

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
BACHARELADO EM ENGENHARIA AEROESPACIAL

LUCCA RONCADOR LUCHETI

DESENVOLVIMENTO DE ALGORITMO DE AUTOMAÇÃO PARA SIMULAÇÕES
CFD EM MODELOS PARAMETRIZADOS DE ASAS DE AERONAVES

Joinville
2025

LUCCA RONCADOR LUCHETI

DESENVOLVIMENTO DE ALGORITMO DE AUTOMAÇÃO PARA SIMULAÇÕES
CFD EM MODELOS PARAMETRIZADOS DE ASAS DE AERONAVES

Trabalho apresentado como requisito para obtenção do título de Bacharel em Engenharia Aeroespacial, no Centro Tecnológico de Joinville, da Universidade Federal de Santa Catarina.

Orientador(a): Dr. Thiago Pontin Tancredi

Joinville
2025

LUCCA RONCADOR LUCHETI

DESENVOLVIMENTO DE ALGORITMO DE AUTOMAÇÃO PARA SIMULAÇÕES
CFD EM MODELOS PARAMETRIZADOS DE ASAS DE AERONAVES

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de Bacharel em Engenharia Aeroespacial, no Centro Tecnológico de Joinville, da Universidade Federal de Santa Catarina.

Joinville (SC), 24 de novembro de 2025.

Banca Examinadora:

Prof. Dr. Thiago Pontin Tancredi
Orientador/Presidente

Prof. Dr. Antônio Otaviano Dourado
Membro(a)
UFSC

Prof. Dr. Victor Simões Barbosa
Membro(a)
UFSC

Dedico este trabalho aos meus queridos pais.

AGRADECIMENTOS

Agradeço, em primeiro lugar, aos colegas que fizeram parte da minha trajetória na graduação, em especial aos amigos da República Sem Saída. Sem eles, a jornada universitária seria muito mais solitária e desgastante. Foram eles que tornaram o cotidiano acadêmico divertido e significativo, permitindo que eu vivenciasse a experiência universitária de uma forma muito mais leve.

Agradeço à Universidade Federal de Santa Catarina e a todos os seus docentes, que contribuíram diretamente para a minha formação acadêmica e pessoal. Graças ao conhecimento, às oportunidades e aos desafios que encontrei dentro da instituição, sinto-me hoje verdadeiramente preparado para encarar problemas complexos e seguir crescendo como profissional e como indivíduo. Em especial, agradeço ao meu orientador, Prof. Dr. Thiago Pontin Tancredi pela confiança depositada em mim ao longo deste trabalho. Agradeço também ao Prof. Antônio Otaviano Dourado, que desde o início da minha graduação se mostrou uma grande inspiração pessoal e profissional, servindo como exemplo de seriedade e dedicação pela engenharia.

Agradeço também à minha namorada, Rafaela, pelo apoio constante em toda a jornada final do curso e, em particular, durante o desenvolvimento deste trabalho. Sua presença, paciência e compreensão foram fundamentais nos momentos de maior pressão e cansaço. O suporte emocional, o incentivo diário e a disposição em me ajudar em tudo o que fosse possível tornaram esse percurso possível.

Por fim, e principalmente, agradeço aos meus pais, minhas maiores inspirações. São eles que sempre deram e continuam dando tudo o que podem por mim, colocando o amor pelos filhos acima de qualquer dificuldade. Foram eles que me instigaram, desde cedo, a buscar o conhecimento, a valorizar o estudo e a encarar os desafios com responsabilidade e coragem. Toda a base sólida que me permitiu chegar até aqui foi construída por eles, e reconheço o quão privilegiado sou por ter os pais que tenho. Este trabalho é, em grande medida, fruto do esforço, do cuidado e do amor que sempre recebi deles.

“ We need to do our part to make the world a better place! ”

Greg, Over The Garden Wall

RESUMO

Recentes e significativos avanços nos recursos computacionais e na capacidade de processamento permitiram o uso de técnicas sofisticadas, como a dinâmica dos fluidos computacional (CFD), ainda nas fases iniciais de projeto; porém, na prática da engenharia, o emprego dessas técnicas costuma se limitar a modelos paramétricos baseados em métodos empíricos. Além disso, o alto custo das ferramentas, a complexidade de modelagem e as diversas iterações necessárias podem tornar o uso do CFD, por vezes, inviável. Visando acelerar o processo inicial de modelagem e testes, bem como reduzir seu custo, apresenta-se neste trabalho uma metodologia automatizada de simulação que integra a modelagem 3D da asa de uma aeronave parametrizada a um fluxo de CFD voltado à determinação do arrasto do corpo. A geometria da asa é descrita por funções matemáticas que caracterizam suas curvas, e o modelo de simulação é construído utilizando a biblioteca de código aberto OpenFOAM. A discretização do domínio e os parâmetros de entrada da simulação e da geração de malha são ajustados iterativamente até que os critérios de qualidade e o valor de y^+ estejam dentro dos limites estabelecidos. Na automação, utiliza-se o solver *simpleFoam* acoplado ao modelo de turbulência $k-\omega$ SST, de modo que, por meio de um código em Python, automatiza-se todo o processo de geração de malha, execução da simulação e extração dos principais resultados. O código recebe sete parâmetros de entrada: seis relacionados à definição geométrica da asa e um correspondente à velocidade do escoamento. Os resultados obtidos a partir de diferentes parâmetros de geometria e escoamento foram comparados com valores empíricos para efeito de validação. Como exemplo de aplicação, é apresentado um estudo paramétrico cuja análise de qualidade de malha foi conduzida manualmente, não sendo identificados problemas significativos na abordagem automatizada.

Palavra-chave: CFD. Python. OpenFoam. asa de aeronave.

ABSTRACT

Recent and significant advances in computational resources and processing power have enabled the use of sophisticated techniques, such as Computational Fluid Dynamics (CFD), already in the early stages of design; however, in engineering practice, the use of these techniques is often restricted to parametric models based on empirical methods. In addition, the high cost of the tools, the modeling complexity, and the numerous iterations required can make the use of CFD unfeasible. Aiming to accelerate the initial process of modeling and testing, as well as to reduce its cost, this work presents an automated simulation methodology that integrates the 3D modeling of an aircraft wing, defined by a parametrized geometry, with a CFD workflow focused on determining the drag of the body. The wing geometry is described by mathematical functions that characterize its curves, and the simulation model is built using the open-source library OpenFOAM. The discretization of the domain and the input parameters for both the simulation and mesh generation are adjusted iteratively until the mesh quality criteria and the value of y^+ fall within predefined limits. In the automation, the *simpleFoam* solver is employed, coupled with the $k-\omega$ SST turbulence model, so that a Python code automates the entire process of mesh generation, simulation execution, and extraction of the main results. The code receives seven input parameters: six related to the geometric definition of the wing and one corresponding to the freestream velocity. The results obtained for different geometric and flow parameters were compared with empirical data for validation purposes. As an application example, a parametric study is presented, in which the mesh quality assessment was conducted manually, and no significant issues were identified in the automated approach.

Keywords: CFD. Python. OpenFoam. airplane wing.

LISTA DE FIGURAS

Figura 1 – Fluxo de trabalho em CFD	20
Figura 2 – Exemplo de malha estruturada H-type bidimensional	21
Figura 3 – Exemplo de malha não estruturada com elementos triangulares	22
Figura 4 – Comparação entre C_L e ângulo de ataque para diferentes topologias de malha	23
Figura 5 – Comparação entre C_L e C_D para diferentes densidades de malha não estruturada	24
Figura 6 – Esquemas de discretização: cell-centered (à esquerda) e cell-vertex (à direita)	26
Figura 7 – Regiões da camada limite turbulenta e a localização das células de malha em função de y^+	30
Figura 8 – Comparação entre modelos comuns de aerofólios e séries NACA	33
Figura 9 – Comparação dos valores simulados e experimentais de C_d e C_l entre os modelos de turbulência	42
Figura 10 – Torque aerodinâmico previsto por diferentes modelos de turbulência em função da velocidade do vento	44
Figura 11 – Estrutura típica de diretórios em um caso OpenFOAM	47
Figura 12 – Fluxo metodológico adotado no estudo	52
Figura 13 – Visualizações do STL da asa de aerofólio NACA 4415 a 10°	61
Figura 14 – Visualizações do STL de diferentes modelos de aerofólio (a) NACA 2406 a 8° (b) NACA 2412 a 6° (c) NACA 4412 a 15° (d) NACA 2212 a 4° (e) NACA 3406 a -8° (f) NACA 5224 a 16° (g) NACA 0012 a 0°	62
Figura 15 – Visão geral da malha do aerofólio NACA 4415 a 10°	63
Figura 16 – Corte transversal da malha do aerofólio NACA 4415 a 10°	64
Figura 17 – Visualização do arquivo STL da asa enflechada	68
Figura 18 – Visualização do arquivo STL da asa enflechada	70
Figura 19 – Gráficos de velocidade e pressão para seções da asa feitas a diferentes distâncias no eixo da envergadura em relação à corda da asa (a) U - 0.2m (b) U - 0.5m (c) U - 0.8m (d) p - 0.2m	72
Figura 20 – Resíduos e coeficientes aerodinâmicos ao longo da simulação para NACA 4415 a 14°	73

LISTA DE QUADROS

Quadro 1 – Resumo das malhas não estruturadas.	24
Quadro 2 – Comparação sucinta de modelos de turbulência para simulações de asas	41
Quadro 3 – Comparação entre os resultados obtidos com DNS, LES e RANS para o escoamento na cavidade com tampa móvel	50

LISTA DE TABELAS

Tabela 1 – y^+ calculado para os modelos simulados	65
Tabela 2 – Comparação dos resultados numéricos e experimentais para diferentes configurações	66
Tabela 3 – Comparação dos resultados numéricos e experimentais para a asa enflechada a 14°	71

LISTA DE ABREVIATURAS E SIGLAS

NACA	National Advisory Committee for Aeronautics
CFD	Computational Fluid Dynamics (Dinâmica dos Fluidos Computacional)
HPC	High Performance Computing (Computação de Alto Desempenho)
CD	Coeficiente de arrasto
CL	Coeficiente de sustentação
CM	Coeficiente de momento de arfagem
CST	Class-Shape Transformation (técnica de parametrização de perfis e asas)
GPL	General Public License (licença pública geral)
SIMPLE	Semi-Implicit Method for Pressure-Linked Equations (algoritmo para acoplamento pressão-velocidade)
STL	StereoLithography (formato de arquivo para representação geométrica tridimensional)

SUMÁRIO

1	INTRODUÇÃO	14
1.1	OBJETIVOS	16
1.1.1	Objetivo geral	16
1.1.2	Objetivos Específicos	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	INTRODUÇÃO À DINÂMICA DOS FLUIDOS COMPUTACIONAL	17
2.2	GERAÇÃO DE MALHA	20
2.2.1	Malhas Estruturadas e Não Estruturadas	21
2.2.2	Ferramentas de Geração de Malha	24
2.2.3	Crerios de Qualidade de Malha e Refinamento	27
2.2.4	Métrica y^+	28
2.3	MODELAGEM DE ASAS DE AERONAVES	31
2.3.1	Séries NACA	32
2.3.2	Geometria e Parametrização de Asas	34
2.4	MODELOS DE TURBULÊNCIA EM CFD	37
2.4.1	Escolha do Modelo de Turbulência para Simulações de Asas	37
2.4.2	Validação de Modelos de Turbulência com Dados Experimentais	40
2.5	OPENFOAM COMO FERRAMENTA DE CFD	45
2.5.1	Solvers e Modelos de Turbulência no OpenFOAM	47
2.6	AUTOMAÇÃO DE PROCESSOS CFD	49
3	METODOLOGIA	52
3.1	MODELAGEM DA ASA	52
3.2	GERAÇÃO DE MALHA	54
3.2.1	Condições de Contorno	54
3.2.2	Integração da malha do domínio com a geometria da asa	55
3.3	ESTUDO PRELIMINAR	56
4	RESULTADOS	59
4.1	MODELO COM SEÇÃO CONSTANTE	59
4.1.1	Configuração das Simulações	60
4.1.2	Qualidade das Malhas e Parâmetros de Simulação	63
4.1.3	Comparação dos Resultados Numéricos com Dados Experimentais	65
4.2	MODELO COM SEÇÃO VARIÁVEL	67
4.2.1	Configuração das Simulações	67
4.2.2	Qualidade das Malhas e Parâmetros de Simulação	69

4.2.3	Comparação dos Resultados Numéricos com Dados Experimentais	70
4.3	ANÁLISE DE CONVERGÊNCIA	73
5	CONCLUSÃO	75
	REFERÊNCIAS	77
	APÊNDICES	81
	APÊNDICE A – GERAÇÃO DO STL DA ASA	82
	APÊNDICE B – GERAÇÃO DA MALHA	84

1 INTRODUÇÃO

A indústria aeroespacial passa por um processo contínuo de evolução e adaptação para atender às complexas demandas de mercado, impulsionadas pelos avanços tecnológicos e pela crescente necessidade de eficiência e sustentabilidade. Esse cenário cria a necessidade de abordagens inovadoras na gestão de qualidade e processos, visando produtos seguros, eficientes e sustentáveis (Pop et al., 2023).

Antes da década de 1960, na engenharia, métodos computacionais eram dificilmente utilizados em análises aerodinâmicas, apesar de já serem comumente empregados em análises estruturais. Até então, a análise aerodinâmica de aeronaves era conduzida quase exclusivamente por meio de testes experimentais em túneis de vento, nos quais modelos físicos eram submetidos a diferentes condições de escoamento para avaliar parâmetros como sustentação, arrasto e estabilidade (Jameson, 1983).

Essas instalações permitiam a visualização do escoamento ao redor de perfis aerodinâmicos e fuselagens, fornecendo dados experimentais sobre coeficientes de sustentação e arrasto. Entretanto, devido às limitações tecnológicas da época, os ajustes geométricos eram realizados manualmente, e cada nova configuração demandava a fabricação de modelos físicos, tornando o processo de desenvolvimento de aeronaves lento e oneroso (Jameson, 1983).

A introdução das séries NACA, por exemplo, foi um marco na engenharia aeronáutica, permitindo a padronização de aerofólios e a utilização de métodos como o Teorema de Theodorsen para prever o comportamento dos perfis aerodinâmicos. Ainda assim, a necessidade de soluções analíticas e experimentais limitava a capacidade de análise para geometrias complexas, uma vez que apenas casos simplificados podiam ser resolvidos matematicamente com precisão (Jameson, 1983).

Foi na década de 1960 que os avanços nas ferramentas computacionais e na capacidade de processamento abriram caminho para o desenvolvimento da aerodinâmica computacional, permitindo a aplicação de modelos matemáticos mais complexos na previsão do comportamento dos fluidos. Inicialmente, os métodos computacionais se concentravam em problemas simplificados, como fluxos potenciais incompressíveis, nos quais a equação de Laplace podia ser aplicada para estimar campos de pressão e velocidades (Jameson, 1983).

Durante a década de 1970, a previsão de escoamentos transônicos tornou-se um dos desafios centrais da aerodinâmica computacional, uma vez que o aumento da velocidade de cruzeiro das aeronaves exigia uma melhor compreensão dos efeitos de compressibilidade. A formação de ondas de choque sobre aerofólios e fuselagens resultava em um aumento expressivo do arrasto aerodinâmico, impactando diretamente

a eficiência do voo (Jameson, 1983).

Já nos anos 1980, com o desenvolvimento de sistemas de transporte espacial, essa tecnologia começou a ser utilizada para simular fluxos hipersônicos e reações químicas fluidas. A aerodinâmica de reentrada atmosférica, um dos desafios mais críticos da engenharia aeroespacial, passou a ser amplamente investigada por meio de simulações numéricas, permitindo análises detalhadas do aquecimento aerodinâmico e da interação fluido-química. Desde então, a simulação CFD se consolidou como uma ferramenta essencial para a análise e design de veículos aéreos e espaciais (Fujii, 2005).

Apesar do progresso significativo, o CFD ainda enfrenta desafios substanciais. Embora seja possível realizar simulações de escoamentos turbulentos e fenômenos de separação de camada limite, a precisão dos resultados depende de fatores como a qualidade da malha computacional e a escolha de modelos de turbulência e condições de contorno adequadas. Além disso, a influência dos efeitos de escala sobre os coeficientes aerodinâmicos ainda não é totalmente compreendida, fazendo com que a validação experimental muitas vezes ainda seja uma etapa essencial para a confiabilidade das previsões numéricas (Fujii, 2005).

Nesse contexto, a complexidade e o tempo elevado necessários para a preparação, execução e análise das simulações CFD fazem desse campo um nicho altamente técnico e especializado, ressaltando a necessidade de abordagens inovadoras. Apesar da existência de metodologias voltadas à automação desse processo, a literatura ainda carece de estudos que explorem soluções open-source e estratégias parametrizadas aplicadas a modelos de asas já consolidados.

Segundo a pesquisa conduzida por Lange et al. (2021), a automação de análises de CFD, combinada com sistemas de computação de alto desempenho (HPC), pode reduzir o tempo de processamento e aumentar a quantidade de iterações de design em um mesmo espaço de tempo. No entanto, apesar dessas vantagens, ainda não havia estudos que quantificassem seus efeitos no desenvolvimento de produtos. A ausência de dados concretos reforça a necessidade de investigações que avaliem sua influência na eficiência dos processos de engenharia.

Para suprir essa lacuna, os autores estudaram uma equipe de competição formada por estudantes chamada "Elefant Racing", da Universidade de Bayreuth, na Alemanha, representando pequenas e médias empresas. Durante várias temporadas, eles forneceram HPC e métodos automatizados para otimizar as simulações CFD da equipe (Lange et al., 2021).

A automação possibilita a execução de múltiplas iterações de projeto sem intervenção manual, garantindo maior consistência nos resultados e aumentando a eficiência do processo. Quando combinada ao uso de HPC, essa abordagem permite a análise de geometrias complexas com maior resolução e menor tempo de proces-

samento. No entanto, a implementação de fluxos automatizados em HPC requer um planejamento cuidadoso, considerando a alocação de recursos computacionais, a integração com ferramentas de modelagem e a validação dos resultados.

A comparação dos indicadores de desempenho antes e depois revelou um aumento de 115% na eficiência aerodinâmica e 600% no número de iterações por temporada. Os resultados demonstram que HPC e automação podem otimizar o desenvolvimento de produtos, reduzindo tempo e custo (Lange et al., 2021).

Dessa forma, o objetivo principal desta pesquisa é o desenvolvimento e a implementação de uma metodologia de automação em Python para o fluxo completo de simulação de CFD aplicada a modelos parametrizados de asas de aeronaves. Esta automação englobará todas as etapas, desde a modificação geométrica e geração de malha até a execução das simulações e o pós-processamento dos resultados aerodinâmicos, visando otimizar a análise de desempenho de perfis aerodinâmicos.

A proposta busca resolver o problema da elevada complexidade e do tempo despendido nas análises iterativas de CFD no design aeroespacial. Ao fornecer uma solução open-source e parametrizada, o trabalho visa demonstrar como a automação aumenta a eficiência e a consistência do processo, permitindo que engenheiros explorem um espaço de projeto mais amplo e atinjam configurações de asas otimizadas com maior agilidade e menor custo operacional.

1.1 OBJETIVOS

1.1.1 Objetivo geral

Desenvolver um algoritmo capaz de executar uma simulação CFD para um tipo de geometria e escoamento, e extrair os principais resultados de forma completamente automatizada, a partir de parâmetros simples de entrada.

1.1.2 Objetivos Específicos

- Definir um modelo de parametrização do domínio do escoamento e da geometria da asa;
- Implementar a geração automática da malha computacional;
- Obter, com a automação, resultados confiáveis e comparáveis a dados empíricos;
- Analisar graficamente os resultados obtidos.

2 FUNDAMENTAÇÃO TEÓRICA

A apresentação deste capítulo visa proporcionar uma base teórica para a compreensão das características do fluxo ao redor de asas de aeronaves e explorar técnicas de automação que aumentem a eficiência e a precisão das simulações CFD, contribuindo para avanços no projeto de veículos aéreos.

A construção da automação proposta neste trabalho exige um entendimento abrangente de todas as etapas envolvidas no processo, desde a definição do problema até a análise e interpretação dos resultados. A implementação de scripts em Python para controlar o software de simulação deve replicar com precisão o fluxo metodológico de uma simulação tradicional, garantindo que cada fase seja corretamente configurada e executada.

Portanto, neste capítulo, é apresentada a fundamentação teórica que sustenta esta pesquisa, abordando os conceitos fundamentais de dinâmica dos fluidos computacional e o uso de ferramentas de automação em Python para otimizar simulações. Serão discutidos os princípios da mecânica dos fluidos que governam o comportamento do escoamento ao redor de asas de aeronaves, focando nas interações entre fluxo e superfície em um contexto aerodinâmico.

Além disso, são revisados trabalhos e artigos relevantes que investigam as aplicações de CFD na engenharia aeroespacial, com ênfase em simulações para projeto e análise de asas de aeronaves. A seção também aborda a utilização de Python como uma linguagem de automação, explorando suas bibliotecas e frameworks que permitem a construção de ferramentas robustas para simplificar processos de simulação e análise de dados.

2.1 INTRODUÇÃO À DINÂMICA DOS FLUIDOS COMPUTACIONAL

A dinâmica dos fluidos computacional (CFD) é um campo voltado à análise de sistemas que envolvem o escoamento de fluidos, a transferência de calor e fenômenos associados, como reações químicas, por meio de simulações numéricas. Sua aplicação abrange diversos setores industriais, desde a engenharia, com estudos aerodinâmicos de aeronaves e hidrodinâmicos de embarcações, até a biomedicina, onde é empregada para investigar o fluxo sanguíneo em artérias e veias (Versteeg; Malalasekera, 2007).

A partir da década de 1960, a indústria aeroespacial começou a integrar essas técnicas no projeto e desenvolvimento de aeronaves e motores a jato, sendo posteriormente aplicadas a diversos setores da engenharia. Hoje, o CFD se consolidou como uma ferramenta essencial para o estudo de sistemas complexos, reduzindo custos e

proporcionando análises detalhadas que complementam experimentos físicos (Versteeg; Malalasekera, 2007).

O CFD pode ser dividido em três etapas principais: pré-processamento, processamento e pós-processamento. Essa estruturação é amplamente adotada, pois permite organizar as etapas da simulação de forma sistemática, facilitando a implementação e a análise dos resultados, além de possibilitar a otimização de cada fase individualmente, garantindo maior controle sobre a precisão e a estabilidade das soluções numéricas (Versteeg; Malalasekera, 2007).

O pré-processamento constitui a etapa inicial da simulação, sendo responsável pela definição do domínio computacional e das condições de contorno. Nessa fase, o modelo físico do problema é estabelecido, incluindo a escolha das propriedades do fluido, como viscosidade e densidade, que influenciam diretamente o comportamento do escoamento. A definição precisa dessas propriedades é fundamental para garantir a acurácia da simulação (Ferziger; Perić, 2002).

O início de qualquer estudo em CFD consiste em caracterizar de forma inequívoca o fenômeno físico a ser investigado. Isso inclui definir o objetivo da simulação (por exemplo, estimar o arrasto, prever padrões de mistura ou avaliar gradientes térmicos) e identificar as variáveis de interesse — tais como velocidade, pressão e temperatura — bem como as escalas de comprimento e tempo que governam o escoamento (Anderson, 1995).

Uma vez delimitado o escopo, procede-se à formulação do modelo matemático. Nesse ponto, selecionam-se as equações de conservação que melhor descrevem o problema (continuidade, quantidade de movimento, energia, transporte de espécies, entre outras) e estabelecem-se hipóteses simplificadoras apropriadas — como escoamento incompressível, regime permanente ou ausência de fontes internas de calor. A escolha criteriosa dessas suposições é fundamental para equilibrar fidelidade física e custo computacional, garantindo que o solver seja capaz de convergir para soluções realistas (Anderson, 1995).

