



**Universidade Federal
do Espírito Santo**

**PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA MECÂNICA
CENTRO TECNOLÓGICO
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO**

CAIO PATRICK PICOLI DE LIMA

**ABORDAGEM MISTA DE VOLUMES FINITOS E ELEMENTOS FINITOS PARA
OTIMIZAÇÃO TOPOLÓGICA EM ESCOAMENTOS TURBULENTOS**

Vitória , ES

2025



**Universidade Federal
do Espírito Santo**

**PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA MECÂNICA
CENTRO TECNOLÓGICO
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO**

CAIO PATRICK PICOLI DE LIMA

**ABORDAGEM MISTA DE VOLUMES FINITOS E ELEMENTOS FINITOS PARA
OTIMIZAÇÃO TOPOLÓGICA EM ESCOAMENTOS TURBULENTOS**

Dissertação apresentada ao Programa de Pós Graduação em Engenharia Mecânica da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Engenharia Mecânica.

Orientador(a): Juan Sergio R. Saenz

Vitória , ES

2025

Ficha catalográfica disponibilizada pelo Sistema Integrado de
Bibliotecas - SIBI/UFES e elaborada pelo autor

D278a de Lima, Caio Patrick Picoli, 1998-
Abordagem mista de Volumes Finitos e Elementos Finitos
para Otimização Topológica em Escoamentos Turbulentos / Caio
Patrick Picoli de Lima. - 2025.
69 f.

Orientador: Juan Sergio Romero Saenz.
Dissertação (Mestrado em Engenharia Mecânica) -
Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Fluidodinâmica Computacional. 2. Turbulência. 3.
Método dos Volumes Finitos. 4. Método dos Elementos
Finitos. 5. Método de Otimização Topológica. I. Saenz, Juan
Sergio Romero. II. Universidade Federal do Espírito Santo.
Centro Tecnológico. III. Título.

CDU: 621



PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA
CENTRO TECNOLÓGICO
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

ABORDAGEM MISTA DE VOLUMES FINITOS E ELEMENTOS FINITOS PARA OTIMIZAÇÃO TOPOLÓGICA EM ESCOAMENTOS TURBULENTOS

CAIO PATRICK PICOLI DE LIMA

COMISSÃO EXAMINADORA

Prof. Dr. Juan Sergio Romero Saenz
(Orientador – PPGEM/UFES)

Prof. Dr. Ramon Silva Martins
(Examinador Interno – PPGEM/UFES)

Prof. Dr. Marcus Vinicius Canhoto Alves
(Examinador Externo – PPGEM/UDESC)

Prof. Dr. Hermes Vazzoler Junior
(Examinador Externo – IFES)



Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Mecânica da Universidade Federal do Espírito Santo como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Mecânica.

Vitória/ES, 21 de fevereiro de 2025.





Aprovação banca Caio Patrick Picoli de Lima

Data e Hora de Criação: 21/02/2025 às 20:54:37

Documentos que originaram esse envelope:

- Aprovação banca Caio Patrick Picoli de Lima.pdf (Arquivo PDF) - 2 página(s)



Hashs únicas referente à esse envelope de documentos

[SHA256]: 397f116595399a2a8c1f05abf18dea33e12c5925d03e14c2021319268f019641

[SHA512]: 522b0149cb9555277b5198948cd0557e24fba637fb29f305030697e26fe39756083036a4ac0d869397142abb052607d7451962a5f18654f1979b98ce591b604e

Lista de assinaturas solicitadas e associadas à esse envelope



ASSINADO - Juan Sergio Romero Saenz (juan.saenz@ufes.br)

Data/Hora: 21/02/2025 - 20:55:20, IP: 177.104.235.39, Geolocalização: [-20.274327, -40.295450]

[SHA256]: 3b6300224843d736b80d8abd87f51c76a69c908d775f2e2a6669a0889f31e7e6



ASSINADO - Ramon Silva Martins (ramon.martins@ufes.br)

Data/Hora: 21/02/2025 - 20:57:02, IP: 131.255.22.234

[SHA256]: 4f4e38f36e9e60392b3988dce50d2da7b8371e07669a6e5c74981435d8f3212f



ASSINADO - Marcus Vinicius Canhoto Alves (marcus.alves@udesc.br)

Data/Hora: 25/02/2025 - 08:51:58, IP: 200.19.106.209

[SHA256]: c8965bd8c0aefa888c937fde88beb4d63c568db2987e8d8108fc1e44d611329e



ASSINADO - Hermes Vazzoler Junior (hermes@ifes.edu.br)

Data/Hora: 25/02/2025 - 19:27:02, IP: 179.217.27.164, Geolocalização: [-20.322713, -40.278425]

[SHA256]: bd1fa1c496ce316504d51b354b331cbd38a03f9935c2d0c0b1c1c09174483310

Histórico de eventos registrados neste envelope

25/02/2025 19:27:02 - Envelope finalizado por hermes@ifes.edu.br, IP 179.217.27.164

25/02/2025 19:27:02 - Assinatura realizada por hermes@ifes.edu.br, IP 179.217.27.164

25/02/2025 19:26:52 - Envelope visualizado por hermes@ifes.edu.br, IP 179.217.27.164

25/02/2025 08:51:58 - Assinatura realizada por marcus.alves@udesc.br, IP 200.19.106.209

25/02/2025 08:51:54 - Envelope visualizado por marcus.alves@udesc.br, IP 200.19.106.209

21/02/2025 20:57:02 - Assinatura realizada por ramon.martins@ufes.br, IP 131.255.22.234

21/02/2025 20:55:20 - Assinatura realizada por juan.saenz@ufes.br, IP 177.104.235.39

21/02/2025 20:54:57 - Envelope registrado na Blockchain por juan.saenz@ufes.br, IP 177.104.235.39

21/02/2025 20:54:57 - Envelope encaminhado para assinaturas por juan.saenz@ufes.br, IP 177.104.235.39

21/02/2025 20:54:38 - Envelope criado por juan.saenz@ufes.br, IP 177.104.235.39

Dedico este trabalho a todos (especialmente professores) que me apoiaram, me suportaram, me motivaram, e, mais importante, nunca me deixaram desistir.

AGRADECIMENTOS

Primeiro, agradeço a Deus por me permitir terminar este trabalho.

Agradeço a minha mãe, Dona Rose, que trabalhou muito duro por toda a sua vida para que eu me tornasse o homem que sou hoje e serei eternamente grato pelo seu apoio financeiro e emocional.

Meus mais sinceros agradecimentos ao meu orientador, Juan, que acreditou em mim e me deu a oportunidade de estar nesse programa de pós graduação e poder fazer minha contribuição com a ciência para o bem da humanidade.

Sou grato a todos os meus amigos, especialmente ao Lucas e ao Marcelo, também orientados de Juan, ao meu irmão Diogo e aos meus familiares pelos momentos felizes entre aulas e pela confiança em mim quando nem eu mesmo acreditei (e também pelas noites em claro fazendo atividades do mestrado).

Agradeço à minha instituição, a Universidade Federal do Espírito Santo (UFES), pelas condições materiais ofertadas e à instituição Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro para a realização deste trabalho. A bolsa por eles ofertada foi essencial para a sobrevivência de minha família.

E por fim, agradeço também a qualquer um que leia este trabalho, pois foi para isso que o mesmo foi escrito.

*“A necessidade é a mãe da inovação.”
(Platão)*

RESUMO

Otimização é um tópico muito presente no imaginário popular. Com respeito à Engenharia, otimização de geometrias é possivelmente o tópico mais difundido entre o público geral, especialmente a respeito de otimização de perfis de asas de avião, e como exemplo deste tópico há o trabalho de Muller (2021), que por otimização reduziu o valor do arrasto quase pela metade. Outro trabalho interessante foi feito por Zheng *et al.* (2021), na otimização do sistema de refrigeração de placas fotovoltaicas a fim de maximizar a eficiência das placas. Dentre os vários métodos de otimização, o que será utilizado neste trabalho será o Método de Otimização Topológica, visto que ele não depende de um conhecimento prévio da geometria ótima. Originalmente, tal método foi desenvolvido a partir do Método dos Elementos Finitos, que apresenta limitações na solução de escoamentos. Tais limitações são superadas ao utilizar o Método dos Volumes Finitos, embora este apresente limitações que impedem seu uso para o Método de Otimização Topológica. O objetivo do presente trabalho é o desenvolvimento da ferramenta criada por Lima, Aguiar e Romero (2024), que utiliza o Método dos Volumes Finitos e o Método dos Elementos Finitos em conjunto para a Otimização Topológica, a fim de que ela otimize escoamentos turbulentos. A solução do escoamento foi realizada através do software OpenFOAM®, a otimização da geometria mediante rotinas personalizadas executadas no Octave e a automatização desse processo foi feita por meio de *Bash Scripting*. Uma vez realizados os testes de otimização de um bocal, de uma curva T e de um joelho em 90°, análises qualitativas com a literatura disponível foram conduzidas. Os resultados obtidos foram qualitativamente idênticos às geometrias de referência com malhas muito menores, o que mostra que a ferramenta é funcional, embora ainda exista espaço para melhorias. Além disso, as geometrias resultantes da otimização dos escoamentos turbulentos apresentaram uma perda de carga menor quando comparados com às geometrias geradas na otimização de escoamentos laminares.

Palavras-chave: Fluidodinâmica Computacional; Turbulência; Método dos Elementos Finitos; Método dos Volumes Finitos; Método de Otimização Topológica.

ABSTRACT

Optimization is a well-known topic in popular imagination. Regarding engineering, geometry optimization is possibly the most widespread topic to the public, especially optimization of aircraft wings, and, as an example of this topic, there is the work of Muller (2021), that, through optimization, almost halved drag on the wing. Another interesting work was conducted by Zheng *et al.* (2021) in the optimization of the cooling system of a photovoltaic panel, in order to maximize its efficiency. Amongst all optimization methods, the one chosen for this work is the Topological Optimization Method, since it does not depend on a previous knowledge of the optimal geometry. Originally, this method was developed using the Finite Elements Method, which has limitations on flow solution. These limitations are overcome by using the Finite Volumes Method, although it has limitations that prevent its use on the Topological Optimization Method. The objective of this work is the development of the tool created by Lima, Aguiar e Romero (2024), which uses the Finite Volume Method and the Finite Element Method together for Topological Optimization, so that it optimizes turbulent flows. The solution of the flow was done using the software OpenFOAM®, the geometry optimization through custom code executed on Octave, and the automatization of this process was done by *Bash Scripting*. Once the optimization tests of a nozzle, a T bend, and a 90° knee had been conducted, qualitative analyses with the available literature were conducted. The results obtained were qualitatively identical to the reference geometries with smaller meshes, which shows that the tool is functional, although there is still space for improvement. Furthermore, the geometries resulting from the optimization of turbulent flows had a smaller pressure drop when compared to the geometries generated from the optimization of laminar flows.

Keywords: Computational Fluid Dynamics; Turbulence; Finite Elements Method; Finite Volumes Method; Topological Optimization Method.

LISTA DE FIGURAS

Figura 1 – Domínio Fixo Estendido de Projeto.	26
Figura 2 – Influência da penalização na propriedade efetiva do material	27
Figura 3 – Fluxograma esquemático das etapas realizadas no processo de otimização	32
Figura 4 – Geometrias e Condições de Contorno utilizadas.	36
Figura 5 – Resultados da Otimização Topológica de um bocal em regime laminar.	37
Figura 6 – Resultados da Otimização Topológica de um joelho em regime laminar.	38
Figura 7 – Resultados da Otimização Topológica de uma curva T em regime laminar.	39
Figura 8 – Resultados da Otimização Topológica de um bocal em regime turbu- lento.	40
Figura 9 – Resultados da Otimização Topológica de um joelho em regime turbu- lento.	41
Figura 10 – Resultados da Otimização Topológica de uma curva T em regime turbulento.	42

LISTA DE TABELAS

Tabela 1 – Parâmetros físicos dos experimentos computacionais realizados. . .	34
Tabela 2 – Aspectos Computacionais das Simulações	35

SUMÁRIO

1	INTRODUÇÃO	14
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Equações Governantes	16
2.2	Método dos Volumes Finitos	21
2.3	Método dos Elementos Finitos	22
2.4	Método de Otimização Topológica	25
2.4.1	Domínio Estendido de Projeto	25
2.4.2	Modelo de Material	26
2.4.3	Função Objetivo	27
2.4.4	Função de Restrição	29
2.4.5	Método das Assíntotas Móveis (MMA)	29
3	METODOLOGIA	31
4	RESULTADOS	37
5	CONSIDERAÇÕES FINAIS	43
	REFERÊNCIAS	44
	APÊNDICE A – ROTINAS DE GERAÇÃO DOS ARQUIVOS DEPENDENTES DA MALHA	47
A.1	<i>kappa</i>	47
A.2	<i>sampleDict</i>	50
	APÊNDICE B – MALHAS UTILIZADAS	52
B.1	Malhas - MVF	52
B.1.1	Bocal	52
B.1.2	Curva T	53
B.1.3	Joelho	54
B.2	Malhas - MEF	56
B.2.1	Bocal	56
B.2.2	Curva T	57
B.2.3	Joelho	58
	APÊNDICE C – CONDIÇÕES DE CONTORNO UTILIZADAS NO MVF	60

C.1	Velocidade	60
C.2	Pressão	61
C.3	Viscosidade Turbulenta	62
	APÊNDICE D – ARQUIVOS DE CONTROLE DO MVF	64
D.1	<i>controlDict</i>	64
D.2	<i>fvSchemes</i>	65
D.3	<i>fvSolution</i>	66
	APÊNDICE E – ROTINA DE AUTOMATIZAÇÃO EM BASH	68

1 INTRODUÇÃO

Otimização é um tópico muito presente no imaginário popular. Além da redução de custos possível de ser obtida por meio de otimizações, elas permitem um melhor uso dos recursos limitados de nosso planeta. Entretanto, tal tópico não é presente apenas nesse imaginário popular. Uma simples busca, na base de dados SCOPUS, pelo termo “otimização”, resulta em mais de 180 mil resultados nas mais diversas áreas, desde Engenharia (responsável por aproximadamente metade dos resultados), até mesmo Psicologia, apenas no ano de 2023.

Com respeito à Engenharia, otimização de geometrias é possivelmente o tópico mais difundido entre o público geral, especialmente a respeito de otimização de perfis de asas de avião. Um bom exemplo é Muller (2021), no qual foi desenvolvido uma otimização do perfil de uma asa de avião, através da Otimização de Forma, que resultou em uma redução do arrasto em 44%. Obviamente, uma redução de arrasto pode ser convertida em uma economia do combustível utilizado pelo veículo.

Embora esse tipo de otimização seja a mais popular, ela é apenas uma pequena fração do que pode ser feito. Outras aplicações interessantes são apresentadas em (Xie *et al.*, 2023), que otimizou a geometria do sistema de resfriamento de placas solares e, consequentemente, aumentou a eficiência de tais placas e em Zheng *et al.* (2021), que otimizou o tanque de combustível de um avião, a fim de reduzir os efeitos do “balançar” do combustível na estabilidade da aeronave. Ambos sendo realizados por meio da Otimização Topológica, que é o método favorito ao tratar de otimização de geometrias, visto que não requer um conhecimento prévio da geometria ótima e pode gerar geometrias que não eram esperadas.

De acordo com Koga (2010), embora conhecido antes, o Método de Otimização Topológica teve seu desenvolvimento muito acelerado no início da década de 90, com o trabalho de Bendsøe e Kikuchi (1988), que objetivou a otimização de estruturas mecânicas. Já a otimização de “escoamentos” e de dispositivos só viria a ser abordada em Borrvall e Petersson (2003). Desde então, a área de Otimização Topológica como um todo desenvolveu-se bastante, especialmente em anos mais recentes. Aproximadamente dois terços dos trabalhos foram publicados no período de 2013 – 2023, e um dos primeiros trabalhos de otimização topológica considerando escoamentos turbulentos pode ser creditado a Yoon (2016).

Normalmente, trabalhos nessa área utilizam o Método dos Elementos Finitos (MEF) para otimizar e resolver o escoamento. Entretanto, a utilização desse método para escoamentos exige modificações da modelagem MEF apenas para atingir a convergên-

cia da simulação do escoamento (Mortensen; Langtangen; Wells, 2011), o que não é necessário ao utilizar a abordagem por Método dos Volumes Finitos (MVF). Entretanto, embora utilizar o MVF resolva esse problema, um contratempo aparece. De acordo com Alonso *et al.* (2022), um problema encontrado na modelagem MVF é o cálculo eficiente das derivadas que são utilizadas na Otimização Topológica. Essas derivadas aparecem naturalmente na abordagem por MEF. O primeiro trabalho a realizar uma abordagem mista entre ambos os métodos a fim de suprir suas limitações foi feito por Alonso *et al.* (2022), utilizando o software OpenFOAM® (e consequentemente MVF) para resolver o escoamento e o software FeniCs (e consequentemente o MEF) para o cálculo das derivadas.

Na mesma linha do trabalho realizado por Alonso *et al.* (2022), foi desenvolvido por Lima, Aguiar e Romero (2024) uma ferramenta que também utiliza o software OpenFOAM® para resolver o escoamento, mas que substitui o FeniCs por rotinas personalizadas no Octave. Essa nova ferramenta usa apenas códigos abertos o que permite uma facilidade maior de manipulação, até mesmo permitindo a substituição do Octave por rotinas em linguagens de mais baixo nível que permitiriam a redução do tempo de execução. Entretanto, essa ferramenta operava apenas com escoamentos em regime laminar.

