

## Ensino

# Escoamento Couette entre placas paralelas: estudo de caso para o ensino de dinâmica de fluidos computacional

Couette flow between parallel plates: a benchmark case for teaching computational fluid dynamics

Livia Flavia Carletti Jatobá<sup>1</sup> , Sthefany Machado Sardinha<sup>1</sup> 

<sup>1</sup>Instituto Politécnico, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, Brasil

## RESUMO

A Dinâmica dos Fluidos Computacional resolve numericamente as equações do escoamento de fluidos. O ensino dessa área abrange métodos numéricos, modelagem e ferramentas computacionais, sendo um desafio interdisciplinar desenvolver conteúdo didático para graduação. O presente trabalho apresenta metodologia para o ensino do método dos volumes finitos e das etapas de uma simulação de dinâmica dos fluidos através do estudo de caso do escoamento Couette entre placas planas. O trabalho apresenta o conteúdo didático para dedução das equações discretas usando o método dos volumes finitos para o escoamento estacionário 1D puramente difusivo. A montagem e solução do sistema algébrico é feita através da ferramenta livre, CalcMe, um aplicativo online para cálculos algébricos. Em seguida, o mesmo caso é resolvido usando uma ferramenta de simulação livre, o OpenFOAM®, e o produto didático é apresentado no formato de um tutorial. Por fim, o presente trabalho adota apenas ferramentas livres e apresenta metodologia que visa mitigar os efeitos da simulação “caixa preta”, através de um material didático explicativo, que visa ampliar o acesso ao conhecimento da solução numérica e usabilidade de uma ferramenta de simulação de dinâmica dos fluidos.

**Palavras-chave:** Método dos volumes finitos; Escoamento Couette; CalcMe; OpenFOAM; CFD

## ABSTRACT

Computational Fluid Dynamics numerically solves the equations of fluid flow. The syllabus covered in the area includes numerical methods, modeling, and computational tools, making it an interdisciplinary challenge to develop material for undergraduate courses. This work presents a methodology for teaching the finite volume method and the steps of a fluid dynamics simulation through the case study of Couette flow between parallel plates. The work presents the didactic content for deriving the discrete equations using the finite volume method for 1D steady-state purely diffusive flow. The assembly and solution of the algebraic system uses a free tool, CalcMe, an online application for algebraic calculations. Then, the

same case is solved using a free simulation tool, OpenFOAM®, and the teaching material is presented in the format of a tutorial. Finally, this work adopts only free tools and presents a methodology aimed at mitigating the effects of “black-box” simulation through explanatory didactic material, which aims to broaden access to knowledge of numerical solution and usability of a fluid dynamics simulation tool.

**Keywords:** Finite volume methods; Couette Flow; CalcMe; OpenFOAM; CFD

## 1 INTRODUÇÃO

A Dinâmica dos Fluidos Computacional, do inglês *Computational Fluid Dynamics* (CFD), é uma área do conhecimento que trata da solução numérica das equações que governam o escoamento de fluidos. O conteúdo abordado no seu ensino é extenso incluindo métodos e algoritmos numéricos, modelagem de fenômenos de transporte e, em alguns casos, linguagem de programação e/ou ferramentas computacionais. A natureza interdisciplinar da área é um desafio, particularmente no desenvolvimento de um curso para a graduação.

A capacidade de modelar os fenômenos físicos utilizando ferramentas matemáticas, computacionais e de simulação é uma competência comum a muitos cursos na área de ciências exatas e engenharia. Assim, o ensino de ferramentas computacionais consiste em uma importante habilidade na formação dos estudantes de cursos como engenharia, física e matemática. O ensino prático através de ferramentas computacionais capacita estudantes na aplicação de conhecimentos matemáticos e tecnológicos à problemas físicos, ao mesmo tempo em que os familiariza com essas novas ferramentas.

Diferentes ferramentas computacionais podem ser adotadas no ensino de Dinâmica dos Fluidos Computacional. A escolha entre aplicativos de simulação CFD, implementação das soluções numéricas usando uma linguagem de programação ou uma ferramenta de cálculo algébrico depende da complexidade dos casos estudados e do objetivo didático. A utilização de ferramentas de código aberto e livre para este fim, promove uma cultura de compartilhamento de conhecimento e educação aberta, ações integradas no presente trabalho e fundamentais para difusão dos conhecimentos desenvolvidos para além do seu ecossistema original.

Uma das primeiras publicações que utilizou CFD para ensinar os conceitos de aerodinâmica foi o trabalho de Hoist (1994), que destaca a capacidade no desenvolvimento de conceitos práticos, a parametrização para otimização de

resultados e a habilidade de visualizar o escoamento em aplicações aerodinâmicas. CFD era uma área do conhecimento restrita a estudantes de pós-graduação (Dales & Willis, 2010), mas que ganhou espaço no currículo de estudantes de graduação em engenharia em diversas universidades (Stern et al., 2006; Hu et al., 2008). Observamos um maior número de publicações didáticas e recursos tecnológicos na área de Dinâmica dos Fluidos Computacional, tanto em inglês (Pawar & San, 2019; Barba & Forsyth, 2018; Barba & Mesnard, 2019; LaRoche et al., 2002) quanto em português (Adair & Jaeger, 2019; Wehmann et al., 2018; Rocha & da Silveira, 2012; Vasata et al., 2011).

O presente trabalho tem como contribuição apresentar metodologia para promover a compreensão dos procedimentos para obtenção de uma solução numérica e respectiva simulação CFD através de um estudo de caso. O objetivo é apresentar produto didático para utilização no ensino de Dinâmica dos Fluidos Computacional, que adota ferramentas computacionais livres, com foco no ensino do método dos volumes finitos através do clássico escoamento Couette entre placas planas. A proposta é apresentar o caso em diferentes níveis de complexidades, desde a construção da solução numérica e solução do sistema algébrico, até a aplicação do caso em uma ferramenta de simulação CFD.

O presente trabalho está organizado da seguinte forma: os trabalhos relevantes são discutidos na seção 2; a metodologia de ensino é apresentada na seção 3; a seção 4 contém o descritivo da dedução das equações discretas do escoamento Couette entre placas planas pelo método dos volumes finitos e a solução usando a ferramenta computacional CalcMe <sup>1</sup>; a seção 5 é o tutorial para simulação CFD do mesmo caso usando o OpenFOAM<sup>®2</sup>. Por fim, as considerações finais são apresentadas na seção 6.

## **2 TRABALHOS RELEVANTES**

Observa-se na literatura o ensino, ou uso, de Dinâmica dos Fluidos Computacional em diferentes abordagens. A primeira tem como foco o ensino dos métodos numéricos e métodos de solução de sistemas algébricos. Esta abordagem é mais usada em cursos de graduação e pós-graduação e espera-se que os discentes já

---

<sup>1</sup> CalcMe é uma calculadora online livre, distribuída pela Wiris, que funciona em qualquer navegador.

<sup>2</sup> O OpenFOAM<sup>®</sup> é um aplicativo livre de dinâmica dos fluidos computacional.

tenham adquiridos alguns conhecimentos prévios, tais como, mecânica dos fluidos, transferência de calor e cálculo diferencial e integral. A parte prática desta abordagem exige o conhecimento de uma linguagem de programação, para implementação das soluções numéricas, ou de ferramentas de cálculo algébrico. Os livros didáticos de Dinâmica dos Fluidos Computacional seguem esta estrutura e os casos estudados são tipicamente a equação de transporte para uma grandeza escalar, como a temperatura ou concentração (Versteeg & Malalasekera, 2007; Maliska, 2017).

A solução numérica da equação de Navier-Stokes já é uma etapa de maior complexidade, acompanhada dos algoritmos de acoplamento pressão-velocidade (Fortuna, 2020; Patankar, 1980). Barba & Forsyth (2018) reportam um repositório, de livre acesso, para ensinar os conceitos da implementação da solução numérica da equação de Navier-Stokes em 12 passos, usando Python, onde o caso do escoamento 2D em uma cavidade e um canal são as etapas finais do produto didático. É possível encontrar ainda repositórios livres com a solução numérica de diferentes problemas da literatura usando a linguagem de programação Julia, onde o caso do escoamento em uma cavidade é novamente um dos casos finais da trilha de aprendizagem (Pawar & San, 2019). Outro exemplo de repositório livre com propostas de solução para o escoamento em uma cavidade é o reportado por Battista (2020), que usa MATLAB e Python como ferramentas. Outras estratégias podem ser encontradas na literatura (Adair & Jaeger, 2019; Vasata et al., 2011) e, de modo geral, as publicações mostram a maior complexidade na solução numérica da equação de Navier-Stokes, particularmente em casos bidimensionais e com acoplamento pressão-velocidade, sugerindo que esta não seja a opção mais apropriada para estudos introdutórios na área.

A segunda estratégia tem como foco o ensino de Dinâmica dos Fluidos Computacional através dos aplicativos de simulação CFD. Os aplicativos de simulação CFD são programas onde as soluções numéricas do escoamento de fluidos já foram implementadas dentro de diversos contextos, tal como um escoamento incompressível ou compressível, laminar ou turbulento, com ou sem troca térmica, monofásico ou multifásico etc. Assim, o estudante não precisa implementar a solução numérica, mas precisa aprender a manusear a ferramenta computacional e conhecer as etapas necessárias para realizar uma simulação CFD. Além disso, é importante

ensinar aos estudantes a fazer as escolhas e definições dos parâmetros da solução numérica adequadas para cada tipo de escoamento, tal como a configuração das condições de contorno, esquemas de interpolação e solução do sistema algébrico. Por fim, tópicos como convergência numérica, convergência de malha e a capacidade de analisar a qualidade dos resultados obtidos também devem ser abordados.