Com o domínio físico e o conjunto de equações definidos, tornam-se necessárias condições de contorno e iniciais que representem fielmente a situação real. Exemplos típicos incluem a condição de não deslizamento em superfícies sólidas, perfis de velocidade ou pressão especificados em entradas e saídas, e, quando pertinente, planos de simetria ou periodicidade. A coerência entre essas condições e as hipóteses assumidas na modelagem evita incompatibilidades numéricas e assegura que os resultados reflitam corretamente o comportamento do sistema (Anderson, 1995).

Em síntese, a fase de definição do problema estabelece o alicerce conceitual para todo o processo de simulação. Uma descrição física bem formulada, acompanhada de um modelo matemático apropriado e condições de contorno consistentes, reduz incertezas e potencializa a confiabilidade dos dados obtidos nas etapas subse-

quentes de malha, solução numérica e análise dos resultados.

Uma das tarefas mais importantes do pré-processamento é a geração da malha, que subdivide o domínio em pequenos volumes de controle onde as equações de conservação serão resolvidas. A qualidade da malha exerce influência direta na precisão dos resultados, sendo necessário garantir a sua adequação ao escoamento em estudo. Regiões com gradientes acentuados, como fronteiras sólidas e zonas de recirculação, exigem malhas mais refinadas para capturar os detalhes do fenômeno físico (Ferziger; Perić, 2002).

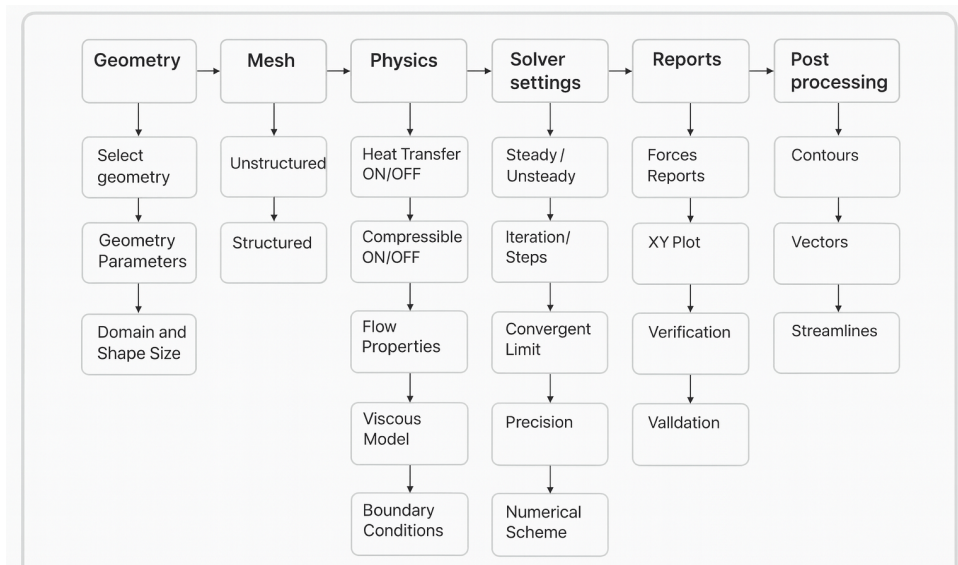
Na fase de processamento, as equações governantes — como as de Navier-Stokes, continuidade e energia — são resolvidas numericamente por meio de métodos como volumes finitos, diferenças finitas ou elementos finitos. O solver utiliza algoritmos iterativos para convergir a solução de forma computacional; porém, a monitoração da convergência e a estabilidade numérica ainda exigem intervenção do usuário para evitar resultados espúrios (Anderson, 1995).

O pós-processamento é dedicado à análise e interpretação dos resultados obtidos na fase anterior, permitindo a visualização de campos de velocidade, pressão, temperatura e outras variáveis relevantes. Essa fase é essencial para validar a simulação, por meio da comparação com dados experimentais ou analíticos, assegurando que os resultados sejam consistentes e aplicáveis ao problema real (Anderson, 1995).

Além disso, recursos como o rastreamento de partículas, animações interativas e exportação de imagens de alta resolução ampliam a capacidade de análise e comunicação dos resultados. Dados alfanuméricos, como valores médios e integrais em superfícies, também são extraídos para validação quantitativa com experimentos ou correlações analíticas. Assim, o pós-processamento conclui o ciclo de simulação, transformando dados brutos em informações úteis para embasar decisões de projeto e pesquisa (Versteeg; Malalasekera, 2007).

A Figura 1 mostra um fluxograma de CFD, elucidando as atividades em cada etapa.

Figura 1 – Fluxo de trabalho em CFD



Fonte – Khuntia e Ahuja (2018).

2.2 GERAÇÃO DE MALHA

A geração de malha em CFD está intimamente ligada à discretização do espaço por onde ocorre o escoamento de um fluido, processo que consiste na divisão do domínio físico em pequenos volumes de controle, sobre os quais são aplicadas as leis de conservação. Essa decomposição possibilita a formulação numérica dos fluxos convectivos e difusivos ao longo das faces que delimitam esses volumes (Blazek, 2015).

A qualidade e a adequação da malha influenciam diretamente a acurácia, estabilidade e eficiência computacional do método de solução adotado. Essa etapa antecede a resolução das equações de conservação e deve considerar não apenas a geometria da região de interesse, mas também as características do escoamento a ser simulado, como regiões de gradientes elevados ou a presença de camadas-limite (Blazek, 2015).

Diferentes geometrias e tipos de escoamento exigem estratégias específicas para a geração da malha. Em domínios simples, pode-se utilizar malhas estruturadas com controle refinado sobre a topologia das células, já em geometrias complexas e com múltiplas escalas, torna-se necessário recorrer a malhas não estruturadas ou híbridas, que permitem maior flexibilidade na adaptação do domínio (Ferziger; Perić, 2002).

A escolha adequada do tipo de malha e o seu refinamento em regiões críticas são elementos fundamentais para o sucesso de uma simulação CFD, pois garantem uma resolução suficiente dos fenômenos físicos relevantes com custo computacional viável. Nos tópicos seguintes, discute-se a classificação das malhas, as ferramentas

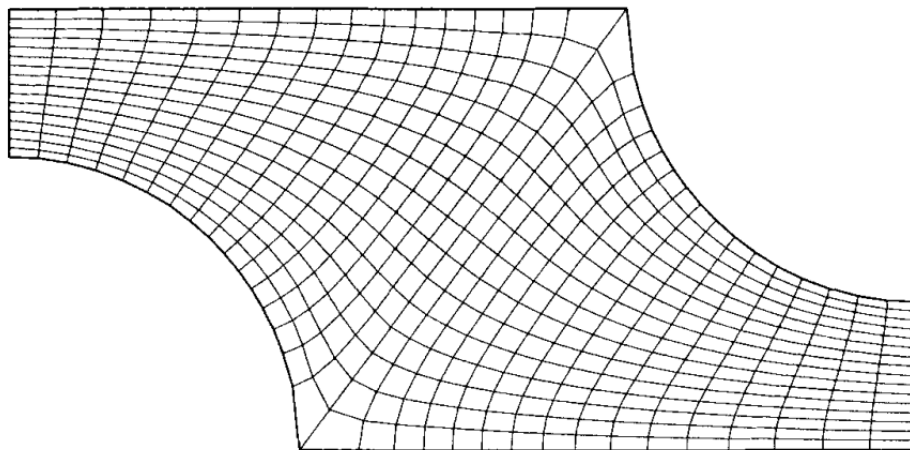
computacionais utilizadas em sua geração, os critérios de qualidade e a relevância da métrica y^+ na modelagem de escoamentos turbulentos próximos às paredes.

2.2.1 Malhas Estruturadas e Não Estruturadas

A estrutura da malha utilizada em uma simulação CFD exerce influência direta sobre a precisão numérica, a eficiência computacional e a capacidade de representação da geometria. De forma geral, as malhas podem ser classificadas em estruturadas e não estruturadas, conforme a organização topológica dos elementos de discretização (Ferziger; Perić, 2002).

Uma malha estruturada, representada pela Figura 2, é definida por linhas de grade regulares que se cruzam uma única vez, permitindo que cada ponto seja indexado por coordenadas inteiras (i, j, k) . Essa organização confere simplicidade na programação, reduz o custo computacional e permite o uso de solvers altamente otimizados, dado que a matriz resultante do sistema algébrico apresenta estrutura regular. Contudo, sua limitação está na dificuldade de representar geometrias complexas sem distorções significativas, o que pode comprometer a qualidade da solução (Ferziger; Perić, 2002).

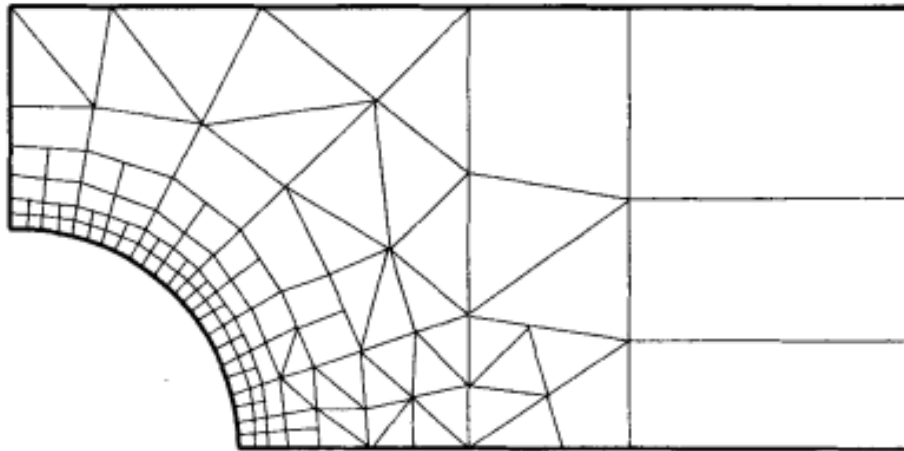
Figura 2 – Exemplo de malha estruturada H-type bidimensional



Fonte – Ferziger e Perić (2002).

Em contrapartida, uma malha não estruturada, representada na Figura 3 é composta por elementos de forma arbitrária, como triângulos e quadriláteros em 2D, ou tetraedros e hexaedros em 3D. A ausência de conectividade regular oferece flexibilidade para representar domínios complexos, permitindo controle local de refinamento e adaptação ao contorno da geometria. Entretanto, essa liberdade estrutural exige estruturas de dados mais complexas e torna o sistema algébrico resultante menos eficiente em termos de memória e tempo de solução (Ferziger; Perić, 2002).

Figura 3 – Exemplo de malha não estruturada com elementos triangulares



Fonte – Ferziger e Perić (2002).

Do ponto de vista da formulação numérica, Ferziger e Perić (2002) destaca que malhas estruturadas permitem discretizações com alto grau de precisão por meio de aproximações centrais simples. Já em malhas não estruturadas, a acurácia depende da qualidade dos elementos e do uso de interpolações mais sofisticadas. Em regiões próximas à parede, por exemplo, o uso de hexaedros ou prismas é preferível para evitar distorções associadas à resolução da camada limite.

A discretização do gradiente em malhas estruturadas é tipicamente obtida por diferenças centrais; em malhas não estruturadas, aproximações equivalentes requerem o cálculo da normal e da área da face entre volumes adjacentes, como mostrado na Equação 1 (Ferziger; Perić, 2002).

$$\frac{\partial \phi}{\partial x} \approx \frac{\phi_E - \phi_P}{\Delta x} \quad (1)$$

em que ϕ é a variável escalar genérica; ϕ_E é o valor no ponto (ou centroide da célula) a leste; ϕ_P é o valor no ponto central da célula considerada; e Δx é a distância, na direção x , entre os centros P e E .

A escolha entre uma malha estruturada ou não estruturada envolve, portanto, um compromisso entre simplicidade computacional e flexibilidade geométrica. Em muitos casos práticos, emprega-se uma malha híbrida, que combina as vantagens de ambos os tipos, utilizando malhas estruturadas em regiões regulares e não estruturadas nas regiões de maior complexidade geométrica (Ferziger; Perić, 2002).

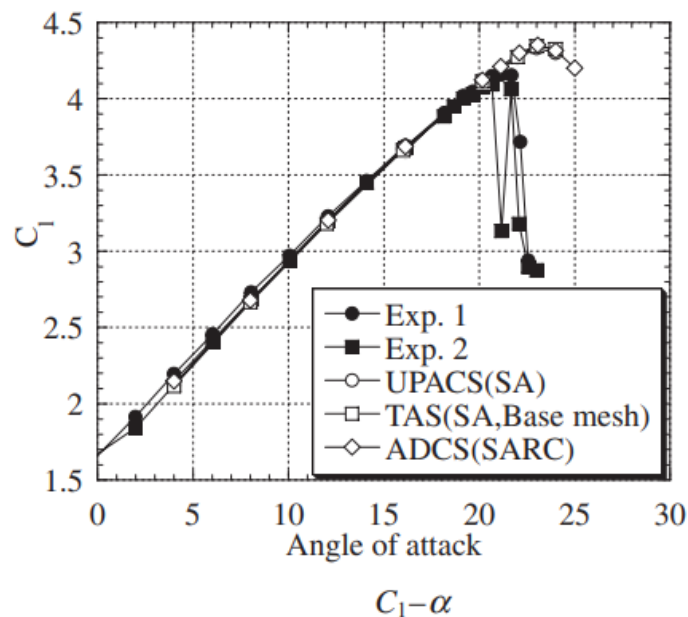
Murayama et al. (2006) realizaram um estudo de validação comparativa entre códigos CFD que utilizam malhas estruturadas e não estruturadas, com aplicação no escoamento ao redor de um aerofólio com três elementos (slat, asa principal e flap). O objetivo principal foi avaliar a precisão dessas abordagens na predição das forças

aerodinâmicas e da física do escoamento em configurações com alta complexidade geométrica.

O estudo utilizou três códigos distintos: dois baseados em malhas estruturadas (UPACS e ADCS) e um baseado em malha não estruturada (TAS). Para garantir a equivalência de comparação, todos utilizaram o mesmo modelo de turbulência (Spalart-Allmaras).

Uma das principais conclusões foi que o coeficiente de sustentação (C_L) apresentou variações inferiores a 1% entre os três códigos, indicando boa robustez na predição global da força vertical, independentemente da topologia da malha, como mostrado na Figura 4. Entretanto, o coeficiente de arrasto (C_D) foi significativamente mais sensível, apresentando diferenças de até 5% entre códigos estruturados e até 10% em malhas não estruturadas menos refinadas.

Figura 4 – Comparação entre C_L e ângulo de ataque para diferentes topologias de malha



Fonte – Murayama et al. (2006).

Observou-se também que a região da esteira gerada pelo slat e pela asa principal apresentava elevada sensibilidade à densidade de pontos na malha não estruturada. Utilizando estratégias de refinamento local baseadas em incremento de entropia como critério, foi possível reduzir o erro no C_D em até 30%, tornando os resultados comparáveis aos obtidos com malha estruturada. A Tabela 1 apresenta a progressão de refinamento empregada (Murayama et al., 2006):

A Figura 5 ilustra a melhoria na previsão do arrasto conforme a malha é refinada. Nota-se que apenas a partir da Mesh 3 os resultados de C_D se aproximam daqueles obtidos com malhas estruturadas. Isso evidencia a importância da distri-

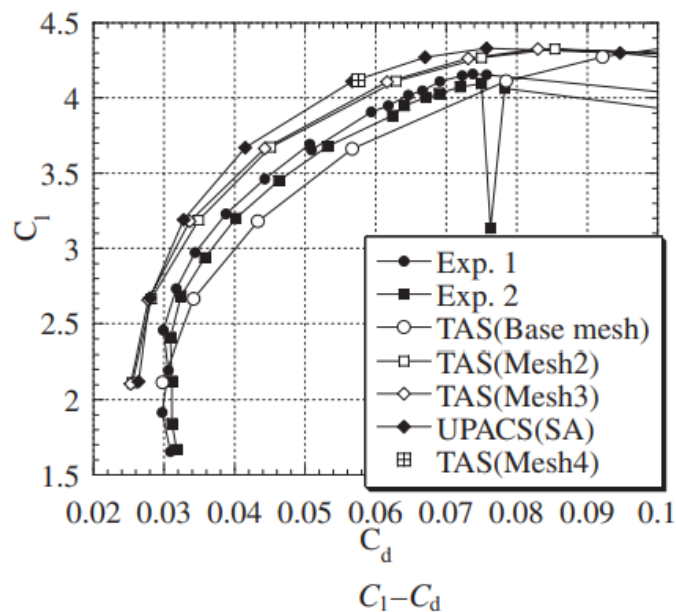
Quadro 1 – Resumo das malhas não estruturadas.

Malha	Pontos	Regiões refinadas
Malha 1	45.997	Malha base, sem refino
Malha 2	75.083	Região externa (50 chords)
Malha 3	117.621	Esteira ($\alpha = 4.01^\circ$)
Malha 4	170.751	Esteira e região do slat ($\alpha = 20.18^\circ$)

Fonte – Murayama et al. (2006).

buição estratégica de pontos em regiões com gradientes elevados, como esteiras e cavidades de slats (Murayama et al., 2006).

Figura 5 – Comparação entre C_L e C_D para diferentes densidades de malha não estruturada



Fonte – Murayama et al. (2006).

A principal vantagem da malha não estruturada observada no estudo foi o tempo de geração. Enquanto a malha estruturada multiblocos demandou cerca de um mês de trabalho manual, a malha não estruturada foi gerada em poucos dias. Essa diferença reforça o potencial das malhas não estruturadas em aplicações industriais, desde que acompanhadas de estratégias robustas de controle de qualidade e refinamento adaptativo (Murayama et al., 2006).

2.2.2 Ferramentas de Geração de Malha

Uma boa ferramenta de geração de malha deve ser capaz de gerar elementos que permitam a aplicação correta dos métodos numéricos, respeitando as propriedades físicas e matemáticas das equações de Navier-Stokes, além de possibilitar refina-

mentos locais e controle de qualidade geométrica dos elementos gerados (Ferziger; Perić, 2002).

De acordo com Ferziger e Perić (2002), uma malha ideal deve satisfazer diversos critérios simultaneamente: adaptação à geometria, alta ortogonalidade entre faces, suavidade na transição entre tamanhos de células e uma razão de aspecto adequada entre dimensões das células adjacentes. Ferramentas computacionais devem, portanto, incorporar algoritmos que respeitem essas diretrizes.

É também essencial que a ferramenta de geração de malha permita a manipulação eficiente de estruturas de dados, especialmente para problemas tridimensionais. Por exemplo, na representação matricial dos sistemas algébricos associados às malhas, a estrutura esparsa das matrizes deve ser explorada para reduzir o consumo de memória e acelerar os cálculos (Ferziger; Perić, 2002).

Vale destacar que ferramentas robustas devem permitir refinamentos adaptativos com base em critérios físicos ou numéricos. Um exemplo clássico é o uso da métrica de erro ou do gradiente da variável de interesse para refinar regiões com grande variação, como zonas de separação ou camadas limite. Essa funcionalidade melhora a resolução local sem aumentar excessivamente o número total de elementos da malha, o que seria computacionalmente inviável (Ferziger; Perić, 2002).

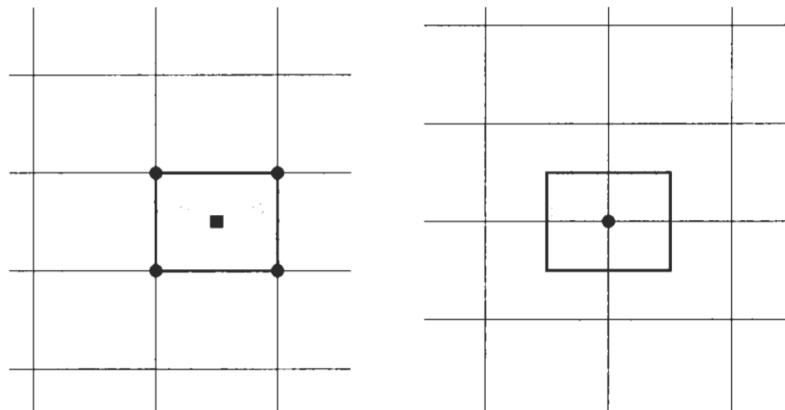
Entre os métodos de discretização espacial mais utilizados destaca-se o Método dos Volumes Finitos (FVM), que utiliza diretamente a forma integral das equações de conservação. Esse método é particularmente vantajoso por garantir, de forma natural, a conservação global das quantidades físicas ao considerar explicitamente os fluxos entre volumes adjacentes (Blazek, 2015). A equação geral de conservação na forma integral é expressa por:

$$\frac{d}{dt} \int_{\Omega} \mathbf{U} d\Omega + \int_{\partial\Omega} (\mathbf{F}_c - \mathbf{F}_d) \cdot \mathbf{n} dS = \int_{\Omega} \mathbf{Q} d\Omega \quad (2)$$

onde \mathbf{U} representa o vetor de variáveis conservativas, \mathbf{F}_c e \mathbf{F}_d são os vetores de fluxos convectivos e difusivos, respectivamente, \mathbf{n} é o vetor normal à superfície dS do volume de controle Ω , e \mathbf{Q} é o vetor de fontes volumétricas.

A definição precisa das variáveis no espaço computacional depende da forma como os volumes de controle são construídos. No esquema *cell-centered*, as variáveis são armazenadas no centro geométrico das células, enquanto no *cell-vertex*, os valores são alocados nos vértices da malha, exigindo a construção de volumes ao redor dos nós (Blazek, 2015). A Figura 6 ilustra essa diferença.

Figura 6 – Esquemas de discretização: cell-centered (à esquerda) e cell-vertex (à direita)



Fonte – Murayama et al. (2006).

Além da construção geométrica, a forma como os fluxos são computados nas faces das células afeta diretamente a precisão e a estabilidade da simulação. Diferentes esquemas numéricos podem ser utilizados, como centrais ou *upwind*, cada um com implicações específicas sobre a dissipação e dispersão numérica do método (Blazek, 2015).

Esquemas centrais realizam a interpolação das variáveis considerando médias entre os valores das células adjacentes à face, resultando em menor dissipação numérica e maior precisão em regiões com gradientes suaves. No entanto, podem apresentar instabilidades em escoamentos com fortes advecções (Blazek, 2015).

Já os esquemas *upwind*, por outro lado, utilizam informações do sentido do escoamento para determinar qual valor da célula será predominante na interpolação, introduzindo dissipação artificial que estabiliza a solução, mas pode suavizar excessivamente gradientes e estruturas finas do fluxo. A escolha entre esses esquemas depende do equilíbrio desejado entre precisão e robustez numérica para o problema em questão (Blazek, 2015).

A precisão espacial do método depende também da ordem do esquema utilizado. O erro de truncamento, definido como a diferença entre a solução exata da equação diferencial e a solução aproximada obtida pelo método numérico, está relacionado à forma como as derivadas são discretizadas e ao tamanho da célula, Δx (Blazek, 2015).

No caso de uma aproximação de primeira ordem, o erro de truncamento é proporcional a Δx , enquanto esquemas de ordem superior convergem mais rapidamente com o refinamento da malha. De modo geral, a relação entre o erro de truncamento e o tamanho da célula pode ser expressa por (Blazek, 2015):

$$\sim \mathcal{O}(\Delta x^p) \quad (3)$$

em que Δx representa o tamanho característico da célula e p é a ordem do esquema numérico utilizado. Esquemas de ordem superior ($p > 1$) podem ser empregados para aumentar a acurácia, mas em geral requerem malhas com qualidade geométrica mais rigorosa, o que reforça a importância de ferramentas robustas de geração de malha para garantir a integridade dos cálculos numéricos (Blazek, 2015).

