



**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
CURSO DE ENGENHARIA AERONÁUTICA**

**Metamodelagem e Machine Learning em estudo de interação
fluido-estrutura de uma esfera em rotação**

Nuno Silas Benrós Santos

Uberlândia
2023

Nuno Silas Benrós Santos

**Metamodelagem e Machine Learning em estudo de interação
fluido-estrutura de uma esfera em rotação**

Projeto de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Aeronáutica da Faculdade de Engenharia Mecânica como pré-requisito para a obtenção do Título de Bacharel em Engenharia Aeronáutica

Área de Concentração: Simulação Numérica, Interação fluido estrutura, Metamodelagem, Machine learning

Linha de pesquisa: Mecânica dos Fluidos Computacional

Orientador: Prof. Dr. João Marcelo Vedovotto

Uberlândia
2023

Nuno Silas Benrós Santos

**Metamodelagem e Machine Learning em estudo de interação
fluido-estrutura de uma esfera em rotação**

Projeto de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Aeronáutica da Faculdade de Engenharia Mecânica como pré-requisito para a obtenção do Título de Bacharel em Engenharia Aeronáutica.

Área de Concentração: Simulação Numérica, Interação fluido estrutura, Metamodelagem

Linha de pesquisa: Mecânica dos Fluidos Computacional

Uberlândia, 20 de Maio de 2023

Banca examinadora:

Prof. Dr. João Marcelo Vedovotto, Orientador (UFU)

Me. Alex José Elias

Me. Felipe Adriano da Silva Gonçalves

Agradecimentos

Agradeço a Deus pela existência de um cosmos que possa ser minimamente interpretado, grato em particular pela minha existência.

Agradeço a minha esposa que tem sido minha companheira nessa viagem que é a vida.

Agradeço o amor, a sabedoria e tudo que minha mãe fez por mim ao longo da minha existência, sou produto dos seus feitos e das suas orações. Agradeço ao meu falecido pai, que me ensinou tanto, e que nos seus últimos dias me deu a motivação para realizar esse sonho de me graduar no curso de Engenharia Aeronáutica.

Agradeço ao meu orientador pelo aprendizado e pela oportunidade de trabalhar com esse tema. Sou grato aos meus colegas do MFLab pelo suporte e ensinamentos que viabilizaram a produção desse trabalho.

Por fim, agradeço a todos os meus amigos, familiares, professores, indivíduos em geral que fizeram parte do meu desenvolvimento, afinal "É preciso uma aldeia inteira para educar uma criança"(Provérbio Africano).

"When I meet God, I am going to ask him two questions: Why relativity ? And why turbulence? I really believe he will have an answer for the first."

Werner Heisenberg

Resumo

A dinâmica de fluidos computacional vem sendo uma das maiores revoluções nos estudos da física aplicada. Entender como se dá a interação entre estruturas e fluidos é um ramo basilar das indústrias como a aeronáutica, marítima e petrolífera. No âmbito do estudo de escoamentos, a análise de uma esfera imersa em um fluido se torna um caso canônico. No presente trabalho foi utilizado um código de CFD para a simulação do comportamento do escoamento sobre uma esfera em várias rotações sobre o próprio eixo. Foi utilizado uma malha adaptativa em combinação com o método da fronteira imersa. Os dados obtidos por sondas ao longo do domínio da simulação foram utilizadas na produção de um metamodelo capaz de prever a rotação da esfera através da variação das velocidades registradas. O metamodelo foi constituído pelo algoritmo de aprendizado de máquina XGBoost, que foi capaz de prever a velocidade de rotação através dos dados da simulação.

Palavras-chave: Metamodelagem, Fronteira Imersa, Machine Learning.

Abstract

Computational fluid dynamics has been one of the biggest revolutions in applied physics studies. Understanding how the interaction between structures and fluids occurs is a basic field of industries such as aeronautics, maritime and oil. Within the scope of the study of flows, the analysis of a sphere immersed in a fluid becomes a canonical case. In the present work, a CFD code was used to simulate the behavior of the flow over a sphere at various rotations about its own axis. An adaptive mesh was used in combination with the immersed boundary method. The data obtained by probes along the simulation domain were used in the production of a metamodel capable of predicting the rotation of the sphere through the variation of the recorded velocities. The metamodel was constituted by the XGBoost machine learning algorithm, which was able to predict the rotation speed through the simulation data.

Keywords: Metamodeling, Immersed Boundary, Machine Learning.

Lista de ilustrações

Figura 1 – escoamento em um aerofólio.	1
Figura 2 – Queda da ponte de Tacoma	2
Figura 3 – Esquema da metodologia do trabalho	4
Figura 4 – Relação entre o tamanho de vórtices e o logaritmo do número de Reynolds	6
Figura 5 – Os quatro regimes de escoamento	6
Figura 6 – Efeito Magnus e Magnus inverso em um bola de golfe	8
Figura 7 – Exemplo de redução de dimensionalidade feito com PCA em um conjunto de dados com 3 variáveis (genes).	12
Figura 8 – Refinamento Localizado	15
Figura 9 – Exemplo de malha composta por três níveis de refinamento.	16
Figura 10 – Domínio da Simulação	16
Figura 11 – Níveis físicos	17
Figura 12 – Representação esquemática da relação entre a malha euleriana e a malha lagrangiana	17
Figura 13 – Malha da esfera	18
Figura 14 – Domínio.	20
Figura 15 – Representação da malha de sondas produzida para obter os dados da simulação	20
Figura 16 – Escoamento $\omega = 0 \text{ rad/s}$	21
Figura 18 – Escoamento $\omega = -20 \text{ rad/s}$	21
Figura 17 – Escoamento $\omega = -10 \text{ rad/s}$	22
Figura 19 – Escoamento $\omega = 10 \text{ rad/s}$	22
Figura 20 – Escoamento $\omega = 20 \text{ rad/s}$	22
Figura 21 – Gráfico porcentagem de variância explicada por Componente Principal	23
Figura 22 – Primeira árvore de decisão com hiperparâmetros default (Autoria Própria)	25
Figura 23 – Gráfico comparação entre Busca aleatória (Random) e Otimização Bayesiana (Bayesian)	26
Figura 24 – Primeira árvore de decisão com hiperparâmetros da otimização Bayesiana	27
Figura 25 – Primeira árvore de decisão com os hiperparâmetros definidos	28
Figura 26 – Histograma das previsões para a rotação $\omega = 12$	29

Lista de tabelas

Tabela 1 – Pârametros Físicos	19
Tabela 2 – Hiperparâmetros default	24
Tabela 3 – Limites inferiores e superiores da otimização	26
Tabela 4 – Hiperparâmetros definidos pela Otimização Randômica	27
Tabela 5 – Hiperparâmetros definidos pela Otimização Bayesiana	27
Tabela 6 – Hiperparâmetros escolhidos para a previsão da rotação	28
Tabela 7 – Medidas de tendência central e dispersão para a previsão da rotação $\omega = 12$	28

Lista de Símbolos

Abreviações

AMR Adaptive Mesh Refinement

CFD Computational Fluid Dynamics

CUBISTA Convergent and Universally Bounded Interpolation Scheme for the Treatment of
Avection

LBOT Nível base da malha

LTOP Nível mais fino da malha

ML Machine Learning

PCA Principal Component Analysis

XGBoost Extreme Gradient Boosting

Constantes Físicas

ω	Rotação	$[rad/s]$
ρ	Massa específica do fluido	$[kg/m^3]$
F	Força lagrangeana, com o sub-índice i ($i = 1, 2, 3$)	$[N/m^3]$
f	Termo fonte genérico; força euleriana, com o sub-índice i ($i = 1, 2, 3$)	$[N/m^3]$
N_{gl}	Número de graus de liberdade	$[-]$
p	Pressão	$[Pa]$
t	Tempo	$[s]$
u	Velocidade, com os sub-índices ij ($i, j = 1, 2, 3$)	$[m/s]$

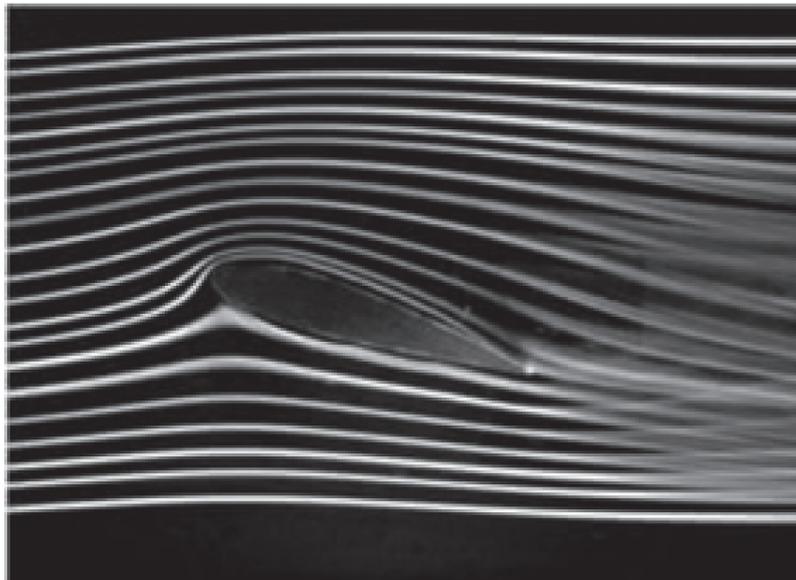
Sumário

Lista de ilustrações		I
Lista de tabelas		II
1	INTRODUÇÃO	1
1.1	Objetivos	3
1.2	Metodologia	3
2	REVISÃO BIBLIOGRÁFICA	5
2.1	Escoamento sobre uma Esfera	5
2.2	Efeito Magnus	7
2.3	Meta-modelagem e Machine learning	9
3	REFERENCIAL TEÓRICO	10
3.1	Modelagem Matemática	10
3.2	Fronteira Imersa	10
3.3	PCA e XGBoost	11
3.3.1	Hiperparâmetros	13
4	PARÂMETROS DE SIMULAÇÃO	15
4.1	Malha	15
4.1.1	Malha euleriana	17
4.1.2	Malha lagrangeana	18
4.2	Parâmetros Físicos	18
4.3	Parâmetros de controle da simulação	19
4.4	Condições de contorno	19
4.5	Sondas	20
5	ANÁLISE DE RESULTADOS	21
5.1	Análise do Escoamento	21
5.2	Machine Learning	23
6	CONCLUSÃO	30
	REFERÊNCIAS	31

1. Introdução

A interação fluido estrutura é um ramo da engenharia que requer um esforço multidisciplinar para compreender de forma satisfatória os fenômenos relacionados. Portanto, aborda problemas complexos que exigem sinergia entre áreas da engenharia, como a engenharia mecânica, computacional e outras. O estudo da interação entre fluido e estrutura tem sido um campo em constante evolução desde o surgimento da mecânica dos fluidos. Esta área de pesquisa tem sido cada vez mais explorada, pois permite analisar o comportamento de um sistema no seu ambiente. O entendimento desta relação é fundamental para a análise de diversos processos, como previsão de vibrações e adaptação de estruturas para ações de fluidos. O estudo de estruturas e escoamentos envolve modelos matemáticos, experimentos e simulações computacionais, com o objetivo de melhorar a compreensão desta relação. Esta área é aplicada em vários campos, como engenharia aeronáutica, indústria de petróleo e gás, engenharia naval, entre outros. Ela envolve analisar como o fluido afeta o comportamento da estrutura e vice-versa. Esta ferramenta pode ser usada para estudar uma variedade de fenômenos, desde o escoamento de ar em torno de uma asa de aeronave (Figura 1) até a vibração de uma ponte devido ao vento (Figura 2).

Figura 1 – Escoamento em um aerofólio.



Fonte: H. Babinski, University of Cambridge (BABINSKY, 2003)

A modelagem matemática é uma ferramenta importante para traduzir a modelagem física para a forma discreta, permitindo a análise destes fenômenos. A interação fluido-estrutura é composta por mecanismos complexos e estuda os efeitos de vários fatores, tais como a intera-

ção entre camadas de fluido, pressão e forças de arrasto, se mostrando crucial para entender e prever o comportamento de muitos sistemas de engenharia, permitindo prever o desempenho, a estabilidade e a segurança desses sistemas.