Visto que uma vasta gama dos escoamentos de interesse industrial ocorrem em regime turbulento, o objetivo do presente trabalho é desenvolver a ferramenta feita por Lima, Aguiar e Romero (2024), permitindo que a mesma também otimize escoamentos turbulentos, utilizando softwares de código aberto. Mais especificamente, busca-se:

- Resolver o escoamento através do Método dos Volumes Finitos, utilizando o software OpenFOAM®;
- Aplicar o modelo de turbulência de de Wray-Agarwal (Wray; Agarwal, 2015) ao trabalho feito por Lima, Aguiar e Romero (2024);
- Exportar os resultados para o software Octave;
- Realizar o cálculo das derivadas através do Método dos Elementos Finitos, sem resolver novamente o escoamento, por meio de rotinas personalizadas no software Octave;
- Otimizar a geometria através do Método das Assíntotas Móveis (Svanberg, 1987);
- Automatizar o processo através de *Bash Scripting*;
- Comparar qualitativamente os resultados obtidos da otimização de um bocal, uma curva T e um joelho em 90°, tanto em regime laminar quanto em regime turbulento, com a literatura disponível.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo descreve-se a fundamentação teórica utilizada no desenvolvimento do presente trabalho. O referencial teórico utilizado foi dividido em 4 partes: Na Seção 2.1 explicita-se e desenvolve-se as equações que descrevem o comportamento dos escoamentos que serão analisados. Já na Seção 2.2 descreve-se o Método dos Volumes Finitos (MVF), que é utilizado para fazer uma aproximação das equações descritas na seção anterior, visto que elas são de alta complexidade de solução.

Na Seção 2.3 descreve-se o Método dos Elementos Finitos (MEF), que, embora tenha a mesma função que o MVF (porém uma diferente abordagem), é necessário pois o método de otimização escolhido, explicado na Seção 2.4, necessita do cálculo dos gradientes de uma função objetivo.

Essas quatro seções juntas compõem o referencial teórico julgado necessário e suficiente para este trabalho.

2.1 Equações Governantes

O escoamento bidimensional, em regime estacionário, de um fluido newtoniano em um meio poroso, desprezados os efeitos de compressibilidade, o efeito gravitacional e o termo de Forchheimer, é descrito pelo conjunto de equações a seguir:

$$\nabla \cdot \mathbf{u} = 0 \quad (2.1)$$

$$\rho(\mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \mu \nabla^2(\mathbf{u}) + k_p \mathbf{u} \quad (2.2)$$

A Equação (2.1) é a Equação da Continuidade, e a Equação (2.2) é a Equação de Brinkman. Nessas equações, ρ representa a massa específica do fluido em um ponto P qualquer, $\mathbf{u}$ sua velocidade, p refere-se à pressão dele naquele ponto, μ expressa sua viscosidade e k_p é a permeabilidade inversa do meio, onde o termo $k_p \mathbf{u}$ representa uma força resistiva que aparece por causa da passagem do fluido em um meio poroso. Tal termo também é conhecido como termo de Darcy. A consideração do escoamento em meio poroso se faz necessária pois é o artifício matemático que vincula a Otimização Topológica à física do problema, e fornece uma ferramenta para controle da geometria de tal forma que uma região de porosidade máxima é tratada como uma região sólida. Tal relação é esclarecida na Seção 2.4.

O termo de Forchheimer é extensamente documentado na literatura, como nos trabalhos de Bae e Kim (2016), Tanner, Gorman e Sparrow (2019) e Lee e Ih (2003). A relevância desse termo, frente ao termo de Darcy que foi abordado até o presente momento, é dependente do valor do número de Reynolds dentro do poro (Shahzad;

Hickel; Modesti, 2022). Esse termo é não linear e, conseqüentemente, mais complexo de ser tratado e, por isso é desprezado neste trabalho. Futuros trabalhos podem avaliar sua interferência no Método de Otimização Topológica.

Existem várias abordagens para resolver o problema da turbulência, dentre elas as mais conhecidas são as abordagens por DNS (Direct Numerical Simulation), LES (Large-Eddy Simulation) e RANS (Reynolds Average Navier-Stokes). A abordagem DNS consiste em resolver as equações governantes em sua plenitude e, por causa das características de um regime turbulento, essa abordagem requer malhas muito finas, o que acarreta um custo computacional muito grande. No outro extremo do espectro, há a abordagem RANS, que consiste em resolver o escoamento médio, diminuindo muito o custo computacional. A abordagem LES é o meio termo entre as duas, onde os maiores vórtices presentes no escoamento são simulados e os menores vórtices são equacionados. Como os problemas de otimização geralmente trabalham com dispositivos em regimes de trabalho, uma análise do escoamento médio nesse contexto é mais coerente. Aliado a isso, um menor custo computacional tornou a abordagem RANS mais atrativa e a utilizada por este trabalho.

De acordo com a abordagem RANS, uma das principais características de um escoamento turbulento é a flutuação aleatória existente nos campos, neste caso de velocidade e de pressão, de tal forma que a velocidade instantânea passa a ser caracterizada por uma componente de velocidade média e uma componente de flutuação, de tal forma que ela pode ser escrita como:

$$\mathbf{u} = \mathbf{u}_m + \mathbf{u}' \quad (2.3)$$

Um raciocínio similar pode ser feito com o campo de pressão:

$$p = p_m + p' \quad (2.4)$$

Onde $\mathbf{u}_m$ é a velocidade média, $\mathbf{u}'$ são as flutuações instantâneas da velocidade, p_m é a pressão média e p' são as flutuações instantâneas da velocidade.

Conforme mencionado anteriormente, a abordagem RANS consiste em analisar o escoamento médio. Para tal, utiliza-se a média temporal tanto da Equação da Conservação de Momento, neste caso a Equação (2.2), quanto da Equação da Conservação de Massa, que é a Equação da Continuidade. Para a Equação da Continuidade, temos:

$$\overline{\nabla \cdot \mathbf{u}} = 0 \quad (2.5)$$

Substituindo a Equação (2.3) na Equação (2.5):

$$\overline{\nabla \cdot (\mathbf{u}_m + \mathbf{u}')} = 0$$

$$\overline{\nabla \cdot \mathbf{u}_m} + \overline{\nabla \cdot \mathbf{u}'} = \nabla \cdot \overline{\mathbf{u}_m} + \nabla \cdot \overline{\mathbf{u}'} = 0$$

Entretanto, a média das variações, $\overline{\mathbf{u}'} = 0$ e a média da velocidade média é a própria velocidade média. Consequentemente, a Equação da Continuidade pode ser reescrita na forma da Equação (2.6)

$$\nabla \cdot \mathbf{u}_m = 0 \quad (2.6)$$

A mesma lógica será aplicada à Equação (2.2):

$$\overline{\rho(\mathbf{u} \cdot \nabla \mathbf{u})} = -\overline{\nabla p} + \overline{\mu \nabla^2(\mathbf{u})} + \overline{k_p \mathbf{u}} \quad (2.7)$$

Cada um dos termos será avaliado separadamente. A Equação (2.8) indica como se decompõe o termo de pressão.

$$-\overline{\nabla p} = -\overline{\nabla(p_m + p')} = -\nabla p_m \quad (2.8)$$

De modo similar, o termo advindo da porosidade e o termo difusivo são decompostos, respectivamente, por:

$$\overline{k_p \mathbf{u}} = k_p \overline{\mathbf{u}_m + \mathbf{u}'} = k_p \mathbf{u}_m \quad (2.9)$$

$$\overline{\mu \nabla^2(\mathbf{u})} = \mu \nabla^2 \overline{\mathbf{u}_m + \mathbf{u}'} = \mu \nabla^2 \mathbf{u}_m \quad (2.10)$$

O termo advectivo pode ser escrito, ao utilizar a regra do produto, como:

$$\overline{\rho \mathbf{u}(\nabla \mathbf{u})} = \rho \overline{\nabla \cdot (\mathbf{u} \mathbf{u}^T) - \mathbf{u}(\nabla \cdot \mathbf{u}^T)} \quad (2.11)$$

Contudo, de acordo com a Equação da Continuidade, $\nabla \cdot \mathbf{u}^T = 0$ e o termo advectivo pode ser escrito como:

$$\overline{\rho \mathbf{u}(\nabla \mathbf{u}^T)} = \rho \overline{\nabla \cdot (\mathbf{u} \mathbf{u}^T)} \quad (2.12)$$

Aplicando a Equação (2.3) na Equação (2.12):

$$\rho \nabla \cdot (\overline{\mathbf{u} \mathbf{u}^T}) = \rho \nabla \cdot ((\mathbf{u}_m + \mathbf{u}')(\mathbf{u}_m + \mathbf{u}')^T) = \rho \nabla \cdot \overline{\mathbf{u}_m (\mathbf{u}_m^T)} + \rho \nabla \cdot (\overline{\mathbf{u}' \mathbf{u}'^T}) \quad (2.13)$$

O termo $\rho \nabla \cdot (\overline{\mathbf{u}' \mathbf{u}'^T})$ compõe o tensor de Reynolds, e pode ser associado a uma tensão de origem turbulenta. Pela Hipótese de Boussinesq, essa tensão é similar à tensão molecular do termo difusivo, de tal forma que:

$$\rho \nabla \cdot (\overline{\mathbf{u}' \mathbf{u}'^T}) = \mu_{turb} \nabla^2 \mathbf{u}_m \quad (2.14)$$

E, portanto, o termo advectivo é dado por:

$$\rho \overline{\nabla \cdot (\mathbf{u} \mathbf{u})} = \rho \mathbf{u}_m \cdot (\nabla \mathbf{u}_m^T) + \mu_{turb} \nabla^2 \mathbf{u}_m \quad (2.15)$$

Assim sendo, aplicando as Equações (2.8), (2.9), (2.10) e (2.15) na Equação (2.7):

$$\rho \mathbf{u}_m (\nabla \mathbf{u}_m^T) = -\nabla p_m + (\mu + \mu_{turb}) \nabla^2 \mathbf{u}_m + k_p \mathbf{u}_m \quad (2.16)$$

As Equações (2.1) e (2.2), que anteriormente representavam um sistema com 3 incógnitas e 3 equações (p e $\mathbf{u}$), geraram uma nova incógnita após a aplicação das decomposições da abordagem RANS, μ_{turb} . A definição dessa incógnita pode ser feita de diversas maneiras, cada uma resultando em um Modelo de Turbulência.

O Modelo de Turbulência a ser utilizado neste trabalho é a Equação de Wray-Agarwal (Wray; Agarwal, 2015) que, modificado para a otimização topológica, como mostrado em (Alonso *et al.*, 2022), é dado por:

$$\begin{aligned} \rho \mathbf{u}_m \cdot \nabla \mu_{turb} = & \underbrace{\nabla \cdot \left[(\sigma_R \rho \mu_{turb}) \nabla \mu_{turb} \right]}_{\text{Transporte de turbulência por tensões viscosas e turbulentas}} + \underbrace{f_1 C_{1k\omega} \rho \frac{\mu_{turb}}{S} (\nabla \mu_{turb} \cdot \nabla S)}_{\text{Dissipação de turbulência na parede}} + \\ & - \underbrace{\rho (1 - f_1) \min \left[C_{2k\epsilon} (\mu_{turb})^2 \left(\frac{\nabla S \cdot \nabla S}{S^2} \right), C_m \nabla \mu_{turb} \cdot \nabla \mu_{turb} \right]}_{\text{Dissipação de turbulência na corrente livre}} + \\ & - \underbrace{\lambda_{\mu_{turb}} k_p \mu_{turb}}_{\text{Atenuação da turbulência no meio poroso}} + \underbrace{\rho C_1 \mu_{turb} S}_{\text{Produção de turbulência}} \end{aligned} \quad (2.17)$$

Onde a variável de interesse na Equação (2.17) é a viscosidade turbulenta, μ_{turb} . Já $\lambda_{\mu_{turb}}$, de acordo com Alonso *et al.* (2022), é um parâmetro ajustável para a intensidade da atenuação da turbulência dentro do material sólido modelado. Neste trabalho, considera-se $\lambda_{\mu_{turb}} = 1$ para simplificar o problema. No trabalho de Alonso *et al.* (2022), uma atenuação específica para cada escoamento foi utilizada. Além disso, S é definido como a deformação média:

$$S = \sqrt{2\mathbf{S} : \mathbf{S}} \quad (2.18)$$

O ponto duplo representa a operação de produto interno duplo e $\mathbf{S}$ é o tensor taxa de deformação, dado pela parte simétrica do gradiente de velocidade e descrito na Equação (2.19)

$$\mathbf{S} = \frac{1}{2} \left((\nabla \mathbf{u}_m) + (\nabla \mathbf{u}_m)^T \right) \quad (2.19)$$

Para evitar uma possível divisão por zero, utiliza-se $S = \max(S, 10^{-16} s^{-1})$ (Wray; Agarwal, 2015). As paredes são consideradas por meio da seguinte função de amorte-

cimento (Wray; Agarwal, 2015):

$$f_\mu = \frac{\chi^3}{\chi^3 + C_\omega^3} \quad (2.20)$$

Onde $\chi = R/\nu$. A função de parede é dada por (Wray; Agarwal, 2015):

$$f_1 = \tanh(arg_1^4) \quad (2.21)$$

$$arg_1 = \left(\frac{\nu + R}{2}\right) \left(\frac{\eta^2}{C_\mu k \omega}\right) \quad (2.22)$$

E:

$$k = \frac{\nu_{turb} S}{\sqrt{C_\mu}} \quad (2.23)$$

$$\omega = \frac{S}{\sqrt{C_\mu}} \quad (2.24)$$

$$\eta = S \max\left(1, \left|\frac{W}{S}\right|\right) \quad (2.25)$$

$$W = \sqrt{2\mathbf{W} : \mathbf{W}} \quad (2.26)$$

$$\mathbf{W} = \frac{1}{2} \left((\nabla \mathbf{u}_m) - (\nabla \mathbf{u}_m)^T \right) \quad (2.27)$$

$$(2.28)$$

As constantes desse modelo, de acordo com Wray e Agarwal (2015), são:

$$C_{1k\omega} = 0,0829$$

$$C_{1k\epsilon} = 0,1284$$

$$C_1 = f_1(C_{1k\omega} - C_{1k\epsilon}) + C_{1k\epsilon}$$

$$\sigma_{k\omega} = 0,72$$

$$\sigma_{k\epsilon} = 1,0$$

$$\sigma_R = f_1(\sigma_{k\omega} - \sigma_{k\epsilon}) + \sigma_{k\epsilon}$$

$$C_{2k\omega} = \frac{C_{1k\omega}}{\kappa^2} + \sigma_{k\omega}$$

$$C_{2k\epsilon} = \frac{C_{1k\epsilon}}{\kappa^2} + \sigma_{k\epsilon}$$

$$\kappa = 0,41$$

$$C_\omega = 8,54$$

$$C_\mu = 0,09$$

$$C_m = 8,0$$

2.2 Método dos Volumes Finitos

As equações explicitadas na seção 2.1 são suficientes para descrever o comportamento do fluido (contanto que as condições impostas sejam atendidas). Entretanto, resolver analiticamente esse conjunto de equações é impossível com o conhecimento atual.

Para resolver esse problema, existe uma vasta gama de métodos que transformam o problema de resolver um sistema de equações diferenciais em algo factível, como um sistema linear de equações, por meio de aproximações.

Dentre esses métodos, os mais conhecidos são o Método das Diferenças Finitas (MDF - FDM), o Método dos Elementos Finitos (MEF - FEM) e o Método dos Volumes Finitos (MVF - FVM), sendo o último o mais utilizado no campo da Dinâmica dos Fluidos Computacional (DFC - CFD) e também o escolhido para encontrar as variáveis de estado (velocidade, pressão e viscosidade turbulenta) neste trabalho, e o desenvolvimento aqui apresentado tem como base o desenvolvimento realizado por Darwish e Moukalled (2016).

