



UNIVERSIDADE FEDERAL DE PERNAMBUCO

DEPARTAMENTO DE ENGENHARIA MECÂNICA
BACHARELADO EM ENGENHARIA NAVAL E OCEÂNICA

**Otimização do Design de Bulbo de Embarcações para
Redução da Resistência ao Avanço Utilizando Algoritmos
Evolucionários e Dinâmica dos Fluidos Computacional**

GUSTAVO SILVEIRA DE ARAÚJO

PROFESSOR ORIENTADOR: MIGUEL ANGEL CELIS CARBAJAL

RECIFE

OUTUBRO, 2024

GUSTAVO SILVEIRA DE ARAÚJO

Otimização do Design de Bulbo de Embarcações para Redução da Resistência ao Avanço Utilizando Algoritmos Evolucionários e Dinâmica dos Fluidos Computacional

Esta monografia é o trabalho de conclusão de curso apresentado à Universidade Federal de Pernambuco com objetivo de obtenção do título de Bacharel em Engenharia Naval e Oceânica.

Professor orientador: Miguel Angel Celis Carbajal

RECIFE

OUTUBRO, 2024

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Araújo, Gustavo Silveira de.

Otimização do Design de Bulbo de Embarcações para Redução da Resistência
ao Avanço Utilizando Algoritmos Evolucionários e Dinâmica dos Fluidos
Computacional / Gustavo Silveira de Araújo. - Recife, 2024.

97 p. : il., tab.

Orientador(a): Miguel Angel Celis Carbajal

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de
Pernambuco, Centro de Tecnologia e Geociências, Engenharia Naval -
Bacharelado, 2024.

Inclui apêndices.

1. Otimização. 2. Design do bulbo. 3. Resistência ao avanço. 4. Dinâmica dos
fluidos computacional (CFD). 5. Algoritmos evolucionários. I. Carbajal,
Miguel Angel Celis . (Orientação). II. Título.

620 CDD (22.ed.)



Universidade Federal de Pernambuco
Departamento de Engenharia Mecânica
Graduação em Engenharia Naval
Centro de Tecnologia e Geociências- CTG



ATA DE SESSÃO DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO - TCC

No dia 25/10/2024, às 10:00:00, de maneira presencial, reuniu-se a banca examinadora para a sessão pública de defesa do Trabalho de Conclusão de Curso em Engenharia Naval da Universidade Federal de Pernambuco, intitulado: **Otimização do Design de Bulbo de Embarcações para Redução da Resistência ao Avanço Utilizando Algoritmos Evolucionários e Dinâmica dos Fluidos Computacional**, elaborado pelo aluno **Gustavo Silveira Araújo**, matrícula 20180018303, composta por **Prof. Miguel Angel Celis Carbajal**, (orientador), **Aluizio Fausto Ribeiro Araújo**, (examinador 1) e **Cesar Augusto Salhua Moreno**, (examinador 2). Após a exposição oral, o candidato foi arguido pelos componentes da banca que em seguida reuniram-se reservadamente, e deliberaram pela aprovação da candidata no Trabalho de Conclusão do Curso de Engenharia Naval da UFPE, atribuindo à monografia a média **9.0** (nove). Para constar, redigi a presente Ata, aprovada por todos os presentes, que vai assinada por mim e pelos demais membros da banca.

Orientador: **Prof. Miguel Angel Celis Carbajal** – Eng. Naval/DEMEC/UFPE Nota: **9.0**

Assinatura: _____

Examinador 1: **Aluizio Fausto Ribeiro Araújo** – CIn / UFPE. Nota: **9.0**

Assinatura: _____

Examinador 2: **Cesar Augusto Salhua Moreno** – Eng. Naval/DEMEC/UFPE Nota: **9.0**

Assinatura: _____

Recife, 25/10/2024

Prof. Adriano Dayvson Marques Ferreira
Coordenador do Curso de Graduação em Engenharia Naval

RESUMO

O presente trabalho busca desenvolver um método sistemático para a otimização preliminar do *design* do bulbo de uma embarcação, focando na redução de resistência ao avanço por geração de ondas. Para isto, uma combinação de dinâmica dos fluidos computacional (CFD) e algoritmos evolucionários (EA) é utilizada, procurando encontrar configurações geométricas que minimizem a resistência ao avanço. Inicialmente, uma embarcação de referência é escolhida e a geometria de seu bulbo é isolada e modificada aleatoriamente com restrições para criar um espaço amostral. Cada modificação tem restrições espaciais e é testada com simulação numérica CFD para avaliar a resistência ao avanço, assim permitindo a criação de uma base de dados para análise. Dos resultados, um algoritmo evolucionário com modelo substitutos treinados pelos dados é implementado para gerar novas geometrias otimizadas, buscando uma geometria com menor resistência ao avanço. Ademais, os dados gerados permitem uma análise da influência da geometria na performance hidrodinâmica da embarcação.

Palavras chave: Otimização, design do bulbo, resistência ao avanço, Dinâmica dos Fluidos Computacional (CFD), algoritmos evolucionários.

ABSTRACT

This work aims to develop a systematic method for the preliminary optimization of a vessel's bulb design, focusing on reducing the resistance to forward motion due to wave formation. For this purpose, a combination of computational fluid dynamics (CFD) and evolutionary algorithms (EA) is used, seeking to find geometric configurations that minimize the resistance to forward motion. Initially, a reference vessel is selected, and its bulb geometry is isolated and randomly modified to create a sample space. Each modification has spatial constraints and is tested with CFD numerical simulations to evaluate the resistance to forward motion, thus allowing the creation of a database for analysis. From the results, a data driven evolutionary algorithm is implemented to generate new optimized geometries, iteratively reducing the resistance to forward motion. In addition, the generated data allow an analysis of the influence of geometric characteristics on the hydrodynamic performance of the vessel.

Keywords: Optimization, bulb design, resistance to forward motion, Computational Fluid Dynamics (CFD), evolutionary algorithms.

LISTA DE FIGURAS

Figura 2.4.1 - Comparação entre amostragem aleatória simples e amostragem por hipercubo latino para um espaço amostral de $\mathbb{R}^2$.	18
Figura 2.6.2.1 - Superfície de um cubo e sua malha tridimensional simples.	23
Figura 2.6.2.2 - Método de célula cortada simples (esquerda) e com definição de tamanho alvo para células cortadas (direita).	24
Figura 2.6.2.3 - Exemplo de camada limite em malha triangular.	24
Figura 2.6.3.1 - Representação do método VOF numa malha bidimensional onde a área hachurada representa preenchimento com fluido.	25
Figura 2.6.4.1 - Comparação de lógica do método explícito e implícito.	26
Figura 2.7.1 - Representação de rede neural artificial.	28
Figura 2.7.2.1 - Fluxograma do funcionamento de um EA genérico.	30
Figura 2.7.2.2 - Fluxograma do DDAE rankeado de forma probabilística.	32
Figura 3.1.1 - Modelo de rebocador Lampião na prova DUNA de 2017.	34
Figura 3.1.2 - Modelo CAD embarcação Superlâmpião e seu bulbo recortado.	34
Figura 3.3.1.1 - Superfícies de um bordo do bulbo e do casco importadas e transformadas em parte única fechada.	38
Figura 3.3.2.1 - Laço da simulação para múltiplos bulbos.	40
Figura 4.1.1 - Pontos de controle do bulbo original (preto) e pontos de controle da malha computacional (vermelho).	42
Figura 4.1.2 - Pontos de controle do bulbo deformados.	43
Figura 4.1.3 - Curvaturas gaussiana (esquerda) e média (direita) de bulbos deformados.	45
Figura 4.2.1 - Vista geral da malha do domínio.	46
Figura 4.2.2 - Vista geral da malha em torno do casco.	46
Figura 4.2.3 - Vista geral da malha próxima ao bulbo e na popa.	47
Figura 4.2.4 - Residuais da simulação.	47
Figura 4.2.5 - Componente da força na direção x do modelo ao longo do tempo com os bulbos modificados a cada quatro segundos.	48
Figura 4.2.6 - Componente da força na direção x entre 30 e 90 segundos.	48

Figura 4.2.7 - Componente da força na direção z do modelo ao longo do tempo com os bulbos modificados a cada quatro segundos.	49
Figura 4.2.7 - Bulbo 66 com menor resistência ao avanço.	51
Figura 4.2.8 - Bulbo 74 com maior resistência ao avanço	51
Figura 4.2.9 - Bulbo 57 com menor empuxo.	51
Figura 4.2.10 - Bulbo 37 com maior empuxo.	52
Figura 4.3.1 - Gráfico de dispersão comparando a resistência ao avanço com CLRP.	53
Figura 4.3.2 - Gráfico de dispersão comparando a resistência ao avanço com CZB .	53
Figura 4.3.3 - Gráfico de dispersão comparando a resistência ao avanço com a magnitude média da curvatura gaussiana.	54
Figura 4.3.4 - Gráfico de dispersão comparando a resistência ao avanço com a magnitude média da curvatura média.	54
Figura 4.4.1 - Comportamento das redes RBF em função do número de neurônios.	55
Figura 4.4.2 - Comparação das previsões dos modelos RBF treinados com 200 x 78 neurônios com os resultados reais dos casos de validação.	56
Figura 4.4.3 - Comparação das previsões dos modelos RBF treinados com 1 x 78 neurônios com os resultados reais dos casos de validação.	56
Figura 4.4.4 - Comparação das previsões dos modelos RBF treinados com 20 x 78 neurônios com os resultados reais dos casos de validação.	57
Figura 4.5.1 - Geometria do bulbo ótimo previsto pelo AE.	57
Figura 4.5.2 - Curvaturas Gaussiana (esquerda) e Média (direita) do bulbo ótimo.	58
Figura 4.5.3 - Gráfico de dispersão comparando a resistência ao avanço com CLPR incluindo os resultados do bulbo ótimo.	59
Figura 4.5.4 - Gráfico de dispersão comparando a resistência ao avanço com CZB incluindo os resultados do bulbo ótimo.	59
Figura 4.5.5 - Gráfico de dispersão comparando a resistência ao avanço com a Magnitude Média da Curvatura Gaussiana incluindo os resultados do bulbo ótimo.	60
Figura 4.5.6 - Gráfico de dispersão comparando a resistência ao avanço com a Magnitude Média da Curvatura Média incluindo os resultados do bulbo ótimo.	60

LISTA DE TABELAS

Tabela 4.1.1 - Propriedades geométricas dos bulbos modificados.	44
Tabela 4.1.2 - Propriedades das curvaturas dos bulbos modificados com exclusão de outliers.	44
Tabela 4.2.1 - Resultados de CFD da resistência ao avanço.	49
Tabela 4.2.2 - Resultados de CFD do empuxo.	50
Tabela 4.2.3 - Resultados de CFD dos bulbos com resistência ao avanço máxima (74) e mínima (66) e empuxo máximo (37) e mínimo (57).	50
Tabela 4.3.1 - Coeficientes das regressões lineares de CLRP e CZB.	52
Tabela 4.5.1 - Propriedades Geométricas do bulbo ótimo encontrado pelo EA.	57

LISTA DE ABREVIATURAS E SIGLAS

EA	<i>Evolutionary Algorithm</i> (Algoritmo Evolucionário)
A_{MD}	Área da secção mestra de uma embarcação.
B_{MD}	Boca moldada na secção mestra de uma embarcação.
CAD	<i>Computer Aided Design</i> (Design Auxiliado por Computador)
CNN	<i>Convolutet Neural Network</i> (Rede Neural Convolucional)
CFD	<i>Computational Fluid Dynamics</i> (Dinâmica dos Fluidos Computacional)
DDEA	<i>Data Driven Evolutionary ALgorithm</i> (Algoritmo Evolucionário Baseado em Dados)
DLNN	<i>Deep Learning Neural Network</i> (Rede Neural de Aprendizagem profunda)
DNS	<i>Direct Numerical Simulation</i> (Simulação Numérica Direta)
FFD	<i>Free Form Deformation</i> (Deformação de Forma Livre)
FPM	<i>Fully Parametric Models</i> (Modelos Completamente Paramétricos)
GAN	<i>Generative Adversarial Network</i> (Rede GeradoraAdversarial)
IC	Inteligência Computacional
IQR	Intervalo Inter Quartil
JSON	<i>JavaScript Object Notation</i> (Notação de Objeto JavaScript)
L_{LWL}	Comprimento na linha de água do casco.
NN	<i>Neural Network</i> (Redes Neurais, também abreviado com ANN)
NURBS	<i>Non Uniform Rational Basis Spline</i> (Spline de Base Racional Não Uniforme)

PMTM	<i>Parametric Mesh Transformation Models</i> (Métodos de Transformação de Malha Paramétricos)
RANS	<i>(Reynolds Averaged Navier Stokes)</i> Média Reynolds de Navier-Stokes
RBF	<i>Radial Basis Function</i> (Funções de Base Radial)
ROM	<i>Reduced Order Methods</i> (Métodos de Ordem Reduzida)
VOF	<i>Volume Of Fluid</i> (Volume de Fluido)

SUMÁRIO

1. INTRODUÇÃO	10
1.1. Objetivo	10
1.2. Estrutura do trabalho	11
1.3. Revisão da literatura	11
1.4. Justificativa e contribuição	13
2. FUNDAMENTAÇÃO TEÓRICA	14
2.1. Resistência ao avanço	14
2.2. Proa bulbosa	14
2.3. Transformações FFD	16
2.4. Amostragem por hipercubo latino	18
2.5. Curvatura Gaussiana e Média	18
2.6. Dinâmica dos fluidos computacional	20
2.6.1. Modelos físicos e matemáticos	20
2.6.2. Malha computacional	23
2.6.3. Considerações para múltiplas fases	25
2.6.4. Regime transiente	26
2.6.5. Análise de convergência	26
2.7. Inteligência computacional	27
2.7.1. Redes neurais artificiais	27
2.7.2. Algoritmos evolucionários	29
3. MÉTODOLOGIA	33
3.1. Seleção de um modelo teste	33
3.2. Modificações do bulbo	34
3.2.1. Obtenção dos parâmetros de comparação do bulbo	36
3.3. Análise CFD	36
3.3.1. STAR CCM+	37
3.3.2. Automatização da análise - Macros	39
3.4. Definição de relações	41
3.5. Análises por Inteligência Computacional	41
4. RESULTADOS	42
4.1. Modificações do bulbo	42
4.2. Análise CFD	46
4.3. Definição de relações	52
4.4. Treinamento das redes RBF	55
4.5. Bulbo encontrado pelo EA	57
5. DISCUSSÕES	61
6. CONCLUSÕES	64
7. REFERÊNCIAS	65
ANEXO I - CONFIGURAÇÕES DA SIMULAÇÃO NO STAR-CCM+	69

1. INTRODUÇÃO

No projeto e *design* naval, o aprimoramento da forma do casco em questão da resistência ao avanço é essencial para diminuir o consumo de combustível e aumentar a eficiência, logo, o retorno financeiro das embarcações. Esta etapa do projeto preliminar de uma embarcação pode demandar muito tempo de engenharia e por tratar do complexo campo de hidrodinâmica, não existe uma formulação simples que encontre a melhor geometria do casco.

Engenheiros navais vem se utilizando de técnicas de previsão como Dinâmica dos Fluidos Computacional (CFD) para prever a resistência ao avanço de uma geometria. Enquanto que estas técnicas podem prever com boa acurácia a resistência ao avanço, não são técnicas de otimização e utilizando apenas CFD um engenheiro pode não ter muita escolha além de utilizar de seu conhecimento e experiência para fazer simulações de diversas geometrias até encontrar uma que satisfaça, mesmo não sendo a solução ótima.

Para a resistência ao avanço, uma característica que se tornou essencial em embarcações com muita resistência devido a geração de ondas, é a presença de um bulbo, que modifica o padrão de ondas gerado pela embarcação, diminuindo sua resistência ao avanço em altas velocidades.

Recentemente, métodos de inteligência computacional vêm sendo empregados no campo acadêmico da engenharia naval para encontrar métodos de otimização da resistência ao avanço, porém, a maioria desses métodos permitem modificações ao formato das geometrias que ou não são aplicáveis a todas embarcações, ou tornam a embarcação não viável para seu uso no mercado real (MIKULIĆ e PARUNOV 2019).

1.1. Objetivo

O presente trabalho tem como objetivo desenvolver um passo a passo para melhorar o *design* do bulbo de uma embarcação no critério de redução da resistência ao avanço em seu estágio preliminar. Propõe-se fazer isso da seguinte forma:

- Selecionar a embarcação em análise.
- Isolar a geometria de seu bulbo e criar um espaço amostral de modificações aleatórias.
- Testar com uma metodologia de dinâmica dos fluidos computacional a resistência ao avanço de cada modificação.

- Criar um algoritmo evolucionário que busque uma nova modificação do bulbo cuja resistência ao avanço seja a ótima.

Ademais, com a obtenção de diversas resistências ao avanço para bulbos modificados, é possível fazer uma análise da relação entre a resistência ao avanço da embarcação e propriedades geométricas de seu bulbo.

1.2. Estrutura do trabalho

O presente relatório está dividido em capítulos, com este capítulo introdutório contextualizando o trabalho e seus objetivos. No capítulo um, busca-se analisar o estado da arte na utilização de metodologias para otimização do formato do casco com utilização de CFD, inteligência artificial e parâmetros geométricos. No capítulo dois (Fundamentação teórica) os principais aspectos teóricos utilizados são expostos e referenciados. No capítulo três (Metodologia), descreve-se os critérios utilizados para seleção do modelo otimizado, o procedimento de modificação e obtenção de deformações de seu bulbo, o método de aplicação de CFD e os testes e modelos computacionais de inteligência artificial. No capítulo quatro (Resultados) se expõe o que foi obtido com as metodologias aplicadas. No capítulo cinco (Discussão) a metodologia e resultados são retrospectivamente analisados e avaliados em seus sucessos e falhas. Finalmente, no capítulo seis (Conclusão) o autor apresenta suas considerações finais e sugestões para trabalhos futuros. Todas as fontes e referências utilizadas no trabalho estão devidamente registradas em sua bibliografia no capítulo 7 (Referências).

1.3. Revisão da literatura

Vernengo *et al.* (2020) fornece uma comparação entre FFD e FPM para modificar modelos de cascos. Este trabalho oferece dados valiosos sobre a eficiência de ambos os métodos para gerar novos recursos e as vantagens e dificuldades de trabalhar com cada um deles.

Salmoiraghi *et al.* (2018) define uma metodologia para aplicar ROM a uma deformação a partir do FFD. Os ROMs exploram o fato de que mudanças suaves e contínuas em uma superfície muitas vezes alteram a propriedade analisada de maneira suave e contínua. Isso permite que um algoritmo estime fenômenos físicos (por exemplo, as propriedades aerodinâmicas de um carro ou a resistência à água de um navio) com menos tempo de processamento.

Zhou *et al.* (2022) desenvolveram uma solução de software para criar superfícies NURBS para navios da série 60 (*i.e.*, um modelo teórico de navio mercante com um único hélice desenvolvido por Todd (1963)). O *software* prova a eficácia de criar múltiplas variações de um modelo base apenas alterando os parâmetros base das curvas geradoras hipotéticas.

Em vez de se concentrar em modificar a geometria de um casco, Yu *et al.* (2015) foca na criação de um novo PMTM para modificar a geometria do casco diretamente como uma malha computacional. Enquanto que Wang e Kim (2023) desenvolveram uma técnica universal para deformação de malhas utilizando curvas de bézier genéricas. Esses métodos são valiosos quando se planeja analisar múltiplas geometrias distintas em análise CFD de alta fidelidade, pois remove a necessidade de processamento não trivial para gerar uma nova malha para cada superfície testada.

Margari *et al.* (2018) desenvolveram uma NN para prever a resistência ao avanço em águas calmas da série sistemática de navios MARAD, proposta por Roseman (1987). A NN alcança menos de 2% (dois por cento) de discrepância em relação à resistência calculada de um navio MARAD. No entanto, a NN treinada não funcionaria em navios fora da série sistemática.

Kim *et al.* (2022) focou em utilizar as imagens obtidas por análise CFD dos campos de pressão e perfis de onda e esteira como indicadores qualitativos para alimentar uma DLNN do tipo CNN enquanto que Ao *et al.* (2023) desenvolveram um método para estimar a resistência também com uma DLNN alimentando-a com dados de geometria e resistência de muitos modelos tridimensionais. Esses modelos tridimensionais foram gerados utilizando transformação linear e FFD em uma superfície de casco base, e sua resistência foi calculada utilizando CFD.

O trabalho de Ao *et al.* (2023) é promissor em termos de sucesso no treinamento de uma NN para estimar a resistência, com o menor poder computacional necessário. Porém, como apenas um tipo específico de formato de casco foi utilizado no treinamento (ou seja, o navio contêiner KRISO), ainda não se sabe se a NN desenvolvida pode estimar corretamente a resistência quando treinada para diferentes formatos de casco.

Khan *et al.* (2023) propõe um método para gerar formatos de cascos a partir de um casco base. Isso é alcançado através de uma GAN, treinada ao receber dados de resistência de onda e geometria para vários tipos de casco.

Este trabalho (KHAN *et al.*, 2023) fornece um excelente método para descrever diferentes tipos de casco como uma malha padronizada (ou seja, mesma resolução). No entanto, essa GAN produziu muitos formatos inválidos (*e.g.*, proas bulbosas e popas de construção impraticável, superfícies que se intersectam). Além disso, como apenas a resistência de onda foi considerada, navios especializados perderam características necessárias (por exemplo, navios contêineres com curvatura excessiva em seus porões, provavelmente resultando em menos espaço útil de carga).

Kim *et al.* (2006) desenvolveram uma NN para estimar quais tipos de placas (*e.g.*, primeira, segunda e terceira superfícies) estarão presentes em cada região do casco, apenas inserindo os dados de curvatura dos pontos da superfície do navio. Como a dificuldade de formar uma placa pode ser estimada pelo tipo de placa, este trabalho prova a viabilidade de utilizar dados de curvaturas para otimizar a construção do casco reduzindo sua dificuldade de formação.

No tópico da dificuldade de formação, Wang *et al.* (2023) criaram uma metodologia matemática para comparar diferentes placas de casco. Utilizando a Análise de Relação Grey, a associação desconhecida entre os dados geométricos da placa (ou seja, centro de curvatura, espessura, largura, comprimento) e o tempo de formação é estimada. Isso nos fornece um método para quantificar e comparar diferentes partes dos cascos em sua dificuldade de construção estimada.

1.4. Justificativa e contribuição

Este trabalho de conclusão de curso, busca encontrar uma metodologia de otimização do bulbo de uma embarcação em seu caráter de redução da resistência ao avanço de forma que não prejudique sua viabilidade de construção, utilizando Algoritmos Evolucionários (EA) informados por dados obtidos de simulações CFD.

2. FUNDAMENTAÇÃO TEÓRICA

Este capítulo busca deixar o leitor ciente dos principais conceitos utilizados durante a execução do trabalho para facilitar a compreensão para os que ainda não exploraram as áreas de análise resistência ao avanço com métodos CFD e aplicações de engenharia de IC.

2.1. Resistência ao avanço

A resistência ao avanço de uma embarcação é a força necessária para rebocá-la por águas calmas. Esta resistência aumenta de forma diretamente proporcional com a velocidade e é formada por diferentes efeitos (MANEN; OOSSANEN, 1988):

Os efeitos de fricção da embarcação se movendo por um fluido viscoso são os maiores componentes únicos da resistência ao avanço, podendo ser maiores que 80% (oitenta por cento) para embarcações de baixa velocidade (MANEN; OOSSANEN, 1988).

A resistência por geração de ondas em que a embarcação ao se mover forma ondas na superfície livre da água e fica sujeita a efeitos da diferença de pressão devido a variação de altura do fluido, é maior em embarcações de alta velocidade (MANEN; OOSSANEN, 1988).

A resistência ocorre devido a turbulência por formação de vórtices, onde se observa a formação de uma “camada limite” ao redor da embarcação entre o escoamento turbulento e laminar (MANEN; OOSSANEN, 1988).

Finalmente, se tem a resistência devido ao ar, que é principalmente significativo em embarcações com grandes obras mortas (*i.e.*, estrutura da embarcação acima da linha d'água.).

Vários métodos são utilizados para estimativas de resistência ao avanço. Molland et al. (2011) lista alguns métodos como utilização de séries sistemáticas como a Série 60, 63 e 64, regressões como as de Holtrop e Mennen, Hollenbach e Van Oortmerssen e métodos de Dinâmica dos Fluidos Computacional CFD.

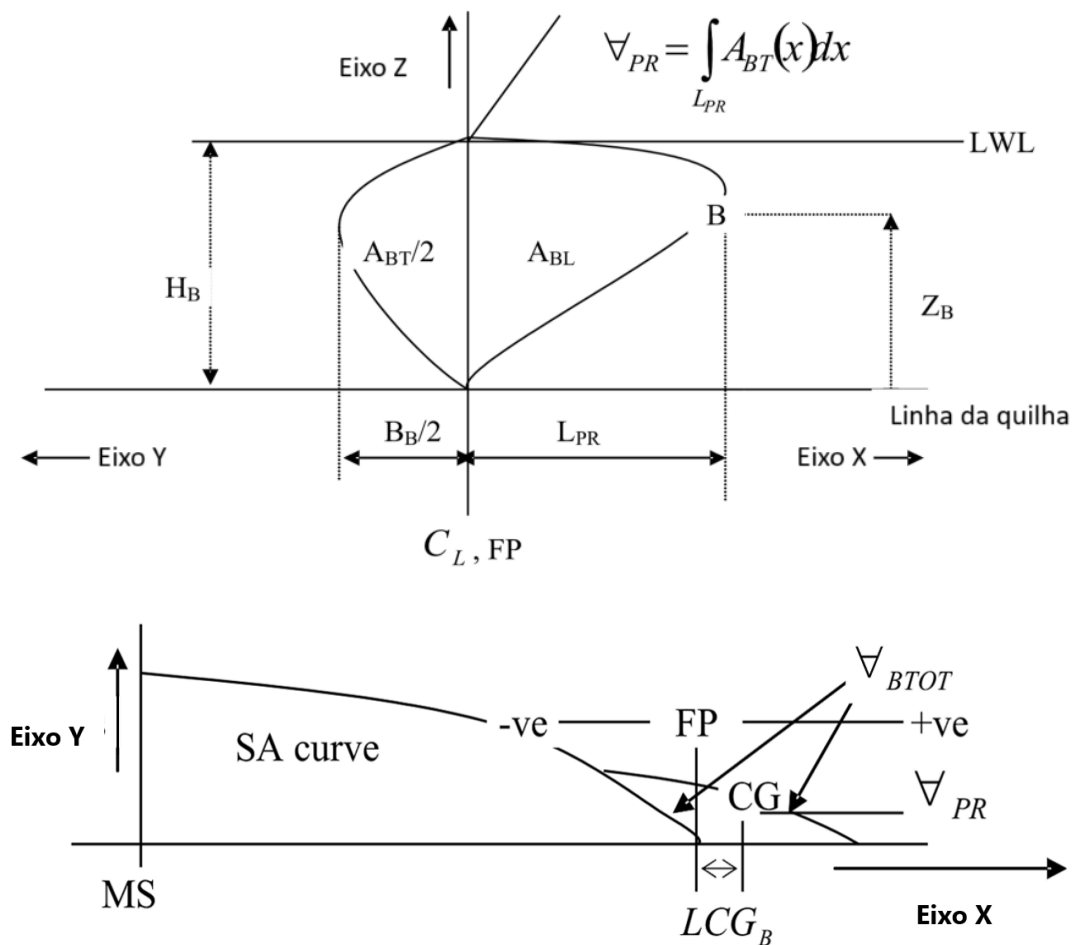
2.2. Proa bulbosa

A proa bulbosa passou a ser desenvolvida após resultados com a aplicação de um tubo de torpedo em uma embarcação militar mostrar resultados positivos de diminuição na resistência ao avanço (SHARMA; SHA, 2005).

