_{n+1}^* = y_n + \frac{5179}{57600}k_1 + \frac{7571}{16695}k_3 + \frac{393}{640}k_4 - \frac{92097}{339200}k_5 + \frac{187}{2100}k_6$$

#### Estimativa e Controle do Erro

Como já mencionado, o erro estimado pelo método de Dormand–Prince é a diferença dos dois resultados encontrados, o primeiro de quarta ordem e o segundo de quinta ordem [15]. De posse da avaliação do erro, o algoritmo pode então realizar a adaptação do passo para que o erro atinja ou mantenha valor dentro dos limites desejados. O valor definido como tolerância para o erro é aqui denotado como  $tol$ . Esse valor pode ser dividido em dois valores distintos: a tolerância absoluta,  $abs$ , e a tolerância relativa,  $rel$ . Essa divisão é necessária, pois para determinadas aplicações é tarefa difícil saber previamente a ordem de grandeza dos valores resultantes, ou seja, o valor absoluto de tolerância pode tornar-se desproporcional à medida que o método numérico avança. Por

isso introduz-se também um erro relativo que é proporcional ao valor resultante,  $y_{n+1}$ . Para este trabalho, definiu-se como tolerância relativa uma fração do tamanho da diagonal da célula que contém o ponto. Essa escolha parte do princípio que dentro de uma célula do domínio o erro tende a ser muito pequeno já que a variação do campo é pequena e portanto o erro cometido ali dentro não compromete os resultados. Então, a equação de  $tol$  é dada como:

$$tol = abs + | y \cdot rel | \quad (4.7)$$

Para aplicações como a deste trabalho, onde o resultado representa uma grandeza vetorial, a estimativa do erro deve ser feita como uma distância Euclidiana, pois os pontos em questão estão em um domínio de três dimensões. Desta forma, a equação 4.8 define o erro como o quociente entre a diferença dos dois resultados do método numérico e o valor de  $tol$  que é o valor de tolerância.

$$erro = \frac{(y_{n+1} - y_{n+1}^*)}{tol} \quad (4.8)$$

Se o erro estimado for menor que a unidade, o ponto é validado e então a iteração termina, e o processo é repetido para a próxima iteração. Se o erro for maior que a unidade, o ponto é invalidado e a mesma iteração é repetida com um novo passo (menor) de integração.

A atualização do passo de integração está, então, diretamente ligada ao erro estimado: se o erro for maior que a unidade, o passo de integração deve ser diminuído para que o erro cometido seja menor; portanto, o passo de integração deve ser multiplicado por um fator positivo menor que a unidade. Caso o erro seja menor que a unidade, o passo também pode ser atualizado, neste caso aumentado por um fator maior que a unidade de modo a acelerar o processo de integração, ainda dentro da tolerância numérica escolhida. O fator pelo qual o passo é multiplicado é representado pela equação:

$$factor = \left( \frac{1}{erro} \right)^{\frac{1}{5}} \quad (4.9)$$

Como a estimativa de erro nesse algoritmo não é exata, mas sim a ordem do erro em função do tamanho do passo, é aconselhável introduzir um



fator de segurança ligeiramente menor que a unidade para evitar que o passo de integração aumente para valores de ordem muito superior quando o erro estimado for muito pequeno [15]. A experiência mostra que não é aconselhável deixar o passo de integração crescer ou diminuir muito rápido, e também não deixar o passo de integração aumentar se o passo anterior foi rejeitado [15]. Acrescentando o fator de segurança, a equação que atualiza o passo de integração é dada pela equação abaixo, onde  $S < 1$  é o fator de segurança:

$$h_{n+1} = S \cdot factor \cdot h_n \quad (4.10)$$

## 5

### Testes e Resultados

Apresentamos nesta seção os resultados obtidos de testes realizados com o método proposto comparando com os resultados de outros métodos. Os resultados são analisados com relação a corretude, desempenho e qualidade das linhas de fluxo geradas. Ainda, analisamos o comportamento dos métodos em regiões de geometria descontínua. Nos testes, para o método proposto, quando não especificado diferente, foi aplicado um valor de tolerância absoluta igual a  $10^{-3}$  e, para tolerância relativa, um valor de 10% do tamanho da diagonal da célula que contém o vértice calculado na iteração anterior.

#### 5.1

##### Lançamento de sementes

Para o traçado de linhas de fluxo é necessário definir onde as sementes que originam as linhas serão lançadas no modelo de reservatório natural de petróleo. Para analisar a efetividade dos poços injetores e produtores, as sementes são lançadas em células de completação dos poços injetores [16]. Para definir a distribuição das sementes pelos poços injetores é feito o seguinte procedimento:

1. Cálculo do somatório dos fluxos que passam pelas faces de todas células de completação de todos poços injetores;
2. Distribuição das sementes, por célula de completação, proporcional à sua participação no somatório dos fluxos de todos os poços injetores;
3. Distribuição das sementes, dentro de cada célula de completação, proporcional aos fluxos positivos das faces;
4. Posicionamento aleatório de cada semente em cada face da célula;

Com base no somatório dos fluxos e o número total de linhas lançadas, avaliamos a vazão correspondente a cada linha. Contando o número de linhas que chega a cada poço projetor, podemos mapear a contribuição efetiva dos poços injetores na produção do campo.

## 5.2

### Modelos para Testes

Primeiramente, apresentam-se aqui os modelos de reservatórios naturais de petróleo para realização dos testes. Eles se diferenciam tanto em dimensões, geometria, número de células, número de poços e proporção de células ativas e inativas.

O *Modelo A* se caracteriza por ter o formato de “caixa” e não apresenta células inativas em seu interior. Ele contém apenas um poço injetor e um poço produtor e é utilizado basicamente para testar a corretude do método proposto, comparado com a solução analítica do método *Tracking Particles*[8]. Ele é ilustrado na Figura 5.1.

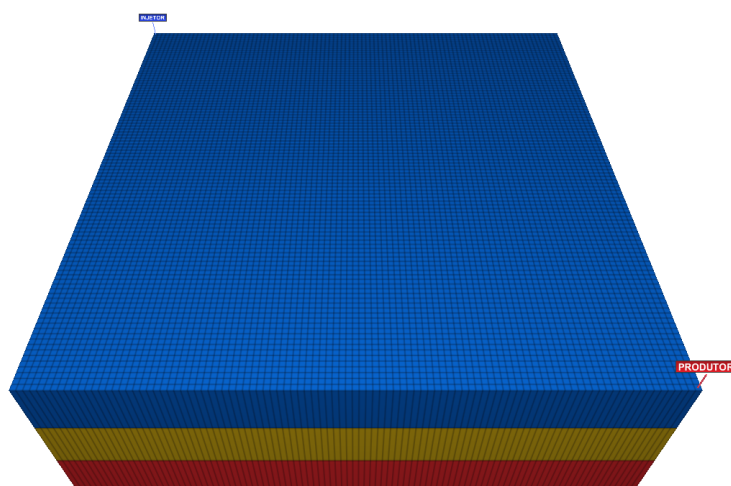


Figura 5.1 – Modelo de Reservatório A.

O *Modelo B* já caracteriza um reservatório real, embora de pequeno porte. Ele apresenta aproximadamente 30.000 células em seu domínio, com células inativas e regiões de falha.

O *Modelo C* é o maior dos três, com mais de 300000 células, apresentando uma geometria complexa, além de uma grande quantidade de células inativas e a presença de regiões de falha. Este é o modelo mais próximo dos modelos de campos atualmente em exploração no Brasil.