O MVF se utiliza da Equação de conservação de uma propriedade escalar ϕ qualquer, que neste trabalho será considerada em regime permanente, sendo expressa pela Equação 2.29.

$$\underbrace{\nabla \cdot (\rho \mathbf{u} \phi)}_{\text{termo convectivo}} = \underbrace{\nabla \cdot (\Gamma^\phi \nabla \phi)}_{\text{termo difusivo}} + \underbrace{Q^\phi}_{\text{termo fonte}} \quad (2.29)$$

Onde Γ^ϕ é o coeficiente de difusão de ϕ e Q^ϕ é o termo fonte da Equação. Considerando um volume de controle V_c , a Equação (2.29) em sua formulação integral representa um balanço da propriedade no mesmo, dado por:

$$\int_{V_c} \nabla \cdot (\rho \mathbf{u} \phi) dV_c = \int_{V_c} \nabla \cdot (\Gamma^\phi \nabla \phi) dV_c + \int_{V_c} Q^\phi dV_c$$

Uma vez aplicado o Teorema da Divergência à Equação, sendo a área superficial do volume de controle dado por S_c e $\hat{\mathbf{n}}$ o vetor normal à superfície, pode ser descrito como:

$$\oint_{S_c} (\rho \mathbf{u} \phi) \cdot \hat{\mathbf{n}} dS_c = \oint_{S_c} (\Gamma^\phi \nabla \phi) \cdot \hat{\mathbf{n}} dS_c + \int_{V_c} Q^\phi dV_c \quad (2.30)$$

É interessante de se notar que a Equação (2.30) é válida em um volume. Conforme o volume de controle escolhido fica menor, mais o balanço da propriedade no volume se aproxima do valor da propriedade no ponto, implicando na necessidade de discretizar o domínio. Agora, para contornar a solução das integrais apresentadas, aplica-se a Quadratura de Gauss, definida, em um domínio Ω para uma dada função F :

$$\int_{\Omega} F d\Omega \approx \sum_{i=1}^{ip} F(i) \Omega \omega_i \quad (2.31)$$

A grande vantagem da Equação (2.31) é que, originalmente, a função era avaliada em todo o volume de controle e após a aplicação da Quadratura de Gauss ela é avaliada em pontos específicos, chamados de pontos de integração, dados por ip . Consequentemente, é necessário fazer uma compensação por causa dessa substituição, que é realizada através dos pesos ω . No contexto do Método dos Volumes Finitos, costuma-se realizar a aproximação por Quadratura de Gauss com apenas um ponto, e para esse caso, o valor do peso é $\omega = 1$. Portanto, após a aplicação desse método, a Equação (2.30) pode ser escrita como:

$$\sum_{i=1}^{n_f} S_c((\rho \mathbf{u} \phi) - (\Gamma^\phi \nabla \phi)) \cdot \hat{\mathbf{n}}_f = Q_c^\phi V_c \quad (2.32)$$

O lado esquerdo representa o fluxo nas faces, que pode ser decomposto em uma parte linear (dependente do valor da propriedade no centro do volume, ϕ_c e do valor na face ϕ_f analisada) e uma parte que não pode ser escrito em função dessas variáveis, de tal forma que:

$$\sum_{i=1}^{n_f} S_c((\rho \mathbf{u} \phi) - (\Gamma^\phi \nabla \phi)) \cdot \hat{\mathbf{n}}_f = FC_f \phi_c + \sum_{i=1}^{n_f} (FF_f \phi_f)_i + FV_f$$

Onde FC_f e FF_f são os coeficientes de linearização do fluxo, respectivamente da propriedade no centro do volume e na face analisada e FV_f é a parte não linearizada. Já a linearização do termo fonte é dado por:

$$Q_c^\phi V_c = FC \phi_c + FV$$

Que, após os arranjos necessários, pode ser escrito como uma Equação algébrica, dada por:

$$a_c \phi_c + \sum_{F \sim n_f} (a_f \phi_f) = b_c \quad (2.33)$$

Onde $a_c = \sum_{F \sim n_f} (FC_f) - FC$, $a_f = FF_f$ e $b_c = \sum_{F \sim n_f} (FV_f) - FV$. Obviamente, o valor dessas constantes depende do termo discretizado e do esquema de discretização utilizado.

2.3 Método dos Elementos Finitos

O desenvolvimento aqui apresentado foi baseado nos trabalhos de Reddy e Gartling (2010) e Romero e Silva (2014). O MEF é derivado de uma classe de métodos chamado Métodos Variacionais, que consistem na aproximação das variáveis de interesse através de uma função, onde o tipo de função utilizado determina o método. Sejam $\tilde{\mathbf{u}}$ e $\tilde{p}$, respectivamente, as aproximações da velocidade média, $\mathbf{u}_m$ e da pressão média, p_m .

Substituindo-as na Equação (2.6) e na Equação (2.16):

$$R_c = \nabla \cdot \tilde{\mathbf{u}} \quad (2.34)$$

$$\mathbf{R}_b = k_p \tilde{\mathbf{u}} + (\mu + \mu_{turb}) \nabla^2 \tilde{\mathbf{u}} - \rho (\nabla \tilde{\mathbf{u}}) \tilde{\mathbf{u}} - \nabla \tilde{p} \quad (2.35)$$

R_c e $\mathbf{R}_b$ são os resíduos resultantes da substituição das aproximações na Equação da Continuidade e na Equação de Brinkman, respectivamente. Um leitor atento notará que não há uma função de aproximação para μ_{turb} . Isso porque, para os propósitos de criação das matrizes necessárias para o Método de Otimização Topológica (que é o objetivo dessa seção) essa grandeza não será considerada uma variável, a fim de diminuir a complexidade do problema. Para amenizar essa consideração, a viscosidade turbulenta (calculada através do MVF) será considerada em cada um dos nós da discretização, como será mostrado posteriormente.

Os resíduos citados são tratados através do Método dos Resíduos Ponderados, que consiste em, dado um domínio Ω , utilizar uma função de ponderação que compensa o resíduo no domínio mencionado. Considerando as funções de ponderação M and $\mathbf{W}$ para a Eq. (2.34) e a Eq. (2.35), temos:

$$\int_{\Omega} R_c M \, d\Omega = 0 \quad \text{e} \quad \int_{\Omega} \mathbf{R}_b \cdot \mathbf{W} \, d\Omega = 0$$

O que significa que a Equação (2.34) e a Equação (2.35) assumem a seguinte forma:

$$\int_{\Omega} (\nabla \cdot \tilde{\mathbf{u}}) M \, d\Omega = 0 \quad (2.36)$$

$$\begin{aligned} \int_{\Omega} (k_p \tilde{\mathbf{u}}) \cdot \mathbf{W} \, d\Omega + \int_{\Omega} ((\mu + \mu_{turb}) \nabla^2 \tilde{\mathbf{u}}) \cdot \mathbf{W} \, d\Omega + \\ - \int_{\Omega} \rho ((\nabla \tilde{\mathbf{u}}) \tilde{\mathbf{u}}) \cdot \mathbf{W} \, d\Omega - \int_{\Omega} (\nabla \tilde{p}) \cdot \mathbf{W} \, d\Omega = 0 \end{aligned} \quad (2.37)$$

Entretanto, usando a regra do produto, o Teorema da Divergência e considerando que Γ é o contorno do domínio, com seu vetor normal dado por $\hat{\mathbf{n}}$:

$$\int_{\Omega} (\nabla^2 \tilde{\mathbf{u}}) \cdot \mathbf{W} \, d\Omega = \oint_{\Gamma} (\nabla \tilde{\mathbf{u}} \cdot \hat{\mathbf{n}}) \cdot \mathbf{W} \, d\Gamma - \int_{\Omega} \nabla \tilde{\mathbf{u}} : \nabla \mathbf{W} \, d\Omega \quad (2.38)$$

E portanto, substituindo a Equação (2.38) na Equação (2.37), é obtido o seguinte sistema de equações:

$$\int_{\Omega} (\nabla \cdot \tilde{\mathbf{u}}) M \, d\Omega = 0 \quad (2.39)$$

$$\begin{aligned} k_p \int_{\Omega} \tilde{\mathbf{u}} \cdot \mathbf{W} \, d\Omega + (\mu + \mu_{turb}) \oint_{\Gamma} (\nabla \tilde{\mathbf{u}} \cdot \hat{\mathbf{n}}) \cdot \mathbf{W} \, d\Gamma + \\ - (\mu + \mu_{turb}) \int_{\Omega} \nabla \tilde{\mathbf{u}} : \nabla \mathbf{W} \, d\Omega - \rho \int_{\Omega} ((\nabla \tilde{\mathbf{u}}) \tilde{\mathbf{u}}) \cdot \mathbf{W} \, d\Omega = \int_{\Omega} (\nabla \tilde{p}) \cdot \mathbf{W} \, d\Omega \end{aligned} \quad (2.40)$$

A Equação (2.39) e a Equação (2.40) representam, respectivamente, as formas fracas da Equação da Continuidade e da Equação de Brinkman. Considerando o campo de velocidade como $\mathbf{u} = [u_1 \ u_2]^T$, onde u_1 é a componente da velocidade no eixo x e u_2 é sua componente no eixo y e utilizando os elementos de Taylor-Hood, onde a velocidade e a pressão são interpolados por funções quadráticas (nove nós por elemento) e lineares (quatro nós por elemento) respectivamente, as aproximações para essas variáveis são dadas pela Equação (2.41):

$$\tilde{\mathbf{u}}_1 = \sum_{j=1}^n \phi_j(u_1)_j = \Psi^T \mathbf{u}_1 \quad \tilde{\mathbf{u}}_2 = \sum_{j=1}^n \phi_j(u_2)_j = \Psi^T \mathbf{u}_2 \quad \tilde{p} = \sum_{j=1}^n \chi_j(p)_j = \chi^T \mathbf{p} \quad (2.41)$$

Onde $\mathbf{u}_1$, $\mathbf{u}_2$ e $\mathbf{p}$ são vetores com os valores nodais da solução aproximada para o campo de velocidade no eixo x, no eixo y e para o campo de pressão definidos na formulação fraca do MEF e Ψ e χ são, respectivamente, as funções de interpolação para a velocidade e para a pressão. Substituindo as aproximações mostradas na Equação (2.41) na Eq. (2.39) e na Eq. (2.40), o sistema de equações algébricas do MEF são obtidas. Para o caso bidimensional, considerando o Método de Galerkin onde o campo desconhecido é representado pelo mesmo tipo de função utilizada para as funções de ponderação, o sistema de equações algébricas pode ser representado pela Equação (2.42):

$$\begin{aligned} & \begin{bmatrix} \mathbf{K}_p & 0 & 0 \\ 0 & \mathbf{K}_p & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{p} \end{Bmatrix} + \begin{bmatrix} \mathbf{C}_1(\mathbf{u}_1) + \mathbf{C}_2(\mathbf{u}_2) & 0 & 0 \\ 0 & \mathbf{C}_1(\mathbf{u}_1) + \mathbf{C}_2(\mathbf{u}_2) & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{p} \end{Bmatrix} \\ & + \begin{bmatrix} 2\mathbf{K}_{11} + \mathbf{K}_{22} & \mathbf{K}_{12} & -\mathbf{Q}_1 \\ \mathbf{K}_{21} & \mathbf{K}_{11} + 2\mathbf{K}_{22} & -\mathbf{Q}_2 \\ -\mathbf{Q}_1^T & -\mathbf{Q}_2^T & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{p} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{0} \end{Bmatrix} \end{aligned} \quad (2.42)$$

Onde os coeficientes das matrizes são:

$$\begin{aligned} \mathbf{K}_p &= k_p \int_{\Omega} \Psi \Psi^T d\Omega & \mathbf{C}_i(\mathbf{u}_i) &= \rho \int_{\Omega} \Psi (\Psi^T \mathbf{u}_i) \frac{\partial \Psi^T}{\partial x_i} d\Omega \\ \mathbf{K}_{ij} &= (\mathbf{I}\mu + \mathbf{M}_t) \int_{\Omega} \frac{\partial \Psi}{\partial x_j} \frac{\partial \Psi^T}{\partial x_i} d\Omega & \mathbf{Q}_i &= \int_{\Omega} \frac{\partial \Psi}{\partial x_i} \chi^T d\Omega & \mathbf{f}_i &= \oint_{\Gamma} h_i \Psi d\Gamma \quad i=1,2 \end{aligned}$$

$\mathbf{M}_t$ é uma matriz diagonal cuja diagonal contém os valores nodais da viscosidade turbulenta (na malha quadrática). O sistema obtido é não-simétrico e não-linear. Para fins de simplificação, será considerado um vetor solução $\mathbf{z}$ que contém todas as variáveis de estado, de tal forma que $\mathbf{z} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \mathbf{p}]^T$. Ele será utilizado na seção 2.4. Assim sendo, com base na Equação (2.42), o sistema não linear pode ser escrito de forma compacta pela Equação (2.43):

$$\mathbf{K}_t \mathbf{z} = \mathbf{f} \quad (2.43)$$

Onde $\mathbf{K}_t$ é a soma total das matrizes coeficientes do sistema não linear e $\mathbf{f}$ é o vetor dos termos independentes.

2.4 Método de Otimização Topológica

O Método de Otimização Topológica (MOT) é um dos métodos de otimização de geometria mais conhecidos, junto com a Otimização Paramétrica e a Otimização de Forma, com a vantagem de que não é necessário um conhecimento prévio da geometria para sua utilização. Ele é baseado em, uma vez dado um domínio estendido de projeto descrito com mais detalhes na Seção 2.4.1, buscar a minimização de uma função objetivo específica, descrita na Seção 2.4.3, levando em consideração algumas funções de restrição, dentre as quais a mais comum é a quantidade de material presente no domínio estendido, a chamada "fração de volume" que é descrita na Seção 2.4.4.

A minimização da função objetivo, que será explicitada a seguir, pode ser feita por algoritmos diversos. Dentre eles, os mais conhecidos são os algoritmos baseados no Critério da Optimalidade, a Programação Linear Sequencial (PSL) e o Método das Assíntotas Móveis (MAM), que será o utilizado neste trabalho. O MAM requer o cálculo dos gradientes da função objetivo com relação à variável de projeto, que aparecem naturalmente na montagem do sistema não linear quando o método a ser escolhido para resolver o escoamento é o MEF (ver Seção 2.3). Entretanto, ele também requer a solução do escoamento, que no presente trabalho será realizada utilizando o MVF (explicitado na Seção 2.2).

2.4.1 Domínio Estendido de Projeto

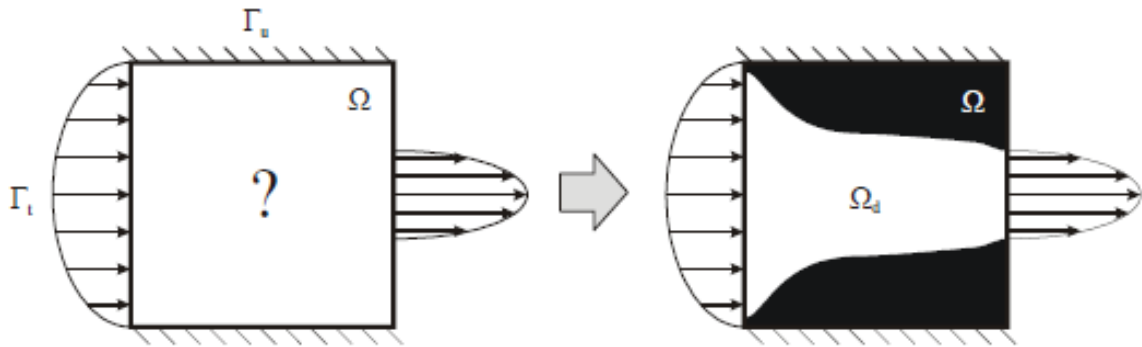
De acordo com o apresentado por (Koga, 2010), o domínio estendido de projeto é uma região do espaço, fixa e pré-determinada, onde o material pode ser "adicionado" ou "retirado" livremente e a solução ótima (Ω_d) é desconhecida *a priori*. Tal domínio gera uma região de trabalho estendida (Ω), em torno da região onde se espera a topologia ótima, de tal forma que seja possível cobrir uma maior região na análise de soluções, garantindo que a solução ótima esteja contida nesse espaço de soluções possíveis.

Além disso, outra vantagem deste domínio estendido é que não é necessário adaptar a malha de projeto conforme a solução obtida ao longo do processo iterativo, uma vez que ela se mantém fixa. Entretanto, pelo fato de ser um domínio estendido, é necessário sempre trabalhar com mais elementos do que o estritamente "necessário",

o que acarreta em um maior custo computacional. Ainda assim, essa estratégia é eficiente e facilita a implementação, sendo a mais utilizada neste tipo de otimização.

A Figura (1) ilustra o conceito aqui apresentado.

Figura 1 – Domínio Fixo Estendido de Projeto.



Fonte: Retirado de (Koga, 2010).

2.4.2 Modelo de Material

Nas aplicações de MOT, o controle do comportamento físico de cada elemento dentro do domínio é dado através da variação de uma pseudo-densidade dos elementos do domínio, onde um elemento cujo valor de pseudo-densidade $\alpha = 1$ significa que o elemento analisado permite o escoamento livre do fluido, enquanto o outro extremo é representado pelo valor de pseudo-densidade $\alpha = 0$, simulando um elemento sólido que impede o escoamento. Como modelado pela primeira vez por Bendsoe (1989), uma propriedade T qualquer pode ser representada pela Equação 2.44.

$$T(\mathbf{x}) = \alpha(\mathbf{x})^p T_0 \quad (2.44)$$

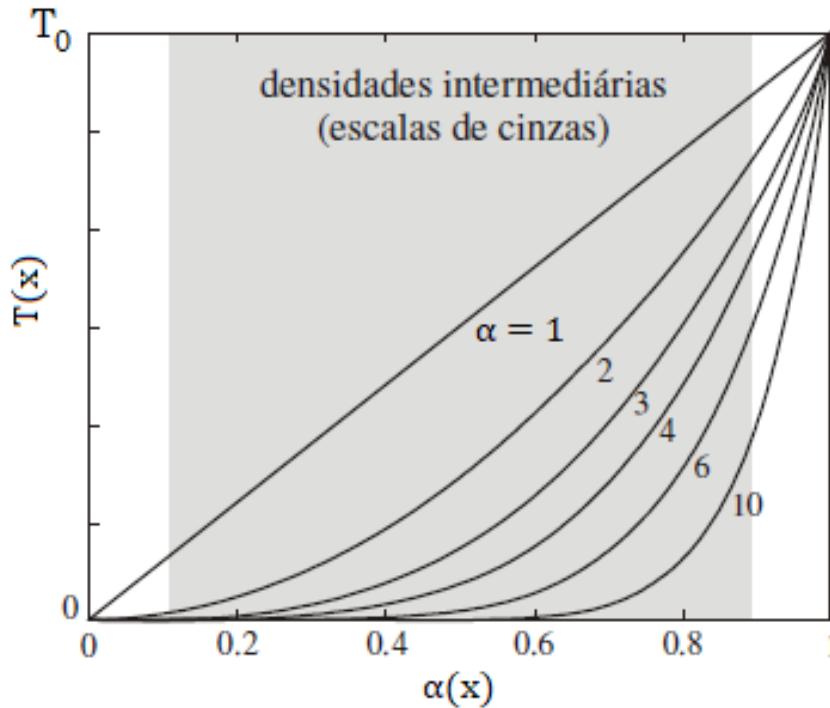
Onde p é uma penalização e T_0 é um valor de referência da propriedade. Utilizando dessa definição, Borrvall e Petersson (2003) define, uma expressão para a constante de proporcionalidade entre a velocidade e a força resistiva do escoamento em um meio fluido, k_p , mencionado na Seção 2.1, em função da pseudo-densidade dada pela Equação (2.45):

$$k_p(\alpha) = k_{p_{max}} + (k_{p_{min}} - k_{p_{max}})\alpha \left(\frac{1+q}{\alpha+q} \right) \quad (2.45)$$

O novo termo, $q > 0$ é um fator de controle da penalização, que surge para resolver um problema comum à otimização topológica: as densidades intermediárias, também conhecido como escalas de cinza. Sob o ponto de vista da manufatura, é desejável que a solução final seja composta apenas de material base (fluido) e de regiões vazias (sólido) (Koga, 2010). E de acordo com a Equação 2.44, a pseudo-densidade pode

obter valores intermediários. Um problema contínuo onde há interesse que ele seja discreto. Assim sendo, a penalização força a solução aos valores extremos de 0 e 1. A Figura 2 mostra a influência da penalização na propriedade efetiva do material.