Portanto, a função principal de uma ferramenta de geração de malha é garantir que a discretização espacial do domínio físico seja compatível com os requisitos do método numérico adotado. A adequação entre a geometria da malha, características do escoamento e esquema numérico é crucial para o sucesso da simulação CFD (Blazek, 2015).

2.2.3 Critérios de Qualidade de Malha e Refinamento

Com o objetivo de avaliar a qualidade de uma malha computacional, adotam-se métricas específicas aplicadas às suas células, buscando padrões geométricos bem definidos. Em malhas não estruturadas, critérios como razão de aspecto, distorção angular, ortogonalidade e suavidade na transição entre elementos são avaliados para minimizar erros de truncamento e garantir a estabilidade numérica da solução (Blazek, 2015).

A razão de aspecto é definida como a razão entre os maiores e menores comprimentos característicos da célula, sendo esse comprimento uma medida representativa do tamanho local da célula em uma direção específica. Em elementos hexaédricos, por exemplo, o comprimento característico pode ser representado pelas dimensões das arestas principais, enquanto em elementos tetraédricos pode-se utilizar o raio da esfera inscrita ou a razão entre a maior e a menor altura do volume (Blazek, 2015).

A ortogonalidade refere-se ao ângulo entre o vetor normal à face de uma célula e o vetor que conecta os centros das células adjacentes. Essa propriedade é essencial para garantir que os fluxos normal e tangencial sejam corretamente representados pelas discretizações centradas. Em geometrias complexas ou com malhas não estruturadas, desvios significativos da ortogonalidade aumentam os erros de interpolação e tornam necessário o uso de correções adicionais nos esquemas de cálculo de fluxos, elevando o custo computacional (Blazek, 2015).

A suavidade na variação do tamanho das células, conhecida como smoothness, evita oscilações numéricas causadas por transições abruptas de escala. Essa propriedade pode ser quantificada por métricas que avaliam a variação do tamanho médio das arestas vizinhas, garantindo uma adaptação progressiva da malha às características do escoamento (Blazek, 2015).

Elementos de má qualidade, como os denominados “slivers” (tetraedros muito achatados com volume próximo de zero) ou “wedges” degenerados (prismas com vértices quase colineares), são comuns em malhas tridimensionais não estruturadas e devem ser evitados. Tais elementos prejudicam a estabilidade numérica e podem comprometer a convergência da simulação. Uma técnica comum para corrigir localmente essas imperfeições é a reconfiguração topológica, conhecida como “troca de arestas” (edge-swapping), que modifica a conectividade entre nós para melhorar a qualidade sem alterar significativamente a malha global (Blazek, 2015).

Refinamento de malha é a principal técnica para redução do erro de discretização, sendo possível de forma global, por zonas, ou adaptativa. Estratégias adaptativas utilizam indicadores físicos, como gradiente de pressão ou entropia, para aumentar a densidade de nós apenas onde necessário, maximizando a eficiência computacional (Blazek, 2015).

Malhas anisotrópicas são aquelas cujas células apresentam alongamento excessivo em uma direção preferencial, sendo utilizadas especialmente em regiões com gradientes intensos em uma única direção, como a camada limite próxima às superfícies sólidas. Esse tipo de malha permite aumentar a resolução local do escoamento na direção perpendicular à parede, sem necessidade de refinar todas as direções igualmente, reduzindo o número total de elementos e o custo computacional (Blazek, 2015).

A escolha da estratégia de refinamento deve levar em conta a ordem do esquema numérico. Para métodos de segunda ordem, o erro de truncamento se comporta como $\mathcal{O}(\Delta x^2)$, sendo possível estimar o erro por comparação entre soluções com diferentes níveis de refinamento:

$$E = \phi_h - \phi_{h/2} \sim \mathcal{O}(h^p) \quad (4)$$

onde p é a ordem do método e ϕ é a variável simulada na malha com passo h . Essa equação serve como base para a validação por convergência espacial (Blazek, 2015).

2.2.4 Métrica y^+

Para a correta resolução do escoamento turbulento nas proximidades de superfícies sólidas, utiliza-se a métrica adimensional y^+ , que é definida para cada célula da malha adjacente à parede. O valor de y^+ quantifica, para uma célula específica, a distância normal entre o centro dessa célula e a parede, expressa em unidades adimensionais baseadas na espessura da subcamada viscosa. Assim, y^+ é uma medida que permite avaliar se aquela célula está posicionada de modo adequado para capturar os gradientes viscosos da camada limite naquele ponto (Moukalled et al., 2016).

Segundo Moukalled et al. (2016), o valor de y^+ é definido como:

$$y^+ = \frac{u_\tau \cdot y}{\nu} \quad (5)$$

em que y representa a distância normal da parede até o centro da célula adjacente, $\nu = \mu/\rho$ é a viscosidade cinemática do fluido, e u_τ é a velocidade de fricção, definida por:

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \quad (6)$$

em que τ_w é a tensão de cisalhamento na parede, e ρ é a densidade do fluido. O cálculo de y^+ depende, portanto, das propriedades do fluido e das condições de escoamento na vizinhança da parede.

A principal aplicação da métrica y^+ é a verificação da adequação da malha na resolução da camada limite turbulenta. Essa camada é geralmente dividida em três regiões, cada uma com características distintas de transporte de momento, ilustradas na Figura 7: a subcamada viscosa (ou laminar), a região buffer e a região logarítmica. O valor de y^+ indica em qual dessas regiões está localizada a primeira célula da malha (Wilcox, 2006).

Modelos turbulentos que resolvem completamente a parede, como os modelos de baixo número de Reynolds, requerem que a primeira célula esteja na subcamada viscosa, o que implica valores de y^+ próximos de 1. Esses modelos calculam explicitamente os gradientes de velocidade e as tensões viscosas em toda a extensão da camada limite, incluindo a região mais próxima da parede, onde os efeitos viscosos predominam. O objetivo de uma simulação com esse tipo de modelo é capturar com alta precisão o comportamento do escoamento junto à superfície sólida, permitindo uma predição detalhada dos esforços de cisalhamento e da transferência de calor na parede (Wilcox, 2006).

Por outro lado, modelos que utilizam funções de parede, como o $k-\varepsilon$ padrão, assumem que a célula está na região logarítmica, o que requer valores de y^+ superiores a 30. Nesses modelos, a estrutura da camada limite muito próxima à parede não é resolvida diretamente; em vez disso, utiliza-se uma formulação empírica (função de parede) para estimar o perfil de velocidade e as tensões na região próxima à superfície. O objetivo, nesse caso, é reduzir o custo computacional, permitindo o uso de malhas mais grossas junto à parede, ao custo de uma menor precisão na resolução dos detalhes da subcamada viscosa (Wilcox, 2006).

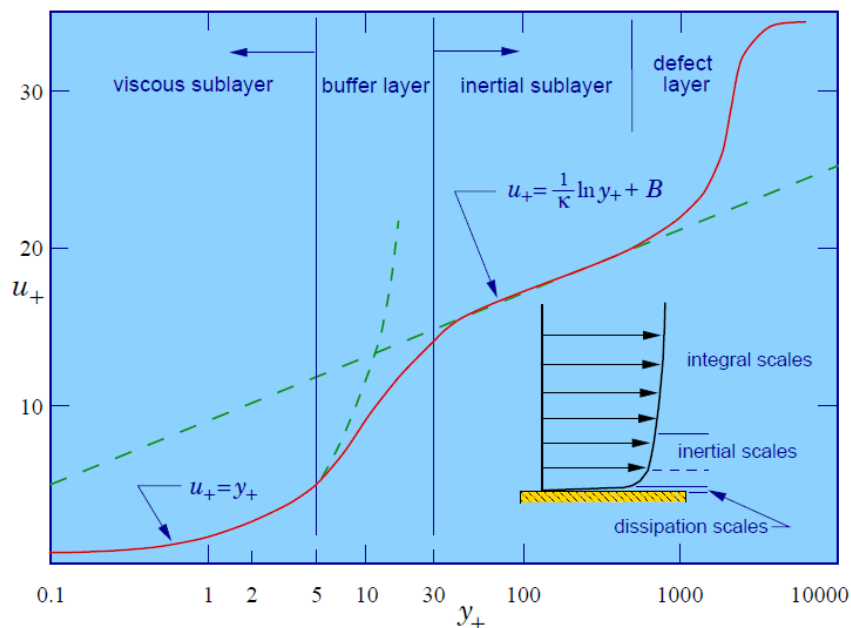
É importante destacar que a análise do y^+ em uma simulação não é feita em um único ponto, mas sim sobre todas as células que estão diretamente adjacentes às superfícies sólidas. A prática comum consiste em calcular a média dos valores de y^+

nessas células, de modo a obter uma estimativa global do posicionamento da malha em relação às paredes (Wilcox, 2006).

Essa média é frequentemente utilizada como critério para avaliar a qualidade do refinamento próximo à parede. Valores muito baixos ou que se situam na chamada região buffer (geralmente entre 5 e 30) devem ser evitados, a menos que o modelo de turbulência tenha sido especificamente validado para operar nesse regime (Wilcox, 2006).

Além disso, o uso de y^+ está associado a formulações adimensionais que descrevem o comportamento de variáveis turbulentas próximas à parede. Uma dessas variáveis é a velocidade adimensional u^+ , definida como a razão entre a velocidade local do fluido, medida a uma distância y da parede, e a velocidade de fricção u_τ , ou seja, $u^+ = u/u_\tau$. Dessa forma, essa grandeza - também ilustrada na Figura 7 - mede o perfil de velocidade normalizado na direção perpendicular à parede e é fundamental para caracterizar a estrutura da camada limite turbulenta (Moukalled et al., 2016).

Figura 7 – Regiões da camada limite turbulenta e a localização das células de malha em função de y^+



Fonte – LearnCAx (2025).

Na região logarítmica da camada limite, o perfil de velocidade segue a lei logarítmica:

$$u^+ = \frac{1}{\kappa} \ln(y^+) + B \quad (7)$$

em que $\kappa \approx 0,41$ é a constante de von Kármán e $B \approx 5,25$ é uma constante empírica, conforme discutido por Moukalled et al. (2016).

Em suma, a métrica y^+ constitui uma ferramenta indispensável na análise de malhas voltadas à modelagem de escoamentos turbulentos. Sua correta interpretação permite garantir a compatibilidade entre o modelo de turbulência adotado e a malha numérica, sendo um fator determinante para a precisão das simulações CFD em zonas de contato com paredes sólidas (Moukalled et al., 2016).

2.3 MODELAGEM DE ASAS DE AERONAVES

A modelagem de asas de aeronaves no design conceitual envolve a consideração de diversos parâmetros que impactam diretamente o desempenho aerodinâmico e estrutural da aeronave. Raymer (1992) descreve como esses parâmetros, como a envergadura, a espessura da asa, o ângulo de ataque e a curvatura, afetam as características de voo e devem ser otimizados para garantir a eficiência. Cada um desses fatores influencia na geração de sustentação e arrasto induzido, além de desempenhar um papel crucial na estabilidade da aeronave.

A envergadura da asa, por exemplo, é um dos parâmetros mais significativos na definição da sustentação e do arrasto da aeronave. Uma maior envergadura pode reduzir o arrasto induzido, melhorando a eficiência aerodinâmica, mas pode exigir um reforço estrutural maior, aumentando o peso da aeronave. A escolha do comprimento da corda da asa e a forma da seção transversal também são fundamentais, pois elas determinam o comportamento do fluxo de ar sobre a asa e a distribuição de pressão ao longo da superfície, afetando diretamente a eficiência do voo (Raymer, 1992).

A modelagem de asas avançadas, como as asas C, é outra consideração importante no design conceitual. Essas asas apresentam uma geometria não convencional que oferece benefícios em termos de redução de arrasto e aumento da sustentação. A curvatura (camber) e o ângulo de ataque são ajustados para maximizar a performance aerodinâmica nessas configurações. A interação entre esses parâmetros deve ser cuidadosamente analisada para otimizar o desempenho em diferentes regimes de voo (Raymer, 1992).

Além de aspectos aerodinâmicos, a modelagem estrutural da asa deve considerar a carga de flexão e os momentos que a asa pode sofrer durante o voo. Raymer (1992) descreve como o comportamento estrutural das asas, que inclui o cálculo de tensões e deformações, pode ser simplificado usando técnicas de modelagem por placas equivalentes. Essas técnicas permitem que as propriedades estruturais da asa, como a espessura do material e a disposição das nervuras, sejam analisadas sem a necessidade de simulações extremamente complexas.

Em termos de otimização do design, o autor destaca a importância de combinar a análise empírica simplificada, utilizada nas fases iniciais de design, com a análise mais detalhada através de elementos finitos nas fases posteriores. Isso per-

mite uma avaliação mais rápida de várias configurações de asas, sem comprometer a precisão dos resultados nas etapas mais avançadas do projeto.

A transição entre os métodos empíricos simplificados e análises mais detalhadas é fundamental para o avanço no design de aeronaves. A abordagem de Raymer (1992) oferece uma visão abrangente dos parâmetros de design de asas, porém, para garantir um processo de otimização mais eficiente, é necessário incluir ferramentas computacionais que tornem possível lidar com geometrias mais complexas e dados estruturais. Nesse contexto, o trabalho de Giles (1995) propõe melhorias no código de análise estrutural ELAPS, integrando métodos de modelagem mais robustos e de rápida execução, ideais para as fases conceituais do design.

O estudo de Giles (1995) foca no aprimoramento de um método computacional para gerar dados estruturais de asas, com o objetivo de reduzir o tempo de modelagem e integrar esses dados eficientemente em códigos multidisciplinares. O código ELAPS (Equivalent Laminated Plate Solution) foi aprimorado para lidar com geometrias avançadas, como as asas C e winglets, e para modelar componentes internos das asas, como nervuras e longarinas. As melhorias feitas no código permitem simular com precisão configurações complexas, proporcionando um ganho significativo na precisão dos resultados (Raymer, 1992).

Ao adicionar capacidades para modelar efeitos de cisalhamento transversal e usar funções de deslocamento múltiplas, o código aprimorado apresenta resultados mais precisos nas análises das tensões e deformações das estruturas das asas. Esses aprimoramentos permitem uma análise mais detalhada de configurações de asas em um tempo reduzido, o que é essencial durante a fase de design conceitual, quando diversas alternativas precisam ser avaliadas rapidamente. Essas melhorias contribuem significativamente para otimizar o desempenho estrutural sem comprometer a eficiência computacional (Giles, 1995).

2.3.1 Séries NACA

O desenvolvimento sistemático de aerofólios pela NACA foi um marco na história da engenharia aeronáutica, visto que, até então, a escolha de perfis era baseada em métodos empíricos e abordagem por tentativa e erro. A criação das séries NACA representou um avanço técnico ao permitir o estudo de geometrias padronizadas, cujas características aerodinâmicas foram extensivamente testadas em túneis de vento com baixos níveis de turbulência (Somers, 1992).

As primeiras séries NACA, como as de quatro e cinco dígitos, foram concebidas para operar com escoamento turbulento, fornecendo altos coeficientes de sustentação, embora com maiores coeficientes de arrasto. Em seguida, com o avanço na compreensão da camada limite, surgiram as séries 6, 7 e superiores, otimizadas para

promover escoamento laminar por maior extensão da corda, reduzindo significativamente o arrasto em condições ideais de superfície (Somers, 1992).

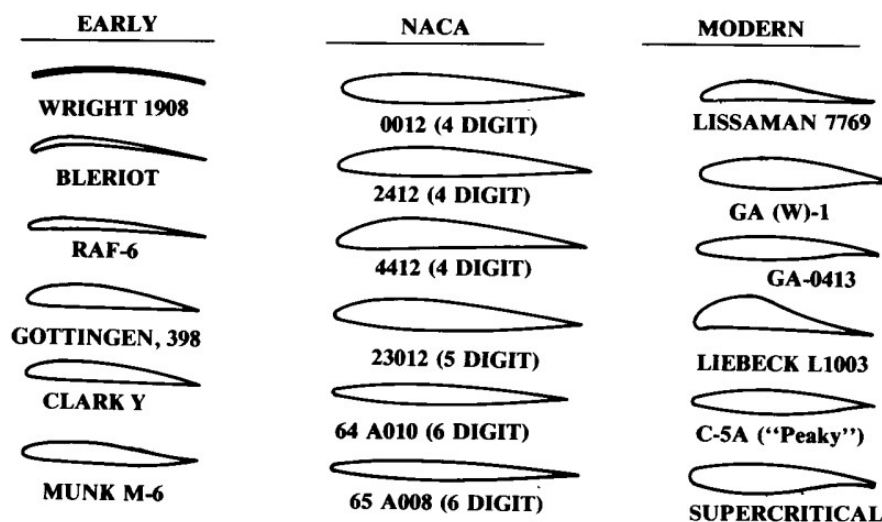
Cada série NACA é definida por uma fórmula que codifica parâmetros geométricos como curvatura máxima, sua posição ao longo da corda e espessura relativa do perfil. Essa nomenclatura padronizada simplifica a seleção de aerofólios no projeto preliminar, permitindo prever com boa precisão suas características básicas antes de simulações mais avançadas ou testes experimentais (Somers, 1992).

O uso de perfis NACA é particularmente útil em fases iniciais de projeto devido à disponibilidade de dados experimentais confiáveis e ao comportamento aerodinâmico previsível. Além disso, sua representação matemática direta permite fácil integração com ferramentas computacionais de modelagem e análise CFD, sendo amplamente utilizada para validação de métodos numéricos (Somers, 1992).

No entanto, a adoção de perfis NACA apresenta limitações quando comparada a aerofólios otimizados via métodos inversos ou parametrizações modernas. As geometrias fixas e o foco em condições de escoamento ideais restringem sua aplicabilidade em contextos mais exigentes, como regimes transônicos ou aeronaves com requisitos de performance específicos. Nesses casos, métodos baseados em otimização numérica ou técnicas como CST tendem a fornecer melhores resultados (Somers, 1992).

A Figura 8 ilustra a comparação entre perfis NACA e outros tipos de aerofólios, destacando as diferenças em termos de curvatura e espessura.

Figura 8 – Comparação entre modelos comuns de aerofólios e séries NACA



Fonte – Raymer (1992).

Apesar dessas limitações, as séries NACA continuam relevantes em contextos educacionais, aplicações de baixa velocidade e validação computacional. Perfis clássicos como o NACA 0012 ou 2412 ainda são amplamente utilizados em estudos

experimentais e em projetos que exigem simplicidade construtiva e resultados bem documentados na literatura (Somers, 1992).

2.3.2 Geometria e Parametrização de Asas

A parametrização de asas de aeronaves é fundamental no design aerodinâmico, pois permite representar matematicamente a geometria da asa para otimização e análise. Durante o desenvolvimento, diferentes configurações de asas precisam ser testadas para garantir desempenho adequado; dessa forma, a escolha de um método eficiente facilita a criação de modelos que representem com precisão as características aerodinâmicas da asa, otimizando o tempo e recursos computacionais (Bloor; Wilson, 1995).

Com o avanço das técnicas numéricas e da capacidade computacional, análises aerodinâmicas complexas passaram a ser feitas nas fases iniciais do design. Bloor e Wilson (1995) observaram que métodos numéricos, antes usados em testes físicos, poderiam ser aplicados na otimização do design, permitindo a avaliação de configurações geométricas por simulação antes da construção física, economizando tempo e custos.

As abordagens numéricas para otimização aerodinâmica podem ser classificadas como design inverso ou otimização numérica direta. No design inverso, parte-se de uma distribuição de pressão desejada e busca-se a geometria correspondente. Na otimização numérica direta, o método consiste em acoplar uma análise aerodinâmica a um esquema de minimização numérica, calculando uma função objetivo que caracteriza o desempenho aerodinâmico (Bloor; Wilson, 1995).

A fim de investigar os impactos da escolha do método de parametrização na eficiência e precisão da otimização aerodinâmica, Mousavi et al. (2007) realizaram um estudo comparativo entre três abordagens distintas: pontos de malha, superfícies B-spline e o método Class Function / Shape Function Transformation (CST). Cada técnica foi aplicada a problemas de design inverso e minimização de arrasto em geometrias tridimensionais típicas da engenharia aeronáutica, permitindo a avaliação do desempenho relativo de cada uma quanto à qualidade dos resultados e ao custo computacional envolvido. Os autores destacaram as vantagens e limitações de cada abordagem, oferecendo subsídios valiosos para a escolha do método mais adequado em contextos distintos de projeto aerodinâmico.

A parametrização com pontos de malha é simples de implementar, mas pode se tornar limitada devido ao grande número de variáveis necessárias para representar a geometria de forma precisa. A independência dos pontos pode gerar superfícies não contínuas, resultando em problemas de estabilidade numérica durante a otimização. Embora flexível, sua aplicabilidade é restrita pela quantidade de variáveis de design necessárias (Mousavi et al., 2007).

As superfícies B-spline oferecem maior controle sobre a geometria com um número reduzido de variáveis de design. A precisão melhora à medida que o número de pontos de controle aumenta, mantendo a suavidade das superfícies e facilitando a otimização com métodos numéricos avançados. No entanto, Mousavi et al. (2007) apontam que a precisão pode ser comprometida se o número de pontos de controle for insuficiente, recomendando-se pelo menos 32 pontos para uma representação precisa da geometria da asa.

O método CST (Class Function / Shape Function Transformation) foi desenvolvido para superar limitações de parametrização em geometrias com grande curvatura, como as bordas de ataque e de fuga das asas. Ele combina uma função de classe, que descreve a geometria básica, com uma função de forma, que ajusta localmente os parâmetros de design. A principal vantagem do CST é sua capacidade de representar formas complexas de forma eficiente, utilizando um número reduzido de variáveis de design, ideal para otimizações rápidas e eficazes (Mousavi et al., 2007).

Com o avanço das técnicas de parametrização e otimização estrutural, novas abordagens têm sido exploradas para lidar com a complexidade das formas geométricas e a distribuição de material. Entre essas técnicas, destacam-se o sistema celular Lindenmayer (LC), os diagramas de Voronoi (VD) e a representação gráfica bijetiva (BG), que oferecem soluções inovadoras para a criação e otimização de estruturas com um número reduzido de variáveis de design (Dexl et al., 2022).

O sistema celular Lindenmayer (LC) é inspirado no crescimento de plantas e utiliza um conjunto de regras de produção para gerar topologias complexas. Nesse sistema, uma célula inicial se divide de acordo com regras definidas, permitindo o crescimento de uma estrutura a partir de elementos simples (Dexl et al., 2022).

É particularmente útil para otimização de topologia, já que pode gerar uma grande variedade de formas com um número reduzido de variáveis. No entanto, a complexidade do controle das variáveis aumenta com a necessidade de modelar detalhadamente a estrutura, o que pode tornar sua implementação mais complexa (Dexl et al., 2022).

Os diagramas de Voronoi (VD) são usados para dividir um espaço em células baseadas em um conjunto de pontos, com cada célula associada ao ponto mais próximo. Essa técnica é eficaz para otimizar estruturas em que a distribuição do material e a conectividade entre os elementos são importantes. A principal vantagem do Voronoi é sua capacidade de simplificar a descrição da estrutura e a facilidade de implementação, já que a definição das células é diretamente associada a pontos de controle (Dexl et al., 2022).

A representação gráfica bijetiva (BG) utiliza pontos e arestas para formar um grafo que descreve a estrutura. Essa técnica é altamente flexível, permitindo a criação de diferentes padrões de conectividade entre os elementos estruturais. A principal

vantagem é o acesso direto aos pontos e arestas, o que facilita a manipulação da estrutura e a adaptação de seus elementos durante a otimização. No entanto, esse método pode resultar em problemas de conectividade e maior complexidade na definição das variáveis de design (Dexl et al., 2022).