Proas bulbosas passaram a se tornar um marco essencial da maioria das embarcações recém construídas devido a seu efeito na diminuição da resistência devido a formação de ondas. A presença de um bulbo modifica o padrão de geração de ondas na proa da embarcação, avançando a posição da criação da crista de onda gerada devido ao avanço da embarcação e diminuindo a pressão da mesma na proa da embarcação. Devido os principais ganhos em resistência serem na diminuição da resistência de ondas, a proa bulbosa é primariamente utilizada em embarcações de alta velocidade ($F_n > 0,2$) (KRACHT, 1978).

Entre esforços para estudar e otimizar os efeitos do bulbo na resistência ao avanço, Sharma e Sha (2005) propõem utilizar os parâmetros descritos por Kracht (1978) design da geometria do bulbo para comparação e validação de resultados de otimização do bulbo, definidos pelas propriedades geométricas vistas na figura 2.3.1 e equações 2.3.1 a 2.3.7.

Figura 2.3.1 - Propriedades para obtenção dos parâmetros do bulbo.



Fonte: Sharma e Sha, 2005.

Parâmetro de Largura	$C_{BB} = B_B/B_{MS}$	Eq. 2.3.1
Parâmetro de Comprimento	$C_{LPR} = L_{PR}/L_{LWL}$	Eq. 2.3.2
Parâmetro de Profundidade	$C_{ZB} = Z_B/H_B$	Eq. 2.3.3
Parâmetro da Secção	$C_{ABT} = A_{BT}/A_{MS}$	Eq. 2.3.4
Parâmetro Lateral	$C_{ABL} = A_{BL}/A_{MS}$	Eq. 2.3.5
Parâmetro Volumétrico	$C_{\forall_{TOT}} = \forall_{BTOT}/\forall_{WL}$	Eq. 2.3.6
Parâmetro do Centroide	$C_{CG} = LCG_B/(L_{WL} Fn)$	Eq. 2.3.7

2.3. Transformações FFD

Transformações FFD como descritas por Sederberg e Parry (1986) são um método para modificar uma geometria 3D de forma contínua e suave. Ao invés de editar-se diretamente a superfície, que muitas vezes é composta por uma quantidade grande de pontos, cria uma *lattice* (i.e., estrutura retangular tipo grade) ao redor da geometria que se deseja modificar e a modificação dos pontos dessa *lattice* modifica a superfície da geometria a geometria de forma proporcional por meio de polinômio de Bernstein (ver equação 2.4.1). um formato polinomial utilizado para interpolação de funções contínuas e base para as curvas de bézier (JOY, 2000).

$$B_{i,n}(t) = \frac{n!}{k!(n-k)!} t^i (1-t)^{n-i} \quad \text{Eq. 2.4.1}$$

Isso se consegue escrevendo as posições dos pontos da superfície no formato das coordenadas da *lattice* (s, t, u), ou seja cada ponto da *lattice* tem $0 \leq (s, t, u) \leq 1$. A *lattice* é dividida em pontos de controle em sua superfície com uma escolhida resolução (l, m, n) e a esses pontos também são atribuídas coordenadas (s, t, u) que logicamente também terão $0 \leq (s, t, u) \leq 1$ (SEDERBERG; PARRY, 1986).

Então, após se mover esses pontos de controle da *lattice* para uma nova posição P_{ijk} , pode-se calcular as novas posições X_{ffd} dos pontos da superfície por meio de um polinômio de Bernstein em produto tensorial tri-variado (ver equação 2.4.2). Essas modificações por natureza são contínuas e suaves, com deformações em um ponto na

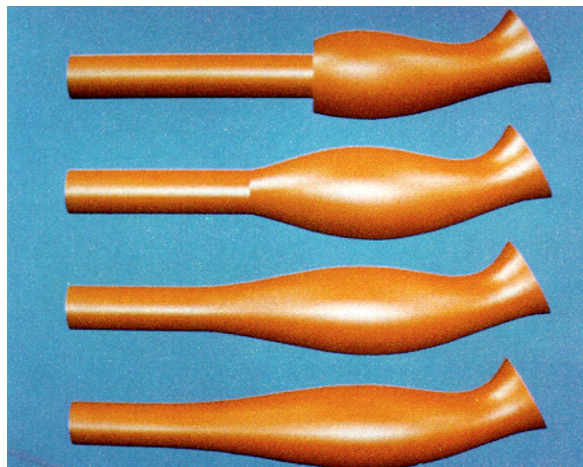
superfície sendo mais correspondentes a translação de pontos de controle próximos a eles (SEDERBERG; PARRY, 1986).

$$X_{ffd} = \sum_{i=0}^l \left\{ \sum_{j=0}^m \left[\sum_{k=0}^n \left(B_{i,l}(s) \cdot B_{j,m}(t) \cdot B_{k,n}(u) \cdot P_{ijk} \right) \right] \right\} \quad \text{Eq. 2.4.2}$$

Porém, existem considerações especiais quando a *lattice* não vai modificar toda a geometria (e.g., se ao invés de se modificar o casco inteiro de uma embarcação, se modifica apenas o bulbo), os pontos onde a geometria que não será deformada e a geometria que será se encontram pode haver uma descontinuidade. Para evitar isso, se utiliza fixação de pontos na junção das duas partes da geometria para garantir a conexão e continuidade (Sederberg e Parry, 1986).

O tipo de continuidade pode ser definido pela quantidade de pontos da extremidade fixados, com c^{-1} representando nenhum ponto e gerando descontinuidade e desconexão da superfície, c^0 representando um ponto e garantindo a conexão, com c^1 dois pontos são fixos garante a tangência e com c^2 três pontos são fixos garantindo a curvatura (ver figura 2.4.1). c^{n-1}

Figura 2.4.1 - Geometria deformada por FFD com continuidades c^{-1} , c^0 , c^1 e c^2 de cima para baixo.



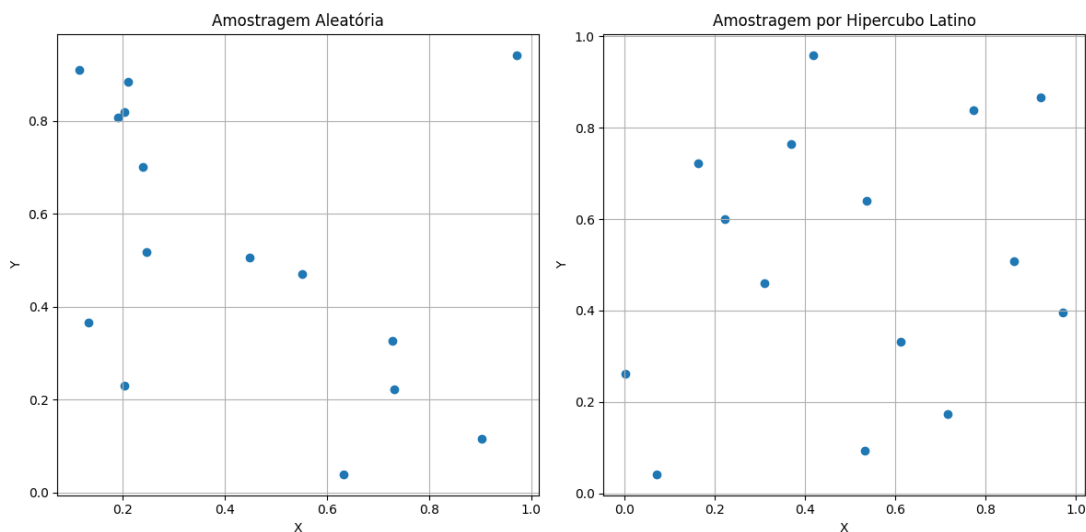
Fonte: Sederberg e Parry, 1986.

2.4. Amostragem por hipercubo latino

Hipercubo latino como explicado por Hadigol e Doostan (2018) é um método de amostragem aleatória que busca preencher o espaço amostral de forma uniforme, essa técnica é útil em problemas de computação em que diversos casos aleatórios com alto requerimento computacional são analisados, pois evita que uma amostragem completamente aleatória não reflita bem todos os casos possíveis, ou que uma quantidade grande de casos muito semelhantes gaste tempo de processamento (ver figura 2.4.1 onde X e Y podem ser variáveis do problema em estudo).

Isso é atingido dividindo todas as $\mathbb{R}^n$ dimensões do espaço amostral em partes iguais em função do número de amostras, de forma que cada uma dessas divisões contenham uma amostra e permitindo boa representação do espaço amostral quando consideradas todas as amostras (HADIGOL; DOOSTAN, 2018).

Figura 2.4.1 - Comparação entre amostragem aleatória simples e amostragem por hipercubo latino para um espaço amostral de $\mathbb{R}^2$.



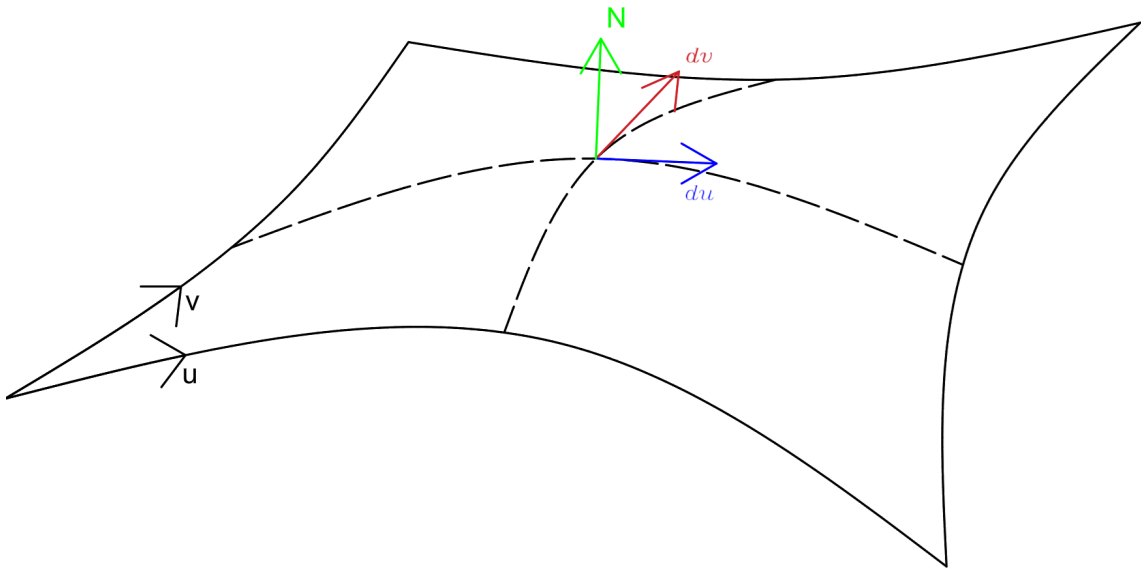
Fonte: Autor, 2024.

2.5. Curvatura Gaussiana e Média

Uma superfície tridimensional pode ser analisada a partir da curvatura de seus pontos, e em função de suas curvaturas gaussianas K e média H . (PRESSLEY, 2001). Esses parâmetros são úteis para definir como uma superfície se curva sobre um ponto e na engenharia naval pode ser utilizado como fator para prever a dificuldade de formação da chapa de um casco (KIM *et al.* 2006).

Para encontrar essas curvaturas em um ponto, é necessário utilizar a primeira e segunda forma fundamental da superfície, essas são propriedades que permitem calcular os comprimentos, ângulos e áreas na superfície com a primeira forma fundamental, e quão curva é a superfície com a segunda. Para obter essas propriedades, se define uma parametrização da superfície em direções u e v , direções nas quais se pode calcular as primeiras e segundas derivadas (ver figura 2.5.1) (PRESSLEY, 2001).

Figura 2.5.1 - Superfície parametrizada em u e v com suas primeiras derivadas e vetor normal.



Fonte: Autor, 2024.

Com as primeiras e segundas derivadas se obtém os parâmetros para se calcular a primeira (equação 2.5.1) e segunda (equação 2.5.2) formas fundamentais, que podem ser utilizadas para encontrar as curvaturas gaussiana e média pelas equações 2.5.3 e 2.5.4 (PRESSLEY, 2001).

$$E = [du, du]$$

$$F = [du, dv]$$

$$G = [dv, dv]$$

$$L = [duu, N]$$

$$M = [duv, N]$$

$$N = [dvv, N]$$

$$I = E du^2 + F du dv + G dv^2 \quad \text{Eq. 2.5.1}$$

$$II = L du^2 + M du dv + N dv^2 \quad \text{Eq. 2.5.2}$$

$$K = \frac{L N - M^2}{E G - F^2} \quad \text{Eq. 2.5.3}$$

$$H = \frac{L G - M F + N E}{2 (E G - F^2)^2} \quad \text{Eq. 2.5.4}$$

2.6. Dinâmica dos fluidos computacional

CFD ou Dinâmica dos Fluidos Computacional trata-se da obtenção da propriedade de escoamentos de forma numérica e vem se tornando cada vez mais popular com o aumento do poder computacional. Metodologias CFD seguem uma ordem geral: definição de uma malha computacional representativa do domínio do escoamento, definição de modelos físicos para calcular as propriedades do escoamento em cada ponto da malha e tratamento e análise dos resultados (Carbajal, 2024).

2.6.1. Modelos físicos e matemáticos

Como diz Anderson (1995) em seu livro *Computational Fluid Dynamics*, toda “Toda CFD, de uma forma ou outra, é baseada nas equações governantes fundamentais da dinâmica dos fluidos - as equações de continuidade, momento e energia.” ou seja, para um modelo CFD funcional, deve-se garantir que estão sendo aplicadas e respeitadas a conservação de massa, do momento e de energia.

Isso conseqüentemente evoca as equações de Navier-Stokes (Equações 2.6.1.1 e 2.6.1.2) que são a base da maioria dos modelos CFD para atender as equações de conservação. Porém, para escoamentos turbulentos com variações em pressão e velocidade (*e.g.*, um navio se movendo na superfície da água), a aplicação direta da equação de Navier-Stokes, DNS, fica exponencialmente cara computacionalmente (CEBECI et al., 2005).

$$\text{Continuidade} \quad \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad \text{Eq. 2.6.1.1}$$

$$\begin{aligned} \text{Momento} \quad \rho \frac{Du}{Dt} &= -\frac{\partial p}{\partial x} + \left(\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} \right) + \rho f_x \\ \rho \frac{Dv}{Dt} &= -\frac{\partial p}{\partial y} + \left(\frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} \right) + \rho f_y \\ \rho \frac{Dw}{Dt} &= -\frac{\partial p}{\partial z} + \left(\frac{\partial \sigma_{zx}}{\partial x} + \frac{\partial \sigma_{zy}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} \right) + \rho f_z \end{aligned} \quad \text{Eq. 2.6.1.2}$$

Onde onde ρ é a densidade, f_ϕ são as forças externas e $\frac{D\phi}{Dt}$ é a derivada material que descreve a mudança da velocidade ao longo da sua trajetória e tempo:

$$\frac{D\phi}{Dt} = \frac{\partial\phi}{\partial t} + f_\phi \nabla\phi \quad \text{Eq. 2.6.1.3}$$

E $\nabla\phi$ é a derivada ao longo dos vetores tangentes ao escoamento que representa o termo convectivo (EMANUEL, 2001).

Para diminuir os recursos computacionais necessários para esse tipo de análise, métodos de transformação da equação de Navier-Stokes são utilizados, como o método de RANS. Essa Média de Reynolds de Navier Stokes substitui as variáveis instantâneas (u, v, w) com a decomposição de Reynolds, pela soma entre sua média temporal ($\bar{u}, \bar{v}, \bar{w}$) e as flutuações (u', v', w') (*i.e.*, a diferença instantânea entre o valor local da variável e sua média temporal) (ANDERSON, 1995). Aplicando nas equações 2.6.1.1 e 2.6.1.2 temos:

$$\text{Continuidade} \quad \frac{\partial \bar{u}}{\partial x} + \frac{\partial \bar{v}}{\partial y} + \frac{\partial \bar{w}}{\partial z} = 0 \quad \text{Eq. 2.6.1.4}$$

$$\begin{aligned} \text{Momento} \quad \rho \frac{D\bar{u}}{Dt} &= -\frac{\partial p}{\partial x} + \mu \nabla^2 \bar{u} + \rho \left(f_x - \frac{\partial(\bar{u}^2)}{\partial x} - \frac{\partial(\bar{u}'v')}{\partial y} - \frac{\partial(\bar{u}'w')}{\partial z} \right) \\ \rho \frac{D\bar{v}}{Dt} &= -\frac{\partial p}{\partial y} + \mu \nabla^2 \bar{v} + \rho \left(f_y - \frac{\partial(\bar{v}'u')}{\partial x} - \frac{\partial(\bar{v}^2)}{\partial y} - \frac{\partial(\bar{v}'w')}{\partial z} \right) \\ \rho \frac{D\bar{w}}{Dt} &= -\frac{\partial p}{\partial z} + \mu \nabla^2 \bar{w} + \rho \left(f_z - \frac{\partial(\bar{w}'u')}{\partial x} - \frac{\partial(\bar{w}'v')}{\partial y} - \frac{\partial(\bar{w}^2)}{\partial z} \right) \end{aligned} \quad \text{Eq. 2.6.1.5}$$

Esta transformação nos permite filtrar as flutuações de alta frequência, porém, é necessário definir um método para lidar com os Tensores de Estresse de Reynolds $\tau'_{ij} \equiv \rho \overline{\phi'_i \phi'_j}$ introduzidos. Para se fazer isso, deve-se aplicar algum modelo de turbulência para aproximar o tensor em função das médias de comprimentos ou viscosidade de vórtices ε_m (ver equação 2.6.1.6) (CEBECI *et al.*, 2005).

$$-\rho \overline{\phi'_i \phi'_j} = \rho \varepsilon_m \frac{\partial \phi_i}{\partial j} \quad \text{Eq. 2.6.1.6}$$

Existem alguns modelos de turbulência aplicados para aproximar a viscosidade de vórtices, eles são classificados pela quantidade de equações diferenciais utilizadas para aproximação de ε_m como modelos de: zero, uma ou duas equações. Cebeci *et al.* (1995) alegam que os modelos de duas equações (ver equação 2.6.1.7) são os que se provam mais adequados para problemas de alta complexidade e com interação entre mais de uma fase de escoamento (*i.e.*, ar e água separados por uma camada limite).

$$\varepsilon_m = \frac{c_\mu k^2}{\varepsilon} \quad \text{Eq. 2.6.1.7}$$

Onde c_μ é uma constante, k e ε são obtidas por equações diferenciais. O modelo de duas equações mais aplicado é o modelo k - ε , onde k é a energia cinética e ε é a taxa de dissipação, obtidos pelas equações de energia cinética turbulenta (equação 2.6.1.8) e de dissipação (2.6.1.9) onde ν é a viscosidade cinemática do fluido.

$$\frac{Dk}{Dt} = \frac{\partial}{\partial x} \left[\left(\nu + \frac{\varepsilon_m}{\sigma_k} \right) \frac{\partial k}{\partial x} \right] + \varepsilon_m \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \frac{\partial \bar{u}_i}{\partial x_j} - \varepsilon \quad \text{Eq. 2.6.1.8}$$

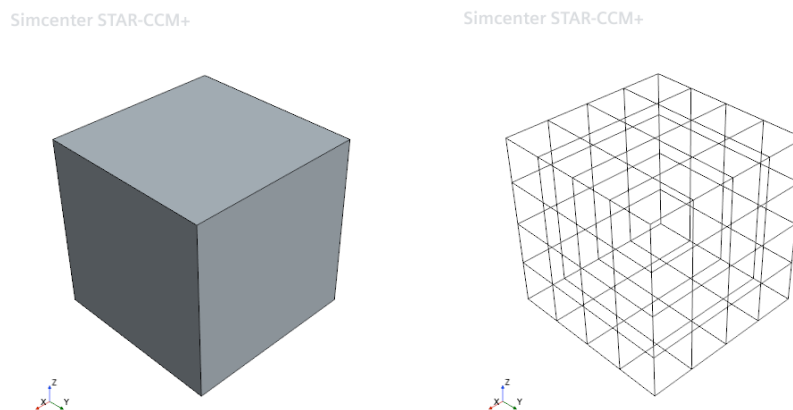
$$\frac{D\varepsilon}{Dt} = \frac{\partial}{\partial x_k} \left[\left(\nu + \frac{\varepsilon_m}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_k} \right] + c_{\varepsilon 1} \frac{\varepsilon}{k} \varepsilon_m \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \frac{\partial \bar{u}_i}{\partial x_j} - c_{\varepsilon 2} \frac{\varepsilon^2}{k} \quad \text{Eq. 2.6.1.9}$$

Para se fechar o problema, aplicam-se condições de contorno dependendo se o problema é de alto ou baixo número de Reynolds, e se o escoamento está próximo a uma interface limite (*i.e.*, paredes). Diversas metodologias de condição de contorno foram testadas incluindo modelos k - ε padrão (JONES; LAUDNER, 1972), de duas camadas (RODI, 1991), e k - ε Realizável, no qual os autores Shih *et al.* (1995) implementam formulações de vorticidade não lineares e funções de amortecimento próximo a paredes, e observam melhor previsão das tensões de Reynolds em regimes como camadas limite e conseqüentemente resultados mais precisos da equações de RANS.

2.6.2. Malha computacional

Para se resolver as equações governantes, é necessário criar uma malha computacional onde diversos elementos chamados células são posicionados e as propriedades do escoamento são calculadas entre eles (ver figura 2.6.2.1). A qualidade da malha computacional utilizada é fundamental para a convergência e acurácia dos resultados obtidos (ANDERSON, 1995).

Figura 2.6.2.1 - Superfície de um cubo e sua malha tridimensional simples.

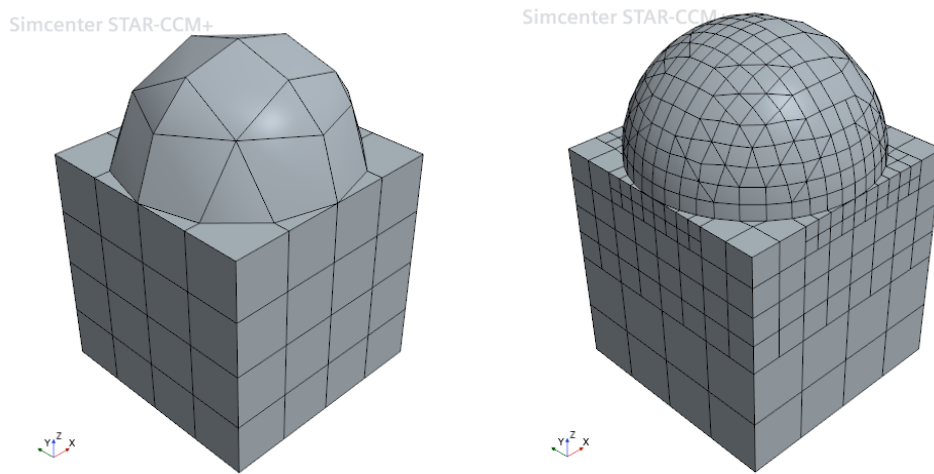


Fonte: Autor, 2024.

Na realidade, para escoamentos no domínio tridimensional, não é possível utilizar uma malha cartesiana simples como exemplificada na figura 2.6.2.1, como um escoamento real conterá curvas, camadas limites, paredes e outras complicações, deve-se utilizar alguma metodologia para mapear a malha (ANDERSON, 1995).

Uma das metodologias mais simples e eficazes é a metodologia de malha cartesiana cortada (*i.e.*, também conhecido como *cut-cell* ou *trimmed cell*), neste método uma malha cartesiana de tamanho alvo S simples é sobreposta sobre o domínio desejado e cortada na intersecção da superfície, desta forma, a maior parte do domínio será composto por células cartesianas no tamanho alvo (CALHOUN e LEVEQUE, 2000). Para se aprimorar a superfície onde existem os limites do domínio, pode se utilizar uma técnica que define que nos pontos onde a célula cartesiana original é cortada, ela se subdivide n vezes até que as células cortadas cheguem a um tamanho alvo especificado, isso permite aumento da densidade da malha nessas regiões de interesse (SIEMENS, 2018).

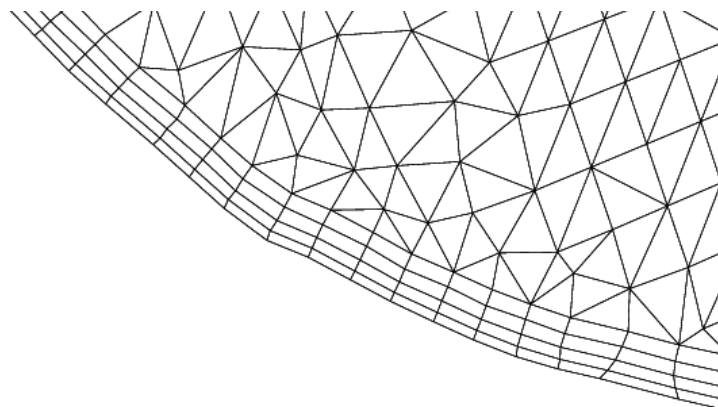
Figura 2.6.2.2 - Método de célula cortada simples (esquerda) e com definição de tamanho alvo para células cortadas (direita).



Fonte: Autor, 2024.

Outra técnica que pode se aplicar em malhas de célula cortada é o uso de camadas prismas. Esta técnica vem se provando ideal para aumentar a precisão da obtenção de forças e velocidades do escoamento em paredes, além de permitir a simulação de fenômenos como o desprendimento de camada limite. Se obtém essas camadas gerando uma subsuperfície na espessura desejada da camada limite, esta tem a própria malha gerada, ortogonal a subsuperfície e reduzida a um fator do tamanho alvo S , que tem suas faces projetadas na superfície original (SIEMENS, 2018).

Figura 2.6.2.3 - Exemplo de camada limite em malha triangular.



Fonte: Svensson, 2006.