## 5.3

### Resultados e Avaliação da Corretude

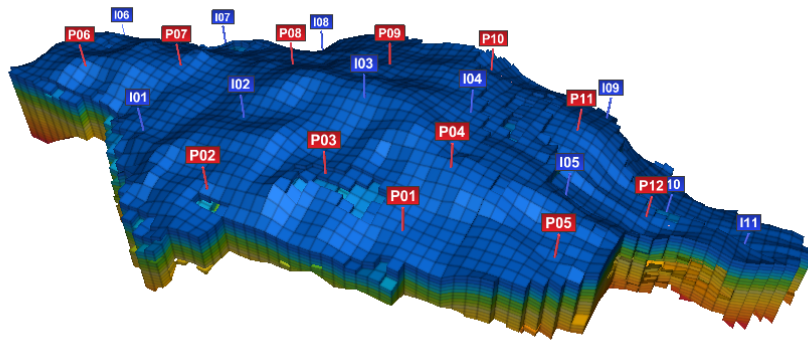


Figura 5.2 – Modelo de Reservatório B.

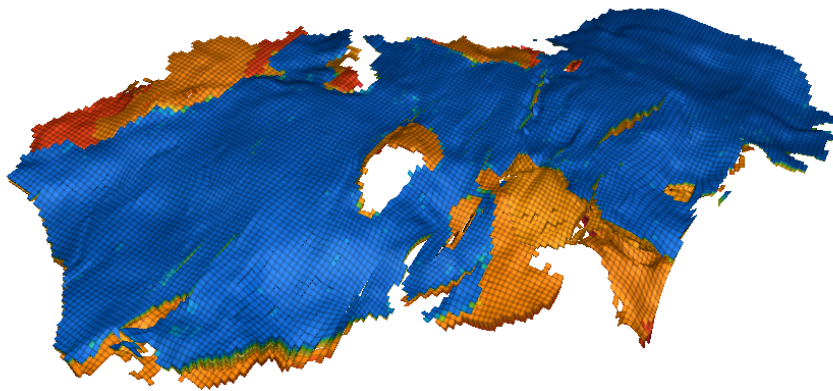


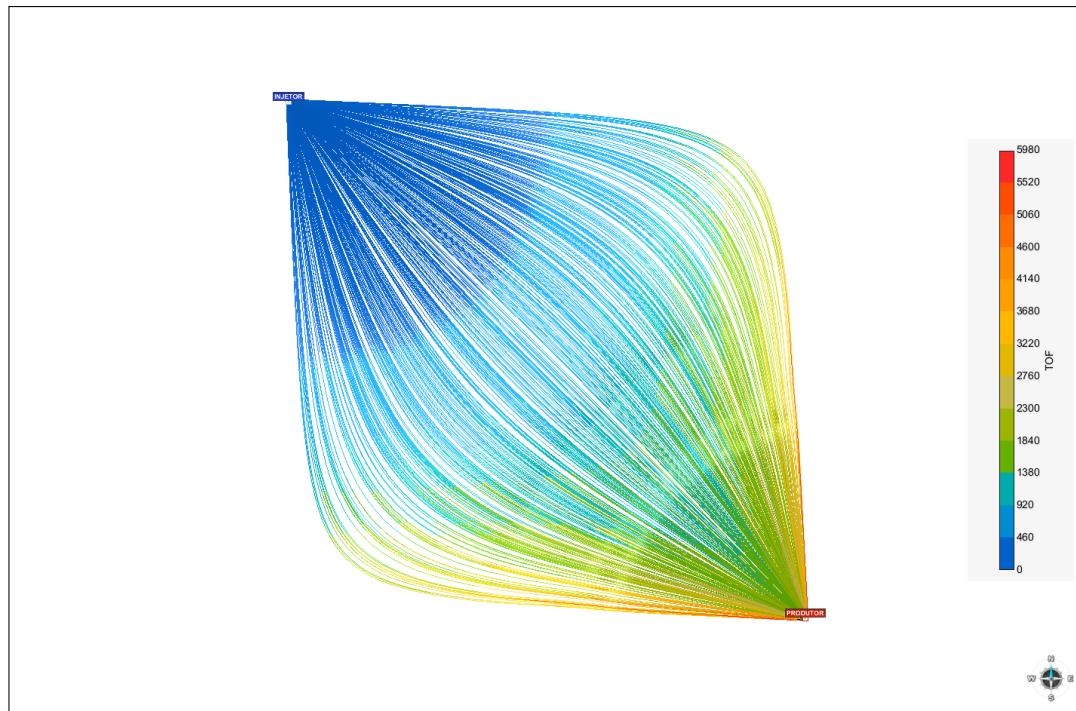
Figura 5.3 – Modelo de Reservatório C.

### 5.3.1 Modelo A

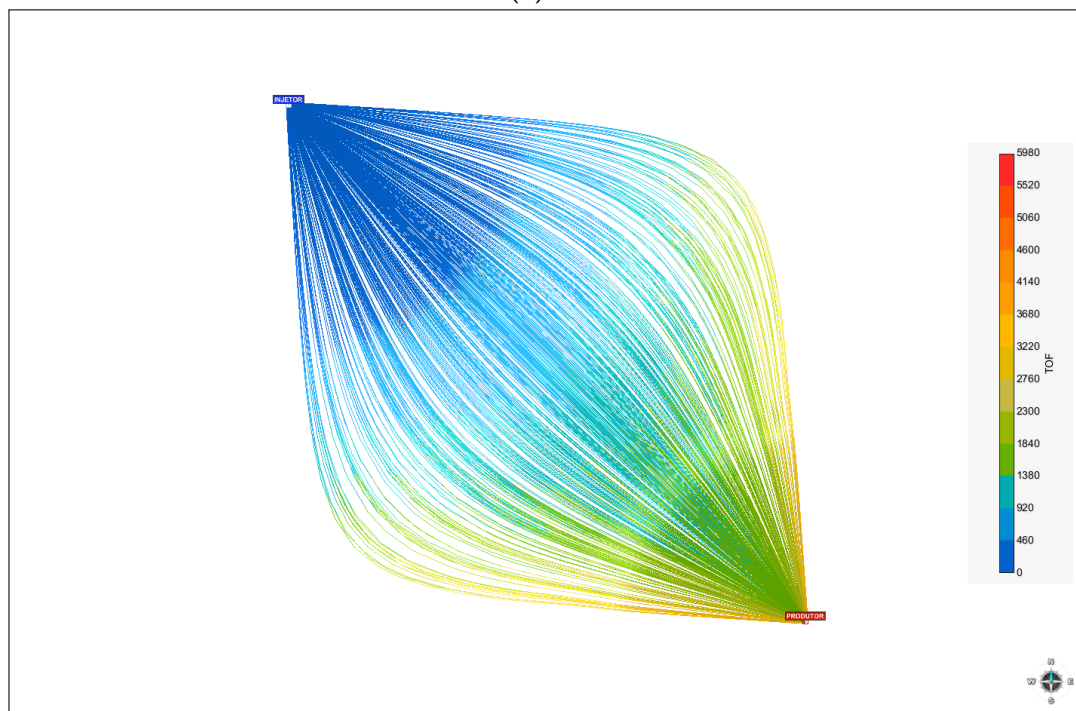
Esse primeiro teste serve para avaliar a corretude obtida com o método proposto em comparação com o método *Particle Tracking*, que só é aplicável a modelos *bloco centrado* (células regulares). Como o método *Particle Tracking* baseia-se numa solução analítica da integral, ele tende a apresentar bom desempenho e corretude, em modelos sem descontinuidade.

Comparamos então os resultados obtidos pelo método *Particle Tracking* e pelo Método Proposto, lançando 500 sementes nas células completadas do poço injetor. As linhas de fluxo obtidas são ilustradas na Figura 5.4. A escala de cores indica tempo de voo das linhas.

Observa-se uma semelhança entre os dois resultados produzidos tanto pelo aspecto da trajetória das linhas quanto pela coloração das mesmas, indicando tempos de voo próximos. Observa-se também que a suavização do campo vetorial feita no método proposto faz com que as linhas se concentrem na região de maiores velocidades que, nesse modelo específico, é a região



(a)



(b)

Figura 5.4 – Comparação entre as linhas geradas pelo método de referência, *Particle Tracking*, em (a) e pelo método Numérico Adaptativo, em (b)

próxima a reta que liga os poços injetor e produtor. Isso faz com que exista uma pequena diferença entre as linhas que estão mais próximas às bordas do modelo.