Os aplicativos de simulação CFD podem ser proprietários ou livres. A vantagem em adotar uma ferramenta proprietária é a interface de usuário mais amigável e maior quantidade de documentações e manuais disponibilizada pelas empresas. A desvantagem é o investimento em licenças. Outra importante discussão presente na literatura é a simulação com erros, ou baixa acurácia, devido a facilidade no uso das ferramentas CFD proprietárias por usuários sem a devida capacitação (Hu et al., 2008; Stern et al., 2006). Os aplicativos livres não possuem custo de licença porém, de modo geral, tem interface menos intuitiva e menor quantidade de documentação e manuais. O desafio desta segunda abordagem consiste em garantir que estudantes sejam capacitados a fazer escolhas adequadas ao longo do processo de simulação.

As publicações dentro do escopo desta segunda abordagem apresentam foco no ensino da Dinâmica dos Fluidos Computacional através de estudo de casos ou na demonstração da ferramenta de simulação para fins didáticos. A publicações de Wang & Gao (2017) e Polanco et al. (2009) trabalham o ensino de CFD através de exemplos, tais como, o escoamento em bomba axial, tanque de mistura e escoamento em torno de um carro. A publicação de Rodríguez-Martín et al. (2019) utiliza o método de aprendizagem baseada em problemas para trabalhar os conceitos de mecânica dos fluidos e CFD, onde o docente atua como orientador em um processo de ensino em que estudante trabalha de forma autônoma. Stern et al. (2006) e Sert & Nakiboglu (2007) apresentam exemplos de interfaces educacionais para o ensino de CFD através de estudos de casos, tais como, escoamento em tubulações, escoamento em uma cavidade e escoamento em torno de um aerofólio.

A comunidade científica brasileira também tem trabalhado para ampliar o acesso às ferramentas CFD (Rocha & da Silveira, 2012; Wehmann et al., 2018; Vasata et al., 2011). Rocha & da Silveira (2012) apresentam estudos de casos que podem ser utilizados como ferramentas didáticas para ensinar verificação e validação de modelos matemáticos. Os autores selecionaram casos clássicos da mecânica dos fluidos, como o escoamento em

torno de um aerofólio, em torno de uma placa plana, um viscosímetro e um caso de misturador com troca térmica. Apresentam materiais e métodos para os experimentos e informações para a simulação CFD com um aplicativo proprietário. Este trabalho é um relevante exemplo de como unir a experimentação e simulação no escoamento de fluidos.

A terceira, e última, abordagem não se concentra no ensino da solução numérica, mas sim na aplicação da Dinâmica dos Fluidos Computacional para aprimorar a compreensão do escoamento de fluidos em diversas disciplinas de graduação (Souza & de Oliveira, 2016; Solmaz & Van Gerven, 2022) e, inclusive, no ensino médio (Vieira & Pires, 2022; Werlang et al., 2008; Clebsch & Mors, 2004). Neste caso, a ferramenta computacional desempenha um papel fundamental no aprofundamento dos fundamentos do escoamento de fluidos. O objetivo não é ensinar os princípios de uma simulação CFD, ou ainda a solução numérica, mas sim adotar de maneira simplificada o simulador como um recurso educacional ou laboratório virtual, proporcionando suporte ao ensino da mecânica dos fluidos. Os trabalhos de Werlang et al. (2008), Souza & de Oliveira (2016) e Dias Fernandes & Moreira (2019) apresentam produtos educacionais para melhorar a conexão entre os conteúdos de dinâmica dos fluidos e vivências práticas. Outro exemplo inovador é o desenvolvimento de ambiente de ensino digital interativo através de realidade virtual para o ensino de CFD (Solmaz & Van Gerven, 2022).

Apesar do relevante número de estudos que ressaltam a importância de métodos de ensino inovadores na área de Dinâmica dos Fluidos Computacional, ainda há espaço para o desenvolvimento de trabalhos que desmistifiquem e simplifiquem a solução numérica da equação de Navier-Stokes. O presente trabalho tem como escopo a primeira e segunda abordagens de ensino e apresenta metodologia que visa mitigar os efeitos da simulação “caixa preta”, através de um material didático explicativo, que visa ampliar o acesso ao conhecimento da solução numérica e usabilidade de uma ferramenta de simulação CFD livre.

### **3 METODOLOGIA**

O método de ensino neste trabalho foi desenvolvido para aplicação em disciplina eletiva do curso de graduação em engenharia mecânica, com carga horária

de 75 horas por semestre. O objetivo de aprendizagem da disciplina é que os estudantes compreendam os princípios fundamentais de uma simulação CFD, os métodos numéricos incorporados em uma ferramenta de simulação e o aprendizado de uma ferramenta de simulação.

O conteúdo didático apresentado a seguir é distribuído com uma carga horária de 10 horas. As aulas anteriores são teóricas e apresentam os princípios de uma simulação CFD, equações que governam o escoamento de fluidos incompressíveis, princípio dos métodos das diferenças finitas e dos volumes finitos. O conteúdo desta publicação trata do primeiro exemplo aplicado para o ensino do método dos volumes finitos. Assim, o mesmo pode ser descrito como material didático para o ensino da solução do escoamento 1D puramente difusivo e estacionário. Foram usadas condições de contorno de 1º tipo e interpolação linear. O material cobre, para o estudo de caso escolhido, a dedução das equações discretas, solução usando ferramenta de cálculo algébrico e o tutorial usando uma ferramenta de simulação CFD. O ensino de métodos de solução de sistemas algébricos está fora do escopo deste trabalho.

De modo geral, os livros didáticos da área utilizam o caso da condução de calor em uma barra de ferro para exemplificar a solução numérica da equação de transporte 1D puramente difusivo e estacionária (Versteeg & Malalasekera, 2007; Maliska, 2017). A proposta aqui é escolher um caso semelhante, mas aplicado ao escoamento de fluidos newtonianos incompressíveis. Assim, o escoamento de Couette entre placas planas com pressão uniforme foi escolhido, pois tem as mesmas características matemáticas que a condução de calor.

Além do escoamento de Couette entre placas paralelas com pressão uniforme, a literatura de mecânica dos fluidos para escoamentos incompressíveis viscosos apresenta o caso do escoamento de Couette com gradiente de pressão e o escoamento de Poiseuille, que é o escoamento com gradiente de pressão entre placas imóveis (Young et al., 2011). Enquanto o escoamento de Couette com pressão uniforme é caracterizado por um perfil de velocidade linear, os demais apresentam uma distribuição não linear de velocidade ao longo da seção transversal das placas. A complexidade adicional do perfil de velocidade não linear e a influência de gradientes de pressão tornam a análise matemática e computacional mais desafiadora em

comparação com o escoamento de Couette com gradiente de pressão nulo. Portanto, a escolha do escoamento Couette com pressão uniforme se dá porque este representa o caso mais simples para a introdução da solução numérica da equação de Navier-Stokes.

A metodologia de ensino faz uso de duas ferramentas computacionais livres: o CalcMe e o OpenFOAM®. O CalcMe, é uma ferramenta gratuita disponível online, desenvolvida em JavaScript, que realiza cálculos algébricos. A ferramenta pode ser acessada através do website <https://calcme.com>, ou integrada à ambientes de aprendizagem virtual, como o Moodle. O OpenFOAM® é a ferramenta computacional escolhida para realizar as simulações CFD. Foi utilizada a versão 10 distribuída pela The OpenFOAM Foundation em <https://openfoam.org>. É importante destacar que o OpenFOAM® não possui interface gráfica e todas as simulações consistem na chamada de executáveis usando o terminal. As aulas são realizadas em laboratório de informática e o sistema operacional usado foi o Ubuntu, uma distribuição do Linux.

O material desenvolvido pode ser apresentado aos estudantes através do método expositivo ou através de métodos ativos, como classe invertida ou aprendizagem baseada em problemas. O material foi organizado em duas etapas: a primeira consiste na dedução das equações discretas e solução usando CalcMe e, a segunda é a solução através do aplicativo de simulação CFD, o OpenFOAM®. Os estudantes têm acesso ao material didático organizado em uma trilha de aprendizagem no Moodle.

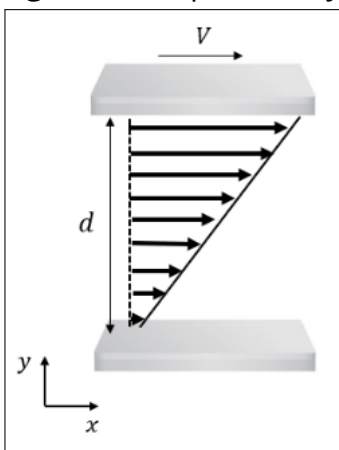
O material didático para a primeira parte consiste em uma apostila, impressa ou em formato digital, com a dedução das equações discretas usando o método dos volumes finitos. Após a apresentação deste conteúdo, os estudantes realizam a solução algébrica usando o CalcMe, integrado a um questionário do Moodle. A descrição e resultados desta primeira etapa são apresentados na seção 4. Em seguida, a configuração da simulação CFD é feita acompanhada de uma apostila e os arquivos do tutorial do OpenFOAM®. A descrição desta etapa e resultados são apresentados na seção 5.



### 3.1 Definição do problema

O problema físico estudado consiste no escoamento laminar de um fluido newtoniano incompressível e plenamente desenvolvido entre duas placas planas paralelas, onde a placa superior desloca-se com velocidade conhecida e a inferior é fixa. A Figura 1 mostra a geometria do problema. A força motriz do escoamento é o gradiente de velocidade entre a placa superior e inferior e, portanto, o transporte difusivo de quantidade de movimento linear. Devido à condição de não deslizamento, o fluido em contato com as placas tem a mesma velocidade das placas. Assim, o fluido em contato com a placa superior transfere quantidade de movimento linear para as camadas de fluido adjacentes até a camada de fluido em contato com a placa inferior, que permanece parada. Neste tipo de problema o gradiente de pressão é nulo, ou seja, a pressão é uniforme. O escoamento é conhecido na literatura como escoamento Couette com pressão uniforme (Young et al., 2011).

