

---

Estabilidade numérica em escoamentos  
viscoelásticos com alto número de Weissenberg

*Irineu Lopes Palhares Junior*

---

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: \_\_\_\_\_

—

**Irineu Lopes Palhares Junior**

## **Estabilidade numérica em escoamentos viscoelásticos com alto número de Weissenberg**

Monografia apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, para o Exame de Qualificação, como parte dos requisitos para obtenção do título de Doutor em Ciências - Ciências de Computação e Matemática Computacional.

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. José Alberto Cuminato

**USP – São Carlos  
Agosto de 2015**

# Resumo

---

Apresentamos o estudo numérico de um método estabilizador para a simulação de escoamentos de fluidos viscoelásticos, confinados e com superfície livre, com alta elasticidade. Nesta formulação, a equação constitutiva do tensor conformação, usada para definir o modelo viscoelástico, é modificada introduzindo uma equação evolutiva para a raiz quadrada do tensor conformação.

Ambos os escoamentos, confinado e com superfície livre, são considerados, usando dois códigos numéricos diferentes. Um código para o método de volumes finitos é usado para escoamentos confinados e outro código com o método de diferenças finitas, desenvolvido no contexto do método *Marker-And-Cell* (MAC), é usado para escoamentos confinados e com superfície livre. A implementação da formulação *square-root* é apresentada para o esquema numérico de diferenças finitas e discutido em termos de sua capacidade e eficiência para o cálculo de escoamentos de fluidos viscoelásticos estacionários e transientes. Os resultados numéricos demonstram que a formulação *square-root* é eficiente na simulação de problemas de referência (*benchmark problems*) para alto número de Weissenberg, tais como o problema da cavidade, o escoamento em uma fenda transversal (*cross-slot*) e o impacto da gota, problema com superfície livre.

Além disso, propusemos um novo método estabilizador para o Problema de Alto Número de Weissenberg denominado *eigendecomposed kernel-conformation*. Nosso método consiste na junção de dois outros métodos estabilizadores, *continuous-eigendecomposition* e *kernel-conformation*. O método é validado mediante o escoamento entre duas placas paralelas (*poiseuille flow*).

Finalmente, motivados pelo comportamento particular do problema do impacto da gota para fluidos viscoelásticos, apresentamos uma forma de incluir os efeitos de tensão superficial e molhamento ao método MAC.

**Palavras-Chave:** Problema de Alto Número de Weissenberg; formulação *square-root*; escoamentos complexos; fluidos viscoelásticos; escoamentos confinados; escoamentos com superfície livre; formulação *eigendecomposed kernel-conformation*; tensão superficial; molhamento.

# Abstract

---

We present a numerical study of a stabilization method for computing confined and free-surface flows of highly elastic viscoelastic fluids. In this approach, the constitutive equation based on the conformation tensor, which is used to define the viscoelastic model, is modified introducing an evolution equation for the square root conformation tensor.

Both confined and free-surface flows are considered, using two different numerical codes. A finite volume method is used for confined flows and a finite difference code developed in the context of the Marker-And-Cell (MAC) method is used for confined and free-surface flows. The implementation of the square-root formulation was performed in finite difference numerical scheme and discussed in terms of its ability and efficiency to compute steady and transient viscoelastic fluid flows. The numerical results show that the square-root formulation performs efficiently in the tested benchmark problems at high-Weissenberg number flows, such as the lid-driven cavity flow, the cross-slot flow and the impacting drop free surface problem.

Furthermore, we propose a new stabilization method based on the High Weissenberg Number Problem called eigendecomposed kernel-conformation. Our method consist of the combination of two previous stabilization techniques, the continuous-eigendecomposition and the kernel-conformation. The validation of this method is performed by the poiseuille flow.

Finally, motivated by the particular behavior of the impacting drop problem to viscoelastic fluid flow, we present a strategy to employ the surface tension and wetting effects to the MAC method.

**Keywords:** High Weissenberg Number Problem; square-root formulaton; complex flows; viscoelastic fluids; confined flows; free surface flows; eigendecomposed kernel-conformation formulation; surface tension; wettability.

---

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>4</b>
1.1	Métodos estabilizadores para o Problema de Alto Número de Weissenberg .	4
1.2	Efeitos de tensão superficial e molhamento . . . . .	7
1.3	Organização do trabalho . . . . .	8
<b>2</b>	<b>Equações governantes</b>	<b>10</b>
<b>3</b>	<b>Estudo do método estabilizador square-root</b>	<b>12</b>
3.1	Metodologia . . . . .	12
3.2	Método numérico . . . . .	14
3.3	Validação: escoamento em uma cavidade . . . . .	16
3.4	Aplicações . . . . .	20
3.4.1	Escoamento confinado: Cross-Slot . . . . .	20
3.4.2	Escoamento com superfície livre: impacto da gota . . . . .	23
<b>4</b>	<b>Estudo do método estabilizador eigendecomposed kernel-conformation</b>	<b>27</b>
4.1	Metodologia . . . . .	27
4.1.1	Transformação de Cayley . . . . .	30
4.1.2	Transformação Kernel . . . . .	32
4.2	Método Numérico . . . . .	33
4.3	Validação: escoamento entre placas paralelas . . . . .	36
<b>5</b>	<b>Emprego da tensão superficial ao método MAC</b>	<b>40</b>
5.1	Metodologia . . . . .	42
5.1.1	Método de projeção . . . . .	42
5.1.2	Conceitos básicos sobre projeção ortogonal . . . . .	45
5.1.3	O método dos mínimos quadrados . . . . .	46
5.1.4	A inclusão da tensão superficial ao método MAC . . . . .	47
5.2	Método Numérico . . . . .	56
<b>6</b>	<b>Considerações finais</b>	<b>57</b>

# Introdução

## 1.1 Métodos estabilizadores para o Problema de Alto Número de Weissenberg

Diversas aplicações em engenharia fazem uso de fluidos viscoelásticos, caracterizados por materiais “macios”, tais como solução de polímero (fluidos contendo moléculas poliméricas com milhares de átomos por macromolécula), suspensões coloidais<sup>1</sup>, gel, emulsão<sup>2</sup>, ou surfactantes. Estes diversos fluidos viscoelásticos podem ser descritos usando apropriadas equações constitutivas, que descrevem o comportamento reológico do material, por exemplo, a relação entre a tensão (força por unidade de área) e a deformação (medida da história da deformação) ou taxa de deformação. Tais equações constitutivas dependem da estrutura do fluido, e podem ser apresentadas em sua forma algébrica, diferencial, integral ou integral-diferencial [1, 2]. Dois números adimensionais, usados para estes fluidos, são frequentemente usados para definir a razão entre duas forças importantes ou a razão entre escalas de tempo, chamados de número de Weissenberg ( $Wi$ ) e número de Debora ( $De$ ). O número de Debora,  $De = \lambda U/H$ , é a razão entre o tempo de relaxação do fluido e a escala de tempo do escoamento [3]. O número de Weissenberg [4], é um parâmetro adimensional que mede o grau de anisotropia<sup>3</sup> ou orientação gerada pela deformação, e é definida como a razão entre as forças elásticas e forças viscosas [5].

Quando um método numérico é aplicado para escoamentos de fluidos viscoelásticos, as equações do movimento e da continuidade são acopladas à equação constitutiva do tensor polimérico, ou tensor não-newtoniano (*extra-stress*),  $\boldsymbol{\tau}$  (ou tensor conformação,  $\mathbf{A}$ ). A inclusão da equação constitutiva não apenas aumenta o número total de graus de

---

<sup>1</sup>As suspensões coloidais, ou coloides, representam a situação em que partículas coloidais (mistura de soluto com solvente) estão presentes na água.

<sup>2</sup>Mistura entre dois líquidos imiscíveis, no qual um deles está na forma de finos glóbulos, espalhados em meio ao outro líquido.

<sup>3</sup>A variação de uma propriedade física em relação a direção.

liberdade do problema, mas também modifica o tipo do resultado obtido pelo sistema de equações governantes [6]. Além disso, o caráter evolucionário e a natureza hiperbólica da equação constitutiva exige que a propriedade simétrica e definida positiva (SPD) do tensor conformação seja preservada [6, 7]. Entretanto, o acúmulo de erros numéricos podem, eventualmente, levar à perda desta propriedade, resultando na perda da topologia evolucionária podendo ocasionar as instabilidades de Hadamard [6]. Este colapso numérico, resultante do crescimento de  $Wi$ , é conhecido como o Problema de Alto Número de Weissenberg ou *High Weissenberg Number Problem* (HWNP), e tem sido um grande desafio às simulações numéricas de escoamentos de fluidos viscoelásticos. O HWNP foi primeiramente identificado por uma quebra do esquema numérico para uma equação constitutiva a nível macroscópico. Esta falha numérica para valores moderados do número de Weissenberg/Debora, vem acompanhada de perda de precisão e da não convergência da solução, principalmente quando a geometria do problema envolve cantos ou pontos de estagnação, devido ao crescimento exponencial das tensões normais nestas regiões, caracterizadas por altas taxas de deformação e baixas velocidades. Portanto, a simulação de fluidos viscoelásticos para alta elasticidade necessita de métodos numéricos apropriados para tratar de maneira adequada a equação constitutiva, garantindo estabilidade, convergência e que preserve a positividade do tensor conformação.

Apesar de o HWNP ainda ser um problema em aberto na Reologia Computacional, diversos métodos estabilizadores foram sendo desenvolvidos em anos recentes. Para a simulação direta de um escoamento turbulento, Sureshkumar e Beris [8] introduziram uma tensão difusiva artificial na equação evolutiva do tensor conformação, resultando em diversos bons resultados com o uso de métodos espectrais [9, 10, 11, 12] (ver também [13, 14, 15, 16]). Vaithianathan et al. [17] desenvolveram outro método que também garante a positividade dos autovalores do tensor conformação, enquanto preserva a extensão máxima dos modelos baseados em moléculas, tais como Oldroyd-B, FENE-P e Giesekus. Para a solução numérica foi combinado o método de diferenças finitas com um esquema pseudo-espectral para um escoamento homogêneo turbulento cisalhante.

Para o caso de um escoamentos laminar, e mais especificamente para escoamentos sem inércia ( $Re \sim 0$ ), Lozinski e Owens [18] apresentaram estimativas teóricas e não-lineares para a tensão e as componentes da velocidade em um cenário geral, para um fluido Oldroyd-B. Os autores trabalharam com a configuração do tensor, e derivaram um método que garante um problema bem posto em termos das instabilidades de Hadamard.

Fattal e Kupferman [19, 20] propuseram uma reformulação nas leis constitutivas que descrevem um fluido não-newtoniano viscoelástico, usando uma transformação tensorial aplicada ao tensor conformação. O novo método, conhecido como *log-conformation*, reformula a equação constitutiva do tensor conformação, que é simétrico e definido positivo, aplicando uma transformação logarítmica, resultando na linearização do crescimento exponencial das tensões próximas às regiões de singularidades. Lee e Xu [21] apresentaram

uma classe de esquemas de discretização para preservar a positividade do tensor conformação, estes foram aplicados em equações constitutivas para fluidos viscoelásticos do tipo *rate-type*, usando uma aproximação semi-lagrangiana e um método de elementos finitos, baseado na observação de que as equações constitutivas *rate-type* podem ser transformadas na forma geral das equações diferenciais de Riccati, além disso, este método tem segunda ordem no tempo e no espaço. Cho [22] propôs uma decomposição vetorial para as equações constitutiva do tensor conformação do tipo Maxwell. Nesta transformação, as equações na forma vetorial são consideradas como somas de *dyadics* do tensor conformação. Esta decomposição preserva a positividade do tensor conformação e evita a solução para o problema dos autovalores em todo passo no tempo, reduzindo o custo computacional. Entretanto, no caso 3D, a decomposição vetorial requer o cálculo de nove componentes ao invés de seis componentes independentes como na formulação do *log-conformation*, limitando sua eficiência para o caso tridimensional, quando comparado com a formulação do *log-conformation*. Balci et al. [23] propôs um método em que a raiz quadrada do tensor conformação é usada para os modelos Oldroyd-B e FENE-P. Eles derivaram uma equação de evolução para a raiz quadrada do tensor conformação, aproveitando-se do fato de o tensor conformação ser simétrico e definido positivo, admitindo uma única decomposição do tipo raiz quadrada que satisfaz uma equação de evolução. Balci et al. [23] aproveitaram que este método pode ser facilmente implementado, uma vez que não há necessidade de se calcular os autovalores e autovetores em todo o passo de tempo, resultando em uma significativa redução no tempo de simulação. Recentemente, Afonso et al. [24] propuseram uma formulação genérica, permitindo a aplicação de uma série de transformações matriciais ao tensor conformação de modo a preservar sua positividade. Um método robusto na solução de escoamentos com alto número de Weissenberg foi apresentado por [25], no qual, usou a formulação não-singular do *log-conformation* baseada na solução pelo método de Newton. Outra modificação da formulação original do *log-conformation* foi proposta em [26], onde um método totalmente implícito é usado para resolver a nova equação constitutiva. A aplicação de todos estes métodos estabilizadores tem surtido bons resultados em termos de estabilidades para uma grande variedade de escoamentos de fluido viscoelástico [27, 28, 29, 30, 31, 32, 33, 34].

No presente estudo, empregamos o método estabilizador proposto por Balci et al. [23] para escoamentos confinado e com superfície livre, usando dois códigos numéricos e métodos. Um método de volumes finitos (FVM) [35, 36, 37, 30] é usado para problemas confinados e um método de diferenças finitas (FDM), no contexto do método Marker-and-Cell (MAC) [38, 39], é usado na simulação de escoamentos confinados e com superfície livre. Os resultados obtidos são discutidos em termos de estabilidade e eficiência computacional para escoamentos transientes de fluidos viscoelásticos. Vale destacar que os resultados apresentados neste trabalho foram submetidos no artigo [40]. Além disso, ainda referente ao estudo dos métodos estabilizadores para o Problema de Alto Número



de Weissenberg, propomos um novo método estabilizador, denominado *eigendecomposed kernel-conformation*. Para este, apresentamos sua formulação e validação com o problema do escoamento entre placas paralelas (*poiseuille flow*).

## 1.2 Efeitos de tensão superficial e molhamento

Um assunto de interesse envolvendo escoamentos com superfície livre são os efeitos de tensão superficial e molhamento. A tensão superficial pode ser definida como uma propriedade física que faz com que a superfície de um líquido funcione como uma membrana elástica [41]. É esta característica, por exemplo, a responsável pelo formato esférico de uma gota de água ou o meio pelo qual um inseto consegue caminhar sobre a água. Por outro lado, o fenômeno de molhamento está associado com a maneira como um líquido se deposita sobre uma superfície sólida. O fenômeno de molhamento é quantificado pelo ângulo de contato, ângulo formado entre o líquido e o substrato. Muitas aplicações industriais necessitam da compreensão destes fenômenos, como, *spray* de resfriamento de superfícies quentes, extintores de incêndio, pesticidas, entre outros [42].

Um problema que tem chamado a atenção na área de reologia computacional, envolvendo os fenômenos de tensão superficial e molhamento, é o impacto da gota esférica de fluido viscoelástico numa superfície sólida. Embora muito estudo tenha sido empregue no entendimento deste problema para fluidos newtonianos, o mesmo não é verdade para os fluidos viscoelásticos, onde a quantidade de trabalhos é escassa, seja em termos de experimentação ou simulação numérica. Apesar disso, trabalhos recentes tem revelado crescimento no interesse desse problema [43, 44, 45, 46, 47], tornando-se uma área de pesquisa promissora.

Dentre os diversos métodos numéricos para a simulação de escoamentos viscoelásticos com superfície livre, podemos destacar Crochet e Keunings [48], que foram os pioneiros na simulação de fluidos viscoelásticos com superfície livre, com o uso de um método de elementos finitos, simulando o problema do inchamento do extrudado (*die-swell*) para um fluido Oldroyd-B com o método iterativo *ad-hoc*. Recentemente, Morrison e Harlen [49] estenderam o método *split Lagrangian Eulerian* para o estudo do problema *jet breakup*. Tomé et al. [50, 51] simularam escoamentos com superfície livre usando o método MAC com a técnica de diferenças finitas, para fluidos do tipo Oldroyd-B e PTT. Fang et al. [52] simularam o problema do impacto da gota em duas dimensões, usando o método *smoothed particle hydrodynamics* (SPH), para um fluido do tipo Oldroyd-B. Entretanto, apesar desses esforços na simulação de escoamentos viscoelásticos com superfície livre, todos estes trabalhos negligenciaram os efeitos de tensão superficial e molhamento, quando necessário.

Ainda com respeito à gota viscoelástica, podemos citar algumas aplicações industriais, como os jatos de tinta (para impressoras e processos de pintura automobilística), ensaios

biológicos e, mais recentemente, a fabricação de *microchips* como o sistema micro-analítico total <sup>4</sup> (*lab-on-chip*) [53, 54]. Segundo Singh et al. [55] o correto entendimento do comportamento dos fluidos viscoelásticos é um elemento chave para o crescimento do campo de micro-fabricação. Desta forma, para a correta representação deste problema torna-se imprescindível o emprego dos efeitos de tensão superficial e molhamento.

Em vista da importância da tensão superficial para a correta representação de um escoamento envolvendo superfície livre, algumas técnicas foram desenvolvidas em anos recentes. Por exemplo, Yadigaroglu e Smith [56] apresentaram um método para incluir os efeitos da tensão superficial no método *Piecewise-Linear Interface Construction* (PLIC-VOF). Nesta técnica, eles calculam a curvatura mediante uma função estimadora que é ajustada mediante o método dos mínimos quadrados <sup>5</sup>. Embora tenham apresentado resultados apenas para o caso 2D, declaram ser possível estender para o caso 3D. Renardy e Renardy [57] (ver também [58]) propuseram melhorias ao popular método *Volume-of-fluid* (VOF) visando eliminar os problemas de acurácia e mesmo de não convergência, resultantes da simulação de problemas envolvendo a força capilar, de modo a criar correntes espúrias. Dentre as melhorias deste novo método, denominado *Parabolic reconstruction of surface tension* (PROST), podemos citar: a inclusão de um novo algoritmo para o cálculo da força de corpo, melhorias ao método de projeção usado no cálculo das equações de Navier-Stokes e um esquema de alta ordem para a advecção da interface. Vale mencionar que o cálculo da curvatura em [57] é feito mediante uma aproximação da interface por uma equação quadrática. Recentemente, Tembely et al. [47] apresentaram resultados para o problema da gota viscoelástica com o fluido FENE-CR. Para as simulações, incorporaram ao método VOF os efeitos de tensão superficial e o ângulo de contato dinâmico, usando a correlação de Kistler [59]. Por fim, podemos citar o trabalho de Mangiavachi et al. [60], que empregaram a tensão superficial ao método MAC. Entretanto para as simulações em [60] foi usado apenas o ângulo de contato estático.

Portanto, neste trabalho, propomos uma maneira de empregar os efeitos de tensão superficial e molhamento ao método MAC na sua formulação implícita [61, 62, 63, 64] para fluidos viscoelásticos. Para o emprego dos efeitos de molhamento, apresentamos apenas o ângulo de contato estático, semelhante ao que foi feito em [60].

### 1.3 Organização do trabalho

O texto está organizado da seguinte forma: no Capítulo 2 apresentamos as equações que governam o escoamento de um fluido viscoelástico, bem como as condições auxiliares usadas nas simulações, no Capítulo 3 apresentamos os resultados e discussões com o método estabilizador *square-root*, no Capítulo 4 propomos a construção de um novo

<sup>4</sup>Dispositivo usado para fazer análises em laboratório com pequenas quantidades de fluido

<sup>5</sup>O emprego da tensão superficial exige o cálculo da curvatura.

---

método estabilizador, *eigendecomposed kernel-conformation*, no Capítulo 5 apresentamos uma maneira de incluir os efeitos de tensão superficial e molhamento ao método MAC, e finalmente, no Capítulo 6 apresentamos as considerações finais.

## Equações governantes

As equações que governam um escoamento transiente, incompressível e isotérmico para fluidos viscoelásticos podem ser escritas, de maneira compacta e em sua forma adimensional, como:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) = -\nabla p + \frac{\beta}{Re} \nabla^2 \mathbf{u} + \nabla \cdot \boldsymbol{\tau} + \frac{1}{Fr^2} \mathbf{g}, \quad (2.1)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (2.2)$$

$$\frac{\partial \mathbf{A}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{A}) = \nabla \mathbf{u} \mathbf{A} + \mathbf{A} \nabla \mathbf{u}^T + \frac{1}{Wi} f(\mathbf{A}) P(\mathbf{A}), \quad (2.3)$$

$$\boldsymbol{\tau} = \xi (\mathbf{A} - \mathbf{I}). \quad (2.4)$$

Nestas equações,  $t$  é o tempo,  $\mathbf{u}$  é o vetor velocidade,  $p$  é a pressão,  $\mathbf{g}$  é o campo de gravidade, e  $\boldsymbol{\tau}$  e  $\mathbf{A}$  são o tensor polimérico e o tensor conformação, respectivamente. Os parâmetros adimensionais  $Re = \frac{\rho LU}{\eta_0}$  e  $Fr = U/\sqrt{gL}$  são os números de Reynolds e Froude, respectivamente, onde  $L$  e  $U$  são escalas apropriadas de comprimento e velocidade,  $g$  é a magnitude do campo de gravidade e  $\rho$  é a densidade do fluido. A contribuição newtoniana do solvente é controlada pela número adimensional  $\beta = \frac{\eta_s}{\eta_0}$ , onde  $\eta_0 = \eta_s + \eta_p$  denota a viscosidade total, enquanto  $\eta_s$  e  $\eta_p$  representam a viscosidade do solvente e a viscosidade do polímero, respectivamente. O número de Weissenberg na equação (2.3) é definido como  $Wi = \lambda U/L$ , onde  $\lambda$  é o tempo de relaxação do fluido. A variável  $\xi$  e as funções  $f(\mathbf{A})$  e  $P(\mathbf{A})$ , escalar e vetorial, respectivamente, nas equações (2.3) e (2.4) são definidas de acordo com modelo constitutivo desejado. Note que o tensor polimérico está normalizado como  $\boldsymbol{\tau} = \boldsymbol{\tau}' / (\rho U^2)$  onde  $\boldsymbol{\tau}'$  representa o tensor polimérico dimensional. Para a pressão, usamos a seguinte normalização,  $p = p' / (\rho U^2)$ .

As condições iniciais usadas para resolver o sistema constituído das equações (2.1-2.4) são:  $\mathbf{u} = 0$  e  $\mathbf{A} = \mathbf{I}$ . Ademais, as condições de contorno na região de entrada do fluido, na região de saída do fluido, nas paredes sólidas e na superfície livre são resumidas como:

- Região de entrada de fluido (*inlets*): a componente normal da velocidade é especificada ( $u_n = u_{inl}$ ), enquanto que a velocidade tangencial é tomada como sendo zero ( $u_t = 0$ ). Além disso, o tensor polimérico é calculado assumindo que o escoamento esteja completamente desenvolvido. Assim, uma vez calculado o valor de  $\boldsymbol{\tau}$ , o tensor conformação pode ser calculado pela relação (2.4).
- Região de saída de fluido (*outlets*): condições de Neumann homogêneo são aplicadas ao campo de velocidade e ao tensor polimérico.
- Paredes sólidas (*walls*): a condição *no-slip* é usada ( $\mathbf{u} = \mathbf{u}_{wall}$ ) para o campo de velocidade.
- Superfície livre (*Moving free surface*): na ausência de tensões superficiais, as componentes normal e tangencial do tensor das tensões total precisam ser contínuas através da superfície, ou seja,

$$\mathbf{n} \cdot (\boldsymbol{\sigma} \cdot \mathbf{n}) = 0, \quad (2.5)$$

$$\mathbf{m} \cdot (\boldsymbol{\sigma} \cdot \mathbf{n}) = 0, \quad (2.6)$$

onde  $\boldsymbol{\sigma}$  é o tensor tensão total, dado por:

$$\boldsymbol{\sigma} = -p\mathbf{I} + \beta \frac{2}{Re} \mathbf{D} + \boldsymbol{\tau}, \quad (2.7)$$

e  $\mathbf{D}$  é o tensor taxa de deformação, definido por  $\mathbf{D} = \frac{1}{2} (\nabla \mathbf{u} + \nabla \mathbf{u}^T)$ . Nas equações (2.5) e (2.6),  $\mathbf{n}$  representa o vetor unitário normal à superfície livre, e  $\mathbf{m}$  é o vetor unitário tangencial à superfície livre.

A equação (2.3) é aplicada nas células de superfície livre para obtermos o tensor conformação, e depois,  $\boldsymbol{\tau}$  é calculado por (2.4).