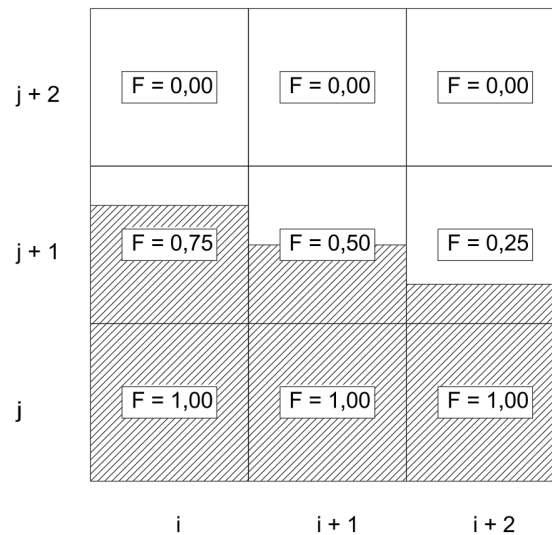
2.6.3. Considerações para múltiplas fases

Em casos de escoamentos com múltiplas fases, surge o problema de acompanhar a interface entre as fases para que as propriedades do fluido (e.g., densidade, viscosidade cinemática) corretas sejam utilizadas. Para isto, Hirt e Nichols (1981) criaram a metodologia VOF, onde eles definem que um campo de frações F se move junto com o escoamento do fluido (equação 2.6.3.1).

$$\frac{\partial F}{\partial t} + u \frac{\partial F}{\partial x} + v \frac{\partial F}{\partial y} + w \frac{\partial F}{\partial z} = 0 \quad \text{Eq. 2.6.3.1}$$

Em que F representa a fração do fluido no nó analisado, onde se $F = 1$, temos que a célula corresponde a uma célula do fluido e $F = 0$ corresponde a uma sem fluido, logo para valores entre zero e um, pode se assumir que existe uma camada limite e se aproximar sua posição, podendo assim se definir quais valores das variáveis do fluido se utilizam nas equações governantes (HIRT; NICHOLS, 1981).

Figura 2.6.3.1 - Representação do método VOF numa malha bidimensional onde a área hachurada representa preenchimento com fluido.



Fonte: Autor, 2024.

O método VOF foi um desenvolvimento essencial para CFD, por ser efetivo e ter baixo requerimento computacional por ser baseado em regiões e evitar os problemas de intersecção de superfícies. Porém, deve-se levar em consideração que VOF não reage

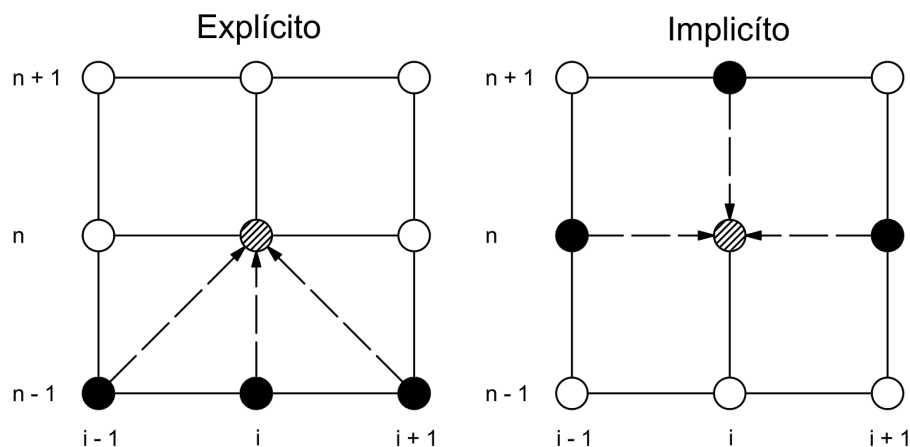
bem a fluidos com múltiplas pequenas fases e em escoamentos com presença de muitas bolhas ou respingos uma malha altamente fina é necessária (HIRT; NICHOLS, 1981).

2.6.4. Regime transiente

Em casos transientes onde a situação e características do escoamento são dependentes do tempo, é preciso saber como os pontos irão afetar e ser afetados por seus vizinhos entre passos no tempo n e seus vizinhos posicionais i . Tradicionalmente, o método direto explícito era utilizado, em que os valores em um ponto i num passo de tempo n eram calculados pelos seu valor e de seus vizinhos $i \pm 1$ no passo de tempo anterior $n - 1$. Porém, esse método resulta em problemas de convergência e requer passos no tempo muito pequenos, o que torna as soluções muito ineficientes e demoradas (VENKATAKRISHNAN e MAVRIPLIS, 1996).

Venkatakrishnan e Mavriplis (1996) sugerem uma método alternativo, implícito (ver figura 2.6.4.1), em que em cada iteração os resultados de um ponto i no passo de tempo n é informado por seu valor no passo de tempo $n + 1$ e de seus vizinhos $i \pm 1$. Apesar de ter um equacionamento mais complicado e logo um custo computacional maior por passo no tempo, o modelo implícito é intrinsecamente estável (*i.e.*, os erros não aumentam de forma descontrolada com o progresso da simulação) e permite passos no tempo maiores, ou seja, a simulação pode abranger um tempo físico maior de forma mais eficiente.

Figura 2.6.4.1 - Comparação de lógica do método explícito e implícito.



Fonte: Autor, 2024.

2.6.5. Análise de convergência

Durante a simulação, as equações como vistas em 2.6.1 irão retornar valores em cada iteração, se compararmos estes valores com os obtidos na próxima iteração, haverá um erro r entre os dois resultados. Podemos calcular o erro médio quadrado (ver equação 2.6.5.1) para todas as N células em cada equação e encontraremos o valor residual (Siemens, 2018).

$$R_{rms} = \sqrt{\frac{1}{N} \sum_i r^2} \quad \text{Eq. 2.6.3.1}$$

O residual é o valor que mostra a convergência das equações, ou seja, quanto menor o residual mais precisamente a simulação está encontrando os resultados. Numa situação ideal, os residuais são zero, porém isto nunca será o caso, pois sempre haverá erros devido a aproximação computacional (CEBECI et al., 2005).

2.7. Inteligência computacional

Inteligência artificial é um campo da computação que busca o desenvolvimento de algoritmos capazes de solucionar problemas complexos utilizando paralelos à inteligência biológica. Ou seja, busca ensinar uma máquina a “raciocinar” seguindo as mesmas lógicas que um ser vivo (ENGELBRECHT, 2007).

Dentro do campo da inteligência artificial, existe a Inteligência Computacional, que é como descreve Engelbrecht (2007, p. 4) “... O estudo de mecanismos adaptativos que permitem ou facilitam o comportamento inteligente em ambientes complexos e mutáveis.”, ou seja, um mecanismo computacional que consegue interpretar e relacionar um espaço de dados complexo com métodos probabilísticos. Pode-se ainda dividir IC em quatro paradigmas que representam as principais ferramentas utilizadas no desenvolvimento de modelos IC: Redes Neurais Artificiais (NN), Algoritmos Evolucionários (AE), inteligência de enxame e sistemas difusos. Em seu escopo este trabalho tratará de NN e AE.

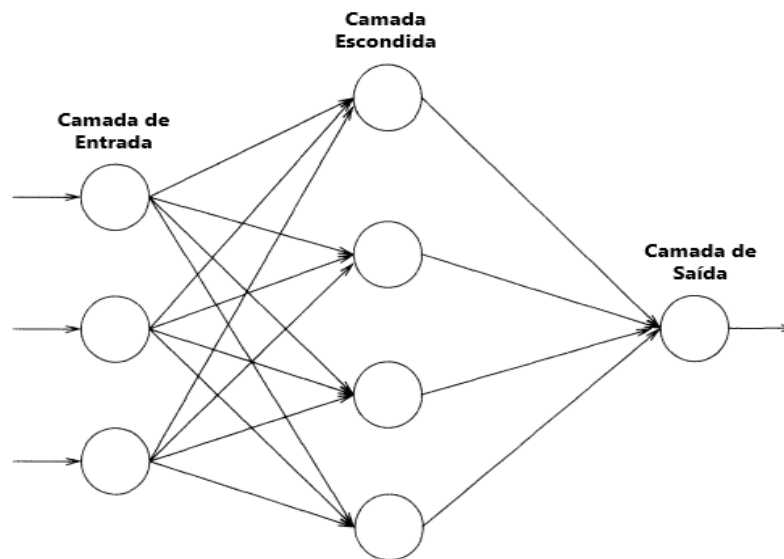
2.7.1. Redes neurais artificiais

NN funcionam buscando emular o processo de funcionamento básico de redes neurais biológicas (e.g., o cérebro humano) criando elementos individuais, chamados

neurônio, capazes de receber informações de entrada de uma base de dados ou de outros neurônios e subsequentemente transmitirem essas informações de forma modificada como uma saída, semelhante a um neurônio biológico (ENGELBRECHT, 2007).

Logo, a conexão desses neurônios artificiais em de diversas camadas forma uma NN. Essas camadas podem ser classificadas como camadas de entrada que recebem os dados originais, camadas escondidas que recebem os dados de outros neurônios, e camadas de saída, onde os resultados são retornados (ENGELBRECHT, 2007).

Figura 2.7.1 - Representação de rede neural artificial.



Fonte: ENGELBRECHT, 2007, p. 9.

A modificação que cada neurônio aplica nos dados de entrada varia para diferentes métodos de redes neurais, mas no geral, tratam-se de multiplicação e soma de matrizes. (ENGELBRECHT,2007)

É possível se dividir redes neurais em diversos tipos, como: redes de camada única, redes *feedforward* multicamadas, redes temporais, redes auto organizadas e as redes RBF (ENGELBRECHT,2007).

Explorando mais a fundo o conceito de redes RBF, ou seja, redes de função de base radial, são redes com uma camada de entrada, uma camada de saída e apenas uma camada escondida que deve ter uma quantidade N de neurônios igual ou maior que a dimensão dos valores de entrada. Seu princípio de funcionamento é a que resultados de saída próximos devem ter resultados de entrada próximos, consequentemente cada neurônio da camada escondida usará uma combinação dos valores de entrada e saída de

um conjunto aleatório de dados históricos como seu centroide, e sua função de ativação, chamada *kernel* (φ), será uma função pertencente a um conjunto de funções conhecidas como funções de base radial. Isto porque em tais funções, a variável está sempre representada como uma distância da entrada a ser analisadas até a entrada do seu centroide (ENGELBRECHT, 2007). Huang, Wang e Ma (2019) citam quatro diferentes tipos de funções de base radial.

Gaussiana

$$\varphi(||x - c||) = \exp\left(-\frac{||x-c||}{2\sigma^2}\right) \quad \text{Eq. 2.7.1.1}$$

Sigma refletido

$$\varphi(||x - c||) = \frac{1}{1 + \frac{||x-c||}{2\sigma^2}} \quad \text{Eq. 2.7.1.2}$$

Multi Quadratica
inversa

$$\varphi(||x - c||) = \frac{1}{\sqrt{(||x-c||)^2 + \sigma^2}} \quad \text{Eq. 2.7.1.3}$$

Multi Quadratica

$$\varphi(||x - c||) = \frac{1}{\varphi(||x - c||)^2 + \sigma^2} \quad \text{Eq. 2.7.1.4}$$

Então, os resultados de φ serão multiplicados por um peso ω_N que é encontrado resolvendo o sistema de equações para os quais os resultados dos dados históricos são comprovados, então temos que a rede RBF treinada nos fornecerá previsões com a equação 2.7.1.5 (ZHEN; GONG; WANG, 2022).

$$Previsão\ RBF = \sum_i^N \varphi_i(||x - c||) \cdot w_i \quad \text{Eq. 2.7.1.5}$$

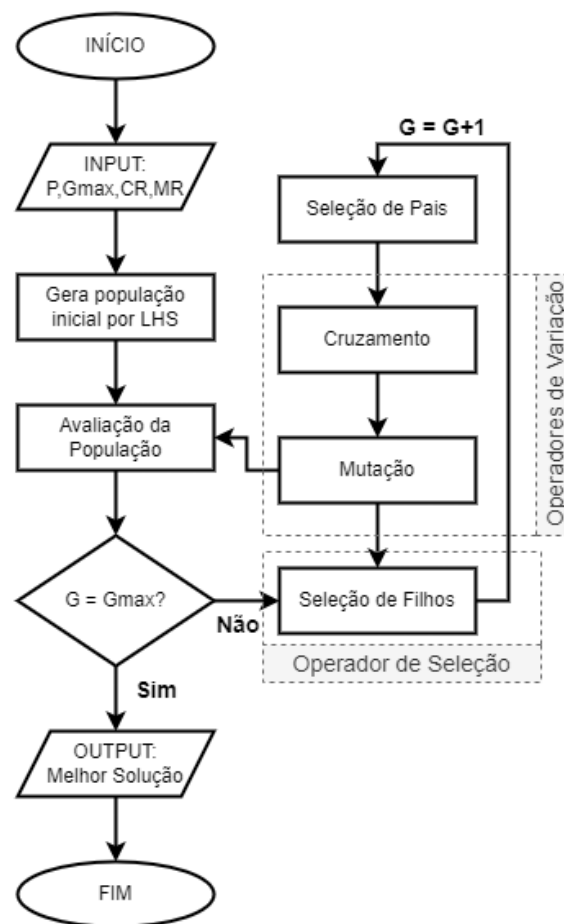
2.7.2. Algoritmos evolucionários

Algoritmos Evolucionários são uma classe de algoritmos usados em inteligência computacional inspirados no processo de seleção natural biológica. Se caracteriza por realizar buscas em um espaço de decisão de um problema e atribuir a diferentes soluções um valor de aptidão ou *fitness* (ENGELBRECHT, 2007).

Neste processo (ver figura 2.7.2.1), uma população de soluções candidatas é criada e operadores de reprodução, variação e seleção são aplicados iterativamente entre várias “gerações”. A reprodução, ou seleção de pais, fará um pareamento de duas ou

mais soluções candidatas para serem combinadas no estágio de cruzamento. Após isso, assim como na reprodução natural, existe uma possibilidade de mutação nas soluções-filhas, que contribui para a manutenção da diversidade na população. Em seguida, a aptidão das soluções-filhas é calculada por meio de uma função de avaliação apropriada para o problema, quantizando a qualidade das soluções criadas nessa geração com um valor denominado *fitness*. Por fim, há uma seleção de soluções, separando as soluções que permanecerão na população na próxima geração das que serão descartadas. É importante destacar que as etapas de avaliação e seleção das soluções candidatas precisam ser consistentes com o problema a ser solucionado (EIBEN; SMITH,2007).

Figura 2.7.2.1 - Fluxograma do funcionamento de um EA genérico.



Fonte: Autor, 2024.

Porém, em situações onde obter o valor de *fitness* não é possível com as entradas, ou computacionalmente muito caro (e.g., onde o *fitness* é a resistência ao avanço de uma geometria), é impossível calcular o *fitness* para toda a população em cada geração. Nesses casos, pode-se utilizar um EA baseado em dados (DDEA) onde ao

invés de se calcular o fitness diretamente, um modelo substituto baseado em dados conhecidos deve ser utilizado. Tal modelo deve ser capaz de extrair informação dos dados e usá-la para fornecer uma previsão aproximada satisfatória para o fitness das soluções candidatas. (ZHEN; GONG; WANG, 2022).

Modelos DDEA podem ser classificados em dois tipos *online* e *offline*, em modelos *online* após um determinado número de gerações, os indivíduos com melhor *fitness* previstos pelos modelos substitutos podem ser testados com valores reais para a solução encontrada. Nestes é possível obter durante a execução do código os resultados reais, assim possibilitando atualização contínua dos modelos substitutos dentro do espaço de soluções ótimas possíveis, que vem sendo reduzido durante a otimização (ZHEN; GONG; WANG, 2022).

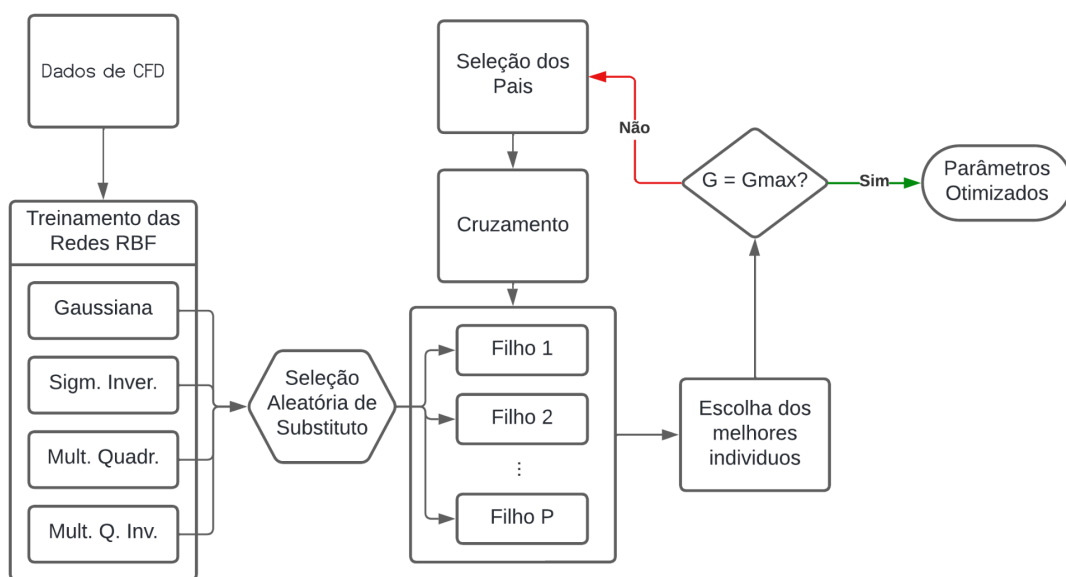
Já modelos *offline* são necessários quando é impossível ou inviável obter os valores reais para novas soluções encontradas pelo EA, e os únicos dados comprováveis utilizados durante a otimização são os iniciais (i.e., chamados de dados históricos na literatura) fornecidos para os modelos substitutos. Enquanto que esses modelos são muito úteis para otimizar esses problemas, são extremamente dependentes do tipo de modelo substituto utilizado, e podem ser prejudicados por regiões do espaço de soluções em que o modelo substituto não captura bem o problema (ZHEN; GONG; WANG, 2022).

Para contornar esse problema, Huang, Wang e Ma (2019) desenvolveram um DDEA de código aberto com quatro modelos substitutos usados intercambiavelmente por meios probabilísticos. De forma simplificada, os dados reais de entrada são utilizados para treinar quatro modelos de redes RBF, diferentes entre si apenas por seu kernel (eq.2.7.1.1-4).

Estes quatro modelos serão teoricamente mais efetivos em diferentes espaços de solução do problema, então, depois da reprodução e mutação, quando são gerados os indivíduos de cada geração, é calculado o valor de *fitness* de cada indivíduo com os quatro modelos substitutos e um dos resultados é escolhido aleatoriamente para ser usado no ranqueamento de cada indivíduo. Ou seja, em cada geração um indivíduo será avaliado por apenas um modelo, mas como há uma probabilidade uniforme de cada modelo ser sorteado, haverá uma mistura da “visão” dos quatro modelos no ranqueamento. Além disso, em outras gerações o modelo sorteado para a solução

candidata pode mudar entre gerações, permitindo uma variação no ranqueamento, e com isso favorecer a diversidade da população. Assim, a dependência em cada um modelo é reduzida e o EA busca otimizar o problema encontrando o indivíduo que os quatro modelos concordam em ter uma boa chance de ser o ótimo (ver figura 2.7.2.2). Para mais detalhes na metodologia o autor recomenda o trabalho original (HUANG; WANG; MA, 2019).

Figura 2.7.2.2 - Fluxograma do DDAE rankeado de forma probabilística.



Fonte: Autor, 2024.

3. MÉTODOLOGIA

Este capítulo busca detalhar o procedimento de modificação do bulbo, obtenção de propriedades geométricas, análise CFD e análise dos resultados por regressões e IC de forma que seja reproduzível pelo leitor.

3.1. Seleção de um modelo teste

A ITTC (2017) define que o uso de ferramentas como CFD para previsão de resistência ao avanço deve ter seu uso comprovado, de forma que a metodologia e aplicação do CFD sejam provados eficazes.

Isto é idealmente feito se comparando resultados obtidos por CFD com os resultados de testes com modelo reduzido, existem resultados prontos disponíveis virtualmente e na literatura (*e.g.*, navios container KRISO) que podem ser utilizados como comprovação (MOLLAND *et al.*, 2011).

Porém, com intuito de futuramente se comprovar com um modelo reduzido os valores obtidos por CFD, um modelo o qual possa ser futuramente construído de forma reduzida e testado pelos alunos da Universidade Federal de Pernambuco é ideal. Portanto deve-se escolher um modelo com bulbo, velocidade elevada suficiente para ter efeitos significativos de sua utilização.

O projeto de extensão Navicula Boat Design do curso de Engenharia Naval da Universidade Federal de Pernambuco constrói modelos de rebocadores de um metro de comprimento para competição de estudantes DUNA, onde os modelos reduzidos competem em uma série de provas para testar seu projeto e construção (DUNA, 2024).

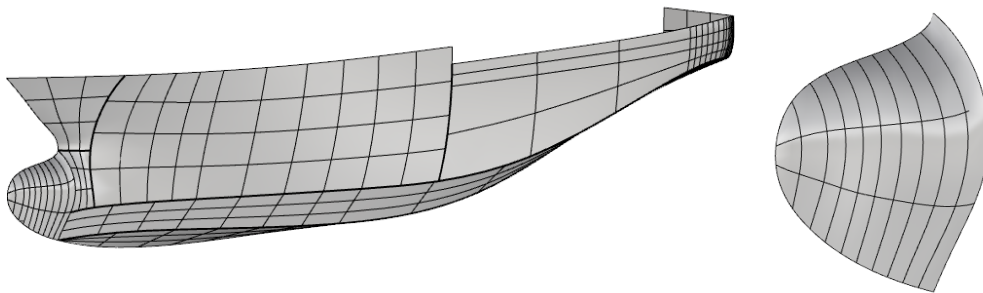
A embarcação Lampião, que foi utilizada na prova de 2017 (ver figura 3.1.1), teve um modelo teórico CAD projetado pelo docente Dr. Miguel Angel Celis Carbajal que continha uma proa bulbosa, denominada Superlampião.

Figura 3.1.1 - Modelo de rebocador Lampião na prova DUNA de 2017.



Fonte: NAVICULA BOAT DESIGN, 2017

Figura 3.1.2 - Modelo CAD embarcação Superlampião e seu bulbo recortado.



Fonte: Autor, 2024.

3.2. Modificações do bulbo

Com a embarcação definida, se utilizará a metodologia de Sederberg e Parry (1986) para modificar a o bulbo. Com a geometria no formato de superfície NURBS no *software* de CAD Rhinoceros, se separa a superfície do bulbo da do casco e a divide na linha de centro da embarcação, pois fazendo as modificações em apenas um lado, é simples espelhar para garantir simetria, e se salva o bulbo como arquivo do Rhinoceros.

Este arquivo do Rhinoceros pode ser convertido para um arquivo tipo JSON com a ferramenta desenvolvida por Bingol e Krishnamurthy (2019) e publicamente distribuída pela plataforma Github.

Bingol e Krishnamurthy (2019) desenvolvem uma livreria para a linguagem de programação Python que permite criação, avaliação e modificação de geometrias

NURBS. Então escreve-se um código em Python para a criação das deformações desejadas.

Exporta-se a geometria do bulbo em JSON com um código do Python e se cria uma *lattice* nas dimensões do bulbo com $(l, m, n) = (3, 3, 3)$ e se exclui o ponto interior para um total de 26 pontos, as coordenadas dos pontos de controle da *lattice* e dos pontos de controle da superfície do bulbo são transformados nas coordenadas (s, t, u) .

A superfície de metade do bulbo que está sendo tratada se conectará com a metade do casco e com sua simetria espelhada. Se objetiva garantir continuidade c^1 nessas conexões, enquanto que é possível apenas fixar duas camadas de pontos da extremidade que se conecta ao casco, fazer isto com os pontos de simetria impediria a deformação no plano de simetria, que é importante para nossa análise, então fixamos apenas na direcção transversal os pontos das duas camadas do plano de simetria.

Apesar de estes garantirem a continuidade, é possível que formas inválidas (*e.g.*, superfícies que se interseccionam o restante da proa, que atravessam o plano de simetria) ou com curvatura muito brusca fossem geradas. Enquanto que é possível que as curvaturas muito bruscas fossem analisadas e descartadas pela nossa análise, isso gastaria muito poder computacional com análises CFD não úteis. Então, se criam critérios para a limitação máxima de cada ponto de controle da *lattice*.

Para começar, universalmente a deformação máxima do ponto de controle da *lattice* em cada direcção (s, t, u) será de 10% (dez por cento) da dimensão da *lattice* nessa direcção, isto se faz pois não se deseja expressivas mudanças de volume do bulbo para evitar divergência da realidade na análise CFD devido a mudanças de trim do modelo. Então, as deformações $\Upsilon_{s,t,u}$ em cada direcção serão limitadas como nas equações 3.2.1 a 3.2.3 para que as deformações sejam contínuas sem intersecção e seguem um formato tradicional de bulbo.

$$- 0,1 s^2 \leq \Upsilon_s \leq 0,1 \quad \text{Eq. 3.2.1}$$

$$- 0,1 t^2 \leq \Upsilon_t \leq 0,1 \quad \text{Eq. 3.2.2}$$

$$-0,1 \left[\log_{1,2}(0,9s) + 0,1 \right] \leq Y_u$$

Eq. 3.2.3

$$Y_u \leq 0,1 \left[\log_{1,2}(0,9s) + 0,1 \right]$$

Utiliza-se o método de amostragem por hipercubo latino para que se distribuam $Y_{s,t,u}$ de cada um dos 26 (vinte e seis) pontos de controle. Ou seja, uma amostragem por hipercubo latino onde $\mathbb{R}^{78}$ para se obter 200 (duzentos) parâmetros de variação do bulbo do bulbo.

Finalmente, se aplicam as deformações nas posições dos pontos de controle da *lattice* e se utiliza a equação de Sederberg e Parry (1986) (Equação 2.4.2) para obter-se novas superfícies do bulbo.

As geometrias são exportadas em formato de JSON e convertidos para arquivos do Rhinoceros com a ferramenta fornecida por Bingol e Krishnamurthy (2019) para análise CFD e as variações $Y_{s,t,u}$ dos pontos de controle são exportadas com arquivo separado por tabulações como parâmetros para alimentar nossos algoritmos de IC.

Essas deformações são exportadas para JSON e transformadas em arquivo do Rhinoceros com a ferramenta fornecida por Bingol e Krishnamurthy (2019).

3.2.1. Obtenção dos parâmetros de comparação do bulbo

As deformações exportadas em JSON são importadas em outro código Python que vai calcular os parâmetros de comparação, neste trabalho serão analisados apenas os parâmetros de comprimento e profundidade (Equações 2.3.2 e 2.3.3).

Para as as curvaturas gaussiana e média, como o equacionamento descrito no capítulo 2.5 encontra esses valores em um ponto de uma superfície, calculamos para todos pontos de controle, assim, como para nós é interessante a magnitude das curvaturas, e não seu sinal, calculamos a média dos valores absolutos.

3.3. Análise CFD

Para se aplicar CFD, pode-se programar os modelos discutidos em 2.6 diretamente em qualquer linguagem de programação desejada, isto traz a vantagem de

poder analisar diretamente e rodar testes comprovando o funcionamento normal do sistema, além de que seria possível integrar diretamente com o código desenvolvido em 3.2 e utilizar teorias de computação evolucionária para otimizar iterativamente a geometria.