Para ratificar a similaridade entre as linhas geradas pelos dois métodos,

faz-se uma comparação do comprimento das linhas. Esta comparação está mostrada na Figura 5.5. Pares de barras do gráfico comparam os comprimentos das linhas obtidos pelos dois métodos. Essa semelhança entre os comprimentos se verifica para todas linhas, mas para uma melhor visualização do gráfico, foram amostradas apenas vinte, escolhidas aleatoriamente. Os comprimentos das linhas geradas pelo Método Proposto são ligeiramente menores que os das linhas geradas pelo *Particle Tracking*. Este fato é explicado justamente pelo efeito da suavização do campo de velocidades que mudou a direção dos vetores mais próximos à borda do modelo de forma que sua orientação fosse mais direcionada ao interior do modelo, sob influencia dos vetores que estão situados mais no interior que possuem velocidades com maior magnitude. Isso gerou linhas com menor curvatura e, conseqüentemente, menor comprimento. O efeito da suavização do campo é mais notado neste exemplo pois a direção do campo não varia muito. Em campos reais, com campos de velocidade complexo, este efeito se reduz. De qualquer forma, é difícil avaliar qual dos resultados reproduz melhor o real comportamento do fluxo nos reservatórios.

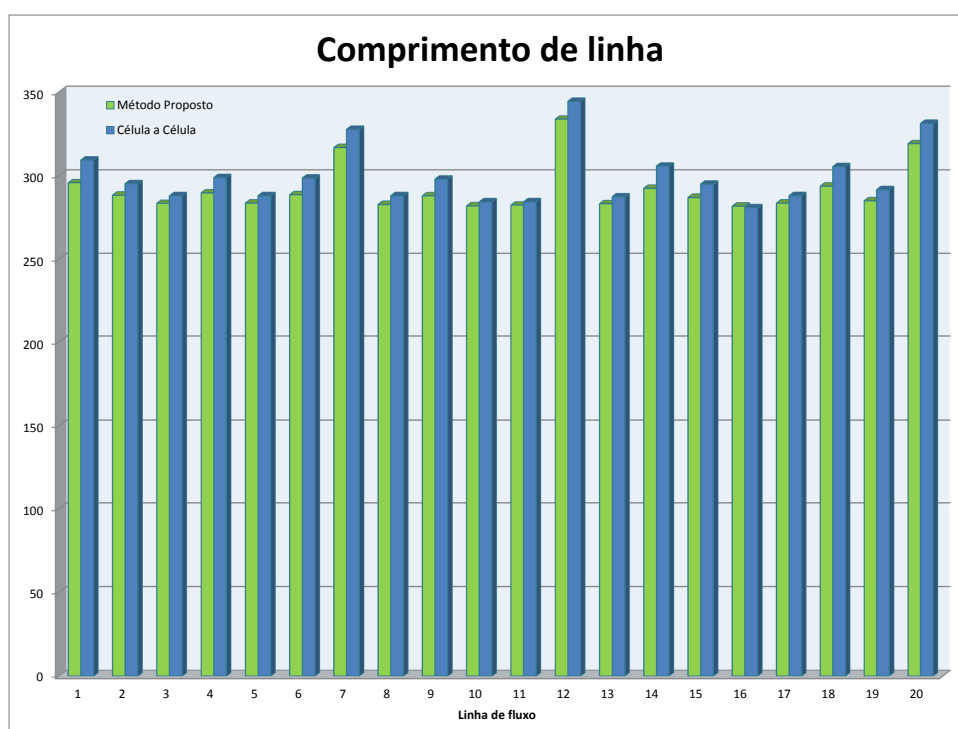


Figura 5.5 – Comparação entre os comprimentos das linhas geradas pelos dois métodos *Particle Tracking* e Método Proposto no modelo de reservatório A

### 5.3.2

#### Modelo B

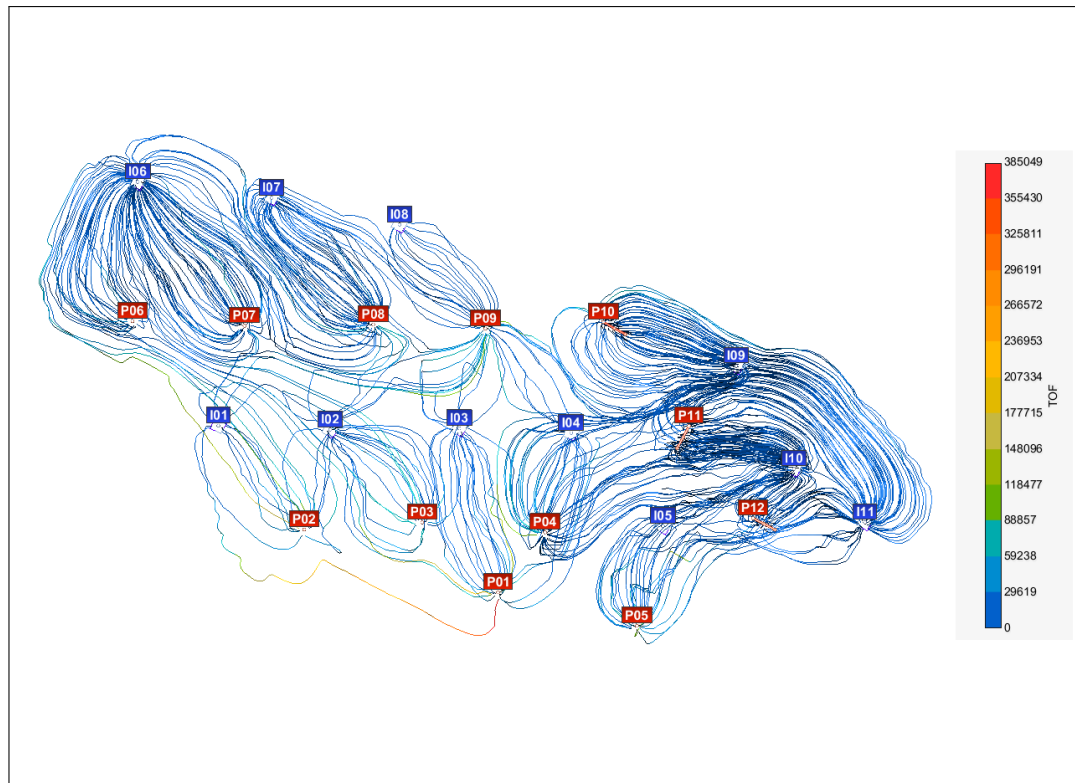
Para o segundo modelo, comparam-se as linhas geradas pelo Método Proposto e o Método Célula a Célula, com solução numérica, aplicado a modelos *corner points* (células irregulares). O método *Particle Tracking* não pode ser testado pois ele se limita a modelos de células regulares. Mais uma vez são lançadas quinhentas sementes distribuídas pelos poços injetores do modelo.

Os resultados para este modelo estão mostrados na Figura 5.6. Percebe-se que no geral a distribuição das linhas ao longo do reservatório é semelhante em ambos resultados, com coloração das linhas também semelhantes, indicando tempos de voo semelhantes. Na Figura 5.7, compara-se uma região do modelo onde ficou evidenciado o efeito da suavização no traçado das linhas. No primeiro resultado produzido pelo método célula-a-célula, nota-se a presença de linhas não suaves devido à descontinuidade do campo de velocidade, enquanto que o segundo resultado apresenta linhas com variações mais brandas no seu traçado, consequência da suavização do campo de velocidades.