# Estudo do método estabilizador square-root

Neste capítulo, apresentamos o método proposto por Balci et al. [23] para escoamentos confinados e com superfície livre. Para as simulações usamos dois códigos numéricos e métodos diferentes. Um código para volumes finitos (FVM), cujos resultados foram obtidos pelos colaboradores do artigo [40] ( ver também [35, 36, 37, 30]), usado exclusivamente para escoamentos confinados, e um código para diferenças finitas (FDM), empregue por nós, desenvolvido no contexto do método MAC, usado para simulação de escoamentos confinados e com superfície livre. A formulação *square-root*, bem como a formulação *log-conformation*, foram implementadas em ambos os esquemas numéricos e os resultados obtidos são discutidos em termos de estabilidade e eficiência, para escoamentos de fluidos viscoelásticos transientes.

## 3.1 Metodologia

Na formulação square root proposta por Balci et al. [23], o tensor conformação é decomposto como:

$$\mathbf{A} = \mathbf{Q}^T \mathbf{Q}. \quad (3.1)$$

A ideia básica deste método é a construção de uma equação de evolução para a (única) matriz simétrica  $\mathbf{Q}$ . Substituindo a equação (3.1) em (2.3), resulta em

$$\mathbf{Q} \frac{D\mathbf{Q}}{Dt} = -\frac{D\mathbf{Q}}{Dt} \mathbf{Q} + \nabla \mathbf{u} \mathbf{Q}^2 + \mathbf{Q}^2 \nabla \mathbf{u}^T + \frac{1}{Wi} f(\mathbf{Q}^2) \mathbf{P}(\mathbf{Q}^2) \quad (3.2)$$

Multiplicando a equação (3.2) por  $\mathbf{Q}^{-1}$ , nos conduz à seguinte equação

$$\frac{D\mathbf{Q}}{Dt} = -\mathbf{Q}^{-1} \frac{D\mathbf{Q}}{Dt} \mathbf{Q} + \mathbf{Q}^{-1} \nabla \mathbf{u} \mathbf{Q}^2 + \mathbf{Q} \nabla \mathbf{u}^T + \frac{1}{W_i} f(\mathbf{Q}^2) \mathbf{Q}^{-1} \mathbf{P}(\mathbf{Q}^2) \quad (3.3)$$

que pode ser reescrita como

$$\begin{aligned} \frac{D\mathbf{Q}}{Dt} - \mathbf{Q} \nabla \mathbf{u}^T - \frac{1}{2W_i} f(\mathbf{Q}^2) \mathbf{Q}^{-1} \mathbf{P}(\mathbf{Q}^2) &= -\mathbf{Q}^{-1} \frac{D\mathbf{Q}}{Dt} \mathbf{Q} + \mathbf{Q}^{-1} \nabla \mathbf{u} \mathbf{Q}^2 + \frac{1}{2W_i} f(\mathbf{Q}^2) \mathbf{Q}^{-1} \mathbf{P}(\mathbf{Q}^2) \\ &= \mathbf{Q}^{-1} \left( -\frac{D\mathbf{Q}}{Dt} + \nabla \mathbf{u} \mathbf{Q} + \frac{1}{2W_i} f(\mathbf{Q}^2) \mathbf{P}(\mathbf{Q}^2) \mathbf{Q}^{-1} \right) \mathbf{Q} \end{aligned} \quad (3.4)$$

Definimos

$$\mathbf{V} \equiv \frac{D\mathbf{Q}}{Dt} - \mathbf{Q} \nabla \mathbf{u}^T - \frac{1}{2W_i} f(\mathbf{Q}^2) \mathbf{Q}^{-1} \mathbf{P}(\mathbf{Q}^2), \quad (3.5)$$

e notando que  $\mathbf{V}$  é um tensor antissimétrico, isto é,  $\mathbf{V} = -\mathbf{V}^T$ , a equação (3.4) pode ser reescrita como função de  $\mathbf{V}$  e  $\mathbf{Q}$  da seguinte forma:

$$\mathbf{V} = \mathbf{Q}^{-1} (-\mathbf{V}^T) \mathbf{Q}, \quad (3.6)$$

ou de maneira equivalente como

$$\mathbf{V} \mathbf{Q}^{-1} = -(\mathbf{V} \mathbf{Q}^{-1})^T. \quad (3.7)$$

Uma observação importante aqui é o fato de  $\mathbf{V} \mathbf{Q}^{-1}$  ser uma matriz antissimétrica.

Finalmente, introduzimos a matriz antissimétrica

$$\mathbf{G} = -\mathbf{Q}^{-1} \mathbf{V}^T \quad (3.8)$$

na equação (3.4), obtendo a seguinte equação de evolução

$$\frac{D\mathbf{Q}}{Dt} = \mathbf{G} \mathbf{Q} + \mathbf{Q} \nabla \mathbf{u}^T + \frac{1}{2W_i} f(\mathbf{Q}^2) \mathbf{Q}^{-1} \mathbf{P}(\mathbf{Q}^2). \quad (3.9)$$

Para resolver a equação (3.9), precisamos escrever uma expressão para  $\mathbf{G}$ . O procedimento usado neste trabalho, para definir a matriz antissimétrica, é o mesmo descrito em [23]. Primeiro, a matriz  $\mathbf{K}$  é definida como

$$\mathbf{K} = \mathbf{G} \mathbf{Q} + \mathbf{Q} \nabla \mathbf{u}^T, \quad (3.10)$$

em seguida, a matriz  $\mathbf{G}$  é construída impondo que  $\mathbf{K}$  seja uma matriz simétrica

$$\mathbf{K}^T = \mathbf{K} \Leftrightarrow \nabla \mathbf{u} \mathbf{Q} - \mathbf{Q} \nabla \mathbf{u}^T = \mathbf{G} \mathbf{Q} + \mathbf{Q} \mathbf{G}. \quad (3.11)$$

Por exemplo, considerando o caso bidimensional em coordenadas cartesianas, a equação (3.11) pode ser usada para obter

$$G_{12} = \frac{\left(Q_{12} \frac{\partial u}{\partial x} + Q_{22} \frac{\partial u}{\partial y}\right) - \left(Q_{11} \frac{\partial v}{\partial x} + Q_{12} \frac{\partial v}{\partial y}\right)}{Q_{11} + Q_{22}}. \quad (3.12)$$

Enquanto que para o caso tridimensional, pode-se calcular  $\mathbf{G}$  a partir do sistema

$$\begin{pmatrix} Q_{22} + Q_{11} & Q_{23} & -Q_{13} \\ Q_{23} & Q_{33} + Q_{11} & Q_{12} \\ -Q_{13} & Q_{12} & Q_{33} + Q_{22} \end{pmatrix} \begin{pmatrix} G_{12} \\ G_{13} \\ G_{23} \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \quad (3.13)$$

onde

$$a = \left(\frac{\partial u}{\partial x} Q_{12} - \frac{\partial v}{\partial x} Q_{11}\right) + \left(\frac{\partial u}{\partial y} Q_{22} - \frac{\partial v}{\partial y} Q_{12}\right) + \left(\frac{\partial u}{\partial z} Q_{23} - \frac{\partial v}{\partial z} Q_{13}\right) \quad (3.14)$$

$$b = \left(\frac{\partial u}{\partial x} Q_{13} - \frac{\partial w}{\partial x} Q_{11}\right) + \left(\frac{\partial u}{\partial y} Q_{23} - \frac{\partial w}{\partial y} Q_{12}\right) + \left(\frac{\partial u}{\partial z} Q_{33} - \frac{\partial w}{\partial z} Q_{13}\right) \quad (3.15)$$

$$c = \left(\frac{\partial v}{\partial x} Q_{13} - \frac{\partial w}{\partial x} Q_{12}\right) + \left(\frac{\partial v}{\partial y} Q_{23} - \frac{\partial w}{\partial y} Q_{22}\right) + \left(\frac{\partial v}{\partial z} Q_{33} - \frac{\partial w}{\partial z} Q_{23}\right) \quad (3.16)$$

Mais detalhes sobre a construção geral das equações em 3D podem ser encontrados em [33].

## 3.2 Método numérico

O método MAC apresentado neste trabalho foi introduzido primeiramente em [38] (ver também [39]). Nesta seção, descrevemos as modificações introduzidas pela formulação *square-root* no contexto de diferenças finitas.

Inicialmente, um desacoplamento entre velocidade e pressão é aplicado. Este método de projeção é amplamente usada na solução das equações de Navier-Stokes (2.1)-(2.2) [65].

A principal modificação introduzida no método de Oishi et al. [38] é a incorporação da equação (3.9) para o tensor raiz quadrada  $\mathbf{Q}$  ao invés de resolver diretamente a equação constitutiva para o tensor polimérico  $\boldsymbol{\tau}$ , ou mesmo a equação do tensor conformação  $\mathbf{A}$ . Desta forma, note que a equação (3.9) pode ser reescrita como



$$\frac{\partial \mathbf{Q}}{\partial t} = \mathbf{z}(\mathbf{u}, \mathbf{Q}, \mathbf{G}) \quad (3.17)$$

onde

$$\mathbf{z}(\mathbf{u}, \mathbf{Q}, \mathbf{G}) = -\nabla \cdot (\mathbf{u}\mathbf{Q}) + \mathbf{G}\mathbf{Q} + \mathbf{Q}\nabla \mathbf{u}^T + \frac{1}{2W_i} f(\mathbf{Q}^2) \mathbf{Q}^{-1} \mathbf{P}(\mathbf{Q}^2). \quad (3.18)$$

Considerando o esquema Runge-Kutta de segunda ordem para a discretização temporal da equação (3.9), temos

$$\frac{\mathbf{Q}^{(n+1)} - \mathbf{Q}^{(n)}}{\delta t} = \frac{1}{2} \left[ \mathbf{z}(\mathbf{u}^{(n)}, \mathbf{Q}^{(n)}, \mathbf{G}^{(n)}) + \mathbf{z}(\mathbf{u}^{(n+1)}, \overline{\mathbf{Q}}^{(n+1)}, \overline{\mathbf{G}}^{(n+1)}) \right], \quad (3.19)$$

onde o valor intermediário  $\overline{\mathbf{Q}}^{(n+1)}$  é calculado com o método de Euler explícito:

$$\frac{\overline{\mathbf{Q}}^{(n+1)} - \mathbf{Q}^{(n)}}{\delta t} = \mathbf{z}(\mathbf{u}^{(n)}, \mathbf{Q}^{(n)}, \mathbf{G}^{(n)}). \quad (3.20)$$

A matriz antissimétrica no estágio intermediário  $\overline{\mathbf{G}}^{(n+1)}$  é calculada por

$$\overline{\mathbf{G}}^{(n+1)} = \frac{\left( \overline{Q}_{12} \frac{\partial u^{(n+1)}}{\partial x} + \overline{Q}_{22} \frac{\partial u^{(n+1)}}{\partial y} \right) - \left( \overline{Q}_{11} \frac{\partial v^{(n+1)}}{\partial x} + \overline{Q}_{12} \frac{\partial v^{(n+1)}}{\partial y} \right)}{\overline{Q}_{11} + \overline{Q}_{22}} \quad (3.21)$$

De modo similar,  $G_{12}^n$  é construído com todos os termos discretizados no nível de tempo  $t = t_n$ .

Portanto, o novo algoritmo, incorporando a formulação *square-root* para o cálculo do tensor conformação, contém os seguintes passos:

1. Dados  $\mathbf{Q}^{(n)}$ ,  $\mathbf{u}^{(n)}$  e  $\mathbf{G}^{(n)}$ , resolvemos a equação de evolução (3.20) para obter o valor intermediário  $\overline{\mathbf{Q}}^{(n+1)}$ ;
2. A partir de  $\overline{\mathbf{Q}}^{(n+1)}$ , calculamos o valor intermediário do tensor conformação usando a equação (3.1). Então, a equação (2.4) é aplicada para o cálculo do tensor polimérico  $\overline{\boldsymbol{\tau}}^{(n+1)}$  que é usado para o cálculo da velocidade intermediária pela equação do momento (sub-passo do método de projeção, ver Capítulo 5, subseção 5.1.1);
3. Aplicamos o método de projeção para obter a velocidade final  $\mathbf{u}^{(n+1)}$  e o campo de pressão  $p^{(n+1)}$  (para mais detalhes ver [38]);
4. Construímos o tensor  $\overline{\mathbf{G}}^{(n+1)}$  pela equação (3.21);
5. Calculamos o valor do tensor raiz quadrada  $\mathbf{Q}^{(n+1)}$ . Neste estágio, o valor final do tensor conformação e do tensor polimérico são obtidos usando as equações (3.1) e (2.4), respectivamente.

**Observação 3.1.** Para iniciar o algoritmo é imposta uma condição inicial homogênea e isotrópica, isto é,  $Q_{t=t_0}^2 = I$ .

**Observação 3.2.** Para o escoamento na superfície livre, o último passo do algoritmo é a advecção da superfície livre. Cada partícula é transportada por um campo de velocidade, a partir de sua posição  $\mathbf{x}^{(n)}$  em  $t = t_n$  para a posição  $\mathbf{x}^{(n+1)}$  em  $t = t_{n+1}$  como

$$\frac{d\mathbf{x}}{dt} = \mathbf{u} \quad (3.22)$$

Esta equação é resolvida usando o esquema RK21 de segunda ordem como descrito em [38].

### 3.3 Validação: escoamento em uma cavidade

Nesta seção, apresentamos a validação da implementação do código para diferenças finitas, juntamente com os resultados para o método de volumes finitos. O problema teste ou de referência (*benchmark*) usado é o escoamento em uma cavidade com tampa móvel [66]. Este escoamento é gerado pelo movimento de uma ou mais paredes de uma cavidade fechada. Existem diversos experimentos e estudos numéricos envolvendo escoamentos em uma cavidade, principalmente para fluidos newtonianos [67]. Por outro lado, para fluidos viscoelásticos o interesse é recente, e tem sido usado, principalmente, para avaliar métodos numéricos para escoamentos com alta elasticidade [20, 68, 69, 70, 71, 72], que requer uma regularização sobre o movimento da tampa devido ao comportamento singular próximos aos cantos.

O problema padrão requer a seguinte regularização para o perfil parabólico na tampa  $u(x, t) = 8[1 + \tanh(8t - 4)]x^2(1 - x)^2$ . As demais paredes permanecem estáticas e a condição de não escorregamento (*no-slip*) é empregue às paredes. Para as simulações, foi fixado o valor de alguns parâmetros, como o número de Reynolds,  $Re = 0.01$ , e a razão entre a viscosidade do solvente e a viscosidade total  $\beta = 0.5$ .

Para avaliar a convergência de ambos os métodos, diferenças finitas (FDM) e volumes finitos (FVM), o escoamento na cavidade foi simulado usando duas malhas uniformes: M1 ( $h = \min(\Delta x, \Delta y) = \frac{1}{128}$ ,  $128 \times 128$  células) e M2 ( $h = \min(\Delta x, \Delta y) = \frac{1}{256}$ ,  $256 \times 256$  células).

Para as figuras desta seção, o perfil da componente  $u$  da velocidade e a componente  $\tau_{11}$  do tensor polimérico foram plotadas ao longo da linha vertical  $x = 0.5$ , enquanto que a componente  $v$  da velocidade foi plotada ao longo da linha horizontal  $y = 0.75$ . A origem do sistema cartesiano foi definida no canto inferior esquerdo da cavidade, e a tampa móvel definida em  $y = 1$ , a partir de  $x = 0$  até  $x = 1$ .

A fim de avaliar as implementações dos códigos, os resultados obtidos foram comparados com resultados da literatura, apresentados por Fattal e Kupferman [20] e Pan

et al. [68]. Os resultados são apresentados nas Figuras 1 e 2 para  $Wi = 1$  e  $Wi = 2$ , respectivamente.

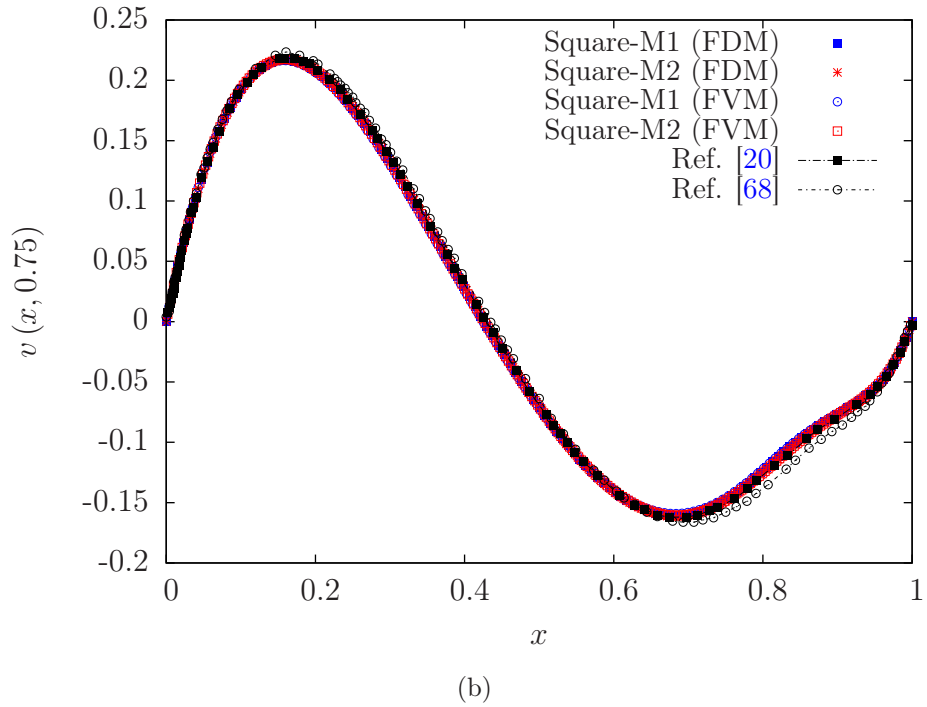
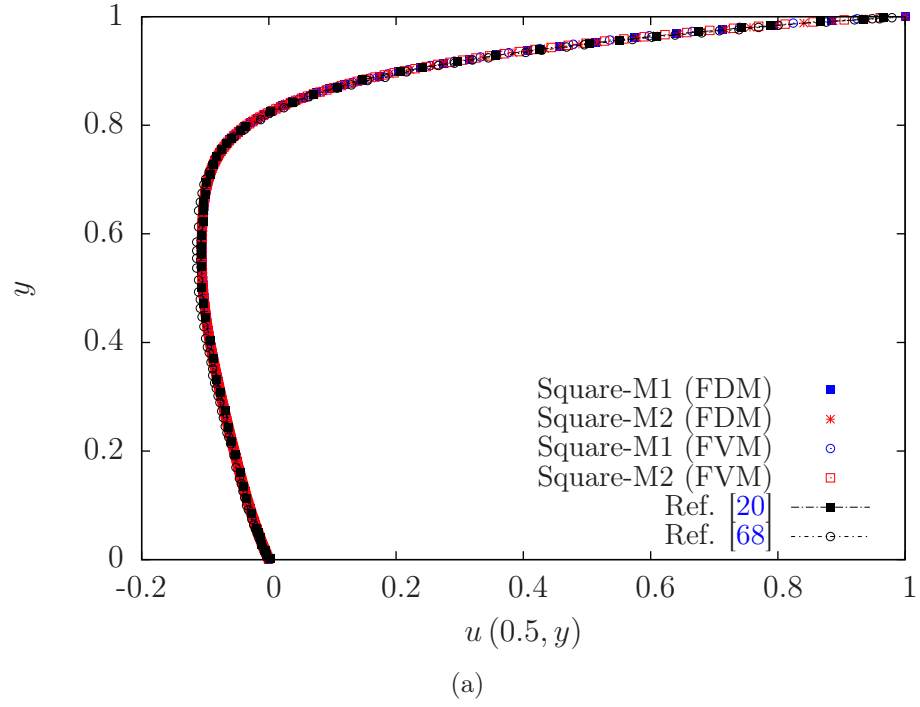
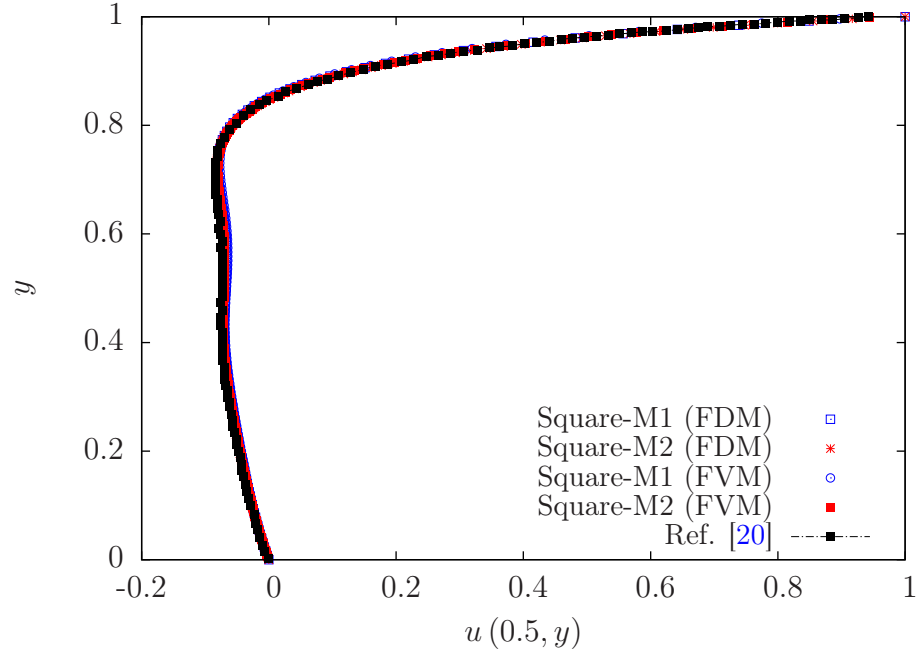
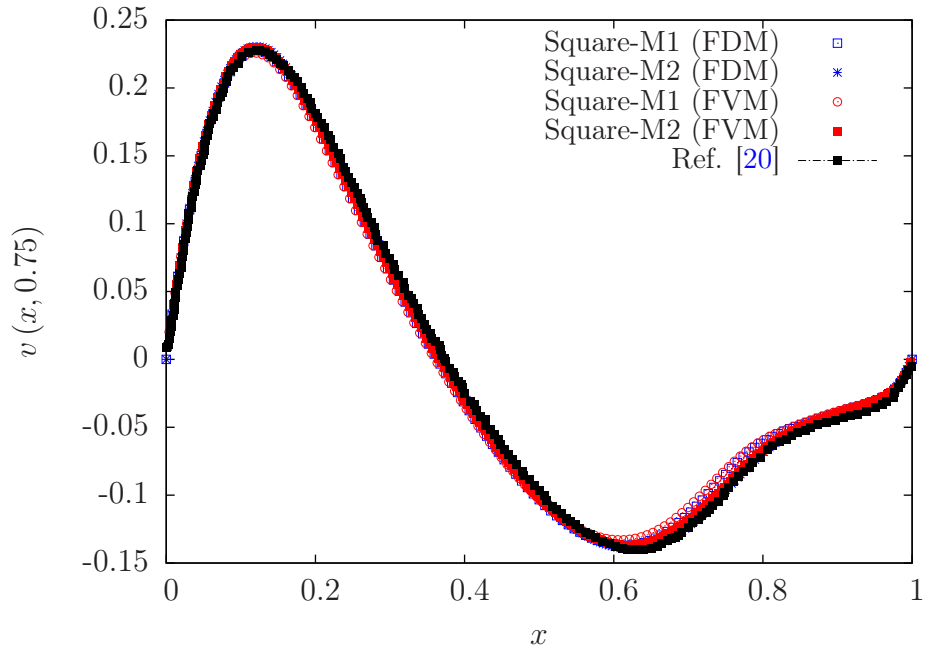


Figura 1: Simulação numérica do escoamento em uma cavidade usando o modelo Oldroyd-B para  $Re = 0.01$ ,  $Wi = 1$  e  $\beta = 0.5$ . Os resultados foram tomados em  $t = 40$ , correspondendo ao estado estacionário do escoamento.

Na Figura 1, foi plotado as componentes  $u$  e  $v$  da velocidade para  $Wi = 1$  no tempo adimensional  $t = 40$ , mostrando que o método *square-root* converge de acordo com os



(a)



(b)

Figura 2: Simulação numérica do escoamento em uma cavidade usando o modelo Oldroyd-B para  $Re = 0.01$ ,  $Wi = 2$  e  $\beta = 0.5$ . Os resultados foram tomados em  $t = 80$ , correspondendo ao estado estacionário do escoamento.

resultados da literatura. Para  $Wi = 2$ , a simulação foi tomada em  $t = 80$ , e de acordo com a Figura 2 os resultados, novamente, apresentam concordância com a literatura.

Ademais, apresentamos dados adicionais para este problema de referência. Foi plotado na Figura 3 a componente  $\tau_{11}$  do tensor polimérico ao longo da linha vertical  $x = 0.5$ ,

ilustrando a convergência com o refinamento de malha do método *square-root* para  $Wi = 1$  e  $Wi = 2$ . Note que, a componente  $\tau_{11}$  do tensor polimérico foi normalizada como  $\tau_{11} = \tau'_{11} / (\rho U^2)$ , onde  $U$  é a velocidade máxima na tampa da cavidade.