Porém, a complexidade de escrever tal algoritmo é considerável, e não seria possível realizar a tempo hábil para esse trabalho. Alternativamente, existe a possibilidade de utilizar um *software* comercial que tenha o equacionamento já definido, onde se precisa apenas inserir as geometrias e definir os parâmetros da simulação. Enquanto que estes *softwares* poupam tempo para montar o sistema, seu código não aberto impede de verificar o equacionamento.

3.3.1. STAR CCM+

Se utiliza o software STAR-CCM+ desenvolvido pela Siemens Digital Industries Software em sua versão v11.06 para rodar as análises CFD dos casos. O STAR-CCM+ é um *software* projetado para CFD com múltiplos equacionamentos físicos já inclusos e interface de usuário em lógica de árvore.

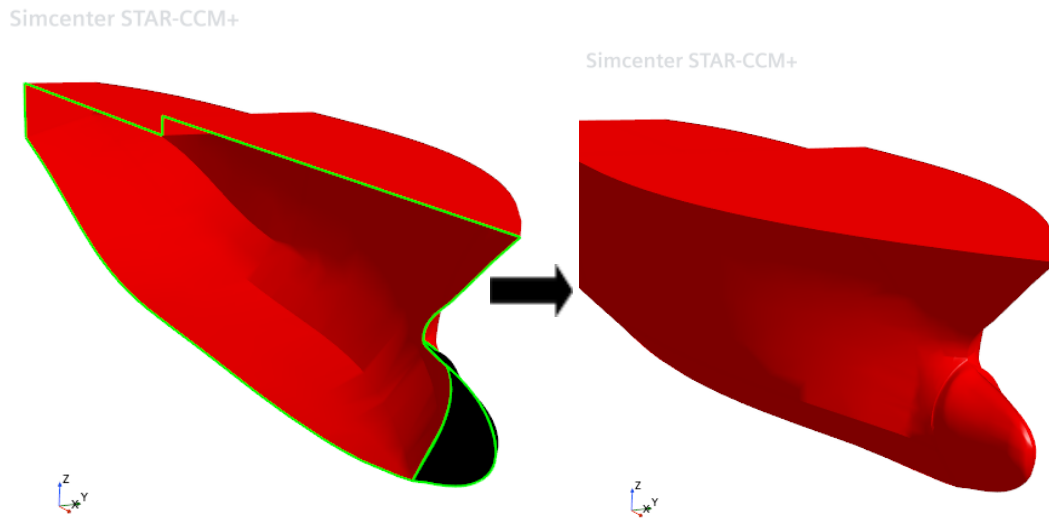
Primeiramente, deve-se importar as superfícies do modelo para o software, tem-se duas superfícies necessárias, o bulbo separado do casco, no qual foram criadas as deformações, e o casco com o recorte para o bulbo. Elas podem ser importadas diretamente dos arquivos do Rhinoceros com o módulo 3D-CAD e unidas com a ferramenta *sew surfaces*.

Porém se tem apenas as superfícies de um bordo da embarcação, para futuramente criarmos o domínio do fluido, a transforma num sólido fechando-a na linha de centro com a ferramenta *fill faces* e após espelhar uma cópia no eixo y, deve-se fazer uma operação de união booleana entre o sólido copiado e o original que é então transformada em parte de geometria na árvore principal da simulação (figura 3.3.1.1).

Cria-se um bloco cujas dimensões de 5,05 x 1,04 x 1,86 metros, suficientemente maiores que as dimensões do modelo (1,00 x 0,32 x 0,20 metros) para que os resultados não sejam afetados por efeitos de parede (e.g., reflexão de ondas nas paredes do domínio). Faz-se uma operação booleana de subtração onde se remove o volume do casco do bloco, o resultado chamado de *subtract* será a base do domínio físico do

fluido. Este domínio é definido criando uma região para o *subtract* em que cada superfície (*i.e.*, as faces do bloco originais e o recorte do casco) são regiões de contorno.

Figura 3.3.1.1 - Superfícies de um bordo do bulbo e do casco importadas e transformadas em parte única fechada.



Fonte: Autor, 2024.

As superfícies a vante e a ré do bloco são definidas como *velocity inlet* e *pressure outlet*, ou seja, a embarcação avançando na verdade se tratará do escoamento se movendo, entrando a uma velocidade fixa pela frente do bloco e saindo pelo fundo sempre quando ultrapassa a pressão inicial. As superfícies laterais do bloco são definidas como planos de simetria, ou seja, a simulação irá assumir que tudo que está acontecendo dentro do domínio naquela superfície se repete de forma inversa por fora, isto é útil para diminuir efeitos de parede. Finalmente, a única parede sólida do sistema é a superfície do casco.

Então, cria-se uma operação de criação de malha automatizada para o *subtract*, onde os modelos criadores de malha principais são, *Trimmed Cell Mesher* e *Prism Layer Mesher* como vistos em 2.6.2. Também se seleciona os modelos de melhoria da malha do STAR-CCM+ *Surface Remesher* e *Automatic Surface Repair*. As configurações podem ser vistas no APÊNDICE 1.

São incluídos quatro controles personalizados, o primeiro é um controle de superfície em que se diminui o tamanho alvo da superfície do casco da embarcação, deixando o casco como um todo com uma malha mais fina. O segundo e terceiro

respectivamente são controles volumétricos que aumentam a densidade da malha na camada limite, para bem representar a geração e padrão de ondas. Finalmente o quarto controle personalizado aumenta ainda mais a densidade da malha na região do bulbo e aumenta a quantidade de camadas prismáticas.

Agora é preciso selecionar as equações governantes que o *software* utilizará para executar a simulação, como base utilizaremos o modelo RANS com turbulência $k-\epsilon$ Realizável e num escoamento multifásico com modelo VOF (ver capítulo 2.6.1).

Configura-se o escoamento multifásico com uma chamada onda VOF plana, ou seja, define-se a condição inicial como um mar plano com escoamento uniforme em que o fluido leve é ar atmosférico com densidade de 1.18415 kg/m^3 e viscosidade dinâmica de $1.85508\text{E-}5 \text{ Pa-s}$, e o fluido pesado como água doce de densidade 997.561 kg/m^3 e viscosidade dinâmica de $8.8871\text{E-}4 \text{ Pa-s}$.

O regime transiente é definido como implícito com um passo temporal definido em função do comprimento e velocidade do modelo.

$$\Delta t = \frac{L}{V 250} \quad \text{Eq. 3.2.1}$$

Com o equacionamento físico definido, se adiciona se condiciona as condições de parada da simulação, que serão o máximo de cinco iterações para cada passo no tempo implícito e um determinado tempo físico que será explicado em 3.3.2.

Com esses parâmetros definidos, resta acionar o monitoramento de forças agindo na superfície do casco na direção longitudinal e vertical, que representaram a resistência ao avanço e o empuxo, e montar o sistema que substituirá e rodará a análise para todas as deformações do bulbo.

3.3.2. Automatização da análise - Macros

O STAR-CCM+ possui a função de rodar um conjunto de comandos de forma automatizada usando roteiros em java (*i.e.*, linguagem de programação orientada a objetos) os quais podem ser criados manualmente, ou com a função de gravar na própria interface de usuário. Então, utilizando a função de gravação se grava no *software* um macro que executa uma vez o laço descrito na figura 3.3.2.1.

Com o macro escrito automaticamente pelo STAR-CCM+, é trivial editá-lo para ser contido dentro do laço: *for (int i = 0; i<200 ; i++) {...}* (i.e., laço do java que será executado para cada valor de *i* entre 0 e 199) onde *i* é o número da deformação do bulbo a ser analisada. Para importar a geometria correta, basta modificar a linha:

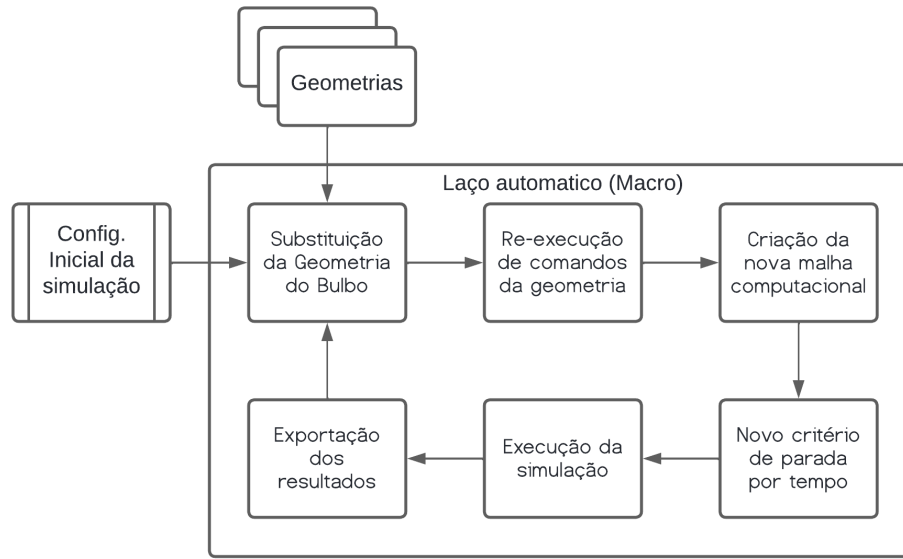
```
“modelReImportCommands_0.loadFileForReplaceBody(importCadFileFeature_0,resolvePath("C:\\localização do arquivo\\Deforma "+i+".3dm"));"
```

Em que dependendo do valor de *i* do laço, um arquivo de bulbo do Rhinoceros diferente será carregado, e garante equivalência das faces e bordas entre bulbos para garantir bom funcionamento das transformações de superfície. Modificações semelhantes são feitas nos comandos de exportação dos resultados para garantir que os resultados de laços antigos não sejam apagados pelo novo. Finalmente, antes do macro rodar a simulação, se define o tempo físico máximo como critério de parada com:

```
“physicalTimeStoppingCriterion_0.setMaximumTime().setValueAndUnits( 5.0 + i * 4.0 , units_0)”
```

Assim, para cada laço, será adicionado quatro segundos ao tempo máximo de parada, ou seja, cada modificação do bulbo será testado por quatro segundos antes de atingir o limite de tempo físico, parar a simulação, exportar as tabelas das forças e residuais em função do passo de tempo e passar para o próximo laço. Para os resultados de resistência ao avanço e empuxo, se fará uma média do último segundo da simulação de cada bulbo, de forma a utilizar os resultados com melhor convergência de cada caso.

Figura 3.3.2.1 - Laço da simulação para múltiplos bulbos.



Fonte: Autor, 2024.

3.4. Definição de relações

Os dados de resistência ao avanço exportados da análise CFD foram importados para um código de Python que comparava os parâmetros do bulbo encontrados com as metodologias listadas em 3.2.1, mais especificamente C_{LPR} , C_{ZB} e as médias dos valores absolutos da curvatura gaussiana e curvatura média de cada bulbo. Se procura relações ou tendências entre a resistência ao avanço e os parâmetros geométricos com uma regressão linear simples (ver equação 3.4.1), onde R^2 será a variância dos resultados a regressão.

$$Y(x) = Y_0 + C x \quad \text{Eq. 3.4.1}$$

3.5. Análises por Inteligência Computacional

Com os resultados exportados da análise CFD, podemos aplicar o código aberto de um DDAE rankeado probabilístico disponibilizado por Huang, Wang e Ma (2019). As únicas modificações feitas foram a inclusão de uma função para importar os resultados com seus parâmetros e um laço que faz uma simples análise de sensibilidade (*i.e.*, variação das constantes das NN para testar sua precisão). Para isto foram separados 35 casos de validação, um laço que treinava as redes RBF e depois compara os

resultados previstos para os casos de validação com os reais e calcula o erro médio quadrado.

Após a análise de resultados (ver capítulo 4.4) as redes RBF foram treinadas com os dados, configuradas para terem 20 x 78 (i.e., 20 vezes a dimensão dos inputs) e o algoritmo evolucionário teve uma população de 200 indivíduos e duração de 150 gerações.

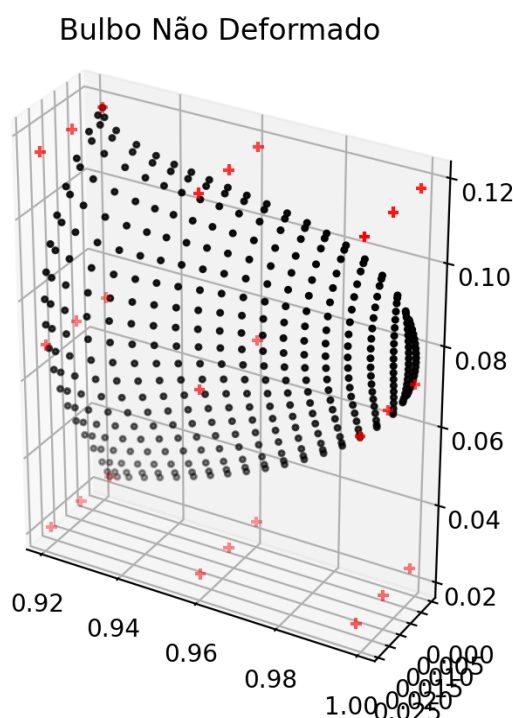
4. RESULTADOS

Neste capítulo será apresentado o que foi obtido seguindo a metodologia como descrito no capítulo 3.

4.1. Modificações do bulbo

A superfície NURBS do bulbo do Superlâmpião, com 400 pontos de controle em esquema 20 x 20 foi importada para o código em Python descrito em 3.2 e teve a *lattice* de transformação FFD gerada em suas extremidades (ver figura 4.1.1).

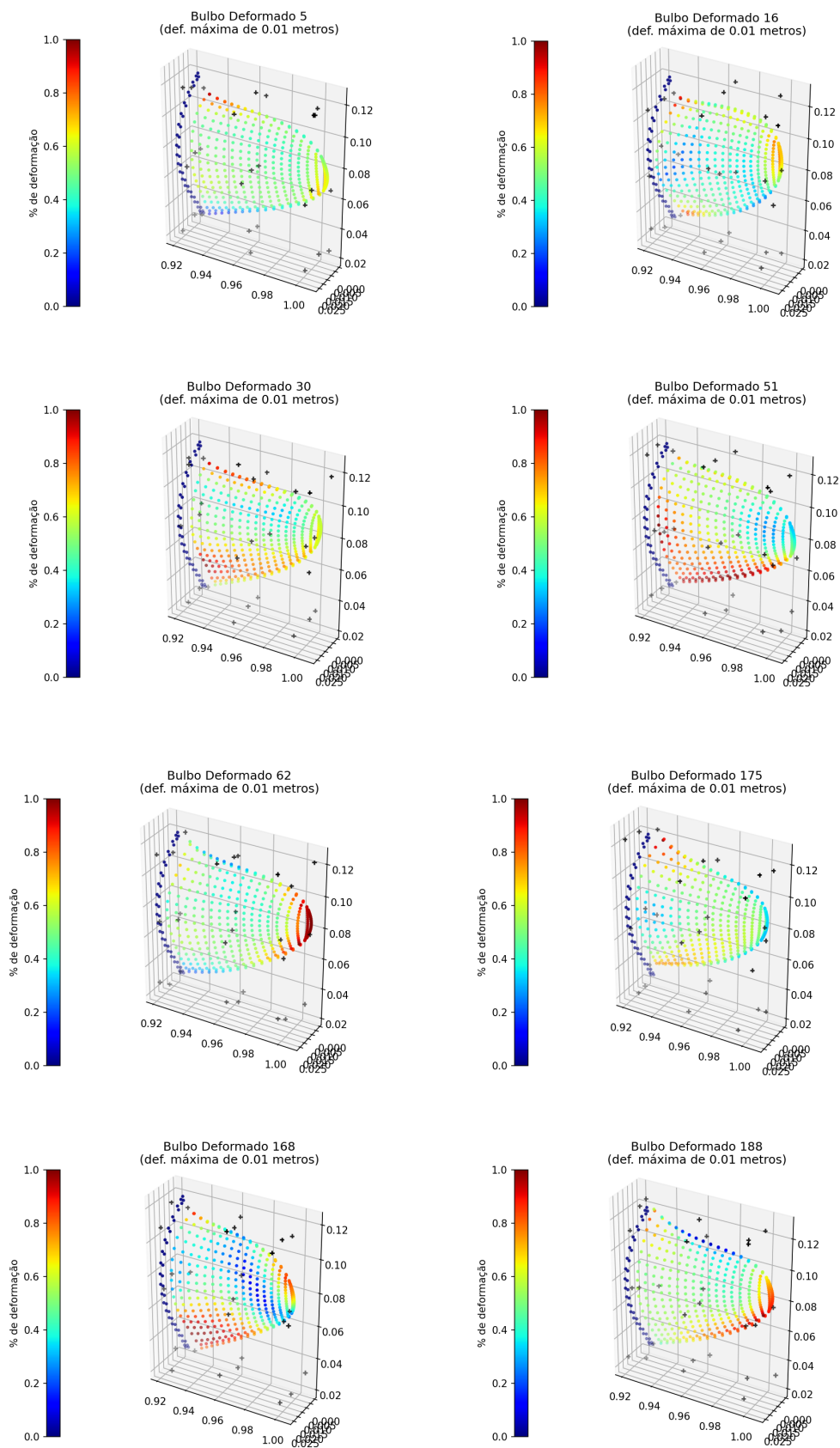
Figura 4.1.1 - Pontos de controle do bulbo original (preto) e pontos de controle da malha computacional (vermelho).



Fonte: Autor, 2024.

São com a aplicação de amostragem por hipercubo latino são geradas 200 (duzentas) matrizes de deformação 26 x 3 com as posições modificadas dos pontos de controle da *lattice* e são utilizadas para modificar os pontos de controle do bulbo original por meio de FFD como descrito em 2.3 se obtendo 200 (duzentas) novas versões dos 400 (quatrocentos) pontos de controle do bulbo exportadas como *Deform 0* até *Deform 199*. A figura 4.1.2 mostra algumas deformações com colorização em cada ponto representando a variação da posição original, os pontos modificados da *lattice* e a linha de conexão com o resto do casco não modificada.

Figura 4.1.2 - Pontos de controle do bulbo deformados.



Fonte: Autor, 2024.

Depois, exportando a geometria para o código de obtenção das propriedades geométricas de cada bulbo (capítulo 3.2) e obtemos o descrito na tabela 4.1.1. Algumas curvaturas gaussianas e médias podem ser observadas na figura 4.1.3.

Tabela 4.1.1 - Propriedades geométricas dos bulbos modificados.

Característica	Mínimo	Máximo	Média	σ_p	$\sigma_{p\%}$
C_{LPR}	0,0717	0,0851	0,0786	0,0026	3,27%
C_{ZB}	0,5953	0,7312	0,6631	0,0284	4,28%
Cr. Gaussiana Magnitude Média	0,0005	6,7612	0,0535	0,5071	947,28%
Cr. Média Magnitude Média	0,1025	6196,2082	35,1926	437,6790	1243,67%

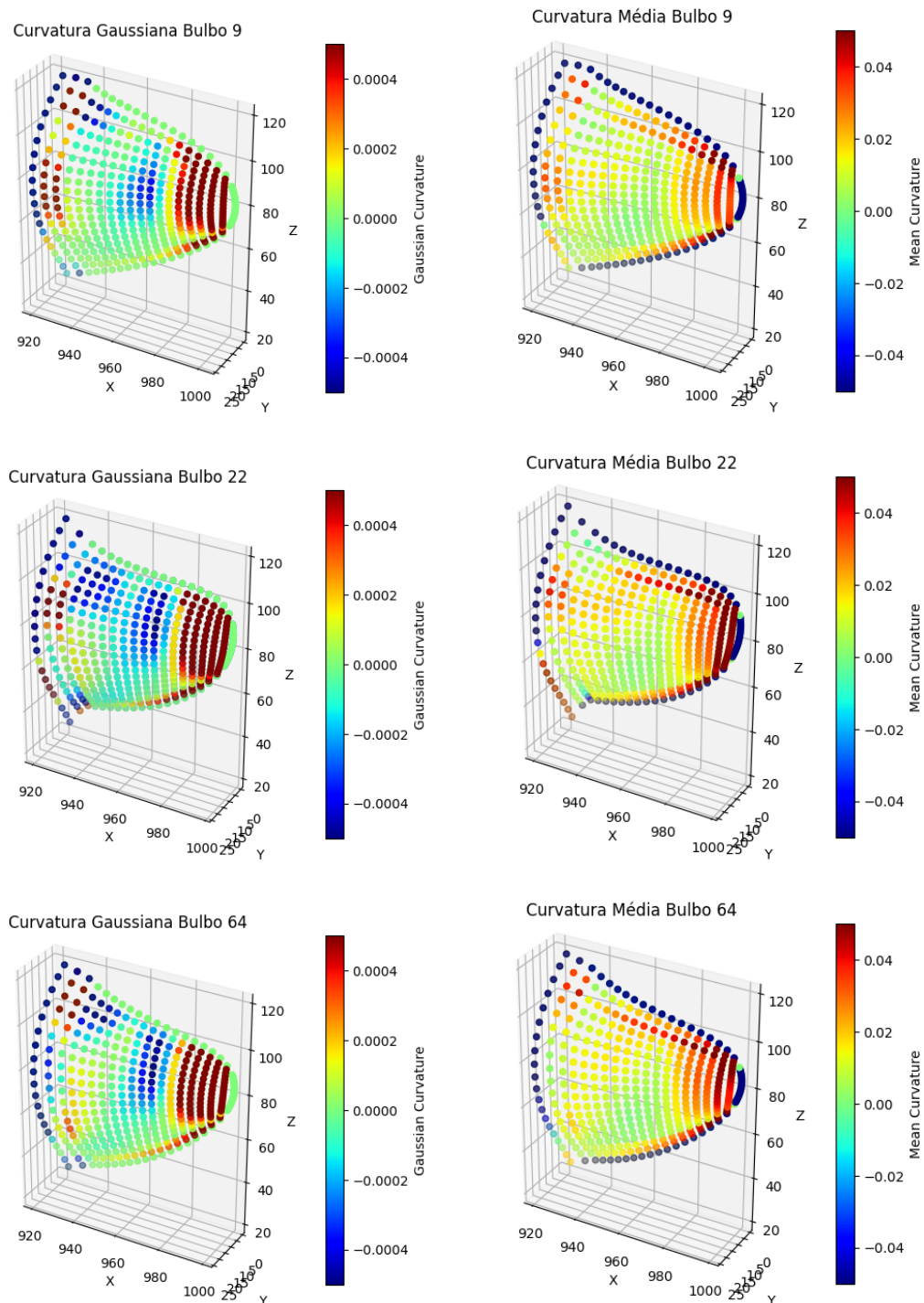
Fonte: Autor, 2024.

Se excluirmos outliers das curvaturas com limite de 5 x IQR (*i.e.*, diferença entre primeiro e terceiro quartil de um grupo de dados) temos:

Tabela 4.1.2 - Propriedades das curvaturas dos bulbos modificados com exclusão de outliers.

Característica	Mínimo	Máximo	Média	σ_p	$\sigma_{p\%}$	Outliers
Cr. Gaussiana Magnitude Média	0,0005	0,0015	0,0008	0,0002	23,61%	12
Cr. Média Magnitude Média	0,1025	1,3281	0,2957	0,2218	75,01%	21

Figura 4.1.3 - Curvaturas gaussiana (esquerda) e média (direita) de bulbos deformados.



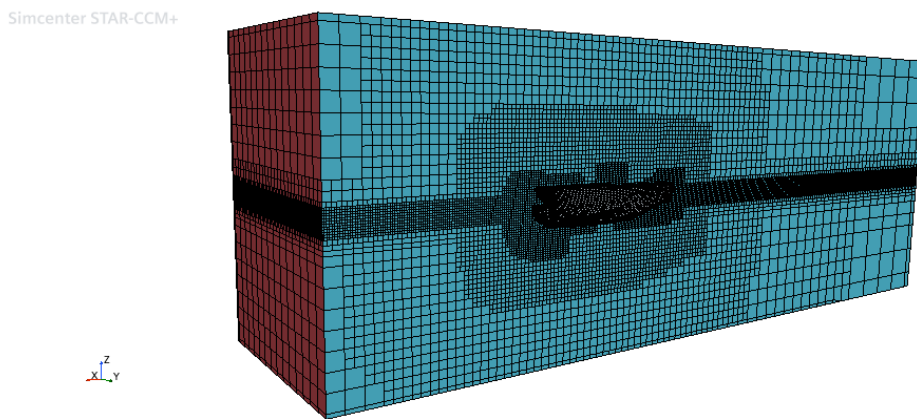
Fonte: Autor, 2024.

4.2. Análise CFD

Com os arquivos da geometria do bulbo exportados e sua pasta referência no macro descrito no capítulo 3.3.2, pode se iniciar a execução do código. As malhas resultantes tinham em torno de 714.000 elementos e a simulação é alocada para 8 processos paralelos com processadores Intel i5-13450HX.

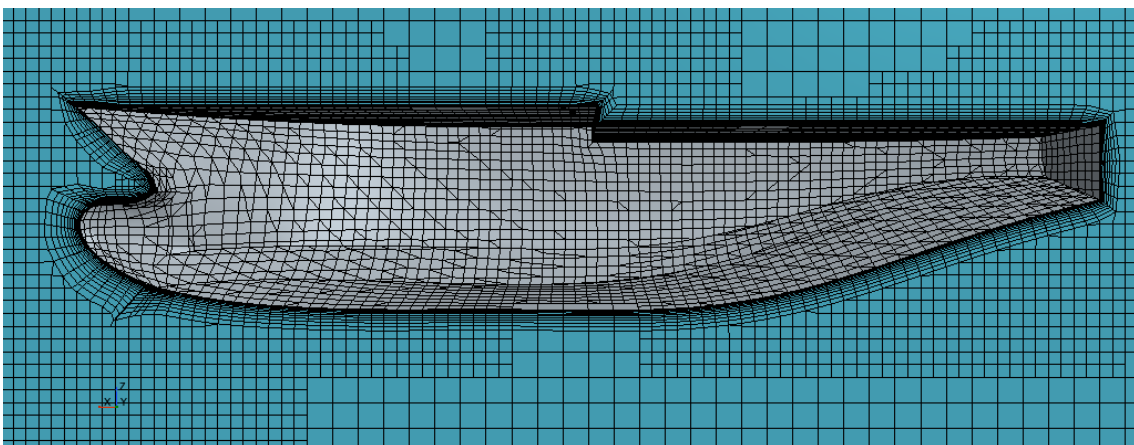
Com em média 714,286 milissegundos por iteração e 17 (dezessete) segundos por geração de malha computacional, o software rodou um total de 200 (duzentos) casos de proa bulbosa em 883.125 iterações cerca de 160 horas com tempo físico de simulação decorrido de 13,35 minutos.

Figura 4.2.1 - Vista geral da malha do domínio.