Figura 2 – Influência da penalização na propriedade efetiva do material



Fonte: Koga (2010), editado

Entretanto, penalizações altas também geram um problema: a convergência da otimização para mínimos locais (Koga, 2010). Para evitar ambos os problemas utiliza-se o Método da Continuação, que consiste em começar o processo de otimização com valores de penalização baixos (por exemplo, 10^{-3}) e, conforme se obtém convergência, aumentar essa penalização (para 10^{-1} , por exemplo). O Método da Continuação aqui utilizado foi automatizado, conforme será explicado no capítulo 3, através de uma progressão geométrica. Tal modelo foi escolhido a fim de facilitar a automatização do processo.

2.4.3 Função Objetivo

Neste trabalho, será considerada a função objetivo desenvolvida por Borrvall e Petersson (2003), ligeiramente adaptada para um caso turbulento, de tal forma que:

$$\Phi = (\mu + \mu_{turb}) \int_{\Omega} \mathbf{u}_m \cdot \nabla^2 \mathbf{u}_m d\Omega + k_p \int_{\Omega} \mathbf{u}_m \cdot \mathbf{u}_m d\Omega = \frac{1}{2} (\mathbf{z}^T \mathbf{C} \mathbf{z}) \quad (2.46)$$

A Equação (2.46) representa a energia cedida pelo fluido ao escoar pelo domínio analisado, e é composta de 2 termos, onde o primeiro representa, simultaneamente, a perda de energia devido à viscosidade do fluido e devido aos efeitos da turbulência e o

segundo reflete a perda de energia oriunda da passagem de um fluido através de um meio poroso. Além disso:

$$\mathbf{C} = \begin{bmatrix} \bar{\mathbf{K}}_d + \bar{\mathbf{K}}_k & 0 \\ 0 & 0 \end{bmatrix} \quad \bar{\mathbf{K}}_d = \begin{bmatrix} 2\mathbf{K}_{11} + \mathbf{K}_{22} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{11} + 2\mathbf{K}_{22} \end{bmatrix} \quad \bar{\mathbf{K}}_k = \begin{bmatrix} \mathbf{K}_p & 0 \\ 0 & \mathbf{K}_p \end{bmatrix}$$

Várias funções objetivos podem ser combinadas de modo a gerar apenas uma função objetivo (ou multiobjetivo), mas neste trabalho apenas a minimização da perda de carga, descrita pela Equação (2.46), será abordada.

Como citado anteriormente, o MAM requer o cálculo do gradiente da função objetivo em relação à variável de projeto para cada um dos elementos do domínio Ω . Neste trabalho, será utilizada a hipótese de “Turbulência Congelada de Taylor”, como feito por Dilgen *et al.* (2018), que diz que a variação das propriedades turbulentas com respeito à variável de projeto é nula, ou seja, $\frac{\partial \mu_{turb}}{\partial \alpha} = 0$. Logo:

$$\frac{\partial \Phi}{\partial \alpha} = \frac{1}{2} \frac{\partial}{\partial \alpha} (\mathbf{z}^T \mathbf{C} \mathbf{z}) = \frac{1}{2} \left(\frac{\partial \mathbf{z}^T}{\partial \alpha} \mathbf{C} \mathbf{z} + \mathbf{z}^T \mathbf{C} \frac{\partial \mathbf{z}}{\partial \alpha} + \mathbf{z} \mathbf{C} \frac{\partial \mathbf{z}^T}{\partial \alpha} \right) = \mathbf{z} \mathbf{C} \frac{\partial \mathbf{z}^T}{\partial \alpha} + \frac{1}{2} \left(\mathbf{z} \frac{\partial \mathbf{C}}{\partial \alpha} \mathbf{z}^T \right) \quad (2.47)$$

Essa manipulação matemática pode ser feita pois a matriz $\mathbf{C}$ é simétrica. O termo $\frac{\partial \mathbf{C}}{\partial \alpha}$ é calculado naturalmente visto que a matriz $\mathbf{C}$ é criada pelo Método dos Elementos Finitos, onde apenas um de seus termos, $\bar{\mathbf{K}}_k$, é dependente de α . De tal forma que essa derivada é dada pela Equação (2.48):

$$\frac{\partial \mathbf{C}}{\partial \alpha} = \begin{bmatrix} \frac{\partial}{\partial \alpha} (\bar{\mathbf{K}}_d + \bar{\mathbf{K}}_k) & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial \alpha} (\bar{\mathbf{K}}_k) & 0 \\ 0 & 0 \end{bmatrix} \quad (2.48)$$

Já o termo $\frac{\partial \mathbf{z}}{\partial \alpha}$ pode ser encontrado através do sistema não linear gerado por MEF. Aplicando o Método de Newton para a resolução de sistemas não lineares na Equação (2.43), temos a existência de um resíduo $\mathbf{r}$, dado por:

$$\mathbf{r} = \mathbf{K}_t \mathbf{z} - \mathbf{f} \quad (2.49)$$

Através desse método, busca-se minimizar o resíduo $\mathbf{r}$, de tal forma que $\frac{d\mathbf{r}}{d\alpha} = 0$. Entretanto, tal resíduo é função tanto de α quanto de $\mathbf{z}$ e, portanto, aplicando a regra da cadeia:

$$\frac{d\mathbf{r}}{d\alpha} = \frac{\partial \mathbf{r}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \alpha} + \frac{\partial \mathbf{r}}{\partial \alpha} \frac{\partial \alpha}{\partial \alpha} = \frac{\partial \mathbf{r}}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \alpha} + \frac{\partial \mathbf{r}}{\partial \alpha} = 0 \quad (2.50)$$

Onde $\frac{\partial \mathbf{r}}{\partial \mathbf{z}}$ é o Jacobiano $\mathbf{J}$ de $\mathbf{r}$. Portanto:

$$\mathbf{J} \frac{\partial \mathbf{z}}{\partial \alpha} = -\frac{\partial \mathbf{r}}{\partial \alpha} = -\frac{\partial}{\partial \alpha} (\mathbf{K}_t \mathbf{z} - \mathbf{f}) = -\frac{\partial \mathbf{K}_t}{\partial \alpha} \mathbf{z} - \mathbf{K}_t \frac{\partial \mathbf{z}}{\partial \alpha} + \frac{\partial \mathbf{f}}{\partial \alpha} \quad (2.51)$$

E conseqüentemente, depois de um arranjo matemático na Equação (2.51), temos:

$$(\mathbf{J} + \mathbf{K}_t) \frac{\partial \mathbf{z}}{\partial \alpha} = \frac{\partial \mathbf{f}}{\partial \alpha} - \frac{\partial \mathbf{K}_t}{\partial \alpha} \mathbf{z} \quad (2.52)$$

Resolvendo o sistema linear expresso na Equação (2.52), encontra-se o valor de $\frac{\partial \mathbf{z}}{\partial \alpha}$ para ser substituído no cálculo do gradiente da função objetivo, expressado pela Equação (2.47).

2.4.4 Função de Restrição

A função limitante necessária para a Otimização Topológica neste trabalho será a quantidade de material disponível. Apenas uma parte do domínio estendido de projeto será sólida, e a outra parte será fluido. Essa parte é denominada "fração de volume". O volume total do domínio estendido de projeto é definido e dado por V_t . Consequentemente, a fração de volume F_v desse domínio estendido é dado por:

$$F_v = \frac{\int_{\Omega} \alpha V d\Omega}{V_t} \quad (2.53)$$

É importante salientar que a fração de volume é um parâmetro da otimização, que deve ser escolhido *a priori*. Além disso, como α é constante para cada um dos elementos da discretização, a integral dada na Equação (2.53) pode ser substituída por um somatório discreto. Portanto, para este processo de otimização, sendo n a quantidade de elementos do domínio, a limitação é dada pela Equação (2.54):

$$\sum_{i=1}^n \alpha_i V_i \leq F_v V_t \quad (2.54)$$

2.4.5 Método das Assíntotas Móveis (MMA)

O Método das Assíntotas Móveis foi desenvolvido por Svanberg (1987) e, portanto, o desenvolvimento aqui apresentado baseia-se nesta referência. Adaptando-o para este projeto, suponha um problema P , definido com a minimização de $\Phi(\alpha)$ (descrito na Equação (2.46)), sujeito à restrição de quantidade de material descrita na Equação (2.54) e ao intervalo de valores para α de $0 \leq \alpha \leq 1$, onde α é o vetor com as variáveis de projeto. Ou seja:

$$\begin{aligned} \text{maximizar: } \Phi_0(\alpha) &= \mathbf{z}^T \mathbf{C} \mathbf{z} \\ \text{sujeito à: } \Phi_1(\alpha) &= \sum_{i=1}^n \alpha_i V_i \end{aligned} \quad (2.55)$$

Onde Φ_0 é a função que se busca minimizar e Φ_1 é a restrição à qual estamos submetidos. O vetor referente a essas duas funções será $\Phi(\alpha)$. A abordagem padrão para resolver esses problemas é:

- Passo 0: Escolha um valor inicial de α , $\alpha^{(0)}$;

- Passo 1: Dado um ponto de iteração $\alpha^{(i)}$, calcule $\Phi(\alpha^{(i)})$ e $\nabla\Phi(\alpha^{(i)})$;
- Passo 2: Gerar um subproblema $P^{(i)}$ através da substituição, em P , de funções Φ que são obtidas através da aproximação de funções explícitas $\Phi^{(i)}$, baseadas nos cálculos obtidos no Passo 1;
- Passo 3: Resolver $P^{(i)}$, fazer com que a solução ótima desse subproblema seja o novo valor de $\alpha^{(i)}$ e retornar ao passo 1.

O processo é interrompido quando algum critério de convergência é atingido. Como esse é o procedimento padrão aplicado na resolução desses problemas, para definir o Método das Assíntotas Móveis, é necessário descrever como as funções $\Phi^{(i)}$ e como o subproblema $P^{(i)}$ deve ser resolvido, uma vez que as funções $\Phi^{(i)}$ foram escolhidas. As equações $\Phi^{(i)}$ são definidas como:

$$\Phi^{(i)}(\alpha) = \mathbf{r}^{(i)} + \sum_{j=1}^n \left(\frac{\mathbf{P}^{(i)}}{\mathbf{b}^{(i)} - \alpha} + \frac{\mathbf{Q}^{(i)}}{\alpha - \mathbf{a}^{(i)}} \right) \quad (2.56)$$

Onde:

$$\mathbf{P}^{(i)} = \begin{cases} \left(\mathbf{b}^{(i)} - \alpha^{(i)} \right)^2 \cdot \nabla\Phi, & \text{se } \nabla\Phi > 0 \\ 0, & \text{se } \nabla\Phi \leq 0 \end{cases} \quad (2.57)$$

$$\mathbf{Q}^{(i)} = \begin{cases} 0, & \text{se } \nabla\Phi \geq 0 \\ -\left(\alpha^{(i)} - \mathbf{a}^{(i)} \right)^2 \cdot \nabla\Phi, & \text{se } \nabla\Phi < 0 \end{cases} \quad (2.58)$$

$$\mathbf{r}^{(i)} = \Phi(\alpha^{(i)}) - \sum_{j=1}^n \left(\frac{\mathbf{P}^{(i)}}{\mathbf{b}^{(i)} - \alpha^{(i)}} + \frac{\mathbf{Q}^{(i)}}{\alpha^{(i)} - \mathbf{a}^{(i)}} \right) \quad (2.59)$$

E $\mathbf{a}^{(i)}$ e $\mathbf{b}^{(i)}$ são parâmetros escolhidos de tal forma que $\mathbf{a}^{(i)} < \alpha^{(i)} < \mathbf{b}^{(i)}$. Eles podem variar entre problemas de otimização ou até mesmo entre iterações do processo de otimização e podem ser considerados como assíntotas, o que permite movê-los, “relaxando” (aumenta a estabilidade) ou “restringindo” (acelera a convergência). Neste trabalho, será utilizado como limite inferior um vetor de zeros e como limite superior um vetor de uns.

Substituindo as Equações (2.56), (2.57), (2.58) e (2.59) na Equação (2.55), gera-se o subproblema $P^{(i)}$. De acordo com Svanberg (1987), esse problema é convexo e um “método dual” pode ser utilizado para resolvê-lo. Mais detalhes a respeito desse passo podem ser encontrados na própria referência, bem como em Fleury (1979) e em Svanberg (1982).

3 METODOLOGIA

Neste capítulo, discute-se a metodologia aplicada ao presente trabalho. Todas as rotinas envolvendo a otimização topológica abordada foram executadas no ambiente virtual WSL (*Windows Subsystem for Linux*).

Antes de começar a otimização é necessário realizar a discretização do domínio, gerando uma malha. Neste trabalho são necessárias duas malhas para cada caso físico, uma para o MVF e outra para o MEF, visto que a malha necessária para o último método precisa ser quadrática. Ambas devem possuir o mesmo número de elementos. O software utilizado para gerar as malhas foi o GMSH, versão 4.13.1.

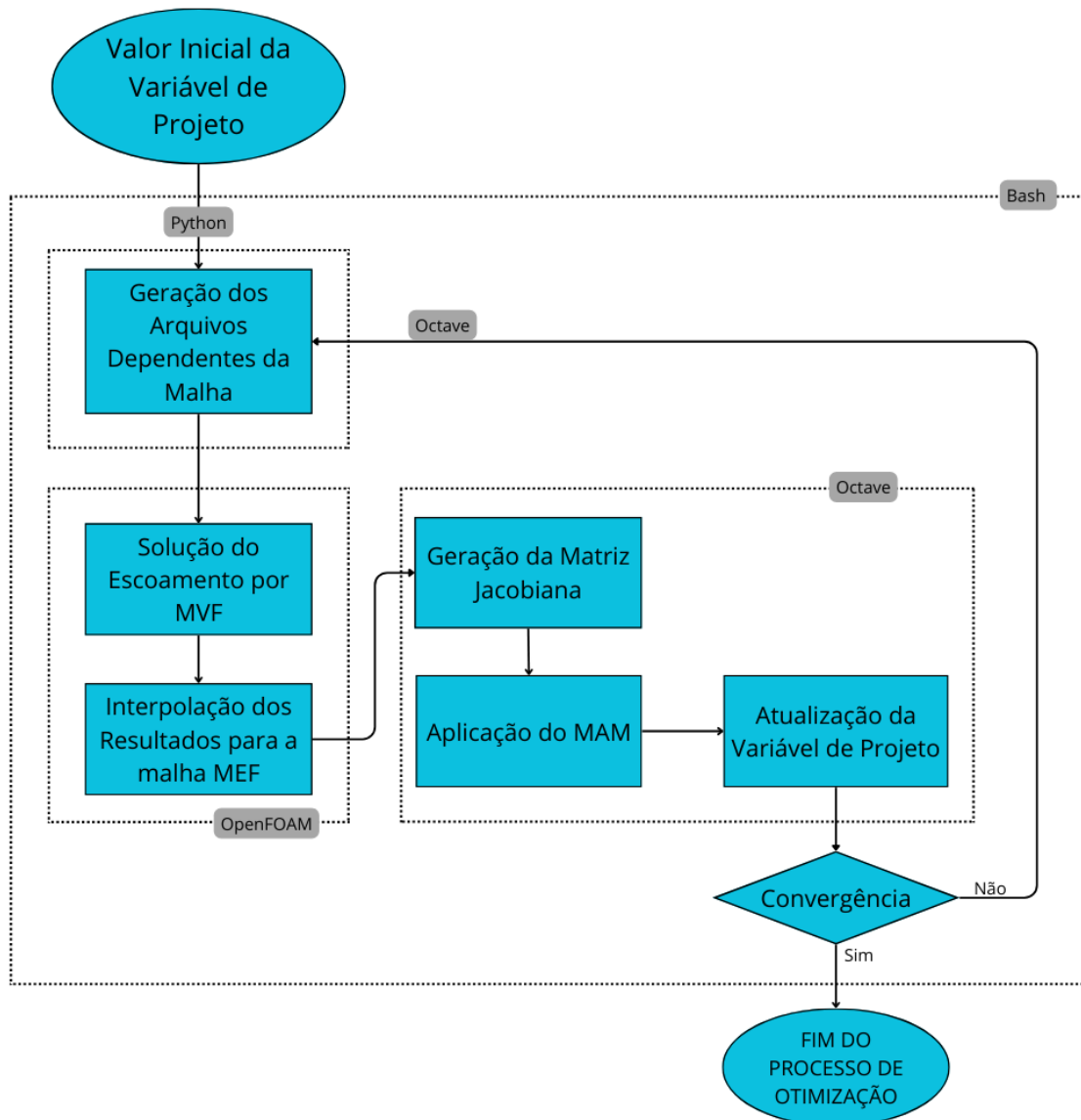
Uma vez discretizado o domínio, se pode começar o processo de otimização, cujo primeiro passo é a criação dos arquivos dependentes da malha. Na primeira iteração eles são gerados por meio do Python, versão 3.10.12. Tais arquivos são: O arquivo que contém os valores de k_p , denominado *kappa* e o arquivo usado na interpolação dos resultados entre as duas malhas, denominado *sampleDict*. Exemplos desses arquivos, bem como a rotina completa para gerá-los, estão dispostos no apêndice A.