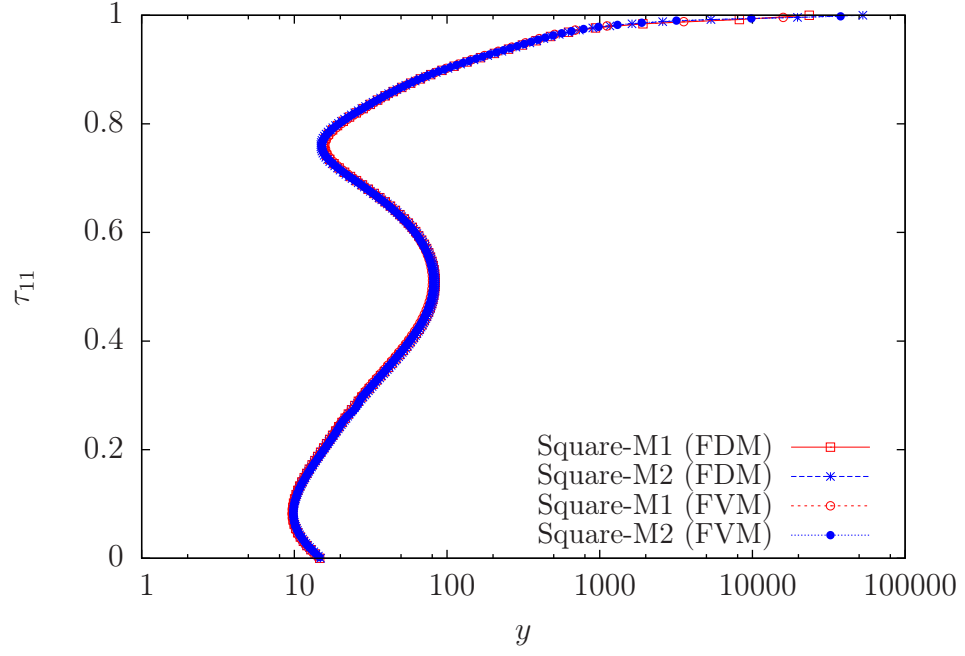
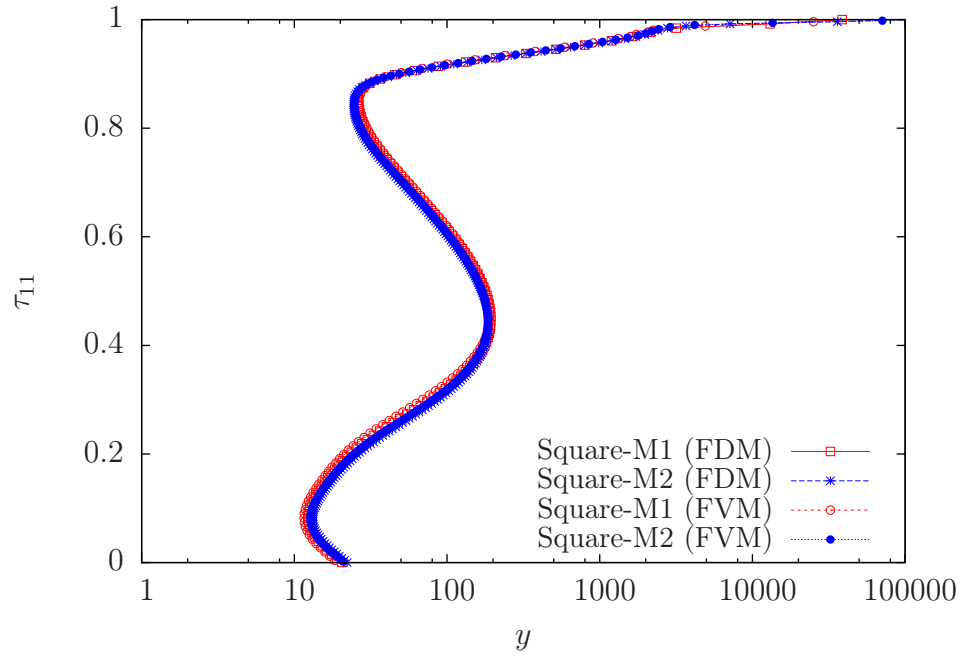
(a)  $Wi = 1$ (b)  $Wi = 2$ 

Figura 3: Simulação numérica do escoamento em uma cavidade usando o modelo Oldroyd-B para  $Re = 0.01$  e  $\beta = 0.5$ . A tensão normal  $\tau_{11}$  é plotada ao longo da linha vertical  $x = 0.5$ .

Além disso, apresentamos na Figura 4 a evolução temporal da energia cinética,  $E = \iint v^2 dx dy$ , para este problema de referência, mostrando novamente que a formulação *square-root* produz resultados similares com a literatura. Além disso, nesta mesma figura foi adicionado os resultados adquiridos com a formulação *log-conformation*. No contexto de diferenças finitas, a versão *log-conformation* nestas simulações foi usada recentemente em [34], enquanto que a versão em volumes finitos foi descrita em [30].

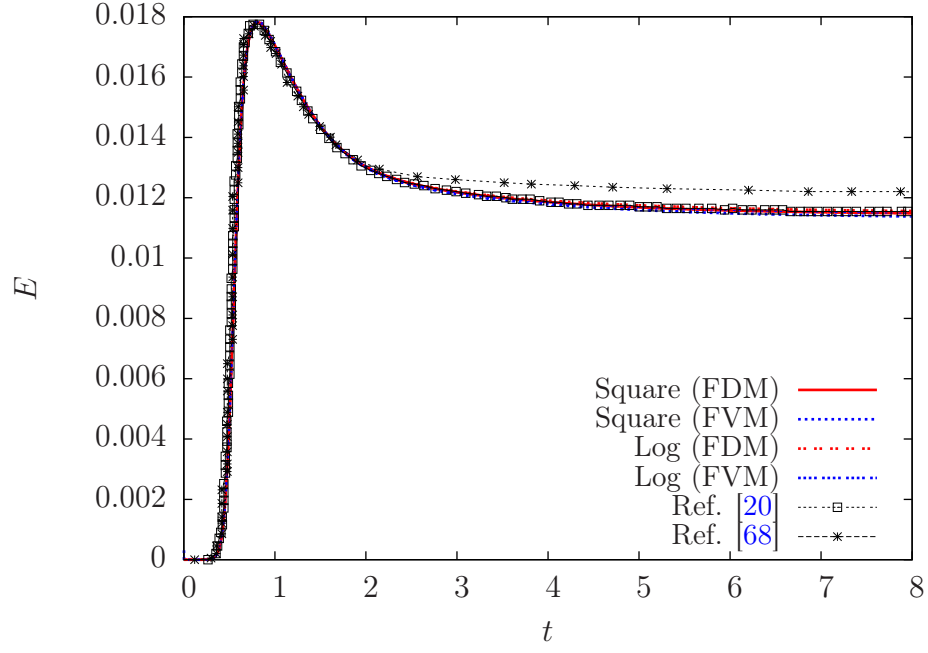
## 3.4 Aplicações

### 3.4.1 Escoamento confinado: Cross-Slot

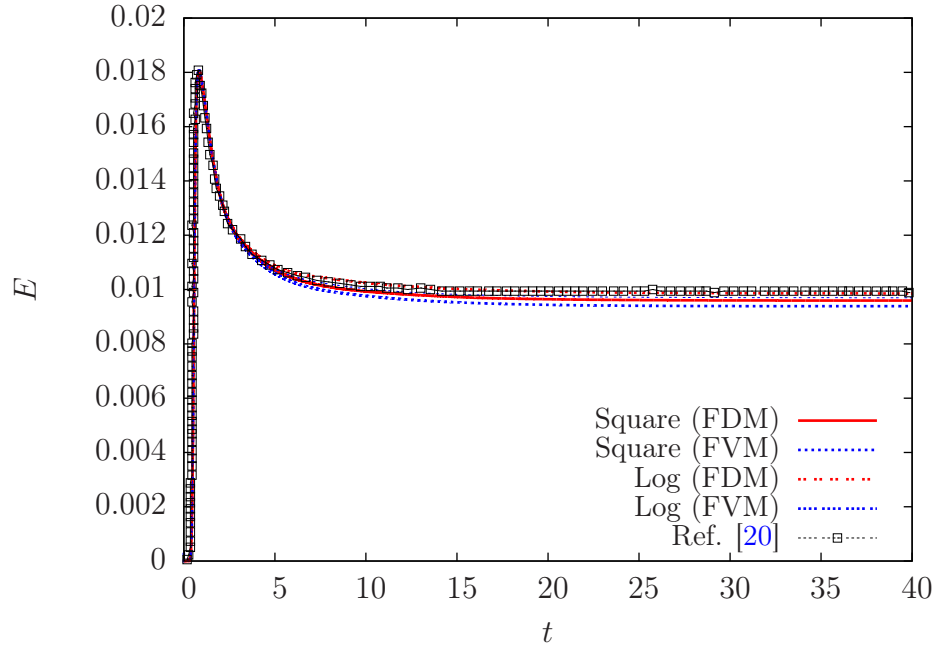
O escoamento em uma fenda transversal (*cross-slot*), e outros escoamentos extensionais<sup>1</sup>, tais como o escoamento em torno de quatro cilindros circulares em movimento de rotação (*four roll mill*) e o aparelho de jatos opostos (*opposed jet apparatus*) tem sido amplamente investigados devido à necessidade de se desenvolver métodos para a medição da viscosidade extensional de soluções poliméricas [73].

Hoje sabemos que para um escoamento de fluido viscoelástico é comum o surgimento de instabilidades elásticas pura, acarretando em um escoamento com irregularidades, mesmo quando as forças inerciais advectivas são menores que as forças viscosas, o chamado *creeping flow*. Um trabalho relevante e motivador para o estudo das instabilidades elásticas foi apresentado por Arratia et al. [74], no qual, mediante resultados experimentais de um escoamento em um microcanal, apresentou o surgimento de instabilidades elásticas no escoamento em uma fenda transversal. Além disso, este trabalho motivou o trabalho numérico de Poole et al. [75] que simulou um escoamento bidimensional em uma fenda transversal para o fluido *Upper-Convected Maxwell* (UCM) usando baixos valores de Reynolds. Poole et al. [75] puderam prever o primeiro tipo de instabilidade (estacionária e assimétrica) em um escoamento bidimensional em uma fenda transversal para um fluido do tipo UCM, conduzindo a uma redução da perda de pressão, além de apresentar o efeito estabilizador da inércia. Este problema, *cross-slot*, foi recentemente proposto como um problema de referência por Cruz et al. [76], devido a sua geometria simples e a, bem definida, instabilidade estacionária assimétrica. Resultados para este problema de referência já foram apresentados para uma ampla variedade de modelos constitutivos do tipo UCM, por exemplo, Oldroyd-B e Phan-Thien and Tanner (PTT), mostrando que no limite quando a inércia é desprezada, ou seja, quando o  $Re$  tende a zero, o fluido com alta viscosidade extensional apresenta dois tipos de instabilidades elásticas pura. Para valores de Deborah ( $De$ ) acima do valor crítico o escoamento torna-se assimétrico, mesmo a geometria do problema sendo perfeitamente simétrica; além disso, para altos valores de

<sup>1</sup>Um escoamento extensional é definido pelo movimento contrário, distanciamento, entre as partículas adjacentes do fluido.



(a)



(b)

Figura 4: Evolução temporal da energia cinética para as formulações *square-root* (Square) e *log-conformation* (Log) comparadas com resultados da literatura. As simulações foram feitas com o modelo Oldroyd-B para  $Re = 0.01$  e  $\beta = 0.5$ : 4(a)  $Wi = 1$  e 4(b)  $Wi = 2$ .

De surge a segunda instabilidade, que é a persistência do escoamento em um comportamento transiente. Recentemente, Cruz e Pinho [77] apresentaram uma solução analítica para o caso bidimensional de escoamentos extensionais, planar estacionário e com pontos de estagnação, com o modelo UCM.

No que segue, apresentamos os resultados numéricos das simulações, para ambos os códigos, diferenças finitas e volumes finitos, com as formulações *square-root* e *log-conformation*. O modelo usado nas simulações é o Oldroyd-B, com  $\beta = 1/9$  e  $Re = 0.01$ . A malha usada nas simulações possui 51 células na região do quadrado central do *cross-slot*, o que conduz à  $\Delta x_{min} = \Delta y_{min} \approx 0.02$ . Para o modelo Oldroyd-B, o valor local do número de Weissenberg,  $Wi = \lambda \dot{\epsilon}$ , no ponto de estagnação pode exceder o valor crítico teórico de  $1/2$  (isto ocorre quando  $De \geq 0.2$  [76]). Assim, as tensões normais no ponto de estagnação podem tornar-se ilimitadas, desde que o tempo de simulação seja infinitamente grande. Este fato pode ser usado para verificar a acurácia do método numérico baseando-se nos valores preditos das tensões normais no ponto de estagnação. A Figura 5 apresenta os resultados obtidos para a primeira diferença entre as tensões normais na forma dimensional,  $N'_1 = (\tau'_{22} - \tau'_{11}) / (\eta_0 U / D)$ , ao longo do eixo  $x$  na posição central  $y' = 0$ , com ambas as formulações, *square-root* e *log-conformation*, para  $De = 0.1, 0.2, 0.3$  e  $0.35$ . Podemos observar que para as simulações onde as tensões extensional tornam-se ilimitadas, isto é, quando  $De \geq 0.2$ , a primeira diferença entre as tensões normais para ambos os métodos apresentam pequenas diferenças, mesmo havendo altos gradientes na vizinhança do ponto de estagnação.

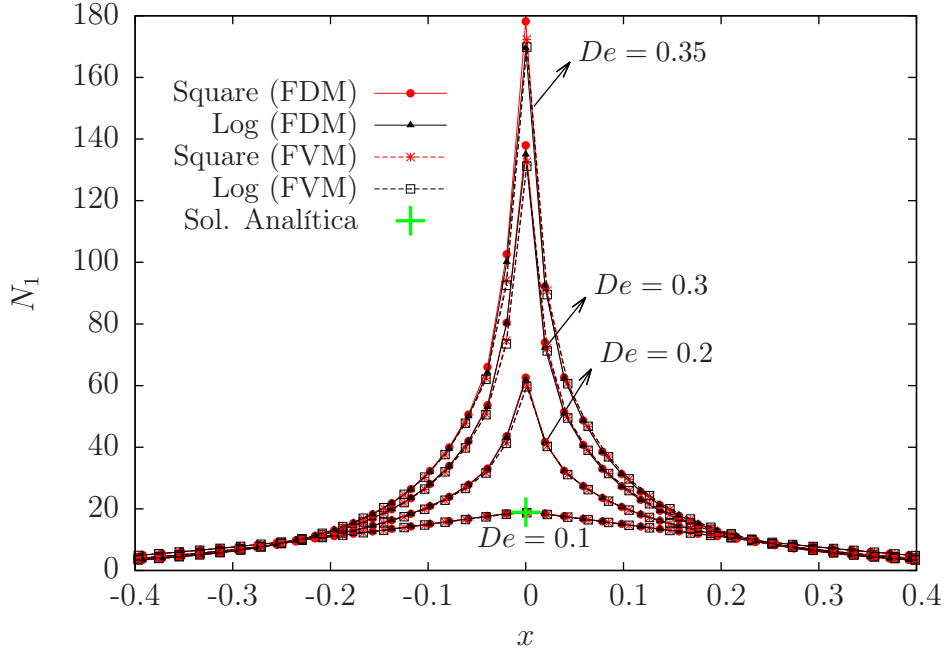


Figura 5: Primeira diferença entre as normais poliméricas,  $(\tau'_{22} - \tau'_{11}) / (\eta_0 U / D)$ , em função de  $x$  em  $y = 0$  para as formulações *log-conformation* e *square-root*:  $De = 0.1, 0.2, 0.3$  e  $0.35$ .

Os resultados obtidos das simulações do *cross-slot* foram comparados com os resultados de Cruz et al. [76], além disso, foi usado o seguinte parâmetro para medir a assimetria do



escoamento [75]:

$$DQ = \frac{Q_2 - Q_1}{Q_2 + Q_1}. \quad (3.23)$$

A taxa total do escoamento por unidade de profundidade em cada entrada de fluido,  $Q = Q_2 + Q_1$ , divide-se em duas correntes de saída,  $Q_1$  e  $Q_2$ . Assim, para um escoamento simétrico  $DQ = 0$ , enquanto que para um escoamento completamente assimétrico, temos  $|DQ| = 1$ , ou seja, o fluido inserido por uma das entradas do canal sai completamente por apenas uma das saídas do canal. Os resultados obtidos para  $DQ$  e  $Wi$ , avaliados no ponto de estagnação, usando as formulações *square-root* e *log-conformation* são apresentados na Tabela 3.1.

Tabela 3.1: Dados para o modelo Oldroyd-B ( $DQ$  e  $Wi$ ), com  $\beta = 1/9$ , cross-slot.

		$De = 0.1$	$De = 0.2$	$De = 0.3$	$De = 0.35$	$De = 0.4$	$De = 0.42$
FVM	$DQ_{log}$	0.000	0.000	0.000	0.000	0.555	0.652
	$DQ_{sqrt}$	0.000	0.000	0.000	0.000	0.554	0.653
FDM	$DQ_{log}$	0.000	0.000	0.000	0.000	0.563	0.660
	$DQ_{sqrt}$	0.000	0.000	0.000	0.000	0.563	0.661
	$DQ$ Ref. [76]	0.000	0.000	0.000	0.000	0.562	0.666
FVM	$Wi_{log}$	0.317	0.517	0.589	0.601	0.506	0.493
	$Wi_{sqrt}$	0.317	0.517	0.589	0.600	0.505	0.493
FDM	$Wi_{log}$	0.313	0.516	0.583	0.576	0.500	0.486
	$Wi_{sqrt}$	0.319	0.516	0.579	0.587	0.502	0.489
	$Wi$ Ref. [76]	0.322	0.522	0.591	0.602	0.497	0.487

Podemos observar que para valores do número de Debora acima do primeiro valor crítico,  $De_{crit} = 0.37$ , o escoamento exibi a instabilidade elástica pura, no qual, o escoamento não atinge o estado estacionário e apresenta assimetria, conduzindo a valores de  $DQ$  não nulos. Para ambas as reformulações da equação do tensor conformação e ambos os métodos (FDM e FVM) os valores de  $DQ$  e  $Wi$  estão próximos dos valores obtidos por Cruz et al. [76] usando um refinamento de malha semelhante. Quando comparado com os valores extrapolados sugeridos por Cruz et al. [76], notamos uma leve diferença nos valores, isto ocorre por causa da malha mais grossa usada nas simulações deste trabalho.

### 3.4.2 Escoamento com superfície livre: impacto da gota

Para este problema, calculamos a evolução temporal de uma gota, com formato circular e em duas dimensões, caindo pela ação da força gravitacional a partir de uma distância  $H$  acima de uma superfície sólida. A Figura 6 ilustra o formato da gota e o campo de velocidade  $u$  em diferentes intervalos de tempo para  $Wi = 1$ .

Algumas dificuldades envolvendo este problema são as instabilidades numéricas para alto nível de elasticidade no escoamento, resultando em uma alta deformação na interface entre o líquido e o ar. Estas características, fazem deste problema um dos mais populares

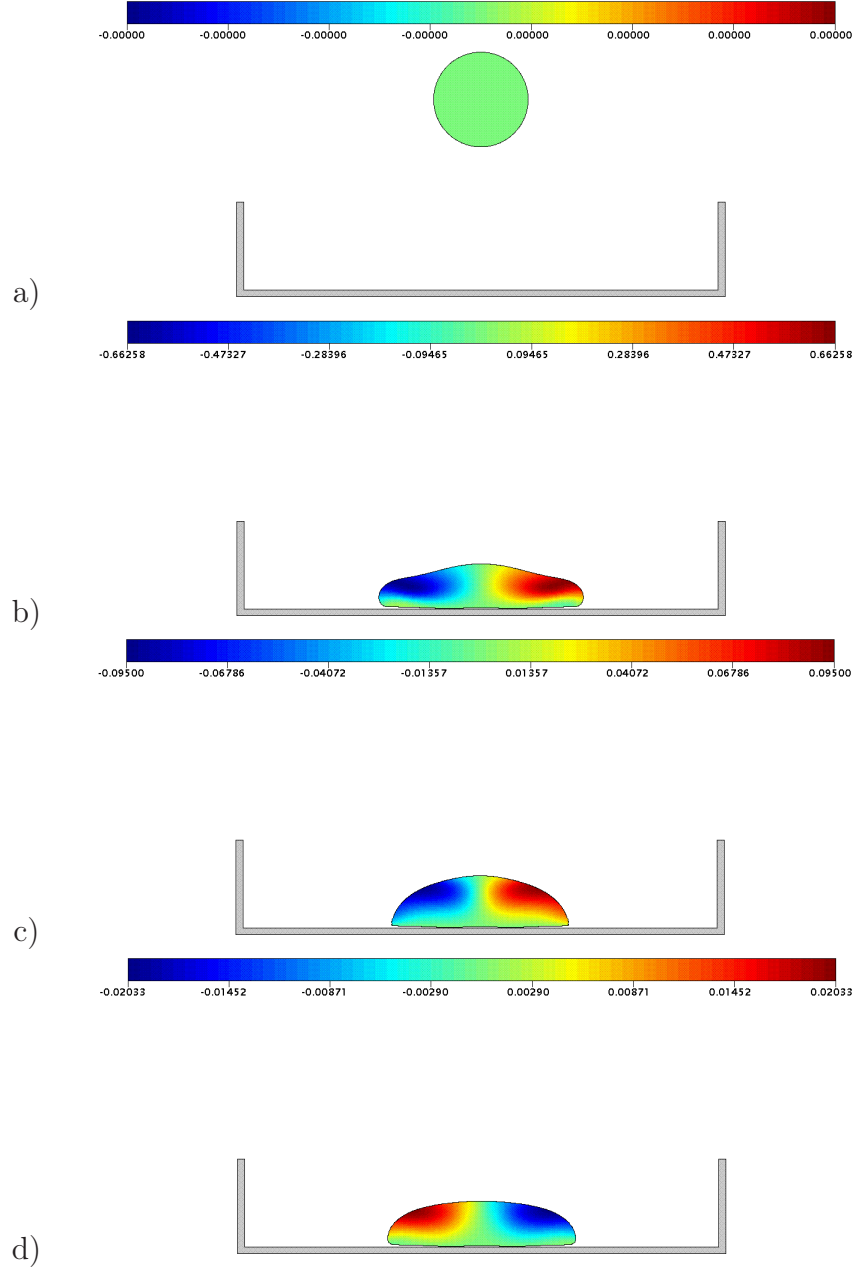


Figura 6: Simulação numérica do problema do impacto de uma gota para o modelo Oldroyd-B ( $Re = 5.0$ ;  $We = 1.0$ ;  $\beta = 0.1$ ;  $Fr = 2.26$ ). Ilustração da gota e seu contorno para a componente  $u$  da velocidade em diferentes tempos: a)  $t = 0$ , b)  $t = 2.0$ , c)  $t = 4.0$ , d)  $t = 6.0$ .

problemas de referência para escoamentos de fluidos viscoelásticos com superfície livre [78, 79, 52, 80]. Entretanto, é escasso na literatura trabalhos envolvendo escoamentos para altos valores de Weissenberg, devido às dificuldades numéricas que envolvem este tipo de escoamento. Desta forma, a fim de obter mais informações e entendimento a respeito da física que envolve este escoamento com superfície livre, será apresentado a solução deste problema com a formulação *square-root* para altos valores de  $Wi$ .

Os parâmetros usados neste estudo são:  $D = 0.02$  m,  $U = 1$  m/s e  $|g| = 9.81$  m/s<sup>2</sup>. Consideramos a gota com velocidade inicial  $v_0 = -1$  m/s a uma distância  $H = 0.04$  m medida do centro da gota até a superfície de impacto.

Primeiramente, verificamos a implementação do código. Para isto, o problema foi simulado com o modelo Oldroyd-B considerando  $Re = 5.0$ ,  $Fr = 2.26$ ,  $Wi = 1.0$  e  $\beta = 0.1$ . Além disso, adotamos três malhas diferentes:  $M1$  com  $h = 0.025$ ,  $M2$  com  $h = 0.0125$  e  $M3$  com  $h = 0.00625$ , onde  $h$  representa o espaçamento entre os pontos da malha, uniformemente espaçados. A Figura 7 apresenta a comparação entre os resultados obtidos e alguns valores da literatura para o diâmetro adimensional da gota. Os resultados com a formulação *square-root*, para as três malhas, mostraram-se com boa convergência com o refinamento de malha, e bom comportamento quando comparado com os outros métodos.

A fim de avaliar a influência da viscoelasticidade do fluido sobre o impacto da gota, a Figura 8 apresenta, para a malha  $M2$ , a evolução temporal do diâmetro da gota para diferentes valores de  $Wi$ . Nesta figura, podemos ver também a inclusão dos resultados para o fluido newtoniano ( $Wi = 0$ ), considerando  $Re = 5$  e  $Re = 50$ . Podemos notar que o crescimento de  $Wi$  conduz ao crescimento do diâmetro da gota, ou seja um espalhamento maior do fluido, como também pode ser observado quando comparado com a Figura 6.