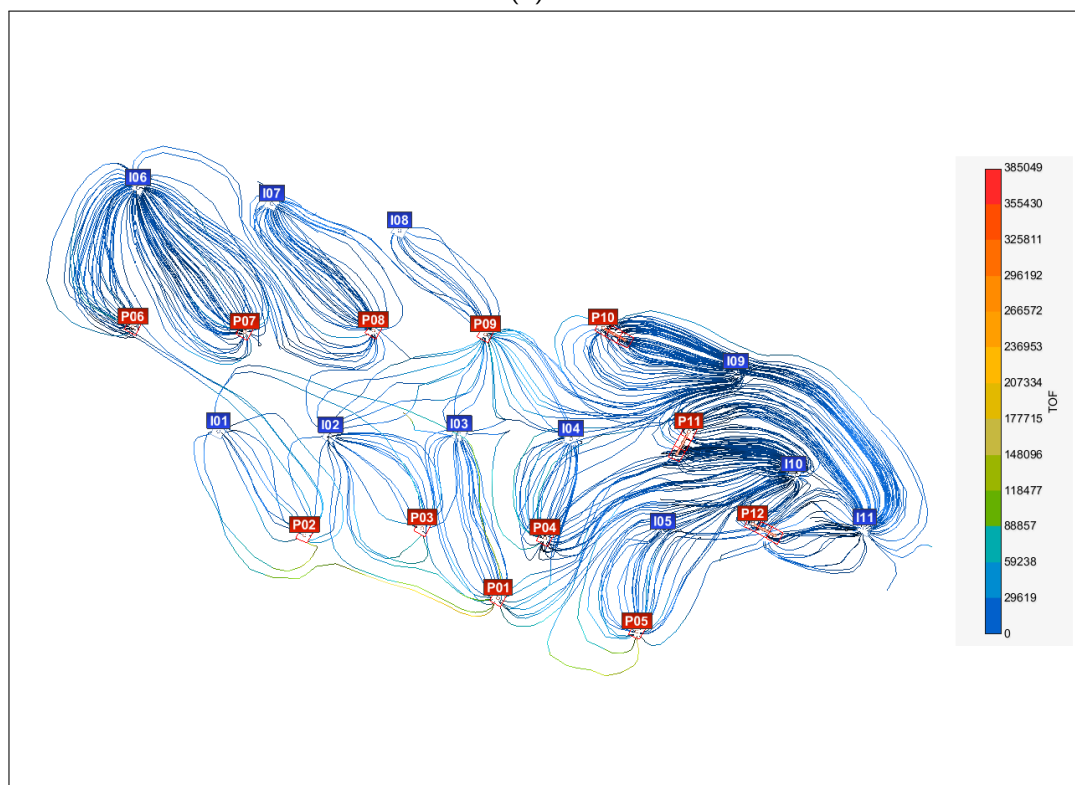
Nos modelos reais de reservatório, nem todas as linhas lançadas alcançam um poço produtor. As razões para as linhas serem interrompidas são diversas: linhas que ultrapassam o tempo máximo de voo especificado, por ter entrado em uma região com velocidade próxima de zero; linhas que ultrapassam a fronteira externa do reservatório; ou linhas que entram em regiões de células inativas no interior do modelo.

Para este modelo, das 500 linhas lançadas, apenas 435 alcançaram poços produtores com o Método Proposto, enquanto que 461 alcançaram com o Método Célula a Célula. Observou-se que grande parte das linhas foram interrompidas, no Método Proposto, por cruzarem a fronteira externa do reservatório. Mais uma vez, este fato é justificado pela suavização do campo. A Figura 5.8 ilustra o que ocorre com a suavização do campo próximo da fronteira. Em geral, a célula da fronteira tem velocidade local tangencial à fronteira. Com a suavização, esta célula ganha um componente de velocidade não tangencial, o que faz com que a linha tenda a colidir com a fronteira do reservatório. Durante a realização desta pesquisa, exploramos duas técnicas de tratamento de colisão com a fronteira externa do reservatório: eliminação de componentes de velocidades normais às fronteiras em células de borda e tratamento de colisão via projeção de partículas fora do domínio. No entanto, devido à complexidade geométrica destas fronteiras em modelos reais, nenhuma das técnicas se mostrou robusta e correta para ser adotada. Por outro lado, as regiões próximas aos limites do domínio do reservatório são, em geral,





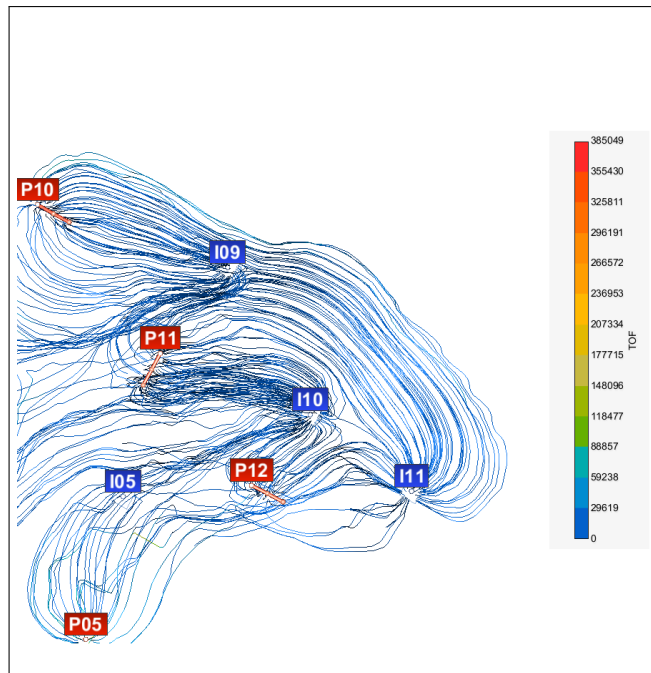
(a)



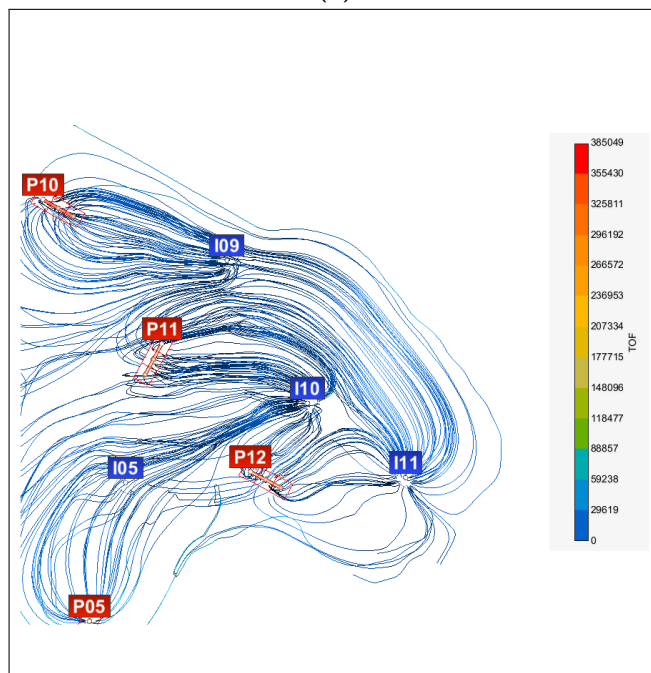
(b)

Figura 5.6 – Comparação entre as linhas geradas pelo Método Célula a Célula em (a) e pelo Método Proposto em (b) para o Modelo de Reservatório B





(a)



(b)

Figura 5.7 – Comparação entre as linhas geradas pelo Método Célula a Célula em (a) e pelo Método Proposto em (b) enfatizando o efeito da suavização do campo de velocidades

regiões menos importantes, onde o campo de velocidade é muito reduzido. Acreditamos que a perda de linhas nestas regiões não afeta a análise da distribuição do fluxo no domínio do reservatório.