Figura 2 – Queda da ponte de Tacoma



Fonte: Retirado de (WASHINGTON, 1940)

A análise do comportamento entre fluidos e estruturas requer um esforço multidisciplinar para compreender de forma satisfatória os fenômenos relacionados. No âmbito da pesquisa neste campo do conhecimento, a abordagem experimental é fundamental. Experimentos são realizados tanto de forma material quanto computacional. A experimentação computacional tem apresentado inúmeras vantagens, graças ao progresso tecnológico exponencial no desenvolvimento de máquinas cada vez mais poderosas. Experimentos físicos, às vezes em escala, requerem uma modelagem física robusta, simulando o ambiente a ser estudado com suposições físicas gerais (para viabilizar o experimento), necessitam de materiais à disposição e espaço em laboratórios controlados onde variáveis externas ao experimento sejam minimizadas. Tudo isso requer um gasto financeiro considerável e grupos de pesquisa aprimorados nas técnicas de instrumentação para que a extração dos dados seja válida. A simulação computacional veio complementar a experimentação física, preenchendo espaços onde o experimento físico é considerado inviável por vários motivos, como o financeiro, a rapidez e a insalubridade (fator de risco). No entanto, é necessário ter recursos humanos habilidosos em modelagem numérica e um poder computacional cada vez maior (o que vem sendo suprido pelo desenvolvimento tecnológico fulminante dos processadores centrais e gráficos). Na experimentação computacional, é necessário que as suposições físicas sejam traduzidas em uma modelagem matemática capaz de representar o problema estudado. Nesse âmbito, o cálculo diferencial se apresenta como uma ferramenta poderosa de trabalho nas engenharias. As equações diferenciais são as ferramentas que permitem traduzir a modelagem física para a discretização numérica - a base da simulação

computacional. Junto com a evolução do hardware dos computadores, o advento do aprendizado de máquina e da metamodelagem revolucionaram a maneira como abordamos a análise de dados. Aproveitando o poder dos algoritmos e da inteligência artificial, existe a possibilidade de processar grandes quantidades de dados com grande velocidade e precisão, permitindo extrair *insights* que antes eram impensáveis de obter. Por meio da metamodelagem, é possível criar modelos que podem ser usados para prever resultados futuros e otimizar processos já estabelecidos. Da mesma forma, também possibilita a criação de modelos adaptáveis, capazes de se ajustar de acordo com o contexto e as variações nos dados. Em suma, Machine Learning e metamodelagem desenvolveram um mundo de possibilidades para a análise de dados, permitindo-nos obter resultados que antes eram inimagináveis.

A esfera, como sólido geométrico simples, é a base de estruturas na indústria. Neste trabalho, é abordado a simulação de uma esfera submersa em um fluido, aplicando o método da fronteira imersa (IB) e utilizando o código de Dinâmica de Fluidos Computacional MFSim, desenvolvido no Laboratório de Mecânica dos Fluidos da Universidade Federal de Uberlândia (MFlab). Os resultados das simulações são obtidos por meio de sondas dispersas ao longo do domínio proposto, que fornecem os parâmetros de entrada para o modelo de Machine Learning (ML) que se propõe a prever a rotação da esfera.

1.1 Objetivos

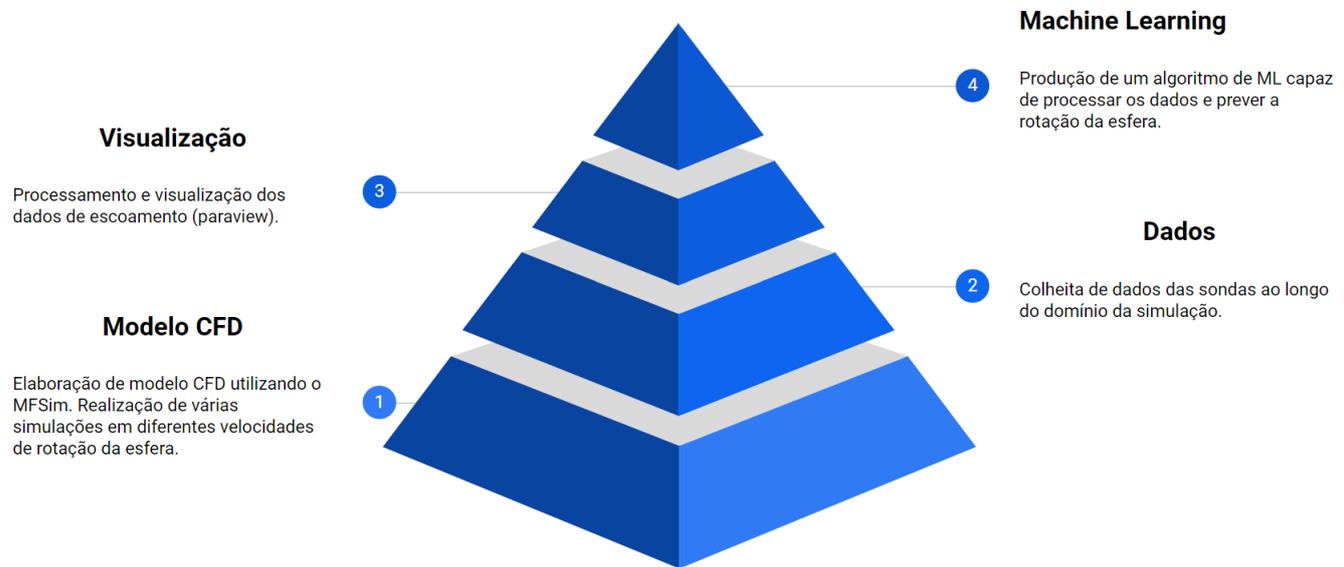
O presente trabalho tem como objetivo geral analisar o comportamento de uma esfera em rotação imersa em um fluido através de simulações numéricas e aplicar uma metamodelagem com Machine learning, com o intuito de obter a rotação da mesma. Para isso, foram estabelecidos objetivos específicos como compreender a interação Fluido-Estrutura de uma esfera em rotação, criar um modelo computacional (modelagem física e numérica) que represente a esfera imersa no fluido, analisar visualmente o escoamento e verificar a influência da velocidade de rotação da esfera no status da interação fluido-estrutura. Além disso, pretende-se aplicar uma metamodelagem com machine learning para prever a rotação da esfera, otimizando assim o processo de análise da interação fluido-estrutura da esfera em rotação.

1.2 Metodologia

Uma pesquisa exploratória foi iniciada com o intuito de elaborar um modelo de Computational Fluid Dynamics (CFD) utilizando a ferramenta de código MFsim, produzida pelo MFlab. Para tal, foram realizadas diversas simulações, variando-se a rotação da esfera e analisando-se o escoamento através de imagens extraídas da simulação. Para a obtenção dos dados, foi utilizada uma matriz de sondas posicionadas ao longo do domínio computacional. Posteriormente, foi desenvolvido um projeto de Machine Learning, onde houve a limpeza e organização dos dados para a alimentação do algoritmo de aprendizado de máquina utilizando a análise das compo-

nentes principais. Ao finalizar o treinamento do algoritmo, foi realizado um teste para avaliar o quão assertivo e representativo dos dados o modelo conseguia ser.

Figura 3 – Esquema da metodologia do trabalho



Fonte: Autoria própria

2. Revisão Bibliográfica

2.1 Escoamento sobre uma Esfera

A análise do escoamento sobre uma esfera é um estudo canônico na área de mecânica dos fluidos.

Começando com os estudos clássicos de Stokes, mais de um século atrás, diversos autores têm desenvolvido modelos para estudar o escoamento sobre uma esfera em rotação. Em 1851, Stokes derivou uma lei a partir da resolução das equações de Navier-Stokes, desenvolvendo um modelo matemático que permitia a análise de corpos esféricos imersos em fluidos em baixos números de Reynolds - regime laminar (STOKES et al., 1851). Esta lei forneceu à comunidade científica uma ferramenta extremamente útil para estudar o fluxo de fluidos, a interação entre fluidos e partículas sólidas e a influência de forças externas. A lei de Stokes foi essencial para o desenvolvimento de técnicas de simulação, bem como para o avanço de muitas áreas, como a hidrodinâmica, a aerodinâmica e a biomecânica.

Em 1956, Taneda investigou os regimes de escoamento de uma esfera submersa em um tanque de água, com um número de Reynolds entre 5 e 300. O experimento concluiu que a formação de vórtices teve início em um valor crítico de Reynolds de 24 (TANEDA, 1956). Além disso, Taneda apresentou uma relação proporcional entre o tamanho dos vórtices e o logaritmo do número de Reynolds (Figura 4).

Achenbach 1972 apresenta um experimento em altos valores de Reynolds, entre 5×10^4 e 6×10^6 Reynolds. Achenbach apresenta um gráfico relacionando o coeficiente de arrasto (medido por strain gauges) e o número de Reynolds. A curva obtida permite caracterizar 4 regimes turbulentos diferentes (Figura 5).

Figura 4 – Relação entre o tamanho de vórtices e o logaritmo do número de Reynolds

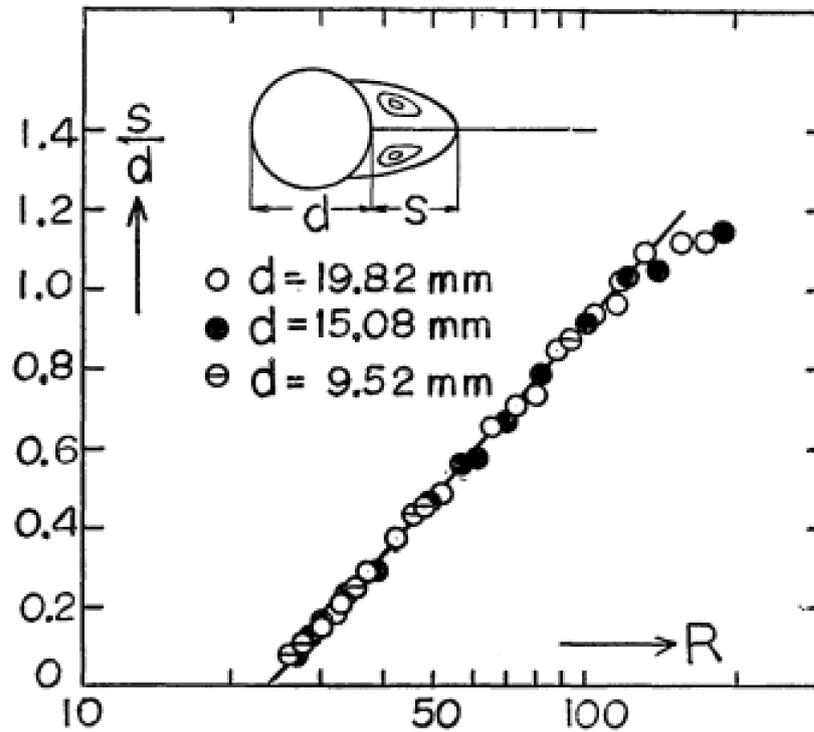
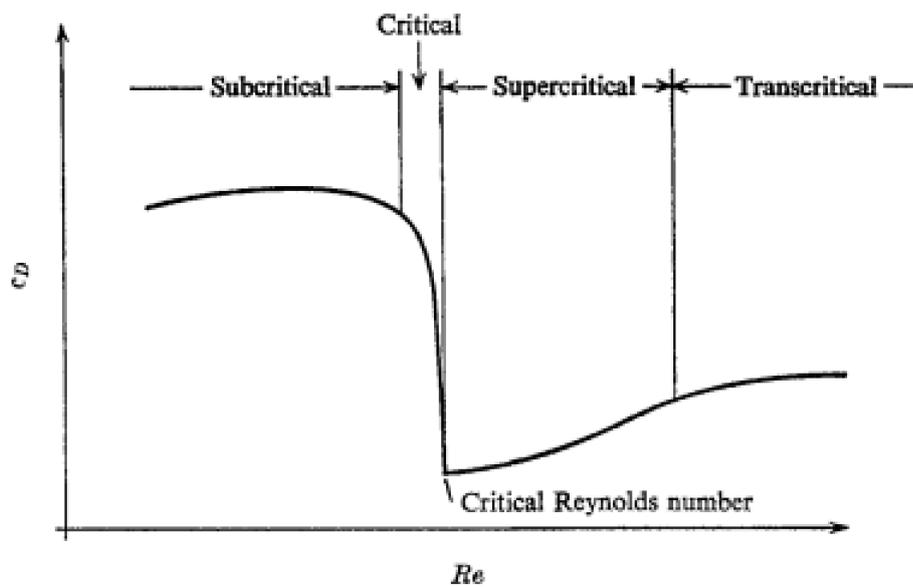


Figura 5 – Os quatro regimes de escoamento



Em um estudo sobre aplicação de CFD para simulação do escoamento sobre uma esfera estacionária, Jones e Clarke (2008) resumem os estudos sobre as faixas dos regimes de escoamento em função do número de Reynolds, apontando certa incerteza nos limites, já que pesquisas distintas apresentaram pequenas variações nas faixas para cada regime de escoamento. Jones e Clarke aplicam esses regimes de escoamento nos valores de Reynolds entre $Re = 20$ e $Re = 1.0 \times 10^6$.