Pode parecer pouco intuitivo, mas para baixos valores de  $Wi$  o crescimento da tensão normal após o impacto é mais rápido (devido ao baixo tempo de relaxação) e isto conduz a um efeito de recuo do fluido devido a viscoelasticidade. O aumento de  $Wi$  causa um aumento nas tensões normais, mas isto ocorre somente para longos tempos e produz resistência por causa da tensão normal no tempo inicial depois do impacto. Em resultado disso, o diâmetro da gota não sofre muitas mudanças para  $Wi \geq 50$ , como ocorreu para valores de  $Wi$  baixos. Em particular, para valores alto de  $Wi$  a tensão polimérica não tem tempo para crescer durante o rápido processo de deformação após o impacto da gota, de modo que o fluido responde de maneira similar ao caso Newtoniano com viscosidade  $\beta\eta_0$ , assim a curva para  $Wi = 500$  se aproxima do caso Newtoniano ( $Wi = 0$ ) com um efetivo número de Reynolds de  $Re = 5/\beta = 50$ , como mostra a Figura 6.

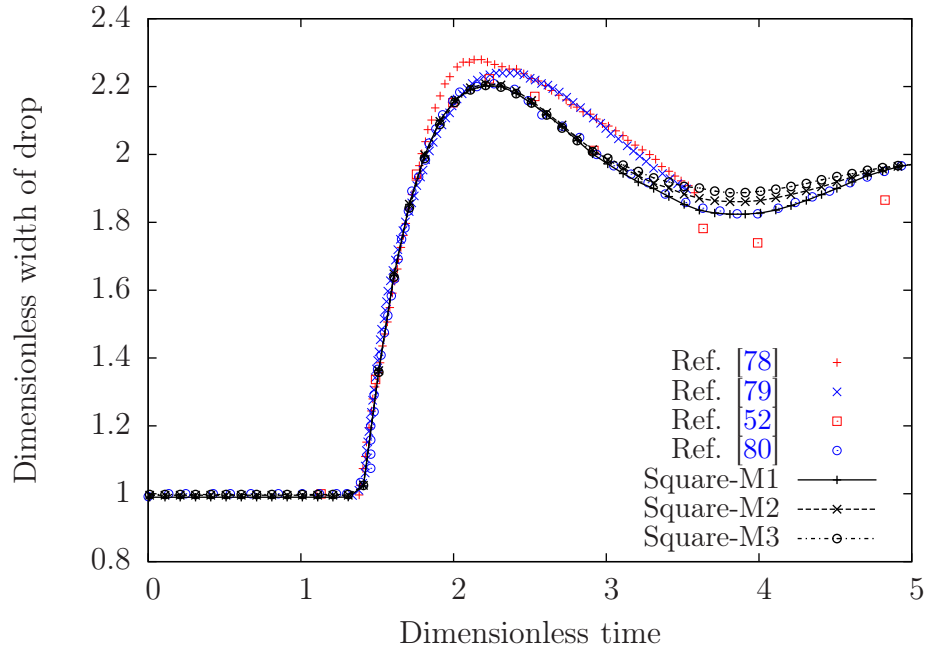


Figura 7: Resultados numéricos da variação temporal do diâmetro normalizada ( $W/L$ ) da gota de fluido Oldroyd-B para  $Re = 5.0$ ,  $Wi = 1.0$ ,  $\beta = 0.1$  e  $Fr = 2.26$ .

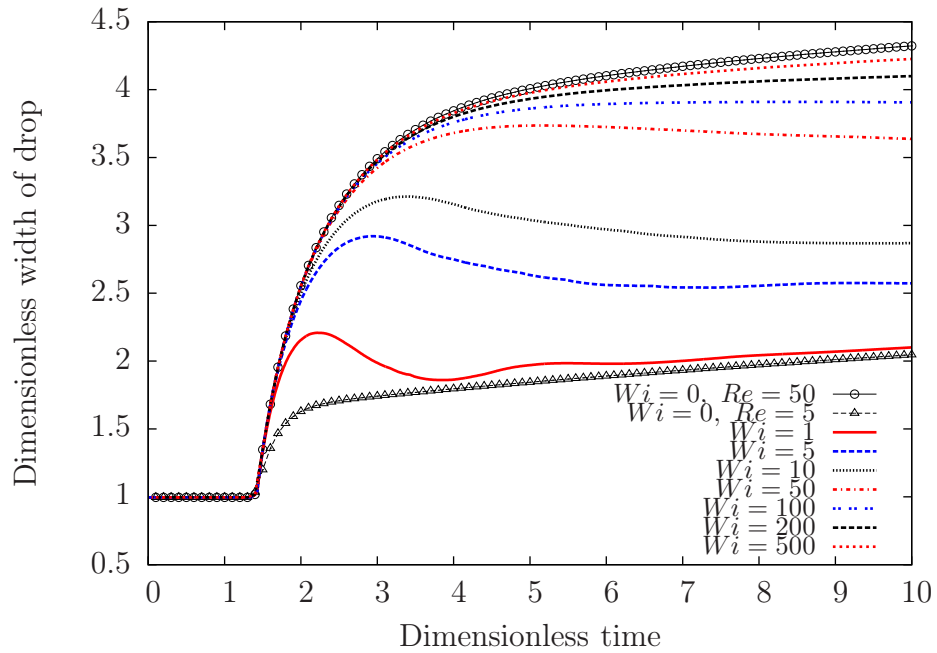


Figura 8: Influência do número  $Wi$  no comportamento do diâmetro normalizado da gota de fluido Oldroyd-B para  $Re = 5.0$ ,  $\beta = 0.1$  e  $Fr = 2.26$ . Os resultados para o caso newtoniano também foram incluídos ( $Wi = 0$ ,  $Re = 5$  e  $Re = 50$ ).

# Estudo do método estabilizador eigendecomposed kernel-conformation

Neste capítulo, apresentamos uma nova formulação para os métodos estabilizadores para escoamentos com alto número de Weissenberg. Faremos uma junção de dois destes métodos, *kernel-conformation*, apresentado por Afonso et al. [24], e *continuous eigendecomposition*, proposto por Vaithianathan e Collins [81], com algumas modificações, visando otimizar sua eficiência. A grande novidade deste método híbrido, além de incorporar estes dois métodos, que chamamos de *eigendecomposed kernel-conformation*, é o modo como calculamos os autovetores, inovando a aplicação da transformação de Cayley usada em [81]. No que segue, apresentamos a formulação do método *eigendecomposed kernel-conformation*, o método numérico e alguns resultados preliminares.

## 4.1 Metodologia

O tensor conformação  $\mathbf{A}$  pode ser decomposto como

$$\mathbf{A} = \mathbf{O}\mathbf{\Lambda}\mathbf{O}^T, \quad (4.1)$$

onde  $\mathbf{O}$  é uma matriz ortogonal, constituída por autovetores ortonormais, e  $\mathbf{\Lambda}$  é a matriz dos autovalores de  $\mathbf{A}$ . Além disso, o tensor  $\nabla\mathbf{u}$  também pode ser decomposto:

- decomposição em uma parcela simétrica e outra antissimétrica [81]:

$$\nabla\mathbf{u} = \mathbf{D} + \mathbf{R}, \quad (4.2)$$

onde  $\mathbf{D} = \frac{1}{2} (\nabla \mathbf{u} + \nabla \mathbf{u}^T)$  é o tensor deformação, matriz simétrica, e  $\mathbf{R} = \frac{1}{2} (\nabla \mathbf{u} - \nabla \mathbf{u}^T)$  é o tensor rotação, matriz antissimétrica.

- decomposição baseado no trabalho de Fattal e Kupferman [19]:

$$\nabla \mathbf{u} = \mathbf{\Omega} + \mathbf{B} + \mathbf{N} \mathbf{A}^{-1}, \quad (4.3)$$

onde  $\mathbf{\Omega}$  e  $\mathbf{N}$  são matrizes antissimétricas e  $\mathbf{B}$  é uma matriz simétrica que comuta com  $\mathbf{A}$ .

Substituindo a equação (4.2) na equação do tensor conformação (2.3), obtemos uma nova equação para  $\mathbf{A}$ :

$$\frac{D\mathbf{A}}{Dt} = (\mathbf{D}\mathbf{A} + \mathbf{A}\mathbf{D}) + (\mathbf{R}\mathbf{A} - \mathbf{A}\mathbf{R}) + \frac{1}{W_i} f(\mathbf{A}) \mathbf{P}(\mathbf{A}), \quad (4.4)$$

ao passo que, substituindo (4.3) em (2.3), obtemos

$$\frac{D\mathbf{A}}{Dt} = (\mathbf{\Omega}\mathbf{A} - \mathbf{A}\mathbf{\Omega}) + 2\mathbf{B}\mathbf{A} + \frac{1}{W_i} f(\mathbf{A}) \mathbf{P}(\mathbf{A}). \quad (4.5)$$

A equação do tensor conformação (4.4), escrita em função de  $\mathbf{D}$  e  $\mathbf{R}$ , foi apresentada por Vathianathan e Collins [81], enquanto que a equação (4.5) foi proposta por Fattal e Kupferman [19].

A partir da equação (4.1), temos que  $\mathbf{\Lambda} = \mathbf{O}^T \mathbf{A} \mathbf{O}$ . Desta forma, podemos escrever a derivada total de  $\mathbf{\Lambda}$  como:

$$\frac{D\mathbf{\Lambda}}{Dt} = \frac{D\mathbf{O}^T}{Dt} \mathbf{A} \mathbf{O} + \mathbf{O}^T \frac{D\mathbf{A}}{Dt} \mathbf{O} + \mathbf{O}^T \mathbf{A} \frac{D\mathbf{O}}{Dt}. \quad (4.6)$$

Substituindo (4.1) e (4.4) em (4.6), resulta na seguinte equação de evolução para  $\mathbf{\Lambda}$ :

$$\frac{D\mathbf{\Lambda}}{Dt} = (\mathbf{\Lambda}\mathbf{H} - \mathbf{H}\mathbf{\Lambda}) + (\tilde{\mathbf{D}}\mathbf{\Lambda} + \mathbf{\Lambda}\tilde{\mathbf{D}}) + (\tilde{\mathbf{R}}\mathbf{\Lambda} - \mathbf{\Lambda}\tilde{\mathbf{R}}) + \frac{1}{W_i} f(\mathbf{\Lambda}) \mathbf{P}(\mathbf{\Lambda}), \quad (4.7)$$

onde  $\tilde{\mathbf{D}} = \mathbf{O}^T \mathbf{D} \mathbf{O}$ ,  $\tilde{\mathbf{R}} = \mathbf{O}^T \mathbf{R} \mathbf{O}$  e  $\mathbf{H}$  é definido como

$$\mathbf{H} = \mathbf{O}^T \frac{D\mathbf{O}}{Dt}. \quad (4.8)$$

Além disso, note que  $\mathbf{H}$  é uma matriz antissimétrica. De fato,

$$\mathbf{0} = \frac{D(\mathbf{O}^T \mathbf{O})}{Dt} = \mathbf{O}^T \frac{D\mathbf{O}}{Dt} + \frac{D\mathbf{O}^T}{Dt} \mathbf{O}, \quad (4.9)$$

que implica  $\mathbf{H} = -\mathbf{H}^T$ .

Por outro lado, substituindo (4.1) e (4.5) em (4.6), temos outra equação evolutiva para  $\Lambda$ :

$$\frac{D\Lambda}{Dt} = (\Lambda\mathbf{H} - \mathbf{H}\Lambda) + (\tilde{\Omega}\Lambda - \Lambda\tilde{\Omega}) + 2\tilde{\mathbf{B}}\Lambda + \frac{1}{W_i}f(\Lambda)\mathbf{P}(\Lambda), \quad (4.10)$$

onde  $\tilde{\Omega} = \mathbf{O}^T\Omega\mathbf{O}$  e  $\tilde{\mathbf{B}} = \mathbf{O}^T\mathbf{B}\mathbf{O}$ .

Observe que, na equação (4.7) as parcelas  $(\Lambda\mathbf{H} - \mathbf{H}\Lambda)$  e  $(\tilde{\mathbf{R}}\Lambda - \Lambda\tilde{\mathbf{R}})$  possuem, cada uma delas, a diagonal principal com entradas nulas, logo não contribuem para a evolução temporal de  $\Lambda$ . Assim, podemos dividir a equação (4.7) em outras duas equações:

$$\begin{cases} \frac{D\Lambda}{Dt} = (\tilde{\mathbf{D}}\Lambda + \Lambda\tilde{\mathbf{D}}) + \frac{1}{W_i}f(\Lambda)\mathbf{P}(\Lambda) \\ (\Lambda\mathbf{H} - \mathbf{H}\Lambda) + (\tilde{\mathbf{D}}\Lambda + \Lambda\tilde{\mathbf{D}}) + (\tilde{\mathbf{R}}\Lambda - \Lambda\tilde{\mathbf{R}}) = \mathbf{0} \end{cases} \quad (4.11)$$

ou em coordenadas cartesianas bidimensionais:

$$\begin{cases} \frac{D\Lambda_{11}}{Dt} = 2\Lambda_{11}\tilde{D}_{11} + \frac{1}{W_i}f(\Lambda)P(\Lambda)_{11} \\ \frac{D\Lambda_{22}}{Dt} = 2\Lambda_{22}\tilde{D}_{22} + \frac{1}{W_i}f(\Lambda)P(\Lambda)_{22} \\ H_{12} = \frac{\tilde{D}_{12}(\Lambda_{11} + \Lambda_{22}) + \tilde{R}_{12}(\Lambda_{22} - \Lambda_{11})}{\Lambda_{22} - \Lambda_{11}} \end{cases} \quad (4.12)$$

Da mesma forma, observando que a equação (4.10) possui parcelas, as matrizes  $(\Lambda\mathbf{H} - \mathbf{H}\Lambda)$  e  $(\tilde{\Omega}\Lambda - \Lambda\tilde{\Omega})$ , tais que a diagonal principal de cada uma delas tem entradas nulas, podemos dividir a equação (4.10) em outras duas equações:

$$\begin{cases} \frac{D\Lambda}{Dt} = 2\tilde{\mathbf{B}}\Lambda + \frac{1}{W_i}f(\Lambda)\mathbf{P}(\Lambda) \\ (\Lambda\mathbf{H} - \mathbf{H}\Lambda) + (\tilde{\Omega}\Lambda - \Lambda\tilde{\Omega}) = \mathbf{0} \end{cases} \quad (4.13)$$

ou em coordenadas cartesianas bidimensionais:

$$\begin{cases} \frac{D\Lambda_{11}}{Dt} = 2\tilde{B}_{11}\Lambda_{11} + \frac{1}{W_i}f(\Lambda)P(\Lambda)_{11} \\ \frac{D\Lambda_{22}}{Dt} = 2\tilde{B}_{22}\Lambda_{22} + \frac{1}{W_i}f(\Lambda)P(\Lambda)_{22} \\ H_{12} = \tilde{\Omega}_{12} \end{cases} \quad (4.14)$$

Observe que as equações (4.12) e (4.14) fornecem-nos duas formas de evoluir os autovalores do tensor conformação  $\mathbf{A}$ , além de fornecer-nos o valor da matriz  $\mathbf{H}$ . Ademais, mediante a definição de  $\mathbf{H}$ , equação (4.8), temos que

$$\frac{D\mathbf{O}}{Dt} = \mathbf{O}\mathbf{H}, \quad (4.15)$$

ou em coordenadas cartesianas bidimensionais:

$$\left\{ \begin{array}{l} \frac{DO_{11}}{Dt} = -O_{12}H_{12} \\ \frac{DO_{12}}{Dt} = O_{11}H_{12} \\ \frac{DO_{21}}{Dt} = -O_{22}H_{12} \\ \frac{DO_{22}}{Dt} = O_{21}H_{12} \end{array} \right. \quad (4.16)$$

Embora seja possível evoluir a matriz dos autovetores  $\mathbf{O}$  pelo conjunto de equações em (4.16), esta não é a melhor estratégia, pois como observado por Vaithianathan e Collins [81], a matriz  $\mathbf{O}$  perde sua propriedade ortogonal com o passar do tempo, resultando na não convergência da solução. Assim, faz-se necessário o uso de alguma estratégia que mantenha a ortogonalidade de  $\mathbf{O}$  com o passar do tempo. Para tanto, será usado a transformação de Cayley [82, 83].

#### 4.1.1 Transformação de Cayley

Qualquer matriz ortogonal  $\mathbf{O}$ , tal que nenhum de seus autovalores seja  $-1$ , pode ser escrita como [82]:

$$\mathbf{O} = (\mathbf{I} - \mathbf{S})^{-1} (\mathbf{I} + \mathbf{S}) \quad (4.17)$$

onde  $\mathbf{S}$  é uma matriz antissimétrica.

Além disso, a matriz  $\mathbf{S}$  também está relacionada com  $\mathbf{O}$  pela relação:

$$\mathbf{S} = (\mathbf{O} - \mathbf{I}) (\mathbf{I} + \mathbf{O})^{-1} \quad (4.18)$$

**Observação 4.1.** *Os autovalores de  $\mathbf{O}$  precisam ser simultaneamente diferentes de  $-1$  para que seja possível o cálculo de  $(\mathbf{I} + \mathbf{O})^{-1}$ . Por exemplo, para o caso em duas dimensões, temos:*

$$\det(\mathbf{I} + \mathbf{O}) = O_{11}O_{22} + O_{11} + O_{22} + 1 - O_{12}O_{21} = O_{11} + O_{22} + 2, \quad (4.19)$$



assim, como  $\text{tr}(O) = O_{11} + O_{22} = d_1 + d_2$ , tal que  $d_i$  é autovalor de  $O$ , precisa-se ter  $d_1 = d_2 \neq -1$ .

Substituindo a relação (4.17) na equação de evolução para  $O$  (4.15), temos que

$$\frac{D[(\mathbf{I} - \mathbf{S})^{-1}(\mathbf{I} + \mathbf{S})]}{Dt} = [(\mathbf{I} - \mathbf{S})^{-1}(\mathbf{I} + \mathbf{S})] \mathbf{H}. \quad (4.20)$$

Pela regra do produto para derivadas, podemos reescrever (4.20) como

$$\frac{D(\mathbf{I} - \mathbf{S})^{-1}}{Dt}(\mathbf{I} + \mathbf{S}) + (\mathbf{I} - \mathbf{S})^{-1} \frac{D(\mathbf{I} + \mathbf{S})}{Dt} = [(\mathbf{I} - \mathbf{S})^{-1}(\mathbf{I} + \mathbf{S})] \mathbf{H}, \quad (4.21)$$

ou ainda

$$(\mathbf{I} - \mathbf{S}) \frac{D(\mathbf{I} - \mathbf{S})^{-1}}{Dt}(\mathbf{I} + \mathbf{S}) + \frac{D(\mathbf{I} + \mathbf{S})}{Dt} = (\mathbf{I} + \mathbf{S}) \mathbf{H}. \quad (4.22)$$

Note que,

$$\frac{D[(\mathbf{I} - \mathbf{S})(\mathbf{I} - \mathbf{S})^{-1}(\mathbf{I} + \mathbf{S})]}{Dt} = \frac{D(\mathbf{I} + \mathbf{S})}{Dt}, \quad (4.23)$$

e, novamente, pela regra do produto para derivadas, temos que

$$(\mathbf{I} - \mathbf{S}) \frac{D(\mathbf{I} - \mathbf{S})^{-1}}{Dt}(\mathbf{I} + \mathbf{S}) = -\frac{D(\mathbf{I} - \mathbf{S})}{Dt}(\mathbf{I} - \mathbf{S})^{-1}(\mathbf{I} + \mathbf{S}). \quad (4.24)$$

Assim, substituindo (4.24) em (4.22) e aplicando algumas manipulações algébricas, obtemos

$$\frac{D\mathbf{S}}{Dt} = (\mathbf{I} + \mathbf{S}) \mathbf{H} [(\mathbf{I} - \mathbf{S})^{-1}(\mathbf{I} + \mathbf{S}) + \mathbf{I}]^{-1}, \quad (4.25)$$

ou em coordenadas cartesianas bidimensionais

$$\frac{\partial S_{12}}{\partial t} + \frac{\partial(uS_{12})}{\partial x} + \frac{\partial(vS_{12})}{\partial y} = \frac{H_{12}}{2}(1 + S_{12}^2). \quad (4.26)$$

Desta forma, podemos evoluir a matriz antissimétrica  $\mathbf{S}$  ao invés de  $O$  e, após sua evolução, retornar o valor para  $O$  através da expressão (4.17), garantindo sua ortogonalidade. Vale ressaltar, que esta aplicação da transformação de Cayley é diferente da abordagem empregue por Vaithianathan e Collins [81]. Em [81], os autores aplicaram a transformação de Cayley à discretização temporal, por Euler explícito, da equação (4.15), resultando na necessidade de armazenar valores de  $\mathbf{S}$  em tempos bem anteriores, isto é,

$$\mathbf{O}^{n+1} = \mathbf{O}^0 \left[ \mathbf{I} - \frac{\delta t}{2} \sum_{k=0}^n \mathbf{S}^k \right]^{-1} \left[ \mathbf{I} + \frac{\delta t}{2} \sum_{k=0}^n \mathbf{S}^k \right] + O(\delta t^2), \quad (4.27)$$

onde  $\mathbf{S}^k = \mathbf{H}^k - \mathbf{O}^{T^k} \nabla \cdot (\mathbf{u}^k \mathbf{O}^k)$ .

### 4.1.2 Transformação Kernel

Outra novidade desta abordagem é o uso de transformações matriciais aplicadas à matriz dos autovalores  $\Lambda$ . Desta forma, unimos a eficiência da formulação *kernel-conformation*, proposta por Afonso et al. [24], com as vantagens do método *continuous eigendecomposition*, proposto por Vaithianathan e Collins [81].

Definimos a transformação kernel  $\mathbb{K}$  como:

$$\mathbb{K}(\mathbf{A}) = \mathbf{O}\mathbb{K}(\Lambda)\mathbf{O}^T, \quad (4.28)$$

tal que  $\mathbb{K}$  é uma transformação contínua, inversível e diferenciável.

Observe que, aplicar a transformação  $\mathbb{K}$  em  $\mathbf{A}$  é o mesmo que aplicar aos autovalores  $\Lambda$ . Além disso, os autovetores  $\mathbf{O}$  de  $\mathbf{A}$ , associados aos autovalores  $\Lambda$ , também são autovetores de  $\mathbb{K}(\mathbf{A})$ , associados aos autovalores  $\mathbb{K}(\Lambda)$ . Baseados nesta observação, podemos construir uma equação de evolução para  $\mathbb{K}(\Lambda)$ , sem nos preocupar com a equação de evolução para  $\mathbf{O}$  (4.15) que permanecerá a mesma.

Aplicando a derivada total em  $\mathbb{K}(\Lambda)$ , temos:

$$\frac{D\mathbb{K}(\Lambda)}{Dt} = \frac{D\Lambda}{Dt} \frac{\partial \mathbb{K}(\Lambda)}{\partial \Lambda}. \quad (4.29)$$

Mas, já conhecemos  $\frac{D\Lambda}{Dt}$  de (4.11) e de (4.13). Assim, a equação para a transformação kernel será:

$$\frac{D\mathbb{K}(\Lambda)}{Dt} = \left[ 2\Lambda\tilde{\mathbf{D}} + \frac{1}{W_i}f(\Lambda)\mathbf{P}(\Lambda) \right] \mathbf{J}, \quad (4.30)$$

ou

$$\frac{D\mathbb{K}(\Lambda)}{Dt} = \left[ 2\Lambda\tilde{\mathbf{B}} + \frac{1}{W_i}f(\Lambda)\mathbf{P}(\Lambda) \right] \mathbf{J}, \quad (4.31)$$

onde

$$\mathbf{J} = \frac{\partial \mathbb{K}(\Lambda)}{\partial \Lambda} \quad (4.32)$$

Como apresentado em [24], diversas transformações matriciais podem ser aplicadas em  $\Lambda$ , como, logaritmo natural, raiz quadrada, linear, seno-hiperbólico, entre outras. Para exemplificar a aplicação destas transformações, apresentamos alguns exemplos:

- Linear,

$$\mathbb{K}(\Lambda) = \xi(\Lambda - \mathbf{I}), \quad (4.33)$$

cuja transformação inversa é

$$\Lambda = \frac{1}{\xi}\mathbb{K}(\Lambda) + \mathbf{I}, \quad (4.34)$$

e

$$J_{ii} = \xi. \quad (4.35)$$

- Logaritmo na base  $a > 0$ ,

$$\mathbb{K}(\Lambda) = \log_a(\Lambda), \quad (4.36)$$

cuja transformação inversa é

$$\Lambda = a^{\mathbb{K}(\Lambda)}, \quad (4.37)$$

e

$$J_{ii} = \frac{\Lambda_{ii}^{-1}}{\ln(a)} \quad (4.38)$$

- Raiz  $k$ -ésima,

$$\mathbb{K}(\Lambda) = \Lambda^{1/k}, \quad (4.39)$$

cuja transformação inversa é

$$\Lambda = \mathbb{K}(\Lambda)^k \quad (4.40)$$

e

$$J_{ii} = \frac{1}{k} \Lambda_{ii}^{(1-k)/k}. \quad (4.41)$$

## 4.2 Método Numérico

Nesta seção, descrevemos as modificações introduzidas ao método MAC pela formulação *eigendecomposed kernel-conformation* no contexto de diferenças finitas. Apresentamos, em conjunto, as duas possibilidades de se evoluir  $\mathbb{K}(\Lambda)$ , pela equação (4.30) ou pela equação (4.31).