Assim que os arquivos são criados, é necessário resolver o escoamento, através do MVF. O software utilizado para tal foi o OpenFOAM®, versão v2012, e o algoritmo utilizado para resolver o escoamento foi um algoritmo baseado no SIMPLE, e as condições de contorno para cada simulação, bem como cada geometria, serão discutidas posteriormente. A interpolação desses valores foi feita também por meio do OpenFOAM, utilizando do arquivo *sampleDict*.

Após a interpolação das velocidades, inicia-se a rotina específica de otimização da geometria, que consiste na geração da matriz jacobiana por meio do MEF e da aplicação da MAM, sendo a última escrita e disponibilizada por Krister Svanberg. As rotinas associadas com essa etapa foram disponibilizadas por Juan Sergio R. Saenz e executadas através do software Octave, versão 7.3.0.

Por fim, executada a rotina de otimização, é necessário atualizar a variável de projeto (e, consequentemente, k_p). O outro arquivo dependente da malha, *sampleDict*, permanece constante por todas as iterações que possam acontecer. A maneira mais fácil de atualizar o arquivo *kappa* neste passo do processo foi por meio software Octave. Por fim, a automatização do processo citado foi feita através de *Bash Scripting*. Um esquemático do processo é dado na Figura (3).

Figura 3 – Fluxograma esquemático das etapas realizadas no processo de otimização



Fonte: O Autor.

Assim como todo experimento físico, os experimentos computacionais também precisam manter uma coerência a fim de serem reproduzidos. Para tal, é necessário se atentar com o valor dos números adimensionais que refletem a física do problema estudado. Neste trabalho, o número adimensional mais importante é o Número de Reynolds, Re , que representa uma relação entre as forças de inércia e as forças viscosas presentes no escoamento, e que será definido aqui pela Equação (3.1):

$$Re = \frac{\rho \cdot u_{max} \cdot L}{\mu} \quad (3.1)$$

Onde ρ é a massa específica do fluido, μ é sua viscosidade, u_{max} é uma velocidade

característica e L é um comprimento característico. Essas variáveis características podem variar entre autores e, neste trabalho, a velocidade característica será considerada a velocidade máxima do perfil parabólico na entrada do domínio e o comprimento característico será considerado o tamanho de tal entrada. Para as otimizações em regime laminar e para o bocal em regime turbulento, o perfil parabólico na entrada será dado pela Equação (3.2):

$$\frac{u(y)}{u_{max}} = \left(4 \cdot \left(\frac{y - y_0}{h} \right) \cdot \left(1 - \left(\frac{y - y_0}{h} \right) \right) \right) \quad (3.2)$$

Com y_0 sendo a menor distância da origem até a entrada e h sendo o tamanho dessa entrada (O perfil parabólico clássico de escoamentos laminares foi utilizado no bocal turbulento pois o trabalho de referência assim o fez). Para escoamentos turbulentos, o perfil parabólico tende a ser um pouco mais achatado. Nesse caso, para as otimizações restantes em regime turbulento, será utilizada a lei da potência de $1/7$, que de acordo com o trabalho de Chant (2005), providenciava e ainda providencia uma relação simples e efetiva para tais perfis de velocidade com gradientes de pressão moderados. Assim sendo, tal perfil é dado pela Equação (3.3).

$$\frac{u(y)}{u_{max}} = \left(4 \cdot \left(\frac{y - y_0}{h} \right) \cdot \left(1 - \left(\frac{y - y_0}{h} \right) \right) \right)^{(1/7)} \quad (3.3)$$

Além disso, assim como na maior parte dos trabalhos realizados na área, será utilizada uma penalização variável a fim de reduzir possíveis escalas de cinza que possam aparecer e dar mais nitidez à imagem. Neste trabalho, a penalização será alterada através de uma progressão geométrica, de tal forma que, após q_n iterações, o novo valor de q será dado por:

$$q_i = (1 + q_{inc}) \cdot q_{i-1} \quad (3.4)$$

Sendo que q_i é o valor de q no novo ciclo de iterações, q_{i-1} o seu valor no ciclo anterior e q_{inc} é o incremento que será feito. Além disso, neste trabalho, optou-se por não utilizar filtros, a fim de evitar a interferência deles com a eficiência da ferramenta desenvolvida.

Na Tabela (1) estão descritos os parâmetros físicos utilizados nos experimentos computacionais realizados.

Tabela 1 – Parâmetros físicos dos experimentos computacionais realizados.

Geometria	Regime	Vel (m/s)	$L^{(1)}$ (m)	$L^{(2)}$ (m)	μ/ρ (m^2/s)	Re	F_v
Bocal	Laminar	1,0	1,0	1,0	1,0	1,0	0,5
	Turbulento	3,0	1,0	1,0	$1 \cdot 10^{-3}$	3000	0,3
Curva T	Laminar	0,5	0,2	1,0	0,1	1,0	0,5
	Turbulento	1,0	0,2	1,0	$4 \cdot 10^{-5}$	5000	0,5
Joelho	Laminar	3,0	1,0	5,0	0,01	300	0,25
	Turbulento	1,0	0,2	1,0	$4 \cdot 10^{-5}$	5000	0,2513

⁽¹⁾ valor referente à Equação (3.1)

⁽²⁾ valor referente à geometria apresentada na Figura (4)

Fonte: O Autor.

A Tabela (2) apresenta os parâmetros computacionais utilizados nas simulações realizadas. A coluna “ q_{low} ” representa o valor inicial da penalização, a coluna “Iter.” refere-se à quantidade total de iterações do processo de otimização, e a coluna “Tempo” refere-se ao tempo total de tal processo.

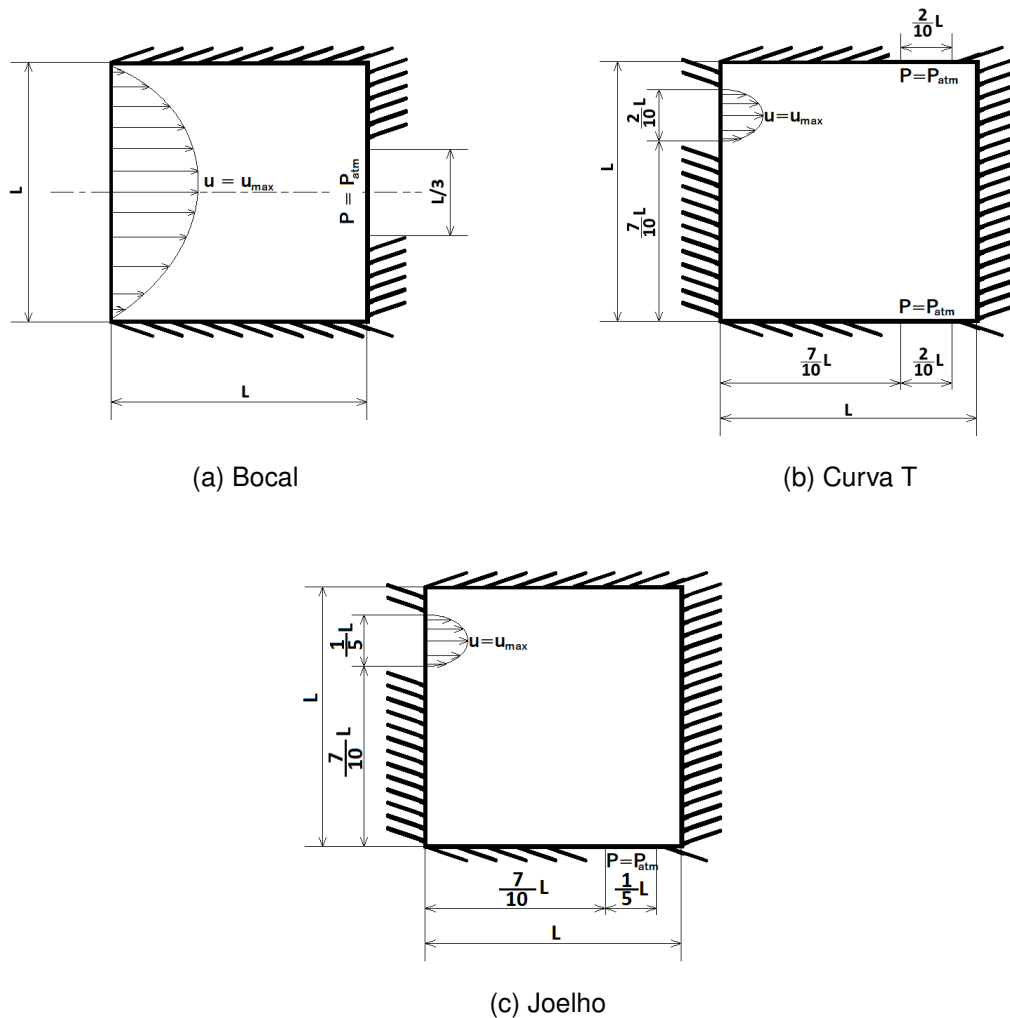
Tabela 2 – Aspectos Computacionais das Simulações

Geometria	Regime	Malha	q_{inc}	q_{low}	q_n	Iter.	Tempo (min)
Bocal	Laminar	Grossa (30x30)	13	$1 \cdot 10^{-6}$	10	60	19
		Fina (51x51)	5	$1 \cdot 10^{-3}$	9	24	50
	Turbulento	Grossa (30x30)	20	$1 \cdot 10^{-6}$	15	60	52
		Fina (51x51)	0,75	$2,5 \cdot 10^{-4}$	4	32	80
Curva T	Laminar	Grossa (30x30)	20	$1 \cdot 10^{-6}$	15	60	15
		Fina (50x50)	2	$1 \cdot 10^{-3}$	12	32	32
	Turbulento	Grossa (30x30)	0,5	$5 \cdot 10^{-7}$	6	60	26
		Fina (50x50)	0,5	$1 \cdot 10^{-5}$	8	32	70
		Finíssima (80x80)	0,5	$5 \cdot 10^{-5}$	8	8	67
Joelho	Laminar	Grossa (30x30)	2	$2,5 \cdot 10^{-6}$	12	40	15
		Fina (50x50)	2	$5 \cdot 10^{-5}$	9	12	19
	Turbulento	Grossa (30x30)	1,05	$1,25 \cdot 10^{-6}$	12	60	21
		Fina (50x50)	0,75	$1 \cdot 10^{-5}$	8	32	70

Fonte: O Autor.

Na Figura (4), serão apresentados os esquemáticos dos experimentos computacionais realizados. Para o leitor interessado em replicar os experimentos, os arquivos de texto utilizados pelo GMSH para criar as malhas estão presentes no Apêndice B e arquivos de texto contendo exemplos das condições de contorno de velocidade, pressão e viscosidade turbulenta, utilizados pelo OpenFOAM®, estão contidos no Apêndice C. Os arquivos que controlam a simulação por MVF feita através desse software estão presente no Apêndice D. Tais arquivos foram adaptados de uma versão do modelo de turbulência de Wray-Agarwal disponível online. No Apêndice E apresenta-se a rotina de automatização escrita em Bash Script. Optou-se por não incluir as rotinas do Octave visto que são muito numerosas.

Figura 4 – Geometrias e Condições de Contorno utilizadas.



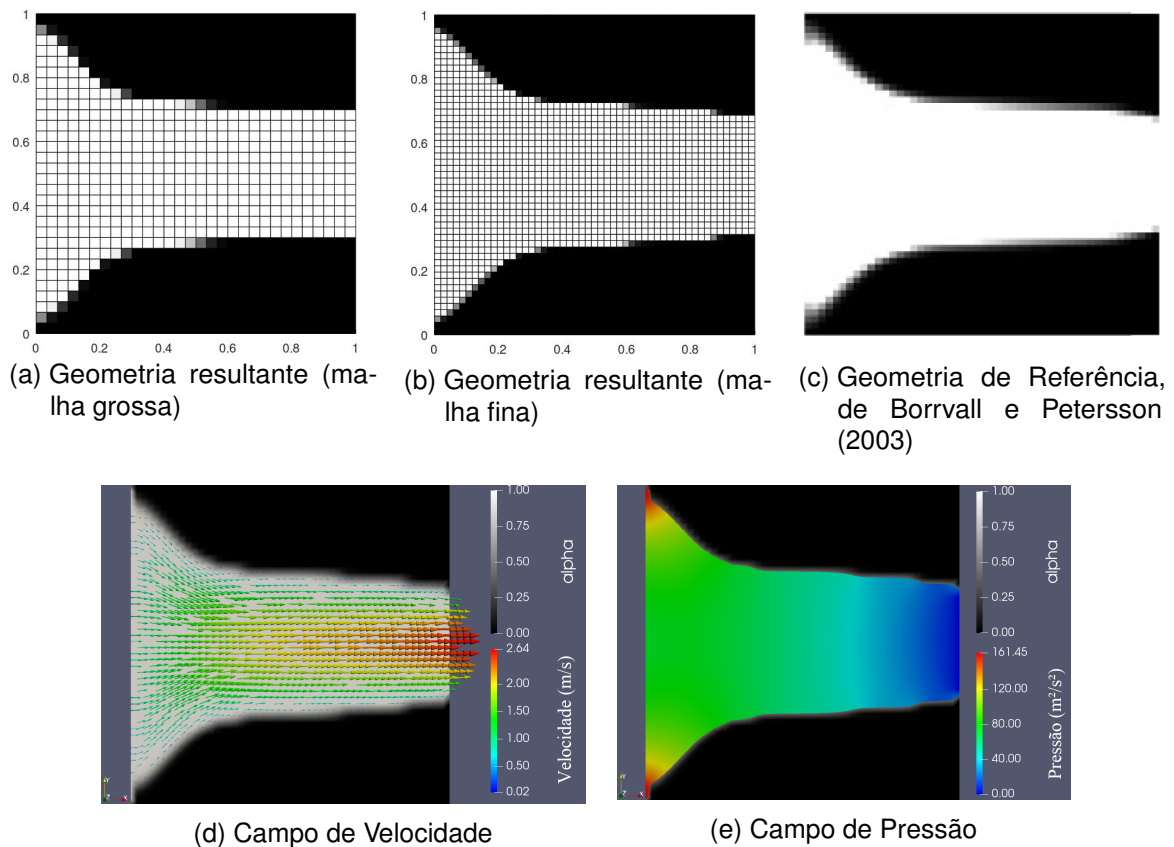
Fonte: O Autor.

4 RESULTADOS

Neste capítulo encontram-se os resultados obtidos das otimizações dos dispositivos descritos no capítulo 3 através da combinação da Otimização Topológica com o Método dos Volumes Finitos. Os campos apresentados neste capítulo foram gerados na geometria mais refinada e os campos de pressão apresentam a variável dividida pela massa específica do fluido, uma pressão específica, e por esse motivo as unidades apresentadas não são as usuais de pressão.

A Figura 5 apresenta os resultados obtidos ao otimizar a geometria de um bocal em regime laminar a fim de reduzir sua perda de carga. Percebe-se que a topologia resultante da presente otimização é, qualitativamente, idêntica à geometria de referência.