Entre Reynolds $Re = 20$ e $Re = 210$ se encontra o regime axis-simétrico permanente. Escoamentos com número de Reynolds inferiores a 24 se apresentam perfeitamente laminares. Acima de $Re = 25$ existe um descolamento do fluxo em relação a esfera, produzindo duas recirculações contra rotativas simétricas em relação ao eixo central da esfera.

Entre $Re > 210$ e $Re < 270$ o escoamento permanece estável porém não mais axis-simétrico, desenvolve-se dois vórtices de cauda, apresentando ainda uma simetria no plano contendo os dois vórtices.

O número de Reynolds entre 270 e 400 resulta em uma transição da esteira planar-simétrica estável para uma esteira planar-simétrica dependente do tempo. A instabilidade primeiro aparece como uma ondulação na esteira de rosca dupla.

No intervalo de $400 < Re < 1000$, a simetria planar do fluxo é eventualmente perdida devido à mudança irregular da localização azimutal em que os loops de vórtice são formados de ciclo para ciclo. O valor exato do número de Reynolds em que isso começa a ocorrer é difícil de especificar com precisão.

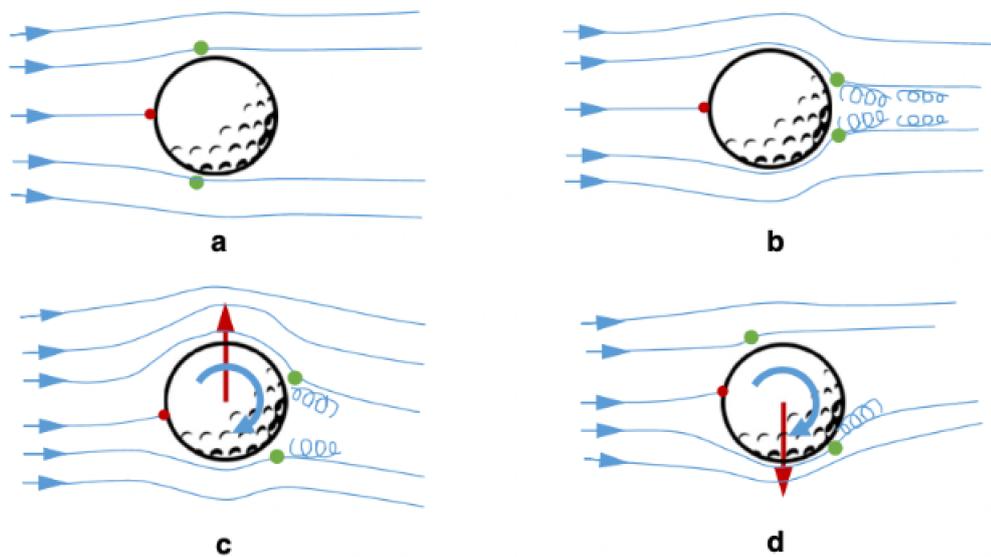
Quando $Re > 1000$, o regime é turbulento. Inicialmente, a emissão de vórtices é periódica, com ciclos regulares de vórtices sendo criados e destruídos. No entanto, ao longo do tempo, a periodicidade desta emissão começa a desaparecer e o escoamento torna-se mais irregular. Ao mesmo tempo, o número de vórtices aumenta, enquanto a amplitude dos vórtices diminui, dando origem a um regime turbulento.

2.2 Efeito Magnus

A análise de um objeto em rotação acarreta o surgimento de um fenômeno já observado e comentado há alguns séculos. Em 1672, Sir Isaac Newton fez o primeiro registro científico desse fenômeno no célebre artigo "New Theory of Light and Colours" (NEWTON, 2014).

O efeito Magnus recebeu o nome do famoso físico alemão Heinrich Gustav Magnus, que o descreveu pela primeira vez em 1853 (SEIFERT, 2012). Esta descoberta trata-se de um fenômeno físico no qual um objeto rotativo, como uma bola, desenvolve uma força perpendicular à direção de seu movimento de rotação. Esta força tem um impacto significativo na trajetória do objeto, desviando-o do caminho previsto. O efeito Magnus é algo muito explorado nos esportes com bola, como beisebol, futebol e golfe, onde os jogadores e atletas precisam prever o desvio das bolas em relação a sua trajetória original.

Figura 6 – Efeito Magnus e Magnus inverso em um bola de golfe



Retirado de (LYU; KENSRUD; SMITH, 2020)

Lyu et al 2020 explicaram o Efeito Magnus em bolas de golf comerciais na camada limite transicional. O fenômeno ocorre devido à transição da camada limite laminar para turbulenta. Em uma esfera sem rotação, a camada limite é laminar e os pontos de descolamento da camada limite ocorrem mais cedo, pois camadas laminares têm maior tendência ao descolamento devido à baixa energia na superfície da esfera. Quando a velocidade do escoamento aumenta, turbulência é gerada na camada limite, retardando o descolamento do escoamento e reduzindo a largura da esteira.

Em esferas com rotação no sentido horário e escoamento da esquerda para a direita, o escoamento na parte superior está favorável ao sentido de rotação, o que oferece maior velocidade em relação à parte inferior, que está contrária ao sentido de rotação. Com isso, o escoamento na parte superior tem seu ponto de descolamento atrasado e o escoamento inferior apresenta um ponto de descolamento mais próximo do ponto de estagnação. O resultado dessa modificação no ponto de estagnação é o surgimento de um gradiente de pressão vertical, o que resulta na Força Magnus. Esse fenômeno apresentou mudança no sentido da Força Magnus em determinadas condições. Quando o escoamento está em transição, de laminar para turbulento, uma rotação no sentido horário pode provocar uma camada limite laminar na parte superior e uma turbulenta na parte inferior, invertendo o Efeito Magnus - o Efeito Magnus Inverso.

Lyu et al observaram o Efeito Magnus Inverso nas bolas de golf em velocidades onde há crise de arrasto, Reynolds entre $5 \times 10^4 < Re < 7 \times 10^4$ e rotações entre $750 < \omega < 2250$ rpm.

2.3 Meta-modelagem e Machine learning

As simulações computacionais permitem avaliar cenários complexos que não são viáveis de produzir de forma analítica (XU et al., 2016). Elas costumam consumir menos recursos do que os experimentos físicos. A maior valia dessas simulações está atrelada ao custo computacional provocado (CHABANET; EL-HAOUZI; THOMAS, 2021) e a capacidade de produzir resultados cada vez mais satisfatórios e representativos da realidade, para isso deve-se sempre procurar a otimização da simulação. Para o efeito da otimização, em cenários complexos, muitas simulações necessitam ser simplificadas, logo, aproximações mais simples são feitas, o que acarreta a elaboração de um modelo de um modelo - o metamodelo (BARTON; MECKESHEIMER, 2006). As técnicas de machine learning desenvolvidas nas últimas décadas permitem agregar a meta-modelagem a capacidade de ingerir grandes quantidades de dados e extrair "insights" que outrora não era possível.

3. Referencial Teórico

3.1 Modelagem Matemática

Segundo Neto 2020 todo modelo matemático deve ser baseado nos princípios fundamentais, que incluem o balanço de massa, o balanço de quantidade de movimento linear (ou Segunda Lei de Newton) e o balanço de energia. Esses princípios fazem parte das leis da Mecânica Clássica Newtoniana, que foram estabelecidas para sistemas fechados, ou seja, aqueles que não perdem nem ganham massa durante qualquer processo físico.

As equações da continuidade 3.1 e de Navier-Stokes 3.2 (abaixo de Mach=15) são usadas para modelar o comportamento da turbulência. É importante entender como essas equações funcionam e como elas são aplicadas para analisar sistemas turbulentos. A continuidade é uma equação fundamental que descreve a conservação de massa em um fluido. A equação de Navier-Stokes é uma equação de movimento que descreve a conservação da quantidade de movimento em um fluido. Ao combinar essas equações, podemos obter a modelagem de um escoamento em diferentes circunstâncias, quer em regime permanente, quanto em transição e em regime turbulento. Considera-se um fluido newtoniano de viscosidade cinemática (ν) em um escoamento incompressível de massa específica (ρ) e um meio isotérmico.

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (3.1)$$

$$\underbrace{\frac{\partial (\rho u_i)}{\partial t}}_{\text{Termo Temporal}} + \underbrace{\frac{\partial (\rho u_j u_i)}{\partial x_j}}_{\text{Termo Advectivo}} = -\frac{\partial p}{\partial x_i} + \underbrace{\frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right]}_{\text{Termo Difusivo}} + \underbrace{f_i}_{\text{Termo Fonte}} \quad (3.2)$$

É importante ressaltar que a turbulência é um fenômeno complexo e ainda há muito a ser descoberto e compreendido sobre ele, pois continua sendo um dos maiores problema da física clássica sem solução definitiva/completa (NETO, 2020).

3.2 Fronteira Imersa

A utilização de malhas produzidas numericamente para simulações de CFD podem ser classificadas em duas classes. O uso de malhas estruturadas, que são conformadas às geometrias imersas no escoamento, é o método mais utilizado devido à sua robustez. No entanto, esse método demanda um esforço computacional considerável e um conhecimento maior para construir uma malha satisfatória, já que o modelo tende a perder qualidade à medida que a geometria estudada se torna mais irregular. O outro método consiste na utilização de malhas cartesianas

independentes da complexidade das geometrias envolvidas. A presença da geometria é considerada através de termos fontes pontuais adicionados à equação de modelagem do balanço de quantidade de movimento linear do sistema. Na simulação fluido-estrutura, o método de fronteira imersa é necessário para impor condições de contorno no subsistema do fluido na interface entre o fluido e a estrutura, e para determinar as forças fluidodinâmicas que atuam sobre a estrutura. O domínio do fluido é discretizado em uma malha cartesiana uniforme, enquanto a superfície da estrutura é discretizada com pontos lagrangeanos. Esses pontos se movimentam dentro do domínio do fluido. A velocidade do fluido nos pontos lagrangeanos (interface entre o fluido e a estrutura) deve ser igual à velocidade dos pontos materiais que pertencem à estrutura. Tanto a velocidade quanto a posição desses pontos lagrangeanos são variáveis no tempo. Portanto, é preciso utilizar um método que localize os pontos lagrangeanos no domínio do fluido e imponha, no fluido, a velocidade do ponto. O método da fronteira imersa, proposto por Wang, Fan e Luo 2008, se coloca nesse contexto. Para levar em consideração a presença da fronteira imersa, é preciso que se adicione um termo fonte $f_i [N m^{-3}]$ à equação de Navier-Stokes filtrada, da seguinte forma:

$$\underbrace{\frac{\partial \bar{u}_i}{\partial t}}_{\text{Termo Temporal}} + \underbrace{\frac{\partial (\bar{u}_i \bar{u}_j)}{\partial x_j}}_{\text{Termo Advectivo}} = -\frac{1}{\rho_f} \frac{\partial \bar{p}}{\partial x_i} + \underbrace{\frac{\partial}{\partial x_j} \left[\nu_{ef} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \right]}_{\text{Termo Difusivo}} + \underbrace{\frac{f_i}{\rho_f}}_{\text{Termo fonte}}, \quad (3.3)$$

onde:

$$f_i(\vec{x}) = \int_{\Omega} (F_k)_i(\vec{x}_k) \delta(\vec{x} - \vec{x}_k) d\vec{x}_k, \quad (3.4)$$

sendo $\delta(\vec{x} - \vec{x}_k)$ é a função delta de Dirac, \vec{x}_k é a posição do ponto lagrangeano e \vec{x} é a posição do ponto euleriano em análise.

O termo f_i o é elo para a comunicação entre as equações de Navier-Stokes e as equações para o movimento da interface. Em outras palavras, esse termo é responsável por permitir que o escoamento reconheça a presença da interface, o que resulta no surgimento de escoamentos coerentes internos e externos a ela. É importante destacar que o termo fonte de força pode ser diferente de zero sobre a interface, mas é sempre nulo fora dela (VILLAR et al., 2007; BARBOSA, 2023).

3.3 PCA e XGBoost

O processo de análise de dados requer uma série de cuidados para garantir a validade dos resultados obtidos. Esse processo é especialmente desafiador quando se lida com conjuntos de dados complexos, como os apresentados neste trabalho, que exigem a redução da alta dimensionalidade para que possam ser processados por algoritmos de aprendizado de máquina.