Primeiramente reescrevemos as equações (4.30) e (4.31), respectivamente, como

$$\frac{\partial \mathbb{K}(\Lambda)}{\partial t} = \mathbf{z}(\mathbf{u}, \mathbb{K}(\Lambda), \Lambda, \tilde{\mathbf{D}}, \mathbf{J}) \quad (4.42)$$

e

$$\frac{\partial \mathbb{K}(\Lambda)}{\partial t} = \mathbf{z}(\mathbf{u}, \mathbb{K}(\Lambda), \Lambda, \tilde{\mathbf{B}}, \mathbf{J}) \quad (4.43)$$

onde

$$\mathbf{z}(\mathbf{u}, \mathbb{K}(\Lambda), \Lambda, \tilde{\mathbf{D}}, \mathbf{J}) = -\nabla \cdot (\mathbf{u} \mathbb{K}(\Lambda)) + \left[ 2\tilde{\mathbf{D}}\Lambda + \frac{1}{W_i} f(\Lambda) \mathbf{P}(\Lambda) \right] \mathbf{J} \quad (4.44)$$

e

$$\mathbf{z}(\mathbf{u}, \mathbb{K}(\Lambda), \Lambda, \tilde{\mathbf{B}}, \mathbf{J}) = -\nabla \cdot (\mathbf{u} \mathbb{K}(\Lambda)) + \left[ 2\tilde{\mathbf{B}}\Lambda + \frac{1}{W_i} f(\Lambda) \mathbf{P}(\Lambda) \right] \mathbf{J}. \quad (4.45)$$

De modo similar, consideremos a equação de evolução para a matriz antissimétrica  $\mathbf{S}$ , reescrita como

$$\frac{\partial \mathbf{S}}{\partial t} = \mathbf{z}(\mathbf{u}, \mathbf{S}, \mathbf{H}), \quad (4.46)$$

onde

$$\mathbf{z}(\mathbf{u}, \mathbf{S}, \mathbf{H}) = -\nabla \cdot (\mathbf{u}\mathbf{S}) + (\mathbf{I} + \mathbf{S}) \mathbf{H} [(\mathbf{I} - \mathbf{S})^{-1} (\mathbf{I} + \mathbf{S}) + \mathbf{I}]^{-1}. \quad (4.47)$$

Considerando o esquema Runge-Kutta de segunda ordem para a discretização temporal das equações (4.42), (4.43) e (4.46), temos, respectivamente,

$$\begin{aligned} & \frac{\mathbb{K}(\boldsymbol{\Lambda})^{(n+1)} - \mathbb{K}(\boldsymbol{\Lambda})^{(n)}}{\delta t} = \\ & = \frac{1}{2} \left[ \mathbf{z} \left( \mathbf{u}^{(n)}, \mathbb{K}(\boldsymbol{\Lambda})^{(n)}, \boldsymbol{\Lambda}^{(n)}, \tilde{\mathbf{D}}^{(n)}, \mathbf{J}^{(n)} \right) + \mathbf{z} \left( \mathbf{u}^{(n+1)}, \bar{\mathbb{K}}(\boldsymbol{\Lambda})^{(n+1)}, \bar{\boldsymbol{\Lambda}}^{(n+1)}, \bar{\tilde{\mathbf{D}}}^{(n+1)}, \bar{\mathbf{J}}^{(n+1)} \right) \right], \end{aligned} \quad (4.48)$$

$$\begin{aligned} & \frac{\mathbb{K}(\boldsymbol{\Lambda})^{(n+1)} - \mathbb{K}(\boldsymbol{\Lambda})^{(n)}}{\delta t} = \\ & = \frac{1}{2} \left[ \mathbf{z} \left( \mathbf{u}^{(n)}, \mathbb{K}(\boldsymbol{\Lambda})^{(n)}, \boldsymbol{\Lambda}^{(n)}, \tilde{\mathbf{B}}^{(n)}, \mathbf{J}^{(n)} \right) + \mathbf{z} \left( \mathbf{u}^{(n+1)}, \bar{\mathbb{K}}(\boldsymbol{\Lambda})^{(n+1)}, \bar{\boldsymbol{\Lambda}}^{(n+1)}, \bar{\tilde{\mathbf{B}}}^{(n+1)}, \bar{\mathbf{J}}^{(n+1)} \right) \right] \end{aligned} \quad (4.49)$$

e

$$\frac{\mathbf{S}^{(n+1)} - \mathbf{S}^{(n)}}{\delta t} = \frac{1}{2} \left[ \mathbf{z} \left( \mathbf{u}^{(n)}, \mathbf{S}^{(n)}, \mathbf{H}^{(n)} \right) + \mathbf{z} \left( \mathbf{u}^{(n+1)}, \bar{\mathbf{S}}^{(n+1)}, \bar{\mathbf{H}}^{(n+1)} \right) \right] \quad (4.50)$$

onde os valores intermediários  $\bar{\mathbb{K}}(\boldsymbol{\Lambda})^{(n+1)}$  e  $\bar{\mathbf{S}}^{(n+1)}$  são calculados, para as equações (4.48), (4.49) e (4.50), respectivamente, com o método de Euler explícito:

$$\frac{\bar{\mathbb{K}}(\boldsymbol{\Lambda})^{(n+1)} - \mathbb{K}(\boldsymbol{\Lambda})^{(n)}}{\delta t} = \mathbf{z} \left( \mathbf{u}^{(n)}, \mathbb{K}(\boldsymbol{\Lambda})^{(n)}, \boldsymbol{\Lambda}^{(n)}, \tilde{\mathbf{D}}^{(n)}, \mathbf{J}^{(n)} \right), \quad (4.51)$$

$$\frac{\bar{\mathbb{K}}(\boldsymbol{\Lambda})^{(n+1)} - \mathbb{K}(\boldsymbol{\Lambda})^{(n)}}{\delta t} = \mathbf{z} \left( \mathbf{u}^{(n)}, \mathbb{K}(\boldsymbol{\Lambda})^{(n)}, \boldsymbol{\Lambda}^{(n)}, \tilde{\mathbf{B}}^{(n)}, \mathbf{J}^{(n)} \right) \quad (4.52)$$

e

$$\frac{\bar{\mathbf{S}}^{(n+1)} - \mathbf{S}^{(n)}}{\delta t} = \mathbf{z} \left( \mathbf{u}^{(n)}, \mathbf{S}^{(n)}, \mathbf{H}^{(n)} \right). \quad (4.53)$$

As demais matrizes  $\bar{\boldsymbol{\Lambda}}^{(n+1)}$ ,  $\bar{\tilde{\mathbf{D}}}^{(n+1)}$ ,  $\bar{\tilde{\mathbf{B}}}^{(n+1)}$ ,  $\bar{\mathbf{J}}^{(n+1)}$  e  $\bar{\mathbf{H}}^{(n+1)}$ , no estágio intermediário, são calculadas, na ordem em que aparecem, por

$$\bar{\boldsymbol{\Lambda}}^{(n+1)} = \mathbb{K}^{-1} \left( \bar{\mathbb{K}}(\boldsymbol{\Lambda})^{(n+1)} \right), \quad (4.54)$$

$$\bar{\tilde{\mathbf{D}}}^{(n+1)} = \bar{\mathbf{O}}^{T(n+1)} \frac{1}{2} \left( \nabla \mathbf{u}^{(n+1)} + \nabla \mathbf{u}^{T(n+1)} \right) \bar{\mathbf{O}}^{(n+1)}, \quad (4.55)$$

$$\bar{\tilde{\mathbf{B}}}^{(n+1)} = \text{diag} \left( \bar{\mathbf{O}}^{T(n+1)} \nabla \mathbf{u}^{(n+1)} \bar{\mathbf{O}}^{(n+1)} \right), \quad (4.56)$$

$$\mathbf{J}^{(n+1)} = \frac{\partial \overline{\mathbb{K}}(\mathbf{\Lambda})^{(n+1)}}{\partial \mathbf{\Lambda}}, \quad (4.57)$$

e por fim,

$$\overline{H}_{12}^{(n+1)} = \frac{\overline{D}_{12}^{(n+1)} \left( \overline{\Lambda}_{11}^{(n+1)} + \overline{\Lambda}_{22}^{(n+1)} \right) + \overline{R}_{12}^{(n+1)} \left( \overline{\Lambda}_{22}^{(n+1)} - \overline{\Lambda}_{11}^{(n+1)} \right)}{\overline{\Lambda}_{22}^{(n+1)} - \overline{\Lambda}_{11}^{(n+1)}}, \quad (4.58)$$

caso escolhamos usar a equação (4.42) para evoluirmos  $\mathbb{K}(\mathbf{\Lambda})$ , do contrário

$$\overline{H}_{12}^{(n+1)} = \overline{\Omega}_{12}^{(n+1)} = \frac{\overline{\Lambda}_{22}^{(n+1)} \overline{M}_{12}^{(n+1)} + \overline{\Lambda}_{11}^{(n+1)} \overline{M}_{21}^{(n+1)}}{\overline{\Lambda}_{22}^{(n+1)} - \overline{\Lambda}_{11}^{(n+1)}}, \quad (4.59)$$

para a versão (4.43) da equação evolutiva de  $\mathbb{K}(\mathbf{\Lambda})$ . As matrizes  $\overline{\mathbf{O}}^{(n+1)}$  e  $\overline{\mathbf{M}}^{(n+1)}$  são definidas, respectivamente, como

$$\overline{\mathbf{O}}^{(n+1)} = \left( \mathbf{I} - \overline{\mathbf{S}}^{(n+1)} \right)^{-1} \left( \mathbf{I} + \overline{\mathbf{S}}^{(n+1)} \right) \quad (4.60)$$

e

$$\overline{\mathbf{M}}^{(n+1)} = \overline{\mathbf{O}}^{T(n+1)} \nabla_{\mathbf{u}^{(n+1)}} \overline{\mathbf{O}}^{(n+1)}. \quad (4.61)$$

De modo similar, as matrizes  $\mathbf{\Lambda}^{(n)}$ ,  $\tilde{\mathbf{D}}^{(n)}$ ,  $\tilde{\mathbf{B}}^{(n)}$ ,  $\mathbf{J}^{(n)}$ ,  $\mathbf{H}^{(n)}$ ,  $\mathbf{O}^{(n)}$  e  $\tilde{\mathbf{M}}^{(n)}$  são construídas com todos os termos discretizados no nível de tempo  $t = t_n$ .

Portanto, o novo algoritmo, incorporando o método *eigendecomposed kernel-conformation* para o cálculo do tensor conformação, contém os seguintes passos:

1. Dados  $\mathbf{\Lambda}^{(n)}$  e  $\mathbf{O}^{(n)}$ , construímos os tensores  $\mathbb{K}(\mathbf{\Lambda})^{(n)}$ , pela aplicação da transformação kernel  $\mathbb{K}$ , e  $\mathbf{S}^{(n)}$  pela relação (4.18).
2. Com  $\mathbb{K}(\mathbf{\Lambda})^{(n)}$  e  $\mathbf{u}^{(n)}$ , resolvemos a equação evolutiva (4.51) ou (4.52), para obtermos o valor intermediário  $\overline{\mathbb{K}}(\mathbf{\Lambda})^{(n+1)}$ ;
3. A partir de  $\overline{\mathbb{K}}(\mathbf{\Lambda})^{(n+1)}$ , construímos a matriz  $\overline{\mathbf{\Lambda}}^{(n+1)}$  pela equação (4.54);
4. Com os valores de  $\mathbf{u}^{(n)}$  e  $\mathbf{O}^{(n)}$ , construímos a matriz  $\tilde{\mathbf{M}}^{(n)}$  pela equação (4.61), que, juntamente com o valor de  $\mathbf{\Lambda}^{(n)}$ , obtemos a matriz  $\mathbf{H}^{(n)}$  pela equação (4.58) ou (4.59);
5. A partir dos valores de  $\mathbf{u}^{(n)}$ ,  $\mathbf{S}^{(n)}$  e  $\mathbf{H}^{(n)}$ , calculamos  $\overline{\mathbf{S}}^{(n+1)}$  pela equação evolutiva (4.53). Então, construímos o valor de  $\overline{\mathbf{O}}^{(n+1)}$  pela expressão (4.60);
6. A partir dos valores  $\overline{\mathbf{\Lambda}}^{(n+1)}$  e  $\overline{\mathbf{O}}^{(n+1)}$ , calculamos o valor intermediário do tensor conformação usando a equação (4.1). Então, a equação (2.4) é aplicada para deter-

minarmos o tensor polimérico  $\overline{\boldsymbol{\tau}}^{(n+1)}$ , que é usado para obtermos uma velocidade intermediária pela equação do momento (sub-passo do método de projeção);

7. Aplica-se o método de projeção para obter a velocidade final  $\mathbf{u}^{(n+1)}$  e o campo de pressão  $p^{(n+1)}$  (para mais detalhes ver [38]);
8. A partir dos valores de  $\mathbf{u}^{(n+1)}$ ,  $\overline{\boldsymbol{\Lambda}}^{(n+1)}$ ,  $\overline{\mathbf{O}}^{(n+1)}$ ,  $\mathbb{K}(\boldsymbol{\Lambda})^{(n+1)}$  e  $\overline{\mathbf{S}}^{(n+1)}$ , construímos todas as matrizes necessárias,  $\overline{\mathbf{D}}^{(n+1)}$  Eq. (4.55),  $\overline{\mathbf{B}}^{(n+1)}$  Eq. (4.56),  $\overline{\mathbf{J}}^{(n+1)}$  Eq. (4.57),  $\overline{\mathbf{H}}^{(n+1)}$  Eq. (4.58) ou Eq. (4.59), e  $\overline{\mathbf{M}}^{(M)}$  Eq. (4.61), no estágio intermediário;
9. Por fim, calculamos  $\mathbb{K}(\boldsymbol{\Lambda})^{(n+1)}$  e  $\mathbf{S}^{(n+1)}$  pelas equações (4.48) ou (4.49) e (4.50), respectivamente, e na sequência, transferimos os valores para  $\boldsymbol{\Lambda}^{(n+1)}$  Eq. (4.18) e  $\mathbf{O}^{(n+1)}$  Eq. (4.17). Neste estágio, o valor final do tensor conformação  $\mathbf{A}^{(n+1)}$  e do tensor polimérico  $\boldsymbol{\tau}^{(n+1)}$  são dados pelas equações (4.1) e (2.4), respectivamente.

**Observação 4.2.** Para iniciar o algoritmo é imposta uma condição inicial homogênea e isotrópica ao tensor conformação, isto é,  $\mathbf{A}_{t=t_0} = \mathbf{I}$ , que resulta em  $\mathbf{O}_{t=t_0} = \mathbf{I}$  e  $\boldsymbol{\Lambda}_{t=t_0} = \mathbf{I}$ . Entretanto, devido ao caculo de  $\mathbf{H}_{t=t_0}$ , no qual, envolve a expressão  $(\Lambda_{22} - \Lambda_{11})$  no denominador, equação (4.58) ou (4.59), é necessário a aplicação de uma pequena perturbação  $\delta\lambda$  a um dos valores de  $\boldsymbol{\Lambda}_{t=t_0}$ .

### 4.3 Validação: escoamento entre placas paralelas

A fim de validar a implementação da formulação *eigendecomposed kernel-conformation* apresentada neste capítulo, simulamos um escoamento entre duas placas paralelas. Na região de entrada do fluido, o perfil analítico da velocidade  $u(y)$  e das componentes do tensor polimérico  $\boldsymbol{\tau}$  são empregues de acordo com as equações abaixo:

$$u(y) = 1 - 4y^2 \text{ e } v = 0 \quad (4.62)$$

$$\tau_{11}(y) = \frac{2Wi}{Re} (1 - \beta) \left( \frac{\partial u}{\partial y} \right)^2, \quad \tau_{12}(y) = \frac{1}{Re} (1 - \beta) \left( \frac{\partial u}{\partial y} \right) \text{ e } \tau_{22} = 0, \quad (4.63)$$

As dimensões da geometria deste escoamento é  $H \times 10H$ , onde  $H$  representa a largura do canal. Em todas as simulações tomamos  $H = 1$ .

Para as simulações os seguintes dados foram usados:  $Re = 0.1$ ,  $Wi = 1$  e  $2$ ,  $\beta = 0.5$  e  $\delta t = 10^{-4}$ . Simulamos este problema usando duas malhas uniformes: M1 ( $\delta x = \delta y = 1/10$ ,  $10 \times 100$ ) e M2 ( $\delta x = \delta y = 1/20$ ,  $20 \times 200$ ). Para as simulações, usamos as duas versões da formulação *eigendecomposed kernel-conformation* para a evolução do tensor  $\mathbb{K}(\boldsymbol{\Lambda})$ . Na primeira versão (Eigen-v1) usamos a equação evolutiva (4.42), enquanto que

na segunda (Eigen-v2), a equação (4.43). Assim, buscando analisar a precisão do nosso código, bem como a convergência, apresentamos os erros relativos  $E(\cdot)$  para as duas malhas usadas, com as componentes  $u$  da velocidade e  $\tau_{11}$  e  $\tau_{12}$  do tensor polimérico, Tabelas 4.1 e 4.2, para  $Wi = 1$  e 2, respectivamente. O cálculo dos erros foi feito usando a norma Euclidiana (norma  $l_2$ ), comparando nossos resultados com a solução analítica, equações (4.62) e (4.63). Note que, apresentamos os valores dos erros relativos apenas para a versão Eigen-v1 da formulação *eigendecomposed kernel-conformation*, Tabelas 4.1 e 4.2, pois, devido à similaridade entre as duas versões do método, os erros relativos ficaram muito próximos de uma versão para outra. Além disso, o número de casas decimais usadas para representar os erros é insuficiente para identificar alguma diferença.

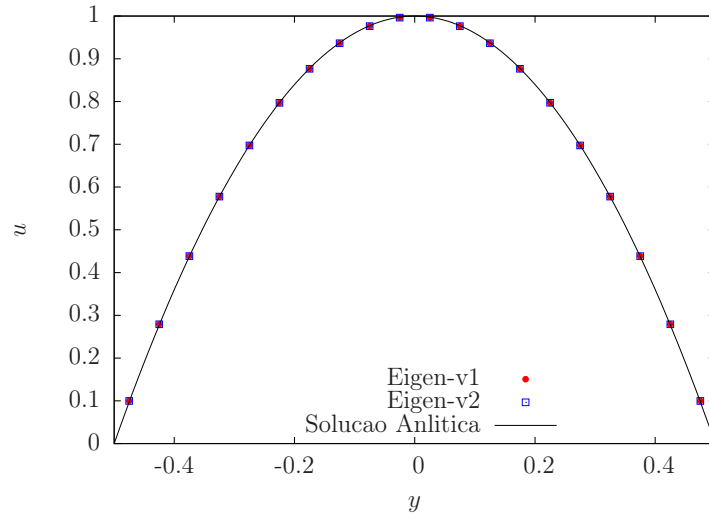
Podemos observar, mediante as Tabelas 4.1 e 4.2, que para os dois valores de  $Wi$ , 1 e 2, o método *eigendecomposed kernel-conformation* apresentou convergência, isto é, da malha M1 para a malha M2 houve um redução no erro relativo. Lembrando que esta observação é válida para as duas versões da formulação, Eigen-v1 e Eigen-v2. De maneira semelhante, as Figuras 9 e 10, confirmam os bons resultados deste método. Nas Figuras 9 e 10 apresentamos os resultados para ambas as versões da formulação *eigendecomposed kernel-conformation* e com os dois valores de Weissenberg  $Wi = 1$  e 2, respectivamente, para as componentes  $u$  da velocidade e  $\tau_{11}$  e  $\tau_{12}$  do tensor polimérico, foram tomados ao longo do eixo  $y$  com  $x = 5$  e no tempo  $t = 50$ .

Malha	$E(u^{M1})$	$E(\tau_{11}^{M1})$	$E(\tau_{12}^{M1})$
M1	$5.8570 \times 10^{-3}$	$2.7834 \times 10^{-2}$	$1.4108 \times 10^{-2}$
M2	$1.5145 \times 10^{-3}$	$1.3416 \times 10^{-2}$	$5.8722 \times 10^{-3}$

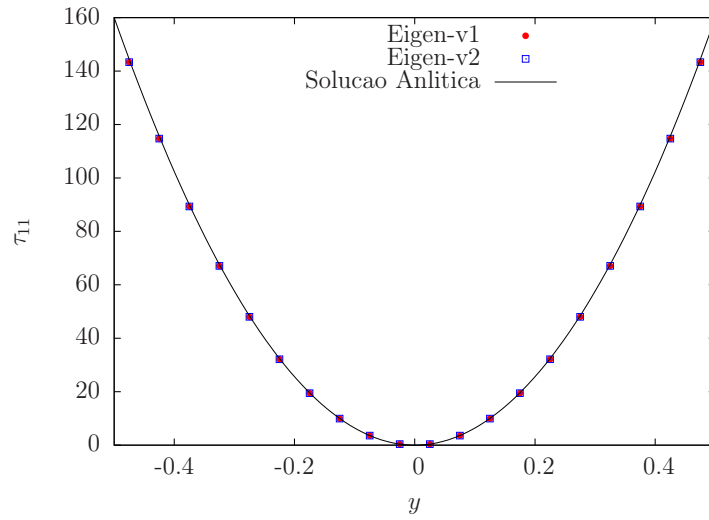
Tabela 4.1: Erros relativo do método Eigen-v1 calculados no meio do canal ( $x = 5$ ) com  $Wi = 1$ .

Malha	$E(u^{M1})$	$E(\tau_{11}^{M1})$	$E(\tau_{12}^{M1})$
M1	$5.8899 \times 10^{-3}$	$2.7807 \times 10^{-2}$	$1.4024 \times 10^{-2}$
M2	$1.6530 \times 10^{-3}$	$1.3870 \times 10^{-2}$	$6.5944 \times 10^{-3}$

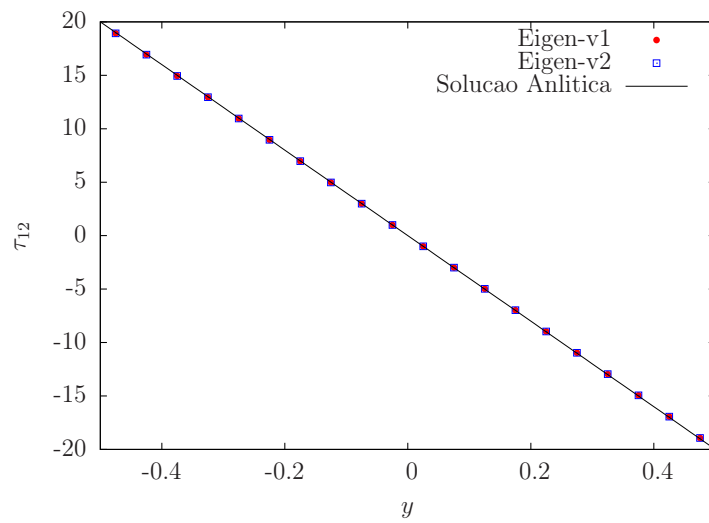
Tabela 4.2: Erros relativo do método Eigen-v1 calculados no meio do canal ( $x = 5$ ) com  $Wi = 2$ .