O estudo de Dexl et al. (2022) concluiu que os métodos baseados em diagramas de Voronoi e sistemas celulares Lindenmayer são mais eficazes do que a representação gráfica bijetiva em termos de otimização e diversidade dos resultados. A principal vantagem dos diagramas de Voronoi foi sua capacidade de garantir maior convergência e reprodutibilidade dos resultados, com a estrutura gerada mostrando uma adaptação precisa aos objetivos da otimização, especialmente no que diz respeito à minimização do arrasto e à adaptação ao perfil aerodinâmico (Dexl et al., 2022).

Por outro lado, o sistema Lindenmayer e os diagramas de Voronoi foram mais eficazes em termos de adaptabilidade estrutural, conseguindo gerar formas mais complexas e com melhor desempenho em relação aos objetivos. A representação gráfica bijetiva, embora útil, resultou em soluções menos diversificadas e com maior probabilidade de problemas de conectividade, o que afetou sua eficiência em otimizações mais complexas (Dexl et al., 2022).

Em resumo, tanto os diagramas de Voronoi quanto os sistemas celulares Lindenmayer provaram ser métodos robustos e eficazes para parametrização de estruturas em otimização de asas morphing, com o Voronoi apresentando a melhor reprodutibilidade e convergência em múltiplas execuções do processo de otimização (Dexl et al., 2022).

Embora o estudo de Mousavi et al. (2007) se concentre em técnicas de parametrização aplicadas a aerofólios e otimização aerodinâmica de formas 2D e 3D, o artigo de Dexl et al. (2022) amplia esse escopo, focando em estruturas morfáveis ativas e seu uso em design multidisciplinar. Enquanto o primeiro explora a eficiência de métodos como B-splines, CST e pontos de malha em problemas clássicos de minimização de arrasto, o segundo trata da parametrização de geometrias de asas em um contexto de tecnologias emergentes, como asas com morfagem ativa, que exigem uma abordagem mais dinâmica e adaptativa.

As diferenças de época também são evidentes, já que Mousavi et al. (2007) trabalharam com limitações computacionais e técnicas de parametrização mais simples, enquanto Dexl et al. (2022), com a evolução das tecnologias computacionais e algoritmos de otimização, exploram novas fronteiras no design de asas, considerando impactos dinâmicos e desempenho aerodinâmico adaptativo. Essas distinções refletem a evolução do campo, onde a parametrização foi aprimorada para incorporar novas variáveis e técnicas de otimização mais avançadas, adequadas às necessidades de projetos mais complexos.

2.4 MODELOS DE TURBULÊNCIA EM CFD

Na dinâmica dos fluidos, turbulência é um regime de escoamento caracterizado por flutuações caóticas e irregulares nas variáveis do fluido, como a velocidade, pressão e temperatura. Em um escoamento turbulento, o fluido apresenta movimentos aleatórios, de pequena e grande escala, que resultam em vórtices e redemoinhos. Isso contrasta com o escoamento laminar, onde as partículas do fluido se movem de maneira suave e ordenada, seguindo trajetórias previsíveis (Versteeg; Malalasekera, 2007).

A turbulência ocorre quando o número de Reynolds — uma medida que relaciona as forças inerciais e viscosas em um fluido — atinge um valor crítico. Todos os fluxos encontrados na prática de engenharia, desde os simples, como jatos bidimensionais, vórtices, fluxos em tubulações e camadas limite em placas planas, até os mais complicados em três dimensões, tornam-se instáveis acima de um determinado número de Reynolds (Versteeg; Malalasekera, 2007).

Em casos simples, as equações de continuidade e de Navier–Stokes podem ser resolvidas analiticamente. No entanto, fluxos mais complexos devem ser tratados numericamente com técnicas de CFD, sem a necessidade de aproximações adicionais. Essas abordagens permitem a análise de fluxos que não podem ser resolvidos de forma exata, principalmente quando a geometria ou as condições são mais complicadas (Versteeg; Malalasekera, 2007).

A maioria dos fluxos de interesse para a engenharia é turbulenta, tornando o regime de turbulência não apenas de interesse teórico, mas essencial para a prática da engenharia de fluidos. Os engenheiros precisam de ferramentas viáveis para representar os efeitos da turbulência, o que torna o estudo do comportamento e da modelagem da turbulência em CFD crucial (Versteeg; Malalasekera, 2007).

2.4.1 Escolha do Modelo de Turbulência para Simulações de Asas

A modelagem da turbulência em escoamentos aerodinâmicos representa um dos maiores desafios na aplicação de CFD, devido à complexidade intrínseca dos fluxos turbulentos e à necessidade de soluções viáveis computacionalmente. Para contornar a impossibilidade de resolver todas as escalas da turbulência em aplicações de engenharia, empregam-se modelos matemáticos que aproximam o comportamento médio do escoamento (Pope, 2000).

A escolha de um modelo de turbulência adequado está diretamente relacionada à natureza do escoamento em estudo, às características da geometria, ao regime de Reynolds e aos objetivos da simulação. Em simulações de asas, é fundamental representar corretamente os efeitos das camadas-limites, separações de fluxo e gradientes de pressão, o que impõe exigências específicas sobre os modelos

empregados (Pope, 2000).

Segundo Pope (2000), tais modelos devem ser não apenas fundamentados em princípios físicos sólidos, mas também consistentes com as propriedades fundamentais das equações de Navier-Stokes. Ainda, o autor destaca que, entre os critérios de avaliação de modelos, estão a precisão na previsão das tensões de Reynolds, a robustez numérica e a capacidade de adaptação a geometrias complexas.

A abordagem estatística, baseada nas equações de Reynolds-Averaged Navier–Stokes (RANS), é amplamente adotada devido à sua viabilidade computacional em simulações de engenharia. Nessa categoria, os modelos de viscosidade turbulenta — como os modelos $k-\varepsilon$ e $k-\omega$ — utilizam hipóteses simplificadoras para estimar o efeito médio da turbulência sobre o escoamento. De acordo com Pope (2000), embora esses modelos não descrevam diretamente as tensões de Reynolds, sua simplicidade permite a aplicação generalizada com custo computacional reduzido.

Por outro lado, modelos mais avançados, como os modelos de tensões de Reynolds (RSM), resolvem equações adicionais para cada componente do tensor de tensões turbulentas, oferecendo maior fidelidade física. Esses modelos são particularmente úteis em escoamentos com anisotropia significativa, como aqueles encontrados em asas com ângulos de ataque elevados ou presença de separações de fluxo. No entanto, o custo computacional e a complexidade numérica aumentam consideravelmente (Pope, 2000).

A escolha entre modelos mais simples, como $k-\varepsilon$ ou Spalart–Allmaras, e abordagens mais sofisticadas, como RSM ou LES, envolve um equilíbrio entre precisão e viabilidade computacional. Em contextos de projeto preliminar ou análise de múltiplas configurações geométricas, a eficiência dos modelos RANS de duas equações pode ser mais vantajosa. Pope (2000) ressalta que, apesar de suas limitações, esses modelos são suficientemente robustos para capturar os principais efeitos da turbulência em escoamentos externos sobre superfícies aerodinâmicas.

Além disso, a forma de tratamento próximo à parede tem papel crucial na precisão das simulações. Estratégias como funções de parede ou modelos com resolução da camada viscosa afetam diretamente a previsão do arrasto e do ponto de separação. A correta implementação dessas condições é especialmente relevante em simulações de perfis aerodinâmicos, onde os gradientes de velocidade são intensos nas regiões adjacentes à superfície sólida (Pope, 2000).

Os modelos de turbulência podem ser classificados segundo a abordagem adotada para representar as escalas turbulentas. O método DNS (Direct Numerical Simulation) resolve diretamente todas as escalas do escoamento, desde as grandes estruturas até as menores escalas dissipativas. Para isso, é necessário que o tamanho da malha e o passo temporal sejam suficientemente pequenos para capturar os efeitos viscosos e inerciais, resultando em um custo computacional elevado e aplicações limi-

tadas a geometrias simples e baixos números de Reynolds (Versteeg; Malalasekera, 2007).

Como alternativa mais viável, a simulação LES (Large Eddy Simulation) resolve explicitamente as grandes escalas do escoamento e modela apenas as pequenas escalas turbulentas que não podem ser capturadas pela malha computacional, conhecidas como escalas sub-malha. Ou seja, enquanto as estruturas turbulentas maiores são resolvidas diretamente pelas equações numéricas, as menores — que ocorrem em dimensões inferiores ao tamanho da malha — são representadas por modelos específicos de turbulência (Wilcox, 2006).

A abordagem mais comum em aplicações de engenharia é o uso de modelos RANS (Reynolds-Averaged Navier-Stokes), que resolvem apenas as médias das variáveis de fluxo e modelam os efeitos das flutuações turbulentas por meio de termos adicionais nas equações de transporte. Entre os modelos RANS, destacam-se os modelos de duas equações, como o $k-\varepsilon$ e o $k-\omega$, por seu equilíbrio entre robustez, custo computacional e aplicabilidade (Wilcox, 2006).

O modelo $k-\varepsilon$ é amplamente utilizado em simulações de escoamentos com regiões predominantemente turbulentas, sendo adequado para escoamentos externos com separações moderadas. Suas equações de transporte são dadas por (Pope, 2000):

$$\frac{\partial k}{\partial t} + U_j \frac{\partial k}{\partial x_j} = P_k - \varepsilon + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] \quad (8)$$

$$\frac{\partial \varepsilon}{\partial t} + U_j \frac{\partial \varepsilon}{\partial x_j} = C_{1\varepsilon} \frac{\varepsilon}{k} P_k - C_{2\varepsilon} \frac{\varepsilon^2}{k} + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_j} \right] \quad (9)$$

em que k representa a energia cinética turbulenta, ε é a taxa de dissipação dessa energia, U_j é a componente j da velocidade média do escoamento, x_j é a coordenada espacial na direção j (com soma implícita sobre j), P_k é a taxa de produção de energia cinética turbulenta, ν é a viscosidade cinemática molecular do fluido, σ_k e σ_ε são constantes adimensionais associadas à difusão turbulenta de k e ε , respectivamente, $C_{1\varepsilon}$ e $C_{2\varepsilon}$ são constantes empíricas do modelo $k-\varepsilon$, e ν_t é a viscosidade turbulenta calculada por $\nu_t = C_\mu \frac{k^2}{\varepsilon}$, onde C_μ é uma constante empírica do modelo (Pope, 2000).

Já o modelo $k-\omega$ é mais indicado para regiões com gradientes de pressão adversos e escoamentos internos com separações próximas à parede, sendo definido pelas seguintes equações (Pope, 2000):

$$\frac{\partial k}{\partial t} + U_j \frac{\partial k}{\partial x_j} = P_k - \beta^* k \omega + \frac{\partial}{\partial x_j} \left[\left(\nu + \sigma_k^* \nu_t \right) \frac{\partial k}{\partial x_j} \right] \quad (10)$$

$$\frac{\partial \omega}{\partial t} + U_j \frac{\partial \omega}{\partial x_j} = \alpha \frac{\omega}{k} P_k - \beta \omega^2 + \frac{\partial}{\partial x_j} \left[\left(\nu + \sigma_\omega^* \nu_t \right) \frac{\partial \omega}{\partial x_j} \right] \quad (11)$$

em que ω é a taxa de dissipação específica da energia cinética turbulenta, β^* , α , β , σ_k^* e σ_ω^* são constantes empíricas do modelo $k-\omega$, e os demais termos são definidos conforme no modelo $k-\varepsilon$. A viscosidade turbulenta é calculada por $\nu_t = \frac{k}{\omega}$ (Pope, 2000).

Em aplicações aerodinâmicas de asas, a escolha pragmática recai nos modelos RANS de duas equações pela boa relação entre custo e fidelidade. Em gradientes de pressão adversos e regiões próximas à parede, o $k-\omega$ SST costuma apresentar desempenho superior, enquanto o $k-\varepsilon$ tende a ser mais estável em escoamentos livres; a decisão sobre funções de parede ou resolução da subcamada viscosa deve seguir o objetivo da simulação e a meta de y^+ (Pope, 2000).

Quando há anisotropia marcada das tensões turbulentas — por curvatura, rotação ou separações extensas — os modelos de Tensões de Reynolds (RSM) podem melhorar a previsão das tensões ao custo de maior rigidez numérica e tempo computacional. Sua adoção é mais adequada em estudos focados onde o ganho físico compensa o aumento de complexidade (Pope, 2000).

O LES é indicado para resolver estruturas coerentes e fenômenos de separação com maior detalhe local. Exige malhas e passos de tempo finos, sobretudo junto à parede, o que limita o domínio ou eleva o custo, sendo mais apropriado para recortes da geometria ou investigações de alta fidelidade (Versteeg; Malalasekera, 2007; Pope, 2000).

Abordagens híbridas RANS–LES, como o DES, aplicam RANS nas proximidades da parede e LES fora da camada limite, produzindo um compromisso eficaz em escoamentos com separação relevante. A eficácia depende fortemente da malha e da zona de troca entre os regimes, recomendando-se estudos de sensibilidade para garantir a ativação correta do modo LES (Wilcox, 2006; Pope, 2000).

Se a localização da transição laminar–turbulento afeta C_L , C_D e o ponto de separação, correções transicionais acopladas a RANS aumentam a aderência física sem perder eficiência (Versteeg; Malalasekera, 2007). Em síntese, a seleção do modelo deve alinhar meta física, orçamento computacional e estratégia de malha, adotando RANS de duas equações como base e escalando para RSM, LES ou DES apenas quando a física do caso exigir tal refinamento (Pope, 2000).

2.4.2 Validação de Modelos de Turbulência com Dados Experimentais

Diversos estudos têm comparado modelos de turbulência tradicionais com dados experimentais para avaliar sua precisão na simulação de escoamentos complexos. Em um estudo recente, Huang et al. (2024) realizaram uma análise comparativa entre cinco modelos de turbulência — IDDES, DDES, DES, LES e URANS — aplicados à simulação do escoamento ao redor de um ônibus.

Quadro 2 – Comparação sucinta de modelos de turbulência para simulações de asas

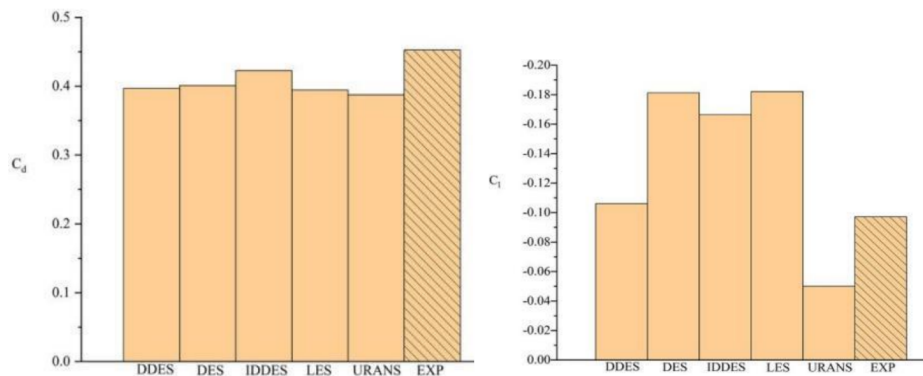
Modelo	Pontos fortes	Limitações e uso recomendado
RANS $k-\varepsilon$	Barato e estável em escoamento livre. Simples de configurar.	Subestima separação e anisotropia. Uso: fases preliminares e casos com separação moderada; funções de parede adequadas.
RANS $k-\omega$ SST	Melhor perto da parede e com gradiente de pressão adverso. Boa previsão de separação incipiente.	Mais sensível no escoamento livre; requer atenção ao y^+ . Uso: aerofólios/asa com efeitos de parede dominantes.
RSM	Captura anisotropia e efeitos de curvatura/rotação.	Custo e rigidez numérica maiores. Uso: ângulos de ataque altos e separações fortes quando RANS de 2 eq. falha.
LES	Resolve estruturas coerentes e transição. Alta fidelidade local.	Custo muito alto; malha e Δt finos, sobretudo junto à parede. Uso: recortes da asa e estudos locais.
DES (RANS– LES)	Compromisso: RANS na parede, LES fora. Bom em separação ampla.	Sensível à malha e à interface RANS/LES; risco de permanecer em RANS. Uso: separação relevante com orçamento intermediário.

Fonte – Pope (2000).

No estudo, os cinco modelos foram avaliados quanto à sua capacidade de prever características aerodinâmicas de um ônibus em escoamento externo. Os testes comparativos incluíram simulações numéricas validadas com dados experimentais de túnel de vento. As métricas avaliadas foram o coeficiente de arrasto (C_d), o coeficiente de sustentação (C_l), campos de pressão, velocidade, linhas de corrente e distribuição de vorticidade (Huang et al., 2024).

O modelo IDDES obteve o melhor desempenho na previsão do coeficiente de arrasto, com $C_d = 0,4226$, apresentando erro de apenas 0,0299 em relação ao valor experimental de $C_d = 0,4525$. Já o modelo URANS subestimou o arrasto significativamente, com valor de $C_d = 0,3877$. Para o coeficiente de sustentação, o modelo DDES apresentou melhor correspondência com o valor experimental ($C_l = -0,0971$), com previsão de $C_l = -0,1061$ — erro absoluto de apenas 0,009. A comparação desses resultados pode ser vista na imagem 9 (Huang et al., 2024).

Figura 9 – Comparação dos valores simulados e experimentais de C_d e C_l entre os modelos de turbulência



Fonte – Huang et al. (2024).

A análise do escoamento na esteira revelou que o modelo IDDES capturou melhor a estrutura dos vórtices gerados na traseira do ônibus. O LES apresentou uma esteira mais uniforme, enquanto o URANS foi incapaz de resolver adequadamente as regiões de baixa pressão e separação de fluxo, resultando em linhas de corrente excessivamente simétricas e ausência de detalhes vorticosos. O DDES mostrou shedding bem definido, mas com menor suavidade no perfil do escoamento (Huang et al., 2024).

Em termos de distribuição de pressão na região traseira, o IDDES capturou valores mais negativos com maior resolução espacial. A distribuição de pressão prevista pelo LES foi mais uniforme, enquanto o URANS não conseguiu reproduzir os gradientes negativos críticos nessa região. Os resultados do artigo demonstram que o IDDES e o LES superaram os demais modelos na previsão dos picos de pressão e sua variação ao longo da superfície do ônibus (Huang et al., 2024).

A visualização da vorticidade com o critério $Q = 1000$ mostrou que o modelo IDDES capturou maior quantidade de vórtices de pequena escala em torno do corpo do ônibus, seguido por DDES e LES. O URANS não conseguiu identificar estruturas coerentes de vórtice. As análises com sondas de pressão em múltiplos planos revelaram que os modelos IDDES e LES apresentaram maiores flutuações de pressão vertical — indicando maior fidelidade na previsão de fenômenos turbulentos não estacionários na esteira (Huang et al., 2024).

Conclui-se que o modelo IDDES apresentou o melhor desempenho global, conciliando precisão na previsão de coeficientes aerodinâmicos com representação detalhada do campo de escoamento. O LES demonstrou maior detalhamento no fluxo, mas com custo computacional mais elevado. O URANS, apesar de eficiente, apresentou erros sistemáticos em regiões de separação e vórtices, limitando sua aplicabilidade a estudos de alta fidelidade em geometria de veículos (Huang et al., 2024).

Em outro estudo, Odier et al. (2017) compararam os modelos RANS e LES com medições experimentais em um turbofan operando em regime nominal. As comparações de perfis médios em três posições axiais mostraram uma concordância razoável entre RANS e LES para todos os resultados.

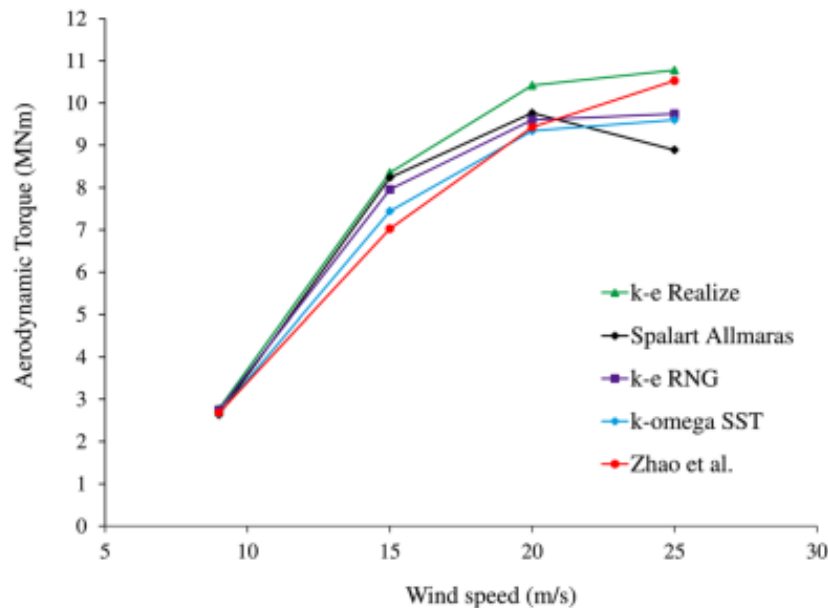
No entanto, observou-se um desvio quando as temperaturas totais numéricas, número de Mach e pressão estática foram comparados aos perfis experimentais, permanecendo uma discrepância a ser explicada. A forma dos perfis numéricos, no entanto, foi consistente com os resultados experimentais (Odier et al., 2017).

Em estudo realizado por Muiruri et al. (2019), quatro modelos de turbulência RANS foram avaliados quanto à capacidade de prever o torque aerodinâmico, o momento fletor na direção da pá (flap-wise bending moment) e a distribuição do coeficiente de pressão sobre uma pá de turbina eólica de 5 MW. Os modelos comparados foram: Spalart–Allmaras, $k-\varepsilon$ RNG, $k-\varepsilon$ Realizable e $k-\omega$ SST.

As simulações foram conduzidas com abordagem de referência em malha móvel (SMRF) em regime estacionário, utilizando ANSYS Fluent, e a validação foi feita com base em resultados numéricos publicados por outros autores, incluindo dados do código FAST.

Para a previsão do torque aerodinâmico, o modelo $k-\omega$ SST apresentou valor de 2664 kNm, em excelente concordância com os 2650–2682 kNm reportados na literatura. Já os modelos $k-\varepsilon$ RNG e $k-\varepsilon$ Realizable superestimaram o torque, com valores de 2740 e 2768 kNm, respectivamente. O modelo Spalart–Allmaras foi considerado adequado para condições de baixa velocidade, mas sua precisão diminuiu em velocidades mais elevadas. Os resultados mostraram que, acima de 15 m/s, o modelo SST manteve boa consistência, enquanto os demais apresentaram desvios crescentes, como mostrado na Figura 10 (Muiruri et al., 2019).

Figura 10 – Torque aerodinâmico previsto por diferentes modelos de turbulência em função da velocidade do vento



Fonte – Muiruri et al. (2019).

Na avaliação do momento fletor flap-wise, o Spalart–Allmaras previu os menores valores em todas as velocidades, o que sugere uma subestimação da carga aerodinâmica. Já os modelos $k-\varepsilon$ Realizable e SST apresentaram comportamento semelhante até 15 m/s, mas divergiram em velocidades mais altas, com o Realizable apresentando flutuações. O modelo SST mostrou uma tendência decrescente do momento após 15 m/s, condizente com estudos prévios que relacionam esse comportamento ao início da separação de fluxo (Muiruri et al., 2019).