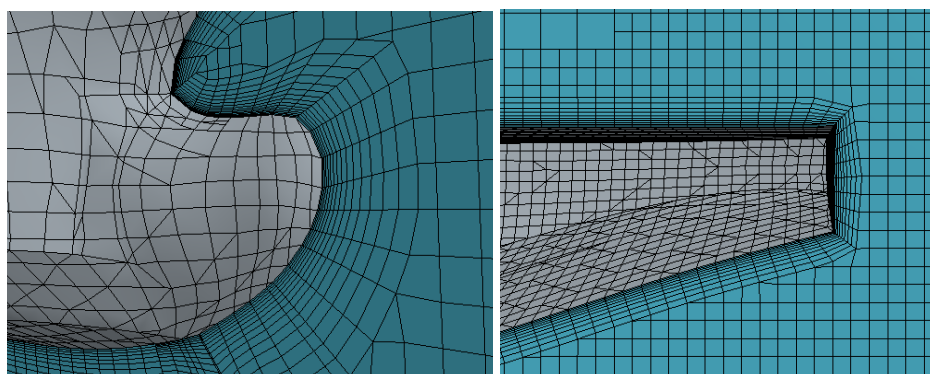
Fonte: Autor, 2024.

Figura 4.2.2 - Vista geral da malha em torno do casco.



Fonte: Autor, 2024.

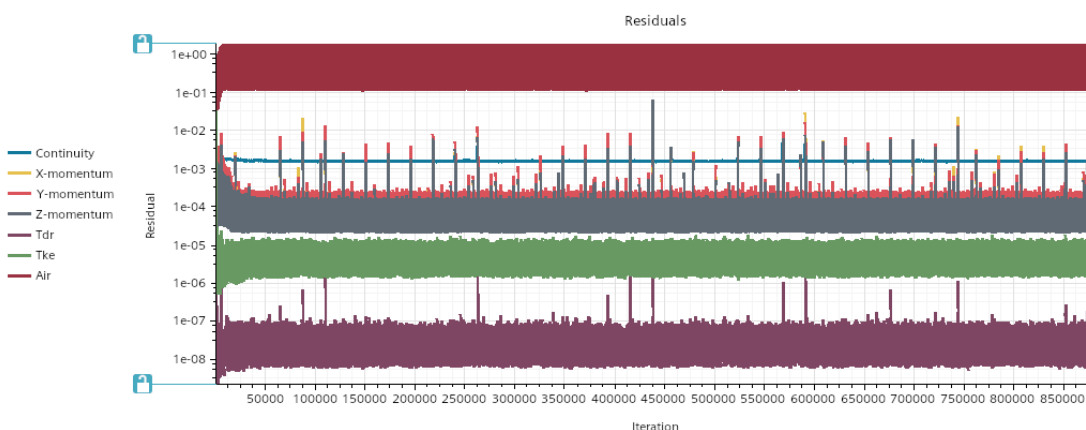
Figura 4.2.3 - Vista geral da malha próxima ao bulbo e na popa.



Fonte: Autor, 2024.

Os residuais vistos na figura 4.2.4 mostram boa convergência com os residuais da continuidade estáveis em torno de $1,5\text{e-}03$, os residuais dos momentos nas três direções cartesianas em torno de $1,0\text{e-}04$ e os residuais de dissipação e energia cinética da turbulência respectivamente em torno de $1,5\text{e-}06$ e $1,3\text{e-}08$. Também se observa claramente o aumento dos valores dos residuais nos pontos em que a geometria do bulbo é substituída, e que ainda assim volta a convergir rapidamente.

Figura 4.2.4 - Residuais da simulação.



Fonte: Autor, 2024.

A componente na direção x de todas as geometrias ao longo do tempo pode ser visualizada na figura 4.2.5, onde se observa um período de não convergência no primeiro minuto de tempo físico e os pontos de não convergência devido a substituição da geometria. Observando com mais detalhe o recorte no tempo da figura 4.2.6 se observa que a força na direção x inicialmente se comporta como uma senoide, na qual a modificação da geometria tem efeito mínimo se comparado com a amplitude inerente à

simulação. Esta amplitude reduz ao longo do tempo físico até aproximadamente 60 segundos onde se estabiliza e se observa que a variação das geometrias é dominante, com variações bruscas na mudança de bulbos e seguidas pela nova geometria se estabilizando.

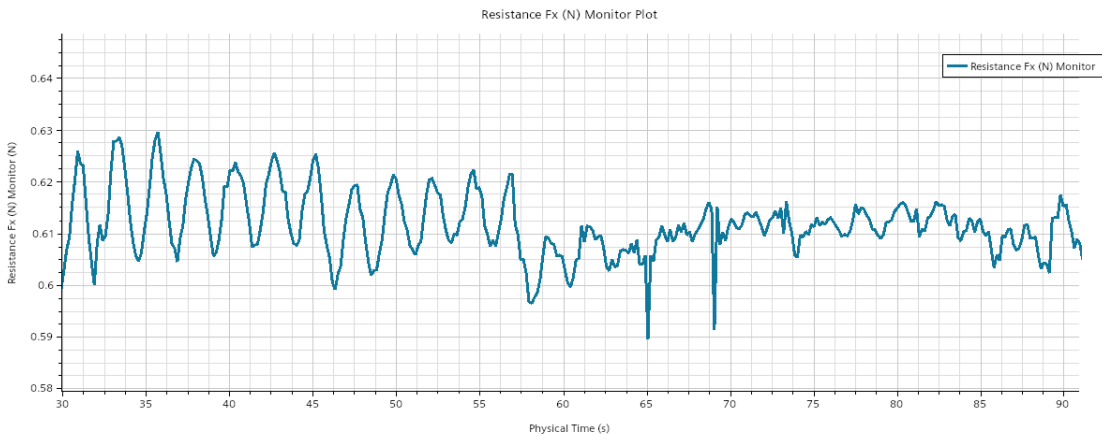
Portanto, excluimos as modificações dos bulbos antes de 65 segundos, um total de 16 (dezesesseis) casos. Como nosso domínio encapsula apenas um bordo da embarcação, a resistência ao avanço do modelo será o dobro desta componente em x . Então, com a média do último segundo físico de cada caso considerado (ver capítulo 3.3.1) obtemos os dados da tabela 4.2.1.

Figura 4.2.5 - Componente da força na direção x do modelo ao longo do tempo com os bulbos modificados a cada quatro segundos.



Fonte: Autor, 2024.

Figura 4.2.6 - Componente da força na direção x entre 30 e 90 segundos.



Fonte: Autor, 2024.

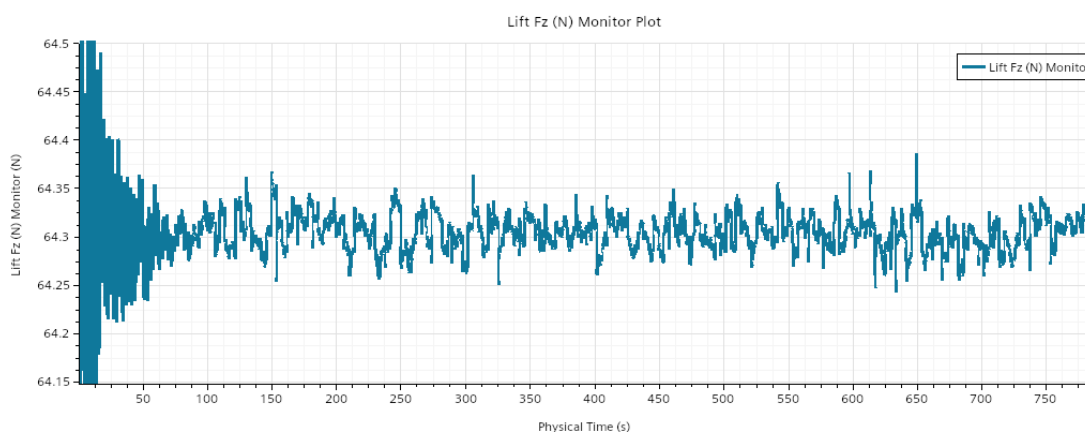
Tabela 4.2.1 - Resultados de CFD da resistência ao avanço.

Resistência ao Avanço (Fx)	Valor (N)	Diferença do origin.	Diferença da Média
Bulbo Original	1,2307	-	+ 0,99%
Máxima (Bulbo 74)	1,2412	+ 0,86 %	+ 1,85 %
Mínima (Bulbo 66)	1,1933	- 3,04 %	- 2,08 %
Média Deformações	1,2186	- 0,98 %	-
Desvio Padrão	0,0080	% média:	0,66 %

Fonte: Autor, 2024.

Semelhante a componente na direção x , observamos a componente na direção z na figura 4.2.7, que mostra o mesmo comportamento periódico até os 60 segundos. Novamente excluindo os 16 primeiros casos e multiplicando os resultados em duas vezes, obtemos os resultados da tabela 4.2.2.

Figura 4.2.7 - Componente da força na direção z do modelo ao longo do tempo com os bulbos modificados a cada quatro segundos.



Fonte: Autor, 2024.

Uma observação em especial aos bulbos de máxima e mínima resistência, respectivamente bulbos 74 e 66, e de máximo e mínimo empuxo, respectivamente 37 e 57, está na tabela 4.2.3 e nas figuras 4.2.7 a 4.2.10.

Tabela 4.2.2 - Resultados de CFD do empuxo.

Empuxo (Fz)	Valor (N)	Diferença do origin.	Diferença da Média
Bulbo Original	128,5266	-	- 0,06 %
Máximo (Caso 37)	128,7015	+ 0,14 %	+ 0,07 %
Mínimo (Caso 57)	128,5233	- 0,00 %	- 0,07 %
Média Deformações	128,6070	+ 0,06 %	-
Desvio Padrão	0,329	% média:	0,03 %

Fonte: Autor, 2024.

Se observa que o bulbo 74 com maior resistência ao avanço possui próximo a conexão com o restante do casco na parte superior uma geometria que cria uma curvatura mais brusca, e no bulbo 66 com menor resistência se observa um volume maior e curvatura mais suave na conexão com o casco. No bulbo com menor empuxo se observa uma diminuição em suas dimensões e volume quando comparado ao de maior empuxo, principalmente na parte inferior e sua conexão com o casco.

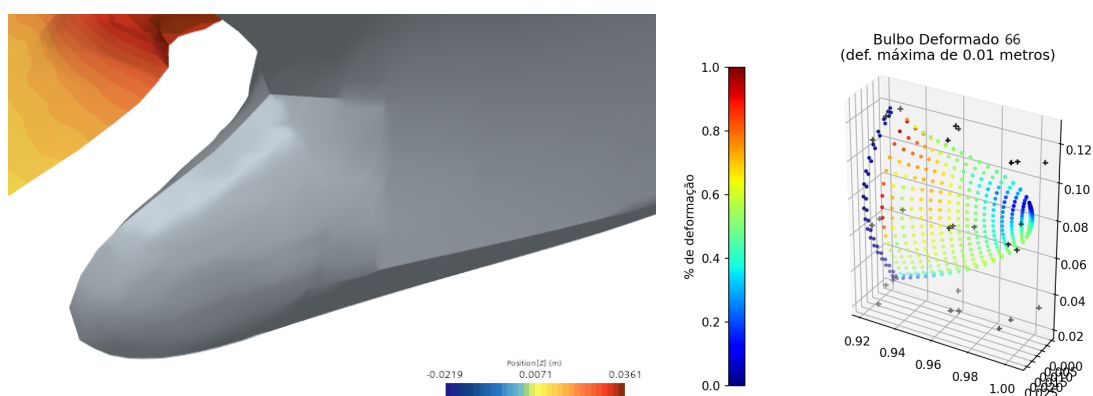
Tabela 4.2.3 - Resultados de CFD dos bulbos com resistência ao avanço máxima (74) e mínima (66) e empuxo máximo (37) e mínimo (57).

Caso	Tempo Físico		Resistência ao Avanço		Empuxo	
	Inicial (s)	Final (s)	Valor	D. Pad*	Valor	D. Pad*
74	297	301	1,2412	0,26 %	128,5233	0,00 %
66	265	269	1,1933	0,16 %	128,6456	0,00 %
57	229	233	1,2284	0,22 %	128,5233	0,01 %
37	149	153	1,2159	0,06 %	128,7015	0,00 %

* Desvio padrão da resistência ao longo do tempo físico, último segundo.

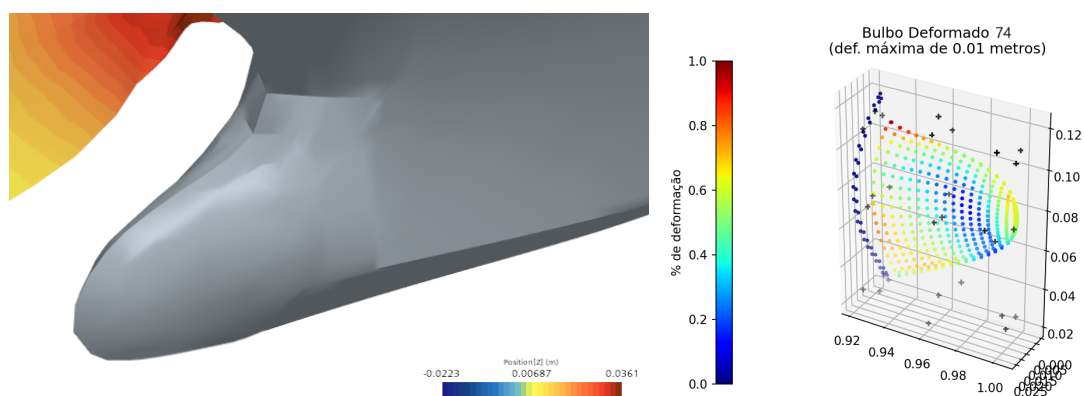
Fonte: Autor, 2024.

Figura 4.2.7 - Bulbo 66 com menor resistência ao avanço.



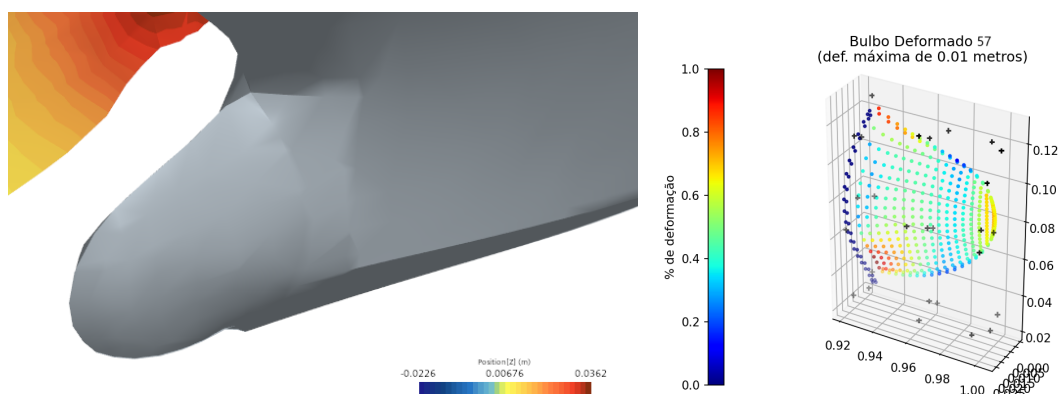
Fonte: Autor, 2024.

Figura 4.2.8 - Bulbo 74 com maior resistência ao avanço



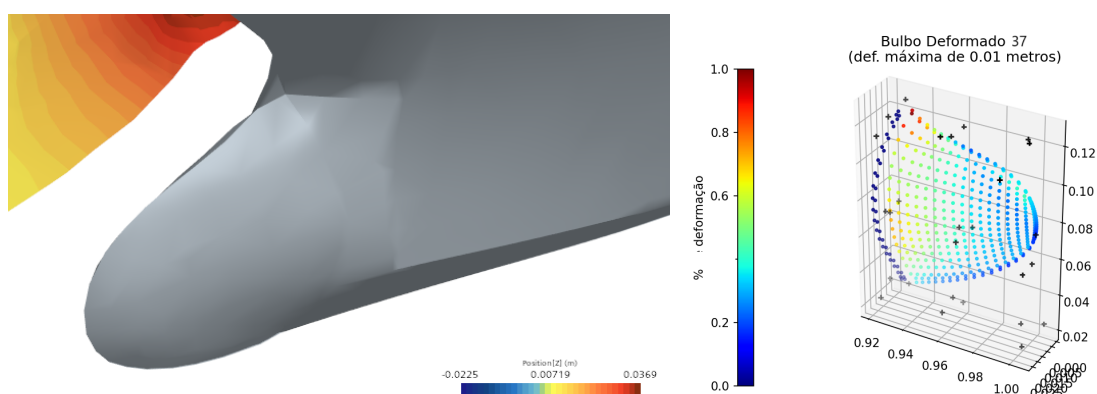
Fonte: Autor, 2024.

Figura 4.2.9 - Bulbo 57 com menor empuxo.



Fonte: Autor, 2024.

Figura 4.2.10 - Bulbo 37 com maior empuxo.



Fonte: Autor, 2024.

4.3. Definição de relações

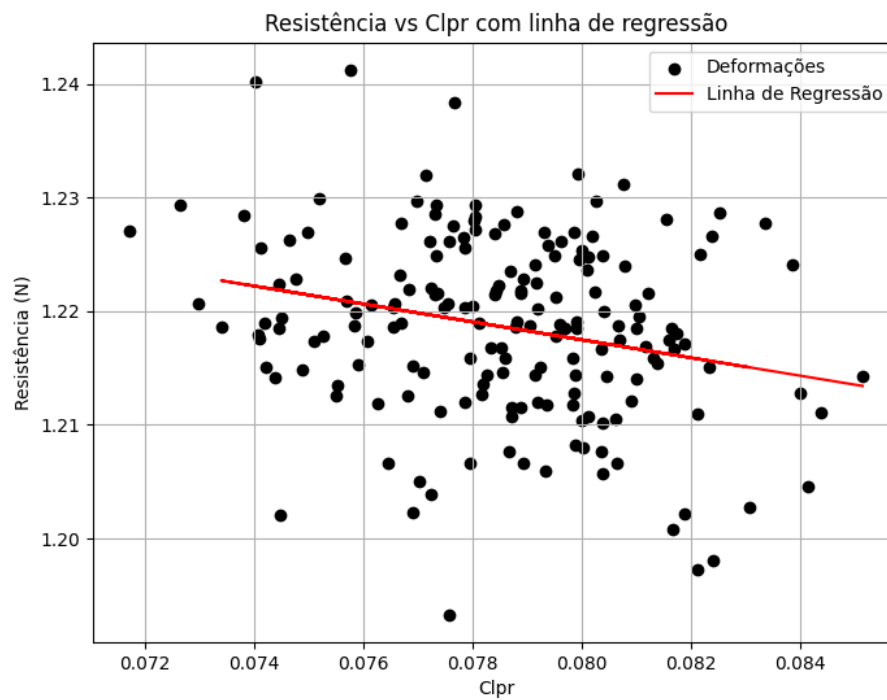
Com os dados de resistência ao avanço, são buscadas relações entre os parâmetros geométricos obtidos em 4.1 e a resistência estimada por CFD (ver figuras 4.3.1 a 4.3.4). Não se encontra uma relação direta com nenhuma das variáveis independentemente, porém, para C_{ZB} e C_{LRP} se encontra com regressão linear uma tendência onde são inversamente proporcionais a resistência ao avanço. As regressões vistas nas figuras 4.3.1 para C_{LRP} e figura 4.3.2 para C_{ZB} terão para a equação 3.4.1 os valores da tabela 4.3.1. Não se encontra tendência significativa para as curvaturas.

Tabela 4.3.1 - Coeficientes das regressões lineares de C_{LRP} e C_{ZB} .

Propriedade Geométrica	Y_0	C	R^2
C_{LRP}	1,2806	-0,7895	0,0143
C_{ZB}	1,2634	-0,0674	0,0458

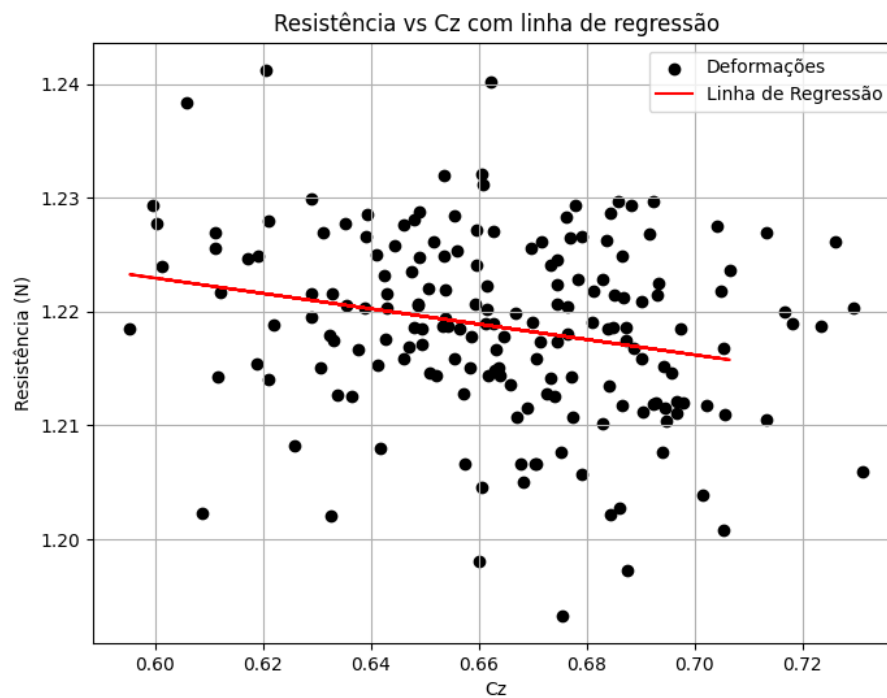
Fonte: Autor, 2024.

Figura 4.3.1 - Gráfico de dispersão comparando a resistência ao avanço com C_{LRP} .



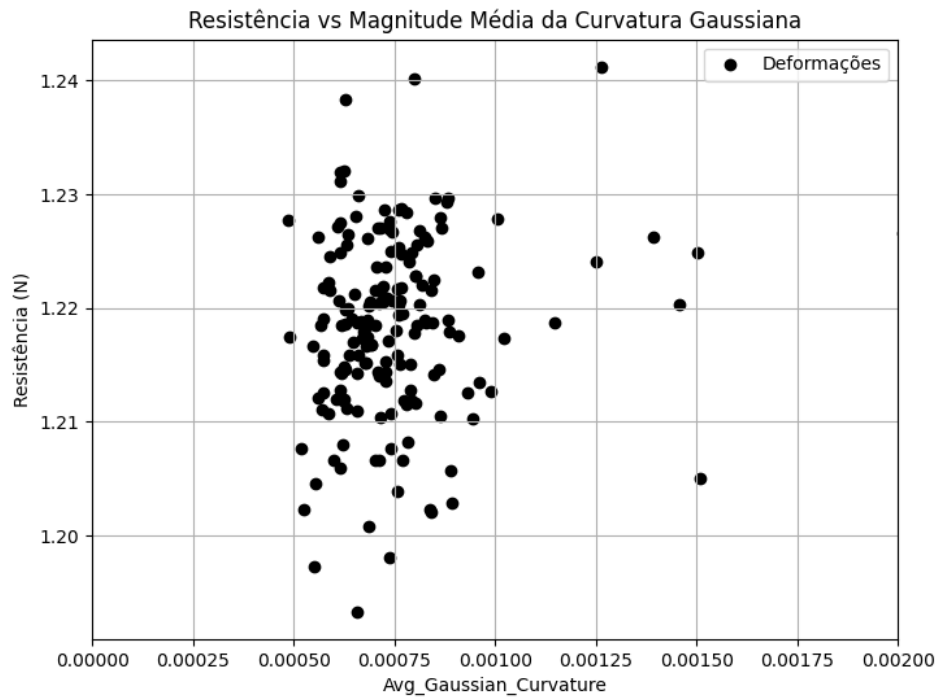
Fonte: Autor, 2024.

Figura 4.3.2 - Gráfico de dispersão comparando a resistência ao avanço com C_{ZB} .



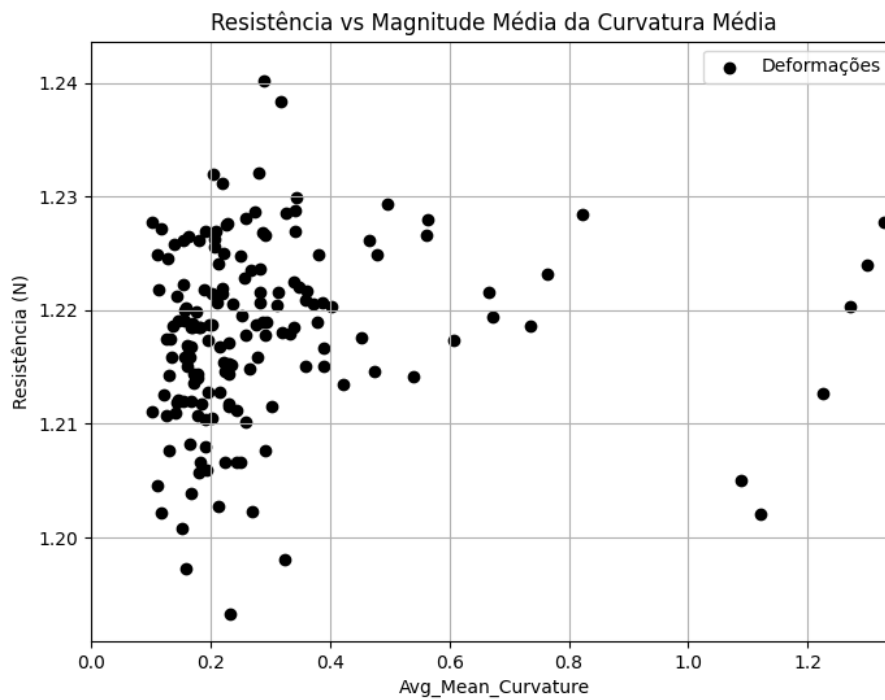
Fonte: Autor, 2024.

Figura 4.3.3 - Gráfico de dispersão comparando a resistência ao avanço com a magnitude média da curvatura gaussiana.



Fonte: Autor, 2024.

Figura 4.3.4 - Gráfico de dispersão comparando a resistência ao avanço com a magnitude média da curvatura média.

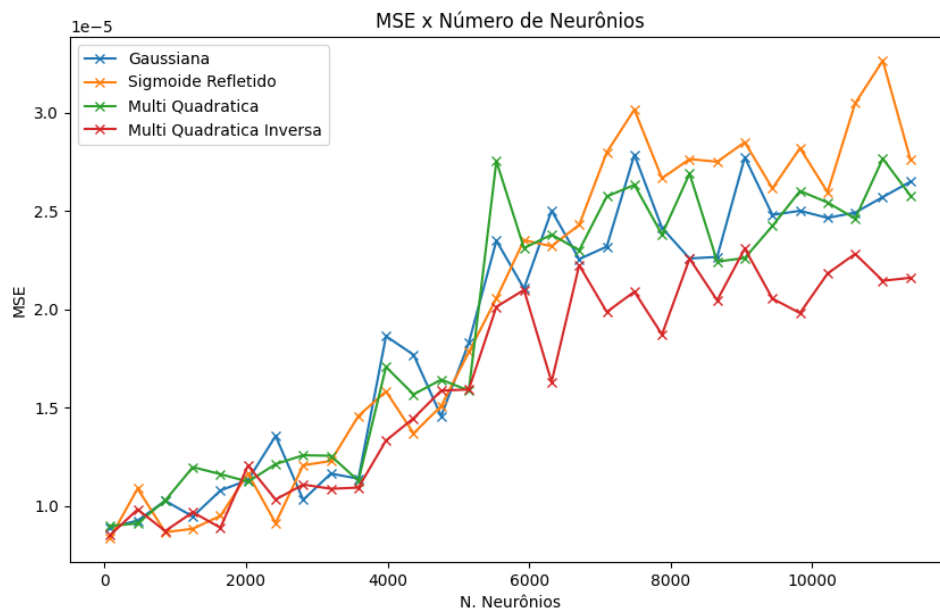


Fonte: Autor, 2024.