(a)



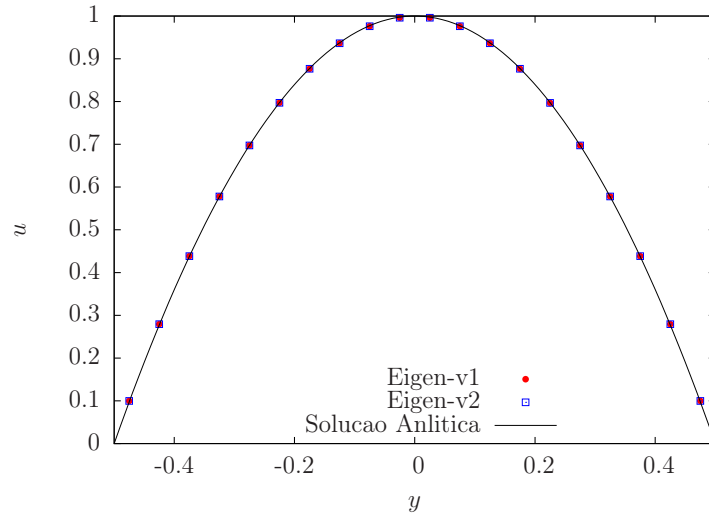
(b)



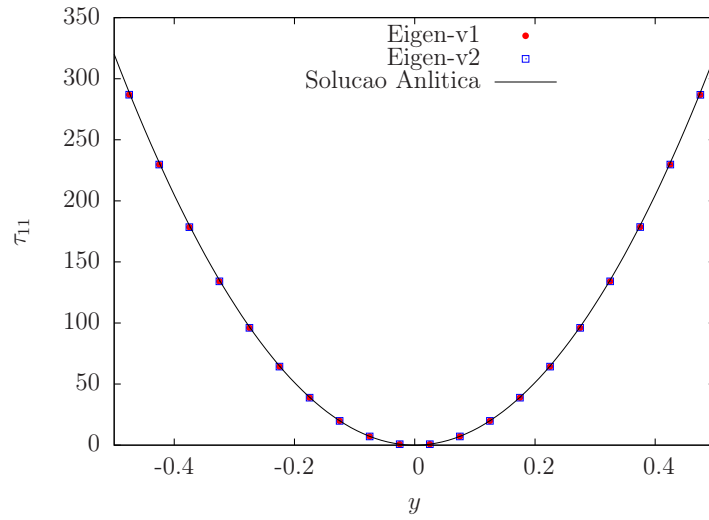
(c)

Figura 9: Solução numérica e analítica obtida ao longo do eixo  $y$  com  $x = 5H$ , usando  $Re = 0.1$ ,  $\beta = 0.5$  e  $Wi = 1$ : 9(a)  $u$ , 9(b)  $\tau_{11}$  e 9(c)  $\tau_{12}$  no tempo  $t = 50$ .

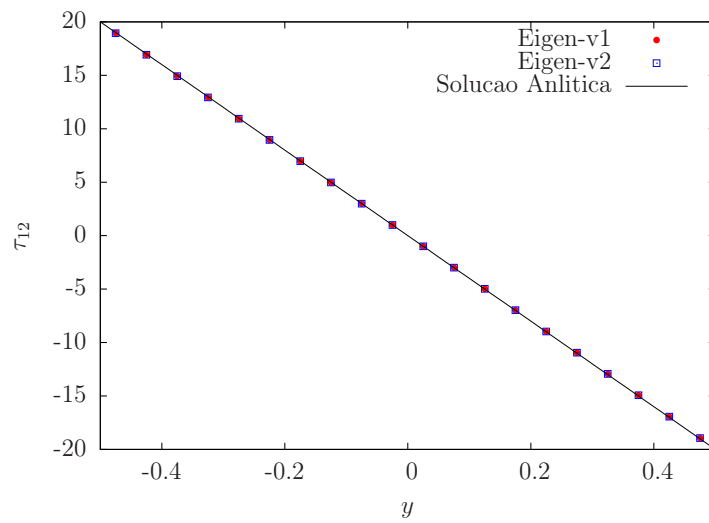




(a)



(b)



(c)

Figura 10: Solução numérica e analítica obtida ao longo do eixo  $y$  com  $x = 5H$ , usando  $Re = 0.1$ ,  $\beta = 0.5$  e  $Wi = 2$ : 10(a)  $u$ , 10(b)  $\tau_{11}$  e 10(c)  $\tau_{12}$  no tempo  $t = 50$ .

## Emprego da tensão superficial ao método MAC

Um fluido pode ser visto como um aglomerado de moléculas em movimento, empregando uma força de atração umas as outras (força de coesão). Desta forma, podemos dizer que as moléculas no interior do fluido sofrem atrações de todas as direções, pelas moléculas vizinhas, enquanto que as moléculas situadas na interface, na superfície do fluido, sofrem atrações laterais e internas apenas, ver Figura 11. Esta diferença de atração, ou desbalanço de forças, faz com que o fluido ajuste seu formato (*shape*) de modo a minimizar a área da superfície, fazendo com que a superfície se comporte como uma membrana elástica [41]. Esta situação explica o fato de a gota assumir um formato esférico, de um inseto conseguir flutuar sobre a água (Figura 12), embora o inseto seja mais denso que a água, e a imiscibilidade entre alguns fluidos.

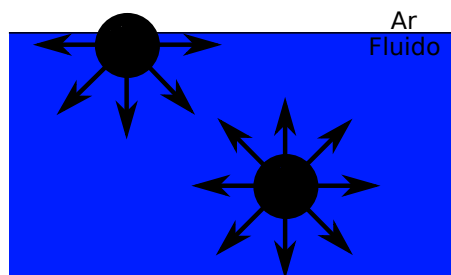


Figura 11: Moléculas de um fluido em contato com o ar (gás).

De modo grosseiro, podemos dizer que a energia de coesão de uma molécula na superfície é reduzida pela metade, quando comparada com as moléculas no interior do fluido. O parâmetro responsável por medir essa energia por unidade de área, na superfície do fluido, é a tensão superficial  $\sigma$ . Por exemplo, se  $E$  é a energia de coesão das moléculas de um fluido em seu interior, essa energia nas moléculas situadas na superfície do fluido

será, grosseiramente,  $E/2$ , resultando em  $\sigma \approx E/(2a^2)$ , onde  $a^2$  representa a “área” de uma molécula na superfície.

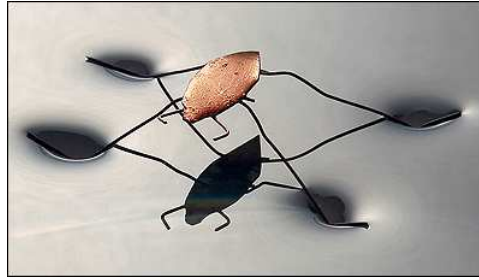


Figura 12: Inseto flutuando sobre a água devido à tensão superficial.

Vale destacar que a tensão superficial pode variar bastante de um fluido para outro, por exemplo, enquanto a tensão superficial da água é  $\sigma \approx 73(mN/m)$ , a tensão superficial do mercúrio é  $\sigma \approx 485(mN/m)$ , considerando a temperatura próxima a  $20^\circ C$ . Isto explica o fato de o mercúrio não se misturar com a água ou de um inseto caminhar com maior facilidade sobre o mercúrio. Muitas aplicações industriais envolvem tensão superficial, por exemplo, em processos de revestimento, secagem de pintura e o movimento das gotas em uma impressora de tinta a jato.

Além dos efeitos de tensão superficial, outra característica importante no estudo do problema do impacto da gota são os efeitos de molhamento, ou molhabilidade (*wettability*), que descrevem como um fluido se deposita sobre uma superfície sólida. Este fenômeno é importante em diversas áreas, como: indústria química (tintas, inseticidas), automobilística (preparação da superfície antes da pintura), entre outras [41]. Este efeito é quantificado pelo ângulo de contato  $\phi$ , formado pela superfície sólida e o líquido, ver Figura 13, indicando o grau de molhamento de um fluido em contato com uma superfície sólida. Se  $\phi \leq 90^\circ$  dizemos que o fluido tem alto molhamento, mas, se  $\phi > 90^\circ$  dizemos que o fluido possui baixo molhamento.

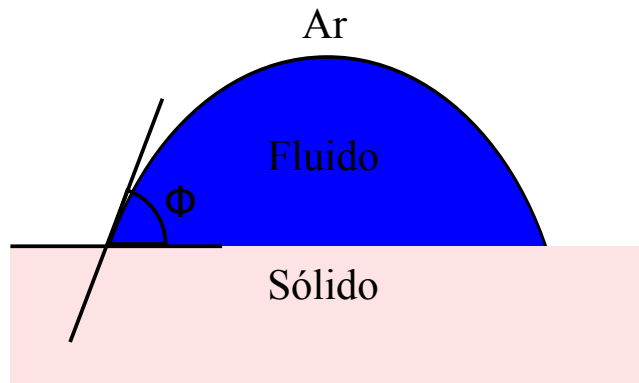


Figura 13: Ângulo de contato.

Os efeitos da tensão superficial são adicionados ao método MAC pelo cálculo da pressão capilar <sup>1</sup>, mediante o cálculo da curvatura na superfície livre, que é incluída nas condições de superfície livre para o campo de velocidade e de pressão [84, 60]. Por outro lado, os efeitos de molhamento são inseridos por meio de modificações no cálculo da curvatura nas células de superfície próximas ao contorno rígido. Desta forma, neste Capítulo, apresentamos os detalhes envolvidos na inclusão destas propriedades ao método MAC na sua versão implícita para fluidos viscoelásticos, expandindo sua aplicabilidade.

## 5.1 Metodologia

### 5.1.1 Método de projeção

Antes de iniciarmos a metodologia referente ao emprego da tensão superficial, apresentamos o método de projeção usado na solução das equações de Navier-Stokes (2.1) e (2.2), que servirá de base para as modificações resultantes da inclusão da tensão superficial. Para isto, apresentamos o Teorema da Decomposição de Helmholtz-Hodge (TDHH) [65]:

**Teorema 5.1.** *Seja  $\Gamma$  uma região com fronteira suave  $\partial\Gamma$  e  $\tilde{\mathbf{u}}$  um campo vetorial definido em  $\Gamma$ . Neste termos, a decomposição de  $\tilde{\mathbf{u}}$  na forma*

$$\tilde{\mathbf{u}} = \mathbf{u} + \nabla\psi \quad (5.1)$$

*existe e é única, sendo  $\psi$  um campo escalar também definido em  $\Gamma$ . Já o campo vetorial  $\mathbf{u}$  é solenoidal e paralelo a  $\partial\Gamma$ , isto é,*

$$\nabla \cdot \mathbf{u} = 0 \quad (5.2)$$

*e, ao longo de  $\partial\Gamma$ ,*

$$\mathbf{u} \cdot \mathbf{n} = 0, \quad (5.3)$$

*onde  $\mathbf{n}$  é o vetor normal a  $\partial\Gamma$  e orientado na direção externa de  $\Gamma$ .*

Com base no TDHH, o método de projeção consiste em definir um campo vetorial  $\tilde{\mathbf{u}}$ , velocidade intermediária, como em (5.1), e aplicar a equação do movimento (2.1) a este campo vetorial, ou seja,

$$\frac{\partial \tilde{\mathbf{u}}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) = -\nabla \tilde{p} + \frac{\beta}{Re} \nabla^2 \tilde{\mathbf{u}} + \nabla \cdot \boldsymbol{\tau} + \frac{1}{Fr^2} \mathbf{g} \quad (5.4)$$

---

<sup>1</sup>A pressão capilar é definida como a diferença na pressão, através da interface, entre dois fluidos imiscíveis

onde  $\tilde{p}$  é um campo de pressão intermediário, em geral, tal que

$$p \neq \tilde{p}. \quad (5.5)$$

Note que o campo vetorial  $\tilde{\mathbf{u}}$  não é necessariamente solenoidal, de modo que este precisar ser corrigido. Desta forma, uma vez que  $\tilde{\mathbf{u}}$  tenha sido modificado, ou corrigido pela relação com  $\nabla\psi$  Eq. (5.1), agora satisfazendo a equação da continuidade (2.2), é necessário corrigir o campo de pressão intermediária  $\tilde{p}$ , para que a equação do movimento também seja satisfeita. Assim, após as devidas correções, o campo de velocidade  $\mathbf{u}$  e o campo de pressão  $p$  devem satisfazer, simultaneamente, as equações (2.1) e (2.2).

Determinamos a correção  $\nabla\psi$ , para  $\tilde{\mathbf{u}}$ , aplicando o operador divergente  $\nabla \cdot$  em ambos os membros de (5.1), isto é,

$$\nabla \cdot \tilde{\mathbf{u}} = \nabla \cdot \mathbf{u} + \nabla \cdot (\nabla\psi), \quad (5.6)$$

pela hipótese do TDHH,  $\mathbf{u}$  é solenoidal, resultando na seguinte equação de poisson para  $\psi$

$$\nabla^2\psi = \nabla \cdot \tilde{\mathbf{u}}. \quad (5.7)$$

Segundo [85], as condições de contorno para  $\psi$  em (5.7) são, em fronteiras rígidas e na região de entrada de fluido,

$$\frac{\partial\psi}{\partial\mathbf{n}} = 0, \quad (5.8)$$

enquanto que, na região de saída de fluido,

$$\psi = 0. \quad (5.9)$$

**Observação 5.1.** *Os dois sistemas linear gerados pelas discretizações das equações (5.4) e (5.7) são resolvidos pelo método dos gradientes bi-conjugados estabilizado (biconjugate gradient stabilized method).*

O método aqui apresentado não é totalmente implícito, pois os termos não-lineares em Navier-Stokes são aproximados explicitamente, isto explica o motivo de não termos usado  $\tilde{\mathbf{u}}$  nos termos convectivos em (5.4). O próximo passo, para a construção do método de projeção para resolver as equações de Navier-Stokes, é a discretização temporal das equações (2.1) e (5.4), para  $\mathbf{u}$  e  $\tilde{\mathbf{u}}$ , respectivamente, permitindo a determinação de uma equação para a correção da pressão intermediária  $\tilde{p}$ . Assim, discretizando no tempo a equação (2.1), tem-se

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\delta t} + \text{conv}(\mathbf{u}\mathbf{u})^n = -\nabla p^{n+1} + \frac{1}{Re}\nabla^2\mathbf{u}^{n+1} + \nabla \cdot \boldsymbol{\tau}^{n+1} + \frac{1}{Fr^2}\mathbf{g}^{n+1}, \quad (5.10)$$

onde  $conv(\mathbf{uu})$  representa o termo não linear. Por outro lado, a discretização temporal de (5.4) é dada por

$$\frac{\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}^n}{\delta t} + conv(\mathbf{uu})^n = -\nabla \tilde{p}^{n+1} + \frac{\beta}{Re} \nabla^2 \tilde{\mathbf{u}}^{n+1} + \nabla \cdot \boldsymbol{\tau}^{n+1} + \frac{1}{Fr^2} \mathbf{g}^{n+1}, \quad (5.11)$$

substituindo  $\tilde{\mathbf{u}}^n = \mathbf{u}^n$  e (5.1) em (5.11), temos

$$\frac{(\mathbf{u}^{n+1} + \nabla \psi^{n+1}) - \mathbf{u}^n}{\delta t} + conv(\mathbf{uu})^n = -\nabla \tilde{p}^{n+1} + \frac{\beta}{Re} \nabla^2 (\mathbf{u}^{n+1} + \nabla \psi^{n+1}) + \nabla \cdot \boldsymbol{\tau}^{n+1} + \frac{1}{Fr^2} \mathbf{g}^{n+1}. \quad (5.12)$$

Finalmente, subtraindo as equações (5.10) e (5.12), resulta em

$$\frac{\nabla \psi^{n+1}}{\delta t} = -\nabla \tilde{p}^{n+1} + \nabla p^{n+1} + \frac{\beta}{Re} \nabla (\nabla^2 \psi), \quad (5.13)$$

ou ainda,

$$\nabla p^{n+1} = \nabla \tilde{p}^{n+1} + \frac{\nabla \psi^{n+1}}{\delta t} - \frac{\beta}{Re} \nabla (\nabla^2 \psi^{n+1}). \quad (5.14)$$

Pelo teorema da divergência (ou de Gauss), a equação (5.14) se reduz a

$$p^{n+1} = \tilde{p}^{n+1} + \frac{\psi^{n+1}}{\delta t} - \frac{\beta}{Re} \nabla^2 \psi^{n+1}, \quad (5.15)$$

que é a equação usada para atualizar a pressão [86], corrigindo a pressão intermediária  $\tilde{p}^{n+1}$ , para que  $\mathbf{u}$  satisfaça (2.1).

Observe que não há necessidade de impormos condições de contorno para a pressão, uma vez que, para determinar a pressão em um ponto, só precisamos de informações da pressão no ponto em questão. Apesar disso, em sua formulação explícita, o método MAC em conjunto com o método de projeção apresentado acima, faz uso da equação (2.5), usada como condição de contorno na superfície livre, para o cálculo da pressão nesta região, ou seja,

$$p = \frac{2\beta}{Re} \left[ \frac{\partial u}{\partial x} n_x^2 + \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + \frac{\partial v}{\partial y} n_y^2 \right] + \tau_{11} n_x^2 + 2\tau_{12} n_x n_y + \tau_{22} n_y^2. \quad (5.16)$$

**Observação 5.2.** *O cálculo da pressão na superfície livre é necessário, uma vez que ao mover a superfície livre as células são reclassificadas, podendo ocasionar de uma célula do tipo superfície livre [S] tornar-se uma célula cheia [F], necessitando de um valor nesta nova célula para atualizar a pressão auxiliar que será usada no passo seguinte, isto é,  $\tilde{p}^{(n+1)} = p^n$ . Para mais detalhes sobre a classificação das células no método MAC ver [85].*

Além disso, para a solução da equação de Poisson (5.7), a formulação explícita usa uma condição de contorno do tipo Dirichle para  $\psi$  ao longo da superfície livre

$$\psi = 0. \quad (5.17)$$

Entretanto, o método implícito, juntamente com a simulação em problemas envolvendo superfície livre, teve seu desempenho reduzido em termos de estabilidade, devido ao uso destas condições de contorno na superfície livre, equações (5.16) e (5.17), no qual acoplava velocidade e pressão [87, 63]. Assim, uma nova modificação introduzida pelo método implícito foi o cálculo da pressão na superfície livre de maneira implícita, ou seja, resolvemos a pressão na superfície livre da mesma maneira que é feita no interior do domínio, pela equação (5.15). Dai, a equação (5.16) juntamente com as equações (5.1) e (5.15) passam a formar a nova condição de superfície livre para  $\psi$ , dada por

$$\begin{aligned} \frac{\psi^{n+1}}{\delta t} - \frac{1}{Re} \left( \frac{\partial^2 \psi^{n+1}}{\partial x^2} + \frac{\partial^2 \psi^{n+1}}{\partial y^2} \right) + \frac{2\beta}{Re} \left[ 2 \frac{\partial^2 \psi^{n+1}}{\partial x \partial y} n_x n_y + \frac{\partial^2 \psi^{n+1}}{\partial x^2} n_x^2 + \frac{\partial^2 \psi^{n+1}}{\partial y^2} n_y^2 \right] = \\ = -\tilde{p}^{n+1} + \frac{2\beta}{Re} \left[ \left( \frac{\partial \tilde{v}^{n+1}}{\partial x} + \frac{\partial \tilde{u}^{n+1}}{\partial y} \right) n_x n_y + \frac{\partial \tilde{u}^{n+1}}{\partial x} n_x^2 + \frac{\partial \tilde{v}^{n+1}}{\partial y} n_y^2 \right] + \tau_{11}^{n+1} n_x^2 + 2\tau_{12}^{n+1} n_x n_y + \tau_{22}^{n+1} n_y^2. \end{aligned} \quad (5.18)$$

Isto posto, estamos em condições de tratar da inclusão da tensão superficial ao método MAC, em sua versão implícita. Antes, porém, apresentamos brevemente alguns conceitos básicos necessários para o correto entendimento da formulação: projeção ortogonal e o método dos mínimos quadrados.

### 5.1.2 Conceitos básicos sobre projeção ortogonal

Sabe-se que a menor distância entre um ponto  $P$  e uma reta  $r$  é o segmento de reta  $\overline{PQ}$ , onde  $Q$  é o ponto de  $r$  tal que  $\overline{PQ}$  é perpendicular a  $r$ , ou seja, a menor distância é a projeção ortogonal de  $P$  sobre  $r$ . Esta mesma ideia se aplica ao conceito de distância entre um vetor  $\mathbf{u}$ , pertencente ao espaço vetorial  $\mathbf{E}$ , ao subespaço vetorial  $\mathbf{E}^* \subset \mathbf{E}$ , tal que  $\mathbf{u} \notin \mathbf{E}^*$ .

Desta forma, suponhamos que o vetor  $\mathbf{v} \in \mathbf{E}^*$  seja a projeção ortogonal de  $\mathbf{u} \in \mathbf{E}$  sobre  $\mathbf{E}^*$ . Além disso, sabemos que  $\mathbf{v} \perp (\mathbf{u} - \mathbf{v})$ , ou seja,

$$\mathbf{v} \cdot (\mathbf{u} - \mathbf{v}) = 0. \quad (5.19)$$

**Observação 5.3.** O operador "." representa o produto interno (ou produto escalar).

Consideremos que o espaço vetorial  $\mathbf{E}$  seja de dimensão finita  $m$ . Logo,  $\mathbf{E}^*$  também tem dimensão finita, e existe uma base  $\mathbf{B} = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{n-1}, \mathbf{e}_n\}$  para  $\mathbf{E}^*$ . Desta forma,

podemos escrever  $\mathbf{v}$  como uma combinação linear dos elementos da base

$$\mathbf{v} = \alpha_1 \mathbf{e}_1 + \alpha_2 \mathbf{e}_2 + \dots + \alpha_{n-1} \mathbf{e}_{n-1} + \alpha_n \mathbf{e}_n, \quad (5.20)$$

tal que os  $\alpha_i$ 's, com  $i = 1, \dots, n$ , são números reais.

Além de termos (5.19), o vetor  $(\mathbf{u} - \mathbf{v})$  é ortogonal a cada elemento da base  $\mathbf{B}$ . Assim, usando este fato, juntamente com a equação (5.20), temos que

$$\begin{cases} \mathbf{e}_1 \cdot (\mathbf{u} - (\alpha_1 \mathbf{e}_1 + \alpha_2 \mathbf{e}_2 + \dots + \alpha_n \mathbf{e}_n)) = 0 \\ \mathbf{e}_2 \cdot (\mathbf{u} - (\alpha_1 \mathbf{e}_1 + \alpha_2 \mathbf{e}_2 + \dots + \alpha_n \mathbf{e}_n)) = 0 \\ \mathbf{e}_3 \cdot (\mathbf{u} - (\alpha_1 \mathbf{e}_1 + \alpha_2 \mathbf{e}_2 + \dots + \alpha_n \mathbf{e}_n)) = 0 \\ \vdots \\ \mathbf{e}_n \cdot (\mathbf{u} - (\alpha_1 \mathbf{e}_1 + \alpha_2 \mathbf{e}_2 + \dots + \alpha_n \mathbf{e}_n)) = 0 \end{cases} \quad (5.21)$$

que resulta no seguinte sistema linear

$$\begin{pmatrix} \mathbf{e}_1 \cdot \mathbf{e}_1 & \mathbf{e}_1 \cdot \mathbf{e}_2 & \mathbf{e}_1 \cdot \mathbf{e}_3 & \dots & \mathbf{e}_1 \cdot \mathbf{e}_n \\ \mathbf{e}_2 \cdot \mathbf{e}_1 & \mathbf{e}_2 \cdot \mathbf{e}_2 & \mathbf{e}_2 \cdot \mathbf{e}_3 & \dots & \mathbf{e}_2 \cdot \mathbf{e}_n \\ \mathbf{e}_3 \cdot \mathbf{e}_1 & \mathbf{e}_3 \cdot \mathbf{e}_2 & \mathbf{e}_3 \cdot \mathbf{e}_3 & \dots & \mathbf{e}_3 \cdot \mathbf{e}_n \\ \vdots & & & & \\ \mathbf{e}_n \cdot \mathbf{e}_1 & \mathbf{e}_n \cdot \mathbf{e}_2 & \mathbf{e}_n \cdot \mathbf{e}_3 & \dots & \mathbf{e}_n \cdot \mathbf{e}_n \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} \mathbf{e}_1 \cdot \mathbf{u} \\ \mathbf{e}_2 \cdot \mathbf{u} \\ \mathbf{e}_3 \cdot \mathbf{u} \\ \vdots \\ \mathbf{e}_n \cdot \mathbf{u} \end{pmatrix} \quad (5.22)$$

Uma vez determinados os coeficientes  $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n$ , podemos calcular  $\mathbf{v}$  por (5.20). Portanto, conhecido o subespaço  $\mathbf{E}^*$ , e uma base para  $\mathbf{E}^*$ , a única dificuldade em se determinar a projeção ortogonal do vetor  $\mathbf{u}$  sobre  $\mathbf{E}^*$  é resolver o sistema linear (5.22).

### 5.1.3 O método dos mínimos quadrados

Dado um conjunto de pares ordenados  $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_m, y_m)\}$  podemos construir uma função  $F$  que melhor aproxima estes pontos. Suponhamos que exista uma função  $f$  ideal, tal que

$$f(x_i) = y_i, \text{ com } i = 1, 2, 3, \dots, m. \quad (5.23)$$

Assim, consideremos uma aproximação  $P_r$  para  $f$

$$P_r(x) = \gamma_0 + \gamma_1 x + \gamma_2 x^2 + \dots + \gamma_r x^r, \quad (5.24)$$

tal que a distância ao quadrado entre  $f$  e  $P_r$

$$\text{dist}(f, P_r)^2 = \|f - P_r\|^2 \quad (5.25)$$



seja mínima.