Além disso, foram avaliadas as distribuições de coeficiente de pressão c_p em três seções da pá: 32%R, 60%R e 92%R. Os modelos apresentaram boas previsões no lado de pressão da pá, mas diferenças significativas no lado de sucção. Nessas regiões, o modelo $k-\varepsilon$ Realizable tendeu a superestimar c_p , enquanto o SST manteve valores mais conservadores e próximos dos estudos de referência. A variação mais crítica ocorreu na seção de 92%R a 20 m/s, onde o SST apresentou menor dispersão entre previsão e literatura (Muiruri et al., 2019).

Com base nesses resultados, o modelo $k-\omega$ SST foi considerado o mais confiável entre os testados, tanto na previsão do torque quanto na análise de cargas e distribuição de pressão. Sua formulação híbrida, que combina características de $k-\varepsilon$ no escoamento livre e $k-\omega$ próximo à parede, mostrou-se especialmente eficaz na região do bordo de ataque e em condições de escoamento complexo. Como conclusão, o estudo recomenda o uso do modelo SST para simulações de pás de turbina eólica de grande porte operando em regime totalmente turbulento (Muiruri et al., 2019).

Esses estudos evidenciam que a escolha do modelo de turbulência deve considerar o equilíbrio entre precisão e custo computacional. Enquanto modelos como o IDDES e LES oferecem maior fidelidade na captura de estruturas turbulentas complexas, eles demandam maior poder computacional. Por outro lado, modelos como o URANS, embora menos precisos em certas situações, são mais eficientes computacionalmente e podem ser adequados para simulações onde a precisão extrema não é crítica.

2.5 OPENFOAM COMO FERRAMENTA DE CFD

O OpenFOAM (Open Source Field Operation and Manipulation) é uma plataforma computacional de código aberto amplamente adotada para simulações de dinâmica dos fluidos. Seu uso tem crescido significativamente tanto em ambientes acadêmicos quanto industriais devido à sua versatilidade, capacidade de personalização e ausência de custos com licenciamento (Chen et al., 2014).

Desenvolvido originalmente no Imperial College London, o OpenFOAM é baseado no método dos volumes finitos e estruturado em C++ com programação orientada a objetos, o que permite representar equações diferenciais parciais de forma simbólica e flexível. Sua arquitetura modular possibilita a implementação de solvers personalizados e o suporte a uma ampla gama de aplicações, incluindo escoamentos incompressíveis e compressíveis, transferência de calor, escoamentos multifásicos, reações químicas e eletromagnetismo (Chen et al., 2014).

Ao contrário de muitos softwares comerciais, o OpenFOAM é distribuído sob licença GNU GPL, o que não apenas elimina os altos custos de licença — que podem ultrapassar dezenas de milhares de dólares anuais — como também proporciona liberdade para modificar, redistribuir e adaptar os códigos conforme a necessidade da aplicação. Tal liberdade é especialmente relevante no meio acadêmico, onde recursos financeiros são frequentemente limitados (Chen et al., 2014).

A flexibilidade do OpenFOAM, no entanto, vem acompanhada de uma curva de aprendizado significativa. Sem interface gráfica integrada, sua utilização exige familiaridade com sistemas Unix/Linux, organização de diretórios e configuração manual de arquivos. Em contrapartida, essa abordagem oferece ao usuário controle total sobre os parâmetros físicos, algoritmos numéricos e condições de contorno, sendo especialmente atrativa para estudos acadêmicos avançados (Chen et al., 2014).

O ambiente computacional estruturado em torno do OpenFOAM inclui ferramentas para todas as etapas do processo: pré-processamento com 'blockMesh' ou 'snappyHexMesh', solução com solvers baseados em métodos de volumes finitos (FVM), e pós-processamento com o uso de softwares como ParaView. Essa integração proporciona uma cadeia completa de simulação, desde a geometria inicial até a

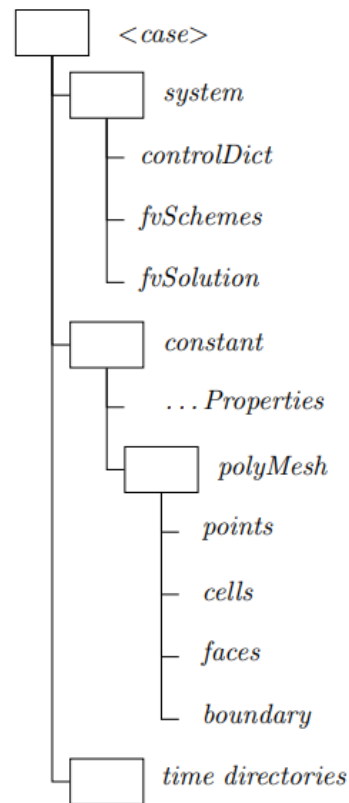
análise dos resultados, tornando o OpenFOAM uma solução independente e autocon-tida (Chen et al., 2014).

Em termos de desempenho, o OpenFOAM permite execução em arquiteturas paralelas de alto desempenho, sendo amplamente utilizado em supercomputadores ao redor do mundo. Um exemplo é a simulação de escoamentos tridimensionais turbu-lentos conduzida pelos próprios autores do artigo de Chen et al. (2014), a qual utilizou 1024 núcleos para resolver, com alta resolução, um problema clássico de cavidade com tampa móvel tridimensional.

Dessa forma, o OpenFOAM não é apenas uma alternativa gratuita aos pa-cotes comerciais, mas uma plataforma de desenvolvimento em constante evolução, apoiada por uma comunidade internacional ativa. Sua natureza aberta e extensível tem permitido avanços significativos em modelagem computacional, tornando-se uma ferramenta essencial na engenharia moderna e nas ciências aplicadas (Chen et al., 2014).

A Figura 11 ilustra a estrutura de diretórios típica de um caso no OpenFOAM. O diretório principal do caso contém subdiretórios essenciais, como '0', 'constant' e 'system'. O diretório '0' armazena os arquivos de condições iniciais e de contorno para cada variável do problema, como velocidade, pressão e temperatura. O diretório 'constant' contém propriedades físicas do fluido, modelos de turbulência e a malha computacional. Já o diretório 'system' abriga arquivos de controle da simulação, in-cluindo parâmetros numéricos, esquemas de discretização e critérios de convergência (Chen et al., 2014).

Figura 11 – Estrutura típica de diretórios em um caso OpenFOAM



Fonte – Chen et al. (2014).

2.5.1 Solvers e Modelos de Turbulência no OpenFOAM

O OpenFOAM disponibiliza um amplo conjunto de solvers padronizados, desenvolvidos para atender a diferentes categorias de escoamentos e fenômenos físicos. Esses solvers são construídos com base no método dos volumes finitos e organizados em estruturas modulares, o que facilita sua adaptação e extensão conforme a complexidade do problema (Chen et al., 2014).

Dentre as categorias disponíveis, destacam-se escoamentos incompressíveis, compressíveis, multifásicos, combustão, transferência de calor, eletromagnetismo e análise de tensões mecânicas. Cada solver é projetado para atuar de forma especializada em uma dessas áreas, permitindo ao usuário selecionar ou modificar a abordagem numérica mais adequada à sua aplicação (Chen et al., 2014).

Seguindo essa organização, destacam-se quatro solvers de base para escoamentos monofásicos: *simpleFoam* e *pimpleFoam* para regimes incompressíveis, e *rhoSimpleFoam* e *rhoPimpleFoam* para regimes compressíveis, cobrindo combinações de análise estacionária e transiente. Todos implementam esquemas segregados com acoplamento pressão-velocidade via correções sucessivas (Patankar, 1980).

A escolha entre escoamento incompressível ou compressível decorre da relevância das variações de densidade no balanço de massa e momento. Quando ρ varia de modo desprezível nas escalas de interesse, adota-se a hipótese incompressível; quando as variações de ρ por efeitos de pressão e temperatura são significativas, impõe-se a formulação compressível com equação de energia (Patankar, 1980).

Regime estacionário implica que as estatísticas ou médias de interesse não dependem explicitamente do tempo e, portanto, busca-se uma solução que satisfaça as equações com derivadas temporais nulas. Regime transiente implica dependência temporal explícita, de modo que as equações são integradas no tempo com passos Δt para capturar a evolução do campo (Patankar, 1980).

O *simpleFoam* resolve escoamentos incompressíveis em regime permanente por meio do algoritmo SIMPLE, que alterna a solução das equações de quantidade de movimento com uma equação de correção de pressão derivada da continuidade. O procedimento consiste em estimar a pressão p nos nós, realizar a predição da velocidade \mathbf{U} , montar uma equação de correção de p , atualizar p e \mathbf{U} com sub-relaxação e repetir até reduzir o desequilíbrio de massa e momento (Patankar, 1980).

A sub-relaxação é um artifício numérico no qual a variável não é atualizada integralmente com o novo valor calculado, mas por uma fração dele combinada com o valor anterior. Esse amortecimento das correções estabiliza o processo iterativo do SIMPLE, reduz oscilações entre iterações e diminui o risco de divergência, ao custo de mais passos até a convergência. O fator de sub-relaxação assume valores entre zero e um: valores menores aumentam a robustez numérica, enquanto valores mais altos aceleram a convergência quando a malha e as condições do problema permitem (Patankar, 1980).

O *pimpleFoam* trata escoamentos incompressíveis transientes combinando a predição de momento com múltiplas correções de pressão no interior de cada passo temporal Δt . A cada avanço temporal, o solver reduz o erro de continuidade por correções sucessivas de p e \mathbf{U} , permitindo maior consistência temporal do acoplamento e controle do número de Courant (Patankar, 1980).

O *rhoSimpleFoam* trata escoamentos compressíveis em regime estacionário, estendendo o SIMPLE para o caso de densidade variável. Além das equações de quantidade de movimento e continuidade, resolve-se a equação de energia para obter a temperatura, e utiliza-se uma equação de estado (por exemplo, gás ideal) para ligar p , ρ e T ; assim, a pressão corrigida ajusta também os fluxos quando a densidade depende de p e T . Esse acoplamento permite capturar efeitos de compressibilidade em problemas sem dependência temporal explícita (Patankar, 1980).

O *rhoPimpleFoam* resolve escoamentos compressíveis transientes. O solver avança no tempo com passos Δt e aplica correções sucessivas de pressão e velocidade dentro de cada passo, enquanto integra a equação de energia para atualizar T .

A equação de estado fecha o sistema a cada Δt , de modo que variações de pressão e temperatura se refletem imediatamente na densidade e nos fluxos. Esse procedimento é adequado quando a evolução temporal e os efeitos de compressibilidade são relevantes, como em transientes térmicos ou variações rápidas de escoamento (Patankar, 1980).

A modelagem da turbulência, essencial em escoamentos de engenharia, está diretamente integrada aos solvers do OpenFOAM por meio de bibliotecas específicas. O software suporta diversas estratégias, desde simulações diretas até abordagens mais econômicas como RANS e LES.

As bibliotecas de turbulência incluem modelos amplamente utilizados, como $k-\varepsilon$, $k-\omega$, Spalart-Allmaras e variantes híbridas como DES (Detached Eddy Simulation), permitindo flexibilidade na escolha do grau de fidelidade e custo computacional (Chen et al., 2014).

Na estrutura do OpenFOAM, a definição do modelo de turbulência é separada da formulação do solver e configurada por meio de arquivos de entrada. Isso permite aplicar diferentes modelos com alterações mínimas na estrutura do caso, mantendo o mesmo solver base. Essa modularidade é um dos principais diferenciais do OpenFOAM, facilitando a experimentação com distintas abordagens sem necessidade de modificar o código-fonte (Chen et al., 2014).

Além dos modelos empíricos, o OpenFOAM também suporta simulações diretas da turbulência, embora com elevado custo computacional. No estudo de Chen et al. (2014), foram aplicadas simulações DNS e LES ao clássico problema da cavidade com tampa móvel, demonstrando a capacidade do software em capturar escoamentos com múltiplas escalas.

Dessa forma, o OpenFOAM se posiciona não apenas como uma ferramenta de simulação, mas também como um ambiente flexível de pesquisa em modelagem turbulenta, sendo amplamente adotado em estudos que buscam equilibrar custo computacional e precisão física.

2.6 AUTOMAÇÃO DE PROCESSOS CFD

Simulações de alta fidelidade tornaram-se essenciais em projetos de engenharia por reduzirem a necessidade de protótipos físicos, acelerando o tempo de desenvolvimento e otimizando o desempenho desde as fases iniciais. No entanto, configurar e executar simulações CFD de forma precisa continua sendo uma tarefa extremamente complexa, que envolve alto investimento de tempo, recursos computacionais e conhecimento técnico especializado (Ligrani et al., 2020).

Historicamente, o uso de ferramentas CFD esteve restrito a uma parcela limitada de profissionais altamente qualificados, dado o alto nível de especialização exigido para gerar malhas de qualidade, ajustar modelos físicos e interpretar correta-

Quadro 3 – Comparação entre os resultados obtidos com DNS, LES e RANS para o escoamento na cavidade com tampa móvel

Modelos de Turbulência	Vantagens	Desvantagens
DNS	Alta precisão. Não necessita correlações empíricas. Descreve todos detalhes do escoamento.	Elevado custo computacional. Difícil definição de condições iniciais e de contorno.
LES	Captura dinâmica dos principais redemoinhos. Menos custoso que DNS. Mais preciso que RANS.	Custo computacional ainda alto. Complexidade em geometrias elaboradas.
RANS	Bom para problemas de engenharia. Custo computacional reduzido.	Não captura detalhes do escoamento. Alta dependência empírica.
DES	Útil em engenharia. Captura efeitos instacionários em escoamentos com separação. Mais geral que RANS.	Não captura detalhes próximos à parede.

Fonte – Chen et al. (2014).

mente os resultados. Como destaca a pesquisa de Freels et al. (2014) com engenheiros mecânicos, mais da metade relatou não se sentir apta a utilizar ferramentas CFD, evidenciando o alto grau de dificuldade associado à sua adoção .

Além disso, o custo envolvido em projetos CFD é substancial: licenças de softwares comerciais podem ultrapassar dezenas de milhares de dólares por ano, enquanto simulações complexas exigem estações de trabalho robustas ou clusters de alto desempenho (Siemens Digital Industries Software, 2023). Como apontado por especialistas da área, o que se contrata, de fato, em um projeto CFD não é apenas o software, mas o tempo, o conhecimento e a experiência do analista responsável Menter (2021).

A automação surge, nesse contexto, como uma resposta estratégica para ampliar a eficiência e a escalabilidade das simulações. Um estudo de caso demonstrou que a integração entre automação e computação de alto desempenho (HPC) permitiu um aumento de 600% no número de iterações de projeto avaliadas durante o mesmo período de desenvolvimento (Smith et al., 2019).

A automação pode ser aplicada em diversas etapas do processo: desde o pré-processamento (geração de malha, definição de condições de contorno), passando pela execução em série de casos paramétricos, até o pós-processamento (extração e análise de resultados). Ao reduzir o esforço manual, esses fluxos automatizados liberam o engenheiro para atuar de forma mais analítica e estratégica, focando na

interpretação dos resultados e não em tarefas repetitivas (Wendt; Casey, 2021).

Contudo, automatizar processos CFD ainda é um desafio considerável. Algumas etapas, como a geração de malhas, continuam exigindo julgamento humano para garantir qualidade e estabilidade numérica, especialmente em geometrias complexas. Embora avanços tenham sido feitos em algoritmos de malha automática, a completa automação desse processo ainda não é plenamente confiável para todos os casos (Frei, 2020). A criação de um pipeline automatizado demanda também investimento inicial em programação, testes e validações, o que só se justifica se o mesmo fluxo for reaproveitado em múltiplas simulações ou estudos de projeto (Eckhardt et al., 2020).

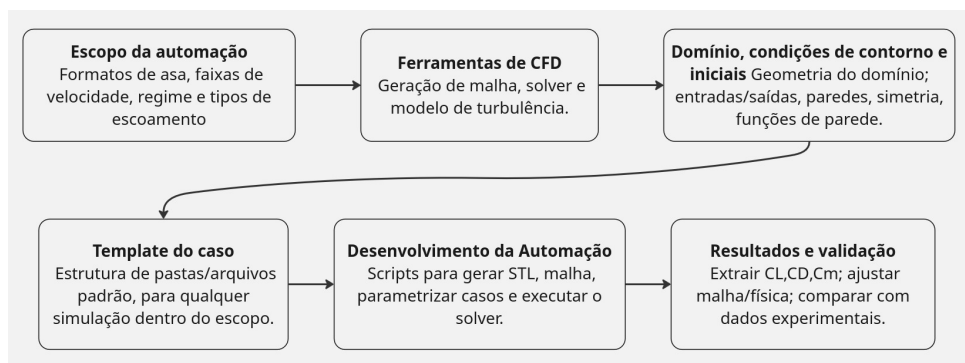
Outro aspecto crítico diz respeito à verificação e validação dos resultados obtidos por meio de simulações automatizadas. A automação pode eliminar erros operacionais humanos, como configurações inconsistentes, mas não assegura por si só a correção física das soluções obtidas. É essencial implementar critérios de verificação — como análise de convergência, independência de malha e conservação de variáveis — e validação contra dados experimentais ou soluções analíticas sempre que possível (Roache, 1998).

Dessa forma, a automação em CFD deve ser encarada não como um substituto da análise técnica, mas como uma ferramenta complementar que, quando aplicada com rigor metodológico, tem o potencial de ampliar a produtividade, reduzir erros e democratizar o uso da simulação numérica no desenvolvimento de projetos de engenharia. A seguir, serão discutidas as principais motivações e possibilidades de automação em CFD, bem como a aplicação prática desses conceitos no ambiente OpenFOAM, com suporte à linguagem Python.

3 METODOLOGIA

Nesse capítulo, são apresentados os procedimentos adotados para a modelagem da asa, geração da malha computacional, configuração das simulações CFD no OpenFOAM, automação do fluxo de trabalho com Python e validação dos resultados obtidos. A metodologia foi estruturada em etapas sequenciais, conforme ilustrado na Figura 12.

Figura 12 – Fluxo metodológico adotado no estudo



Fonte – Autor (2025).

3.1 MODELAGEM DA ASA

Para a modelagem da asa, optou-se pelo emprego de aerofólios da família NACA, pela baixa complexidade geométrica, formulação analítica explícita e ampla confiabilidade reportada na literatura. A parametrização direta de espessura relativa, camber e posição da máxima espessura facilita a geração automática de geometrias, o que possibilita a varredura de configurações de forma sistemática e garante reprodutibilidade no passo a passo do CFD.

Assim, definiram-se parâmetros geométricos e de voo como entradas, com o objetivo de implementar um algoritmo que gere automaticamente o arquivo STL de uma asa 3D a partir desses valores. Foram considerados: (i) o código NACA do aerofólio, (ii) a envergadura (span), (iii) a corda na raiz, (iv) a corda na ponta, (v) o deslocamento longitudinal, (vi) o ângulo de ataque e (vii) a velocidade de voo. Esses parâmetros descrevem o plano médio e a variação de corda, permitindo a construção paramétrica da geometria e sua exportação em STL.

A forma do aerofólio NACA de 4 dígitos utilizada no algoritmo é descrita pela Equação 12. Para um código NACA $abcd$ (com dígitos a, b, c, d), definem-se:

$$m = \frac{a}{100}, \quad p = \frac{b}{10}, \quad t = \frac{cd}{100}, \quad x \in [0, 1]. \quad (12)$$

onde m é a curvatura máxima da linha média, dada como fração da corda, p é a posição do camber máximo ao longo da corda, medida desde o bordo de ataque e t é a espessura máxima do aerofólio, também como fração da corda. A distribuição de semiespessura (metade da espessura) ao longo da corda normalizada é mostrada na Equação 13.

$$y_t(x) = 5t \left(0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4 \right). \quad (13)$$

A seguir, são apresentados o cálculo da linha média (camber, Equação 14) e sua derivada (Equação 15), do ângulo da tangente ao camber (Equação 16) e das coordenadas das superfícies superior (Equação 17) e inferior (Equação 18), todos utilizados na geração da geometria de forma automatizada:

$$y_c(x) = \begin{cases} \frac{m}{p^2} (2px - x^2), & 0 \leq x < p, \\ \frac{m}{(1-p)^2} ((1-2p) + 2px - x^2), & p \leq x \leq 1. \end{cases} \quad (14)$$

$$\frac{dy_c}{dx} = \begin{cases} \frac{2m}{p^2} (p - x), & 0 \leq x < p, \\ \frac{2m}{(1-p)^2} (p - x), & p \leq x \leq 1. \end{cases} \quad (15)$$

$$\theta(x) = \arctan\left(\frac{dy_c}{dx}\right). \quad (16)$$

$$x_u(x) = x - y_t(x) \sin\theta(x), \quad y_u(x) = y_c(x) + y_t(x) \cos\theta(x). \quad (17)$$

$$x_\ell(x) = x + y_t(x) \sin\theta(x), \quad y_\ell(x) = y_c(x) - y_t(x) \cos\theta(x). \quad (18)$$

Com base nessas equações, o contorno bidimensional do aerofólio NACA é fechado pela combinação das superfícies superior e inferior, permitindo a extrusão ao longo da envergadura com variação linear da corda entre raiz e ponta e eventual deslocamento longitudinal. Esse procedimento permite a geração da geometria tridimensional da asa em formato STL, incorporando ângulo de ataque e demais parâmetros definidos, o que possibilita a reprodução automática de diferentes perfis NACA, assegurando reprodutibilidade, flexibilidade na variação de casos de estudo e eficiência na preparação das geometrias para simulações CFD.

Diversas bibliotecas em Python foram avaliadas para a implementação do algoritmo de geração automática da geometria da asa. Optou-se pela utilização do *NumPy-STL*, devido à sua simplicidade, eficiência e integração direta com a linguagem Python, além de ser amplamente adotada na comunidade científica para manipulação de arquivos STL.

3.2 GERAÇÃO DE MALHA

O *blockMesh* define o envelope do escoamento. O domínio foi dimensionado em função do comprimento característico L , calculado a partir das cordas de raiz c_r e de ponta c_t , como mostra a equação 19.

$$\lambda = \frac{c_t}{c_r} \quad L = \frac{2}{3} c_r \frac{1 + \lambda + \lambda^2}{1 + \lambda} \quad (19)$$

Adotaram-se dimensões explícitas $L_x = 10 L$ e $L_y = L_z = 8 L$, conforme a parametrização dos *vertices*. A resolução foi fixada por $(N_x, N_y, N_z) = (30, 24, 24)$, implicando passos $\Delta_x = L_x/N_x$, $\Delta_y = L_y/N_y$ e $\Delta_z = L_z/N_z$. Empregou-se *multi-grading* em *simpleGrading* para concentrar células nas regiões de interesse: no eixo x , partições (25%, 25%, 50%) com razões (0,2, 0,4, 2); no eixo y , (35%, 30%, 35%) com (0,25, 0,5, 4); no eixo z , (70%, 30%) com (0,8, 0,2). Essa configuração substitui afastamentos genéricos por múltiplos de L nos *vertices* e assegura transição suave entre a vizinhança da asa e as regiões mais afastadas do domínio.

3.2.1 Condições de Contorno

As condições de contorno do domínio foram definidas de modo a representar fielmente o escoamento ao redor da asa e garantir a correta resolução das variáveis físicas nas interfaces entre o fluido e as superfícies sólidas. O domínio foi dividido em três regiões principais: *freestream*, que compreende as superfícies de entrada do escoamento, bem como as superfícies laterais, superior e inferior; *outFlow*, correspondente à superfície de saída do fluido situada atrás da asa; e *wall*, que representa a própria superfície da asa.