Para lidar com essa complexidade, o PCA (Principal Component Analysis) é uma técnica matemática de análise multivariada amplamente utilizada para reduzir a complexidade do conjunto de dados, representando-o em um espaço de dimensão menor. O método consiste em

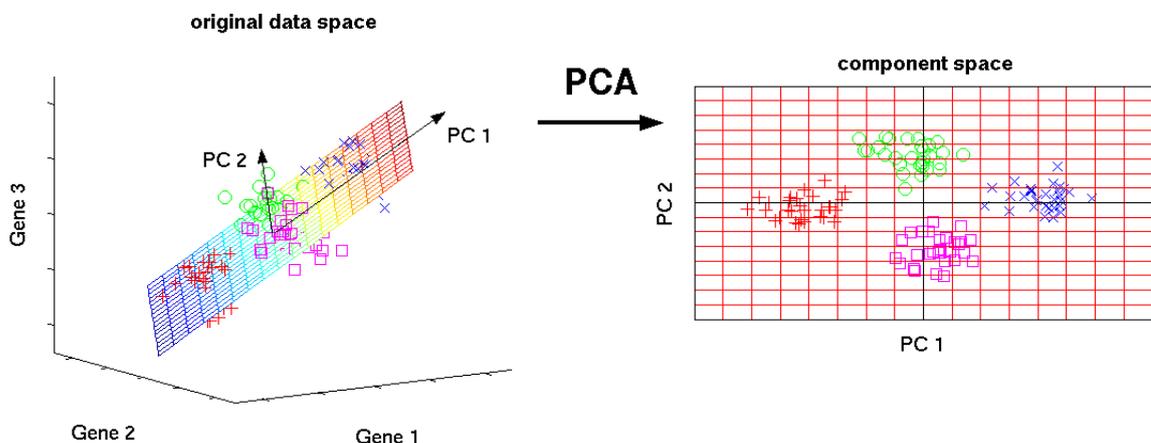
encontrar as direções ou componentes principais responsáveis por explicar a maior parte da variação nos dados. Essas direções são calculadas a partir da matriz de covariância dos dados originais e representam combinações lineares dos recursos originais (SCHOLZ, 2006).

As direções principais são ordenadas em termos de sua importância em explicar a variação nos dados, com a primeira direção principal representando a maior parte da variação. Uma vez que as direções principais são encontradas, a PCA pode ser usada para reduzir a dimensão dos dados originais, mantendo a maior parte da variação. Isso é feito projetando os dados originais nas direções principais, o que resulta em um conjunto de dados transformado com menos dimensões.

No entanto, para garantir a validade dos resultados obtidos através da PCA é fundamental normalizar os dados para que todos os recursos tenham a mesma escala e possam ser comparados adequadamente.

Os novos recursos gerados pela PCA são chamados de componentes principais. Com a redução da dimensionalidade do conjunto de dados, é possível melhorar a eficiência computacional dos algoritmos de aprendizado de máquina e, ao mesmo tempo, preservar a maior parte das informações relevantes nos dados originais. Por isso, é uma técnica importante para a análise de grandes conjuntos de dados e um conhecimento valioso para aqueles que trabalham com Aprendizado de Máquina e Análise de Dados.

Figura 7 – Exemplo de redução de dimensionalidade feito com PCA em um conjunto de dados com 3 variáveis (genes).



Retirado de Scholz (SCHOLZ, 2006)

Graças ao avanço da ciência de dados, existem uma grande variedade de algoritmos de machine learning a disposição. O algoritmo escolhido neste trabalho foi XGBoost. XGBoost (Extreme Gradient Boosting) é um algoritmo de aprendizado de máquina de aumento de gradiente que é usado para classificação e regressão em conjuntos de dados estruturados. O algoritmo foi criado por Tianqi Chen e Carlos Guestrin em 2016. O XGBoost funciona construindo uma sequência de árvores de decisão que são treinadas de forma iterativa para corrigir os erros de

previsão em cada iteração. Além disso, o XGBoost usa uma técnica chamada de boosting (FRIEDMAN, 2001) para combinar várias árvores de decisão em uma sequência de modelos. Cada árvore é adicionada à sequência com um peso proporcional ao seu desempenho no conjunto de dados de treinamento. Essa técnica ajuda a melhorar a precisão da previsão e a reduzir o viés do modelo.

Para treinar o modelo, precisamos definir a função objetivo para medir o quão bem o modelo se ajusta aos dados de treinamento. A função objetivo está definida na equação 3.5

$$obj = \sum_{i=1}^n L(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (3.5)$$

Onde obj é a função objetivo a ser minimizada, n é o número de exemplos nos dados de treinamento, $L(y_i, \hat{y}_i)$ é a função de perda que mede a diferença entre as previsões do modelo \hat{y}_i e os valores observados y_i para o i -ésimo exemplo. K é o número de árvores de decisão no conjunto de modelos f , $\Omega(f_k)$ é o termo de regularização que penaliza modelos mais complexos para a k -ésima árvore f_k .

3.3.1 Hiperparâmetros

A otimização dos hiperparâmetros do modelo XGBoost é crucial para alcançar um modelo de alta precisão e evitar problemas como overfitting ou underfitting.

Os hiperparâmetros controlam como o algoritmo de aprendizado é executado e como o modelo é construído. Alterar um hiperparâmetro pode ter um grande impacto no desempenho do modelo, por isso é importante encontrar a combinação ideal de hiperparâmetros que permita que o modelo atinja o melhor desempenho possível.

Por exemplo, se o número de árvores ($n_estimators$) for muito baixo, o modelo pode ser subtreinado e ter uma precisão ruim. Por outro lado, se o número de árvores for muito alto, o modelo pode estar sujeito a overfitting e não generalizar bem para novos dados.

O mesmo acontece com outros hiperparâmetros, como a taxa de aprendizado (eta), a profundidade máxima das árvores (max_depth), a quantidade mínima de peso que cada folha da árvore deve ter (min_child_weight), e a fração de amostras selecionadas aleatoriamente para cada árvore durante o treinamento ($subsample$).

A otimização de hiperparâmetros pode ser realizada por meio de tentativa e erro, mas isso pode ser demorado e muitas vezes ineficaz. Em vez disso, é possível usar técnicas mais avançadas, como busca aleatória, busca em grade ou otimização bayesiana, para encontrar a combinação ideal de hiperparâmetros.

Em resumo, otimizar os hiperparâmetros do XGBoost é importante para maximizar o desempenho do modelo e garantir que ele generalize bem para novos dados:

1. $n_estimators$ - Este hiperparâmetro define o número de árvores de decisão que serão criadas pelo modelo XGBoost. Um número maior de árvores pode aumentar a precisão

do modelo, mas também pode aumentar o tempo de treinamento e o risco de overfitting. É importante encontrar um equilíbrio entre a precisão e a eficiência do modelo.

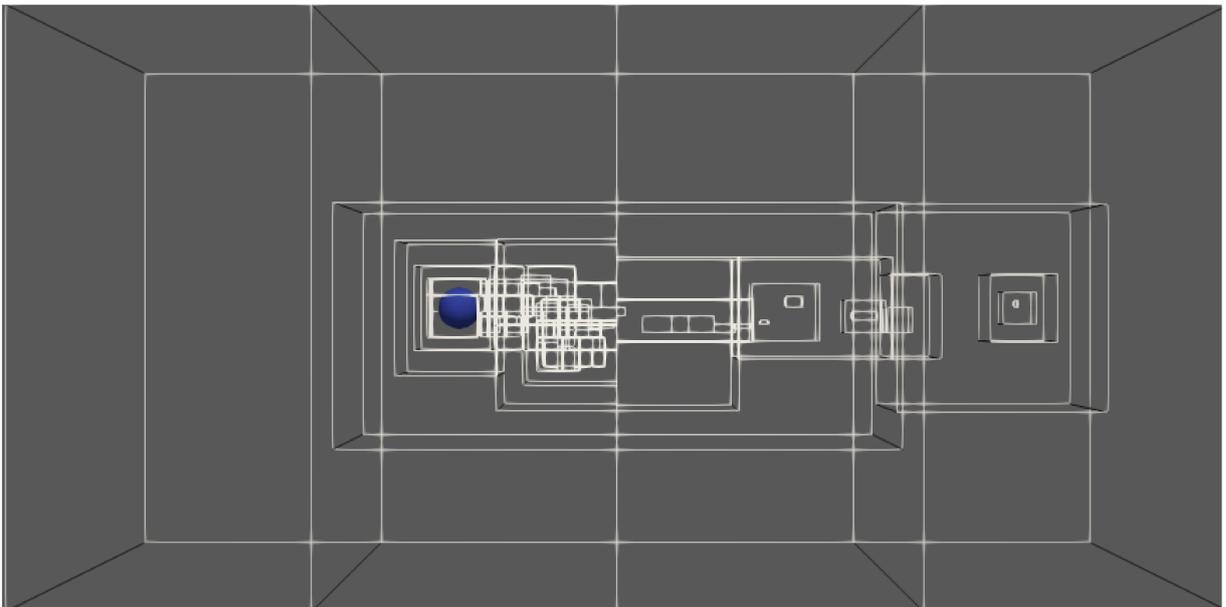
2. *eta* - Também conhecido como taxa de aprendizado (learning rate), este hiperparâmetro controla a magnitude da atualização que é feita em cada iteração do algoritmo de treinamento. Uma taxa de aprendizado menor pode levar a uma convergência mais lenta, mas também pode ajudar a evitar overfitting. Por outro lado, uma taxa de aprendizado maior pode acelerar a convergência, mas também aumenta o risco de overfitting.
3. *max_depth* - Este hiperparâmetro controla a profundidade máxima das árvores de decisão no modelo. Uma profundidade maior pode permitir que o modelo capture interações mais complexas entre as variáveis, mas também pode levar a overfitting e aumentar o tempo de treinamento. Novamente, é importante encontrar um equilíbrio entre a precisão e a eficiência do modelo.
4. *min_child_weight* - Este hiperparâmetro controla a quantidade mínima de peso que cada folha da árvore deve ter. Um valor maior para este parâmetro pode ajudar a evitar overfitting, já que cada folha terá que ter pelo menos uma quantidade mínima de amostras para ser considerada. No entanto, um valor muito alto pode levar a um modelo subtreinado. Em tarefas de regressão *min_child_weight* corresponde a quantidade mínima de amostras em cada folha.
5. *subsample* - Este hiperparâmetro controla a fração de amostras que são selecionadas aleatoriamente para cada árvore durante o treinamento. Um valor menor para este parâmetro pode ajudar a evitar overfitting, já que cada árvore terá que aprender a partir de um subconjunto menor de amostras. No entanto, um valor muito baixo pode levar a um modelo subtreinado.

4. Parâmetros de simulação

4.1 Malha

A simulação foi feita utilizando uma malha adaptativa, devido à necessidade de diminuir o tempo necessário para a solução numérica de problemas envolvendo mecânica dos fluidos. Surge, portanto, a necessidade de utilizar um refinamento localizado, a fim de obter uma malha refinada apenas na região de interesse. O refinamento adaptativo envolve a adaptação da malha e a otimização em pontos que exigem maior refinamento tais como zonas de instabilidades no escoamento, turbulências intensas e proximidades das paredes, economizando assim o poder computacional em áreas menos necessárias. Para o refinamento localizado, é empregada a técnica de Refinamento de Malha Adaptativa (Adaptive Mesh Refinement - AMR), na qual uma malha composta é definida como uma sequência hierárquica de níveis aninhados, progressivamente mais finos, como conceituado por Roma et al (1999).

Figura 8 – Refinamento Localizado



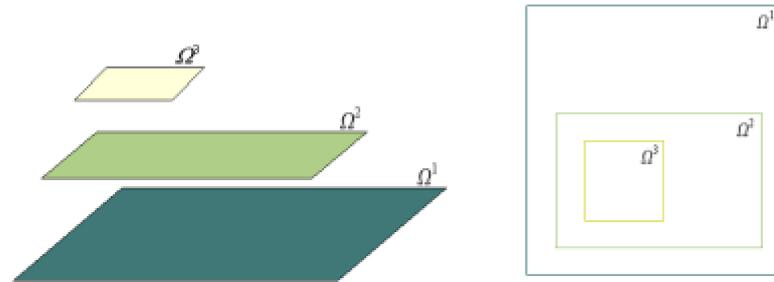
Fonte: Autoria Própria

Foram utilizados métodos Multigrid para a resolução dos sistemas lineares envolvidos na solução numérica. A variante Multinível-Multigrid possibilita o manuseio de malhas refinadas localmente, para a qual deve-se definir o nível de refinamento mais fino e o nível base.

As equações de Navier-Stokes são resolvidas utilizando uma malha bloco-estruturada com refinamento local. A definição principal de malha está relacionada ao nível mais grosso, cha-

mado de LBOT. O domínio é então discretizado com a malha escolhida e, caso se deseje refinar determinadas regiões do domínio, malhas mais finas são sobrepostas a essa malha grosseira. O nível mais fino da malha é chamado de LTOP.