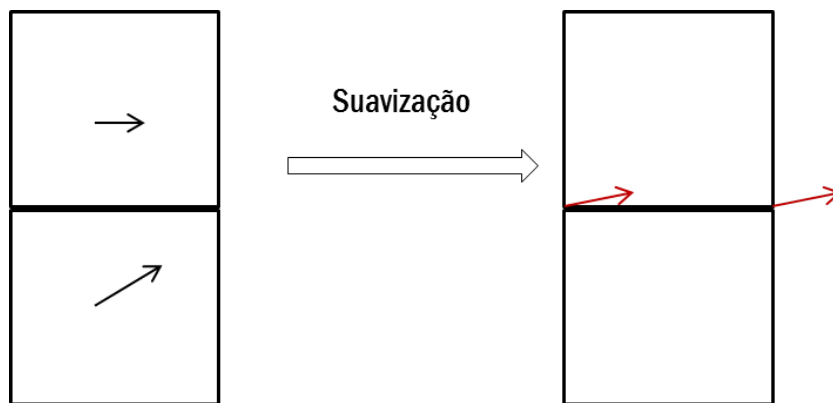


Figura 5.8 – Exemplo do efeito da suavização em duas células vizinhas. Se a célula A estiver na borda do modelo, as velocidades suavizadas em vermelho podem levar as linhas de fluxo que entrarem nessa célula a saírem do modelo

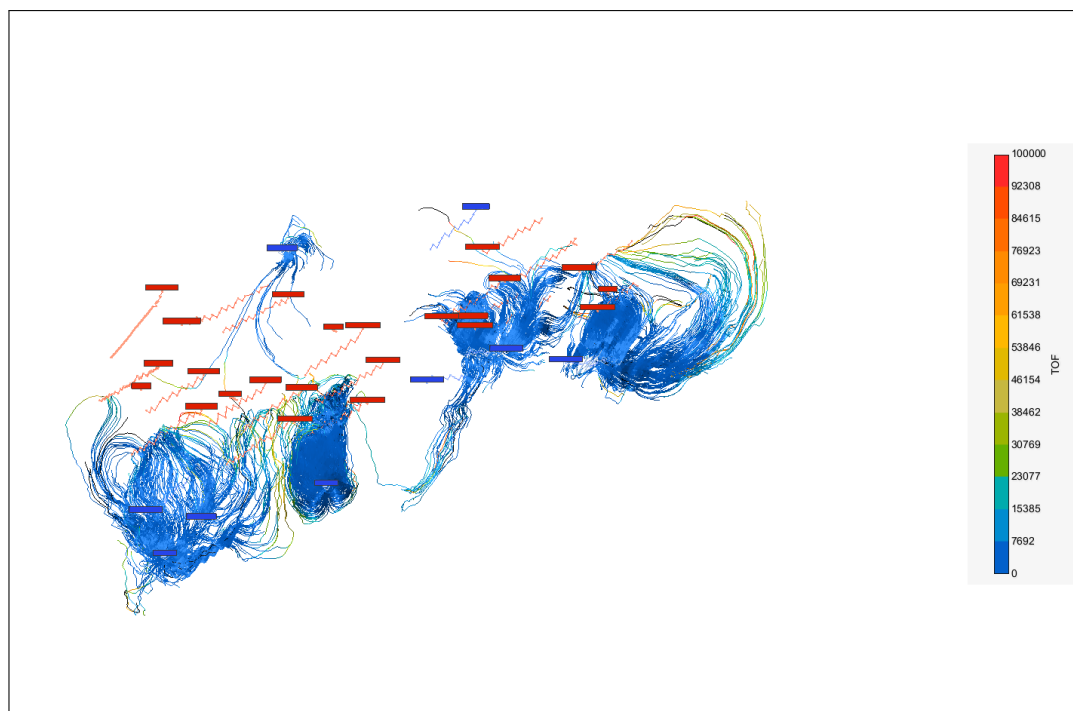
### 5.3.3 Modelo C

Para o Modelo C, também foram geradas 5000 linhas. A Figura 5.9 mostra as linhas geradas em cada um dos métodos. Percebe-se que as linhas geradas pelo Método Proposto têm aspecto mais suave do que as geradas pelo Método Célula a Célula.

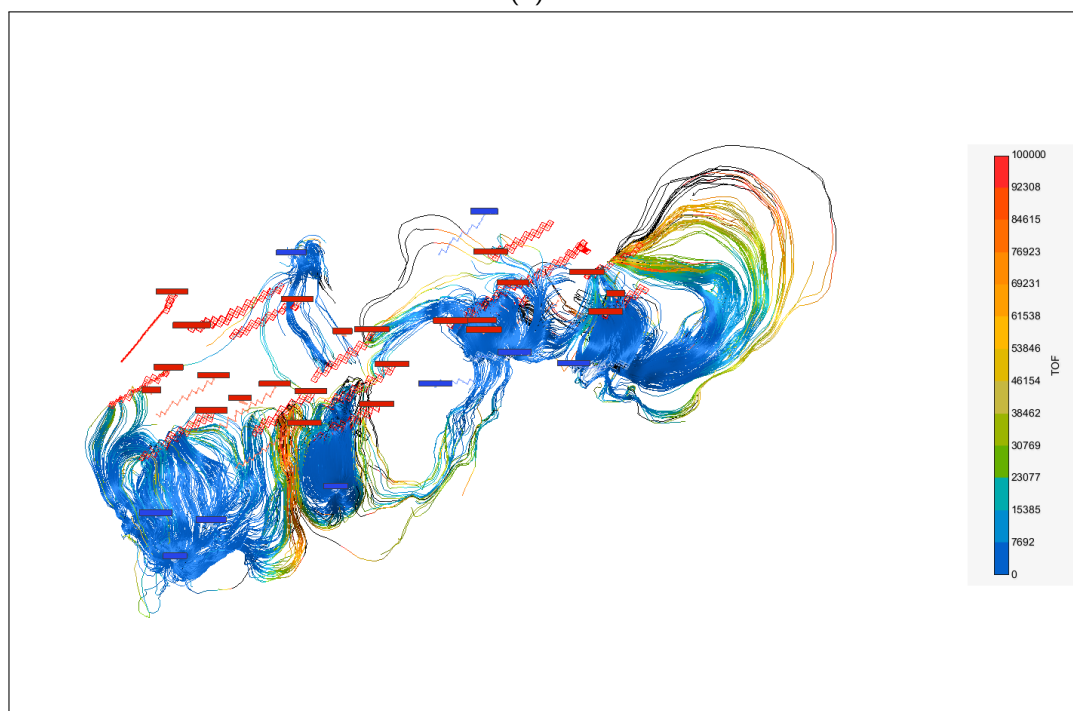
Para este modelo, o método proposto apresenta uma vantagem significativa. Das 5000 linhas lançadas, 3160 alcançaram poços produtores pelo Método Proposto, enquanto apenas 2003 não foram interrompidas pelo Método Célula a Célula. Como se trata de um modelo com regiões de geometria complexa, com muitas células inativas no interior do domínio, o Método Célula a Célula sofre com a interrupção de linhas que atingem células inativas. No Método Proposto, a suavização do campo por si só ajuda as linhas contornarem a fronteira das células inativas no interior do domínio. Além disso, como o campo suavizado é representado nos vértices das células, a primeira camada de células inativas apresenta velocidades não nulas, devido aos vértices compartilhados com células ativas. Isso permite que as linhas avancem em regiões inativas próximas do domínio do reservatório. Outros experimentos, conduzidos com modelo reais de campos da camada de pré-sal do Brasil, indicam que as linhas interrompidas nos métodos convencionais célula-a-célula tornam inviável a análise do fluxo por traçado de linhas.

## 5.4 Avaliação de Desempenho

Para fazer a avaliação do desempenho do método proposto, faz-se aqui uma comparação entre as abordagens do Método Proposto e do Método Célula a Célula, que resolve a equação da linha através de um método numérico não-



(a)



(b)

Figura 5.9 – Comparação entre as linhas geradas pelo Método Célula a Célula em (a) e pelo Método Numérico Adaptativo em (b) para o Modelo de Reservatório C

adaptativo. Para o primeiro modelo também foi testado o *Particle Tracking*, já que ele se direciona a este tipo de modelo.