4.4. Treinamento das redes RBF

Como explicado em 3.4, inicialmente foi feita uma análise de sensibilidade para se decidir o número de neurônios ideal para se treinar a rede, para isto dos 184 (cento e oitenta e quatro) dados históricos utilizados, já excluindo os 16 (dezesesseis) casos explicado em 4.2, foram-se separados 35 (trinta e cinco) casos como casos de validação e se obteve a relação vista em 4.4.1 onde o erro médio cresce claramente com o número de neurônios.

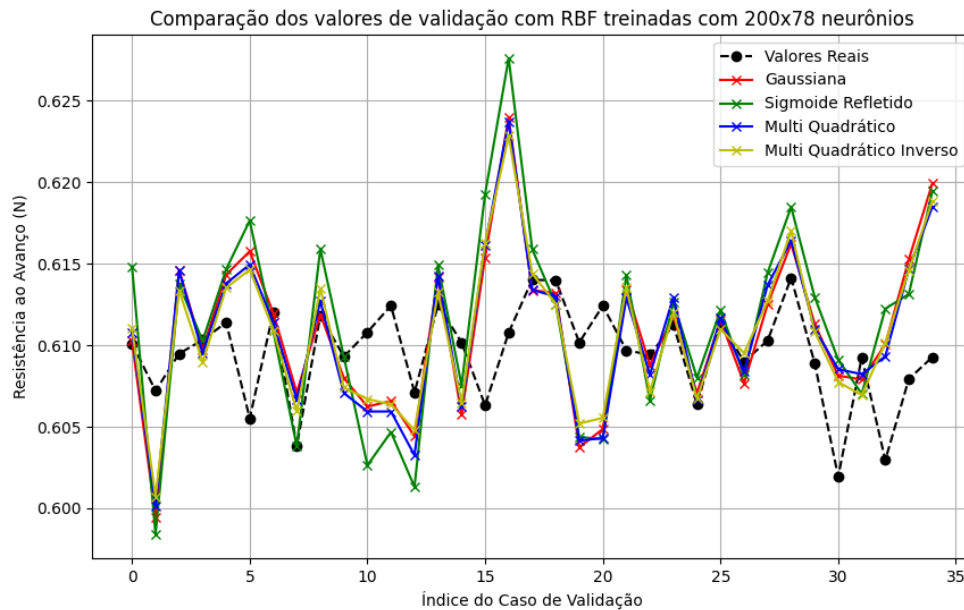
Figura 4.4.1 - Comportamento das redes RBF em função do número de neurônios.



Fonte: Autor, 2024.

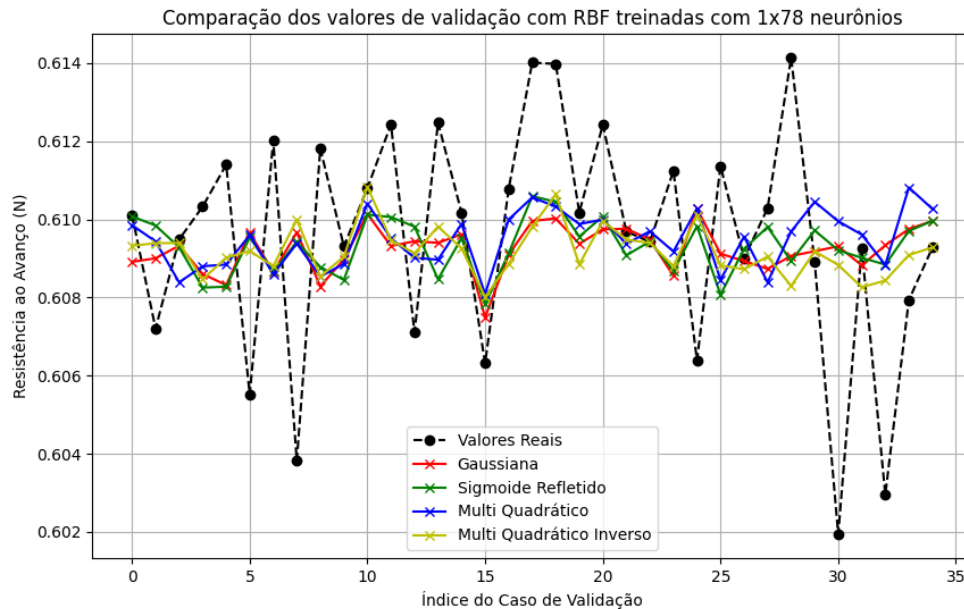
Podemos observar com mais detalhes que redes RBF com altos números de neurônios, como as da figura 4.4.2, com 20 x 78 neurônios têm comportamento que sugere *overfitting*, em que todas as redes têm resultados semelhantes mas errôneos em alguns pontos, com valores muito além ou abaixo dos resultados reais. Porém, na situação com menor erro médio quadrado possível, em que o número de neurônios é 78 (i.e., pois como visto em 2.7.1, deve ser pelo menos igual à dimensão das entradas) se é visto na figura 4.4.3 que os resultados ficam muito limitados a média dos resultados, e teria dificuldade de prever valores das extremidades mínimas e máximas, como nosso objetivo.

Figura 4.4.2 - Comparação das previsões dos modelos RBF treinados com 200 x 78 neurônios com os resultados reais dos casos de validação.



Fonte: Autor, 2024.

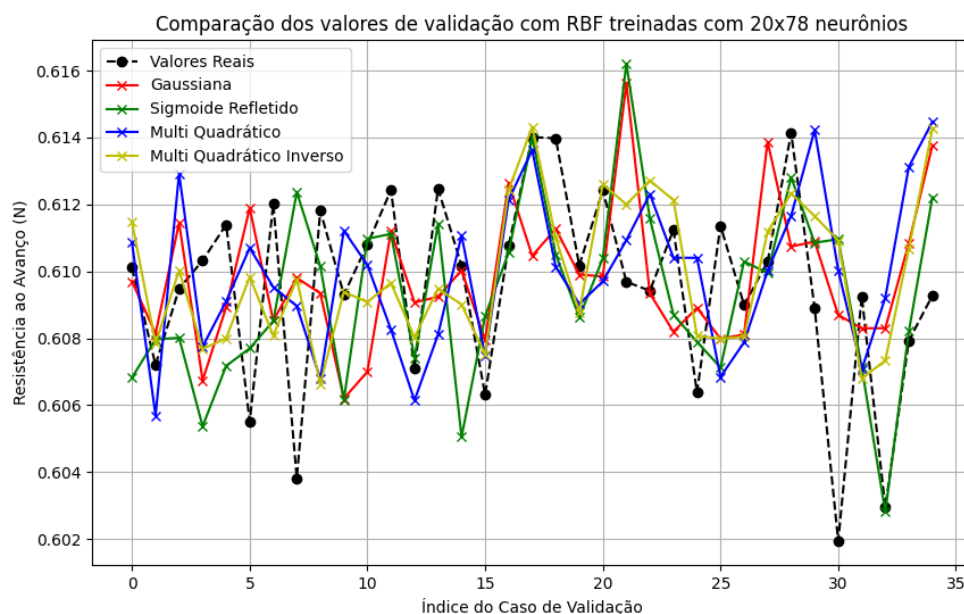
Figura 4.4.3 - Comparação das previsões dos modelos RBF treinados com 1 x 78 neurônios com os resultados reais dos casos de validação.



Fonte: Autor, 2024.

Finalmente, a quantidade de neurônios utilizada para treinar as redes RBF que serão modelo substituto para o EA é de 20 x 78, onde como visto na figura 4.4.3, mantém boa diversidade dos resultados das redes RBF e captura mais pontos extremos.

Figura 4.4.4 - Comparação das previsões dos modelos RBF treinados com 20 x 78 neurônios com os resultados reais dos casos de validação.

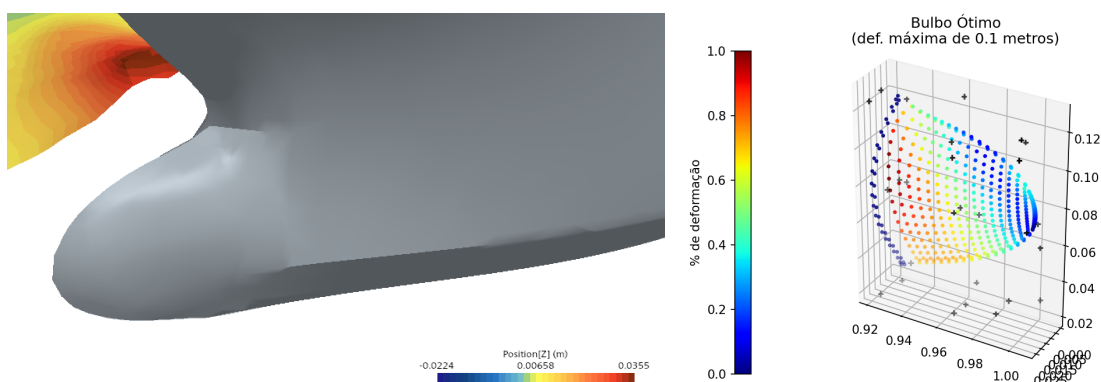


Fonte: Autor, 2024.

4.5. Bulbo encontrado pelo EA

O EA utilizado continha 200 (duzentos) indivíduos e duração de 150 (cento e cinquenta) gerações, prevendo como ótimo (*i.e.*, menor resistência ao avanço) o bulbo da figura 4.5.1, que prediz ter uma resistência ao avanço de 1,2023 N. Utilizando as mesmas técnicas descritas em 2.2 e 2.5, se preenche a tabela 4.5.1 com as características geométricas de comparação, e as curvaturas podem ser visualizadas na figura 4.5.2.

Figura 4.5.1 - Geometria do bulbo ótimo previsto pelo AE.



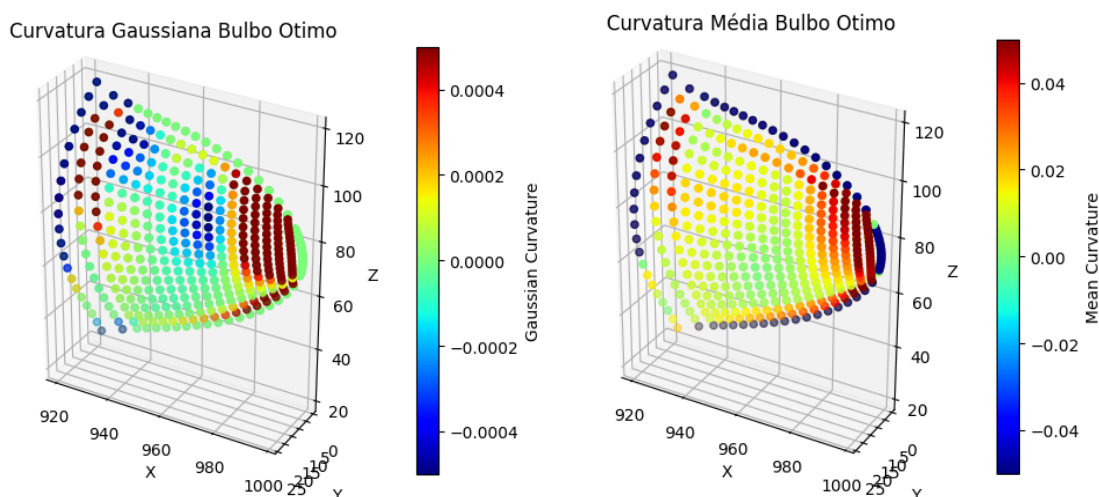
Fonte: Autor, 2024.

Tabela 4.5.1 - Propriedades Geométricas do bulbo ótimo encontrado pelo EA.

C_{LPR}	C_{ZB}	Crv. Gaussian Magnitude Média	Crv. Média Magnitude Média
0,0773	0,6206	0,0008	0,4824

Fonte: Autor, 2024.

Figura 4.5.2 - Curvaturas Gaussiana (esquerda) e Média (direita) do bulbo ótimo.

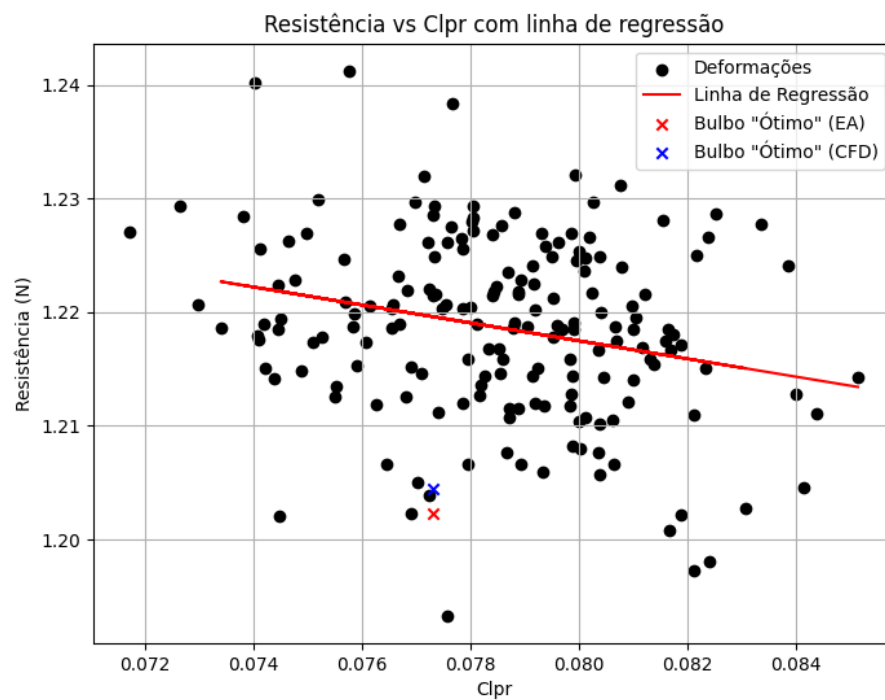


Fonte: Autor, 2024.

Temos então que o bulbo “ótimo” encontrado pelo EA tem uma resistência ao avanço 1,34% (um vírgula trinta e quatro por cento) menor que a média dos resultados obtidos, porém com seis bulbos aleatórios, 3% (três por cento) dos analisados, tendo uma resistência ao avanço inferior. Se observa que a EA não concordou com a regressão em tentar diminuir os fatores de C_{LPR} e C_{ZB} .

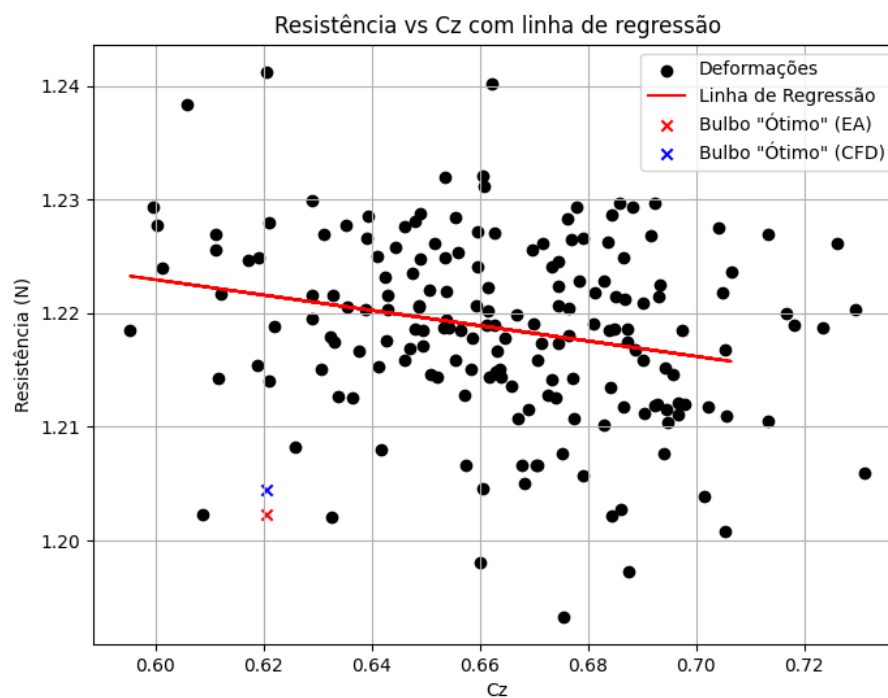
Ao exportarmos o bulbo “ótimo” sugerido para o STAR-CCM+ e o aplicarmos na metodologia, encontramos uma resistência ao avanço de 1,2045 N com desvio padrão no tempo físico de análise de 0,0012 N, ou seja, uma diferença de 0,0022 N ou 1,84 vezes o desvio padrão do resultado CFD. Logo, tem nove deformações aleatórias com resultados de resistência ao avanço inferiores, 5% (cinco por cento) dos analisados. Incluindo estes bulbos nos gráficos de dispersão de comparação dos resultados temos as figuras 4.5.3 a 4.5.6.

Figura 4.5.3 - Gráfico de dispersão comparando a resistência ao avanço com C_{LPR} incluindo os resultados do bulbo ótimo.



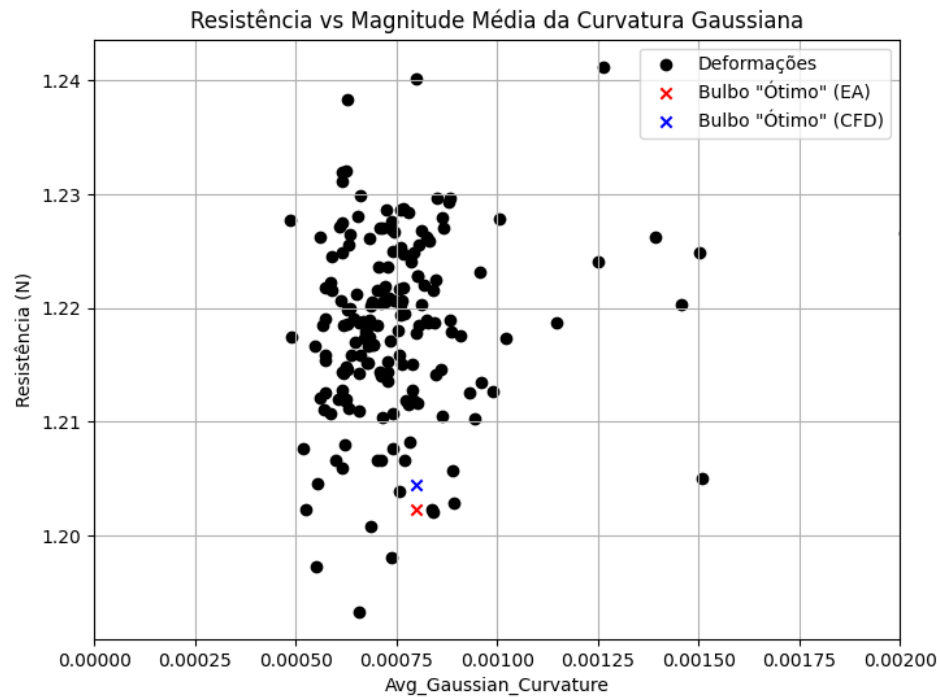
Fonte: Autor, 2024.

Figura 4.5.4 - Gráfico de dispersão comparando a resistência ao avanço com C_{ZB} incluindo os resultados do bulbo ótimo.



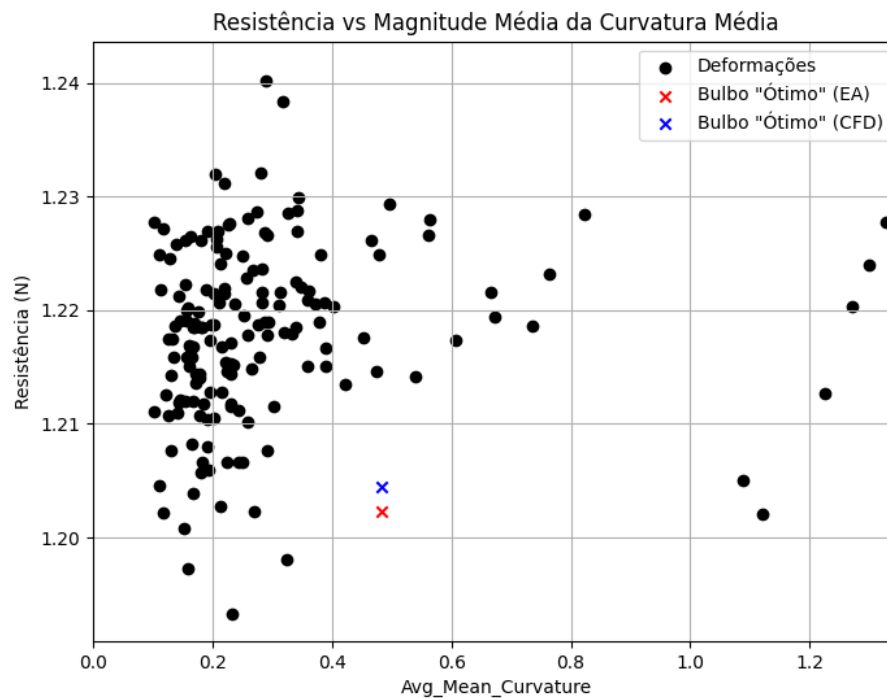
Fonte: Autor, 2024.

Figura 4.5.5 - Gráfico de dispersão comparando a resistência ao avanço com a Magnitude Média da Curvatura Gaussiana incluindo os resultados do bulbo ótimo.



Fonte: Autor, 2024.

Figura 4.5.6 - Gráfico de dispersão comparando a resistência ao avanço com a Magnitude Média da Curvatura Média incluindo os resultados do bulbo ótimo.



Fonte: Autor, 2024.

5. DISCUSSÕES

Primeiramente, as deformações FFD se provaram capazes de gerar diversas versões de bulbos que se assemelham aos observados na prática de arte naval, sem formas com construção impossível ou deformidades aparente, indicando a boa adaptação do FFD para esta superfície e dos critérios de restrições adotados.

Porém, 10% (dez por cento) em cada coordenada da *lattice* foi um limite de deformação possivelmente muito restritivo, e não permitiu que geometrias muito diferentes fossem geradas. Ademais, é possível que a relação inversamente proporcional dos parâmetros geométricos C_{LPR} e C_{ZB} com a resistência ao avanço possua na verdade um caráter curvo fora dos espaços de deformação analisados, onde se poderá encontrar regiões de mínimo que guiaram futuros projetos na criação de um bulbo para embarcações. Por isso, trabalhos futuros devem investigar maiores limites de deformação.

Ao analisar as propriedades geométricas dos bulbos criados, foram observados geometrias com *outliers* em suas curvaturas gaussiana e média, essas ocorreram devido ao método do critério c^1 de continuidade adotado, porém, não se observou efeito significativo desses outliers na resistência ao avanço.

Como se observa que as curvaturas, em especial a curvatura gaussiana, não possui relação observável com a resistência ao avanço, é interessante o desenvolvimento de métodos que permitam a otimização simultânea da otimização da resistência ao avanço com a diminuição da curvatura gaussiana, de forma a utilizar critérios como os descritos por Wang et al. (2023) para encontrar um bulbo simultaneamente efetivo e de mais fácil construção.

Na análise com CFD, foi possível detectar a variação da resistência ao avanço com os diferentes bulbos testados, porém, alguns parâmetros da simulação podem ser melhorados para obtenção de resultados mais precisos. Inicialmente, para se obter uma maior quantidade de casos, a malha computacional ficou com menos de um milhão de elementos, para esse problema podendo ser considerada uma malha grossa, e se vê que as deformações mais sutis da geometria não foram capturadas, isso pode também ter impossibilitado detectar uma relação entre as curvaturas e a resistência ao avanço.

Como explicado em 4.2, os primeiros 60 segundos da simulação tinha um padrão senoidal como variação dominante nos valores das forças, não as mudanças da geometria do bulbo, portanto, é interessante incluir critérios de parada baseados em convergência para o começo do laço de substituição de bulbos e para cada bulbo dentro do laço.

Técnicas podem ser empregadas para isolar os efeitos de modificação do bulbo na resistência ao avanço, como: considerar apenas as componentes das forças causadas pela pressão, ou seja, ignorar a parte viscosa na qual os efeitos do bulbos são menos relevantes. Ou fazer uma análise do perfil de onda e observar como se modifica com a deformação dos bulbos.

O procedimento CFD foi demorado computacionalmente, mesmo considerando a quantidade de recursos de *hardware* aplicados. Foi-se observado durante a aplicação dos modelos substitutos RBF que enquanto que o erro das previsões diminuía significativamente com a quantidade de casos adicionados à base de dados, até se aproximar de 100 (cem) casos, quando retornos de precisão foram sendo observados menos significativos por cada caso adicionado. Portanto, para o procedimento sugerido ter mais aplicabilidade real, é necessário fazer uma análise de sensibilidade na quantidade real de casos a serem testados, principalmente ao se considerar aumentar a precisão da malha computacional no CFD.

Sobre o treinamento das redes RBF, se observa que elas não predizem bem todos os pontos de validação, isso era esperado e uma das vantagens de se aplicar um modelo de EA que utiliza mais de um modelo substituto. Porém, é válido buscar a aplicação de outros modelos como modelos de escala dupla sugeridos por Ku, Zhen e Gong (2023), ou o desenvolvimento de um modelo dedicado.

Ainda sobre as redes RBF, a análise de sensibilidade é inicialmente passável, porém, uma métrica como a correlação de ranque de Kendall (MARDEN; KENDALL; GIBBONS, 1992) seria mais informativa.

Foi encontrado com o EA empregado um bulbo ‘ótimo’ que não provou ter a menor resistência ao avanço dentro de todos os bulbos gerados aleatoriamente, nem em sua previsão de resistência pelo EA, nem em seu teste com CFD. Porém, a previsão do EA e o resultado CFD foram condizentes um com o outro.

É possível que o EA tenha entrado em uma região de mínimo local dentro do espaço amostral, e aplicar técnicas que preservam melhor a diversidade ao longo das gerações (*e.g.*, aumentando o número de indivíduos) pode chegar a encontrar um bulbo ainda melhor.

Outra possibilidade seria reduzir o espaço de busca, utilizando deformações maiores do bulbo inicial para encontrar a região de valores de resistência ao avanço mínimas em função dos parâmetros geométricos do bulbo, e buscar mais resultados CFD nestas regiões, diminuindo o espaço de busca para o EA.

Finalmente, também existe a chance de esta falha em prever o melhor resultado seja uma limitação dos modelos substitutos utilizados, como se observa na figura 4.4.3, os modelos têm dificuldade em prever os valores extremos menores da resistência ao avanço.

6. CONCLUSÕES

A metodologia proposta oferece melhorar a resistência ao avanço de uma embarcação utilizando técnicas de CFD e EA modificando o bulbo sem afetar a viabilidade construtiva da embarcação.