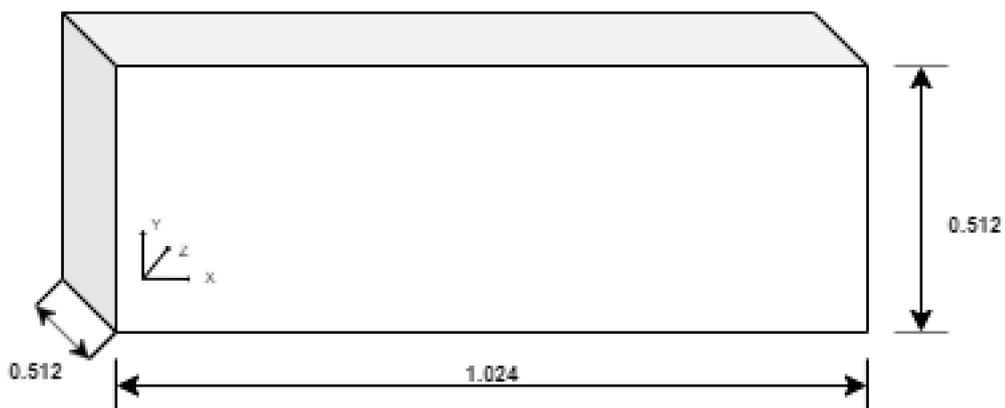
Figura 9 – Exemplo de malha composta por três níveis de refinamento.



Fonte: Villar (2007)

O domínio do problema é um paralelepípedo retangular definido por $[0; 1.024] \times [0; 0.512] \times [0; 0.512]$. Dentro deste domínio, há uma esfera com centro em $(0,36, 0,256, 0,256)$ e diâmetro de 0,04 m. São definidos 5 níveis físicos, sendo que o nível LBOT é $32 \times 16 \times 16$. Além disso, é adotada uma malha adaptativa e são usados dois critérios de refinamento: fronteira imersa (ib-points) e vorticidade.

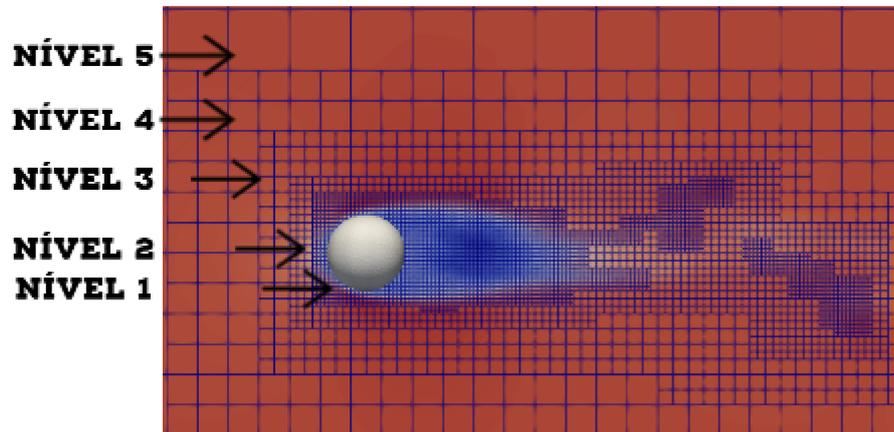
Figura 10 – Domínio da Simulação



Fonte: Autoria Própria

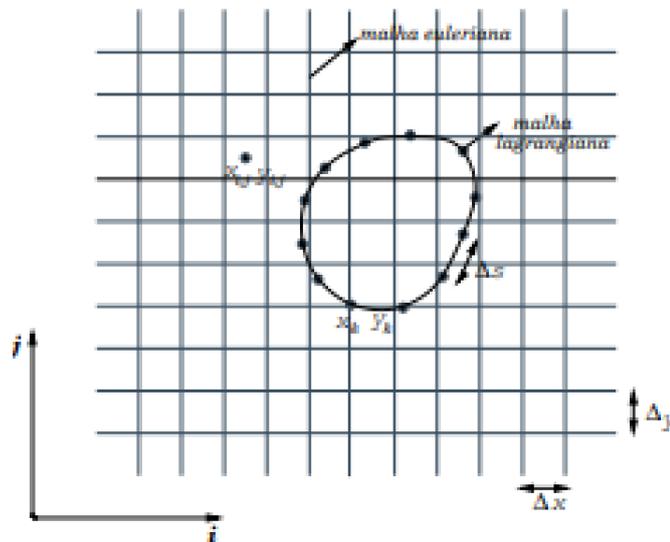
A aproximação numérica das equações de Navier-Stokes é realizada utilizando um método de discretização do tipo Malha euleriana, ao passo que as equações que descrevem o movimento da interface são discretizadas utilizando uma malha Lagrangiana móvel, independente da malha euleriana.

Figura 11 – Níveis físicos



Fonte: Autoria Própria

Figura 12 – Representação esquemática da relação entre a malha euleriana e a malha lagrangiana



Fonte: (VILLAR et al., 2007)

4.1.1 Malha euleriana

A malha euleriana descreve o domínio a ser estudado contemplando os 5 níveis físicos propostos. Para se obter o número de células no nível mais fino (n_{ltop}) deve-se calcular:

$$n_{ltop} = n_{lbot} * 2^{(n_{pl}-1)} \quad (4.1)$$

Onde n_{lbot} é o número de células no nível base e n_{pl} é o número de níveis físicos desejados. Sendo $LBOT$ igual a $32 \times 16 \times 16$, temos que $LTOP$ é $512 \times 256 \times 256$, $dx = dy = dz$ é o espaçamento da malha euleriana que é igual em todas as direções. O volume do cubo do nível

mais fino da malha euleriana é dado por

$$dx^3 = \left(\frac{1.024}{512}\right)^3 = 8 * 10^{-9} \quad (4.2)$$

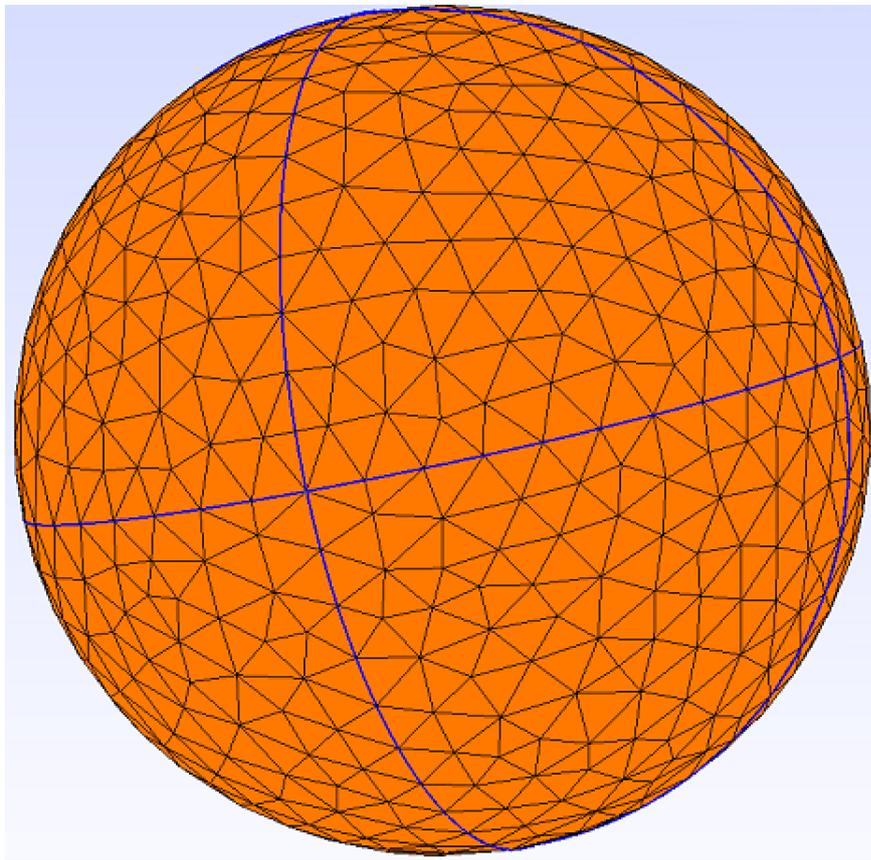
4.1.2 Malha lagrangeana

A malha lagrangiana é a malha que limita os contornos da fronteira imersa. Por meio do software Gmsh, o arquivo da geometria no formato .stl é gerado. O volume do prisma de base triangular da malha que cobre a esfera é dado por:

$$\frac{l^3\sqrt{3}}{4} = dx^3 \quad (4.3)$$

A partir da equação 4.2 obtemos que $l = 0.0026$. A malha lagrangiana construída pode ser visualizada a seguir, onde o tamanho das arestas dos triângulos é 0,0026.

Figura 13 – Malha da esfera



Fonte: Autoria própria

4.2 Parâmetros Físicos

As propriedades físicas necessárias para desenvolver a simulação estão atreladas ao número de Reynolds,

O número de Reynolds é definido como:

$$\text{Re} = \frac{\rho u L}{\mu} \quad (4.4)$$

Onde Re é o número de Reynolds, ρ é a densidade do fluido, u é a velocidade característica do fluido, L é a dimensão característica do sistema e μ é a viscosidade dinâmica do fluido. O número de Reynolds foi setado a priori no valor de 1000. Foi considerado um ρ de 2,5, o comprimento característico da esfera consiste no seu diâmetro $L=0,04$. A viscosidade μ apresenta o valor de 0,0001. Os dados forma sumarizados na tabela 1. O tempo físico total da simulação foi de 1 segundo.

Tabela 1 – Pârametros Físicos

Parâmetro	Valor
Re	1000
ρ	2,5
u	1
L	0,04
μ	0,001

4.3 Parâmetros de controle da simulação

O número de Reynolds selecionado para essa simulação permite que a equação de Navier-Stokes seja resolvida completamente, já que foi observado, durante a simulação, que os efeitos da turbulência são desprezíveis nessa faixa. O tempo físico escolhido foi de 1 segundo. O termo temporal é discretizado utilizando o método semi-implícito sem truque. O Termo advectivo foi discretizado pelo esquema CUBISTA (Convergent and Universally Bounded Interpolation Scheme for the Treatment of Avection).

4.4 Condições de contorno

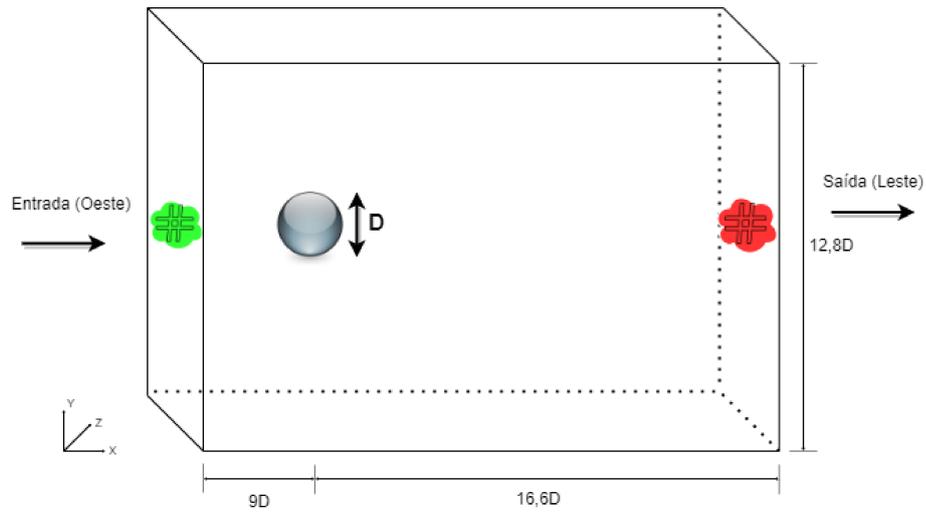
Dado que o fluido flui na direção x, podemos identificar a face oeste como a entrada. Foi adotada uma condição de Dirichlet para impor que a velocidade na entrada é igual ao valor unitário.

Na face leste, que corresponde à saída, foi aplicada uma condição advectiva. Esta condição é capaz de lidar com recirculações que ocorrem nas imediações da fronteira e é frequentemente empregada na face de saída.

Em relação às outras faces, a condição de Neumann foi definida para a velocidade.

Em relação à pressão, a condição de Neumann foi estabelecida para a face oeste, enquanto que nas demais faces foi aplicada uma condição de Dirichlet com valor zero. Para as propriedades físicas, foi adotada uma condição de contorno com derivada nula, tendo em vista que o valor dessas propriedades varia em função do número de Reynolds.

Figura 14 – Domínio.