Figura 1 – Representação simplificada do escoamento entre placas planas



A Equação da Continuidade,

$$\nabla \cdot \mathbf{U} = 0 \quad (1)$$

e a Equação de Navier-Stokes simplificada, com viscosidade constante e força gravitacional desprezível, são as equações que governam este escoamento

$$\nabla \cdot (\nabla \mathbf{U}) = 0. \quad (2)$$

O sistema de coordenadas cartesianas é considerado e a velocidade tem componente diferente de zero apenas na direção  $x$ , ou seja,  $U_y = U_z = 0$ . Assim, a Equação da Continuidade determina que  $\partial U_x / \partial x = 0$  e a Equação (2) pode ser simplificada por

$$\frac{d^2 U_x}{dy^2} = 0 \quad (3)$$

e a componente  $x$  da velocidade é função apenas de  $y$  ( $U_x(y)$ ). A condição de contorno para a velocidade é de primeiro tipo, onde as velocidades são conhecidas nas fronteiras superior e inferior e  $d$  é a distância entre as placas:  $U_x(y = 0) = 0$  ;  $U_x(y = d) = V$ . A solução analítica para o perfil de velocidade é

$$U_x(y) = \frac{V}{d}y \quad (4)$$

onde  $V$  e  $d$  são constantes.

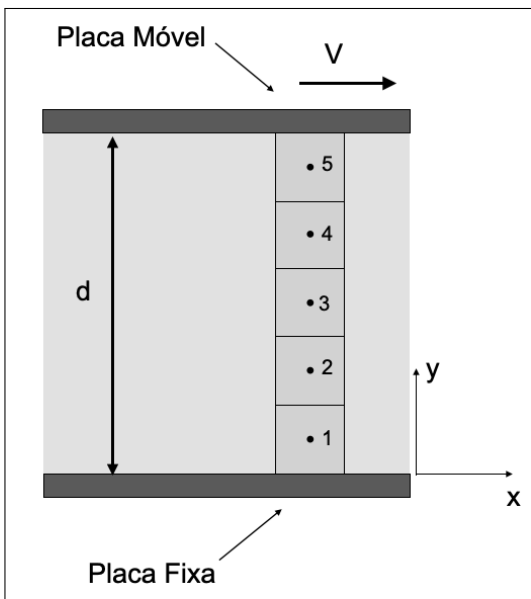
A caracterização do problema é apresentada aos estudantes pois o principal objetivo didático consiste em ensinar as etapas necessárias para verificação da solução numérica com a solução analítica. O problema do escoamento Couette entre placas planas foi escolhido pois trata-se de um caso clássico, amplamente abordado em disciplinas de mecânica dos fluidos e simplificado, uma vez que a pressão é uniforme e a única equação que precisa ser resolvida é a para componente  $x$  da velocidade. Assim, toda a complexidade de novo aprendizado para os estudantes reside no entendimento da solução numérica pelo método dos volumes finitos.

#### 4 MVF: ESCOAMENTO COUETTE ENTRE PLACAS PLANAS

O Método dos Volumes Finitos (MVF) consiste em um método de solução numérica para equações diferenciais parciais usado em Dinâmica dos Fluidos Computacional (Versteeg & Malalasekera, 2007). A primeira etapa consiste na discretização, ou construção da malha, para a solução numérica. A Figura 2 mostra a geometria e malha para o escoamento entre as placas planas onde  $d$  é a distância entre as placas.

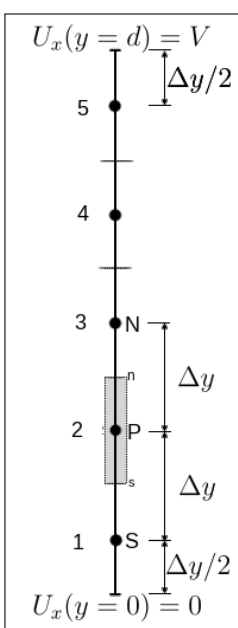
A Figura 3 mostra a notação adotada para descrever a malha uniforme, onde  $\Delta y = d/N$ ,  $N = 5$  é o número de volumes de controle da malha. Considere a seguinte notação:  $P$  é o centro do volume de controle,  $N$  é o centro do volume de controle à norte de  $P$  e  $S$  é o centro do volume de controle à sul de  $P$ . As faces do volume de

Figura 2 – Geometria do escoamento entre placas planas com malha 1D em  $y$



controle  $P$  são,  $A_n$  e  $A_s$ , para as faces à norte e sul, respectivamente (Versteeg & Malalasekera, 2007). A dedução das equações discretas pelo MVF tem as seguintes etapas: integração da equação para cada volume de controle, escolha de funções de interpolação para aproximar os valores das variáveis nas faces a partir dos valores dos volumes vizinhos e construção do sistema algébrico com uma equação por volume de controle. Este processo é descrito a seguir para o caso estudado.

Figura 3 – Malha 1D em  $y$  com 5 volumes de controle



#### 4.1 Equação discreta para volumes internos

A primeira etapa consiste na integração da equação de conservação no volume de controle e a transformação da integral em volume para integral em área, usando o Teorema de Gauss,

$$\int_{V_c} \nabla \cdot (\nabla U_x) dV = \sum_{A_f} \int_{A_f} (\nabla U_x \cdot \mathbf{n}) dA = 0. \quad (5)$$

O escoamento entre placas planas é um problema 1D em  $y$ , ou seja, temos apenas as faces norte e sul em cada volume de controle, e a Equação (5) fica

$$\int_{A_n} (\nabla U_x \cdot \mathbf{n}) dA + \int_{A_s} (\nabla U_x \cdot \mathbf{n}) dA = 0. \quad (6)$$

Em seguida, o resultado da integral do fluxo difusivo  $(\nabla U_x \cdot \mathbf{n})$  é substituído por um valor aproximado

$$(\nabla U_x \cdot \mathbf{n})_n A_n + (\nabla U_x \cdot \mathbf{n})_s A_s = 0 \quad (7)$$

onde o subscrito  $n$  é a aproximação na face norte e o subscrito  $s$  é a aproximação na face sul. Uma função de interpolação será escolhida para calcular o valor do fluxo aproximado. Note que a operação tensorial em  $U_x$  é

$$\nabla U_x \cdot \mathbf{n} = \frac{dU_x}{dx} n_x + \frac{dU_x}{dy} n_y + \frac{dU_x}{dz} n_z \quad (8)$$

onde a normal para cada face é dada por:

$$\mathbf{n}_n = (0, 1, 0) ; \quad \mathbf{n}_s = (0, -1, 0). \quad (9)$$

Assim, sabendo que a malha para o problema é uniforme e  $A_n = A_s = A$ , a Equação (7) fica

$$\left( \frac{dU_x}{dy} \right)_n A - \left( \frac{dU_x}{dy} \right)_s A = 0. \quad (10)$$

A etapa seguinte consiste na escolha de funções de interpolação para aproximar as derivadas de  $U_x$  nas faces norte e sul, usando os valores dos volumes vizinhos. A face sul tem como os volumes vizinhos  $S$  e  $P$  e, a face norte, os volumes vizinhos são  $P$  e  $N$ , conforme mostra a Figura 3.

Uma função de interpolação linear é usada para aproximar as derivadas.

Sabendo que  $\Delta y$  é a distância entre os pontos  $N - P$  e  $P - S$ , respectivamente, uma aproximação linear para a derivada na face norte é

$$\left(\frac{dU_x}{dy}\right)_n \cong \frac{(U_x)_N - (U_x)_P}{\Delta y} \quad (11)$$

e na face sul é

$$\left(\frac{dU_x}{dy}\right)_s \cong \frac{(U_x)_P - (U_x)_S}{\Delta y}. \quad (12)$$

A aproximação linear pode ser explicada pela interpolação por diferenças centrais para uma função  $\phi$  qualquer, dada por,

$$\left(\frac{d\phi}{dy}\right)_i \cong \frac{\phi_{i+1} - \phi_{i-1}}{2h} \quad (13)$$

onde  $h$  é a distância entre o ponto  $i$  e  $i + 1$  e  $i - 1$ , respectivamente. Considere a face norte o índice  $i$ , o centro de volume  $P$  é  $i - 1$  e o centro do volume norte  $N$  é  $i + 1$ . Neste caso a distância entre a face norte e os pontos  $P$  e  $N$ , respectivamente, é  $\Delta y/2$  e a aproximação para a componente  $x$  da velocidade fica:

$$\left(\frac{dU_x}{dy}\right)_n \cong \frac{(U_x)_N - (U_x)_P}{2(\Delta y/2)} = \frac{(U_x)_N - (U_x)_P}{\Delta y}. \quad (14)$$

A mesma análise é válida para a face sul.

A próxima etapa consiste em substituir as aproximações das derivadas na equação de conservação,

$$\left(\frac{(U_x)_N - (U_x)_P}{\Delta y}\right) A - \left(\frac{(U_x)_P - (U_x)_S}{\Delta y}\right) A = 0 \quad (15)$$

e rearranjar os termos da seguinte forma

$$\left(\frac{A}{\Delta y}\right) (U_x)_N - \left(\frac{A}{\Delta y}\right) (U_x)_P - \left(\frac{A}{\Delta y}\right) (U_x)_P + \left(\frac{A}{\Delta y}\right) (U_x)_S = 0. \quad (16)$$

A solução de sistemas algébricos é mais adequada para diagonal positiva. Os termos da diagonal serão aqueles dados pelo ponto  $P$ . Assim, a equação discreta é descrita por

$$-\underbrace{\left(\frac{A}{\Delta y}\right)}_{a_S}(U_x)_S + \underbrace{\left(\frac{A}{\Delta y} + \frac{A}{\Delta y}\right)}_{a_P=a_S+a_N}(U_x)_P - \underbrace{\left(\frac{A}{\Delta y}\right)}_{a_N}(U_x)_N = \underbrace{0}_{S_P} \quad (17)$$

onde foi utilizada a notação dos coeficiente  $a_S$ ,  $a_N$ ,  $a_P$  e  $S_P$  (Versteeg & Malalasekera, 2007).