Consideremos o seguinte produto interno:

$$f \cdot g = \sum_{k=1}^m f(x_k) g(x_k). \quad (5.26)$$

Desta forma, podemos escrever (5.25) como

$$\begin{aligned} \text{dist}(f, P_r)^2 &= (f - P_r) \cdot (f - P_r) = \\ &= \sum_{k=1}^m [f(x_k) - P_r(x_k)]^2 = \sum_{k=1}^m [y_k - (\gamma_0 + \gamma_1 x + \gamma_2 x^2 + \gamma_3 x^3 + \dots + \gamma_r x^r)]^2 \end{aligned} \quad (5.27)$$

Veja que ainda não determinamos como calcular os coeficientes de  $P_r$ . Entretanto, já vimos que a menor distância entre um vetor e um subespaço vetorial é a projeção ortogonal do vetor sobre o subespaço. Assim, o polinômio  $P_r$ , que minimiza  $\text{dist}(f, P_r)^2$ , é a projeção ortogonal de  $f$  sobre o subespaço  $K_r(x)$ , subespaço vetorial dos polinômios de grau menor ou igual a  $r$ . Desta forma, tomando a base canônica  $\mathbf{B} = \{1, x, x^2, x^3, \dots, x^r\}$  de  $K_r(x)$ , podemos calcular  $P_r$  pela determinação de seus coeficientes mediante o sistema (5.22), ou seja,

$$\begin{pmatrix} 1.1 & 1.x & 1.x^2 & \dots & 1.x^r \\ x.1 & x.x & x.x^2 & \dots & x.x^r \\ x^2.1 & x^2.x & x^2.x^2 & \dots & x^2.x^r \\ \vdots & & & & \\ x^r.1 & x^r.x & x^r.x^2 & \dots & x^r.x^r \end{pmatrix} \begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_r \end{pmatrix} = \begin{pmatrix} 1.y_0 \\ 1.y_1 \\ 1.y_2 \\ \vdots \\ 1.y_m \end{pmatrix} \quad (5.28)$$

#### 5.1.4 A inclusão da tensão superficial ao método MAC

Nesta subseção, apresentamos o processo de inclusão das forças superficiais ao escoamento. Para isto, vamos empregar a pressão capilar  $p_{cap}$  à condição de superfície livre (2.5) da seguinte forma (ver [88] pág. 150 e [89] pág. 382)

$$\mathbf{n} \cdot (\boldsymbol{\sigma} \cdot \mathbf{n}) = p_{cap}, \quad (5.29)$$

onde  $p_{cap} = -\frac{\kappa}{We}$ , tal que  $\kappa$  é a curvatura adimensional, e  $We = \frac{\rho LU^2}{\sigma}$  é o número adimensional de Weber ( $\sigma$  é o coeficiente de tensão superficial). Assim, a pressão capilar é acrescentada à equação (5.18) como um termo fonte, usado na condição de contorno para o potencial auxiliar  $\psi$  ao longo da superfície livre:

$$\begin{aligned}
& \frac{\psi^{n+1}}{\delta t} - \frac{1}{Re} \left( \frac{\partial^2 \psi^{n+1}}{\partial x^2} + \frac{\partial^2 \psi^{n+1}}{\partial y^2} \right) + \frac{2\beta}{Re} \left[ 2 \frac{\partial^2 \psi^{n+1}}{\partial x \partial y} n_x n_y + \frac{\partial^2 \psi^{n+1}}{\partial x^2} n_x^2 + \frac{\partial^2 \psi^{n+1}}{\partial y^2} n_y^2 \right] = \\
& = -\tilde{p}^{n+1} + \frac{2\beta}{Re} \left[ \left( \frac{\partial \tilde{v}^{n+1}}{\partial x} + \frac{\partial \tilde{u}^{n+1}}{\partial y} \right) n_x n_y + \frac{\partial \tilde{u}^{n+1}}{\partial x} n_x^2 + \frac{\partial \tilde{v}^{n+1}}{\partial y} n_y^2 \right] + \\
& + \tau_{11}^{n+1} n_x^2 + 2\tau_{12}^{n+1} n_x n_y + \tau_{22}^{n+1} n_y^2 - p_{cap}^n.
\end{aligned} \tag{5.30}$$

Note que a pressão capilar está no tempo  $t = t_n$ , uma vez que, no momento do cálculo de  $\psi$ , a superfície livre ainda não foi movimentada, resultando que as partículas marcadoras, usadas no cálculo da curvatura, ainda estarão no tempo  $t = t_n$ .

Vale ressaltar que na formulação explícita, apresentada por Mangiavacchi et al. [60], a condição (5.29) é usada para o cálculo da pressão na superfície livre, enquanto que na formulação implícita, será usada como condição de contorno na superfície livre para o potencial auxiliar  $\psi$ . Entretanto, a pressão na superfície livre ainda receberá os efeitos da pressão capilar, pois a atualização da pressão na superfície livre em  $t = t_{n+1}$  envolve o valor de  $\psi^{n+1}$ , equação (5.15).

O cálculo da curvatura em uma dada célula de superfície é feito mediante a aproximação de uma curva, um polinômio de segundo grau, à superfície livre usando o método dos mínimos quadrados. Antes, porém, é usado uma aproximação linear, ou seja, aproximamos a superfície livre por uma reta, também pelo método dos mínimos quadrados, para o cálculo do vetor normal à superfície livre  $\mathbf{n}_s$ .

**Observação 5.4.** *Para o cálculo das condições de contorno na superfície livre, é usado a antiga aproximação para a normal no centro da célula [90]:  $\mathbf{n}_c = (1, 0)$ , somente quando a face direita da célula surface ([S]) está em contato com uma célula vazia ([E]),  $\mathbf{n}_c = (0, 1)$  somente quando a face superior da célula [S] está em contato com uma célula [E],  $\mathbf{n}_c = (-1, 0)$  somente quando a face esquerda da célula [S] está em contato com uma célula [E],  $\mathbf{n}_c = (0, -1)$  somente quando a face inferior da célula [S] está em contato com uma célula vazia [E],  $\mathbf{n}_c = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)$  somente quando as faces direita e superior da célula [S] estão em contato com células [E], e assim por diante.*

#### 5.1.4.1 Cálculo da curvatura $\kappa$

Definimos uma distancia  $\omega$  a partir do ponto central  $\mathbf{x}_c$  da célula [S] em questão, e tomamos todas as partículas virtuais,  $(x_i, y_i)$  com  $i = 1, \dots, m$ , dentro deste raio. O tamanho de  $\omega$  é tomado de modo a termos partículas dentro da célula [S] considerada, bem como de células [S] vizinhas.

Como primeiro passo para determinarmos a curvatura  $\kappa$  na célula  $[S]$ , vamos aproximar a superfície livre por uma reta

$$y = P_1(x) = \alpha_0 + \alpha_1 x \quad (5.31)$$

pelo método dos mínimos quadrados. Recorrendo ao que foi apresentado nas subseções sobre projeção ortogonal e o método dos mínimos quadrados, consideremos a função  $f \in \mathbf{E} = C(\mathbb{R})$ , espaço vetorial das funções contínuas, tal que  $y_i = f(x_i)$  para  $i = 1, \dots, m$ . Queremos obter a projeção ortogonal de  $f$  sobre o subespaço  $\mathbf{K}_1(x)$ , espaço vetorial dos polinômios de grau menor ou igual a 1. Além disso, sabemos que a base canônica de  $\mathbf{K}_1(x)$  é  $\mathbf{B} = \{1, x\}$ . Portanto, podemos determinar os coeficientes do polinômio de primeiro grau (5.31) através do sistema linear

$$\begin{pmatrix} 1.1 & 1.x \\ x.1 & x.x \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = \begin{pmatrix} 1.y \\ x.y \end{pmatrix} \quad (5.32)$$

onde os valores da matriz dos coeficientes e do vetor independente são determinados pelo produto interno (5.26).

Note que os pontos  $(x_i, P_1(x_i))$ , para  $i = 1, \dots, m$ , da equação (5.31), reescrita, por simplicidade, como

$$\alpha_0 + \alpha_1 x - y = 0, \quad (5.33)$$

representam a curva de nível da função

$$f(x, y) = 0, \quad (5.34)$$

onde  $f(x, y) = \alpha_0 + \alpha_1 x - y$ .

Calculando o vetor gradiente de  $f(x, y)$ , temos

$$\nabla f(x, y) = (\alpha_1, -1). \quad (5.35)$$

Sabemos que o vetor gradiente é ortogonal à curva de nível formada pelos pontos  $(x_i, P_1(x_i))$ , para  $i = 1, \dots, m$ . Desta forma, o vetor normal à superfície livre é determinado por

$$\mathbf{n}_s = \left( \frac{\alpha_1}{\sqrt{\alpha_1^2 + 1}}, -\frac{1}{\sqrt{\alpha_1^2 + 1}} \right). \quad (5.36)$$

**Observação 5.5.** Se a matriz dos coeficientes do sistema linear (5.32) for singular (não inversível), a melhor aproximação para a reta será  $x = c$  (uma reta na direção vertical), cujo vetor normal é  $\mathbf{n}_s = (1, 0)$ . Entretanto, durante a simulação esse vetor é ajustado de modo a concordar com o vetor por célula  $\mathbf{n}_c$ .

O próximo passo, como uma forma de “facilita” os cálculos subsequentes do método dos mínimos quadrados, iremos aplicar uma translação e uma rotação aos eixos coordenados  $xOy$ . Primeiramente, aplicamos a translação, afim de termos a origem do sistema cartesiano no centro  $\mathbf{x}_c = (x_c, y_c)$  da célula de superfície considerada, ou seja,

$$\begin{cases} x^* = x - x_c \\ y^* = y - y_c \end{cases} \quad (5.37)$$

Para realizarmos a rotação, devemos lembrar-nos que o vetor normal pode ser escrito como

$$\mathbf{n}_s = (n_x, n_y) = (\cos(\alpha), \sin(\alpha)), \quad (5.38)$$

onde  $\alpha$  é o ângulo formado pelo vetor  $\mathbf{n}_s$  e a parte positiva do eixo  $Ox$ , considerando o ângulo no sentido anti-horário. A rotação que queremos aplicar ao sistema  $x^*Oy^*$ , para gerarmos o sistema  $\xi O\eta$ , resultará que  $\eta$  ficará alinhado ao vetor  $\mathbf{n}_s$ . Além disso, queremos destacar que a relação existente entre o ângulo  $\alpha$ , formado pelo vetor normal  $\mathbf{n}_s$ , e o ângulo de rotação  $\theta$ , do sistema cartesiano, é

$$\theta = \alpha - 90^\circ. \quad (5.39)$$

Assim, temos que

$$\begin{cases} \xi = x^* \cos(\theta) + y^* \sin(\theta) \\ \eta = y^* \cos(\theta) - x^* \sin(\theta) \end{cases} \quad (5.40)$$

Aplicando a relação (5.39), resulta em

$$\begin{cases} \xi = x^* \sin(\alpha) - y^* \cos(\alpha) \\ \eta = y^* \sin(\alpha) + x^* \cos(\alpha) \end{cases} \quad (5.41)$$

Finalmente, pelas equações (5.37) e (5.38), temos

$$\begin{cases} \xi = (x - x_c) n_y - (y - y_c) n_x \\ \eta = (x - x_c) n_x + (y - y_c) n_y \end{cases} \quad (5.42)$$

O passo seguinte para o cálculo da curvatura  $\kappa$  é a determinação de um polinômio de segundo grau,

$$\eta = P_2(\xi) = \gamma_0 + \gamma_1 \xi + \gamma_2 \xi^2, \quad (5.43)$$

para aproximar a superfície livre, semelhante ao que fizemos para o polinômio de grau um. Desta forma, determinamos os coeficientes de  $P_2(\xi)$  por resolver o sistema

$$\begin{pmatrix} 1.1 & 1.\xi & 1.\xi^2 \\ \xi.1 & \xi.\xi & \xi.\xi^2 \\ \xi^2.1 & \xi^2.\xi & \xi^2.\xi^2 \end{pmatrix} \begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \end{pmatrix} = \begin{pmatrix} 1.\eta \\ \xi.\eta \\ \xi^2.\eta \end{pmatrix} \quad (5.44)$$

O produto interno dos elementos da matriz dos coeficientes e do vetor independente são calculados por (5.26). Portanto, uma vez determinado o polinômio  $P_2(\xi)$ , a curvatura é calculada pela relação [91, 92]:

$$\kappa = -\frac{P_2''(\xi)^2}{\left(1 + P_2'(\xi)^2\right)^{3/2}}. \quad (5.45)$$

Note que  $P_2'(\xi) = \gamma_1 + 2\gamma_2\xi$  e  $P_2''(\xi) = 2\gamma_2$ . Entretanto, a curvatura é um conceito pontual, ou seja, precisamos definir um ponto, dentre a sequência de pontos  $(\xi_1, \xi_2, \dots, \xi_m)$ , para avaliar a curvatura. O ponto escolhido, será o ponto de máximo da função  $P_2(\xi)$ , ou seja, onde  $P_2'(\xi) = 0$ . Assim, segue que

$$\kappa = -2\gamma_2. \quad (5.46)$$

Se o sistema (5.44) for singular, então a curvatura é tomada como sendo zero. Além disso, o sinal da curvatura é determinado pelo sinal da expressão  $\mathbf{n}_c \cdot \mathbf{n}_s$ . Se  $\mathbf{n}_c \cdot \mathbf{n}_s < 0$ , então o sinal da curvatura será o inverso, isto é,  $\kappa = 2\gamma_2$ .

#### 5.1.4.2 O emprego do ângulo de contato estático $\phi$

Nas regiões em que a superfície livre estiver em contato com o contorno rígido, ou a parede, teremos a influência do ângulo de contato. Assim, iremos modificar o cálculo da curvatura nas células de superfície livre adjacentes às células do contorno rígido, de modo a inserir as informações do ângulo de contato. Consideremos os seguintes parâmetros usados na dedução das modificações no cálculo da curvatura, conforme ilustrado na Figura 14:

- $\mathbf{n}_1$  é o vetor normal à superfície livre no ponto  $\mathbf{x}_1$ , localizado na célula de superfície livre  $[S2]$  vizinha à célula considerada  $[S1]$
- $\mathbf{n}_2$  é o vetor normal à superfície livre no ponto de contato  $\mathbf{x}_2$ , localizado no encontro da superfície livre com o contorno rígido
- $\mathbf{n}_3$  é o vetor normal externo ao contorno rígido
- $\beta$  é a distância entre o contorno rígido e o ponto  $\mathbf{x}_1$
- $\mathbf{x}_0$  é o ponto de encontro (a interseção) entre as retas que passam por  $\mathbf{n}_1$  e  $\mathbf{n}_2$

- $\phi$  é o ângulo de contato

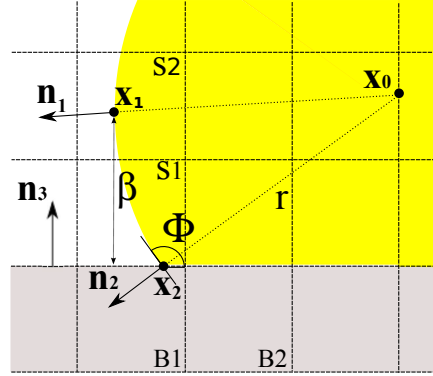


Figura 14: Esquema para o cálculo do ângulo de contato  $\phi$ .

Na representação da gota no tempo inicial  $t_0$ , e um pouco antes de atingir a superfície sólida, a gota possui exatamente um formato esférico (representação no computador). Com base nisto e supondo que ela mantenha este formato esférico após o impacto, ou algo próximo disso, construímos as seguintes relações

$$\begin{cases} \mathbf{x}_1 - \mathbf{x}_0 = r \mathbf{n}_1 \\ \mathbf{x}_2 - \mathbf{x}_0 = r \mathbf{n}_2 \end{cases} \quad (5.47)$$

onde  $r = \|\mathbf{x}_1 - \mathbf{x}_0\| = \|\mathbf{x}_2 - \mathbf{x}_0\|$  é o raio da circunferência  $R(\mathbf{x}_0, r)$  de centro  $\mathbf{x}_0$  e raio  $r$ , ver Figura 14. Ou seja, tomando as retas que passam por  $\mathbf{n}_1$  e  $\mathbf{n}_2$ , o ponto de encontro dessas duas retas, denotado por  $\mathbf{x}_0$ , será o centro da circunferência que intercepta os pontos  $\mathbf{x}_1$  e  $\mathbf{x}_2$ .

Subtraindo as duas expressões na equação (5.47), resulta em

$$\mathbf{x}_1 - \mathbf{x}_2 = r (\mathbf{n}_1 - \mathbf{n}_2). \quad (5.48)$$

Note que  $\beta$ , a distância entre o contorno rígido e o ponto  $\mathbf{x}_1$ , pode ser definida como:

$$\beta = (\mathbf{x}_1 - \mathbf{x}_2) \cdot \mathbf{n}_3, \quad (5.49)$$

que representa a projeção ortogonal do vetor  $(\mathbf{x}_1 - \mathbf{x}_2)$  na direção  $\mathbf{n}_3$ . Substituindo a equação (5.48) em (5.49), resulta em

$$\beta = r (\mathbf{n}_1 - \mathbf{n}_2) \cdot \mathbf{n}_3, \quad (5.50)$$

que implica em

$$r = \beta (\mathbf{n}_1 - \mathbf{n}_2) \cdot \mathbf{n}_3. \quad (5.51)$$

Desta forma, determinamos a curvatura  $\kappa$  que é definida como o inverso do raio da circunferência

$$\kappa = \frac{1}{r} = \frac{(\mathbf{n}_1 - \mathbf{n}_2) \cdot \mathbf{n}_3}{\beta}. \quad (5.52)$$

Observe que ainda não esclarecemos como determinar  $\mathbf{n}_2$ ,  $\mathbf{n}_3$  e  $\beta$ . Lembrando que o vetor  $\mathbf{n}_1$  é obtido no cálculo de  $\kappa$  na célula  $[S2]$ .

O vetor  $\mathbf{n}_3$  é calculado mediante a aproximação de uma reta  $y = \alpha_0 + \alpha_1 x$  ao contorno rígido pelo método dos mínimos quadrados. Primeiramente determinamos três pontos na face da célula de contorno  $[B1]$  em contato com  $[S1]$ . Estes pontos são obtidos pela interseção das seis retas

$$\begin{aligned} x &= r_{i+1/2}; \\ x &= r_i; \\ x &= r_{i-1/2}; \\ y &= r_{j+1/2}; \\ y &= r_j; \\ y &= r_{j-1/2}, \end{aligned} \quad (5.53)$$

com a face de  $[B1]$  em contato com  $[S1]$ , considerando  $[B1]$  com coordenadas  $(i, j)$ . Assim, uma vez determinados os pontos sobre  $[B1]$ , podemos usar o método dos mínimos quadrados para obter  $y = \alpha_0 + \alpha_1 x$ , semelhante ao que fizemos para determinar a reta usada no cálculo de  $\mathbf{n}_s$ . Desta forma, o vetor  $\mathbf{n}_3$  é obtido por

$$\mathbf{n}_3 = \left( \frac{\alpha_1}{\sqrt{\alpha_1^2 + 1}}, -\frac{1}{\sqrt{\alpha_1^2 + 1}} \right). \quad (5.54)$$

**Observação 5.6.** *Se o sistema linear usado no cálculo dos coeficientes de  $y = \alpha_0 + \alpha_1 x$  for singular, a melhor aproximação será a reta  $x = c$ , que resultará em  $\mathbf{n}_3 = (1, 0)$ . Entretanto, este valor será ajustado mediante comparação com o vetor normal por célula, definido no centro de  $[B1]$ .*

Com a equação  $y = \alpha_0 + \alpha_1 x$ , descrevendo a superfície de  $[B1]$  em contato com  $[S1]$ , podemos determinar o ângulo formado entre o contorno rígido e o eixo  $x$  positivo da seguinte forma:

$$\theta = \arctg(\alpha_1). \quad (5.55)$$

**Observação 5.7.** *O cálculo, ou mesmo o sinal, de  $\theta$  irá sofrer modificações dependendo da direção/sentido do vetor  $\mathbf{n}_3$ .*

Uma vez conhecidos  $\theta$  e  $\phi$  (parâmetro de entrada), podemos calcular o vetor  $\mathbf{n}_2$ . De fato, fazemos

$$\phi_1 = \frac{\pi}{2} - \phi, \quad (5.56)$$

e, em seguida

$$\phi_{final} = \pi - \phi_1 + \theta, \quad (5.57)$$

para o lado esquerdo da gota, e

$$\phi_{final} = \phi_1 + \theta, \quad (5.58)$$

para o lado direito da gota. Por fim, calculamos  $\mathbf{n}_2$  por relações trigonométricas, isto é,

$$\mathbf{n}_2 = (\text{sen}(\theta_{final}), \cos(\theta_{final})). \quad (5.59)$$

Finalmente, podemos calcular a distância  $\beta$ , que será a distância entre o ponto  $\mathbf{x}_1$  e a reta  $y = \alpha_0 + \alpha_1 x$ , no contorno rígido. Para isto, escrevemos a reta  $y = \alpha_0 + \alpha_1 x$  como

$$\alpha_1 x - y + \alpha_0 = 0, \quad (5.60)$$

chamada equação geral da reta. Antes, porém, consideremos a equação da reta na sua forma vetorial, escrita como

$$h : \mathbf{x} = \mathbf{s} + t\mathbf{w}, \quad (5.61)$$

onde  $\mathbf{s}$  é um vetor pertencente a reta  $h$ ,  $\mathbf{w}$  é o vetor diretor e  $t \in \mathbb{R}$  (conjunto dos números reais), ver Figura 15.

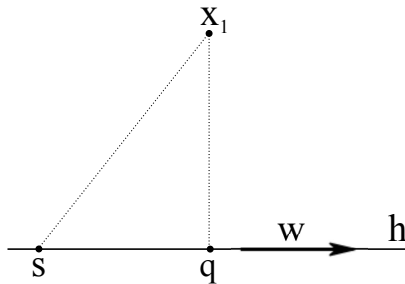


Figura 15: Esquema para o cálculo da distância entre uma reta e um ponto.

Sabemos que a norma do produto vetorial entre os vetores  $\mathbf{s}\mathbf{x}_1$  e  $\mathbf{s}\mathbf{q}$  é a área do paralelogramo gerado por estes vetores. Então, podemos escrever a área do triângulo  $\Delta\mathbf{s}\mathbf{x}_1\mathbf{q}$  como

$$\frac{\|\mathbf{s}\mathbf{x}_1 \wedge \mathbf{s}\mathbf{q}\|}{2} = \frac{\|\mathbf{s}\mathbf{q}\| \|dist(\mathbf{x}_1, h)\|}{2}, \quad (5.62)$$

que implica

$$dist(\mathbf{x}_1, h) = \frac{\|\mathbf{s}\mathbf{x}_1 \wedge \mathbf{s}\mathbf{q}\|}{\|\mathbf{s}\mathbf{q}\|}. \quad (5.63)$$

Note que, podemos escrever o vetor  $\mathbf{s}\mathbf{q}$  em função do vetor diretor  $\mathbf{w}$  da seguinte forma:

$$\mathbf{s}\mathbf{q} = t_0\mathbf{w}, \quad (5.64)$$



onde  $t_0$  é um número real. Assim, a equação (5.63) pode ser reescrita como

$$\text{dist}(\mathbf{x}_1, h) = \frac{\|\mathbf{s}\mathbf{x}_1 \wedge t_0 \mathbf{w}\|}{\|t_0 \mathbf{w}\|} = \frac{|t_0|}{|t_0|} \frac{\|\mathbf{s}\mathbf{x}_1 \wedge \mathbf{w}\|}{\|\mathbf{w}\|}, \quad (5.65)$$

que implica

$$\text{dist}(\mathbf{x}_1, h) = \frac{\|\mathbf{s}\mathbf{x}_1 \wedge \mathbf{w}\|}{\|\mathbf{w}\|}. \quad (5.66)$$

Desta forma, precisamos determinar um vetor diretor  $\mathbf{w}$  e um ponto  $\mathbf{s}$  da reta que descreve o contorno rígido. Note que, a equação vetorial (5.61) pode ser reescrita, na forma paramétrica, como

$$\begin{cases} x = s_1 + tw_1 \\ y = s_2 + tw_2, \end{cases} \quad (5.67)$$

que implica em

$$w_2x - w_1y + (w_1s_2 - w_2s_1) = 0. \quad (5.68)$$

Assim, comparando as equações (5.60) e (5.68), segue que

$$\mathbf{w} = (1, \alpha_1). \quad (5.69)$$

Além disso, tomando  $x = 0$  em (5.60), resulta em  $y = \alpha_0$ . Assim, podemos tomar  $\mathbf{s} = (0, \alpha_0)$ . Uma vez determinado o vetor  $\mathbf{w}$  e o ponto  $\mathbf{s}$ , podemos calcular o produto vetorial  $\mathbf{s}\mathbf{q} \wedge \mathbf{w}$

$$\mathbf{s}\mathbf{q} \wedge \mathbf{w} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ 1 & \alpha_1 & 0 \\ (0 - x_1) & (\alpha_0 - y_1) & 0 \end{vmatrix} = \vec{k}(\alpha_0 - y_1 + x_1\alpha_1). \quad (5.70)$$

Portanto, o valor de  $\beta$  é dado por

$$\beta = \frac{|\alpha_0 - y_1 + x_1\alpha_1|}{\sqrt{1 + \alpha_1^2}}. \quad (5.71)$$

**Observação 5.8.** *Se a equação que representa o contorno rígido em contato com a superfície livre for  $x = c$  (sistema singular), o valor de  $\beta$  será*

$$\beta = |c - x_1|. \quad (5.72)$$

Veja que o procedimento até aqui não é totalmente geral, pois o ângulo de contato não varia, seria apenas um parâmetro de entrada fixo. Assim, um próximo passo deste projeto, será investigar uma forma de inserir o cálculo do ângulo de contato a cada passo no tempo.