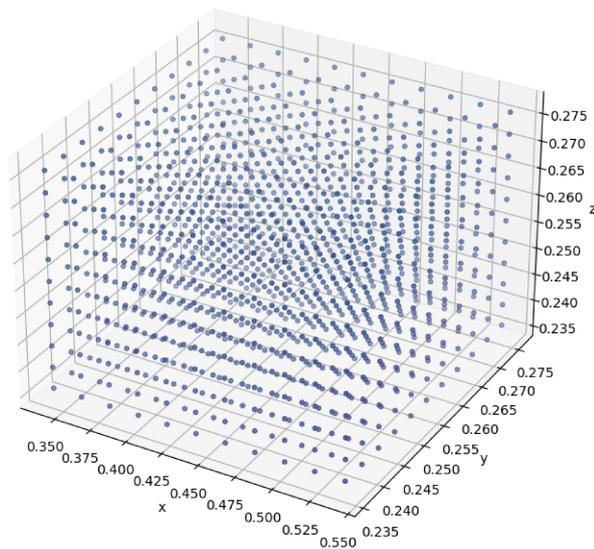


Fonte: Autoria Própria

4.5 Sondas

Para a obtenção dos dados utilizados na visualização do escoamento sob a esfera e para aplicação no algoritmo de aprendizado de máquina, foi implementado um conjunto de sondas na simulação. O número de sondas utilizado foi de 1331 ao longo das três dimensões do domínio x , y e z . Cada sonda obtém o tempo e os dados da velocidade nos três eixos, u , v e w . A escolha por trabalhar com um quantidade elevada de sondas se dá devido a necessidade de um conjunto de dados expressivo para a melhor efetividade do processo de predição do algoritmo de ML.

Figura 15 – Representação da malha de sondas produzida para obter os dados da simulação



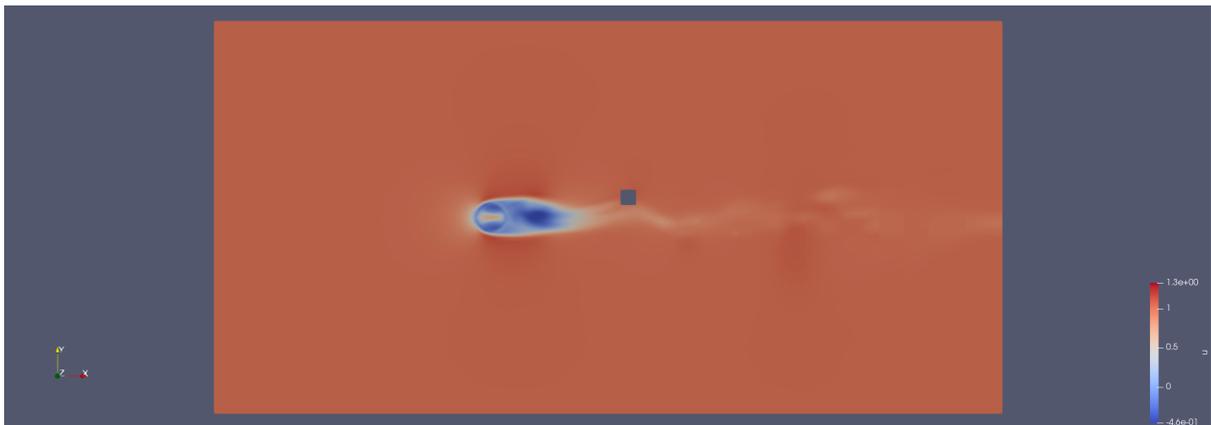
Fonte: Autoria Própria

5. Análise de Resultados

5.1 Análise do Escoamento

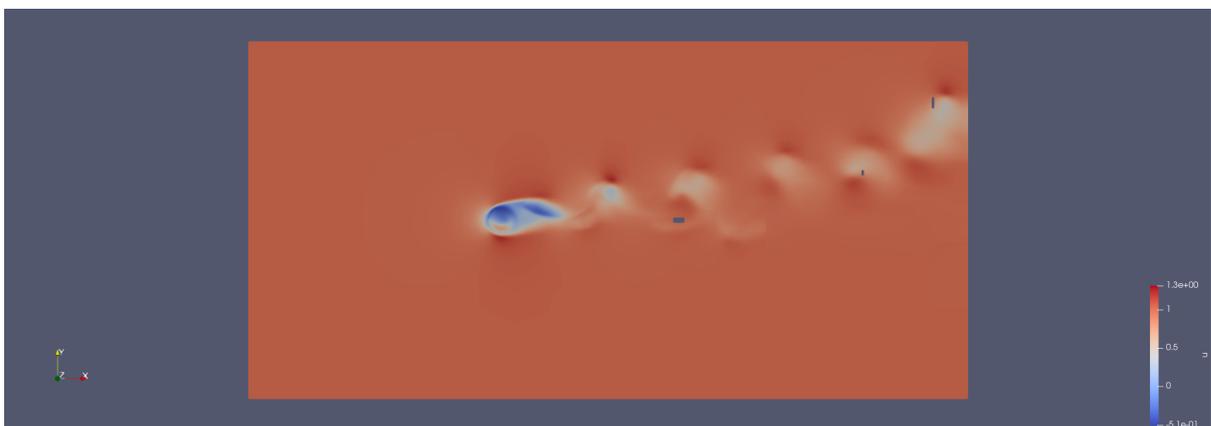
Várias simulações foram realizadas onde se varia a velocidade de rotação da esfera (ω). As velocidades foram selecionadas em um intervalo entre $\omega = 25 \text{ rad/s}$ no sentido horário e $\omega = -25 \text{ rad/s}$, sentido anti-horário. As figuras obtidas mostram o efeito da velocidade de rotação no escoamento. Observa-se que o escoamento é modificado consoante a velocidade de rotação, obtendo uma trajetória diagonal, tendo como parâmetro a situação onde a velocidade angular ω é nula. A amplitude do ângulo do escoamento bem como a produção de vórtices depende da intensidade da velocidade angular.

Figura 16 – Escoamento $\omega = 0 \text{ rad/s}$

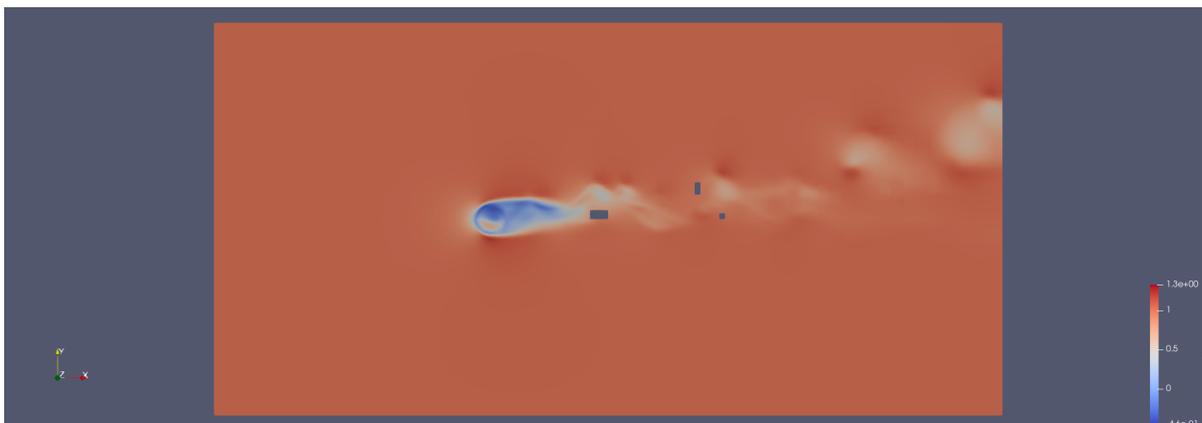


Fonte: Autoria própria

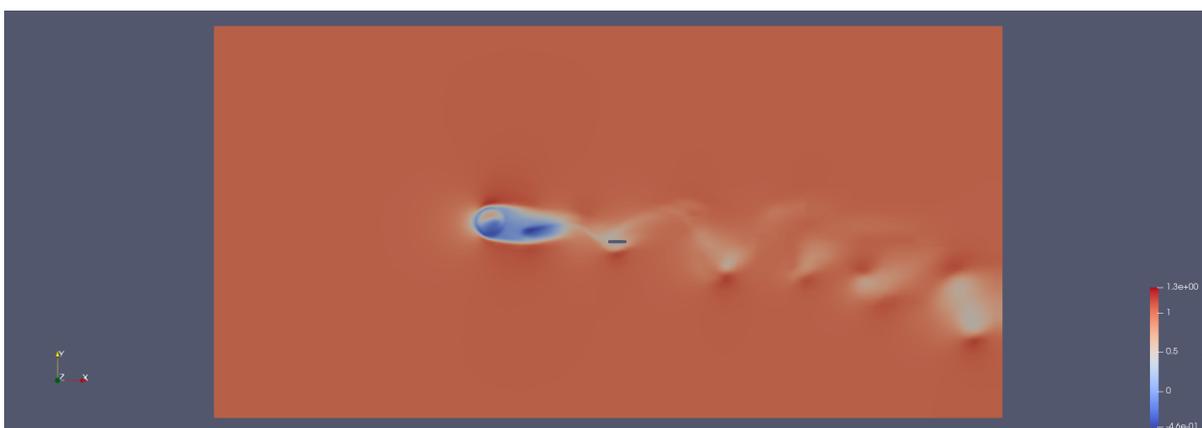
Figura 18 – Escoamento $\omega = -20 \text{ rad/s}$



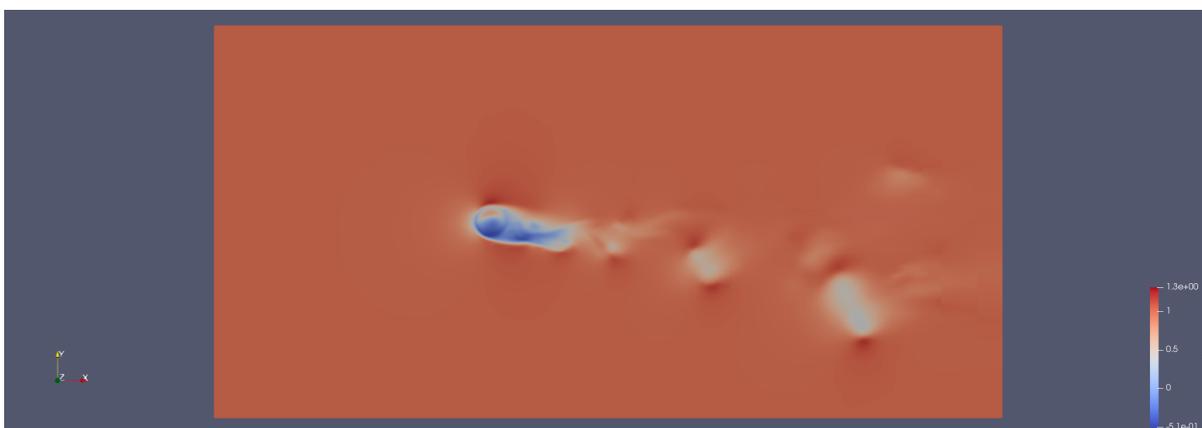
Fonte: Autoria própria

Figura 17 – escoamento $\omega = -10 \text{ rad/s}$ 

Fonte: Autoria própria

Figura 19 – escoamento $\omega = 10 \text{ rad/s}$ 

Fonte: Autoria própria

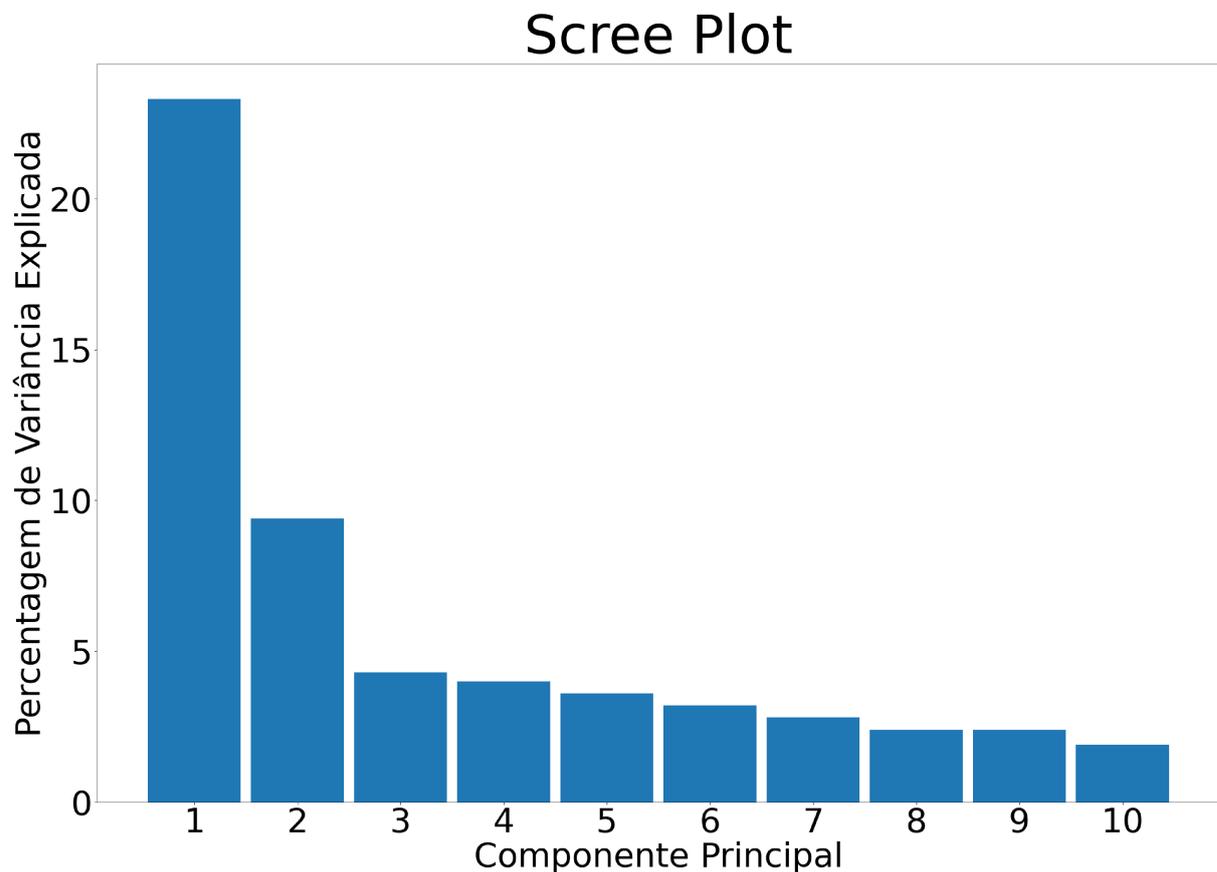
Figura 20 – escoamento $\omega = 20 \text{ rad/s}$ 