Estas equações são válidas para os volumes internos, onde  $P = 2, 3$  e  $4$  e podem ser descritas por

$$-a_S(U_x)_S + a_P(U_x)_P - a_N(U_x)_N = 0 \quad (18)$$

onde

$$a_S = \frac{A}{\Delta y} ; a_N = \frac{A}{\Delta y} ; S_P = 0 \quad (19)$$

e

$$a_P = a_S + a_N + S_P. \quad (20)$$

As equações para construção do sistema algébrico são:

- $P = 2$  onde  $S = 1$  e  $N = 3$

$$-\frac{A}{\Delta y}(U_x)_1 + \frac{2A}{\Delta y}(U_x)_2 - \frac{A}{\Delta y}(U_x)_3 + 0(U_x)_4 + 0(U_x)_5 = 0 \quad (21)$$

- $P = 3$  onde  $S = 2$  e  $N = 4$

$$0(U_x)_1 - \frac{A}{\Delta y}(U_x)_2 + \frac{2A}{\Delta y}(U_x)_3 - \frac{A}{\Delta y}(U_x)_4 + 0(U_x)_5 = 0 \quad (22)$$

- $P = 4$  onde  $S = 3$  e  $N = 5$

$$0(U_x)_1 + 0(U_x)_2 - \frac{A}{\Delta y}(U_x)_3 + \frac{2A}{\Delta y}(U_x)_4 - \frac{A}{\Delta y}(U_x)_5 = 0. \quad (23)$$

## 4.2 Equação discreta para volumes no contorno

As equações para os volumes no contorno,  $P = 1$  e  $P = 5$ , são deduzidas seguindo as mesmas etapas, mas os coeficientes encontrados são diferentes por causa da condição de contorno. Vamos começar pelo volume de controle  $P = 1$ . Neste caso não temos face sul e sim a face do contorno, onde a condição de contorno determina que  $U_x = 0$  em  $y = 0$ . A face norte é aquela entre os volumes  $1 - 2$ . A

integração da equação neste volume é

$$\left(\frac{dU_x}{dy}\right)_{n=(1-2)} A - \left(\frac{dU_x}{dy}\right)_{y=0} A = 0 \quad (24)$$

onde a face norte é a face entre os volumes 1 – 2 e a face  $y = 0$  é a do contorno.

Uma função de interpolação linear é adotada para aproximar o valor das derivadas nas faces. Na face norte temos os volumes vizinhos 2 e 1 separados por uma distância  $\Delta y$

$$\left(\frac{dU_x}{dy}\right)_{n=(1-2)} \cong \frac{(U_x)_2 - (U_x)_1}{\Delta y}. \quad (25)$$

Já a face do contorno  $y = 0$  não tem volume vizinho à sul, apenas a norte  $P = 1$ . Porém, o valor de  $U_x$  em  $y = 0$  é conhecido e podemos escrever a aproximação linear pela diferença entre o valor no ponto 1 e o valor na face  $y = 0$ , dividido pela distancia entre esses dois pontos, que é  $\Delta y/2$

$$\left(\frac{dU_x}{dy}\right)_{y=0} \cong \frac{(U_x)_1 - (U_x)_{y=0}}{\Delta y/2}. \quad (26)$$

A aproximação da derivada na face do contorno, pode ser explicada a partir da aproximação por diferença avançada. Note que a aproximação para a derivada no ponto  $i$  por diferença avançada entre o ponto  $i$  e  $i + 1$  é dada por

$$\left(\frac{d\phi}{dy}\right)_i \cong \frac{\phi_{i+1} - \phi_i}{h} \quad (27)$$

onde  $h$  é a distância entre os pontos  $i$  e  $i + 1$ . Considerando a face do contorno  $y = 0$  o ponto  $i$  e o ponto no centro do volume  $P = 1, i + 1$ . A distância entre estes pontos é  $\Delta y/2$  e a aproximação para a derivada na face do contorno é tal qual a Equação (26).

Substituindo as aproximações das derivadas na Equação (24), a equação discreta para  $P = 1$ , fica

$$\left(\frac{(U_x)_2 - (U_x)_1}{\Delta y}\right) A - \left(\frac{(U_x)_1 - (U_x)_{y=0}}{\Delta y/2}\right) A = 0 \quad (28)$$

que pode ser reescrita por

$$\left(\frac{A}{\Delta y}\right) (U_x)_2 - \left(\frac{A}{\Delta y}\right) (U_x)_1 - \left(2\frac{A}{\Delta y}\right) (U_x)_1 + \left(2\frac{A}{\Delta y}\right) (U_x)_{y=0} = 0 \quad (29)$$

e para uma equação com diagonal positiva é dada por:

$$\underbrace{\left(\frac{A}{\Delta y} + 2\frac{A}{\Delta y}\right)}_{a_N+S_P}(U_x)_1 - \underbrace{\left(\frac{A}{\Delta y}\right)}_{a_N}(U_x)_2 = \underbrace{\left(2\frac{A}{\Delta y}\right)}_{S_P}(U_x)_{y=0}. \tag{30}$$

Assim, a equação para  $P = 1$  é

$$-a_S(U_x)_S + a_P(U_x)_P - a_N(U_x)_N = S_P(U_x)_{y=0} \tag{31}$$

onde

$$a_S = 0 ; a_N = \frac{A}{\Delta y} ; S_P = 2\frac{A}{\Delta y} \tag{32}$$

e

$$a_P = a_S + a_N + S_P. \tag{33}$$

Sabendo que  $U_x(y = 0) = 0$ , a equação fica:

$$\left(\frac{A}{\Delta y} + 2\frac{A}{\Delta y}\right)(U_x)_1 - \left(\frac{A}{\Delta y}\right)(U_x)_2 + 0(U_x)_3 + 0(U_x)_4 + 0(U_x)_5 = 0. \tag{34}$$

O procedimento agora é repetido para o volume de controle  $P = 5$ , vizinho a face da fronteira  $y = d$ . Neste caso, não temos face norte, e sim a face do contorno, onde a condição de contorno determina que  $U_x = V$  em  $y = d$ . A integração da equação neste volume é

$$\left(\frac{dU_x}{dy}\right)_{y=d} A - \left(\frac{dU_x}{dy}\right)_{s=(4-5)} A = 0 \tag{35}$$

onde a face sul é a face entre os volumes 4 – 5 e a face  $y = d$  é a do contorno.

Substituindo as aproximações lineares das derivadas, onde a derivada na face sul, ou seja, entre os volumes 4 – 5 é

$$\left(\frac{dU_x}{dy}\right)_{s=4-5} \cong \frac{(U_x)_5 - (U_x)_4}{\Delta y} \tag{36}$$

e a derivada para a face do contorno é

$$\left(\frac{dU_x}{dy}\right)_{y=d} \cong \frac{(U_x)_{y=d} - (U_x)_5}{\Delta y/2} \tag{37}$$

a equação para  $P = 5$  fica

$$\left(\frac{(U_x)_{y=d} - (U_x)_5}{\Delta y/2}\right) A - \left(\frac{(U_x)_5 - (U_x)_4}{\Delta y}\right) A = 0 \tag{38}$$



e pode ser rearranjada por

$$\left(2\frac{A}{\Delta y}\right)(U_x)_{y=d} - \left(2\frac{A}{\Delta y}\right)(U_x)_5 - \left(\frac{A}{\Delta y}\right)(U_x)_5 + \left(\frac{A}{\Delta y}\right)(U_x)_4 = 0 \quad (39)$$

e, finalmente, para a diagonal positiva

$$-\underbrace{\left(\frac{A}{\Delta y}\right)}_{a_S}(U_x)_4 + \underbrace{\left(\frac{A}{\Delta y} + 2\frac{A}{\Delta y}\right)}_{a_S+S_P}(U_x)_5 = \underbrace{\left(2\frac{A}{\Delta y}\right)}_{S_P}(U_x)_{y=d}. \quad (40)$$

Assim, a equação para  $P = 5$  pode ser escrita por

$$-a_S(U_x)_S + a_P(U_x)_P - a_N(U_x)_N = S_P(U_x)_{y=d} \quad (41)$$

onde

$$a_S = \frac{A}{\Delta y} ; a_N = 0 ; S_P = 2\frac{A}{\Delta y} \quad (42)$$

e

$$a_P = a_S + a_N + S_P. \quad (43)$$

Sabendo que  $U_x(y = d) = V$ , a equação fica:

$$0(U_x)_1 + 0(U_x)_2 + 0(U_x)_3 - \left(\frac{A}{\Delta y}\right)(U_x)_4 + \left(\frac{A}{\Delta y} + 2\frac{A}{\Delta y}\right)(U_x)_5 = \left(2\frac{A}{\Delta y}\right)V. \quad (44)$$

### 4.3 Solução do sistema algébrico usando o CalcMe

Uma vez deduzida todas as equações discretas, um exemplo com valores numéricos é apresentado aos estudantes. Os estudantes acessam o Moodle da disciplina e o respectivo questionário do problema. A Figura 4(a) mostra como é a interface para os estudantes.