Como principal resultado temos uma metodologia para otimização de bulbos de uma embarcação, que busca aplicar de forma prática técnicas de inteligência computacional no cenário naval. A metodologia foi aplicada a uma embarcação na escala reduzida que representa um rebocador, para competição denominado Superlâmpião (vide figura 3.1.1) e encontrou-se múltiplas modificações para o bulbo (ver figura 4.1.2) utilizando FFD, para cada bulbo calculou-se suas propriedades geométricas (ver tabela 4.1.1 e 4.1.2), e utilizou CFD para estimar a resistência ao avanço de forma numérica (ver tabela 4.2.1). Buscou-se relações entre as propriedades geométricas de cada bulbo com a resistência ao avanço e encontrou-se uma leve relação inversamente proporcional entre a resistência ao avanço e os parâmetros geométricos (comprimento e altura), nenhuma relação significativa com as curvaturas da geometria (ver figuras 4.3.1 a 4.3.4) que poderiam ser pesquisados a futuro. Foram treinadas quatro redes RBF com base em uma simples análise de sensibilidade (veja figura 4.4.1) para servirem de modelos substitutos de um DDEA *offline*, que com 200 (duzentos) indivíduos em 150 (cento e cinquenta) gerações encontrou um bulbo “ótimo”, que na realidade não foi o melhor das geometrias analisadas, mas sua resistência prevista estava entre os melhores 3% (três por cento) dos bulbos analisados. A resistência ao avanço (R_t) com o bulbo encontrado pelo DDEA foi estimada e é encontrado um erro de 0,18% do valor, previsto pelo DDEA, porém com o bulbo ainda estando entre os melhores 5% (cinco por cento) dos analisados.

A metodologia proposta abre caminho para aplicações mais realistas de técnicas de inteligência computacional na indústria naval.

Finalmente sugere-se para trabalhos futuros a ampliação das deformações a geometria incluindo toda a proa, ou todo o casco, incluindo devidos critérios de restrição para não se perder a forma útil do casco. Utilizar a metodologia para outras embarcações e comparar os resultados em diferentes modelos, e tentar treinar um modelo de IC genérico. Ademais, programação de uma análise CFD própria, pode permitir a criação de um procedimento semi-automatizado generalista e utilização de modelos de DDEA *online*.

7. REFERÊNCIAS

ANDERSON, J. **Computational Fluid Dynamics**. Nova Iorque, NY, USA: McGraw-Hill Professional, 1995.

AO, Y., LI, Y., GONG, J., & LI, S. (2023). **An artificial intelligence-aided design (AIAD) of ship hull structures**. Journal of Ocean Engineering and Science, 8(1), 15–32. <https://doi.org/10.1016/J.JOES.2021.11.003>

BINGOL, O. R.; KRISHNAMURTHY, A. **NURBS-Python: An open-source object-oriented NURBS modeling framework in Python**. SoftwareX, v. 9, p. 85–94, 2019.

CALHOUN, D.; LEVEQUE, R. J. **A Cartesian grid finite-volume method for the advection-diffusion equation in irregular geometries**. Journal of computational physics, v. 157, n. 1, p. 143–180, 2000.

CEBECI, T. *et al.* **Computational fluid dynamics for engineers**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

EIBEN, A. E.; SMITH, J. E. **Introduction to evolutionary computing**. 1. ed. Berlin, Germany: Springer, 2007.

EMANUEL, G. (2001). **Analytical fluid dynamics**. 2. ed. CRC Press. pp. 6–7. [ISBN 0-8493-9114-8](https://doi.org/10.1080/084939114883888888).

ENGELBRECHT, A. P. **Computational Intelligence: An Introduction**. 2. ed. Hoboken, NJ: Wiley-Blackwell, 2007.

HADIGOL, M.; DOOSTAN, A. **Least squares polynomial chaos expansion: A review of sampling strategies**. Computer methods in applied mechanics and engineering, v. 332, p. 382–407, 2018.

HIRT, C. W.; NICHOLS, B. D. **Volume of fluid (VOF) method for the dynamics of free boundaries**. Journal of computational physics, v. 39, n. 1, p. 201–225, 1981.

HUANG, P.; WANG, H.; MA, W. **Stochastic ranking for offline data-driven evolutionary optimization using radial basis function networks with multiple kernels**. 2019 IEEE Symposium Series on Computational Intelligence (SSCI). Anais...IEEE, 2019.

ITTC. **Uncertainty Analysis in CFD Verification and Validation Methodology and Procedures.** [s.l]. 2017.

JONES, W. P.; LAUNDER, B. E. **The prediction of laminarization with a two-equation model of turbulence.** International journal of heat and mass transfer, v. 15, n. 2, p. 301–314, 1972.

JOY, K. I. **On-Line Geometric Modeling Notes: BERNSTEIN POLYNOMIALS.** , 2000.

KHAN, S., GOUCHER-LAMBERT, K., KOSTAS, K., & KAKLIS, P. (2023). **ShipHullGAN: A generic parametric modeller for ship hull design using deep convolutional generative model.** Computer Methods in Applied Mechanics and Engineering, 411, 116051. <https://doi.org/10.1016/J.CMA.2023.116051>

KIM, J.-H. et al. **Prediction of the superiority of the hydrodynamic performance of hull forms using deep learning.** International journal of naval architecture and ocean engineering, v. 14, n. 100490, p. 100490, 2022.

KIM, S. Y., MOON, B. Y., KIM, D. E., & SHIN, S. C. (2006). **Automation of hull plates classification in ship design system using neural network method.** *Mechanical Systems and Signal Processing*, 20(2), 493–504. <https://doi.org/10.1016/J.YMSSP.2004.06.008>

KRACHT, A. M. **Design of Bulbous Bows.** SNAME Transactions, v. 86, n. 1978, p. 197–217, 1978.

KU, J.; ZHEN, H.; GONG, W. **Offline data-driven optimization based on dual-scale surrogate ensemble.** Memetic computing, v. 15, n. 2, p. 139–154, 2023.

MANEN, J. D.; VAN OOSSANEN, P. **Resistance. Em: LEWIS, E. V. (Ed.). Principles of Naval Architecture.** Alexandria VA, USA: The Society of Naval Architects and Marine Engineers, 1988. v. 2p. 1–128.

MARDEN, J. I.; KENDALL, M.; GIBBONS, J. D. **Rank correlation methods (5th ed.).** Journal of the American Statistical Association, v. 87, n. 417, p. 249, 1992.

MARGARI, V., KANELLOPOULOU, A., & ZARAPHONITIS, G. (2018). **On the use of Artificial Neural Networks for the calm water resistance prediction of**

MARAD Systematic Series' hullforms. *Ocean Engineering*, 165, 528–537.
<https://doi.org/10.1016/J.OCEANENG.2018.07.035>

MIKULIĆ, A., & PARUNOV, J. (2019). **A review of artificial intelligence applications in ship structures.** In *Trends in the Analysis and Design of Marine Structures* (pp. 515–523). <https://doi.org/10.1201/9780429298875-59>

MOLLAND, A. F.; TURNOCK, S. R.; HUDSON, D. A. **Ship Resistance and Propulsion - Practical Estimation of Ship Propulsive Power.** New York, USA: Cambridge University Press, 2011.

NAVICULABOATDESIGN. **Postagem no Instagram de 2 de Junho de 2017.** Disponível em: <<https://www.instagram.com/p/BU2XsYPBTbY/>>. Acesso em: 7 out. 2024.

PRESSLEY, A. **Elementary Differential Geometry.** 2001. ed. Londres, England: Springer, 2001.

RODI, W. **Experience with two-layer models combining the k-epsilon model with a one-equation model near the wall.** 29th Aerospace Sciences Meeting. Anais...Reston, Virigina: American Institute of Aeronautics and Astronautics, 1991.

ROSEMAN, D. P. *The MARAD systematic series of full-form ship models.* 1987

SALMOIRAGHI, F., SCARDIGLI, A., TELIB, H., & ROZZA, G. (2018). **Free-form deformation, mesh morphing and reduced-order methods: enablers for efficient aerodynamic shape optimisation.** *International Journal of Computational Fluid Dynamics*, 32(4–5), 233–247. <https://doi.org/10.1080/10618562.2018.1514115>

SEDERBERG, T. W.; PARRY, S. R. **Free-form deformation of solid geometric models.** Computer graphics, v. 20, n. 4, p. 151–160, 1986.

SHARMA, R.; SHA, O. P. **Practical hydrodynamic design of bulbous bows for ships.** Naval engineers journal, v. 117, n. 1, p. 57–76, 2005.

SIEMENS. Simcenter **STARCCM+ 11.06 User Guide.** , 2018.

SHIH, T.-H. *et al.* **A new k- ϵ eddy viscosity model for high reynolds number turbulent flows.** Computers & fluids, v. 24, n. 3, p. 227–238, 1995.

SVENSSON, J. et al. **Feasibility of patient specific aortic blood flow CFD simulation. Medical image computing and computer-assisted intervention: MICCAI ...** International Conference on Medical Image Computing and Computer-Assisted Intervention, v. 9, n. Pt 1, p. 257–263, 2006.

TODD, F. H.. *Series 60 Methodical Experiments With Model of Single-Screw Merchant Ships*. 1963

VENKATAKRISHNAN, V.; MAVRIPLIS, D. J. **Implicit method for the computation of unsteady flows on unstructured grids**. Journal of computational physics, v. 127, n. 2, p. 380–397, 1996.

VERNENGO, G., VILLA, D., GAGGERO, S., & VIVIANI, M. (2020). **Interactive design and variation of hull shapes: pros and cons of different CAD approaches**. *International Journal on Interactive Design and Manufacturing*, 14(1), 103–114. <https://doi.org/10.1007/s12008-019-00613-3>

WANG, S., XU, Z., WANG, J., LI, R., & WANG, Y. (2023). **Forming Difficulty Evaluation for Curved Hull Plates Based on Grey Relational Analysis**. *Applied Sciences (Switzerland)*, 13(9). <https://doi.org/10.3390/app13095233>

WANG, S.; KIM, Y. **Adaptive grid deformation method for CFD application to hull optimization**. IOP conference series. Materials science and engineering, v. 1288, n. 1, p. 012043, 2023.

YU, Y., LI, K., LIN, Y., & CHEN, M. (2015). **A New Mesh Transformation Method for Hull Structure Shape Optimization Based on Parametric Technology**. *International Offshore and Polar Engineering Conference*.

ZHEN, H.; GONG, W.; WANG, L. **Offline data-driven evolutionary optimization based on model selection**. Swarm and evolutionary computation, v. 71, n. 101080, p. 101080, 2022.

ZHOU, H., FENG, B., LIU, Z., CHANG, H., & CHENG, X. (2022). **NURBS-Based Parametric Design for Ship Hull Form**. *Journal of Marine Science and Engineering*, 10(5). <https://doi.org/10.3390/jmse10050686>

ANEXO I - CONFIGURAÇÕES DA SIMULAÇÃO NO STAR-CCM+

Summary Report: LoopTest_V2_FINISHED

Session Summary

Date 9 de out. de 2024 01:08:59
File size 4,1e+02 MB

Software Summary

Simulation Properties

1 LoopTest_V2_FINISHED

+1 Continua	Continua	1
\-1 Physics 1	Regions	[Dominio]
	Interfaces	[]
	Point Sets	[]
	Active	TRUE
	Motion Always Active	FALSE
	Tags	[]
+1 Models		
+1 Cell Quality Remediation		
+2 Gradients	Limit GradVar By Beta	TRUE
	Boundary Cell Eigenvalues Ratio Tolerance	0.06
	Limiter Method	Venkatakrishnan
	Custom Accuracy Level Selector	2.0
	Maximum Reconstruction Coefficient	1.0
	Two Pass Velocity Gradient	FALSE
	Use TVB Gradient Limiting	FALSE
	Acceptable Field Variation (Factor)	0.05
+3 Gravity		
+4 High y+ Wall Treatment		
+5 Implicit Unsteady	Continuum Iteration	883125
	Continuum Physical Time	7.849.999.999.979.110
+6 K-Epsilon Turbulence		
+7 Multiphase		
+1 Phases		
+1 Air		
\-1 Models		
+1 Constant Density		
+2 Gas		
\-1 Air	Database Material	Air (Air) [Standard/Gases]
	Tags	[]
\-1 Material Properties		
+1 Density	Method	Constant
\-1 Constant	Value	1.18415 kg/m^3
\-2 Dynamic Viscosity	Method	Constant
\-1 Constant	Value	1.85508E-5 Pa-s
\-3 Turbulent		
\-2 Water		
\-1 Models		
+1 Constant Density		
+2 Liquid		
\-1 H2O	Database Material	H2O (Water) [Standard/Liquids]
	Tags	[]

`-1 Material Properties		
`+-1 Density	Method	Constant
`-1 Constant	Value	997.561 kg/m^3
`-2 Dynamic Viscosity	Method	Constant
`-1 Constant	Value	8.8871E-4 Pa-s
`-3 Turbulent		
`-2 Mixture	Tags	[]
`-1 Material Properties		
`-1 Dynamic Viscosity	Method	Volume-Weighted Mixture
`-1 Volume-Weighted Mixture		
`+-8 Multiphase Interaction		
`-1 Phase Interactions		
`+-9 Realizable K-Epsilon	Buoyancy Production of Dissipation	Boundary Layer Orientation
	Cmu	0.09
	C1e	1.44
	C2e	1.9
	Ct	1.0
	Sigma_k	1.0
	Sigma_e	1.2
	Sarkar	2.0
	Tke Minimum	1.0E-10
	Tdr Minimum	1.0E-10
	Secondary Gradients	On
	Convection	2nd-order
	Normal Stress Term	FALSE
	Curvature Correction Option	Off
`+-10 Reynolds-Averaged Navier-Stokes		
`+-11 Segregated Flow	Flux Update Dissipation Factor	0.5
	Positivity Rate Limit	0.2
	Flow Boundary Diffusion	TRUE
	Unsteady Flux Dissipation Corrections	FALSE
	Limit Acoustic-CFL Option	Per-Model
	Secondary Gradients	On
	Convection	2nd-order
	Delta-V Dissipation	Off
`+-12 Solution Interpolation	Per Part Mapping	FALSE
	Interpolation Method	Nearest neighbor
	Legacy Method	FALSE
	Conservation Correction	Disable
`+-13 Three Dimensional		
`+-14 Turbulent		
`+-15 VOF Waves	Damping Constant 1	10.0 /s
	Damping Constant 2	10.0 /m
	Damping Exponent	2.0
	Enable Theory Comparison	FALSE
	Forcing Constant	10.0 /s
`-1 Waves		
`-1 Free Surface	Point On Water Level	[0.0, 0.0, 0.0] m
	Vertical Direction	[0.0, 0.0, 1.0]
	Current	[-0.9, 0.0, 0.0] m/s
	Wind	[-0.9, 0.0, 0.0] m/s

	Light Fluid Density	1.18415 kg/m ³
	Heavy Fluid Density	997.561 kg/m ³
	Tags	[]
+16 VOF Wave Zone Distance	VOF Wave Zone Distance Method	Implicit Tree
+17 Volume of Fluid (VOF)	Convection	HRIC
	Face Density Reconstruction	1st-order
-1 HRIC	CFL_l	0.5
	CFL_u	1.0
	Sharpening Factor	0.0
	Angle Factor	0.05
	Enable HRIC Gradient Smoothing	FALSE
-18 Wall Distance	Wall Distance Method	Implicit Tree
+2 Reference Values		
+1 Gravity	Value	[0.0, 0.0, -9.81] m/s ²
+2 Reference Altitude	Value	[0.0, 0.0, 0.0] m
+3 Minimum Allowable Wall Distance	Value	1.0E-6 m
+4 Maximum Allowable Absolute Pressure	Value	1.0E8 Pa
+5 Reference Density	Value	1.0 kg/m ³
+6 Minimum Allowable Absolute Pressure	Value	1000.0 Pa
-7 Reference Pressure	Value	101325.0 Pa
-3 Initial Conditions		
+1 Pressure	Method	Field Function
-1 Field Function	Scalar Function	Hydrostatic Pressure of Heavy Fluid of Free Surface
+2 Turbulence Specification	Method	K + Epsilon
+3 Turbulent Dissipation Rate	Method	Constant
-1 Constant	Value	0.1 m ² /s ³
+4 Turbulent Kinetic Energy	Method	Constant
-1 Constant	Value	0.001 J/kg
+5 Velocity	Method	Field Function
	Coordinate System	Laboratory
-1 Field Function	Vector Function	Velocity of Free Surface
-6 Volume Fraction	Method	Composite N-1
-1 Composite N-1		
-1 Air	Method	Field Function
-1 Field Function	Scalar Function	Volume Fraction of Light Fluid of Free Surface
+2 Regions		
	Part Selection Priority	[Dominio]
	Regions	1
-1 Dominio	Index	0
	Allow Per-Part Values	FALSE
	Physics Continuum	[Physics 1]
	Parts	[Subtract]
	Type	Fluid Region
	Topology	VOLUME
	Tags	[]
+1 Boundaries	Part Surface Selection Priority	[Dominio: Subtract.Hull.Default, Dominio: Subtract.Block Dominio.Top, Dominio: Subtract.Block Dominio.Lateral, Dominio: Subtract.Block Dominio.Outflow, Dominio: Subtract.Block Dominio.SymPlane, Dominio: Subtract.Block Dominio.Bottom, Dominio: Subtract.Block Dominio.Inflow]

	Boundaries	7
+1 Subtract.Block Dominio.Bottom	Index	7
	Interfaces	
	Part Surfaces	[Subtract.Block Dominio.Bottom]
	Type	Symmetry Plane
	Allow Per-Surface Values	FALSE
	Topology	SURFACE
	Tags	[]
`1 Physics Conditions		
`1 VOF Wave Damping Boundary Option	Method	No
+2 Subtract.Block Dominio.Inflow	Index	8
	Interfaces	
	Part Surfaces	[Subtract.Block Dominio.Inflow]
	Type	Velocity Inlet
	Allow Per-Surface Values	FALSE
	Topology	SURFACE
	Tags	[]
+1 Physics Conditions		
+1 Flow Direction Specification	Method	Boundary-Normal
+2 Reference Frame Specification	Option	Lab Frame
+3 Turbulence Specification	Method	Intensity + Viscosity Ratio
+4 Velocity Specification	Method	Magnitude + Direction
`5 VOF Wave Damping Boundary Option	Method	Yes
`2 Physics Values		
+1 Turbulence Intensity	Method	Constant
`1 Constant	Value	0.01
+2 Turbulent Viscosity Ratio	Method	Constant
`1 Constant	Value	10.0
+3 Velocity Magnitude	Method	Field Function
`1 Field Function	Scalar Function	Velocity of Free Surface: Magnitude
`4 Volume Fraction	Method	Composite N-1
`1 Composite N-1		
`1 Air	Method	Field Function
`1 Field Function	Scalar Function	Volume Fraction of Light Fluid of Free Surface
+3 Subtract.Block Dominio.Lateral	Index	4
	Interfaces	
	Part Surfaces	[Subtract.Block Dominio.Lateral]
	Type	Symmetry Plane
	Allow Per-Surface Values	FALSE
	Topology	SURFACE
	Tags	[]
`1 Physics Conditions		
`1 VOF Wave Damping Boundary Option	Method	No
+4 Subtract.Block Dominio.Outflow	Index	5
	Interfaces	
	Part Surfaces	[Subtract.Block Dominio.Outflow]
	Type	Pressure Outlet
	Allow Per-Surface Values	FALSE
	Topology	SURFACE
	Tags	[]

+1 Physics Conditions		
+1 Backflow Specification	Direction	Boundary-Normal
	Pressure	Environmental
	Scalars	Specified
+2 Pressure Outlet Option	Option	None
+3 Reference Frame Specification	Option	Lab Frame
+4 Turbulence Specification	Method	Intensity + Viscosity Ratio
`5 VOF Wave Damping Boundary Option	Method	Yes
`2 Physics Values		
+1 Pressure	Method	Field Function
`1 Field Function	Scalar Function	Hydrostatic Pressure of Heavy Fluid of Free Surface
+2 Turbulence Intensity	Method	Constant
`1 Constant	Value	0.01
+3 Turbulent Viscosity Ratio	Method	Constant
`1 Constant	Value	10.0
`4 Volume Fraction	Method	Composite N-1
`1 Composite N-1		
`1 Air	Method	Field Function
`1 Field Function	Scalar Function	Volume Fraction of Light Fluid of Free Surface
+5 Subtract.Block Dominio.SymPlane	Index	6
	Interfaces	
	Part Surfaces	[Subtract.Block Dominio.SymPlane]
	Type	Symmetry Plane
	Allow Per-Surface Values	FALSE
	Topology	SURFACE
	Tags	[]
`1 Physics Conditions		
`1 VOF Wave Damping Boundary Option	Method	No
+6 Subtract.Block Dominio.Top	Index	3
	Interfaces	
	Part Surfaces	[Subtract.Block Dominio.Top]
	Type	Symmetry Plane
	Allow Per-Surface Values	FALSE
	Topology	SURFACE
	Tags	[]
`1 Physics Conditions		
`1 VOF Wave Damping Boundary Option	Method	No
`7 Subtract.Hull.Default	Index	2
	Interfaces	
	Part Surfaces	[Subtract.Hull.Default]
	Type	Wall
	Allow Per-Surface Values	FALSE
	Topology	SURFACE
	Tags	[]
+1 Physics Conditions		
+1 Reference Frame Specification	Option	Region Reference Frame
+2 Shear Stress Specification	Method	No-Slip
+3 Tangential Velocity Specification	Method	Fixed
+4 VOF Wave Damping Boundary Option	Method	No

`-5 Wall Surface Specification	Method	Smooth
`-2 Physics Values		
`-1 Standard Wall Function	E	9.0
	Kappa	0.42
+2 Phase Conditions		
+1 Air	Type	Fluid Region
	Tags	[]
`-1 Physics Conditions		
`-1 Momentum Source Option	Momentum Source Option	None
`-2 Water	Type	Fluid Region
	Tags	[]
`-1 Physics Conditions		
`-1 Momentum Source Option	Momentum Source Option	None
+3 Physics Conditions		
+1 Damping Length Specification	Length Specification Option	REGION
+2 Initial Condition Option	Option	Use Continuum Values
+3 Momentum Source Option	Momentum Source Option	None
+4 Motion Specification Option	Option	Motion Specification
+5 Turbulence Source Option	Turbulence Source Option	None
+6 VOF Wave Zone Option	VOF Wave Zone Option	DAMPING
`-7 Volume Fraction Source Option	Phase Source Term	FALSE
`-4 Physics Values		
+1 Axis	Coordinate System	Laboratory
	Origin	[0.0, 0.0, 0.0] m
	Direction	[0.0, 0.0, 1.0]
+2 Motion Specification	Motion	Stationary
	Reference Frame	Lab Reference Frame
+3 Orientation Manager		
`-1 Local Orientation 1	Coordinate System	Laboratory
	Tags	[]
+1 Principal Axis 1	Method	Constant
`-1 Constant	Value	[1.0, 0.0, 0.0]
`-2 Principal Axis 2	Method	Constant
`-1 Constant	Value	[0.0, 1.0, 0.0]
`-4 VOF Wave Damping Length	Method	Constant
`-1 Constant	Value	0.5 m
+3 Representations		
+1 Latest Surface/Volume	Representation	Volume Mesh
	Tags	[]
+2 Geometry	Tags	[]
+1 Automated Mesh.Remesh	Tags	[]
+2 Input Surface	Tags	[]
`-3 Latest Surface	Tags	[]
`-3 Volume Mesh	Cells	711866
	Interior Faces	2167341
	Vertices	770871
	Tags	[]
+1 Finite Volume Regions		
`-1 Dominio	Cells	711866
	Interior Faces	2167341
	Vertices	770871

	Edges	0
`1 Finite Volume Boundaries		
+-1 Subtract.Block Dominio.Bottom	Faces	1121
+-2 Subtract.Block Dominio.Inflow	Faces	1839
+-3 Subtract.Block Dominio.Lateral	Faces	6349
+-4 Subtract.Block Dominio.Outflow	Faces	1684
+-5 Subtract.Block Dominio.SymPlane	Faces	11550
+-6 Subtract.Block Dominio.Top	Faces	1124
`7 Subtract.Hull.Default	Faces	3368
`2 Cell Sets		
+4 Automation		
+1 Parameters		
+2 Field Functions		
+3 Simulation Operations	Selected	[]
+4 Filters		
+5 Tags		
+6 Stages	Active Stage	[]
+7 Update Events	Event Count	0
	Event Names	
`8 Time Scales		
+5 Contacts		
+6 Parts		
+1 Block Bulb	Metadata	{}
	Index	6
	Color	java.awt.Color[r=112,g=128,b=144]
	Is Shell	FALSE
	Region	[]
	Contacts	[]
	Descriptions	[Root]
	Face Count	12
	Coordinate System	Laboratory
	Corner 1	[0.9, -0.17, -0.1] m,m,m
	Corner 2	[1.05, 0.0, 0.02] m,m,m
	Tags	[]
+-1 Surfaces		
`1 Block Surface	Index	17
	Metadata	{}
	Boundary	[]
	Color	java.awt.Color[r=112,g=128,b=144]
	Tags	[]
`2 Curves		
`1 Block Curve	Index	8
	Tags	[]
+2 Block Coarse	Metadata	{}
	Index	3
	Color	java.awt.Color[r=112,g=128,b=144]
	Is Shell	FALSE
	Region	[]
	Contacts	[]
	Descriptions	[Root]
	Face Count	12

	Coordinate System	Laboratory	
	Corner 1	[-3.1042907625278184, -1.725270681202507, -0.14488504946526598] m,m,m	
	Corner 2	[2.5841539868697696, 0.28123937443595315, 0.09730994271058793] m,m,m	
	Tags	[]	
+-1 Surfaces			
-1 Block Surface	Index	8	
	Metadata	{}	
	Boundary	[]	
	Color	java.awt.Color[r=112,g=128,b=144]	
	Tags	[]	
-2 Curves			
-1 Block Curve	Index	3	
	Tags	[]	
+-3 Block Dominio	Metadata	{}	
	Index	2	
	Color	java.awt.Color[r=112,g=128,b=144]	
	Is Shell	FALSE	
	Region	[]	
	Contacts	[]	
	Descriptions	[Root]	
	Face Count	12	
	Coordinate System	Laboratory	
	Corner 1	[-2.85153644195698, -1.4205946478170892, -0.9225904445015318] m,m,m	
	Corner 2	[2.1953011955100696, 0.933343256884831] m,m,m	0.0,
	Tags	[]	
+-1 Surfaces			
+-1 Bottom	Index	6	
	Metadata	{}	
	Boundary	[]	
	Color	java.awt.Color[r=112,g=128,b=144]	
	Tags	[]	
+-2 Inflow	Index	7	
	Metadata	{}	
	Boundary	[]	
	Color	java.awt.Color[r=112,g=128,b=144]	
	Tags	[]	
+-3 Lateral	Index	3	
	Metadata	{}	
	Boundary	[]	
	Color	java.awt.Color[r=112,g=128,b=144]	
	Tags	[]	
+-4 Outflow	Index	4	
	Metadata	{}	
	Boundary	[]	
	Color	java.awt.Color[r=112,g=128,b=144]	
	Tags	[]	