Fonte: Autoria própria

5.2 Machine Learning

Para o processo de construção de um modelo de aprendizado de máquina foi utilizado os dados das simulações onde $\omega = \{-25; -20; -15; -12; -10; -5; 0; 5; 10; 15; 25\}$ (rad/s), para o teste foi utilizado a rotação de $\omega = 12 rad/s$ para previsão. A extração dos dados da simulação através das sondas obteve, inicialmente, um conjunto de dados com 17988 linhas e 3995 colunas. As variáveis de interesse foram as velocidades u, v e z . Os dados foram tratados e normalizados, implementando-se a seguir uma análise de componentes principais. O propósito do *PCA* consistia em reduzir a quantidade de variáveis, ou dimensões, a serem implementadas no algoritmo de aprendizado de máquina. Com a análise de componentes principais foi possível reduzir as 3993 variáveis em uma combinação de 55 componentes principais. O gráfico 21 apresenta as primeiras 10 componentes principais.

Figura 21 – Gráfico porcentagem de variância explicada por Componente Principal



Fonte: Autoria própria

A Primeira componente principal representa 23,3% da variação dos dados, enquanto a segunda componente principal representa 9,4% da variação do conjunto de dados original. O restante das componentes principais não ultrapassam o valor de 5%. Todas as componentes principais foram utilizadas no treinamento do algoritmo.

O método utilizado para analisar o erro do algoritmo foi a raiz quadrada do erro médio quadrado. A raiz quadrada do erro médio quadrado (RMSE) pode ser escrita como:

$$\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (5.1)$$

Onde y_i é o valor observado, \hat{y}_i é o valor previsto pelo modelo e n é o número total de observações.

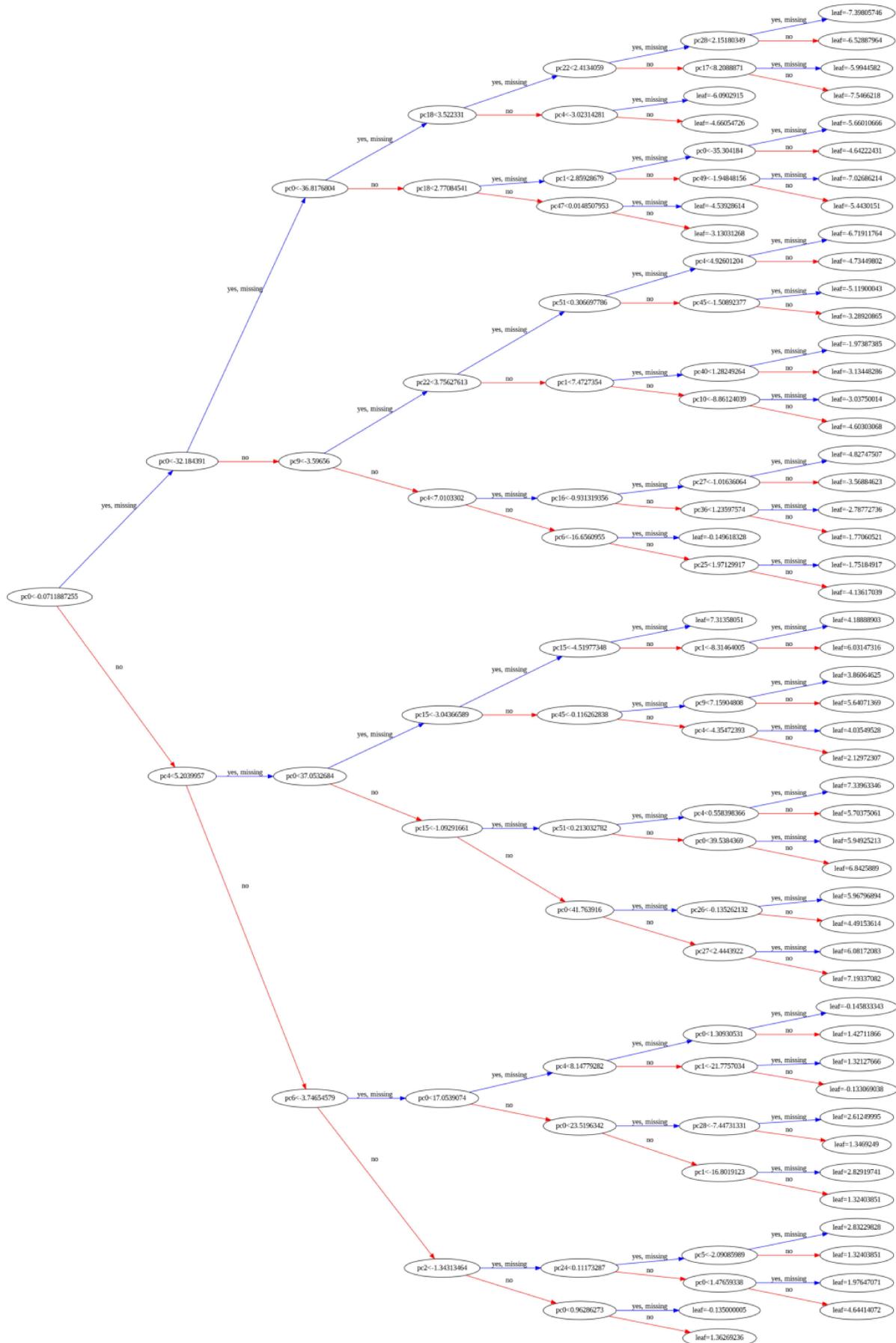
Inicialmente o algoritmo foi implementado utilizando os valores pré definidos do XGBoost, com 100 árvores de decisão. Os valores padrões do XGBoost das variáveis escolhidas para otimização estão apresentadas na tabela 2. Nesta primeira interação o RMSE foi igual a 2.29

Tabela 2 – Hiperparâmetros default

Hiperparâmetro	Valor
<i>n_estimators</i>	100
<i>eta</i>	0,3
<i>max_depth</i>	6
<i>min_child_weight</i>	1
<i>subsample</i>	1

A Primeira árvore de decisão foi plotada, demonstrando o processo de decisão do algoritmo. Observa-se que a árvore possui várias camadas, desenvolvendo um processo de decisão complexo, que se torna suscetível a vieses, podendo provocar o overfitting e a não generalização da previsão feita.

Figura 22 – Primeira árvore de decisão com hiperparâmetros default (Autoria Própria)



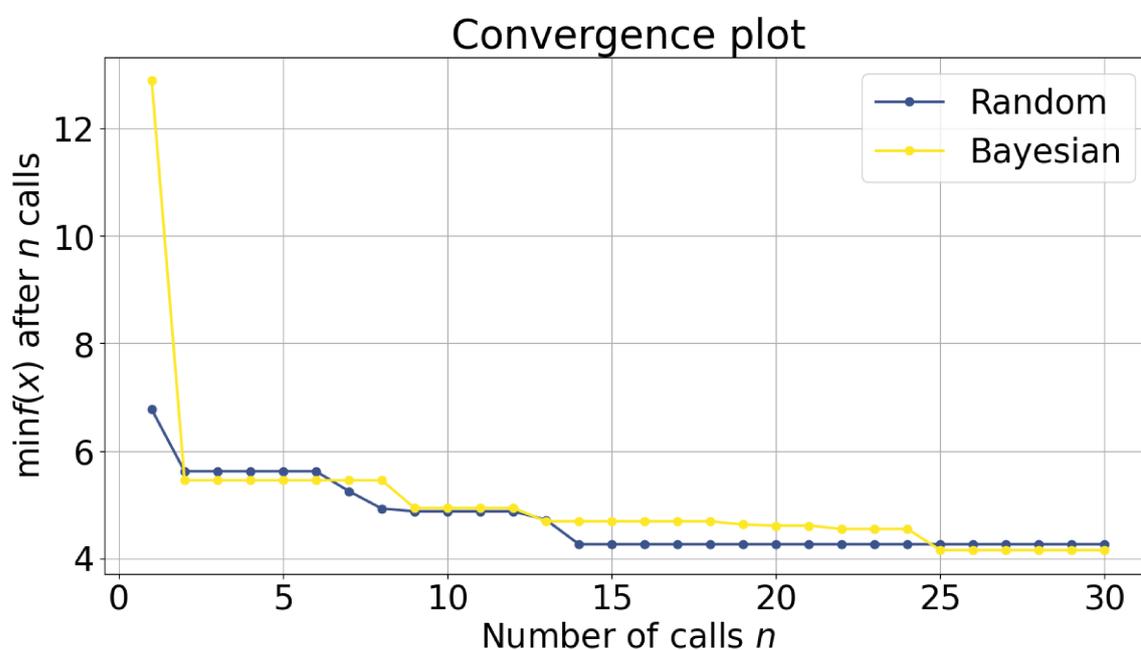
Para Otimizar os hiperparâmetros foi proposta a comparação entre dois métodos: Busca aleatória e Otimização Bayesiana. Em ambos os casos é dado um intervalo de valores na qual certo hiperparâmetro deve ser definido. A busca aleatória procura alterar esses valores de forma randômica a cada interação no intuito de encontrar o menor erro possível. Na otimização Bayesiana, o processo também inicia-se com uma busca aleatória, porém, após algumas interações, o método absorve as informações das interações anteriores para alimentar o seu modelo probabilístico, com o intuito de prover previsões cada vez mais precisas. A tabela 3 indica os limites propostos para cada hiperparâmetro otimizado.

Tabela 3 – Limites inferiores e superiores da otimização

Hiperparâmetro	Limite inferior	Limite superior
<i>n_estimators</i>	80	1000
<i>eta</i>	0,01	0.5
<i>max_depth</i>	2	15
<i>min_child_weight</i>	2	20
<i>subsample</i>	0.5	1

Ambos os modelos foram definidos com um limite de 30 interações. A comparação entre os modelos encontra-se na figura 23, onde se compara o erro médio quadrado ao longo das interações.

Figura 23 – Gráfico comparação entre Busca aleatória (Random) e Otimização Bayesiana (Bayesian)



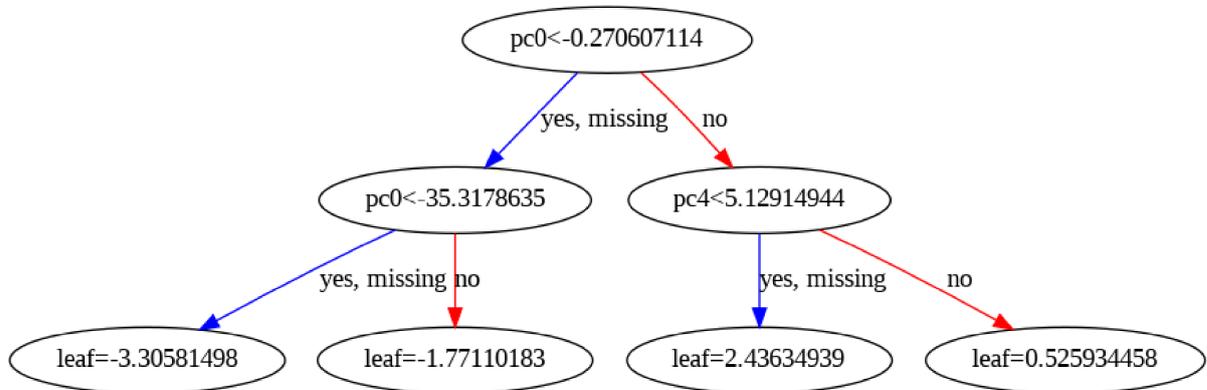
Fonte: Autoria própria

Neste caso específico, onde o número de interações é igual em ambos modelos de otimização, o processo de otimização bayesiana se mostrou mais eficaz, ao obter um erro menor, em

comparação a otimização aleatória.