A solução do problema começa pela definição das variáveis: área  $A$ , distância entre as placas  $d$ , número de divisões da malha  $N$ , velocidade da placa superior  $v$  e velocidade da placa inferior  $v_0$ . Os estudantes são orientados à escrever as equações de  $P = 1$  a 5, onde  $A = 1 \text{ m}^2$ ,  $d = 0.01 \text{ m}$  e  $\Delta y = d/5$ ,

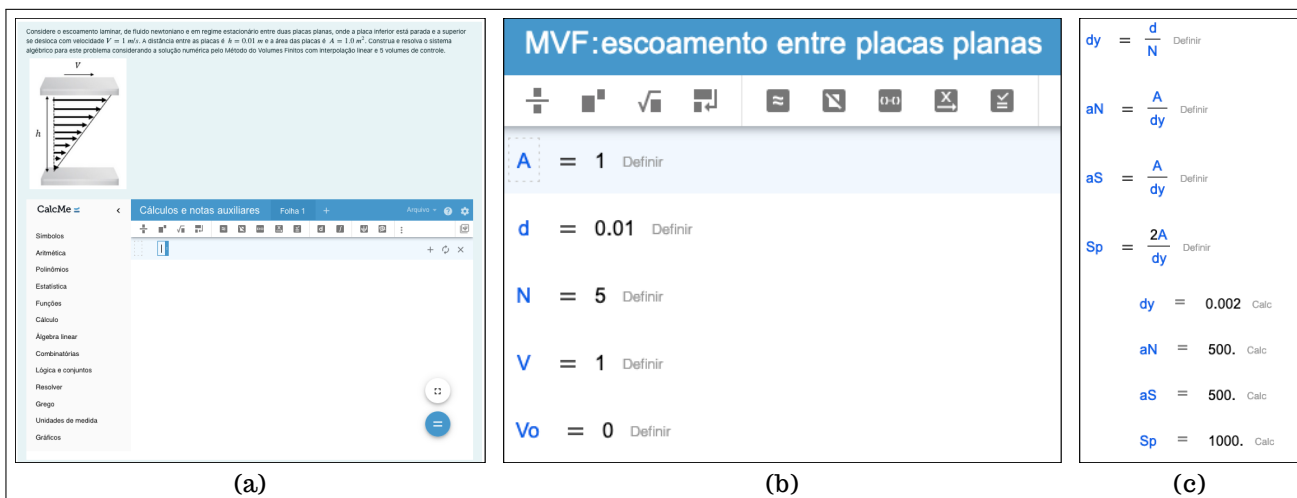
$$\begin{aligned}
 1500(U_x)_1 - 500(U_x)_2 + 0(U_x)_3 + 0(U_x)_4 + 0(U_x)_5 &= 0 \\
 -500(U_x)_1 + 1000(U_x)_2 - 500(U_x)_3 + 0(U_x)_4 + 0(U_x)_5 &= 0 \\
 0(U_x)_1 - 500(U_x)_2 + 1000(U_x)_3 - 500(U_x)_4 + 0(U_x)_5 &= 0 \\
 0(U_x)_1 + 0(U_x)_2 - 500(U_x)_3 + 1000(U_x)_4 - 500(U_x)_5 &= 0 \\
 0(U_x)_1 + 0(U_x)_2 + 0(U_x)_3 - 500(U_x)_4 + 1500(U_x)_5 &= 1000
 \end{aligned} \tag{45}$$

Este conjunto de equações pode ser escrito tal qual o sistema algébrico abaixo, que tem matriz de coeficiente tri-diagonal

$$\begin{pmatrix}
 1500 & -500 & 0 & 0 & 0 \\
 -500 & 1000 & -500 & +0 & 0 \\
 0 & -500 & 1000 & -500 & 0 \\
 0 & 0 & -500 & 1000 & -500 \\
 0 & 0 & 0 & -500 & 1500
 \end{pmatrix} \times \begin{pmatrix} (U_x)_1 \\ (U_x)_2 \\ (U_x)_3 \\ (U_x)_4 \\ (U_x)_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1000 \end{pmatrix} \tag{46}$$

A Figura 4(b) mostra a interface do CalcMe para a definição das variáveis, que deve ser feita na coluna à esquerda da folha de cálculo. A expressão para cálculo dos coeficientes do sistema algébrico é feita em seguida, na Figura 4(c).

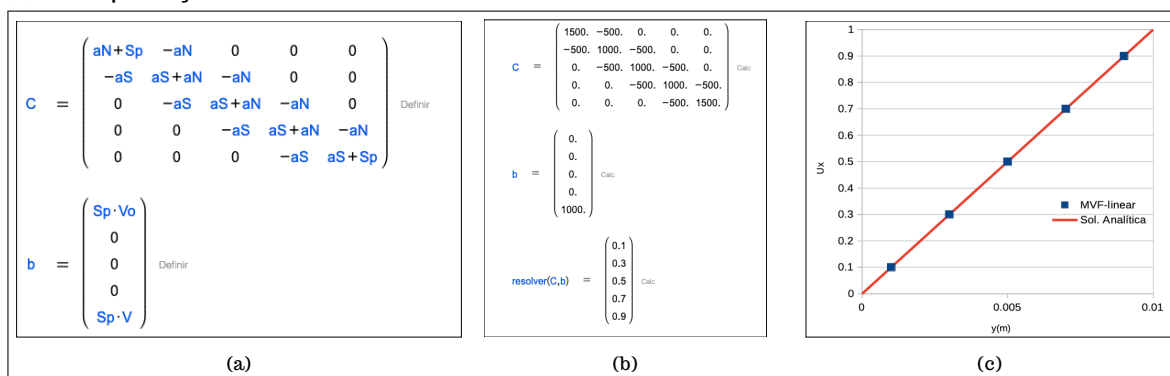
Figura 4 – Interface do CalcMe: (a) no Moodle; (b) definição das variáveis; (c) cálculo dos coeficientes



A matriz dos coeficientes e do termo fonte são definidos, conforme a Figura 5(a). A solução do sistema algébrico é feita usando a função `resolver()` do CalcMe, conforme mostra a Figura 5(b). Por fim, os estudantes constroem um gráfico de comparação entre

a solução analítica e a solução numérica obtida, conforme a Figura 5(c). Note que, como o perfil de velocidade é linear, uma malha de apenas cinco volumes reproduz a solução analítica. A atividade é encerrada com os estudantes, que podem fazer o *download* da memória de cálculo para estudos futuros.

Figura 5 – Etapa de solução numérica do sistema algébrico: (a) definição; (b) solução; (c) comparação



## 5 TUTORIAL DO OPENFOAM®: ESCOAMENTO COUETTE ENTRE PLACAS PLANAS

A seguir, são descritas as etapas para a solução do escoamento entre placas planas usando o OpenFOAM®, aplicativo livre de simulação CFD. O conteúdo segue o formato de um tutorial, onde a estrutura principal são as etapas de uma simulação CFD: construção da geometria e malha, definição das propriedades do fluido, condições de contorno e inicial, configuração da solução numérica, solução do escoamento e análise da convergência numérica. Por fim, a solução numérica é comparada com a solução analítica.

Os estudantes são inicialmente introduzidos ao Linux, principais comandos de terminal e conceitos básicos do OpenFOAM®, como estrutura de diretórios de um caso e paradigma de trabalho sem interface gráfica. A estrutura de um caso consiste nos diretórios e arquivos, também chamados de dicionários, obrigatórios para executar a simulação, que são: diretório `constant` com informações de malha e propriedades do fluido, diretório `system` com os arquivos de configuração da solução numérica e diretório com condição de contorno e inicial.

Neste tutorial, são utilizados os utilitários `blockMesh`, `foamLog` e `postProcess` para as etapas de pré e pós-processamento, e o `solver icoFoam` para a solução do escoamento. O `gnuplot` foi a ferramenta complementar utilizada para gerar gráficos.

## 5.1 Geometria e malha

A geometria e a malha deste tutorial são construídas simultaneamente usando o utilitário `blockMesh`. O dicionário `blockMeshDict`, localizado no diretório `system` do caso, contém as instruções necessárias para criar a geometria e malha. A primeira instrução no `blockMeshDict` determina a unidade de comprimento da geometria em relação ao metro, conforme mostra trecho do arquivo no Dicionário 1. Um caso com dimensões em centímetros ou milímetros, teria o valor unitário substituído por 0.01 ou 0.001, respectivamente.

### Dicionário 1 – Definição do comprimento da geometria no `blockMeshDict`

---

```
convertToMeters 1;
```

---

Em seguida, são declaradas os valores das variáveis adotadas neste tutorial, conforme o trecho do arquivo no Dicionário 2. A distância  $d = 0.1 \text{ m}$  é definida através da variável `dCanal`. As demais distâncias são os comprimentos nas direções  $x$  e  $z$ , respectivamente, definidas como `xCanal` e `zCanal`. A solução numérica para o escoamento entre placas planas é realizada para uma discretização com `nY` divisões em  $y$ . Note que, como o escoamento é plenamente desenvolvido, a solução é para em uma seção  $x$  e  $z$  qualquer da placa plana e esses comprimentos foram arbitrariamente definidos como  $0.4 \text{ m}$  e  $0.1 \text{ m}$ , respectivamente, e possuem uma única divisão.

### Dicionário 2 – Definição das variáveis no `blockMeshDict`

---

```
xCanal 0.4;          //comprimento do canal na direcao x
dCanal 0.1;          //distancia entre as placas na direcao y
zCanal 0.1;          //distancia na direcao z
nY 5;                //numero de divisoes em y
```

---

A próxima etapa consiste em listar as coordenadas de todos os vértices. A Figura 6 mostra a geometria da placa plana e seus respectivos vértices. O Dicionário 3 é o trecho do `blockMeshDict` onde a declaração dos vértices é feita, e as variáveis `xCanal`, `dCanal` e `zCanal` definidas anteriormente usando a notação `$`.

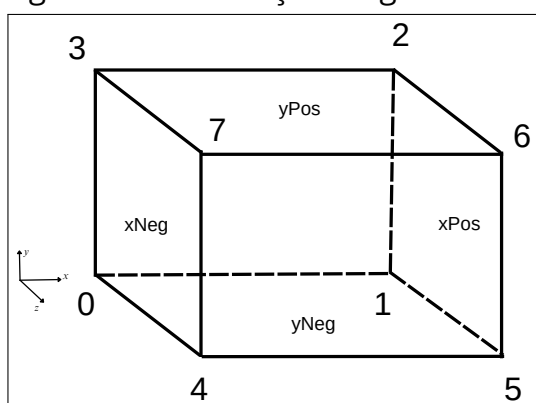
Dicionário 3 – Definição dos vértices no blockMeshDict.

```

vertices
(
    (0          0          0)          //vertice 0
    ($xCanal   0          0)          //vertice 1
    ($xCanal   $hCanal   0)          //vertice 2
    (0          $hCanal   0)          //vertice 3
    (0          0          $zCanal)   //vertice 4
    ($xCanal   0          $zCanal)   //vertice 5
    ($xCanal   $hCanal   $zCanal)   //vertice 6
    (0          $hCanal   $zCanal)   //vertice 7
);

```

Figura 6 – Construção da geometria usando o blockMesh



A construção da malha é dada a partir da definição de blocos dentro da geometria. Neste caso, foi construído um único bloco hexaédrico com uma única divisão nas direções  $x$  e  $z$ , e  $nY$  na direção  $y$ . O Dicionário 4 mostra o trecho do `blockMeshDict` para criar um bloco, onde primeiro é definido o tipo da malha (`hex = hexaédrica`). Depois os vértices que constroem o bloco são declarados, seguidos do número de divisões em cada direção ( $x, y, z$ ). Por fim, é declarado uma opção de refino de malha em cada direção. Não foi adotado refino de malha nas fronteiras e, por esse motivo, os parâmetros do `simpleGrading` são unitários.