Para o campo de velocidade, na região *freestream* foi aplicada a condição de contorno *fixedValue*, fixando o valor da velocidade de entrada parametrizado conforme o caso estudado. Essa abordagem garante que o escoamento incidente seja mantido constante ao longo da simulação, estabelecendo o número de Reynolds e a intensidade do fluxo principal. Na superfície *outFlow*, foi utilizada a condição *zeroGradient*, que permite que o gradiente normal da velocidade seja nulo, ou seja, não impõe restrições ao valor da velocidade na saída, permitindo que o fluido escoe livremente sem reflexões artificiais ou restrições numéricas. Na superfície *wall*, correspondente à asa,

foi adotada a condição de não deslizamento (*noSlip*), impondo velocidade nula na interface fluido-sólido, o que é fundamental para a correta formação da camada limite e para a predição dos esforços de cisalhamento.

Para o campo de pressão, o valor interno foi inicializado como *uniform 0*. Nas superfícies *freestream* e *wing*, foi aplicada a condição *zeroGradient*, permitindo que o gradiente normal da pressão seja nulo e que o campo se ajuste naturalmente às variações induzidas pelo escoamento. Na superfície *outFlow*, foi utilizada a condição *fixedValue*, fixando o valor da pressão em zero, o que estabelece uma referência para o campo de pressão e facilita o fechamento do balanço de massa no domínio.

Para as variáveis de turbulência, k e ω , o valor calculado na simulação foi fixado na região *freestream* por meio da condição *fixedValue*, assegurando que o nível de turbulência incidente seja consistente com o regime de escoamento desejado. Na superfície *outFlow*, foi utilizada a condição *zeroGradient*, permitindo a saída livre das variáveis turbulentas sem imposição de gradientes artificiais. Na superfície da asa (*wing*), foram aplicadas as funções de parede (*wallFunctions*), que ajustam o comportamento das variáveis turbulentas de acordo com o modelo de turbulência adotado e com a resolução da malha junto à parede, garantindo a correta predição dos efeitos viscosos e turbulentos na camada limite.

No caso da viscosidade turbulenta (ν_t), o valor foi definido como *uniform 0* em todo o domínio. Embora o campo ν_t seja utilizado principalmente em simulações com modelos de turbulência RANS, como o $k-\omega$ SST, no contexto do *simpleFoam* essa definição inicial é adequada, pois o valor de ν_t é recalculado iterativamente pelo solver a partir das equações de transporte de k e ω . Dessa forma, a condição inicial uniforme não compromete a evolução do campo turbulento, sendo suficiente para garantir a estabilidade numérica e a correta ativação dos modelos de parede.

Essas escolhas de condições de contorno asseguram que o escoamento seja corretamente representado nas interfaces do domínio, permitindo a formação da camada limite na asa, a entrada e saída livre do fluido e a adequada resolução das variáveis turbulentas, em conformidade com as melhores práticas para simulações CFD em perfis aerodinâmicos.

3.2.2 Integração da malha do domínio com a geometria da asa

A ferramenta *snappyHexMesh* foi empregada em três fases complementares. Na fase de *castellation*, a malha de fundo é refinada nas vizinhanças da superfície e removem-se os volumes de controle que incidem sobre o interior do sólido, garantindo que o domínio fluido represente apenas a região externa ao corpo. Essa etapa define a resolução volumétrica inicial ao redor da geometria e prepara o campo para o alinhamento geométrico.

Na fase de *snap*, os vértices e faces próximos à superfície STL são projetados em direção à geometria alvo, reduzindo o erro de representação e melhorando a conformidade da malha com borda de ataque, bordo de fuga e curvaturas relevantes. Ajusta-se a posição local dos nós para elevar a ortogonalidade e evitar colapsos, preservando a qualidade métrica sem alterar o conteúdo topológico obtido na *castellation*.

Na fase de *addLayers*, são acrescentadas camadas prismáticas ao longo das paredes, com espessura e razão de expansão definidas para capturar gradientes normais à superfície e atender metas de y^+ . Essas camadas ampliam a resolução na subcamada viscosa e na região logarítmica, mantendo transição suave para a malha hexaédrica do volume e preservando critérios de qualidade (não-ortogonalidade, *skewness* e razão de aspecto).

Para viabilizar o cálculo automático em cada fase, estabelecem-se previamente limites, níveis de refinamento, tolerâncias geométricas e critérios de transição entre tamanhos de células. Esses parâmetros guiam a geração ao indicar onde a malha deve concentrar elementos, como suavizar variações entre regiões adjacentes e quais condições de qualidade precisam ser respeitadas. A definição consistente desses valores assegura que a geometria seja capturada com fidelidade e que o custo computacional permaneça dentro do orçamento.

Entre os parâmetros específicos, destaca-se o *firstLayerThickness*, que define a espessura do primeiro volume de controle adjacente à asa. Esse valor é calculado a partir de uma meta de y^+ desejável e inserido pela automação, com base na relação

$$\Delta y_1 \approx \frac{y_*^+ \nu}{u_\tau}, \quad u_\tau \approx U_\infty \sqrt{\frac{C_f}{2}}, \quad C_f \approx 0,026 Re^{-1/7},$$

em que ν é a viscosidade cinemática, U_∞ a velocidade de referência e $Re = U_\infty c / \nu$ um número de Reynolds característico. A escolha adequada de Δy_1 posiciona o primeiro ponto dentro da subcamada viscosa ou da região logarítmica alvo, o que é crucial para a acurácia do modelo $k-\omega$ SST e para a predição de esforços de superfície.

Todo o fluxo foi automatizado por *scripts* sequenciais, descritos no Apêndice A. O pipeline executa *blockMesh*, *surfaceFeatures*, *snappyHexMesh* em passes controlados e *checkMesh* entre fases. A automação padroniza nomes de arquivos, registra *logs* e preserva a reprodutibilidade para diferentes números de Mach e ângulos de ataque.

3.3 ESTUDO PRELIMINAR

A seleção de solvers restringiu-se a *simpleFoam* e *rhoSimpleFoam*, em consonância com a natureza estacionária dos casos estudados e com a automação construída para campos médios em regime permanente. Solvers transientes (*pimpleFoam* e *rhoPimpleFoam*) foram deliberadamente excluídos nesta etapa e reservados como

possibilidade para trabalhos futuros, por exigirem integração temporal e maior custo por iteração.

Ensaio comparativos entre *simpleFoam* e *rhoSimpleFoam* foram conduzidos ao longo de números de Mach crescentes até o limite do escopo. Adotou-se *simpleFoam* para $M \leq 0,3$, quando as variações de densidade são desprezíveis, e *rhoSimpleFoam* para $0,3 < M \leq 0,7$, quando os efeitos de compressibilidade passam a influenciar os coeficientes aerodinâmicos sem exigir tratamento transiente. A decisão baseou-se na convergência dos resíduos e na estabilização dos coeficientes de força, com prioridade para o menor custo computacional compatível com a precisão requerida.

O modelo de turbulência adotado foi o RANS $k-\omega$ SST, pela boa resposta em gradiente adverso de pressão, capacidade de resolver a região próxima à parede com $y^+ \approx 1$ e uso consolidado em aerodinâmica de perfis e asas. Modelos alternativos, como Spalart–Allmaras (menor custo, porém menos confiável no início de separação), famílias $k-\varepsilon$ clássicas (mais sensíveis sob gradiente adverso) e RSM/DES (maior custo e complexidade de malha e calibração), foram preteridos para manter o foco no objetivo do estudo e no orçamento computacional disponível.

O ângulo de ataque é incorporado por rotação consistente das seções em torno de um eixo de referência, mantendo a orientação dos eixos do domínio. As pontas são seladas por superfícies na corda e na ponta da asa para garantir um sólido fechado e o resultado é discretizado em malha triangular e exportado em formato STL. Todo o processo é automatizado em *Python*, como indica o Apêndice B, mas é descrito aqui em termos puramente geométricos, independentes de implementação.

Ao longo das fases, a consistência global é garantida por *meshQualityControls* (limites para *non-orthogonality*, *skewness*, volumes mínimos e concavidade). Após a geração, procede-se à verificação de métricas de qualidade e, se necessário, ajustam-se níveis de refinamento, gradings e parâmetros de camada até atender simultaneamente às metas de resolução geométrica (bordo de ataque/fuga), de camada limite (y^+) e de custo computacional.

As condições de contorno foram escolhidas com base em seu significado físico. Na entrada, impõe-se velocidade uniforme, o que fixa a intensidade do escoamento livre e, junto à escala geométrica, determina o número de Reynolds do caso. Na saída, define-se uma pressão de referência, permitindo que o campo de pressão ajuste-se naturalmente sem impor gradientes artificiais. Nas superfícies laterais e em teto/fundo afastados, adota-se condição de *slip* para representar escoamento distante sem atrito, reduzindo efeitos de parede no campo principal. Na asa, aplica-se *noSlip* para impor velocidade nula na parede (condição de não deslizamento), viabilizando a formação da camada limite e a correta predição de esforços cisalhantes.

Os coeficientes aerodinâmicos são obtidos pelo *function object* de forças, cal-

culando C_L e C_D a partir das integrais de pressão e cisalhamento na asa. Utiliza-se S como área de referência projetada e a densidade ρ e velocidade U_∞ consistentes com o ramo escolhido; no ramo compressível (quando aplicável), ativam-se *thermophysicalProperties* e campo de temperatura T com modelo de gás ideal e viscosidade dependente de T . Essa configuração garante reprodutibilidade e permite escalar o estudo para outras velocidades apenas alternando automaticamente o solver conforme o número de Mach.

Por fim, os arquivos de pós processamento gerados pelo OpenFOAM são utilizados para geração de gráficos. São extraídos os coeficientes aerodinâmicos, métricas de qualidade de malha e estatísticas de convergência, consolidando-os em figuras de forma a facilitar a avaliação dos resultados.

4 RESULTADOS

Este capítulo apresenta os resultados numéricos obtidos com a automação desenvolvida. Foram geradas simulações para diferentes geometrias de asa, ângulos de ataque e números de Mach do escoamento. A configuração de cada caso seguiu o procedimento metodológico previamente descrito.

Os coeficientes aerodinâmicos de interesse (C_L , C_D , C_M), além de métricas de qualidade de malha foram extraídos diretamente dos arquivos de pós-processamento e sintetizados em tabelas e figuras. As séries temporais e estatísticas de convergência foram igualmente coletadas e consolidadas. Os resultados são confrontados com dados experimentais da literatura, de modo a avaliar consistência, tendências e desvios relativos.

Para todas as configurações simuladas nesse contexto, foram adotados valores de densidade e viscosidade dinâmica do ar a nível do mar e pressão atmosférica padrão, conforme descrito por Çengel e Boles (2013). A densidade do ar foi definida como $\rho = 1.225 \text{ kg/m}^3$, enquanto a viscosidade dinâmica foi fixada em $\mu = 1.8 \times 10^{-5} \text{ Pa.s}$. A partir desses valores, obteve-se uma viscosidade cinemática de $\nu = 1.471 \times 10^{-5} \text{ m}^2/\text{s}$.

Finalmente, o eixo x foi orientado no sentido do escoamento, com sentido do bordo de ataque ao bordo de fuga, o eixo y na direção vertical para cima e o eixo z na direção transversal (da envergadura), no sentido da raiz à ponta, formando um sistema de coordenadas retangular ortogonal destro. A seguir, são apresentados os resultados obtidos para o modelo de asa com seção constante.

4.1 MODELO COM SEÇÃO CONSTANTE

Ostowari e Naik (1985) realizaram um estudo com o objetivo de caracterizar o desempenho de asas em condições de estol, levando em consideração variáveis como a razão de aspecto, a espessura da lâmina e o número de Reynolds. O estudo foi realizado por meio de testes em túnel de vento com lâminas de corda constante e seções de aerofólio da série NACA 44XX (iniciadas por "44").

Foram testadas lâminas com diferentes razões de aspecto (6, 9, 12 e infinito), em uma faixa de ângulos de ataque de -10° a 110° e números de Reynolds variando de 0.25×10^6 a 1.0×10^6 . A espessura dos aerofólios, medida em porcentagem em relação à corda dele, que corresponde aos dois últimos dígitos da série NACA de 4 dígitos, foi variada entre 18, 15, 12 e 09 (Ostowari; Naik, 1985).

4.1.1 Configuração das Simulações

Foram realizadas simulações para quatro configurações distintas de aerofólios, variando o tipo de aerofólio e o ângulo de ataque. As 4 configurações testadas envolveram os aerofólios NACA 4412 e NACA 4415, cada um com ângulos de ataque de 10° e 15° .

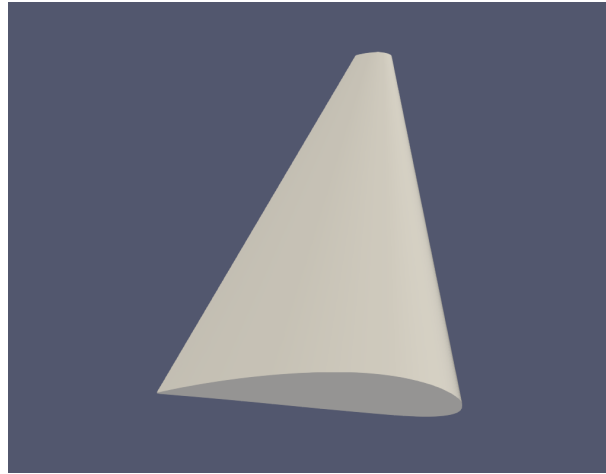
Para todas as configurações, o aerofólio foi modelado com uma corda constante de 0.2 m, tanto na raiz quanto na ponta da asa, caracterizando uma geometria de seção constante. A envergadura da asa foi definida como 1.2 m, resultando em uma razão de aspecto $AR = 6$, e o enflechamento em x foi definido como 0.

Além disso, o escoamento foi simulado em condições de regime subcrítico, com velocidade de entrada do fluido definida como $U_\infty = 36.77$ m/s, de modo a alcançar um número de Reynolds de $Re = 0.5 \times 10^6$. Essas configurações de entrada replicam as usadas nos experimentos de Ostowari e Naik (1985), garantindo fidelidade na validação dos resultados.

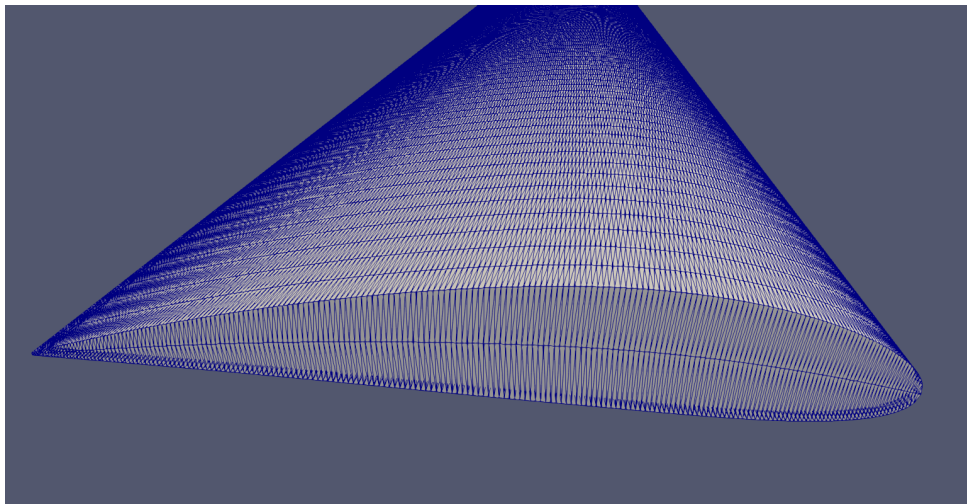
As figuras abaixo ilustram o arquivo STL da asa gerado pela automação para a configuração do aerofólio NACA 4415 com ângulo de ataque de 10° . A Figura 13a apresenta a superfície lisa do objeto, destacando a geometria tridimensional. A Figura 13b ilustra a malha na superfície de forma mais aproximada. Finalmente, a Figura 13c mostra uma vista lateral da asa, evidenciando a curvatura do aerofólio e a inclinação em relação ao fluxo de entrada.

Figura 13 – Visualizações do STL da asa de aerofólio NACA 4415 a 10°

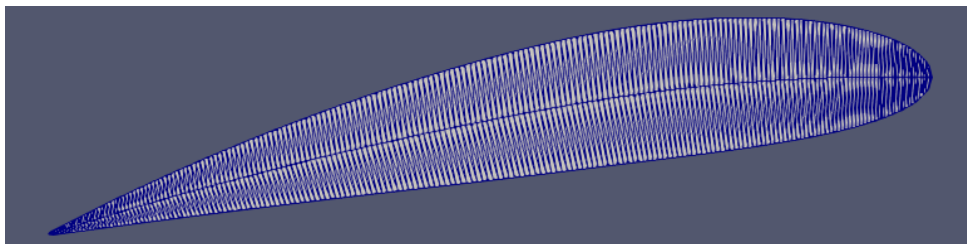
(a)



(b)



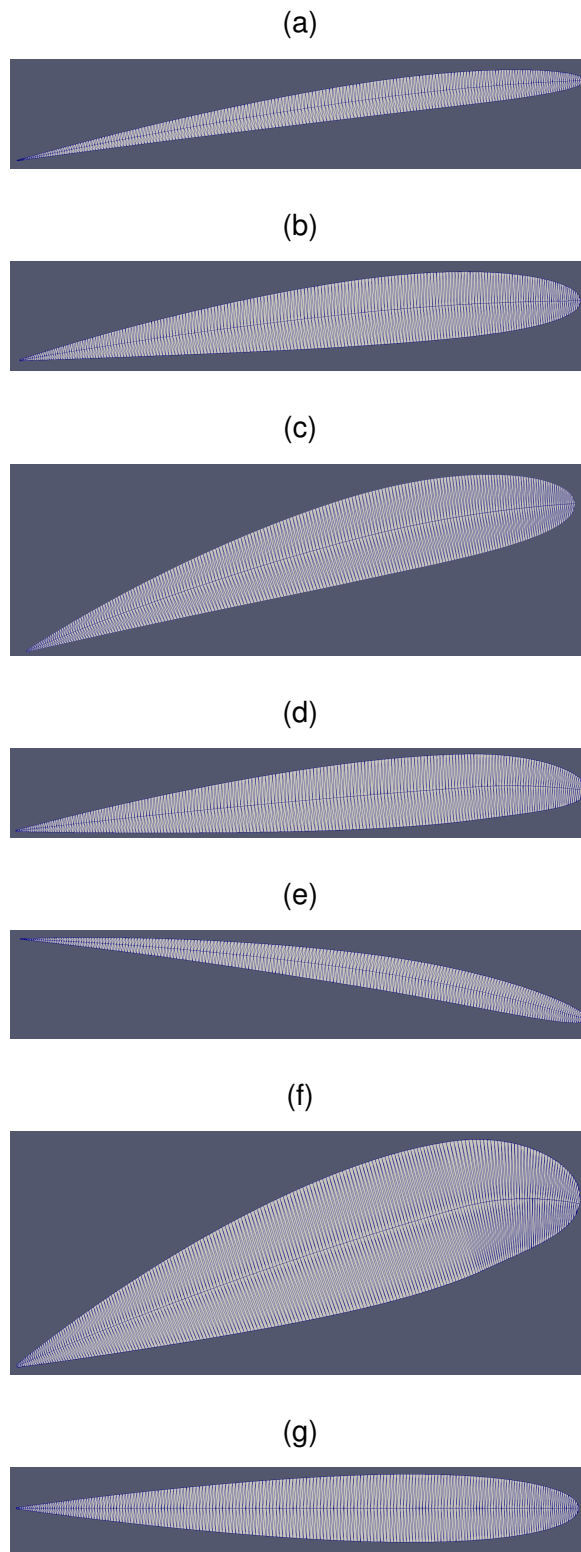
(c)



Fonte – Autor (2025).

Para efeito de comparação, as Figuras 14a, 14b e 14c mostram uma imagem lateral da malha gerada no arquivo STL de diferentes modelos NACA com ângulos de ataque variados, para visualização da geração do aerofólio e sua inclinação.

Figura 14 – Visualizações do STL de diferentes modelos de aerofólio **(a)** NACA 2406 a 8° **(b)** NACA 2412 a 6° **(c)** NACA 4412 a 15° **(d)** NACA 2212 a 4° **(e)** NACA 3406 a -8° **(f)** NACA 5224 a 16° **(g)** NACA 0012 a 0°



Fonte – Autor (2025).

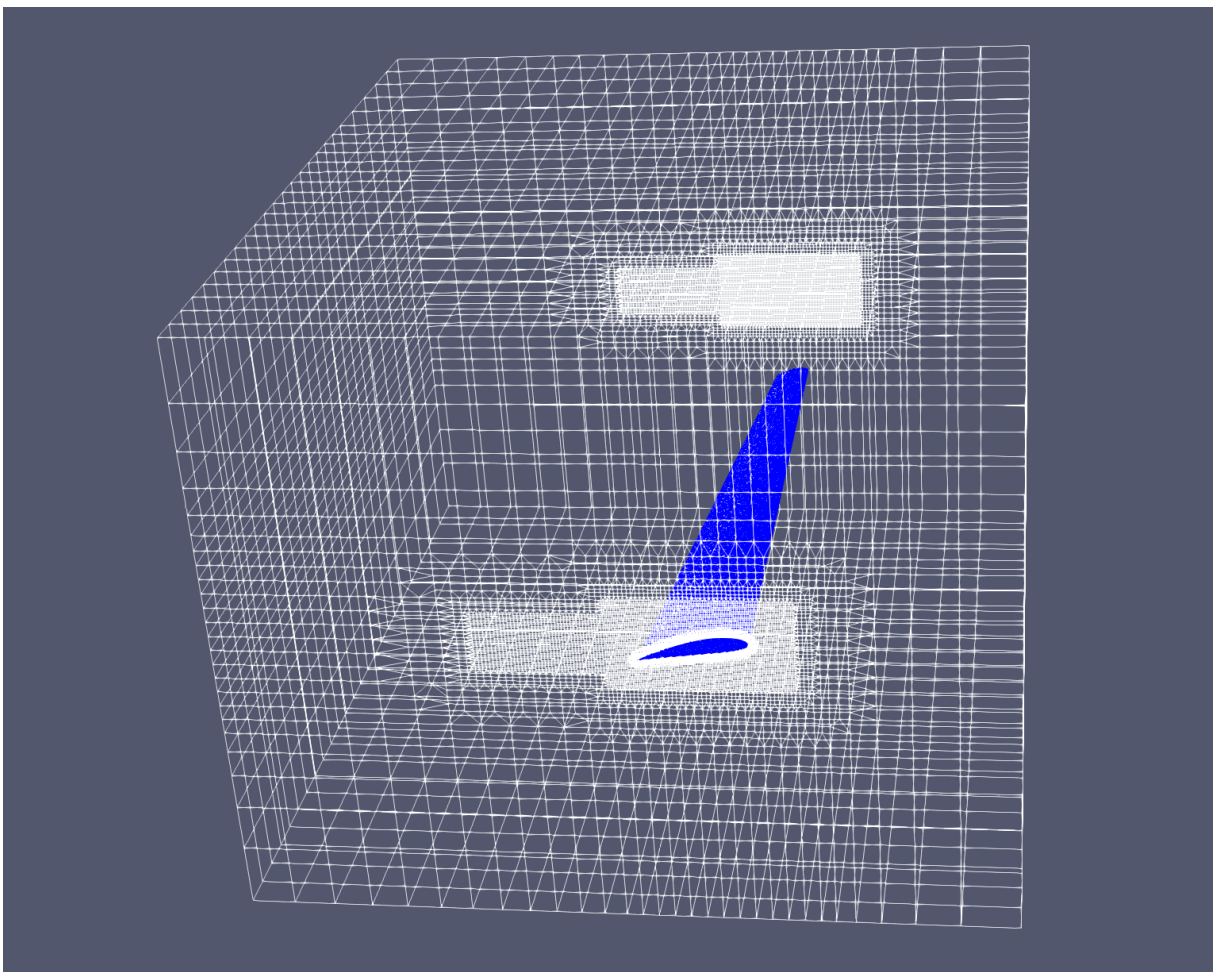
4.1.2 Qualidade das Malhas e Parâmetros de Simulação

Para todas as configurações simuladas, foram geradas malhas contendo aproximadamente 6.500.000 células. A distribuição das células na malha foi composta por cerca de 5.900.000 hexaedros, 230.000 prismas e 370.000 poliedros.