+-5 SymPlane	Index	5
	Metadata	{}
	Boundary	[]
	Color	java.awt.Color[r=112,g=128,b=144]
	Tags	[]
`6 Top	Index	2
	Metadata	{}
	Boundary	[]
	Color	java.awt.Color[r=112,g=128,b=144]
	Tags	[]
`2 Curves		
`1 Block Curve	Index	2
	Tags	[]
+-4 Block Fine	Metadata	{}
	Index	4
	Color	java.awt.Color[r=112,g=128,b=144]
	Is Shell	FALSE
	Region	[]
	Contacts	[]
	Descriptions	[Root]
	Face Count	12
	Coordinate System	Laboratory
	Corner 1	[-3.1042907625278184, -1.725270681202507, -0.05] m,m,m
	Corner 2	[2.5841539868697696, 0.28123937443595315, 0.05] m,m,m
	Tags	[]
+-1 Surfaces		
`1 Block Surface	Index	9
	Metadata	{}
	Boundary	[]
	Color	java.awt.Color[r=112,g=128,b=144]
	Tags	[]
`2 Curves		
`1 Block Curve	Index	4
	Tags	[]
+-5 Hull	Metadata	{}
	Index	1
	Color	java.awt.Color[r=255,g=0,b=0]
	Is Shell	FALSE
	Region	[]
	Contacts	[]
	Descriptions	[Root]
	Face Count	11366
	Tags	[]
+-1 Surfaces		
`1 Default	Index	1
	Metadata	{}
	Boundary	[]
	Color	java.awt.Color[r=112,g=128,b=144]
	Tags	[]

`2 Curves		
`1 Default	Index	1
	Tags	[]
`6 Subtract	Metadata	{}
	Index	5
	Color	java.awt.Color[r=112,g=128,b=144]
	Is Shell	FALSE
	Region	[Dominio]
	Contacts	[]
	Descriptions	[Root, Automated Mesh.Remesh]
	Face Count	5094
	Operation	Subtract
	Tags	[]
+7 3D-CAD Models		
`1 3D-CAD Model 1	Distinguish Bodies Color Palette	High Contrast Color Palette
	Part Update Method	UPDATE_GEOMETRY
	Tags	[]
+1 Body Groups		
`1 Hull	Name	Hull
	Color	java.awt.Color[r=128,g=128,b=128]
	Opacity	1.0
	Display Resolution	VERY_COARSE
	Tags	[]
+2 Features		
+-1 XY	Error Message	
	Origin	[0.0, 0.0, 0.0] m
	X-Axis	[1.0, 0.0, 0.0]
	Y-Axis	[0.0, 1.0, 0.0]
	Tags	[]
+-2 YZ	Error Message	
	Origin	[0.0, 0.0, 0.0] m
	X-Axis	[0.0, 1.0, 0.0]
	Y-Axis	[0.0, 0.0, 1.0]
	Tags	[]
+-3 ZX	Error Message	
	Origin	[0.0, 0.0, 0.0] m
	X-Axis	[0.0, 0.0, 1.0]
	Y-Axis	[1.0, 0.0, 0.0]
	Tags	[]
+4 Global Origin	Error Message	
	Position	[0.0, 0.0, 0.0]
	Tags	[]
+5 Lab Coordinate System	Error Message	
	Origin	[0.0, 0.0, 0.0]
	X-axis Direction	[1.0, 0.0, 0.0]
	Y-axis Direction	[0.0, 1.0, 0.0]
	Tags	[]
+6 ImportCad 1	Error Message	
	Exchange Settings	{NX=0, STEP=0, SE=0, CGR=0, SW=0, JT=0, ACIS=0, IFC=0, CATIAV5=0, IGES=0, CATIAV4=0, 3DXML=0, CREO=0, INV=0}

	File Name	C:\--FOLDERPATH--\Superlampiao_Hull_V6_NoBulb.3dm
	Tags	[]
+-7 ImportCad 2	Error Message	
	Exchange Settings	{NX=0, STEP=0, SE=0, CGR=0, SW=0, JT=0, ACIS=0, IFC=0, CATIAV5=0, IGES=0, CATIAV4=0, 3DXML=0, CREO=0, INV=0}
	File Name	C:\--FOLDERPATH--\Deform 195.3dm
	Tags	[]
+-8 SewSheetBodies 1	Error Message	
	Sewing Tolerance	0.001 m
	Attempt to Form Solid	TRUE
	Sew Bodies Independently	FALSE
	Tags	[]
+-9 Fill Surface 1	Error Message	
	Set Tangency at all Edges	FALSE
	Tolerance	0.001 m
	Tags	[]
+-10 MirrorBody 1	Error Message	
	Coordinate System Source	Reference Coordinate System
	Normal	[0.0, 1.0, 0.0] m,m,m
	Position	[0.0, 0.0, 0.0] m,m,m
	Action	Copy
	Body Group	FALSE
	Tags	[]
+-11 UniteBodies 1	Error Message	
	Merge Type	TolerantFaces
	Tolerance	1.0E-5 m
	Use Auto-Matching	TRUE
	Transfer Face Names	TRUE
	Transfer Body Names	FALSE
	Tags	[]
`12 MoveBody 1	Error Message	
	Direction Type	Specified
	Distance	0.2 m
	Translation Vector	[0.0, 0.0, -0.11] m,m,m
	Action	Move
	Body Group	FALSE
	Tags	[]
+-3 Design Filters		
`4 Design Parameters		
+8 Operations		
+-1 Subtract	Link Output Part Name	FALSE
	Input Parts	[Block Dominio, Hull]
	Target Part	[Block Dominio]
	Perform CAD Boolean	TRUE
	Output Parts	Manager
	Tags	[]
+-1 Tessellation Options	Tessellation Options	Medium
`2 Tolerance Type	Tolerance Type	Precise
	Tolerance	1.0E-5 m

`2 Automated Mesh	Per-Part Meshing	FALSE
	Mesher Execution Mode	Serial
	Input Parts	[Subtract]
	Perform Local Meshing	FALSE
	Preserve Surface Perimeters	None
	Verbose Output	FALSE
	Tags	[]
+1 Meshers		
+1 Surface Remesher	Meshing Method	Enhanced Quality Triangle
	Perform Curvature Refinement	TRUE
	Perform Proximity Refinement	TRUE
	Perform Compatibility Refinement	FALSE
	Create Aligned Meshes	TRUE
	Minimum Face Quality	0.2
	Field Function based Refinement	[]
+2 Automatic Surface Repair	Connected Surface Count Limit	None
	Connected Surface Size Limit(s)	None
	Minimum Face Quality	0.2
+3 Trimmed Cell Mesher	Core Cell Type	Hexahedra
	Field Function based Refinement	[]
	Coordinate System	Laboratory
	Perform Mesh Alignment	TRUE
	Size Dependent Growth Rates	All cells
	Run Post Mesh Optimizer	TRUE
`4 Prism Layer Mesher	Stretching Function	Geometric Progression
	Distribution Mode	Stretch Factor
+2 Default Controls		
+1 Base Size	Base Size	0.4 m
	Base Size	0.4 m
+2 CAD Projection	Project to CAD	TRUE
+3 Target Surface Size	Size Type	Relative to base
	Percentage of Base	50.0
	Absolute Size	0.2 m
+4 Minimum Surface Size	Size Type	Relative to base
	Percentage of Base	25.0
	Absolute Size	0.1 m
+5 Surface Curvature	Enable Curvature Deviation Distance	FALSE
	# Pts/circle	36.0
	Max # Pts/circle	200.0
	Curvature Deviation Distance	0.01 m
+6 Surface Proximity	Search Floor	0.0 m
	# Points in gap	2.0
	Enable Search Ceiling	FALSE
	Search Ceiling	1.0E10 m
	Search Direction	INSIDE
+7 Surface Growth Rate	Surface Growth Rate	SLOW
	User Specified Value	1.3
+8 Auto-Repair Minimum Proximity	Minimum Proximity	0.01
+9 Maximum Core/Prism Transition Ratio	Limit Cell Size by Prism Layer Thickness	FALSE
+10 Volume Growth Rate	Default Growth Rate	Very Slow
	Surface Growth Rate	Very Slow

+-11 Maximum Cell Size	Size Type	Relative to base
	Percentage of Base	100.0
	Absolute Size	0.4 m
+-12 Mesh Alignment Location	Location	[0.0, 0.0, 0.0] m
`13 Post Mesh Optimization	Optimize Boundary Vertices	FALSE
	Optimize Cell Topology	FALSE
`3 Custom Controls		
+-1 Surface Control Hull	Enable Control	TRUE
	Controls Display Mode	All
	Part Surfaces	[Subtract.Hull.Default]
	Apply Only to Contacting Area	FALSE
	Tags	[]
+-1 Controls		
+-1 Target Surface Size	Target Surface Size	Custom
+-2 Minimum Surface Size	Minimum Surface Size	Parent
+-3 Surface Curvature	Curvature	Parent
+-4 Surface Proximity	Proximity	Parent
+-5 Edge Proximity	Proximity	Parent
+-6 Surface Growth Rate	Surface Growth Rate	Parent
+-7 Surface Remeshing	Surface Remeshing	Parent
+-8 Meshing Method	Meshing Method	Parent
+-9 Prism Layers	Prism Layers	Custom
`1 Customize	Customize Number of Layers	TRUE
	Customize Total Thickness	TRUE
	Customize Distribution	FALSE
	Override Boundary Defaults	FALSE
	Customize Minimum Thickness Percentage	FALSE
	Customize Boundary March Angle	FALSE
	Customize Concave Angle Limit	FALSE
	Customize Convex Angle Limit	FALSE
+-10 Trimmer Surface Growth Rate	Trimmer Surface Growth Rate	Parent
+-11 Remesher Wake Refinement	Specify wake refinement options	FALSE
`12 Trimmer Wake Refinement	Specify Trimmer Wake Refinement Options	FALSE
`2 Values		
+-1 Target Surface Size	Size Type	Relative to base
	Percentage of Base	5.0
	Absolute Size	0.02 m
`2 Custom Prism Values		
+-1 Number of Prism Layers	Number of Prism Layers	10
`2 Prism Layer Total Thickness	Size Type	Relative to base
	Percentage of Base	5.0
	Absolute Size	0.02 m
+-2 Volumetric Control Bulb	Enable Control	TRUE
	Controls Display Mode	All
	Parts	[Block Bulb]
	Tags	[]
+-1 Controls		
+-1 Surface Remesher	Customize Size	TRUE
+-2 Prism Layer Mesher	Customize Number of Layers	TRUE
	Customize Total Thickness	TRUE
	Customize Stretching	FALSE

	Customize Minimum Thickness Percentage	FALSE
	Customize Boundary March Angle	TRUE
	Customize Concave Angle Limit	FALSE
	Customize Convex Angle Limit	FALSE
`-3 Trimmer	Customize Isotropic Size	FALSE
	Customize Anisotropic Size	FALSE
`-2 Values		
+-1 Custom Size	Size Type	Relative to base
	Percentage of Base	3.0
	Absolute Size	0.012000000000000002 m
`-2 Custom Prism Values		
+-1 Boundary March Angle	Boundary March Angle	70.0 deg
+-2 Number of Prism Layers	Number of Prism Layers	15
`-3 Prism Layer Total Thickness	Size Type	Relative to base
	Percentage of Base	5.0
	Absolute Size	0.02 m
+3 Volumetric Control FS Coarse	Enable Control	TRUE
	Controls Display Mode	All
	Parts	[Block Coarse]
	Tags	[]
+-1 Controls		
+-1 Surface Remesher	Customize Size	FALSE
+-2 Prism Layer Mesher	Customize Number of Layers	FALSE
	Customize Total Thickness	FALSE
	Customize Stretching	FALSE
	Customize Minimum Thickness Percentage	FALSE
	Customize Boundary March Angle	FALSE
	Customize Concave Angle Limit	FALSE
	Customize Convex Angle Limit	FALSE
`-3 Trimmer	Customize Isotropic Size	FALSE
	Customize Anisotropic Size	TRUE
`-2 Values		
`-1 Trimmer Anisotropic Size	Relative/Absolute	Relative to base
	Custom X Size	FALSE
	Custom Y Size	FALSE
	Custom Z Size	TRUE
`-1 Relative Z Size	Percentage of Base	10.0
	Absolute Size	0.04 m
`-4 Volumetric Control FS Fine	Enable Control	TRUE
	Controls Display Mode	All
	Parts	[Block Fine]
	Tags	[]
+-1 Controls		
+-1 Surface Remesher	Customize Size	FALSE
+-2 Prism Layer Mesher	Customize Number of Layers	FALSE
	Customize Total Thickness	FALSE
	Customize Stretching	FALSE
	Customize Minimum Thickness Percentage	FALSE
	Customize Boundary March Angle	FALSE
	Customize Concave Angle Limit	FALSE
	Customize Convex Angle Limit	FALSE

`3 Trimmer	Customize Isotropic Size	FALSE
	Customize Anisotropic Size	TRUE
`2 Values		
`1 Trimmer Anisotropic Size	Relative/Absolute	Relative to base
	Custom X Size	TRUE
	Custom Y Size	TRUE
	Custom Z Size	TRUE
+1 Relative X Size	Percentage of Base	5.0
	Absolute Size	0.02 m
+2 Relative Y Size	Percentage of Base	5.0
	Absolute Size	0.02 m
`3 Relative Z Size	Percentage of Base	5.0
	Absolute Size	0.02 m
+9 Descriptions	Number of Children	5
+1 Root	Described Parts	[Hull, Block Dominio, Block Coarse, Block Fine, Subtract, Block Bulb]
+2 Automated Mesh.Remesh	Described Parts	[Subtract]
	Faces	8344
	Vertices	4174
+3 Input Surface		
+4 Latest Surface	Described Parts	[Hull, Block Dominio, Block Coarse, Block Fine, Subtract, Block Bulb]
	Faces	19758
	Vertices	9891
	Preview Mesh Operation Parts	FALSE
`5 Latest Surface/Volume	Described Parts	[Hull, Block Dominio, Block Coarse, Block Fine, Subtract, Block Bulb]
+10 Coordinate Systems		
`1 Laboratory	Tags	[]
`1 Local Coordinate Systems		
+11 Parameterizations		
+12 Tables	Tables	0
+13 Units	Preferred System	Systeme International
+14 Custom Trees	Initial Tree View	[]
+15 Volume Shapes		
+16 Idealizations	Region selection priority	[Dominio Symmetry Idealization 1]
	Idealizations	1
`1 Dominio Symmetry Idealization 1	Regions	[Dominio]
	Generating Region	[Dominio]
	Electric Fields	[]
	Magnetic Fields	[]
	Tags	[]
+17 Color Palettes		
+1 High Contrast Color Palette	Number of Colors	28
	Swatches	[java.awt.Color[r=255,g=0,b=0], java.awt.Color[r=0,g=255,b=0], java.awt.Color[r=0,g=0,b=255], java.awt.Color[r=255,g=255,b=0], java.awt.Color[r=255,g=128,b=0], java.awt.Color[r=160,g=32,b=240], java.awt.Color[r=255,g=255,b=255], java.awt.Color[r=255,g=192,b=203], java.awt.Color[r=189,g=252,b=201], java.awt.Color[r=175,g=238,b=238], java.awt.Color[r=240,g=230,b=140],

			<pre>java.awt.Color[r=255,g=228,b=181], java.awt.Color[r=221,g=160,b=221], java.awt.Color[r=192,g=192,b=192], java.awt.Color[r=219,g=112,b=147], java.awt.Color[r=0,g=201,b=87], java.awt.Color[r=0,g=255,b=255], java.awt.Color[r=255,g=227,b=3], java.awt.Color[r=255,g=176,b=15], java.awt.Color[r=186,g=85,b=211], java.awt.Color[r=128,g=128,b=105], java.awt.Color[r=176,g=48,b=96], java.awt.Color[r=34,g=139,b=34], java.awt.Color[r=95,g=158,b=160], java.awt.Color[r=255,g=215,b=0], java.awt.Color[r=255,g=97,b=3], java.awt.Color[r=143,g=94,b=153], java.awt.Color[r=115,g=74,b=18]]</pre>
	Tags		[]
+-2 Legacy Plot Color Palette	Number of Colors		13
			<pre>[java.awt.Color[r=255,g=0,b=0], java.awt.Color[r=0,g=255,b=0], java.awt.Color[r=0,g=0,b=255], java.awt.Color[r=255,g=200,b=0], java.awt.Color[r=0,g=255,b=255], java.awt.Color[r=0,g=0,b=0], java.awt.Color[r=178,g=0,b=0], java.awt.Color[r=0,g=178,b=0], java.awt.Color[r=0,g=0,b=178], java.awt.Color[r=8,g=46,b=84], java.awt.Color[r=178,g=140,b=0], java.awt.Color[r=160,g=32,b=240], java.awt.Color[r=92,g=36,b=110]]</pre>
	Swatches		
	Tags		[]
^-3 Siemens Color Palette	Number of Colors		30
			<pre>[java.awt.Color[r=15,g=120,b=155], java.awt.Color[r=229,g=192,b=76], java.awt.Color[r=219,g=83,b=90], java.awt.Color[r=96,g=106,b=117], java.awt.Color[r=226,g=137,b=77], java.awt.Color[r=127,g=70,b=100], java.awt.Color[r=104,g=153,b=98], java.awt.Color[r=232,g=170,b=184], java.awt.Color[r=165,g=143,b=111], java.awt.Color[r=127,g=178,b=172], java.awt.Color[r=0,g=85,b=125], java.awt.Color[r=168,g=133,b=45], java.awt.Color[r=153,g=49,b=64], java.awt.Color[r=50,g=57,b=63], java.awt.Color[r=170,g=97,b=48], java.awt.Color[r=81,g=45,b=67], java.awt.Color[r=71,g=102,b=66], java.awt.Color[r=173,g=104,b=127], java.awt.Color[r=112,g=94,b=75], java.awt.Color[r=77,g=124,b=115], java.awt.Color[r=70,g=170,b=193], java.awt.Color[r=234,g=208,b=150], java.awt.Color[r=234,g=172,b=185], java.awt.Color[r=170,g=180,b=188], java.awt.Color[r=239,g=194,b=163], java.awt.Color[r=186,g=131,b=165], java.awt.Color[r=165,g=198,b=158], java.awt.Color[r=239,g=203,b=217], java.awt.Color[r=204,g=188,b=168], java.awt.Color[r=180,g=214,b=208]]</pre>
	Swatches		
	Tags		[]
+-18 Data Set Functions	Data Directory		function_data
	In-core surface FFTs		FALSE
+-19 User Code			
+-20 Data Focus			

+21 Layouts		
-1 default		
+22 Data Mappers	Use Legacy Volume Mapper	FALSE
+23 Motions	Motion Preview Time	0.0 s
	Visual Motion Transforms	[]
	Motion Preview Parts	[]
-1 Stationary	Tags	[]
+24 Reference Frames		
-1 Lab Reference Frame	Tags	[]
+25 Screenplays		
+26 Derived Parts	Derived Parts	1
-1 FreeSurface Isosurface	Parts	[Dominio]
	Scalar Field	Volume Fraction of Water
	Mode	ISOVALUE_SINGLE
	Tags	[]
-1 Value	Isovalue	0.5
+27 Summaries		
+28 Monitors	Monitors	11
	Monitors To Print	[Continuity, X-momentum, Y-momentum, Z-momentum, Tke, Tdr, Air, Lift Fz (N) Monitor, Resistance Fx (N) Monitor]
	Output Direction	Horizontal
	Heading Print Frequency	10
+1 Iteration	Maximum Plot Samples	5000
	Tags	[]
+2 Lift Fz (N) Monitor	Report	[Lift Fz (N)]
	Enabled	TRUE
	Value Type	Total Value
	Trigger	Time Step
	Normalization Option	Off
	Maximum Plot Samples	5000
	Tags	[]
-1 Time-Step Frequency	Frequency	1
	Start	0
	Enable Stop	FALSE
	Stop	0
+3 Physical Time	Maximum Plot Samples	5000
	Tags	[]
+4 Resistance Fx (N) Monitor	Report	[Resistance Fx (N)]
	Enabled	TRUE
	Value Type	Total Value
	Trigger	Time Step
	Normalization Option	Off
	Maximum Plot Samples	5000
	Tags	[]
-1 Time-Step Frequency	Frequency	1
	Start	0
	Enable Stop	FALSE
	Stop	0
+29 Reports	Reports	2
+1 Lift Fz (N)	Units	N

	Coordinate System	Laboratory
	Direction	[0.0, 0.0, 1.0]
	Force Option	Pressure + Shear
	Reference Pressure	0.0 Pa
	Number of Bands	0
	Parts	[Dominio: Subtract.Hull.Default]
	Representation	Volume Mesh
	Smooth Values	FALSE
	Account for Idealization	FALSE
	Tags	[]
`-2 Resistance Fx (N)	Units	N
	Coordinate System	Laboratory
	Direction	[-1.0, 0.0, 0.0]
	Force Option	Pressure + Shear
	Reference Pressure	0.0 Pa
	Number of Bands	0
	Parts	[Dominio: Subtract.Hull.Default]
	Representation	Volume Mesh
	Smooth Values	FALSE
	Account for Idealization	FALSE
	Tags	[]
+30 Solvers	Cache Solver Settings	TRUE
+1 Implicit Unsteady	Time-Step	<1 m> / <0.9 m/s> * 1. / 250.
	Temporal Discretization	1st-order
	Freeze Time	FALSE
`-1 Stopping Criteria	Verbose	FALSE
+2 Partitioning	Solver Frozen	FALSE
	Partitioning Method	Per-Continuum
+3 Wall Distance	Solver Frozen	FALSE
	Verbosity	0
	Minimum Tree Size Threshold	500000
+4 VOF Wave Zone Distance	Solver Frozen	FALSE
	Verbosity	0
	Minimum Tree Size Threshold	500000
+5 Segregated Flow	Pressure Gradient Inverse Distance Weighting	TRUE
	Implicit Scheme	SIMPLE
	Freeze Flow	FALSE
	Reconstruction Frozen	FALSE
	Reconstruction Zeroed	FALSE
	Enable Enhanced Stability Treatment	FALSE
	Temporary Storage Retained	FALSE
	Pressure Corrections: Bad Cell Minimum Scaling	0.8
	Pressure Corrections: Acceptable Cell Volume Change	1
	Velocity Corrections: Maximum Unlimited Velocity	20.0 m/s
	Velocity Corrections: Acceptable Velocity Increase Rate [<1]	0.15
	Continuity Initialization	FALSE
+1 Velocity	Under-Relaxation Factor	0.8

	Dynamic Local Under-Relaxation	FALSE
+-1 Under-Relaxation Factor Ramp	Ramp Method	No Ramp
^-2 AMG Linear Solver	Max Cycles	30
	Verbosity	NONE
	Enable Direct Solver	FALSE
	Maximum Direct Solver Equations	32
	Convergence Tolerance	0.1
	Epsilon	0.0
	Cycle Type	Flex Cycle
	Group Size Control	Auto
	Group Size	4
	Relaxation Scheme	Gauss-Seidel
	Acceleration Method	None
	Scaling	Disabled
^-1 Flex Cycle	Restriction Tolerance	0.9
	Prolongation Tolerance	0.5
	Sweeps	1
^-2 Pressure	Number of Non-Orthogonality Correctors	0
	Under-Relaxation Factor	0.2
	Pressure Reference Location	Automatic Selection
+-1 Under-Relaxation Factor Ramp	Ramp Method	No Ramp
^-2 AMG Linear Solver	Max Cycles	30
	Verbosity	NONE
	Enable Direct Solver	FALSE
	Maximum Direct Solver Equations	32
	Convergence Tolerance	0.1
	Epsilon	0.0
	Cycle Type	V Cycle
	Group Size Control	Auto
	Group Size	4
	Relaxation Scheme	Gauss-Seidel
	Acceleration Method	Conjugate Gradient
	Scaling	Auto
^-1 V Cycle	Pre-Sweeps	1
	Post-Sweeps	1
	Max Levels	50
+6 Segregated VOF	Use Multi-Step	FALSE
	Temporary Storage Retained	FALSE
	Solution Strategy	SINGLE_STEP
	Update Frequency	Every Iteration
^-1 Single-Step	Under-Relaxation Factor	0.9
+-1 Under-Relaxation Factor Ramp	Ramp Method	No Ramp
^-2 AMG Linear Solver	Max Cycles	30
	Verbosity	NONE
	Enable Direct Solver	FALSE
	Maximum Direct Solver Equations	32
	Convergence Tolerance	0.1
	Epsilon	0.0
	Cycle Type	V Cycle
	Group Size Control	Auto
	Group Size	4

	Relaxation Scheme	Gauss-Seidel
	Acceleration Method	None
	Scaling	Disabled
`-1 V Cycle	Pre-Sweeps	1
	Post-Sweeps	1
	Max Levels	50
+-7 K-Epsilon Turbulence	Solver Frozen	FALSE
	Reconstruction Frozen	FALSE
	Reconstruction Zeroed	FALSE
	Temporary Storage Retained	FALSE
	Under-Relaxation Factor	0.8
	Boundary Layer Initialization	FALSE
+1 Under-Relaxation Factor Ramp	Ramp Method	No Ramp
`-2 AMG Linear Solver	Max Cycles	30
	Verbosity	NONE
	Enable Direct Solver	FALSE
	Maximum Direct Solver Equations	32
	Convergence Tolerance	0.1
	Epsilon	0.0
	Cycle Type	Flex Cycle
	Group Size Control	Auto
	Group Size	4
	Relaxation Scheme	Gauss-Seidel
	Acceleration Method	None
	Scaling	Disabled
`-1 Flex Cycle	Restriction Tolerance	0.9
	Prolongation Tolerance	0.5
	Sweeps	1
`-8 K-Epsilon Turbulent Viscosity	Solver Frozen	FALSE
	Under-Relaxation Factor	1.0
	Maximum Ratio	100000.0
+31 Stopping Criteria	Verbose	FALSE
+1 Maximum Inner Iterations	Enabled	TRUE
	Maximum Inner Iterations	5
	Logical Rule	Or
	Criterion Satisfied	TRUE
	Tags	[]
+2 Maximum Physical Time	Enabled	TRUE
	Maximum Physical Time	785.0 s
	Logical Rule	Or
	Criterion Satisfied	TRUE
	Tags	[]
+3 Maximum Steps	Enabled	FALSE
	Maximum Steps	1000
	Logical Rule	Or
	Criterion Satisfied	TRUE
	Tags	[]
`-4 Stop File	Enabled	TRUE
	Stop Inner Iterations	TRUE
	Path	ABORT
	Logical Rule	Or

	Criterion Satisfied	FALSE
	Tags	[]
+-32 Solution Histories		
+-33 Solution Views		
^-1 Current Solution	Iteration	883125
	Time Step	176625
	Solution Time	7.849.999.999.979.110
	Tags	[]
+-34 Layout Views		
Solution		
Accumulated CPU Time over all processes (s)	4.464.847.213.008.580	
Elapsed Time (s)	6.072.924.419.986.340	1686923449996
Time Level	176625	
Solution Time	7.849.999.999.979.110	