Tabela 5.1 – Resultados de tempo de execução para o Modelo A

Modelo A		
Método	Tempo (s)	Linhas Completas
Célula a célula	5,853335	500
Adaptativo com Tolerância de $10^{-3}$	1,089062	500
Particle Tracking	1,029329	500

Tabela 5.2 – Resultados de tempo de execução para o Modelo B

Modelo B		
Método	Tempo (s)	Linhas Completas
Célula a célula	7,8504491	461
Adaptativo com Tolerância de $10^{-3}$	1,526087	435
Adaptativo com Tolerância de $10^{-2}$	1,482003	426

#### 5.4.1

##### Tempo Total de Execução

Uma avaliação relevante é verificar o tempo total de execução de cada método para traçar um determinado número de linhas em um dado reservatório. Este tipo de teste evidencia a performance geral de cada método. O método *Particle Tracking* é testado para o Modelo A e é um bom parâmetro de comparação por apresentar tempos de execução ótimo para este tipo de modelo.

Os resultados para o Modelo A estão demonstrados na Tabela 5.1. Verifica-se que o Método Proposto faz o traçado das linhas em tempo muito menor que o Método Célula a Célula, se aproximando inclusive do *Particle Tracking* que é um método analítico, que geralmente apresenta menor custo computacional em seu algoritmo. Este desempenho superior do Método Proposto pode ser explicado pelo fato de que o campo de velocidades no Modelo A apresenta baixas variações de uma célula pra outra, ou seja, é o domínio discreto ideal para se interpretar como um meio contínuo. Assim é o melhor caso para se utilizar um método que resolve a equação da linha de forma numérica e adaptativa, pois existe baixa ocorrência de erro do método numérico, e portanto necessita-se de poucas iterações para completar o traçado de uma linha.

Os resultados para o Modelo B estão demonstrados na Tabela 5.2. Nota-se que ao aumentar a precisão do método adaptativo, ou seja, ao diminuir-se a tolerância, verifica-se que o tempo de execução também aumenta como era esperado.

Para o Modelo C, os resultados apresentam novamente tempos parecidos, mas neste caso um fato diferente influenciou os tempos de execução: o relativo

Tabela 5.3 – Resultados de tempo de execução para o Modelo C

Modelo C		
Método	Tempo (s)	Linhas Completas
Célula a Célula	44,045192	2003
Adaptativo com Tolerância de $10^{-3}$	54,411112	3160
Adaptativo com Tolerância de $10^{-2}$	54,085093	3125

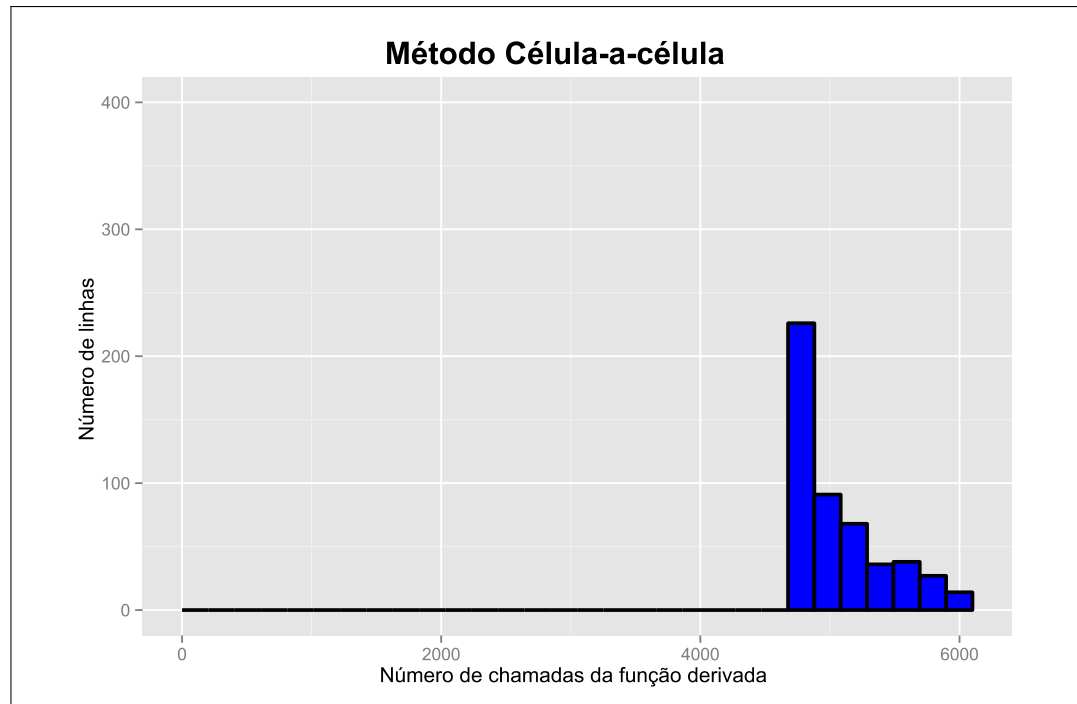
baixo número de linhas de fluxo que atingiram poços produtores. Isto se explica pelas características do modelo que tem uma geometria mais complexa que o Modelo B, além de uma grande quantidade de células inativas e regiões de falha. Os tempos totais de execução devem ser interpretados considerando-se o número de linhas que alcançaram poços produtores, embora seja importante salientar que as linhas descartadas também consomem processamento, e seria inadequado simplesmente descartar as linhas interrompidas numa análise de desempenho.

#### 5.4.2

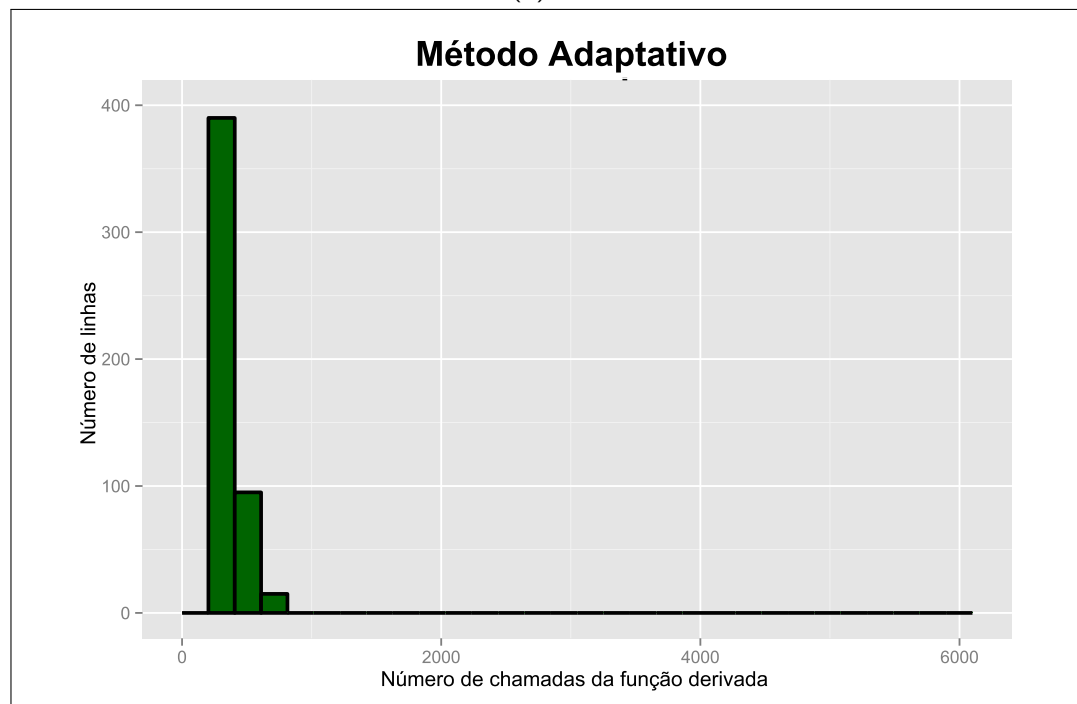
##### Número de Chamadas da Função Derivada por cada Método

Como duas abordagens testadas utilizam um método numérico para resolver a equação diferencial da linha de fluxo, uma boa métrica para avaliar o desempenho de cada algoritmo é quantificar o número de chamadas da função derivada realizadas ao traçar um determinado número de linhas de fluxo em um dado modelo de reservatório. Nesta seção, compara-se o esforço que cada método realiza para traçar quinhentas linhas ao longo do reservatório, saindo de um poço injetor até chegar a um poço produtor. Os resultados estão apresentados sob a forma de histogramas, Figuras 5.10, 5.11 e 5.12.