As malhas geradas foram submetidas ao teste de qualidade *checkMesh*, um procedimento padrão do *OpenFOAM*, que verifica a integridade e a conformidade da malha com os critérios necessários para simulações numéricas precisas e estáveis. Todas as malhas passaram com sucesso nesse teste, indicando ausência de erros críticos, como aspectos geométricos degenerados ou problemas de conectividade entre as células.

As figuras abaixo ilustram a malha gerada para a configuração do aerofólio NACA 4415 com ângulo de ataque de 10° . A Figura 15 apresenta uma visão geral da malha no domínio de escoamento, destacando a distribuição das células ao redor do aerofólio.

Figura 15 – Visão geral da malha do aerofólio NACA 4415 a 10°

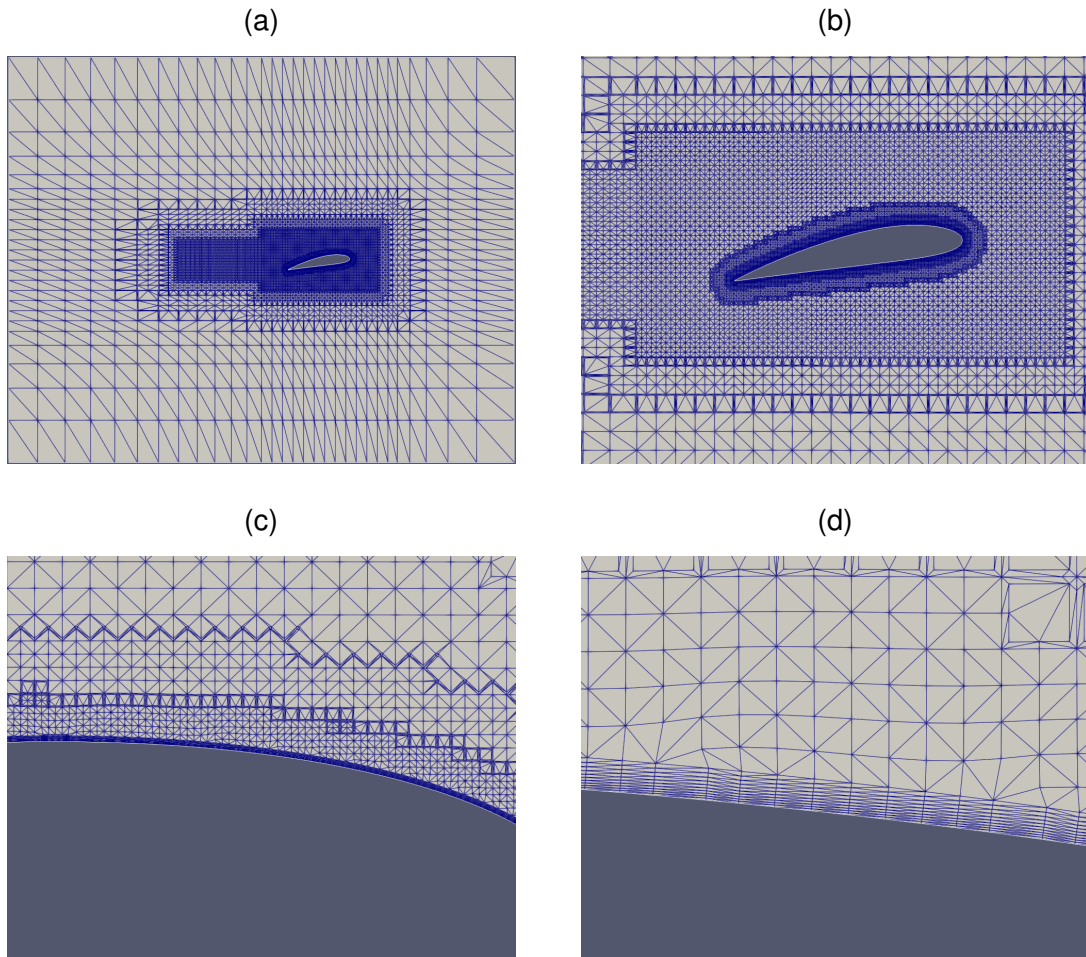


Fonte – Autor (2025).

Já a Figura 16 mostra um corte transversal da malha com ampliações diferen-

tes, evidenciando a densidade de células na região próxima à superfície do aerofólio, onde a camada limite se desenvolve, permitindo a visualização das camadas prismáticas adicionadas para melhorar a resolução nessa região crítica.

Figura 16 – Corte transversal da malha do aerofólio NACA 4415 a 10°



Fonte – Autor (2025).

A Tabela 1 apresenta os valores de y^+ para as configurações simuladas, incluindo os valores mínimo, médio e máximo. Esses valores foram calculados para todas as células na região próxima às superfícies dos aerofólios, com o objetivo de avaliar a resolução da camada limite e garantir que os fenômenos físicos na interface fluido-sólido fossem modelados com precisão.

Para as configurações simuladas, os valores médios de y^+ variam entre 18.67 e 21.40, o que está dentro de uma faixa aceitável para o modelo de turbulência k - ϵ utilizado, que geralmente requer valores de y^+ médios na ordem de 30 ou inferiores para uma representação precisa da camada limite. No entanto, os valores mínimos e máximos apresentam uma discrepância significativa em relação aos valores médios. Por exemplo, para o caso do aerofólio NACA 4412 com ângulo de ataque de

Tabela 1 – y^+ calculado para os modelos simulados

NACA / α	y^+ min.	y^+ med.	y^+ max.
4412 / 10°	0.67	21.07	114.78
4412 / 15°	0.88	18.67	118.07
4415 / 10°	0.51	21.40	115.41
4415 / 15°	0.72	19.05	112.07

Fonte – Autor (2025).

10°, o y^+ mínimo foi de 0.67, enquanto o y^+ máximo alcançou 114.78, representando uma diferença considerável.

Essa variação entre os valores mínimo e máximo de y^+ pode indicar uma não uniformidade na resolução da camada limite ao longo da superfície do aerofólio. Valores muito baixos de y^+ podem gerar imprecisões na representação da camada limite laminar, enquanto valores muito altos podem comprometer a precisão na região de maior gradiente de velocidade próximo à superfície, especialmente em áreas de separação do escoamento.

Os valores mínimos de y^+ , inferiores a 1 em algumas configurações (por exemplo, 0.51 para o aerofólio NACA 4415 com ângulo de ataque de 10°), são adequados para simulações que utilizam aproximações diretas da camada limite, como o modelo de parede (*wall functions*) do *OpenFOAM*. No entanto, essa adequação depende de uma uniformidade maior dos valores ao longo da superfície. Por outro lado, os valores máximos de y^+ , como os 118.07 obtidos para o aerofólio NACA 4412 com ângulo de ataque de 15°, ultrapassam o limite recomendado para o modelo *k-epsilon*, que geralmente opera melhor com valores não superiores a 100.

Apesar das limitações identificadas, os resultados obtidos fornecem uma base consistente para análise aerodinâmica das configurações simuladas.

4.1.3 Comparação dos Resultados Numéricos com Dados Experimentais

As tabelas 2a, 2b, 2c, 2d comparam os resultados numéricos obtidos para os três coeficientes de interesse citados com os dados experimentais reportados por Ostowari e Naik (1985).

Tabela 2 – Comparação dos resultados numéricos e experimentais para diferentes configurações

(a) Modelo NACA 4412 - 10°

	Numérico	Experimental	Erro (%)
C_d	0.0557	0.06	-7.17
C_l	0.8796	0.81	+8.59
C_m	-0.0788	-0.07	-12.57

Fonte Autor (2025).

(b) Modelo NACA 4412 - 15°

	Numérico	Experimental	Erro %
C_d	0.2118	0.18	+17.67
C_l	0.9360	0.95	-1.47
C_m	-0.1313	-0.11	-19.36

Fonte Autor (2025).

(c) Modelo NACA 4415 - 10°

	Numérico	Experimental	Erro %
C_d	0.0532	0.05	+6.40
C_l	0.8808	0.82	+7.41
C_m	-0.0632	-0.06	+5.33

Fonte Autor (2025).

(d) Modelo NACA 4415 - 15°

	Numérico	Experimental	Erro %
C_d	0.2287	0.20	+14.35
C_l	1.1468	1.20	-4.43
C_m	-0.0814	-0.07	-16.29

Fonte Autor (2025).

Os dados indicam um padrão coerente com escoamentos tridimensionais próximos ao estol. Em $\alpha = 10^\circ$ há superestimação de sustentação, enquanto em $\alpha = 15^\circ$ o C_L se aproxima do experimental e o C_D cresce além do medido. Já o momento de arfagem é subestimado em todos os casos, com discrepâncias maiores em ângulos mais elevados.

Essa tendência é típica de RANS estacionário com hipótese de transição totalmente turbulenta: atraso de separação em ângulos moderados eleva C_L ; já em

ângulos maiores, a separação estabelecida amplia perdas viscosas e induz momento mais picado, elevando C_D e $|C_M|$. A distribuição de y^+ não uniforme observada reforça esse diagnóstico, pois compromete a consistência do tratamento de parede e a captura dos gradientes na camada limite e na esteira.

4.2 MODELO COM SEÇÃO VARIÁVEL

A comparação exclusiva com asas de seção constante seria insuficiente para validar um pipeline tridimensional, pois não captura os efeitos de plano de asa, como afilamento e enflechamento. Assim, inclui-se um modelo com seção variável, coerente com a literatura de asas finitas, tornando viável a avaliação da automação frente a fenômenos de indução e distribuição de sustentação típicos de geometria 3D.

Para esse fim, adota-se como referência experimental o estudo de Malik et al. (2019), que ensaiou em túnel de vento uma meia-asa NACA 4412 afilada, com enflechamento de 30° e número de Reynolds baseado na corda aerodinâmica média de 4×10^5 . Além disso, o trabalho reporta coeficientes C_L , C_D e C_M para a asa lisa e para variações de rugosidade, dessa forma, a configuração sem rugosidade se faz apropriada para confronto direto com os resultados tridimensionais gerados pela automação proposta.

Por fim, vale ressaltar que o estudo emprega diedro de 5° na meia-asa, o que não é parametrizável na versão atual da automação geométrica. Porém, ele será desprezado neste trabalho. Em deriva nula ($\beta = 0$) e sem rolagem, o diedro afeta sobretudo a estabilidade lateral (C_{l_β}) e tem impacto desprezível nos coeficientes longitudinais. O efeito dominante é apenas a projeção da sustentação na vertical: $L_v = L \cos \gamma$; para $\gamma = 5^\circ$, $\cos \gamma \approx 0,9962$, o que implica variação $< 0,4$ em C_L e de mesma ordem em C_D e C_m . Essa diferença fica abaixo das incertezas experimentais e numéricas usuais e não altera a área de referência adotada. Assim, usar $\gamma = 0^\circ$ mantém a simetria do domínio e a parametrização sem impacto relevante nos resultados.

4.2.1 Configuração das Simulações

Foi executada uma simulação a $\alpha = 14^\circ$, ângulo correspondente ao estol com C_L máximo reportado por Malik et al. (2019). Para reproduzir a configuração experimental, adotou-se NACA 4412, envergadura $b = 0,90$ m, corda na raiz $c_r = 0,40$ m, corda na ponta $c_t = 0,10$ m e deslocamento longitudinal $x_{\text{off}} = 0,52$ m medido a partir do bordo de ataque. O escoamento de entrada foi fixado em $U_\infty = 20 \text{ m s}^{-1}$.

A geometria é afilada com razão $\lambda = c_t/c_r = 0,25$, e o comprimento característico foi tomado como a corda aerodinâmica média (MAC). Para asa trapezoidal, a

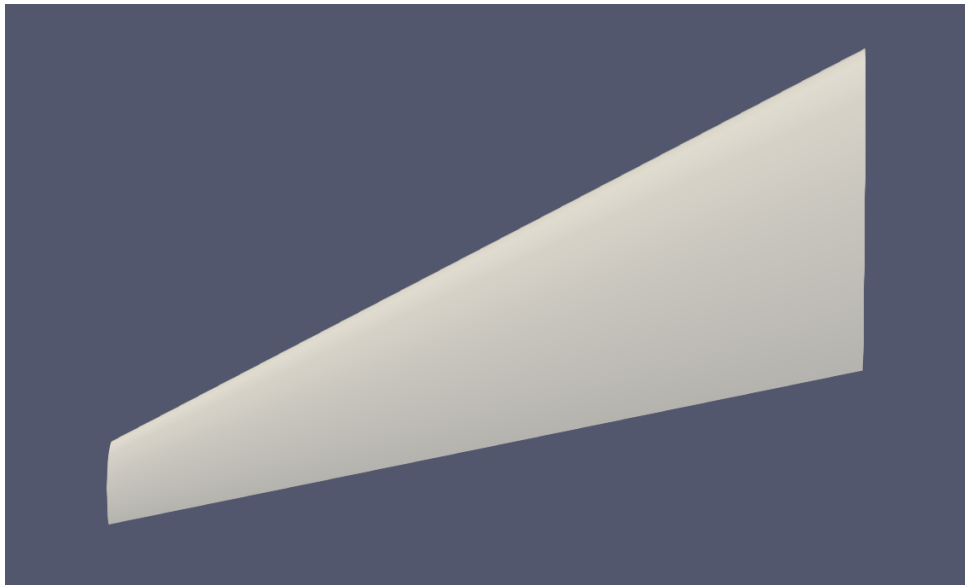
MAC é

$$\bar{c} = \frac{2}{3} c_r \frac{1 + \lambda + \lambda^2}{1 + \lambda}. \quad (20)$$

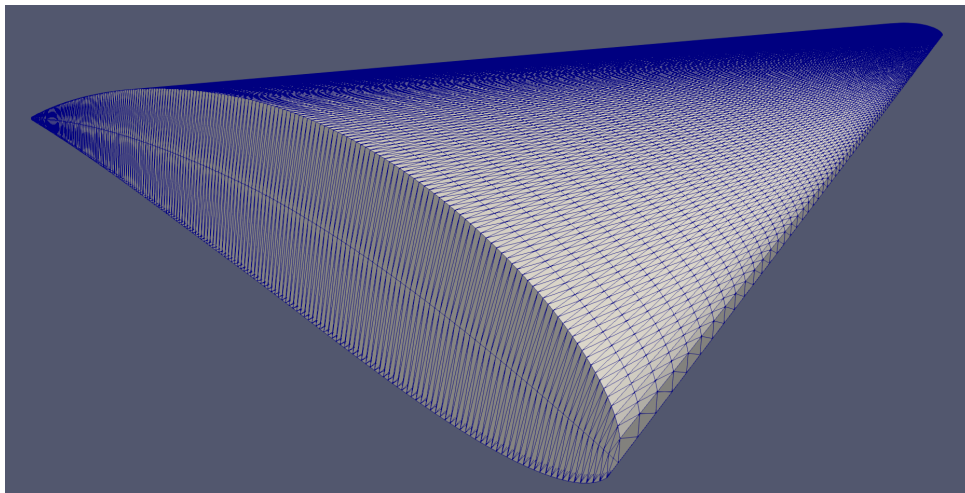
no caso estudado, $c_r = 0,40$ m e $c_t = 0,10$ m, resultando em $\bar{c} = 0,28$ m. A Figura 17 mostra o arquivo STL da asa gerada pela automação.

Figura 17 – Visualização do arquivo STL da asa enflechada

(a) Vista superior da geometria da asa



(b) Vista isométrica da malha da asa.



Fonte – Autor (2025).

Com $\rho = 1,225$ kg m⁻³ e $\nu = 1,471 \times 10^{-5}$ m² s⁻¹, o número de Reynolds é dado

por

$$\text{Re} = \frac{U_{\infty} \bar{c}}{\nu}. \quad (21)$$

para $U_{\infty} = 20 \text{ m s}^{-1}$ e $\bar{c} = 0,28 \text{ m}$, obtém-se $\text{Re} \approx 3,8 \times 10^5$, em alinhamento com a ordem de grandeza do estudo experimental.

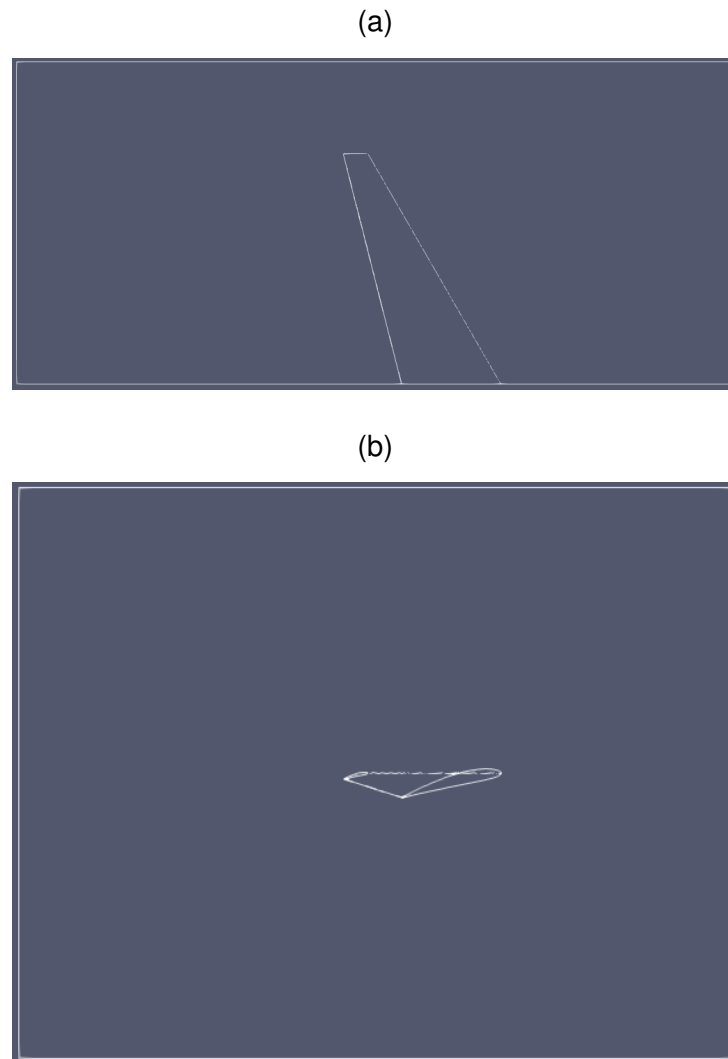
Para os coeficientes aerodinâmicos, adotou-se a mesma convenção de referência do artigo, utilizando $c_{\text{ref}} = \bar{c}$ e a área de referência reportada por Malik et al. (2019). Essa escolha garante consistência na comparação direta dos valores de C_L , C_D e C_M entre os resultados numéricos e a base experimental.

4.2.2 Qualidade das Malhas e Parâmetros de Simulação

A malha gerada para a configuração da asa enflechada continha aproximadamente 5.290.956 células, distribuídas em 4.767.924, 41.083 prismas e 481.949 poliedros. A malha passou com sucesso no teste de qualidade *checkMesh* do *OpenFOAM*, indicando que estava adequada para a simulação numérica.

A Figura 18 apresenta ângulos diferentes da geometria gerada, traçando linhas nas arestas características do domínio e da asa. A Figura 18a apresenta uma vista superior, já a Figura 18b apresenta uma vista lateral.

Figura 18 – Visualização do arquivo STL da asa enflechada



Fonte – Autor (2025).

O y^+ médio calculado para a malha da asa enflechada foi de aproximadamente 36.48, o que está dentro da faixa recomendada para o modelo de turbulência *k-epsilon* utilizado na simulação. No entanto, assim como nas configurações anteriores, os valores mínimo e máximo de y^+ apresentaram uma variação significativa em relação ao valor médio, indicando uma não uniformidade na resolução da camada limite ao longo da superfície da asa.

4.2.3 Comparação dos Resultados Numéricos com Dados Experimentais

A Tabela 3 compara os resultados numéricos obtidos para os coeficientes aerodinâmicos C_L , C_D e C_M com os dados experimentais reportados por Malik et al. (2019) para a asa enflechada a um ângulo de ataque de 14° .

Tabela 3 – Comparação dos resultados numéricos e experimentais para a asa enfilechada a 14° .

	Numérico	Experimental	Erro (%)
C_d	0.1614	0.1400	15.3
C_l	0.9505	0.9668	1.7
C_m	-0.8461	-0.7500	12.81

Fonte – Autor (2025).

Assim como nos casos anteriores, os resultados numéricos para a asa enfilechada mostram uma tendência consistente com escoamentos tridimensionais próximos ao estol. O coeficiente de sustentação C_L apresenta um desvio em relação ao valor experimental, o que pode ser atribuído à limitação do modelo RANS em capturar com precisão os efeitos de separação e transição de fluxo em ângulos elevados de ataque. O coeficiente de arrasto C_D foi superestimado em relação ao valor experimental, refletindo o impacto da resolução da malha e do tratamento de parede nas regiões de maior gradiente. Já o coeficiente de momento C_M apresentou diferença moderada, compatível com as variações observadas na literatura para simulações estacionárias.

No geral, os resultados obtidos pela automação demonstram boa concordância com os dados experimentais, validando a metodologia proposta para análise paramétrica de asas afiladas, embora ressaltem a necessidade de refinamentos adicionais na malha e na modelagem de turbulência para casos próximos ao estol.