## 5.2 Método Numérico

Nesta seção, apresentamos os passos para o cálculo do campo de velocidade  $\mathbf{u}$ , campo de pressão  $p$  e o tensor polimérico  $\boldsymbol{\tau}$  juntamente com a inclusão da tensão superficial. Assim, dados os valores  $\mathbf{u}^{(n)}$ ,  $p^{(n)}$  e  $\boldsymbol{\tau}^{(n)}$ , os passos para o cálculo destas propriedades ao longo do tempo são:

1. Cálculo do tensor polimérico no tempo intermediário  $\overline{\boldsymbol{\tau}}^{(n+1)}$ . Aqui podemos usar algum dos métodos estabilizadores apresentados nos capítulos anteriores;
2. Calculamos a velocidade intermediária  $\tilde{\mathbf{u}}^{(n+1)}$  a partir da equação (5.11);
3. Determinamos a correção  $\nabla\psi^{(n+1)}$  para  $\tilde{\mathbf{u}}$ , mediante o cálculo da equação de Poisson (5.7) utilizando a nova condição de contorno (5.30), que inclui a pressão capilar;
4. A velocidade  $\mathbf{u}^{(n+1)}$  é corrigida pela equação (5.1);
5. A pressão  $p^{(n+1)}$  é calculada pela equação (5.15), também usada na superfície livre;
6. Neste momento, as partículas marcadoras são movimentadas, pela equação (3.22);
7. Por fim, um novo valor para o tensor polimérico  $\boldsymbol{\tau}^{(n+1)}$  é obtido com o segundo passo do método Runge-Kutta de dois passos, preparando para um novo ciclo.

## Considerações finais

Este trabalho apresentou a aplicação do método estabilizador *square-root* para o cálculo de escoamento de fluido viscoelástico confinado e com superfície livre. Os resultados demonstraram a habilidade desta metodologia em termos de eficiência e estabilidade para alto número de Weissenberg, usando o modelo Oldroyd-B. O estudo numérico incluiu o problema da cavidade, escoamento em uma fenda transversal e o problema do impacto da gota.

O método numérico demonstrou estabilidade e eficiência para todos os problemas estudados neste trabalho, o limite do número de Weissenberg aumentou com o uso da formulação *square-root*, quando comparado com o método padrão baseado no tensor polimérico, que usualmente diverge para valores moderados de Weissenberg. A precisão e estabilidade da formulação *square-root* mostrou-se semelhante à formulação *log-conformation*, oferecendo uma metodologia alternativa para abordar o problema de alto número de Weissenberg.

Além disso, neste trabalho, obtemos resultados preliminares com o método *eigendecomposed kernel-conformation*. Estes resultados, mediante comparação com a solução analítica, permitiram a validação desta nova formulação. Assim, um próximo passo deste projeto será a aplicação de um estudo numérico ao método *eigendecomposed kernel-conformation*, semelhante ao que foi feito ao método *square-root*.

Após realizado o estudo e formulação dos efeitos da tensão superficial e de molhamento, mediante o ângulo de contato, um próximo passo deste projeto será a validação da implementação da tensão superficial e do ângulo de contato estático, por meio do problema da gota apoiada. Após isso, será estudado uma forma de empregar o ângulo de contato dinâmico, e realizar um estudo do problema do impacto da gota viscoelástica, considerando os efeitos da tensão superficial e molhamento, juntamente com os métodos estabilizadores para alto número de Weissenberg. Finalmente, aproveitando-se do cálculo do vetor normal unitário na superfície do fluido, usado no cálculo da tensão superficial,

aplicaremos este vetor no cálculo das condições de superfície livre, substituindo o vetor normal por célula, visando melhorar a precisão dos cálculos nesta região.

---

## Bibliografia

- [1] R. B. Bird, R. C. Armstrong, and O. Hassager, *Dynamics of Polymeric Liquids. Vol. 1- Fluid Mechanics*. New York: John Wiley & Sons, 2nd ed., 1987.
- [2] R. B. Bird, C. F. Curtiss, R. C. Armstrong, and O. Hassager, *Dynamics of Polymeric Liquids. Vol. 2- Kinetic Theory*. New York: John Wiley & Sons, 2nd ed., 1987.
- [3] M. Reiner, "The Deborah number," *Physics Today*, vol. 17(1), p. 62, 1964.
- [4] J. L. White, "Dynamics of viscoelastic fluids, melt fracture, and the rheology of fiber spinning," *J. Appl. Polym. Sci.*, vol. 8, pp. 2339–2357, 1964.
- [5] R. J. Poole, "The Deborah and Weissenberg numbers. The British Society of Rheology," *Rheology Bulletin*, vol. 53, pp. 32–39, 2012.
- [6] D. D. Joseph, *Fluid Dynamics of Viscoelastic Liquids*. New York: Springer-Verlag, 1990.
- [7] J. M. M. F. Dupret, "Loss of evolution in the flow of viscoelastic fluid," *J. Non-Newton. Fluid Mech.*, vol. 20, pp. 143–171, 1986.
- [8] R. Sureshkumar and A. N. Beris, "Effect of artificial stress diffusivity on the stability of numerical calculations and the flow dynamics of time-dependent viscoelastic flows," *J. Non-Newton. Fluid Mech.*, vol. 60(1), pp. 53–80, 1995.
- [9] A. N. Beris and R. Sureshkumar, "Simulation of time-dependent viscoelastic channel Poiseuille flow at high Reynolds numbers," *Chem. Eng. Sci.*, vol. 51, pp. 1451–1471, 1997.
- [10] R. Sureshkumar, A. N. Beris, and R. A. Handler, "Direct numerical simulation of turbulent channel flow of a polymer solution," *Phys. Fluids*, vol. 9, pp. 743–755, 1997.
- [11] C. D. D. R. Sureshkumar and A. N. Beris, "Direct numerical simulation of viscoelastic turbulent channel flow exhibiting drag reduction: effect of the variation of rheological parameters," *J. Non-Newton. Fluid Mech.*, vol. 79, pp. 433–468, 1998.

- 
- [12] C. D. D. R. Sureshkumar, A. N. Beris, and R. A. Handler, “Budgets of reynolds stress, kinetic energy and streamwise enstrophy in viscoelastic turbulent channel flow,” *Phys. FLuids*, vol. 13, pp. 1016–1027, 2001.
  - [13] B. Eckhardt, J. Kronjager, and J. Schumacher, “Stretching of polymers in a turbulent environment,” *Comput. Phys. Commun.*, vol. 147, pp. 538–543, 2002.
  - [14] P. Ilg, E. de Angelis, I. V. Karlin, C. M. Casciola, and S. Succi, “Polymer dynamics in wall turbulent flow,” *Europhys. Lett.*, vol. 58, pp. 616–662, 2002.
  - [15] T. Min, J. Y. Yoo, and H. Choi, “Effect of spatial discretization schemes on numerical solutions of viscoelastic fluid flows,” *J. Non-Newton. Fluid Mech.*, vol. 100, pp. 27–47, 2001.
  - [16] Y. Dubief and S. K. Lele, “Direct numerical simulation of polymer flow,” in *Center for Turbulence Research Annual Research Briefs, NASA Ames/Stanford University*, 2001.
  - [17] T. Vaithianathan, A. Robert, J. G. Brasseur, and L. R. Collins, “An improved algorithm for simulating three-dimensional viscoelastic turbulence,” *J. Non-Newt. Fluid Mech.*, vol. 140(1), pp. 3–22, 2006.
  - [18] A. Lozinski and R. G. Owens, “An energy estimate for the oldroyd-b model: theory and applications,” *J. Non-Newt. Fluid Mech.*, vol. 112(2-3), pp. 161–176, 2003.
  - [19] R. Fattal and R. Kupferman, “Constitutive laws for the matrix-logarithm the conformation tensor,” *J. Non-Newton. Fluid Mech.*, vol. 223(2-3), pp. 281–85, 2004.
  - [20] R. Fattal and R. Kupferman, “Time-dependent simulation of viscoelastic flow at high weissenberg number using the log-conformation representation,” *J. Non-Newton. Fluid Mech.*, vol. 26, pp. 23–37, 2005.
  - [21] Y. J. Lee and J. Xu, “New formulations, positivity preserving discretizations and stability analysis for non-newtonian flow models,” *Computers Methods in Applied Mechanics and Engineering*, vol. 195(9-12), pp. 1180–1206, 2006.
  - [22] K. S. Cho, “Vector decomposition of the evolution equations of the conformation tensor of maxwellian fluids,” *Korea-Australia Rheology Journal*, vol. 21(2), pp. 143–146, 2009.
  - [23] N. Balci, B. Thomases, M. Renardy, and C. R. Doering, “Symmetric factorization of the conformation tensor in viscoelastic fluid models,” *J. Non-Newtonian Fluid Mech.*, vol. 166, pp. 546–553, 2011.

- 
- [24] A. M. Afonso, F. T. Pinho, and M. A. Alves, “The kernel-conformation constitutive laws,” *J. Non-Newton. Fluid Mech.*, vol. 167-168, pp. 30–37, 2012.
- [25] P. Saramito, “On a modified non-singular log-conformation formulation for johnson-segalman viscoelastic fluids,” *J. Non-Newton. Fluid Mech.*, vol. 211, pp. 16–30, 2014.
- [26] P. Knechtges, M. Behr, and S. Elgeti, “Fully-implicit log-conformation formulation of constitutive laws,” *J. Non-Newton. Fluid Mech.*, vol. 214, pp. 78–87, 2014.
- [27] Y. Kwon, “Finite element analysis of planar 4:1 contraction flow with the tensor-logarithmic formulation of differential constitutive equations,” *Korea-Australia Theology Journal*, vol. 16, pp. 183–191, 2004.
- [28] O. M. Coronado, D. Arora, M. Behr, and M. Pasquali, “A simple method for simulating general viscoelastic fluid flows with an alternate log-conformation formulation,” *J. Non-Newton. Fluid Mech.*, vol. 147, pp. 189–199, 2007.
- [29] R. Gu  nette, A. Fortin, A. Kane, and J. F. H  tu, “An adaptive remeshing strategy for viscoelastic fluid flow simulations,” *J. Non-Newton. Fluid Mech.*, vol. 153, pp. 34–45, 2008.
- [30] A. Afonso, P. J. Oliveira, F. T. Pinho, and M. A. Alves, “The log-conformation tensor approach in the finite-volume method framework,” *J. Non-Newton. Fluid Mech.*, vol. 157, pp. 55–65, 2009.
- [31] A. Jafari, N. Fi  tier, and M. O. Deville, “A new extended matrix logarithm formulation for the simulation of viscoelastic fluids by spectral elements,” *Comput. Fluids*, vol. 39, pp. 1425–1438, 2010.
- [32] H. Damanik, J. Hron, A. Ouazzi, and S. Turek, “A monolithic fem approach for the log-conformation reformulation (lcr) of viscoelastic flow problems,” *J. Non-Newton. Fluid Mech.*, vol. 165, pp. 1105–1113, 2010.
- [33] X. Chen, H. Marshall, M. Schafer, and D. Bothe, “A comparasion of stabilization approaches for finite-volume simulation of viscoelastic fluid flow,” *International Journal of Computational Fluid Dynamics*, vol. 27, pp. 229–250, 2013.
- [34] F. P. Martins, C. M. Oishi, A. M. Afonso, and M. A. Alves, “A numerical study of the kernel conformation transformation for transient viscoelastic fluid flows.” submitted, 2014.
- [35] P. J. Oliveira, F. T. Pinho, and G. A. Pinto, “Numerical simulation of non-linear elastic flows with a general collocated finite-volume method,” *J. Non-Newtonian Fluid Mech.*, vol. 79, p. 1, 1998.

- 
- [36] M. A. Alves, F. T. Pinho, and P. J. Oliveira, “Effect of a high resolution differencing scheme on finite-volume predictions of viscoelastic flows,” *J. Non-Newton. Fluid Mech.*, vol. 93, pp. 297–314, 2000.
  - [37] M. A. Alves, P. J. Oliveira, and F. T. Pinho, “A convergent and universally bounded interpolation scheme for the treatment of advection,” *International Journal for Numerical Methods*, vol. 41, pp. 47–75, 2003.
  - [38] C. M. Oishi, F. P. Martins, M. F. Tomé, J. A. Cuminato, and S. Mckee, “Numerical solution of the extended pom-pom model for viscoelastic free surface flows,” *J. Non-Newton. Fluid Mech.*, vol. 166, pp. 165–179, 2011.
  - [39] J. A. Cuminato, C. M. Oishi, and R. A. Figueiredo, *Implicit method for simulating low Reynolds number free surface flows: Improvements on the MAC-type methods*. Japan: Springer, 2014.
  - [40] I. L. Palhares, C. M. Oishi, A. M. Afonso, M. A. Alves, and F. T. Pinho, “Numerical study of the square root conformation tensor formulation for confined and free-surface viscoelastic fluid flows.” submitted, 2015.
  - [41] P.-G. de Gennes, F. Brochard-Wyart, and D. Quéré, *Capillarity and Wetting Phenomena Drops, Bubbles, Pearls, Waves*. Springer, 2003.
  - [42] M. Pasandideh-Fard, Y. M. Qiao, S. Chandra, and J. Mostaghimi, “Capillary effects during droplet impact on a solid surface,” *Phys. Fluids*, vol. 8, pp. 650–659, 1996.
  - [43] D. Bortolo, A. Boudaoud, G. Narcy, and D. Bonn, “Dynamics of non-newtonian droplets,” *Phys. Review Letters*, vol. 99, pp. 1–4, 2007.
  - [44] J. Cooper-White, R. C. Crooks, and D. V. Boger, “A drop impact study of worm-like viscoelastic surfactant solutions,” *Colloids and Surfaces*, vol. 210, pp. 105–123, 2002.
  - [45] S. Gier, S. Dorbolo, D. Terwagne, N. Vandewalle, and C. Wagner, “Bouncing of polymeric droplets on liquid interfaces,” *Phys. Review*, vol. 86, pp. 1–7, 2012.
  - [46] A. S. Moita, D. Herrmann, and A. L. N. Moreira, “Experimental characterization of the post-impact behaviour of non-newtonian droplets,” in *17<sup>th</sup> International Symposium on Applications of Laser Techniques to Fluid Mechanics*, (Lisboa, Portugal), 2014.
  - [47] M. Tembely, D. Vadillo, A. Soucemarianadin, and A. Dolatabadi, “On the modeling of viscoelastic droplet impact dynamics,” in *4<sup>th</sup> Micro and Nano Flows Conference UCL*, (London, UK), 2014.



- 
- [48] M. J. Crochet and R. Keunings, “Finite element analysis of die swell of a highly elastic fluid,” *J. Non-Newton. Fluid Mech.*, vol. 10, pp. 339–356, 1982.
  - [49] N. F. Morrison and O. G. Harlen, “Viscoelastic in inkjet printing,” *Rheol. Acta*, vol. 49, pp. 619–632, 2010.
  - [50] M. F. Tomé, N. Mangiavacchi, J. A. Cuminato, A. Castelo, and S. McKee, “A finite difference technique for simulating unsteady viscoelastic free surface flows,” *J. Non-Newtonian Fluid Mech.*, vol. 106, pp. 61–106, 2002.
  - [51] M. F. Tomé, G. S. Paulo, F. T. Pinho, and M. A. Alves, “Numerical solution of the ptt constitutive equation for unsteady three-dimensional free surface flows,” *J. Non-Newton. Fluid Mech.*, vol. 165, pp. 247–262, 2010.
  - [52] J. Fang, R. G. Owens, L. Tacher, and A. Parriaux, “A numerical study of the sph method for simulating transient viscoelastic free surface flows,” *J. Non-Newt. FLuid Mech.*, vol. 139, pp. 68–84, 2006.
  - [53] W. Clemens, W. Fix, J. Ficker, A. Knobloch, and A. Ullmann, “From polymer transistor toward printed electronics,” *J. Mater. Res.*, vol. 19, pp. 1963–1973, 2004.
  - [54] J. Xu and D. Attinger, “Drop on demand in a microfluidic chip,” *J. Micromech. Microeng.*, vol. 18, p. 65020, 2008.
  - [55] M. Singh, H. M. Haverinen, P. Dhagat, and G. E. Jabbour, “Inkjet printing process and its applications,” *Adv. Mater.*, vol. 22, pp. 673–685, 2010.
  - [56] M. Meier, G. Yadigaroglu, and B. L. Smith, “A novel technique for including surface tension in plic-vof methods,” *Eur. J. Mech.*, vol. 21, pp. 61–73, 2002.
  - [57] Y. Renardy and M. Renardy, “Prost: A parabolic reconstruction of surface tension for the volume-of-fluid method,” *J. Comput. Phys.*, vol. 183, pp. 400–421, 2002.
  - [58] S. Popinet and S. Zaleski, “A front tracking algorithm for accurate representation of surface tension,” *Internat. J. Numer. Methods Fluids*, vol. 30, pp. 775–793, 1999.
  - [59] S. F. Kistler, *In Hydrodynamics of Wetting*. Inc.: New York: J. C. Berg, ed. wettability; m. dekker ed., 1993.
  - [60] N. Mangiavacchi, A. Castelo, M. F. Tomé, J. A. Cuminato, and M. L. B. S. McKee, “An effective implementation of surface tension using the marker and cell method for axisymmetric and planar flows,” *SIAM J. Sci. Comput.*, vol. 26, pp. 1340–1368, 2005.

- 
- [61] C. M. Oishi, “Análise e implementação de métodos implícitos no sistema freeflow2d,” Master’s thesis, ICMC/USP, São Carlos, 2004.
  - [62] C. M. Oishi, *Análise e implementação de metodos implícitos e de projeção para escoamentos com superficie livre*. PhD thesis, ICMC/USP, São Carlos, 2008.
  - [63] C. M. Oishi, J. A. Cuminato, V. Ferreira, M. F. Tomé, A. F. Castelo, N. Mangia-vacchi, and S. Mckee, “A stable semi-implicit method for free surface flows,” *J. of Applied Mech.*, vol. 73, pp. 940–947, 2006.
  - [64] C. M. Oishi, J. A. Cuminato, J. Y. Yuan, and S. Mckee, “Stability of numerical schemes on staggered grids,” *Numerical Linear Algebra with Applications*, vol. 15, pp. 945–967, 2008.
  - [65] A. Chorin and J. Marsden, *A mathematical introduction to fluid mechanics*. Springer, 2000.
  - [66] F. Pan and A. Acrivos, “Steady flows in a rectangular cavities,” *J. Non-Newton. Fluid Mech.*, vol. 28(4), pp. 165–179, 1967.
  - [67] P. Shankar and M. Deshpande, “Fluid mechanics in the driven cavity,” *Ann. Rev. Fluid Mech.*, vol. 32(1), pp. 93–136, 2000.
  - [68] T. W. Pan, J. Hao, and R. Glowinski, “On the simulation of a time-dependent cavity flow of an oldroyd-b fluid,” *Int. J. Numer. Meth. Fluids*, vol. 60, pp. 791–808, 2009.
  - [69] K. Yapici, B. Karasozen, and Y. Uludog, “Finite volume simulation of viscoelastic laminar ow in a lid-driven cavity,” *J. Non-Newton. FLui Mech.*, vol. 164(1), pp. 51–65, 2009.
  - [70] R. Poole, A. Afonso, F. T. Pinho, P. Oliveira, and M. A. Alves, “Scaling of purely-elastic instabilities in viscoelastic lid-driven cavity flow,” in *Scaling of purely-elastic instabilities in viscoelastic lid-driven cavity flow*, XVIth International Workshop of Numerical methods in Non-Newtonian Flows, 2010.
  - [71] J. Su, J. Ouyang, X. Wang, B. Yang, and W. Zhou, “Lattice boltzmann method for the simulation of viscoelastic fluid flows over a large range of weissenberg numbers,” *J. Non-Newton. FLui Mech.*, vol. 194, pp. 42–59, 2013.
  - [72] F. Habla, M. W. Tan, J. Hablberger, and O. Hinrichsen, “Numerical simulation of the viscoelastic flow in a three-dimensional lid-driven cavity using the log-conformation reformulation in openfoam,” *J. Non-Newton. FLui Mech.*, vol. 212, pp. 47–62, 2014.

- 
- [73] G. G. Fuller, C. A. Cathey, B. Hubbard, and B. E. Zebrowski, “Extensional viscoelasticity measurements for low viscosity fluids,” *J. of Rheology*, vol. 31, pp. 235–249, 1987.
  - [74] P. E. Arratia, C. C. Thomas, J. D. Diorio, and J. P. Gollub, “Elastic instabilities of polymer solutions in cross-channel flow,” *Physical Review Letters*, pp. 144–502, 2006.
  - [75] R. J. Poole, M. A. Alves, and P. J. Oliveira, “Purely elastic flow asymmetries,” *Physical Review Letters*, vol. 99, pp. 164–503, 2007.
  - [76] F. A. Cruz, R. J. Poole, A. M. Afonso, F. T. Pinho, P. J. Oliveira, and M. A. Alves, “A new viscoelastic benchmark flow: stationary bifurcation in a cross slot,” *J. Non-Newt. Fluid Mech.*, pp. 214–57, 2014.
  - [77] D. O. A. Cruz and F. T. Pinho, “Analytical solution of steady 2d wall-free extensional flows of ucm fluids.” submitted, 2015.
  - [78] A. Raffee, M. T. Manzari, and M. Hossini, “An incompressible sph method for simulation of unsteady viscoelastic free-surface flows,” *Int. J. Non-Linear Mech.*, vol. 42, pp. 1210–1223, 2007.
  - [79] T. Jiang, J. Ouyang, B. Yang, and J. Ren, “The sph method for simulating a viscoelastic drop impact and spreading on an inclined plate,” *Comp. Mech.*, vol. 45, pp. 573–583, 2010.
  - [80] C. M. Oishi, F. P. Martins, M. F. Tomé, and M. A. Alves, “Numerical simulation of drop impact and jet buckling problems using the extended pom-pom model,” *J. Non-Newt. Fluid Mech.*, vol. 169–170, pp. 91–103, 2012.
  - [81] T. Vaithianathan and R. L. R. Collins, “Numerical approach to simulating turbulent flow of a viscoelastic polymer solution,” *Journal of Computational Physics*, vol. 187, pp. 1–21, 2003.
  - [82] A. Iserles, *Lie-group methods*. Acta Numer., 2000.
  - [83] J. Gallier, “Remarks on the cayley representation of orthogonal matrices and on perturbing the diagonal of a matrix to make it invertible,” tech. rep., Cornell University Library, Philadelphia, USA, 12 2013.
  - [84] A. C. Filho, N. Mangiavacchi, M. F. Tomé, J. A. Cuminato, A. O. Fortuna, J. de Oliveira, V. G. Ferreira, and S. Mckee, “Surface tension implementation for gensmac2d,” *J. of the Brazilian Society of Mech. Sciences*, vol. 23, 2001.
  - [85] M. F. Tomé, *GENSMAC: a multiple free surface fluid flow solver*. PhD thesis, University of Ayrathclyde, Scotland, 1993.

- 
- [86] D. L. Brown, R. Crotez, and M. L. Minion, “Accurate projection methods for the incompressible navier-stokes equations,” *J. Comp. Physics*, vol. 168, pp. 464–499, 2001.
- [87] F. P. Martins, “Desenvolvimento de um método numérico implícito para a simulação de escoamentos viscoelásticos com superfícies livres,” Master’s thesis, ICMC/USP, São Carlos, 2009.
- [88] G. K. Batchelor, *An introduction to fluid dynamics*. Cambridge: Cambridge University Press, 1967.
- [89] J. H. Ferziger and M. Peric, *Computational methods for fluid dynamics*. New York: Springer, 3rd edition ed., 2002.
- [90] M. Sussman and P. Smereka, “Axisymmetric free boundary problems,” *J. Fluid Mech.*, vol. 341, pp. 269–294, 1988.
- [91] H. Alencar and W. Santos, *Geometria diferencial das curvas planas*. Rio de Janeiro: Publicações IMPA, 2003.
- [92] P. V. Araújo, *Geometria diferencial*. Rio de Janeiro: Publicações IMPA, 1998.