A figura 24 apresenta a primeira árvore de decisão obtida na otimização bayesiana.

Figura 24 – Primeira árvore de decisão com hiperparâmetros da otimização Bayesiana



Fonte: Autoria própria

Os valores sugeridos pela otimização dos hiperparâmetros estão disponíveis nas tabelas 4 e 5.

Tabela 4 – Hiperparâmetros definidos pela Otimização Randômica

Hiperparâmetro	Valor
<i>n_estimators</i>	498
<i>eta</i>	0,15
<i>max_depth</i>	9
<i>min_child_weight</i>	9
<i>subsample</i>	0.57

Tabela 5 – Hiperparâmetros definidos pela Otimização Bayesiana

Hiperparâmetro	Valor
<i>n_estimators</i>	80
<i>eta</i>	0.15
<i>max_depth</i>	2
<i>min_child_weight</i>	2
<i>subsample</i>	0.74

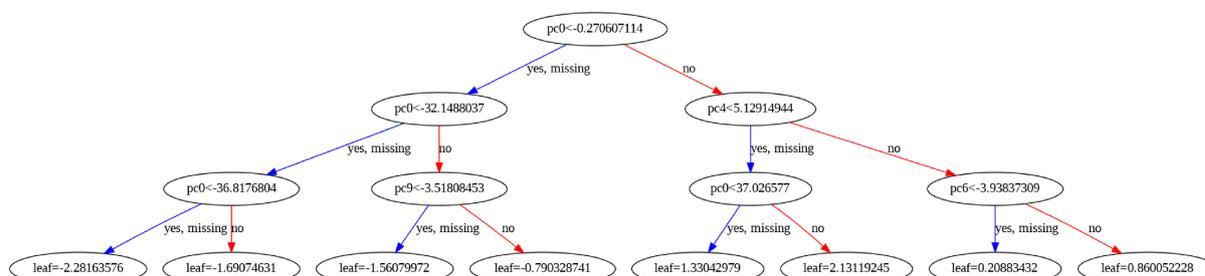
No processo de aprendizado de máquina, o objetivo não se limita a apenas obter o menor erro possível, mas sim, assegurar que o modelo possa ser generalizado, ou seja, impedir o overfitting. Tendo isso em consideração, utilizando-se dos valores fornecidos pela otimização, procurou-se ajustar os hiperparâmetros de forma conservadora. Os valores finais dos hiperparâmetros após ajustes se encontram na tabela 6

Tabela 6 – Hiperparâmetros escolhidos para a previsão da rotação

Hiperparâmetro	Valor
$n_estimators$	300
eta	0.1
max_depth	3
min_child_weight	17
$subsample$	0.7

Com os hiperparâmetros otimizados o valor do raiz quadrada do erro médio quadrado (RMSE) foi de 1.55, diminuindo o erro em relação aos valores pré-definidos do algoritmo. A primeira árvore de decisão foi gerada e se encontra na figura 25

Figura 25 – Primeira árvore de decisão com os hiperparâmetros definidos



Fonte: Autoria própria

Os Valores da previsão máxima e mínima, bem como as medidas de tendência central (média e mediana) e dispersão (variância e desvio padrão) estão apresentadas na tabela 7

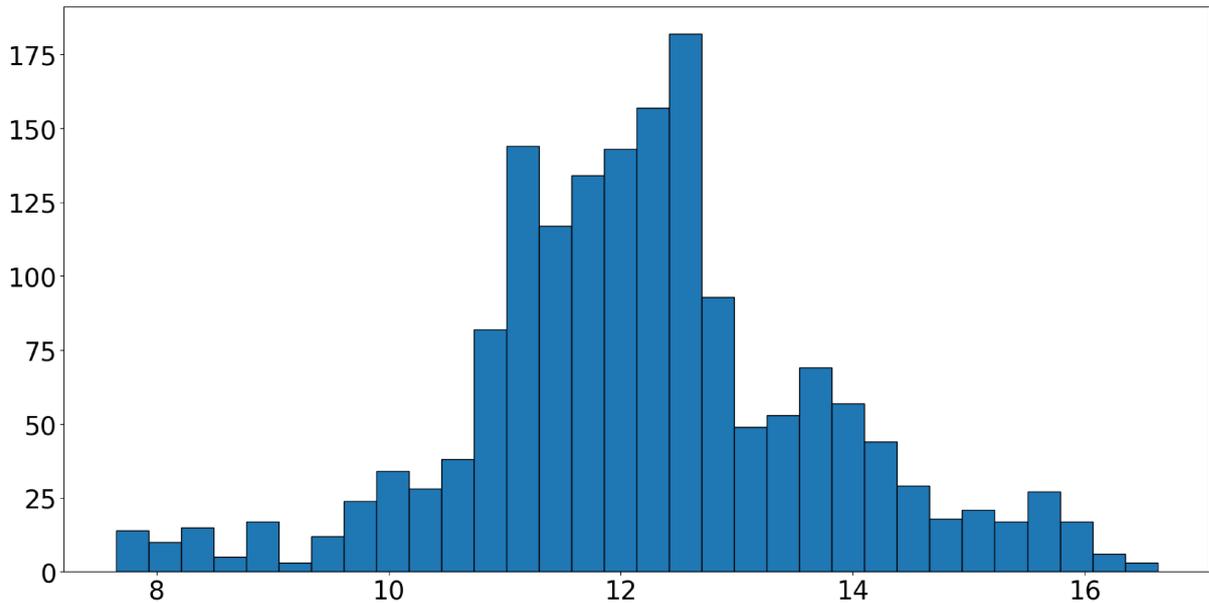
Tabela 7 – Medidas de tendência central e dispersão para a previsão da rotação $\omega = 12$

Medida	Valor
Média (\bar{x})	12.2
Mediana (\tilde{x})	12.15
Máximo (max)	16.63
Mínimo (min)	7.65
Variância (σ^2)	2.38
Desvio Padrão (σ)	1.54

O histograma apresentado na figura 26 permite visualizar a dispersão das 1662 previsões feitas. A quantidade de classes foi obtida utilizando o método Freedman–Diaconis:

$$\text{Número de barras} = \frac{\text{Amplitude dos dados}}{2 \times IQR \times \sqrt[3]{n}} \quad (5.2)$$

Onde IQR é o intervalo interquartil (a diferença entre o terceiro e o primeiro quartil) e n é o número de observações na amostra. A quantidade de classe foi definida em 32 com uma largura de 0.28.

Figura 26 – Histograma das previsões para a rotação $\omega = 12$ 

Fonte: Autoria própria

As métricas estatísticas e o histograma permite a conclusão de que a previsão feita pelo algoritmo XGboost foi satisfatória. Algumas hipóteses foram levantadas para justificar o erro das previsões:

1. A variação nos resultados da previsão podem ser explicados com base no instante de coleta dos dados dos sensores, já que o processo inicia-se no tempo $t = 0s$, onde o escoamento ainda não está desenvolvido.
2. Ao lidar com equações diferenciais em engenharia, os dados frequentemente exibem autocorrelação, o que implica em observações correlacionadas ao longo do tempo. O uso de modelos de machine learning é útil nesses casos, permitindo capturar as complexas relações entre as variáveis. No entanto, a autocorrelação pode resultar em desvios da média, levando a previsões enviesadas. Para mitigar esse problema, é necessário incorporar informações temporais nos modelos, utilizar técnicas adequadas para séries temporais e analisar os resíduos do modelo. Essas abordagens visam capturar a estrutura temporal dos dados e melhorar a precisão das previsões.

6. Conclusão

A análise de escoamento de uma esfera apresenta um conjunto de fenômenos interessantes de alta aplicabilidade na engenharia.

No presente trabalho foi possível agregar a análise numérica e o aprendizado de máquina em um era onde os dados estão cada vez mais abundantes. Como era esperado, o efeito Magnus foi observado no escoamento sob um esfera em rotação, variando-se a rotação da mesma, quer no sentido horário, como anti-horário.

O modelo matemático foi desenvolvido utilizando as equações de Navier-Stokes e foi possível a utilização de malhas adaptativas que produziram uma economia no gasto computacional. Para se simular um objeto esférico inserido em um fluido foi utilizado o método da Fronteira Imersa e a interação fluido estrutura é dividida computacionalmente em uma malha euleriana e uma malha lagrangiana. O algoritmo XGBoost foi utilizado e otimizado consoante as necessidades do problema em questão, produzindo resultados satisfatórios. O metamodelo desenvolvido foi capaz de prever a rotação da esfera através das informações obtidas ao longo do domínio computacional onde o escoamento se desenvolveu.

Referências

- ACHENBACH, E. Experiments on the flow past spheres at very high reynolds numbers. *Journal of fluid mechanics*, Cambridge University Press, v. 54, n. 3, p. 565–575, 1972.
- BABINSKY, H. How do wings work? *Physics Education*, v. 38, p. 497, 11 2003.
- BARBOSA, P. C. P. d. F. Modelagem matemático-numérico-computacional do transporte e deposição de sólidos em processo de perfuração em águas profundas. Universidade Federal de Uberlândia, 2023.
- BARTON, R. R.; MECKESHEIMER, M. Metamodel-based simulation optimization. *Handbooks in operations research and management science*, Elsevier, v. 13, p. 535–574, 2006.
- CHABANET, S.; EL-HAOUZI, H. B.; THOMAS, P. Coupling digital simulation and machine learning metamodel through an active learning approach in industry 4.0 context. *Computers in Industry*, Elsevier, v. 133, p. 103529, 2021.
- CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. [S.l.: s.n.], 2016. p. 785–794.
- FRIEDMAN, J. H. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, JSTOR, p. 1189–1232, 2001.
- JONES, D. A.; CLARKE, D. *Simulation of flow past a sphere using the fluent code*. [S.l.], 2008.
- LYU, B.; KENSRUD, J.; SMITH, L. The reverse magnus effect in golf balls. *Sports Engineering*, Springer, v. 23, p. 1–7, 2020.
- NETO, A. d. S. *Escoamentos turbulentos : análise física e modelagem teórica*. Uberlândia: Composer, 2020.
- NEWTON, I. A letter of mr. isaac newton, professor of the mathematicks in the university of cambridge; containing his new theory about light and colors: sent by the author to the publisher from cambridge, febr. 6. 1671/72; in order to be communicated to the r. society. *Philosophical transactions of the Royal Society of London*, The Royal Society London, v. 6, n. 80, p. 3075–3087, 2014.
- ROMA, A. M.; PESKIN, C. S.; BERGER, M. J. An adaptive version of the immersed boundary method. *Journal of computational physics*, Elsevier, v. 153, n. 2, p. 509–534, 1999.
- SCHOLZ, M. *Approaches to analyse and interpret biological profile data*. Tese (Doutorado) — Universität Potsdam, 2006.
- SEIFERT, J. A review of the magnus effect in aeronautics. *Progress in aerospace sciences*, Elsevier, v. 55, p. 17–45, 2012.

STOKES, G. G. et al. On the effect of the internal friction of fluids on the motion of pendulums. Pitt Press Cambridge, 1851.

TANEDA, S. Experimental investigation of the wake behind a sphere at low Reynolds numbers. *Journal of the physical society of Japan*, The Physical Society of Japan, v. 11, n. 10, p. 1104–1108, 1956.

VILLAR, M. M. et al. Análise numérica detalhada de escoamentos multifásicos bidimensionais. Universidade Federal de Uberlândia, 2007.

WANG, Z.; FAN, J.; LUO, K. Combined multi-direct forcing and immersed boundary method for simulating flows with moving particles. *International Journal of Multiphase Flow*, Elsevier, v. 34, n. 3, p. 283–302, 2008.

WASHINGTON, T. *Suspension bridge collapses into the Tacoma Narrows*. 1940. <<https://lccn.loc.gov/2006687436>>.

XU, J. et al. Simulation optimization in the era of industrial 4.0 and the industrial internet. *Journal of Simulation*, Taylor & Francis, v. 10, n. 4, p. 310–320, 2016.