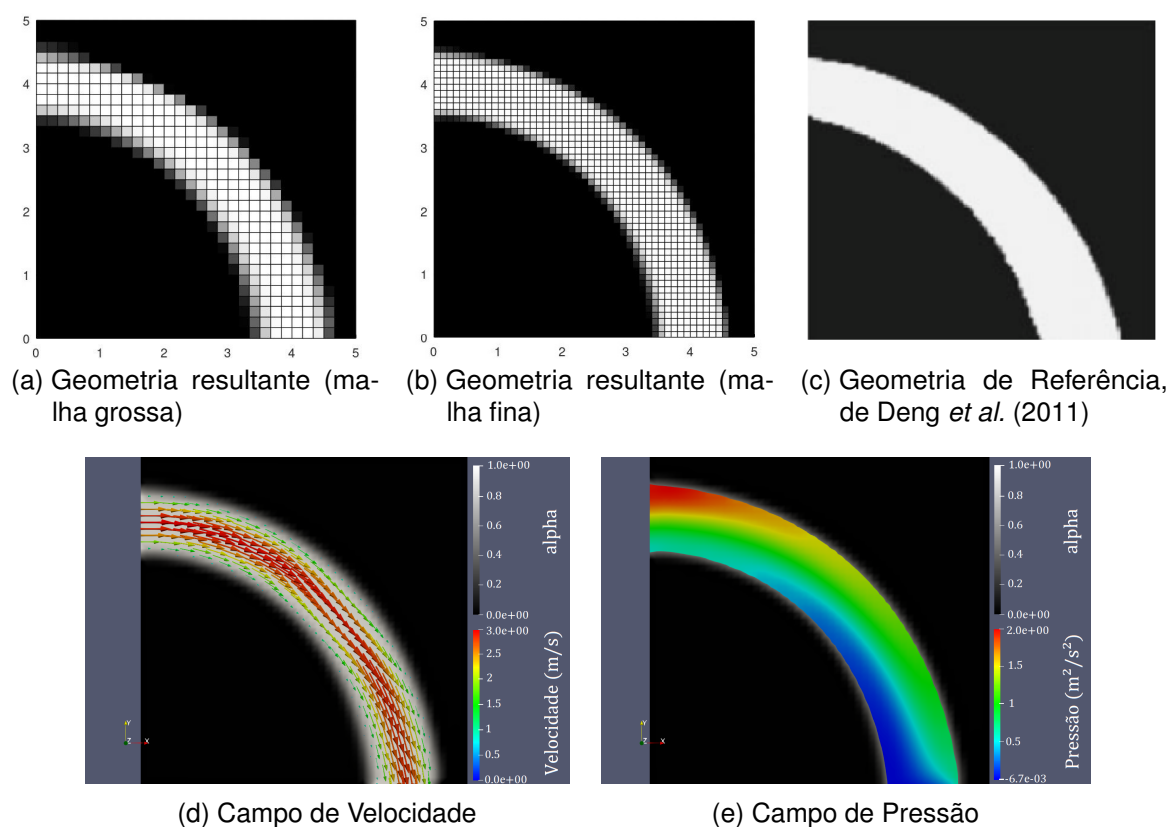
Figura 5 – Resultados da Otimização Topológica de um bocal em regime laminar.



Fonte: O Autor.

A Figura 6 apresenta os resultados obtidos ao otimizar a geometria de um joelho em regime laminar a fim de reduzir a perda de carga neste dispositivo. Assim como a otimização do bocal, a otimização do joelho também é, qualitativamente, idêntica à geometria de referência.

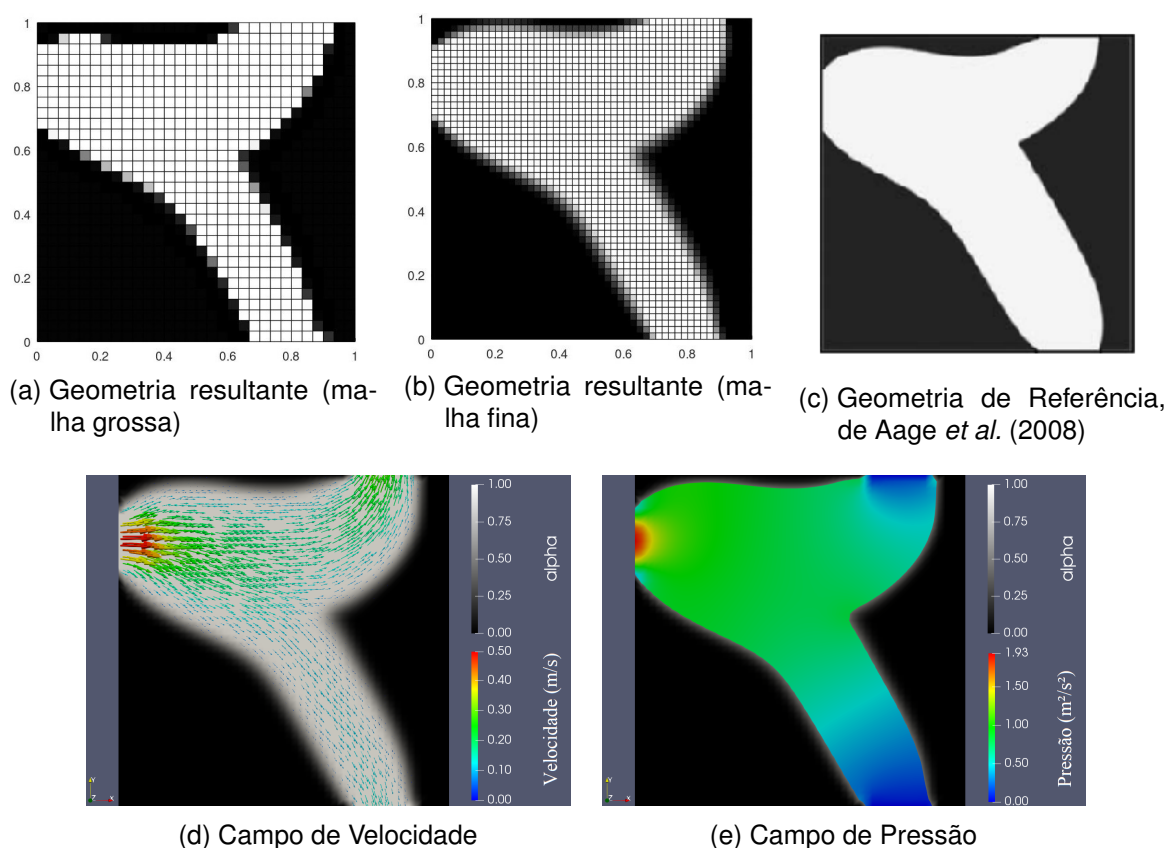
Figura 6 – Resultados da Otimização Topológica de um joelho em regime laminar.



Fonte: O Autor.

A Figura 7 apresenta os resultados obtidos ao otimizar a geometria de uma curva em T em regime laminar a fim de reduzir a perda de carga neste dispositivo. Embora exista uma pequena diferença na altura da "quina" encontrada através da otimização, a topologia encontrada é extremamente similar à topologia de referência. Essa pequena diferença pode ser facilmente justificada devido à diferença nas funções objetivo utilizadas, visto que a função objetivo do trabalho de referência também buscava um fluxo específico nas saídas e, portanto, é possível (e esperado) ver na geometria obtida neste trabalho um fluxo muito maior na saída superior do que na saída inferior.

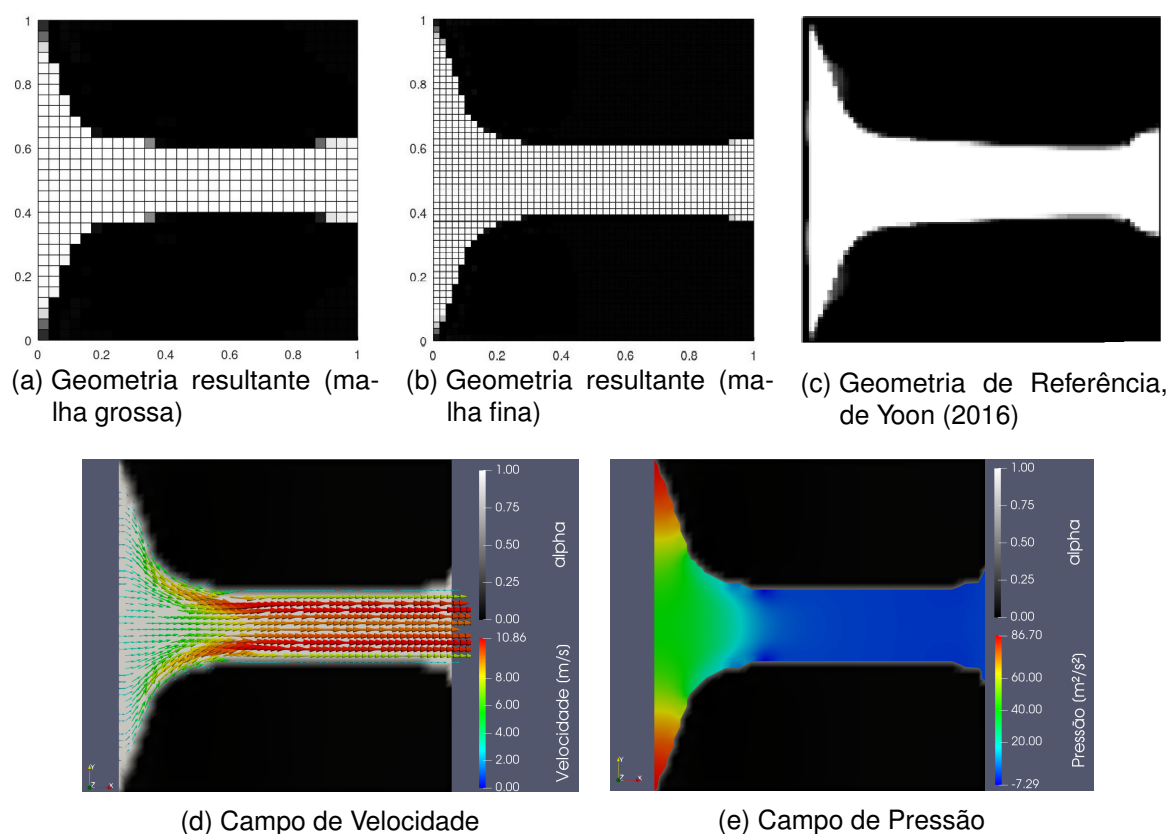
Figura 7 – Resultados da Otimização Topológica de uma curva T em regime laminar.



Fonte: O Autor.

A Figura 8 apresenta os resultados obtidos ao otimizar a geometria de um bocal em regime turbulento a fim de reduzir a perda de carga no mesmo. Percebe-se que a topologia resultante da presente otimização é, qualitativamente, idêntica à geometria de referência.

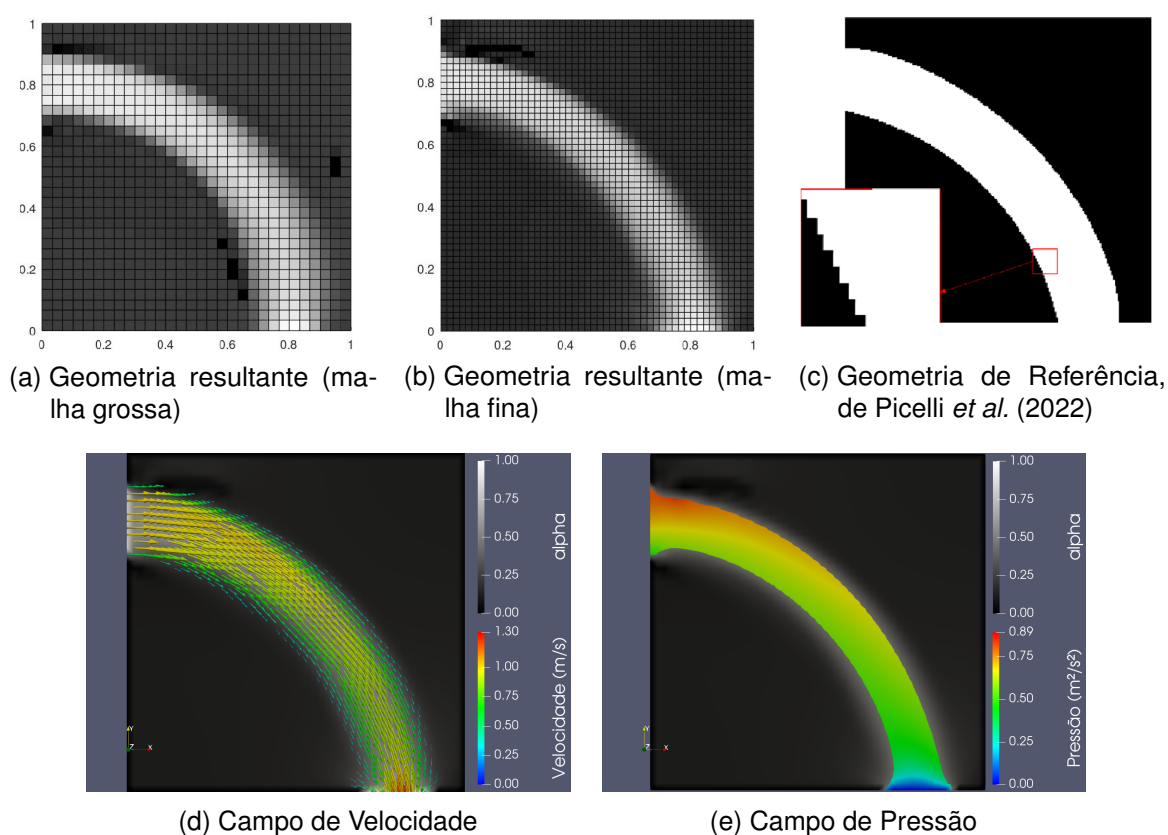
Figura 8 – Resultados da Otimização Topológica de um bocal em regime turbulento.



Fonte: O Autor.

A Figura 9 apresenta os resultados obtidos ao otimizar a geometria de um Joelho em regime turbulento a fim de reduzir a perda de carga no mesmo. Percebe-se que a topologia resultante da presente otimização é, qualitativamente, idêntica à geometria de referência, embora ela se torne mais semelhante conforme a malha é mais refinada. Também é possível perceber falta de nitidez na imagem, o que aponta para a necessidade da utilização de filtros. Como a nitidez dos resultados melhora com o refino da malha, essa pode ser uma boa estratégia caso haja poder computacional suficiente (a malha utilizada no trabalho de referência é uma malha quadrada 200x200).

Figura 9 – Resultados da Otimização Topológica de um Joelho em regime turbulento.

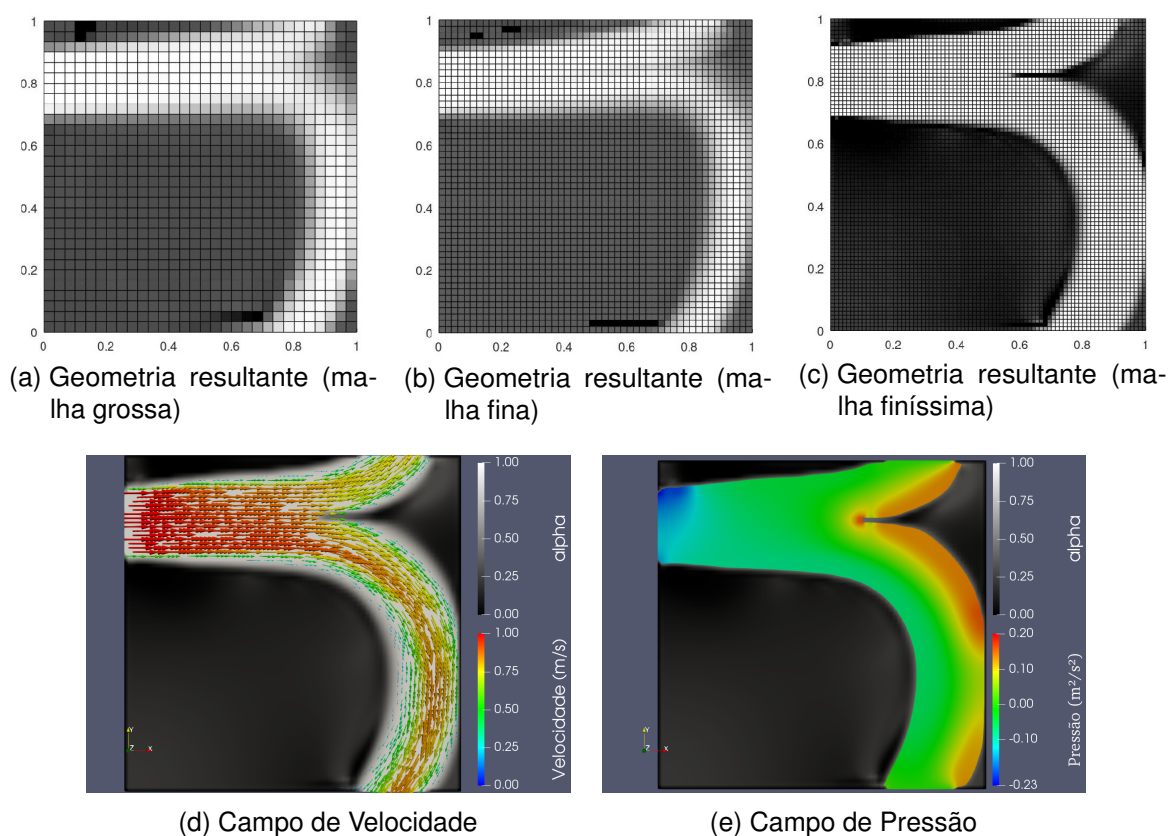


Fonte: O Autor.

A Figura 10 apresenta os resultados obtidos ao otimizar a geometria de uma curva T em regime turbulento a fim de reduzir a perda de carga no mesmo. Até o presente momento não foi encontrada uma referência para este caso em regime turbulento. Assim como na otimização de um joelho turbulento, neste caso também é possível perceber falta de nitidez na imagem (ao menos em geometrias mais grosseiras), o que aponta para a necessidade da utilização de filtros e/ou uma maior discretização do domínio.

Um ponto interessante de se notar nessa otimização é que ao contrário do que aconteceu com o caso laminar, aqui há um fluxo similar de fluido em ambas as saídas, mesmo sem impor essa condição na função objetivo.

Figura 10 – Resultados da Otimização Topológica de uma curva T em regime turbulento.



Fonte: O Autor.

5 CONSIDERAÇÕES FINAIS

Com base nos resultados obtidos é possível afirmar que a ferramenta desenvolvida neste trabalho atingiu os objetivos propostos adequadamente, com uma malha igual ou menor do que as utilizadas nas referências. Além disso, as otimizações em regimes turbulentos geraram geometrias com uma perda de carga menor do que a gerada pelas otimizações em regimes laminares, com reduções de 44% chegando até 77%.