Analisando os histogramas, percebe-se que para os três modelos, o número de chamadas da função derivada é sempre menor para o Método Proposto e essa diferença chega a ser mais considerável no primeiro modelo, onde se encontra o domínio discreto mais propício a se utilizar o Método Proposto.

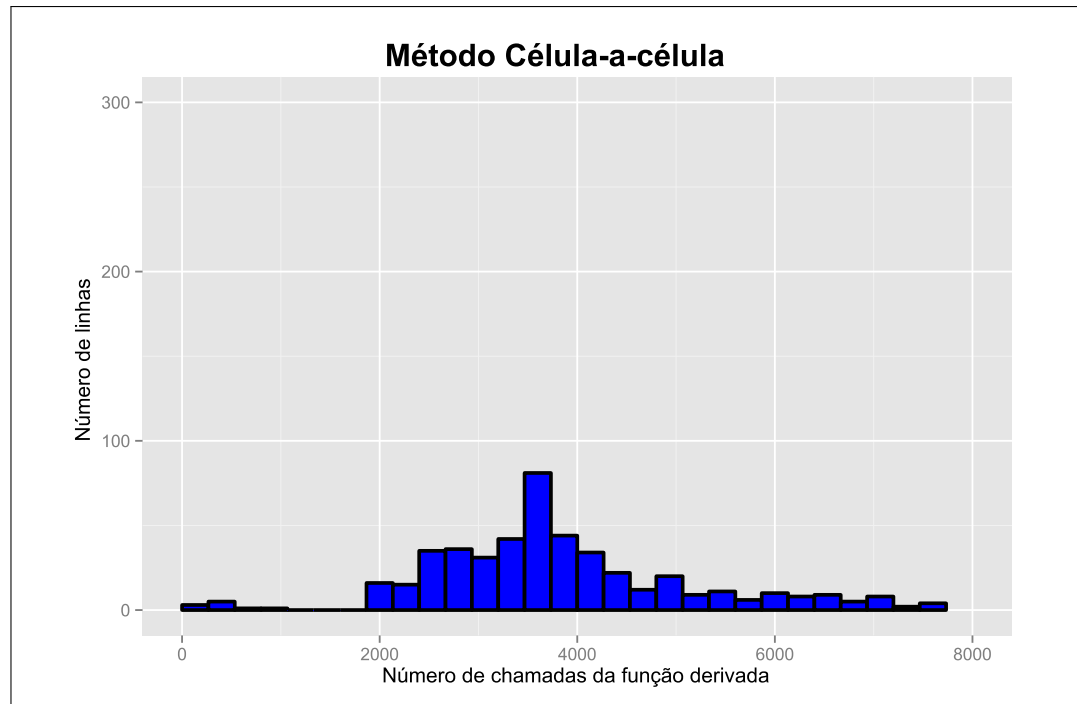


(a)

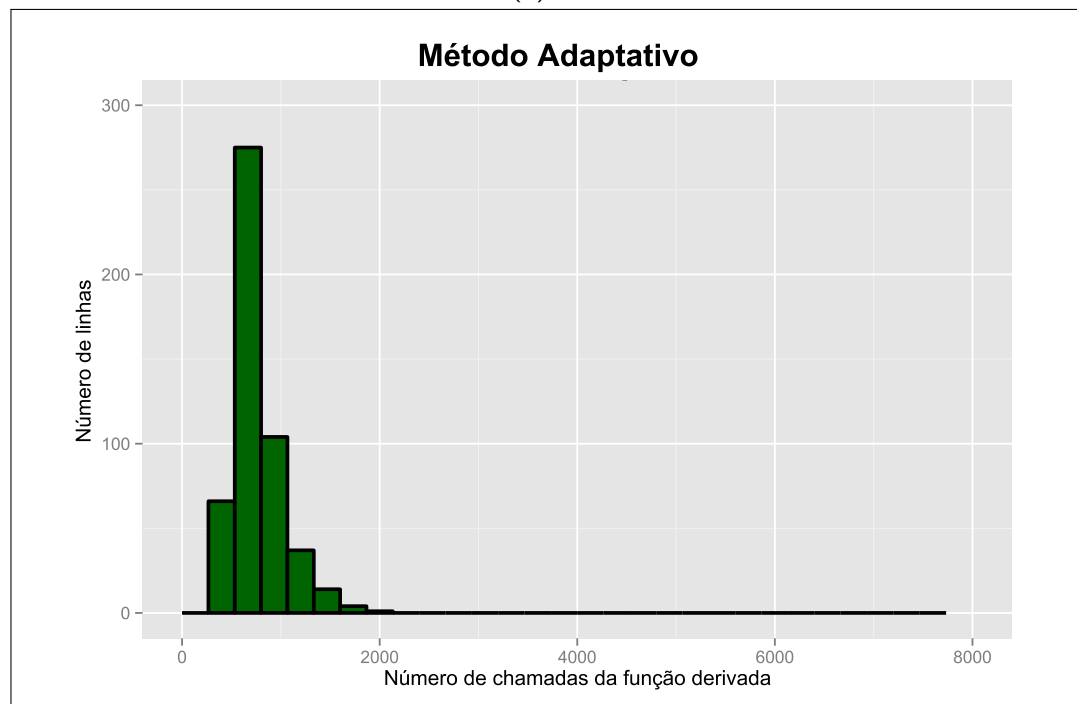


(b)

Figura 5.10 – Histogramas do número de chamadas da função derivada realizadas pelos Métodos: Célula a Célula em (a) e Adaptativo em (b) no modelo A

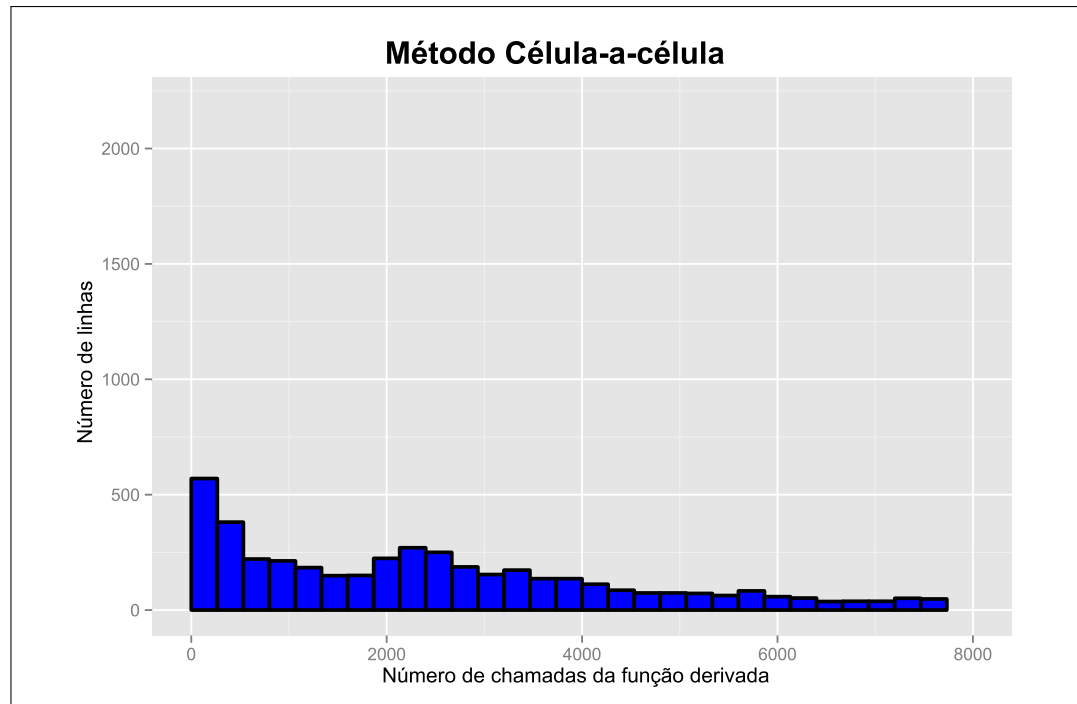


(a)