Dicionário 4 – Definição de um bloco no blockMeshDict

```

blocks
(
    hex (0 1 2 3 4 5 6 7) (1 $nY 1) simpleGrading (1 1 1)
);

```

A última etapa de construção da malha consiste na declaração das fronteiras do domínio. O Dicionário 5 é o trecho do `blockMeshDict` onde é feita a declaração das fronteiras. A declaração de uma fronteira começa pela escolha de um nome (qualquer)

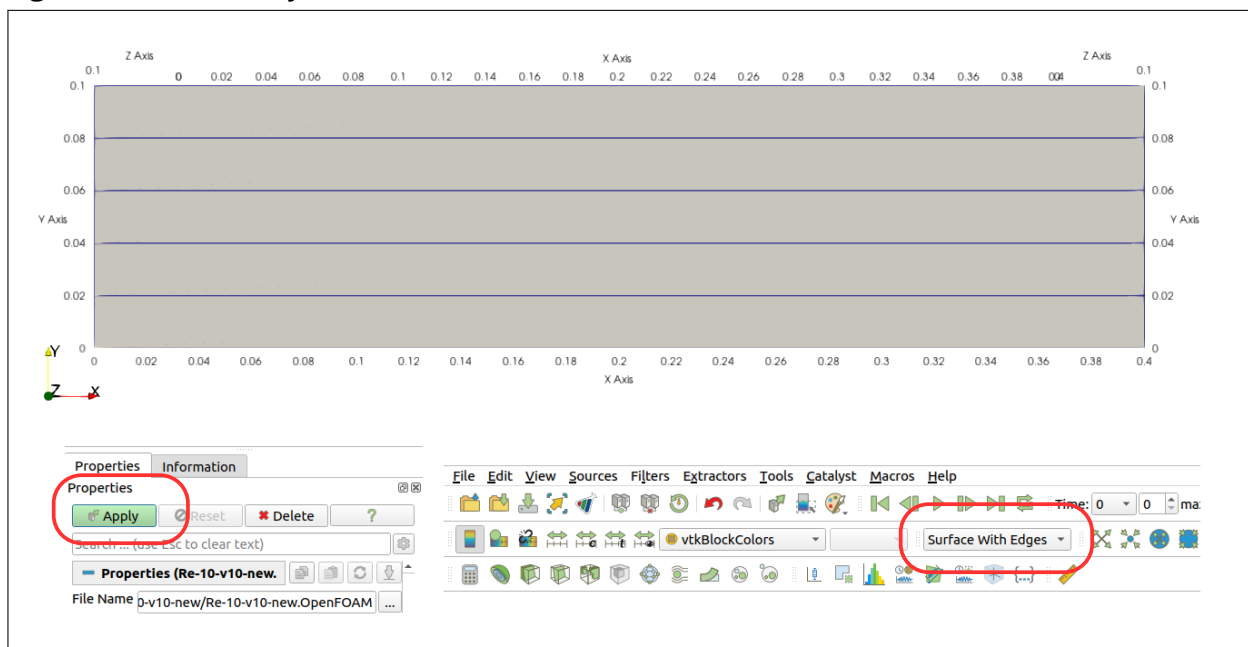
para identificá-la. Este nome será utilizado para definir as condições de contorno para velocidade e pressão. Em seguida utilizaremos a regra da mão direita para determinar os vértices que serão utilizados para construir uma fronteira. A declaração é feita no sub-dicionário `boundary` onde a ordem de cada vértice é tal qual a regra da mão direita, com a normal sempre apontando para fora do volume. Nesta malha, cinco fronteiras são criadas: `yPos`, `yNeg`, `xPos`, `xNeg` e `zPlan`. Em seguida, cada fronteira é identificada através de uma característica topológica `type` e os vértices das faces que a compõem, na ordem consistente com a normal apontando para fora da geometria.

#### Dicionário 5 – Definição das fronteiras no `blockMeshDict`

```
boundary (
  yPos
  {
    type patch;
    faces
    (
      (3 7 6 2)
    );
  }
  yNeg
  {
    type patch;
    faces
    (
      (0 1 5 4)
    );
  }
  xNeg
  {
    type cyclic;
    neighbourPatch xPos;
    faces
    (
      (0 4 7 3)
    );
  }
  xPos
  {
    type cyclic;
    neighbourPatch xNeg;
    faces
    (
      (1 2 6 5)
    );
  }
  zPlan
  {
    type empty;
    faces
    (
      (3 2 1 0)
      (4 5 6 7)
    );
  }
);
```

As faces que compõem as fronteiras são identificadas através dos vértices que a constroem. Assim, através da Figura 6, é possível identificar que a fronteira superior é formada pela face de vértices (3 7 6 2), que recebeu o nome de `yPos`. A escolha da característica topológica da fronteira, ou `type`, depende do tipo de condição de contorno das variáveis. Neste caso, a fronteira superior e inferior (`yPos` e `yNeg`) são todas do tipo `patch`, que é uma característica topológica genérica para cálculos diversos. As fronteiras laterais (`xPos` e `xNeg`) são do tipo `cyclic`, ou seja, os valores das variáveis na direção  $x$  são uniformes. Esta característica topológica é utilizada para reproduzir matematicamente o escoamento desenvolvido em  $x$  e a sua configuração é feita em pares, através da declaração do `neighbourPatch`, conforme mostra o Dicionário 5. Por fim, os planos no eixo  $z$ , são do tipo `empty`. Essa característica topológica existe pois, no OpenFOAM®, todas as geometrias são tridimensionais. Assim, quando um problema precisa ser simplificado para bidimensional, os planos normais ao plano cuja solução é desejada recebe a característica topológica `empty`. Neste caso, busca-se uma solução no plano  $xy$  e, por esse motivo, as fronteiras `zPlan` recebem o `type empty`.

Figura 7 – Visualização da malha usando o ParaView



Fonte: autores (2024)

Uma vez editado o arquivo `blockMeshDict`, a geometria e malha podem ser construídas através da execução do comando `blockMesh` no diretório do caso. Assim, em um terminal no diretório do caso, execute o comando: `blockMesh`. A execução deste comando cria um novo diretório `constant/polyMesh` com os arquivos da malha.

A visualização da geometria e malha são feitas usando o aplicativo ParaView. Assim, em um terminal no diretório do caso, execute o comando `paraFoam` para visualizar a sua geometria e malha. O aplicativo ParaView será carregado e, escolha a opção *Apply* e *Surface with Edges*, para visualizar a malha, conforme a Figura 7.

## 5.2 Propriedades, condições de contorno e inicial

A próxima etapa de pré-processamento consiste na definição das propriedades do fluido, do regime de escoamento (laminar ou turbulento) e das condições de contorno para velocidade e pressão. Este tutorial tem escoamento laminar com número de Reynolds igual a 10.

As propriedades do fluido são especificadas no dicionário `physicalProperties` no diretório `constant`. É necessário especificar a viscosidade cinemática ( $\nu = \text{nu}$ ), conforme mostra o trecho do arquivo no Dicionário 6. Observe que a propriedade viscosidade cinemática é declarada através de uma lista de expoentes de unidades (`dimensionSet`), `[0 2 -1 0 0 0 0]`, e um escalar `0.01`. Valor `0.01` é aquele calculado para Reynolds 10 ( $Re = dV/\nu$ ), onde a velocidade é  $V = 1 \text{ m/s}$  e a distância entre as placas é  $0.1 \text{ m}$ . A lista de expoentes são para as unidades primárias, na seguinte ordem: massa, comprimento, tempo, temperatura, quantidade de matéria, corrente e intensidade luminosa. Assim, a viscosidade cinemática tem unidade em  $\text{m}^2/\text{s}$ .

### Dicionário 6 – Definição das propriedades do fluido no `physicalProperties`

<code>nu</code>	<code>[0 2 -1 0 0 0 0]</code>	<code>0.01;</code>
-----------------	-------------------------------	--------------------

A condição de contorno e inicial dos campos de velocidade ( $U$ ) e pressão ( $p$ ) são especificadas em um arquivo para cada variável no diretório `0`, que é o instante inicial do caso. O Dicionário 7 apresenta a especificação da condição de contorno e inicial do campo de velocidade. A primeira definição do arquivo é a lista `dimensionSet` que representa a unidade da grandeza ( $\text{m/s}$ ), em seguida o valor da velocidade no campo interno, ou seja, nos centros dos volumes de controle do domínio computacional (`internalField`). Neste caso, o campo interno é inicializado como uniforme e nulo. As



condições de contorno são especificadas para cada fronteira a partir do nome escolhido na etapa de construção de malha. Recorde que as fronteiras criadas neste caso foram: *yPos*, *yNeg*, *xPos*, *xNeg* e *zPlan* (Dicionário 5).

A condição de contorno para velocidade na fronteira *yPos* e *yNeg* é do tipo valor fixo, declarada como *fixedValue*. A fronteira *yPos* é aquela que tem velocidade na direção *x* igual a  $1\text{ m/s}$ . Já na fronteira *yNeg* a velocidade é igual a zero. Os planos *xPos* e *xNeg* terão os valores uniformes em *x*, portanto, a condição de contorno é do tipo *cyclic*. Por fim, a fronteira *zPlan* é do tipo *empty* para que a solução do escoamento seja apenas no plano *xy*.