Para trabalhos futuros, uma análise mais adequada das penalizações se faz necessária, ou até mesmo um maior desenvolvimento da ferramenta criada por Lima, Aguiar e Romero (2024) e refinada neste trabalho, a fim de reduzir a dependência da penalização. A utilização de filtros mais refinados pode ser feita a fim de reduzir as escalas de cinza de maneira menos agressiva. Filtros de densidade são comuns na literatura na otimização de estruturas e podem ser utilizados em trabalhos futuros.

Também é interessante o estudo da aplicação de aceleradores de convergência para o processo de otimização, visto que para malhas relativamente pequenas os problemas de Otimização Topológica costumam ser muito demorados, como é possível de ver na Tabela 2, onde uma interação do processo de otimização chegou a demorar mais de 8 minutos (curva T em regime turbulento, na malha finíssima).

Por fim, para trabalhos futuros, há os seguintes pontos de melhoria/ inovação:

- Otimização de geometrias tridimensionais. Embora boa parte dos problemas tridimensionais possam ser simplificados por problemas bidimensionais, para uma gama de problemas é inviável. Este ponto já é presente na literatura, mas tal consideração neste trabalho requer uma pequena adaptação ao código gerado ;
- Funções multiobjetivos que, embora sejam, comuns na literatura, não foram abordadas neste trabalho e requerem mais adaptações a este código. Em especial, a minimização da concentração de espécie química a fim de minimizar a concentração de dióxido de carbono e aumentar o conforto térmico em ambientes;
- Utilização das geometrias utilizadas por Otimização Topológica como palpites iniciais a serem utilizados pela Otimização de Forma, visto que esta é mais refinada do que aquela, com aplicações principalmente na indústria aeronáutica onde pequenas variações podem resultar em uma grande economia de combustível;
- Análise da influência do Termo de Forchheimer, desprezado no equacionamento feito na Seção 2.1, na Otimização Topológica em escoamentos turbulentos, visto que neles tal termo costuma sobrepujar a influência do Termo de Darcy.

REFERÊNCIAS

- AAGE, N.; POULSEN, T. H.; GERSBORG-HANSEN, A.; SIGMUND, O. Topology optimization of large scale stokes flow problems. *Structural and Multidisciplinary Optimization*, Springer, v. 35, p. 175–180, 2008. Citado na página 39.
- ALONSO, D. H.; SAENZ, J. S. R.; PICELLI, R.; SILVA, E. C. N. Topology optimization method based on the wray–agarwal turbulence model. *Structural and Multidisciplinary Optimization*, Springer, v. 65, n. 3, p. 82, 2022. Citado 2 vezes nas páginas 15 e 19.
- BAE, Y.; KIM, Y. I. Numerical modeling of anisotropic drag for a perforated plate with cylindrical holes. *Chemical Engineering Science*, Elsevier, v. 149, p. 78–87, 2016. Citado na página 16.
- BENDSOE, M. P. Optimal shape design as a material distribution problem. *Structural optimization*, Springer, v. 1, p. 193–202, 1989. Citado na página 26.
- BENDSØE, M. P.; KIKUCHI, N. Generating optimal topologies in structural design using a homogenization method. *Computer methods in applied mechanics and engineering*, Elsevier, v. 71, n. 2, p. 197–224, 1988. Citado na página 14.
- BORRVALL, T.; PETERSSON, J. Topology optimization of fluids in stokes flow. *International journal for numerical methods in fluids*, Wiley Online Library, v. 41, n. 1, p. 77–107, 2003. Citado 4 vezes nas páginas 14, 26, 27 e 37.
- CHANT, L. J. D. The venerable 1/7th power law turbulent velocity profile: a classical nonlinear boundary value problem solution and its relationship to stochastic processes. *Applied Mathematics and Computation*, Elsevier BV, v. 161, n. 2, p. 463–474, fev. 2005. ISSN 0096-3003. Disponível em: <<http://dx.doi.org/10.1016/j.amc.2003.12.109>>. Citado na página 33.
- DARWISH, M.; MOUKALLED, F. *The finite volume method in computational fluid dynamics: an advanced introduction with OpenFOAM® and Matlab®*. [S.l.]: Springer, 2016. Citado na página 21.
- DENG, Y.; LIU, Z.; ZHANG, P.; LIU, Y.; WU, Y. Topology optimization of unsteady incompressible navier–stokes flows. *Journal of Computational Physics*, Elsevier, v. 230, n. 17, p. 6688–6708, 2011. Citado na página 38.
- DILGEN, C. B.; DILGEN, S. B.; FUHRMAN, D. R.; SIGMUND, O.; LAZAROV, B. S. Topology optimization of turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, Elsevier, v. 331, p. 363–393, 2018. Citado na página 28.
- FLEURY, C. Structural weight optimization by dual methods of convex programming. *International Journal for Numerical Methods in Engineering*, Wiley Online Library, v. 14, n. 12, p. 1761–1783, 1979. Citado na página 30.
- KOGA, A. A. *Projeto de dispositivos de microcanais utilizando o método de otimização topológica*. Tese (Doutorado) — Universidade de São Paulo, 2010. Citado 4 vezes nas páginas 14, 25, 26 e 27.

- LEE, S.-H.; IH, J.-G. Empirical model of the acoustic impedance of a circular orifice in grazing mean flow. *The Journal of the Acoustical Society of America*, Acoustical Society of America, v. 114, n. 1, p. 98–113, 2003. Citado na página 16.
- LIMA, C. P. Picoli de; AGUIAR, U.; ROMERO, J. S. Topological optimization based on the finite volume method in conjunction with the finite element method. In: *Proceedings of the 20th Brazilian Congress of Thermal Sciences and Engineering*. ABCM, 2024. (CIT24). Disponível em: <<http://dx.doi.org/10.26678/ABCM.ENCIT2024.CIT24-0193>>. Citado 4 vezes nas páginas 8, 9, 15 e 43.
- MORTENSEN, M.; LANGTANGEN, H. P.; WELLS, G. N. A fenics-based programming framework for modeling turbulent flow by the reynolds-averaged navier–stokes equations. *Advances in Water Resources*, Elsevier, v. 34, n. 9, p. 1082–1101, 2011. Citado na página 15.
- MULLER, J. Radical reduction of aircraft fuel consumption by optimizing aerofoil by new evolutionary algorithms. *Preprints*, Preprints, June 2021. Disponível em: <<https://doi.org/10.20944/preprints202106.0219.v1>>. Citado 3 vezes nas páginas 8, 9 e 14.
- PICELLI, R.; MOSCATELLI, E.; YAMABE, P. V. M.; ALONSO, D. H.; RANJBARZADEH, S.; GIORIA, R. dos S.; MENEGHINI, J. R.; SILVA, E. C. N. Topology optimization of turbulent fluid flow via the tobs method and a geometry trimming procedure. *Structural and Multidisciplinary Optimization*, Springer, v. 65, n. 1, p. 34, 2022. Citado na página 41.
- REDDY, J. N.; GARTLING, D. K. *The finite element method in heat transfer and fluid dynamics*. [S.l.]: CRC press, 2010. Citado na página 22.
- ROMERO, J.; SILVA, E. C. N. A topology optimization approach applied to laminar flow machine rotor design. *Computer Methods in Applied Mechanics and Engineering*, Elsevier, v. 279, p. 268–300, 2014. Citado na página 22.
- SHAHZAD, H.; HICKEL, S.; MODESTI, D. Permeability and turbulence over perforated plates. *Flow, Turbulence and Combustion*, Springer, v. 109, n. 4, p. 1241–1254, 2022. Citado na página 17.
- SVANBERG, K. An algorithm for optimum structural design using duality. *Applications*, Springer, p. 161–177, 1982. Citado na página 30.
- SVANBERG, K. The method of moving asymptotes—a new method for structural optimization. *International journal for numerical methods in engineering*, Wiley Online Library, v. 24, n. 2, p. 359–373, 1987. Citado 3 vezes nas páginas 15, 29 e 30.
- TANNER, P.; GORMAN, J.; SPARROW, E. Flow–pressure drop characteristics of perforated plates. *International Journal of Numerical Methods for Heat & Fluid Flow*, Emerald Publishing Limited, v. 29, n. 11, p. 4310–4333, 2019. Citado na página 16.
- WRAY, T. J.; AGARWAL, R. K. Low-reynolds-number one-equation turbulence model based on k-omega closure. *AIAA Journal*, v. 53, n. 8, p. 2216–2227, 2015. Citado 4 vezes nas páginas 15, 19, 20 e 60.

XIE, L.; LI, H.; ZHANG, Y.; LIU, X.; ZHAO, Y. Topology optimization and numerical analysis of cold plates for concentrating photovoltaic thermal management. *Case Studies in Thermal Engineering*, v. 52, p. 103713, 2023. ISSN 2214-157X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2214157X23010195>>. Citado na página 14.

YOON, G. H. Topology optimization for turbulent flow with spalart–allmaras model. *Computer Methods in Applied Mechanics and Engineering*, Elsevier, v. 303, p. 288–311, 2016. Citado 2 vezes nas páginas 14 e 40.

ZHENG, S.; GAO, F.; ZHANG, Z.; LIU, H.; LI, B. Topology optimization on fuel tank rib structures for fuel sloshing suppression based on hybrid fluid–solid sph simulation. *Thin-Walled Structures*, v. 165, p. 107938, 2021. ISSN 0263-8231. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0263823121003098>>. Citado 3 vezes nas páginas 8, 9 e 14.

APÊNDICE A – ROTINAS DE GERAÇÃO DOS ARQUIVOS DEPENDENTES DA MALHA

Neste apêndice, detalha-se as rotinas de criação dos arquivos que dependem da discretização utilizada: *sampleDict* e *kappa*.

A.1 *kappa*

O arquivo *kappa* é gerado em através de 2 softwares diferentes: Python e Octave. A rotina desenvolvida em Python, cujo parâmetro é um vetor contendo os valores de *kappa* é:

```
def generatekappa(list):

    f = open("kappa", "w")
    with open("kappa", "w") as w:
        w.write(r"""/*-----*-- C++
        ↪ -----*\
        ===== |
        \ \      / F ield      | OpenFOAM: The Open Source CFD Toolbox
        \ \      / O peration   | Website: https://openfoam.org
        \ \      / A nd          | Version: dev
        \ \ /      M anipulation |
        \*-----*/

FoamFile
{
    version      2.0;
    format       ascii;
    class        volScalarField;
    object       kappa;
}
// * * * * *

kInlet          0;

dimensions      [0 0 -1 0 0 0 0];

internalField    nonuniform List<scalar>""")

        cells=len(list)

        w.write("\n"+str(cells)+"\n(")

        for i in range(cells):
            w.write("\n"+str(list[i]))

        w.write(r"""
);
boundaryField
{
    inlet
```

```

{
    type            fixedValue;
    value           uniform 0;
}

outlet
{
    type            fixedValue;
    value           uniform 0;
}

fixedWalls
{
    type            fixedValue;
    value           uniform 10000;
}

frontAndBack
{
    type            empty;
}
}

// ***** //""")
f.close()

```

Já a rotina utilizada pelo Octave para geração desse arquivo é dada por:

```

function write_kappa(att_kappa) %writes new kappa file for FVM simulation
%=====
%Creating the text of the kappa file%
%-----%Header%-----%
i=1;
text(i) = {'/*-----*- C++
    ↪ ↪ -*-----*'};
i++;
text(i) = {' ===== | '};
i++;
text(i) = {' \\ / F i e l d | OpenFOAM: The Open Source CFD Toolbox '};
i++;
text(i) = {' \\ / O p e r a t i o n | Website: https://openfoam.org '};
i++;
text(i) = {' \\ / A n d | Version: dev '};
i++;
text(i) = {' \\ / M a n i p u l a t i o n | '};
i++;
text(i) =
    ↪ {'/*-----*'};
    ↪
i++;
text(i) = {'FoamFile '};
i++;
text(i) = {'{ '};
i++;
text(i) = {' version 2.0; '};
i++;
text(i) = {' format ascii; '};

```



```

i++;
text(i) = { '      } '};
i++;
text(i) = { ' '};
i++;
text(i) = { '      fixedWalls '};
i++;
text(i) = { '      { '};
i++;
text(i) = { '          type          fixedValue ; '};
i++;
text(i) = { '          value          uniform 10000; '};
i++;
text(i) = { '      } '};
i++;
text(i) = { ' '};
i++;
text(i) = { '      frontAndBack '};
i++;
text(i) = { '      { '};
i++;
text(i) = { '          type          empty ; '};
i++;
text(i) = { '      } '};
i++;
text(i) = { ' '};
i++;
text(i) = { ' '};
i++;
text(i) = { ' '};
i++;
text(i) = { ' // *****
    ↪ // '};
i++;

%-----%Formatting%-----%

text(:) = strcat(text(:), "\n"); %adding new line to all lines
[rows, lines] = size (text);
chars = length(char(text(lines)));
text(lines) = char(text(lines))(1:chars-1); %removing new line from the last line
%-----%

                %Writing kappa file%
fid = fopen("kappa", "w");
for i = 1:lines
    fputs(fid, char(text(i)));
endfor

fclose(fid);
%=====
endfunction

```

A.2 *sampleDict*

O arquivo *sampleDict* é gerado por meio do seguinte código, escrito em Python, cujo argumento é a lista de nós presente no arquivo da malha MEF gerada pelo GMSH:

```

def generatesample(list):

    lines = len(list)

    f = open("sampleDict","w")

    with open("sampleDict", "w") as w:
        w.write(r"""/*-----*- C++
                ↪ -----*\
| =====|
|  \ \    /  F ield      | OpenFOAM: The Open Source CFD Toolbox |
|  \ \    /  O peration  | Version: v2012                        |
|   \ \    /   A nd      | Website: www.openfoam.com             |
|    \ \    /  M anipulation |                                   |
\*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "system";
    object       sampleDict;
}
// * * * * *

type sets;

interpolationScheme cellPointFace;

setFormat      raw;

sets
(
    data
    {
        type cloud;
        axis xyz;
        ordered no;
        points (
""")

        for i in range(lines):
            w.write(str(list[i]))
        w.write(r""");

    }
);

fields          (U p);

// ***** //""")

    f.close()

```

APÊNDICE B – MALHAS UTILIZADAS

Neste apêndice encontram-se os arquivos de texto utilizados pelo GMSH para criar as malhas estruturais utilizadas neste trabalho. Como exemplificado no capítulo 3, cada simulação utiliza uma malha para o MEF e uma malha para o MVF e, consequentemente, é assim que este apêndice será dividido.

B.1 Malhas - MVF

Nesta seção deste apêndice estão descreve-se os arquivos que criaram as malhas utilizadas pelo software OpenFOAM® para resolver o escoamento.

B.1.1 Bocal

```
//Gmsh project created on Mon Aug 28 17:41:47 2023
SetFactory ("OpenCASCADE");
L=1;
k = 10;           //number of cells on the exit
n=3*k;           //number of cells on y axis
m = n;           //number of cells on x axis
hz=0.1*(L/n);
//+
Point(1) = {0, 0, 0, 1.0};
//+
Point(2) = {L, 0, 0, 1.0};
//+
Point(3) = {L, L/3, 0, 1.0};
//+
Point(4) = {L, 2*L/3, 0, 1.0};
//+
Point(5) = {L, L, 0, 1.0};
//+
Point(6) = {0, L, 0, 1.0};
//
Line(1) = {1, 2};
//+
Line(2) = {2, 3};
//+
Line(3) = {3, 4};
//+
Line(4) = {4, 5};
//+
Line(5) = {5, 6};
//+
Line(6) = {6, 1};
//+
Curve Loop(1) = {1, 2, 3, 4, 5, 6};
//+
Plane Surface(1) = {1};
//+
Transfinite Surface {1} = {1, 2, 5, 6};
//+
Transfinite Curve {2, 3, 4} = (k+1) Using Progression 1;
```

```
//+
Transfinite Curve {6} = (n+1) Using Progression 1;
//+
Transfinite Curve {5, 1} = (m+1) Using Progression 1;
//+
Recombine Surface {1};
//+
Extrude {0, 0, hz} {
  Surface{1}; Layers {1}; Recombine;
}
//+
Physical Surface("inlet", 19) = {7};
//+
Physical Surface("outlet", 20) = {4};
//+
Physical Surface("fixedWalls", 21) = {3, 2, 5, 6};
//+
Physical Surface("frontAndBack", 22) = {8, 1};
//+
Physical Volume("internal", 23) = {1};
```