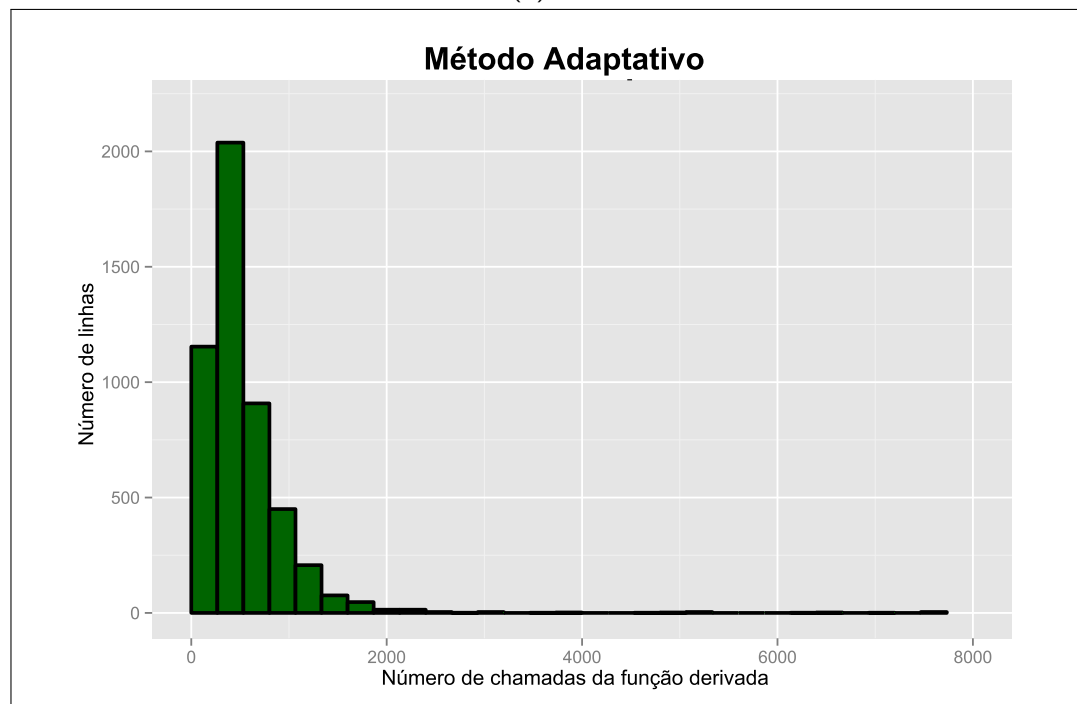


(b)

Figura 5.11 – Histogramas do número de chamadas da função derivada realizadas pelos Métodos: Célula a Célula em (a) e Adaptativo em (b) no modelo B



(a)



(b)

Figura 5.12 – Histogramas do número de chamadas da função derivada realizadas pelos Métodos: Célula a Célula em (a) e Adaptativo em (b) no modelo C



## 6

### Conclusão

Este trabalho apresentou um novo método para traçado de linhas de fluxo em modelos de reservatórios naturais de petróleo. Esse método se apresenta como boa alternativa aos tradicionais métodos célula-a-célula, que se caracterizam por fazer a integração de cada segmento de linha limitado ao domínio de uma célula. O método proposto foi desenvolvido baseado na ideia de tratar o modelo de reservatório discreto com um campo de velocidade contínuo e, como consequência, foi possível traçar as linhas sem a restrição de que um segmento da linha esteja limitado ao domínio de uma célula do modelo.

A partir dessa ideia, foi necessário, primeiramente, eliminar as discontinuidades do campo de velocidades através da suavização do mesmo. A suavização do campo de velocidades foi realizada com uso da técnica de média esférica de vetores. Também foi necessário localizar eficientemente um ponto qualquer no modelo de reservatório e, para isso, utilizou-se uma estrutura espacial de aceleração. Para interpolar o valor da velocidade no interior da célula, empregou-se interpolação linear esférica. Para assegurar precisão numérica, utilizou-se um método numérico adaptativo, preservando o erro numérico de integração em cada passo inferior a um valor de tolerância pré-estabelecido.

Os resultados atestaram que o método proposto neste trabalho se mostrou melhor que um método célula-a-célula convencional para modelos reais de reservatório, tanto pelo maior desempenho quanto pelo menor número de linhas interrompidas nas regiões de células inativas. Como trabalho futuro, pode ser feito estudo de como melhorar o método proposto de forma a minimizar o efeito da suavização do campo de velocidades nas células de borda do modelo evidenciado nos testes.

## Referências Bibliográficas

- [1] TECGRAF/PUC-RIO, I.. **Geresim: Sistema de gerenciamento de simulação de reservatórios**. Versão 3.6, Junho 2015.
- [2] ABRAHAM, F. R.. **Visualização de modelos massivos de reservatórios naturais de petróleo**. Master's thesis, Departamento de Informatica, Pontificia Universidade Catolica, Rio de Janeiro, Sept. 2011.
- [3] FRANCESCHIN, B. B.. **Visualização de Seções de corte Arbitrárias de malhas Não estruturadas**. Master's thesis, Departamento de Informatica, Pontificia Universidade Catolica, Rio de Janeiro, p. 26–29, Apr. 2012.
- [4] LAN HAI-QIANG, Z. Z.-J.. **Seismic wavefield modeling in media with fluid-filled fractures and surface topography**. *APPLIED GEOPHYSICS*, 9(3):301, 2012.
- [5] A.FELIPPA, C.. **Advanced finite element methods**, 2013.
- [6] DAM, E. B.; KOCH, M. ; LILLHOLM, M.. **Quaternions, interpolation and animation**. Technical report, 1998.
- [7] BUSS, S. R.; FILLMORE, J. P.. **Spherical averages and applications to spherical splines and interpolation**. *ACM Trans. Graph.*, 20(2):95–126, Apr. 2001.
- [8] POLLOCK, D. W.. **Semianalytical computation of path lines for finite-difference models**. *Ground Water*, 26(6):743–750, 1988.
- [9] PREVOST, M.; EDWARDS, M. ; BLUNT, M.. **Streamline tracing on curvilinear structured and unstructured grids**. *SPE Reservoir Evaluation And Engineering*, 7:139–148, 2002.
- [10] HAEGLAND, H.; DAHLE, H. K.; EIGESTAD, G. T.; A. LIE, K. ; AAVATSMARK, I.. **Improved streamlines and time-of-flight for streamline simulation on irregular grids**. *Adv. Water Resour*, 30:10–1016.

- [11] JIMENEZ, E.; SABIR, K.; DATTA-GUPTA, A.; KING, M. J. ; OTHERS. **Spatial error and convergence in streamline simulation.** SPE Reservoir Evaluation And Engineering, 10(03):221–232, 2007.
- [12] UENG, S.-K.; SIKORSKI, C. ; MA, K.-L.. **Efficient streamline, stream-ribbon, and streamtube constructions on unstructured grids.** Visualization and Computer Graphics, IEEE Transactions on, 2(2):100–110, Jun 1996.
- [13] HILL, S.. **Tri-linear interpolation.** In: Heckbert, P. S., editor, GRAPHICS GEMS IV, p. 521–525. Academic Press, 1994.
- [14] DORMAND, J.; PRINCE, P.. **A family of embedded runge–kutta formulae.** Journal of Computational and Applied Mathematics, 6(1):19 – 26, 1980.
- [15] PRESS, W. H.; TEUKOLSKY, S. A.; VETTERLING, W. T. ; FLANNERY, B. P.. **Numerical Recipes 3rd Edition: The Art of Scientific Computing.** Cambridge University Press, New York, NY, USA, 3 edition, 2007.
- [16] THIELE, M. R.; BATYCKY, R. P.. **Using streamline-derived injection efficiencies for improved waterflood management.** SPE Reservoir Evaluation and Engineering, p. 187–196, Apr. 2006.