#### Dicionário 7 – Arquivo de condição de contorno/inicial para velocidade no diretório 0

```

dimensions      [0 1 -1 0 0 0 0];
internalField   uniform (0 0 0);
boundaryField
{
    yPos
    {
        type      fixedValue;
        value     uniform (1 0 0);
    }
    yNeg
    {
        type      fixedValue;
        value     uniform (0 0 0);
    }
    xNeg
    {
        type      cyclic;
    }
    xPos
    {
        type      cyclic;
    }
    zPlan
    {
        type      empty;
    }
}

```

O Dicionário 8 apresenta a especificação da condição de contorno e inicial do campo de pressão. Note que o campo de pressão especificado na solução do escoamento incompressível no OpenFOAM® é um campo de pressão relativa (manométrica) dividido pela massa específica do fluido, ou seja, a pressão é na verdade  $p/\rho$ . Por este motivo, a unidade do campo de pressão é  $m^2/s^2$ . A condição de contorno adotada para a pressão nas fronteiras *yPos*, *yNeg* é do tipo *zeroGradient*, onde o valor da fronteira é igual ao valor do volume vizinho. A condição de contorno dos planos *xPos*, *xNeg* são do tipo *cyclic*, que repetem os valores na direção *x* para

reproduzir o escoamento desenvolvido. Por fim, a fronteira `zPlan` é do tipo `empty` para que a solução do escoamento seja apenas no plano  $xy$ .

Dicionário 8 – Arquivo de condição de contorno/inicial para pressão relativa no diretório 0

---

```

dimensions      [0 2 -2 0 0 0 0];
internalField   uniform 0;
boundaryField
{
    yPos
    {
        type      zeroGradient;
    }
    yNeg
    {
        type      zeroGradient;
    }
    xNeg
    {
        type      cyclic;
    }
    xPos
    {
        type      cyclic;
    }
    zPlan
    {
        type      empty;
    }
}

```

---

### 5.3 Configurações da solução numérica

As configurações da solução numérica são as escolhas dos esquemas de interpolação, métodos de solução dos sistemas algébricos, tolerâncias, número de iterações no acoplamento pressão-velocidade, tempo inicial, final e passo de tempo da simulação. Estas definições são feitas nos dicionários no diretório `system` do caso.

Os esquemas de interpolação utilizados neste tutorial foram: método implícito de Euler na integração temporal e linear nas demais discretizações. O Dicionário 9 mostra as declarações no arquivo `fvSchemes`.

## Dicionário 9 – Definição dos esquemas de interpolação no fvSchemes no diretório

system

---

```

ddtSchemes
{
    default          Euler;
}
gradSchemes
{
    default          Gauss linear;
}
divSchemes
{
    default          none;
    div(phi,U)      Gauss linear;
}
laplacianSchemes
{
    default          Gauss linear corrected;
}

```

---

Os métodos para a solução dos sistemas algébricos foram: Gauss-Seidel e Gradiente Conjugado para a velocidade e pressão, respectivamente. Um pré-condicionador diagonal foi configurado na solução da pressão. As tolerâncias de  $10^{-6}$  e  $10^{-7}$  foram configuradas para a velocidade e pressão, respectivamente. Note que, apesar do campo de pressão ser uniforme, as configurações para pressão precisam estar presentes, inclusive o sub-dicionário PISO, que controla o acoplamento pressão-velocidade, e a declaração do valor da pressão relativa de referência, que é nula. O Dicionário 10 mostra o arquivo `fvSolutions` com as configurações citadas.

## Dicionário 10 – Definição dos parâmetros solução no fvSolution no diretório system

---

```

solvers
{
    U
    {
        solver          smoothSolver;
        smoother        symGaussSeidel;
        tolerance       1e-06;
        relTol          0;
    }
    pFinal
    {
        solver          PCG;
        preconditioner  DIC;
        tolerance       1e-07;
        relTol          0;
    }
}
PISO
{
    pRefCell          0;
    pRefValue         0;
}

```

---

Por fim, os parâmetros de controle da simulação, como tempo inicial, final, passo de tempo e instantes de tempo salvos, são configurados no dicionário `controlDict`, conforme trecho do arquivo no Dicionário 11. O passo de tempo (`deltaT`), configurado para o caso foi calculado considerando o critério de Courant unitário para o caso.

#### Dicionário 11 – Definição dos parâmetros solução no `controlDict` no diretório `system`

```
application      icoFoam;
startFrom        startTime;
startTime        0;
stopAt           endTime;
endTime          1;
deltaT           0.02;
writeControl     adjustableRunTime;
writeInterval    0.1;
purgeWrite       0;
writeFormat      ascii;
writePrecision   6;
writeCompression off;
timeFormat       general;
timePrecision    6;
```

## 5.4 Solução do escoamento

A solução do escoamento é dada pela execução do *solver* escolhido, que no caso foi o `icoFoam`. Este é o *solver* padrão do OpenFOAM® para a solução transiente de escoamento laminar de fluido Newtoniano. A execução do *solver* irá montar o sistema algébrico da solução numérica das equações que governam o escoamento, considerando os esquemas de interpolação escolhidos no `fvSchemes` e a malha do caso. Assim, em um terminal no diretório do caso execute o comando: `icoFoam > log`.

Este comando irá executar o *solver* do OpenFOAM® e armazenar os dados de saída no terminal no arquivo `log`. Os diretórios para a solução transiente serão criados e os respectivos campos de velocidade e pressão salvos, até a simulação atingir o seu tempo final configurado no `controlDict`.

## 5.5 Convergência Numérica

Uma vez que a execução do comando no terminal está concluída, a próxima etapa consiste em avaliar a convergência da solução numérica. A análise da convergência numérica consiste em acompanhar os resíduos das soluções dos sistemas algébricos e uma variável de interesse. A convergência numérica depende das características do escoamento e, neste caso, o escoamento atinge o regime estacionário. Neste tutorial,

apenas o resíduo para  $U_x$  é relevante, uma vez que a pressão é uniforme e as demais componentes da velocidade são nulas.

Os valores dos resíduos iniciais e finais para solução de cada sistema algébrico é resultado da execução do comando `icoFoam`, que foi armazenado no arquivo `log`. O Dicionário 12 abaixo mostra a trecho simplificado do arquivo `log` no primeiro passo de tempo e no instante de tempo final. Note que, no primeiro instante de tempo, o resíduo inicial (`Initial residual`) para  $U_x$  tem valor unitário, o resíduo final (`Final residual`) tem ordem de grandeza semelhante a tolerância para  $U$  especificada no `fvSolution` e 6 iterações foram necessárias para esta convergência. Já no último instante de tempo, o resíduo inicial é da ordem de grandeza de  $10^{-5}$ . Pode-se afirmar, que esta simulação tem uma convergência numérica da ordem de  $10^{-5}$  no último instante de tempo.

#### Dicionário 12 – Resultado da solução do escoamento usando o `icoFoam`

---

```
Starting time loop
Time = 0.02s

Courant Number mean: 0 max: 0
smoothSolver: Solving for Ux, Initial residual = 1, Final
  residual = 2.83e-07, No Iterations 6
...
Time = 1s

Courant Number mean: 0.024996 max: 0.0449981
smoothSolver: Solving for Ux, Initial residual = 1.74885e-05,
  Final residual = 1.32559e-07, No Iterations 2
...
End
```

---

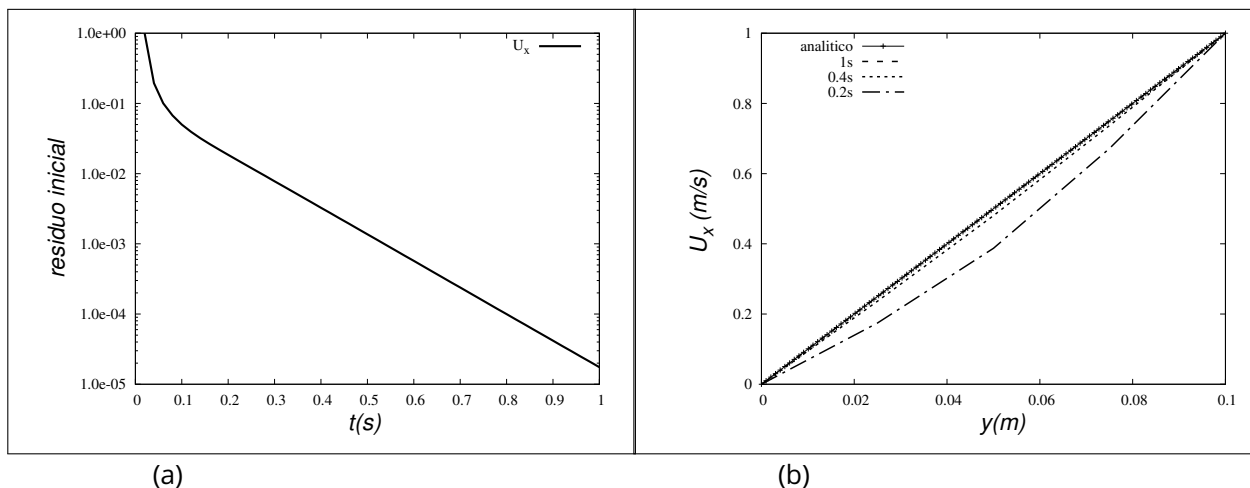
Apesar do valor do resíduo inicial no último instante de tempo já ser um parâmetro válido para análise da convergência numérica e regime estacionário, um gráfico do resíduo inicial ao longo do tempo é um importante resultado para análise da convergência numérica. O OpenFOAM® possui um utilitário para capturar as informações de resíduos do arquivo `log`, o `foamLog`. Assim, em um terminal, execute o comando: `foamLog > log`.

O utilitário irá gerar um novo diretório `logs`, contendo os arquivos necessários para construir um gráfico do resíduo inicial de  $U_x$  em relação ao tempo. Foi utilizado o aplicativo `gnuplot` para construir o gráfico.