B.1.2 Curva T

```
// Gmsh project created on Mon Aug 28 17:41:47 2023
SetFactory("OpenCASCADE");
L=1;
d = 4*(L/5); //distance from center of opening to origin
h = L/5; //size of the openings

lambda = 3; //discretization, any positive integer bigger than 0
k = 2*lambda; //number of cells on the exit
q = (7/2)*k; //number of cells on the biggest wall
r = (1/2)*k; //number of cells on the smallest wall
n=(k+q+r); //total number of cells

hz=0.1*(L/n);

//+
Point(1) = {0, 0, 0, 1.0};
//+
Point(2) = {(d-(h/2)), 0, 0, 1.0};
//+
Point(3) = {(d+(h/2)), 0, 0, 1.0};
//+
Point(4) = {L, 0, 0, 1.0};
//+
Point(5) = {L, L, 0, 1.0};
//+
Point(6) = {(d+(h/2)), L, 0, 1.0};
//
Point(7) = {(d-(h/2)), L, 0, 1.0};
//
Point(8) = {0, L, 0, 1.0};
//
Point(9) = {0, (d+(h/2)), 0, 1.0};
//
Point(10) = {0, (d-(h/2)), 0, 1.0};
//
Line(1) = {1, 2};
```

```

//+
Line(2) = {2, 3};
//+
Line(3) = {3, 4};
//+
Line(4) = {4, 5};
//+
Line(5) = {5, 6};
//+
Line(6) = {6, 7};
//+
Line(7) = {7, 8};
//+
Line(8) = {8, 9};
//+
Line(9) = {9, 10};
//+
Line(10) = {10, 1};
//+
Curve Loop(1) = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
//+
Plane Surface(1) = {1};
//+
Transfinite Surface {1} = {1, 4, 5, 8};
//+
Transfinite Curve {2, 6, 9} = (k+1) Using Progression 1;
//+
Transfinite Curve {1, 7, 10} = (q+1) Using Progression 1;
//+
Transfinite Curve {3, 5, 8} = (r+1) Using Progression 1;
//+
Transfinite Curve {4} = (n+1) Using Progression 1;
//+
Recombine Surface {1};
//+
Extrude {0, 0, hz} {
    Surface{1}; Layers {1}; Recombine;
}
//+
Physical Surface("inlet", 19) = {10};
//+
Physical Surface("outlet", 20) = {3, 7};
//+
Physical Surface("fixedWalls", 21) = {2, 4, 5, 6, 8, 9, 11};
//+
Physical Surface("frontAndBack", 22) = {12, 1};
//+
Physical Volume("internal", 23) = {1};

```

B.1.3 Joelho

```

// Gmsh project created on Mon Aug 28 17:41:47 2023
SetFactory("OpenCASCADE");
L=1;
d = 0.7*L;      //distance of the exit from the origin
h = 0.2*L;      //size of the exit

lambda = 5;     //discretization

```



```

k = 2*lambda;    //number of cells on the exit
r = (1/2)*k;     //number of cells on the line after the exit
q = (7/2)*k;     //number of cells on the line before the exit
n=(k+r+q);       //number of cells on y axis (10*lambda)

hz=0.1*(L/n);
//+
Point(1) = {0, 0, 0, 1.0};
//+
Point(2) = {d, 0, 0, 1.0};
//+
Point(3) = {(d+h), 0, 0, 1.0};
//+
Point(4) = {L, 0, 0, 1.0};
//+
Point(5) = {L, L, 0, 1.0};
//+
Point(6) = {0, L, 0, 1.0};
//
Point(7) = {0, (d+h), 0, 1.0};
//
Point(8) = {0, d, 0, 1.0};
//
Line(1) = {1, 2};
//+
Line(2) = {2, 3};
//+
Line(3) = {3, 4};
//+
Line(4) = {4, 5};
//+
Line(5) = {5, 6};
//+
Line(6) = {6, 7};
//+
Line(7) = {7, 8};
//+
Line(8) = {8, 1};
//+
Curve Loop(1) = {1, 2, 3, 4, 5, 6, 7, 8};
//+
Plane Surface(1) = {1};
//+
Transfinite Surface {1} = {1, 4, 5, 6};
//+
Transfinite Curve {2, 7} = (k+1) Using Progression 1;
//+
Transfinite Curve {1, 8} = (q+1) Using Progression 1;
//+
Transfinite Curve {3, 6} = (r+1) Using Progression 1;
//+
Transfinite Curve {4, 5} = (n+1) Using Progression 1;
//+
Recombine Surface {1};
//+
Extrude {0, 0, hz} {
    Surface{1}; Layers {1}; Recombine;
}
//+
Physical Surface("inlet", 19) = {8};

```

```
//+
Physical Surface("outlet", 20) = {3};
//+
Physical Surface("fixedWalls", 21) = {2, 4, 5, 6, 7, 9};
//+
Physical Surface("frontAndBack", 22) = {10, 1};
//+
Physical Volume("internal", 23) = {1};
```

B.2 Malhas - MEF

Nesta seção deste apêndice descreve-se os arquivos que criaram as malhas utilizadas pelo software Octave, após a solução do escoamento, a fim de encontrar a geometria ótima. É importante ressaltar que essas malhas devem possuir ordem quadrática.

B.2.1 *Bocal*

```
// Gmsh project created on Mon Aug 28 17:41:47 2023
SetFactory("OpenCASCADE");
L=1;
k = 10;           //number of cells on the exit
n=3*k;           //number of cells on y axis
m = n;           //number of cells on x axis
hz=0.1*(L/n);
//+
Point(1) = {0, 0, 0, 1.0};
//+
Point(2) = {L, 0, 0, 1.0};
//+
Point(3) = {L, L/3, 0, 1.0};
//+
Point(4) = {L, 2*L/3, 0, 1.0};
//+
Point(5) = {L, L, 0, 1.0};
//+
Point(6) = {0, L, 0, 1.0};
//
Line(1) = {1, 2};
//+
Line(2) = {2, 3};
//+
Line(3) = {3, 4};
//+
Line(4) = {4, 5};
//+
Line(5) = {5, 6};
//+
Line(6) = {6, 1};
//+
Curve Loop(1) = {1, 2, 3, 4, 5, 6};
//+
Plane Surface(1) = {1};
//+
Transfinite Surface {1} = {1, 2, 5, 6};
//+
Transfinite Curve {2, 3, 4} = (k+1) Using Progression 1;
//+
```

```

Transfinite Curve {6} = (n+1) Using Progression 1;
//+
Transfinite Curve {5, 1} = (m+1) Using Progression 1;
//+
Recombine Surface {1};
//+//+
Physical Surface("internal", 10) = {1};
//+
Physical Curve("inlet", 1) = {6};
//+
Physical Curve("outlet", 2) = {3};
//+
Physical Curve("fixedWalls", 3) = {4, 5, 2, 1};
//+

```

B.2.2 Curva T

```

// Gmsh project created on Mon Aug 28 17:41:47 2023
SetFactory("OpenCASCADE");
L=1;
d = 4*(L/5); //distance from center of opening to origin
h = L/5; //size of the openings

lambda = 3; //discretization, any positive integer bigger than 0
k = 2*lambda; //number of cells on the exit
q = (7/2)*k; //number of cells on the biggest wall
r = (1/2)*k; //number of cells on the smallest wall
n=(k+q+r); //total number of cells

hz=0.1*(L/n);

//+
Point(1) = {0, 0, 0, 1.0};
//+
Point(2) = {(d-(h/2)), 0, 0, 1.0};
//+
Point(3) = {(d+(h/2)), 0, 0, 1.0};
//+
Point(4) = {L, 0, 0, 1.0};
//+
Point(5) = {L, L, 0, 1.0};
//+
Point(6) = {(d+(h/2)), L, 0, 1.0};
//
Point(7) = {(d-(h/2)), L, 0, 1.0};
//
Point(8) = {0, L, 0, 1.0};
//
Point(9) = {0, (d+(h/2)), 0, 1.0};
//
Point(10) = {0, (d-(h/2)), 0, 1.0};
//
Line(1) = {1, 2};
//+
Line(2) = {2, 3};
//+
Line(3) = {3, 4};
//+
Line(4) = {4, 5};

```

```

//+
Line(5) = {5, 6};
//+
Line(6) = {6, 7};
//+
Line(7) = {7, 8};
//+
Line(8) = {8, 9};
//+
Line(9) = {9, 10};
//+
Line(10) = {10, 1};
//+
Curve Loop(1) = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
//+
Plane Surface(1) = {1};
//+
Transfinite Surface {1} = {1, 4, 5, 8};
//+
Transfinite Curve {2, 6, 9} = (k+1) Using Progression 1;
//+
Transfinite Curve {1, 7, 10} = (q+1) Using Progression 1;
//+
Transfinite Curve {3, 5, 8} = (r+1) Using Progression 1;
//+
Transfinite Curve {4} = (n+1) Using Progression 1;
//+
Recombine Surface {1};
//+
Physical Curve("inlet", 1) = {9};
//+
Physical Curve("outlet", 2) = {2, 6};
//+
Physical Curve("fixedWalls", 3) = {1, 3, 4, 5, 7, 8, 10};
//+
Physical Surface("internal", 10) = {1};

```

B.2.3 Joelho

```

// Gmsh project created on Mon Aug 28 17:41:47 2023
SetFactory("OpenCASCADE");
L=1;
d = 0.7*L;      //distance of the exit from the origin
h = 0.2*L;      //size of the exit

lambda = 5;     //discretization

k = 2*lambda;   //number of cells on the exit
r = (1/2)*k;    //number of cells on the line after the exit
q = (7/2)*k;    //number of cells on the line before the exit
n=(k+r+q);      //number of cells on y axis (10*lambda)

hz=0.1*(L/n);
//+
Point(1) = {0, 0, 0, 1.0};
//+
Point(2) = {d, 0, 0, 1.0};
//+
Point(3) = {(d+h), 0, 0, 1.0};

```

```

//+
Point(4) = {L, 0, 0, 1.0};
//+
Point(5) = {L, L, 0, 1.0};
//+
Point(6) = {0, L, 0, 1.0};
//
Point(7) = {0, (d+h), 0, 1.0};
//
Point(8) = {0, d, 0, 1.0};
//
Line(1) = {1, 2};
//+
Line(2) = {2, 3};
//+
Line(3) = {3, 4};
//+
Line(4) = {4, 5};
//+
Line(5) = {5, 6};
//+
Line(6) = {6, 7};
//+
Line(7) = {7, 8};
//+
Line(8) = {8, 1};
//+
Curve Loop(1) = {1, 2, 3, 4, 5, 6, 7, 8};
//+
Plane Surface(1) = {1};
//+
Transfinite Surface {1} = {1, 4, 5, 6};
//+
Transfinite Curve {2, 7} = (k+1) Using Progression 1;
//+
Transfinite Curve {1, 8} = (q+1) Using Progression 1;
//+
Transfinite Curve {3, 6} = (r+1) Using Progression 1;
//+
Transfinite Curve {4, 5} = (n+1) Using Progression 1;
//+
Recombine Surface {1};
//+

Physical Curve("inlet", 1) = {7};
//+
Physical Curve("outlet", 2) = {2};
//+
Physical Curve("fixedWalls", 3) = {1, 3, 4, 5, 6, 8};
//+
Physical Surface("internal", 10) = {1};

```

APÊNDICE C – CONDIÇÕES DE CONTORNO UTILIZADAS NO MVF

Neste apêndice retrata-se os arquivos de texto contendo as condições de contorno utilizadas pelo software OpenFOAM para resolver o escoamento. O arquivo *kappa*, embora possua condições de contorno, está presente no apêndice A.1 já que depende da malha. Como a única condição de contorno que muda entre as simulações é o cálculo da velocidade parabólica da entrada (o que ocorre justamente pela configuração física das entradas), apenas um exemplo de arquivo será fornecido. Para futuras simulações, as configurações físicas da entrada devem ser levadas em consideração.

No caso da velocidade, é possível trocar entre o perfil parabólico laminar e turbulento (equacionado utilizando a regra da potência de $1/7$), bastando comentar uma das linhas. As condições de contorno para a viscosidade turbulenta foram as sugeridas pelos autores do modelo de turbulência utilizado (Wray; Agarwal, 2015).

As condições de contorno aqui apresentadas foram as utilizadas na otimização do joelho em regime turbulento.

C.1 Velocidade

```

/*----- C++ -----*/
| ===== |
| \ \      /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
| \ \      /  O peration     | Version: v2012 |
| \ \      /  A nd           | Website: www.openfoam.com |
|  \ \     /  M anipulation  | |
|-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        volVectorField;
    object       U;
}
// * * * * *

dimensions      [0 1 -1 0 0 0 0];

internalField    uniform (0 0 0);

boundaryField
{
    inlet
    {
        type      codedFixedValue;
        value      uniform (0 0 0);
        name       parabolicVelocity1;

        code
        #{

```



```
// * * * * *

dimensions      [0 2 -2 0 0 0 0];

internalField    uniform 0;

boundaryField
{
    inlet
    {
        type      zeroGradient;
    }

    outlet
    {
        type      fixedValue;
        value      uniform 0;
    }

    fixedWalls
    {
        type      zeroGradient;
    }

    frontAndBack
    {
        type      empty;
    }
}

// * * * * *
```

C.3 Viscosidade Turbulenta

```
/*-----*-- C++ --*-----*/
| ===== |
| \ \      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
| \ \      / O p e r a t i o n | Version: plus |
| \ \      / A n d             | Web: www.OpenFOAM.com |
| \ \      / M a n i p u l a t i o n |
|=====|
FoamFile
{
    version      2.0;
    format        ascii;
    class        volScalarField;
    object       nuTilda;
}
// * * * * *

dimensions      [0 2 -1 0 0 0 0];

internalField    uniform 5e-5;

boundaryField
{
    inlet
    {
        type      fixedValue;
    }
}
```



```
        value          uniform 5e-5;
    }
    outlet
    {
        type            zeroGradient;
    }
    fixedWalls
    {
        type            fixedValue;
        value            uniform 0;
    }
    frontAndBack
    {
        type            empty;
    }
}

// ***** //
```

APÊNDICE D – ARQUIVOS DE CONTROLE DO MVF

Neste apêndice retrata-se os arquivos de texto que controlam a simulação por MVF feita no software OpenFOAM®, especificamente os arquivos *controlDict*, que controla os parâmetros da simulação, *fvSchemes*, que especifica quais esquemas de interpolação são utilizados para cada termo das equações governantes e *fvSolution* que determina a tolerância mínima para cada uma dessas interpolações.

D.1 *controlDict*

```

/*-----*- C++ -*-----*/
| ===== |
| \ \      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
| \ \      / O p e r a t i o n | Version: v2012 |
| \ \      / A n d | Website: www.openfoam.com |
| \ \ / M a n i p u l a t i o n | |
/*-----*/

FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "system";
    object       controlDict;
}
// * * * * *

application      BrinksimpleFoam;

startFrom        startTime;

startTime        0;

stopAt           endTime;

endTime          200;

deltaT           1;

writeControl      timeStep;

writeInterval     50;

purgeWrite        0;

writeFormat       ascii;

writePrecision    6;

writeCompression off;

timeFormat        general;

```



```

snGradSchemes
{
    default            corrected;
}

wallDist
{
    method meshWave;
}

// *****

```

D.3 *fvSolution*

```

/*----- C++ -----*/
| ===== |
| \ \      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
| \ \      / O p e r a t i o n | Version: plus |
| \ \      / A n d      | Web: www.OpenFOAM.com |
| \ \      / M a n i p u l a t i o n |
| \ \      / |
/*-----*/

FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "system";
    object       fvSolution;
}

// *****

solvers
{
    p
    {
        solver      GAMG;
        tolerance    1e-16;
        relTol       0.01;
        smoother     GaussSeidel;
    }

    "(U|k|epsilon|omega|f|v2|nuTilda|RWA)"
    {
        solver      smoothSolver;
        smoother     symGaussSeidel;
        tolerance    1e-16;
        relTol       0.01;
    }
}

SIMPLE
{
    nNonOrthogonalCorrectors 0;
    consistent yes;

    residualControl
    {
        p            1e-3;
    }
}

```

```

        U          1e-4;
        "(k|epsilon|omega|f|v2|nuTilda|FWA)" 1e-4;
    }
}

relaxationFactors
{
    equations
    {
        U          0.8; // 0.9 is more stable but 0.95 more convergent
        ".*"       0.6; // 0.9 is more stable but 0.95 more convergent
    }
}

// ***** //
```

APÊNDICE E – ROTINA DE AUTOMATIZAÇÃO EM BASH

Neste apêndice encontra-se a rotina de automatização escrita em Bash Script.

```
#!/usr/bin/bash

bash clear_routine.sh          #clear old files , if exist
python3 generatefiles.py       #generates new files for this simulation and mesh
mv kappa 0                     #moves "kappa" file to "0" folder
    ↪
mv sampleDict system           #moves "sampleDict" file to "system" folder

cd 'OptBib'
octave start_q.m               #starts penalty's value
cd '..'

SECONDS=0
while [ ! -f 'OptBib/stop' ]; do

    BrinksimpleFoam > log
    ↪                               #solves the flow
    postProcess -func sampleDict -latestTime > /dev/null    #interpolate
    ↪ velocities (>/dev/null "sends" the reports of the post processing
    ↪ operations to a null file , so it doesn't show on the console)
    read -r iter < 'MOTiter'

    cd 'OptBib'
    ↪
    ↪                               #change directory to optimizer's
    ↪ directory
    octave MOT_NS_Q2Q1.m
    ↪                               #runs the optimizer
    read -r residue < 'residue'
    printf "Iteration number %d completed. Residue: %f\n" $iter $residue

    if [ ! -f "stop" ]; then
        ↪                               #moving new files
        mv kappa '../0'
        mv MOTiter '..'
        mv MOTxold '..'
    fi

    cd '..'
    ↪                               #retuns to original directory

    foamListTimes -rm
    ↪                               #removes old simulation
    rm -r "postProcessing"
    ↪                               #removes old interpolation

done

duration=$SECONDS
echo "Optimization process finished in $((duration / 60)) minutes and $((duration %
    ↪ 60)) seconds."
```

```
BrinksimpleFoam > log
↪ #solves the flow

latestTime=$(foamListTimes -latestTime | tail -n 1) #gets latest timestep
cp 'OptBib/alpha' $latestTime #
↪ copies alpha file to latest time, for visualization

echo -en "\007"
↪ #beep noise to mark end of code
```