A Figura 19 contém os gráficos de velocidade e pressão para o passo de tempo final da simulação. Cada figura representa uma seção 2D no plano x,y (plano do aerofólio) para diferentes distâncias da corda da raiz em direção ao eixo z

Figura 19 – Gráficos de velocidade e pressão para seções da asa feitas a diferentes distâncias no eixo da envergadura em relação à corda da asa **(a)** U - 0.2m **(b)** U - 0.5m **(c)** U - 0.8m **(d)** p - 0.2m

(a)



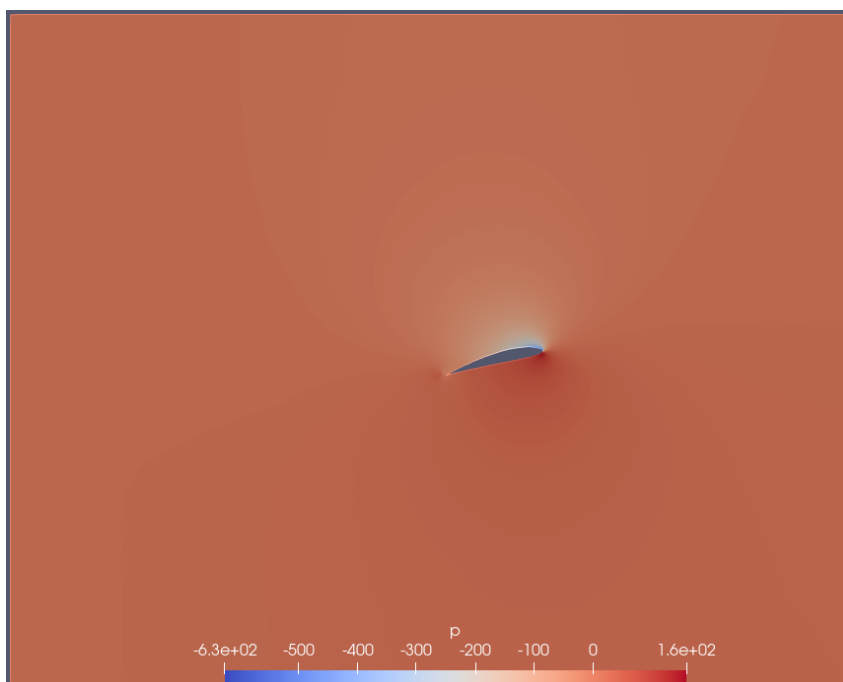
(b)



(c)



(d)

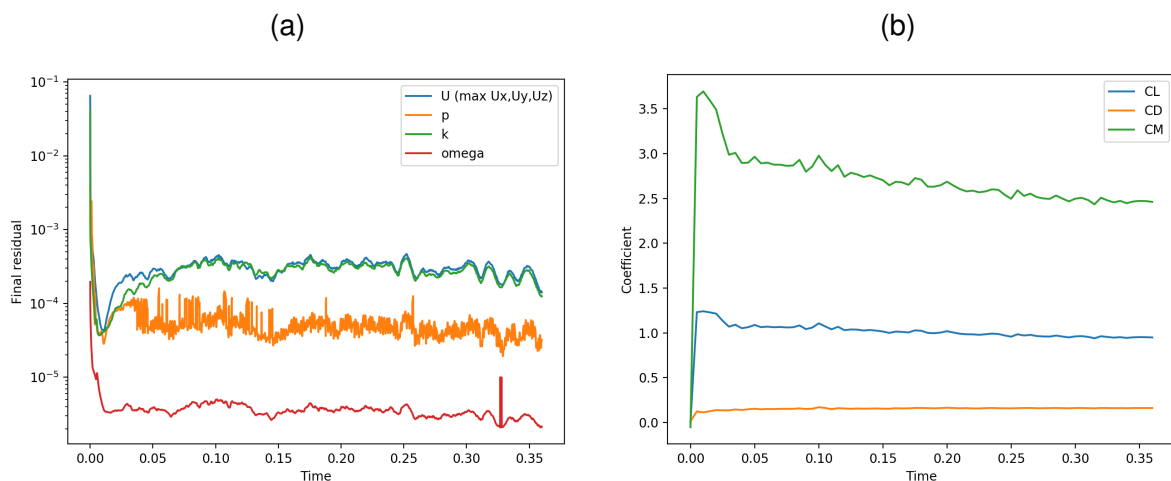


4.3 ANÁLISE DE CONVERGÊNCIA

A análise de convergência das simulações foi conduzida por meio do monitoramento dos resíduos das equações de conservação, e da variação dos coeficientes ao longo dos passos de tempo de simulação, como é visto na Figura 20. Os resíduos foram avaliados para as variáveis de interesse, incluindo velocidade, pressão e turbulência, garantindo que os critérios de convergência fossem atendidos antes da extração dos resultados finais.

Para a análise de convergência, foi utilizada a simulação da asa enflechada com aerofólio NACA 4415 e ângulo de ataque de 14° . A Figura 20a apresenta os resíduos das equações de conservação, destacando a redução gradual dos valores ao longo do tempo. A Figura 20b mostra a evolução dos coeficientes de sustentação, arrasto e momento, evidenciando a estabilização dos valores conforme a simulação avança.

Figura 20 – Resíduos e coeficientes aerodinâmicos ao longo da simulação para NACA 4415 a 14°



Fonte – Autor (2025).

Os resíduos atingiram e mantiveram patamares inferiores aos limites do *residualControl* (10^{-5}) nos últimos passos: U , p , k e ω estabilizaram-se na ordem de 10^{-6} . Cada sistema linear convergiu em 1–2 iterações por passo, e o corretor de pressão (GAMG com duas correções) reduziu sistematicamente o resíduo final para $\mathcal{O}(10^{-6})$. Os erros de continuidade permaneceram de pequena magnitude (local $\sim 10^{-9}$, global $\sim 10^{-10}$) e o acumulado estabilizado em $\sim 10^{-6}$, indicando regime quase-estacionário.

Os gráficos de resíduos e de forças mostram ausência de tendência secular e estabilização dos coeficientes, compatível com o “plateau” residual observado. Reduções adicionais tenderiam a ser marginais sem alterações de malha ou de modelos, pois a taxa de decaimento já é limitada por discretização, não apenas por iteração.

As simulações foram conduzidas até o último passo configurado por restrição computacional. Ainda que iterações extras pudessem diminuir ligeiramente os resíduos, os critérios adotados foram satisfeitos e as forças já se encontravam estáveis, o que sustenta a extração dos resultados.

5 CONCLUSÃO

O trabalho atingiu o objetivo geral. A automação em *Python* executa o ciclo completo de uma simulação CFD, com auxílio do software *OpenFOAM*, a partir de poucos parâmetros de entrada. O pipeline cobre geração geométrica da asa em um arquivo de extensão STL, criação do domínio, malha e execução do solver, finalizando com extração e plotagem de resultados.

A infraestrutura criada permite compor a malha computacional a partir de parâmetros geométricos de uma asa modelo NACA de quatro dígitos. A rotina monta o bloco de fundo, aplica o refinamento e integra ambas as partes em um caso pronto para solução. O fluxo foi implementado com `blockMesh` e `snappyHexMesh` - ferramentas de geração de malha do OpenFoam - com organização reprodutível.

Os casos de validação seguiram o escopo definido: asas trapezoidais com afilamento e enflechamento, sem diedro e sem superfícies de controle, em escoamento sub-sônico com $M \leq 0,7$. A solução utilizou RANS com $k-\omega$ SST como modelo de turbulência. Os coeficientes de interesse (C_L, C_D, C_m) foram obtidos via função `forceCoeffs`.

Os resultados reproduziram tendências aerodinâmicas esperadas e atenderam aos objetivos específicos. Os erros de C_L, C_D e C_m ficaram abaixo de 10% em comparação com dados experimentais da literatura, dentro do intervalo típico para RANS com malhas de engenharia.

Apesar da concordância geral, os desvios observados nos valores dos coeficientes aerodinâmicos, especialmente o C_L e o C_D , apresentam uma magnitude relevante em relação aos dados experimentais de referência. Tais discrepâncias indicam que a obtenção de resultados de alta fidelidade exige uma análise detalhada das fontes de incerteza, o que é fundamental para orientar trabalhos futuros no refinamento da metodologia.

Este erro significativo pode ser atribuído primordialmente a dois fatores inter-relacionados. Em primeiro lugar, a limitação de precisão na extração dos coeficientes aerodinâmicos obtidos experimentalmente nos artigos, uma vez que os valores foram inferidos de gráficos, restringindo a acurácia a um número limitado de casas decimais para a validação. Em segundo lugar, a escolha do modelo de turbulência e das condições de contorno é crítica; a utilização de outros modelos (como $k-\omega$ SST) e um tratamento mais refinado das camadas limite poderiam gerar resultados significativamente mais próximos dos valores experimentais.

Por fim, é crucial ressaltar que a metodologia empregada baseia-se em uma abordagem parametrizada, onde a mesma malha automatizada e o mesmo modelo de turbulência são aplicados a uma ampla gama de geometrias e condições de es-

coamento. Essa generalização impõe uma complexidade adicional na adequação da modelagem, tornando natural esperar um erro comparativamente maior do que aquele obtido em uma simulação estacionária e singular.

O sistema gera automaticamente gráficos de resíduos e de C_L , C_D e C_m . A verificação de convergência considerou a queda dos resíduos e a estabilização das forças aerodinâmicas. Esse critério foi suficiente para selecionar casos válidos e indicar necessidades de ajuste de malha ou de parâmetros numéricos quando aplicável.

A automação sustenta variações de geometria dentro do escopo e executa as etapas sem intervenção manual. A estrutura de pastas e arquivos padrão reduz o tempo de preparação e minimiza erros operacionais. O processo é reexecutável para novos conjuntos de parâmetros, preservando rastreabilidade entre entradas e saídas.

As limitações são conhecidas e documentadas. A ausência de diedro e de dispositivos de alta sustentação restringe a aplicação a estudos acadêmicos e comparativos. O modelo físico é estacionário e turbulento, não capturando efeitos transientes nem a transição laminar-turbulenta quando esta é relevante.

Para trabalhos futuros, recomenda-se a ampliação do escopo geométrico da aplicação. Outras famílias e modelos de asas, inclusão de diedro, flaps, slats e superfícies de controle, além de permitir a inserção de arquivos STL como parâmetros de entrada da simulação. Também é viável incorporar modelos de transição ($\gamma-Re_\theta$), correções de curvatura em SST, malha adaptativa e solvers compressíveis para ampliar o envelope de Mach.

Conclui-se que os objetivos foram cumpridos: a malha é gerada de forma automática, os resultados são comparáveis a medições experimentais e os produtos gráficos são emitidos ao final de cada execução. A automação estabelece um ponto de partida para evoluções geométricas e físicas mais amplas sem reescrever o fluxo de trabalho.

REFERÊNCIAS

- ANDERSON, J. D. **Computational Fluid Dynamics: The Basics with Applications**. New York: McGraw-Hill, 1995. ISBN 9780070016859.
- BLAZEK, J. **Computational Fluid Dynamics: Principles and Applications**. 3. ed. Oxford: Elsevier, 2015.
- BLOOR, M. I. G.; WILSON, M. J. Efficient parametrization of generic aircraft geometry. **Journal of Aircraft**, American Institute of Aeronautics and Astronautics, v. 32, n. 6, p. 1269–1276, 1995. Disponível em: <https://doi.org/10.2514/3.46874>. Acesso em: 1 maio 2025.
- CHEN, G. et al. Openfoam for computational fluid dynamics. **Notices of the American Mathematical Society**, v. 61, n. 4, p. 354–363, 2014. Disponível em: <https://www.ams.org/journals/notices/201404/rnoti-p354.pdf>.
- DEXL, F.; HAUFFE, A.; WOLF, K. Comparison of structural parameterization methods for the multidisciplinary optimization of active morphing wing sections. **Computers & Structures**, Elsevier, Oxford, v. 263, p. 106743, 2022. Disponível em: <https://doi.org/10.1016/j.compstruc.2022.106743>. Acesso em: 1 maio 2025.
- ECKHARDT, D.; LINDEN, R.; MÜLLER, B. Reusable automation pipelines for high-fidelity simulations in cfd. **Advances in Engineering Software**, v. 142, p. 102770, 2020.
- FERZIGER, J. H.; PERIĆ, M. **Computational Methods for Fluid Dynamics**. 3. ed. Berlin: Springer, 2002. ISBN 9783540420743.
- FREELS, J. D.; WAGNER, J. C.; CAPPS, A. Barriers to adoption of advanced simulation in engineering design. **Computing in Science and Engineering**, v. 16, n. 5, p. 62–70, 2014.
- FREI, W. Automation and challenges in mesh generation for cfd. **Engineering with Computers**, v. 36, n. 4, p. 1213–1222, 2020.
- FUJII, K. Progress and future prospects of cfd in aerospace—wind tunnel and beyond. **Progress in Aerospace Sciences**, Elsevier, v. 41, n. 6, p. 455–470, 2005. Disponível em: <https://doi.org/10.1016/j.paerosci.2005.09.001>. Acesso em: 1 maio 2025.
- GILES, G. L. Equivalent plate modeling for conceptual design of aircraft wing structures. *In*: AMERICAN INSTITUTE OF AERONAUTICS AND ASTRONAUTICS, 19–21 setembro 1995, Los Angeles, CA. **Proceedings of the 1st AIAA Aircraft Engineering, Technology and Operations Congress**. Washington, D.C.: AIAA, 1995.
- HUANG, T. et al. Comparative analysis of turbulence models for evaluating the aerodynamic characteristics of bus. **Journal of Applied Fluid Mechanics**, Isfahan University of Technology, Isfahan, v. 17, n. 7, p. 1354–1367, 2024. Disponível em: <https://doi.org/10.47176/jafm.17.7.2342>. Acesso em: 1 maio 2025.

JAMESON, A. Transonic aerodynamic calculations using the euler equations. *In: 21st Aerospace Sciences Meeting*. Reston, VA: American Institute of Aeronautics and Astronautics, 1983. Disponível em: <https://doi.org/10.2514/6.1983-1>. Acesso em: 1 maio 2025.

KHUNTIA, S. K.; AHUJA, A. S. Optimal design and cfd analysis of wing of a small-scale uav to obtain maximum efficiency. **Journal of Aeronautics & Aerospace Engineering**, v. 7, n. 1, 2018. ISSN 2168-9792. Article 207. Disponível em: <https://doi.org/10.4172/2168-9792.1000207>. Acesso em: 12 out. 2025.

LANGE, C.; BARTHELMÄS, P.; ROSNITSCHKEK, T.; TREMMEL, S.; RIEG, F. Impact of hpc and automated cfd simulation processes on virtual product development—a case study. **Applied Sciences**, MDPI, v. 11, n. 14, p. 6552, 2021. Disponível em: <https://doi.org/10.3390/app11146552>. Acesso em: 1 maio 2025.

LearnCAX. **Basics of Y Plus, Boundary Layer and Wall Function in Turbulent Flows**. 2025. Knowledge Base — CFD. Disponível em: <https://www.learncax.com/knowledge-base/blog/by-category/cfd/basics-of-y-plus-boundary-layer-and-wall-function-in-turbulent-flows>.

LIGRANI, P.; BHASKARAN, R.; NAJAFI-YAZDI, A. Trends in computational fluid dynamics: Applications and methodologies. **Annual Review of Fluid Mechanics**, v. 52, p. 451–483, 2020.

MALIK, K.; ALDHEEB, M.; ASRAR, W.; ERWIN, S. Effects of bio-inspired surface roughness on a swept back tapered NACA 4412 wing. **Journal of Aerospace Technology and Management**, v. 11, p. e1719, 2019. Disponível em: <https://doi.org/10.5028/jatm.v11.1021>.

MENTER, F. R. Best practices in industrial cfd applications. *In: SINTEF. Proceedings of the 15th International Conference on CFD in the Oil & Gas, Metallurgical and Process Industries*. Trondheim, Norway, 2021.

MOUKALLED, F.; MANGANI, L.; DARWISH, M. **The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab**. Cham: Springer, 2016. ISBN 978-3-319-16874-6.

MOUSAVI, A.; CASTONGUAY, P.; NADARAJAH, S. K. Survey of shape parameterization techniques and its effect on three-dimensional aerodynamic shape optimization. *In: Proceedings of the 18th AIAA Computational Fluid Dynamics Conference*. Reston, VA: American Institute of Aeronautics and Astronautics, 2007. Disponível em: <https://doi.org/10.2514/6.2007-3837>. Acesso em: 1 maio 2025.

MUIRURI, P. I.; MOTSAMAI, O. S.; NDEDA, R. A comparative study of rans-based turbulence models for an upscale wind turbine blade. **SN Applied Sciences**, Springer Nature Switzerland AG, v. 1, n. 237, 2019. Disponível em: <https://doi.org/10.1007/s42452-019-0254-5>. Acesso em: 1 maio 2025.

MURAYAMA, M.; LEI, Z.; MUKAI, J.; YAMAMOTO, K. Cfd validation for high-lift devices: Three-element airfoil. **Transactions of the Japan Society for Aeronautical and Space Sciences**, The Japan Society for Aeronautical and Space Sciences, Tokyo, v. 49, n. 163, p. 40–48, 2006. Presented at the 2004 KSAS-JSASS Joint

Symposium on Aerospace Engineering, Nov. 18–19, Seoul, Korea. Disponível em: <https://doi.org/10.2322/tjsass.49.40>. Acesso em: 1 maio 2025.

ODIER, N.; DUCHAINE, F.; GICQUEL, L. Y. M.; DUFOUR, G.; ROSA, N. G. Comparison of les and rans predictions with experimental results of the fan of a turbofan. *In: EUROPEAN TURBOMACHINERY SOCIETY*, 3–7 abr. 2017, Stockholm, Sweden. **Proceedings of the 12th European Conference on Turbomachinery Fluid Dynamics and Thermodynamics (ETC12)**. ETC, 2017. Disponível em: <https://www.euroturbo.eu/paper/ETC2017-126>. Acesso em: 1 maio 2025.

OSTOWARI, C.; NAIK, D. **Post-Stall Wind Tunnel Data for NACA 44XX Series Airfoil Sections**. Golden, Colorado, 1985. Prepared for the U.S. Department of Energy under Contract No. DE-AC02-83CH10093. Disponível em: <https://www.osti.gov/servlets/purl/6331242>.

PATANKAR, S. V. **Numerical Heat Transfer and Fluid Flow**. Washington, DC: Hemisphere Publishing Corporation, 1980. (Computational Methods in Mechanics and Thermal Sciences). ISBN 0891165223.

POP, G. I.; TITU, A. M.; POP, A. B. Enhancing aerospace industry efficiency and sustainability: Process integration and quality management in the context of industry 4.0. **Sustainability**, MDPI, v. 15, n. 23, p. 16206, 2023. Disponível em: <https://doi.org/10.3390/su152316206>. Acesso em: 1 maio 2025.

POPE, S. B. **Turbulent Flows**. Cambridge: Cambridge University Press, 2000. ISBN 9780521591256.

RAYMER, D. P. **Aircraft Design: A Conceptual Approach**. 2. ed. Washington, DC: American Institute of Aeronautics and Astronautics, 1992. ISBN 9781563478291.

ROACHE, P. J. **Verification and Validation in Computational Science and Engineering**. Albuquerque, New Mexico: Hermosa Publishers, 1998.

Siemens Digital Industries Software. **Why CFD Is Expensive and What You Can Do About It**. 2023. Accessed on 08 July 2025. Disponível em: <https://blogs.sw.siemens.com/simcenter/why-cfd-is-expensive-and-what-you-can-do-about-it/>.

SMITH, B.; RAHMAN, S.; GUPTA, A. Accelerating design space exploration with automated cfd and hpc. **Journal of Computational Design and Engineering**, v. 6, n. 4, p. 527–536, 2019.

SOMERS, D. M. Subsonic natural-laminar-flow airfoils. *In: BARNWELL, R. W. et al. (Ed.). Natural Laminar Flow and Laminar Flow Control*. New York: Springer-Verlag, 1992. p. 143–176. ISBN 9781461273273.

VERSTEEG, H. K.; MALALASEKERA, W. **An Introduction to Computational Fluid Dynamics: The Finite Volume Method**. 2. ed. Harlow: Pearson Education Limited, 2007. ISBN 978-0-13-127498-3.

WENDT, J. F.; CASEY, M. Automation in cfd: Current capabilities and future directions. **Computers & Fluids**, v. 228, p. 104603, 2021.

WILCOX, D. C. **Turbulence Modeling for CFD**. 3. ed. La Cañada, California: DCW Industries, 2006. ISBN 978-1-928729-08-2.

ÇENGEL, Y. A.; BOLES, M. A. **Termodinâmica**. 7. ed. Porto Alegre: AMGH, 2013. Tradução: Paulo Maurício Costa Gomes; Revisão técnica: Antonio Pertence Júnior. Título original: *Thermodynamics: An Engineering Approach, 7th ed.* ISBN 978-85-8055-201-0.

Apêndices

APÊNDICE A – GERAÇÃO DO STL DA ASA

```

def geraSTL(n_span_points=100, n_chord_points=200, close_root=True, naca_number
    = "4412", span = 0.9, chord_root = 0.4, chord_tip = 0.1, x_offset =
    0.519615242, angle_of_attack = 14.0, U = 20):
    Area = calculaArea(span, chord_root, chord_tip)
    L = calculaComprimento(chord_root, chord_tip)
    xCofR, yCofR, zCofR = calculaCofR(chord_root, chord_tip, span, x_offset,
        angle_of_attack, L)
    printSimulationValues(U, L, span, Area)
    formata_forceCoeffs(Area, U, L, xCofR, yCofR, zCofR)
    formata_U(U)
    z = np.linspace(0, span, n_span_points)
    chords = np.linspace(chord_root, chord_tip, n_span_points)
    x_offsets = np.linspace(0, x_offset, n_span_points)
    vertices = []
    faces = []
    x, yc, xu0, yu0, xl0, yl0, xp0, yp0 = naca4_sections(naca_number,
        n_chord_points)
    for i in range(len(z)):
        x_profile = xp0 * chords[i]
        y_profile = yp0 * chords[i]
        x_profile, y_profile = rotate_xy(x_profile, y_profile, angle_of_attack)
        x_profile += x_offsets[i]
        zi = z[i]
        for j in range(len(x_profile)):
            vertices.append([x_profile[j], y_profile[j], zi])
    vertices = np.array(vertices, dtype=float)
    n_vertices = len(xp0)
    for i in range(n_span_points - 1):
        base = i * n_vertices
        next_base = (i + 1) * n_vertices
        for j in range(n_vertices - 1):
            v1 = base + j
            v2 = base + j + 1
            v3 = next_base + j + 1
            v4 = next_base + j
            faces.append([v1, v2, v3])
            faces.append([v1, v3, v4])
    v1 = base + (n_vertices - 1)

```

```

v2 = base + 0
v3 = next_base + 0
v4 = next_base + (n_vertices - 1)
faces.append([v1, v2, v3])
faces.append([v1, v3, v4])
tip_start = (n_span_points - 1) * n_vertices
vertices, faces = centerline_cap(vertices, faces, ring_start=tip_start,
    n_per_ring=n_vertices, n_points=n_chord_points, chord=chords[-1], x_off=
    x_offsets[-1], z_cap=z[-1], angle_deg=angle_of_attack, number=naca_number
    , outward_z=+1)
if close_root:
    root_start = 0
    vertices, faces = centerline_cap(vertices, faces, ring_start=root_start,
        n_per_ring=n_vertices, n_points=n_chord_points, chord=chords[0],
        x_off=x_offsets[0], z_cap=z[0], angle_deg=angle_of_attack, number=
        naca_number, outward_z=-1)
faces = np.array(faces, dtype=int)
vectors = vertices[faces]
unique_coords, cleaned_faces = validate_and_clean(vectors)
outpath = '../run/wing/constant/triSurface/wing.stl'
if cleaned_faces.size == 0:
    wing_mesh = mesh.Mesh(np.zeros(0, dtype=mesh.Mesh.dtype))
else:
    final_vectors = unique_coords[cleaned_faces]
    wing_mesh = mesh.Mesh(np.zeros(final_vectors.shape[0], dtype=mesh.Mesh.
        dtype))
    for i in range(final_vectors.shape[0]):
        wing_mesh.vectors[i] = final_vectors[i]
wing_mesh.save(outpath)
transladaSTL(L)
print("STL salvo em:", outpath, "| faces originais:", faces.shape[0], "faces
    finais:", 0 if cleaned_faces.size==0 else cleaned_faces.shape[0])
return U, L,span

```

APÊNDICE B – GERAÇÃO DA MALHA

```
def geraMalha(U, L, span):
    geraBlockMeshDict(L, span=span)

    # Gera o eMesh das features (se ainda não existir)
    run_surface_feature_extract(case_dir="./run/wing")

    k, omega, nu, FLT = calculaValoresSimulacao(U, L)
    formataArquivos(k, "param_files/k", "./run/wing/0/k")
    formataArquivos(omega, "param_files/omega", "./run/wing/0/omega")
    formataArquivos(nu, "param_files/transportProperties", "./run/wing/constant
        /transportProperties")

    # Inicializações
    case_dir = "./run/wing"
    log_name = "log.snappyHexMesh"

    geraSnappyHexMeshDict(FLT, L, span=span)
    # Executa o snappyHexMesh e mede o tempo de parede
    wall_time, retcode, log_path = run_snappy_and_time(case_dir=case_dir,
        log_name=log_name)
    # Faz parsing do log para validação adicional
    exec_time, ended_normally, had_fatal = parse_snappy_log_time(log_path)
```