A Figura 8(a) mostra a análise da convergência numérica através do resíduo inicial de  $U_x$  ao longo da simulação. Note que, no início da simulação os valores dos resíduos são altos e, ao longo do tempo, reduzem até a ordem de grandeza de  $10^{-5}$ . Como o valor especificado para a tolerância na solução da velocidade foi de  $10^{-6}$ , a

convergência do resíduo inicial para  $10^{-5}$  indica que o escoamento atingiu o estado estacionário. A análise de uma variável de interesse ao longo do tempo, a fim de confirmar a convergência para o regime estacionário e comparação com a solução analítica é feita a seguir.

Figura 8 – Resultados da simulação



Legenda: (a) Resíduo inicial ao longo da simulação e (b) comparação do perfil de velocidade

## 5.6 Análise dos resultados

Uma vez verificada a convergência numérica da solução, a próxima etapa consiste em avaliar a variável de interesse no campo. Neste tutorial, a variável de interesse é  $U_x$  ao longo da distância  $y$ . O utilitário `postProcess` do OpenFOAM® será utilizado para capturar os valores de  $U_x$  ao longo de  $y$ . Um arquivo adicional, chamado `sampleU` e localizado no diretório `system`, é necessário para especificar o local da geometria onde os valores de  $U_x$  devem ser capturados. O Dicionário 13 mostra o trecho do dicionário `sampleU`. Note que uma linha chamada de `linhA`, do ponto `start` ao ponto `end`, captura os valores de  $U$  e da coordenada  $y$  ao longo de 5 pontos (`nPoints`). Em um terminal no diretório do caso, execute o comando `postProcess -func sampleU` para capturar os valores da velocidade.

## Dicionário 13 – Dicionário sampleU no diretório system

```

type sets;
libs ("libsampling.so");
interpolationScheme cellPointFace;
setFormat raw;
sets
(
    linhaA
    {
        type          lineUniform;
        nPoints      5;
        axis          y;
        start         (0.2  0    0.05);
        end           (0.2  0.1  0.05);
    }
);
fields
(
    U
);

```

Um vez concluída a execução do comando, um novo diretório `postProcess` será criado no diretório do caso. Dentro deste diretório `postProcess`, os resultados do dicionário `sampleU` são armazenados em um diretório de mesmo nome, para cada instante de tempo. O `gnuplot` foi utilizado novamente para construir o gráfico de  $U_x$  em função de  $y$  para diferentes instantes de tempo e comparar com a solução analítica, conforme mostra a Figura 8(b). Observe que, no instante de tempo  $0.2\text{ s}$  o perfil de velocidade ainda está bem distante da solução no estado estacionário. Já no instante de tempo  $1\text{ s}$ , não é possível fazer distinção visual entre o perfil de velocidade simulado e o analítico.

## 6 CONSIDERAÇÕES FINAIS

Foi apresentado conteúdo didático para o ensino do método dos volumes finitos e das etapas de uma simulação CFD usando o escoamento Couette entre placas planas como caso de referência. Foram utilizadas ferramentas computacionais livres no presente trabalho. Espera-se que esta publicação contribua para atenuar a acentuada curva de aprendizado observada no ensino de Dinâmica dos Fluidos Computacional e, em particular, no uso do OpenFOAM®.

## REFERÊNCIAS

Adair, D. & Jaeger, M. (2019). An efficient strategy to deliver understanding of both numerical and practical aspects when using navier-stokes equations to solve fluid mechanics problems. *Fluids*, 4(4).

- Barba, L. A. & Forsyth, G. F. (2018). CFD Python: the 12 steps to Navier-Stokes equations. *Journal of Open Source Education*, 1(9):21.
- Barba, L. A. & Mesnard, O. (2019). Aero Python: classical aerodynamics of potential flow using Python. *Journal of Open Source Education*, 2(15):45.
- Battista, N. A. (2020). Suite-CFD: An array of fluid solvers written in matlab and python. *Fluids*, 5(1).
- Clebsch, A. B. & Mors, P. M. (2004). Explorando recursos simples de informática e audiovisuais: uma experiência no ensino de fluidos. *Rev. Bras. Ens. Fis.*, 26:323–333.
- Dales, R. & Willis, L. (2010). Application of Computational Fluid Dynamics (CFD) through project-based learning. *Higher Education Academy Engineering Subject Centre, Loughborough University*.
- Dias Fernandes, T. F. D. & Moreira, N. L. (2019). Uma interface de controle para a Fluidodinâmica Computacional. *Rev. Bras. Ens. Fis.*, 41(4):1–11.
- Fortuna, A. O. (2020). *Técnicas Computacionais para Dinâmica dos Fluidos*. LTC.
- Hoist, T. L. (1994). Computational Fluid Dynamics Uses in Fluid Dynamics / Aerodynamics Education. Technical report, National Aeronautics and Space Administration.
- Hu, J., Zhang, L., & Xiong, X. (2008). Teaching Computational Fluid Dynamics (CFD) to design engineers. In *ASEE Annual Conference and Exposition*. American Society for Engineering Education.
- LaRoche, R. D., Hutchings, B. J., & Muralikrishnan, R. (2002). Flowlab: Computational fluid dynamics (CFD) framework for undergraduate education. In *ASEE Annual Conference Proceedings*, pages 1235 – 1238. American Society for Engineering Education.
- Maliska, C. R. (2017). *Transferência de Calor e Mecânica dos Fluidos Computacional*. LTC.
- Patankar, S. (1980). *Numerical Heat Transfer and Fluid Flow*. CRC Press.
- Pawar, S. & San, O. (2019). CFD Julia: A learning module structuring an introductory course on computational fluid dynamics. *Fluids*, 4(3).



- Polanco, G., García, N., & Rojas, L. (2009). Teaching how to use the CFD approach by an example: Hydrodynamics within a passenger car compartment in motion. In *Proceedings of FEDSM2009 ASME 2009 Fluids Engineering Division Summer Meeting*, pages 251–257. ASME.
- Rocha, P. A. C. & da Silveira, J. V. P. (2012). Estudo e aplicação de simulação computacional em problemas simples de mecânica dos fluidos e transferência de calor. *Rev. Bras. Ens. Fis.*, 34.
- Rodríguez-Martín, M., Rodríguez-González, P., Sánchez-Patrocínio, A., & Sánchez, J. R. (2019). Short CFD simulation activities in the context of fluid-mechanical learning in a multidisciplinary student body. *Applied Sciences*, 9:1–17.
- Sert, C. & Nakiboglu, G. (2007). Use of computational fluid dynamics (CFD) in teaching fluid mechanics. In *ASEE Annual Conference and Exposition*. American Society for Engineering Education.
- Solmaz, S. & Van Gerven, T. (2022). Interactive CFD simulations with virtual reality to support learning in mixing. *Computers and Chemical Engineering*, 156:107570.
- Souza, P. V. S. & de Oliveira, P. M. C. (2016). Túneis de vento numéricos. *Rev. Bras. Ens. Fis.*, 38.
- Stern, F., Xing, T., Yarbrough, D. B., Rothmayer, A., Rajagopala, G., Otta, S. P., Caughey, D., Bhaskaran, R., Smith, S., Hutchings, B., & Moeykens, S. (2006). Hands-On CFD Educational Interface for Engineering Courses and Laboratories. *Journal of Engineering Education*, (January):63–83.
- Vasata, D., Galante, G., Rizzi, R. L., & Zara, R. A. (2011). Solução computacional do problema da cavidade cúbica através das equações de Navier-Stokes tridimensionais. *Rev. Bras. Ens. Fis.*, 33(2).
- Versteeg, H. K. & Malalasekera, W. (2007). *Introduction to Computational Fluid Dynamics - THE FINITE VOLUME METHOD*. Pearson Education Limited.

Vieira, C. A. & Pires, M. (2022). Aerodinâmica de autos para o ensino médio na abordagem dos três momentos pedagógicos. *Rev. Bras. Ens. Fis.*, 44.

Wang, Z. & Gao, D. (2017). The Application of CFD in Teaching of Fluid Machinery. In *Proceedings of the 3rd International Conference on Arts, Design and Contemporary Education (ICADCE)*. Atlantis Press.

Wehmann, C. F., Rocha, P. A. C., da Silva, M. E. V., Araújo, F. A. A., & Correia, D. L. (2018). Estudo e aplicação de simulação computacional em problemas simples de mecânica dos fluidos e transferência de calor - Parte II: Problemas clássicos de transmissão de calor. *Rev. Bras. Ens. Fis.*, 40(2):1-9.

Werlang, R. B., de Souza Schneider, R., & da Silveira, F. L. (2008). Uma experiência de ensino de física de fluidos com o uso de novas tecnologias no contexto de uma escola técnica. *Rev. Bras. Ens. Fis.*, 30:1-9.

Young, D. F., Munson, B. R., & Okiishi, T. H. (2011). *A Brief Introduction to Fluid Mechanics*. John Wiley & Sons.

## **Contribuições dos autores**

### **1 – Livia Flavia Carletti Jatobá (Autor Correspondente)**

Doutora em Engenharia Química, Professora Adjunta

<https://orcid.org/0000-0003-1653-8289> • [liviajatoba@iprj.uerj.br](mailto:liviajatoba@iprj.uerj.br)

Contribuição: Conceituação, Análise Formal, Obtenção de financiamento, Metodologia, Administração do projeto, Software, Supervisão, Validação, Visualização, Escrita – primeira redação

### **2 – Sthefany Machado Sardinha**

Estudante do curso de Engenharia Mecânica, Bolsista

<https://orcid.org/0009-0003-2047-0528> • [sthefanymachados1@gmail.com](mailto:sthefanymachados1@gmail.com)

Contribuição: Software, Validação, Visualização, Escrita – tutorial no OpenFOAM

## Como citar este artigo

Jatobá, L. F. C., & Sardinha, S. M. (2024). Escoamento Couette entre placas paralelas: estudo de caso para o ensino de dinâmica de fluidos computacional. *Ciência e Natura*, Santa Maria, v. 46, e87078. DOI: <https://doi.org/10.5902/2179460X87078>.