

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO TECNOLÓGICO DE JOINVILLE  
CURSO DE ENGENHARIA NAVAL

ATILIO DA ROSA SCUSSEL

QUANTIFICAÇÃO DE INCERTEZAS NAS PREDIÇÕES DE INÍCIO DE CAVITAÇÃO  
NO ENSAIO DO PROPULSOR DUISBURG P1570

Joinville

2022

ATILIO DA ROSA SCUSSEL

QUANTIFICAÇÃO DE INCERTEZAS NAS PREDIÇÕES DE INÍCIO DE CAVITAÇÃO  
DO CASO DE TESTE DO PROPULSOR DUISBURG P1570

Trabalho apresentado como requisito para obtenção do título de bacharel no Curso de Graduação em Engenharia Naval do Centro Tecnológico de Joinville da Universidade Federal de Santa Catarina.

Orientador: Dr. Thiago Pontin Tancredi

Coorientador: Dr. Artur Lidtke

Joinville

2022

Ficha de identificação da obra elaborada pelo autor, através do Programa de  
Geração Automática da Biblioteca Universitária da UFSC.

Scussel, Atilio

QUANTIFICAÇÃO DE INCERTEZAS NAS PREDIÇÕES DE INÍCIO DE  
CAVITAÇÃO DO CASO DE TESTE DO PROPULSOR DUISBURG P1570 /  
Atilio Scussel ; orientador, Thiago Pontin Tancredi,  
coorientador, Artur Lidtke, 2022. 150 p.

Trabalho de Conclusão de Curso (graduação) -  
Universidade Federal de Santa Catarina, Campus Joinville, Graduação  
em Engenharia Naval, Joinville, 2022.

Inclui referências.

1. Engenharia Naval. 2. Cavitação. 3. Propulsor Duisburg.  
4. Análise de incertezas. I. Pontin Tancredi,  
Thiago. II. Lidtke, Artur. III. Universidade Federal de Santa  
Catarina. Graduação em Engenharia Naval. IV. Título.

ATILIO DA ROSA SCUSSEL

QUANTIFICAÇÃO DE INCERTEZAS NAS PREDIÇÕES DE INÍCIO DE CAVITAÇÃO  
DO CASO DE TESTE DO PROPULSOR DUISBURG P1570

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de bacharel em Engenharia Naval, na Universidade Federal de Santa Catarina, Centro Tecnológico de Joinville.

Joinville (SC), 14 de Março de 2022.

**Banca Examinadora:**

---

Dr. Thiago Pontin Tancredi  
Orientador/Presidente

---

Dr. Lucas Weihmann  
Membro  
Universidade Federal de Santa Catarina

---

Dr. Filipe Dutra da Silva  
Membro  
Universidade Federal de Santa Catarina

Dedico este trabalho a minha família, amigos e orientadores.

## **AGRADECIMENTOS**

Agradeço aos meus pais e irmã por todo o suporte dado durante minha graduação e por sempre incentivarem minha educação.

Agradeço meu orientador Dr. Thiago Pontin Tancredi por ter sido desde o início receptivo, acreditar no tema escolhido e aceitar me orientar. Ainda, agradeço-o pelo auxílio durante a execução do projeto e pelas discussões e ensinamentos sobre o tema desde antes do início da pesquisa.

Agradeço também meu coorientador Dr. Artur Lidtke pela proposta de tema, pelo auxílio e suporte dado durante o desenvolvimento da tese e pela oportunidade de poder me desenvolver ainda mais com trabalho realizado e a orientação recebida.

Como sabemos, existem incógnitas conhecidas. Há coisas que sabemos que sabemos. Também sabemos que existem incógnitas desconhecidas. Isso quer dizer que sabemos que existem algumas coisas que não sabemos. Mas também existem incógnitas desconhecidas, aquelas que não sabemos que não conhecemos.

(RUMSFEL, 2002 apud SULLIVAN, 2015, p. 49, tradução nossa).

## RESUMO

A análise de sensibilidade é vital para a verificação de modelos preditivos, além de contribuir para o aprimoramento de novos produtos, pois permite determinar os efeitos que as incertezas de projeto causam nas características de desempenho analisadas. Nesse contexto, o presente estudo busca identificar e quantificar as incertezas presentes nas análises de início de cavitação dos tipos nuvem e folha do propulsor Duisburg P1570 operando totalmente submerso. Para a determinação da influência das variáveis no desempenho do propulsor, propõem-se a realização de simulações CFD utilizando o programa ReFRESHCO e a abordagem não-intrusiva de Galerkin. Por fim, apresenta-se uma estrutura em Python, a qual estabelece um modelo interpolador (baseado na Amostragem por Hipercubo Latino) capaz de reduzir consideravelmente o custo computacional das análises. Os módulos em Python permitem ainda calcular os índices de Sobol (incertezas paramétricas), as incertezas de entrada e as incertezas devido ao refino de malha (incertezas de discretização). As variáveis de entrada analisadas foram: a rotação do hélice (RPM), a pressão de vapor da água ( $P_v$ ) e o refino de malha (número de elementos), enquanto os parâmetros de desempenho analisados foram: o coeficiente de empuxo ( $K_T$ ), o coeficiente de torque ( $K_Q$ ), a eficiência do propulsor ( $\eta$ ) e o índice de cavitação ( $C_{VINDEX}$ ). Os resultados mostraram que as variações observadas nos resultados dos parâmetros  $K_T$ ,  $K_Q$  e  $\eta$  devem-se pouco às incertezas de discretização (5%) e à incerteza no valor de  $P_v$  (menos de 1%), sendo majoritariamente causadas pela incerteza na rotação (mais de 94%). Entretanto, o  $C_{VINDEX}$  apresenta um resultado diferente, sendo majoritariamente afetado pelas incertezas de discretização (99%) e pouco afetado pelas incertezas de entrada (1%).

**Palavras-chave:** Quantificação de incertezas. Cavitação. Propulsor. Simulação numérica. Propulsor Duisburg.

## ABSTRACT

The sensitivity analysis is relevant for studies of verification and improvement of product, since it is possible to obtain the effects that the analyzed variables have on the outputs, leading to more efficient and effective solutions. The present study seeks to identify and quantify the uncertainties present in the analysis of cloud and sheet cavitation onset for the Duisburg propeller test case. It is proposed the use of CFD simulations using the ReFRESKO software and Galerkin's non-intrusive approach. Finally, a framework in Python was developed to establish a surrogate model based on Latin Hypercube Sampling (LHS), which considerably reduces the computational cost of the analysis. Python modules also allow calculating Sobol indices (parametric uncertainties), input uncertainties and uncertainties due to mesh refinement (discretization uncertainties). The input variables analyzed were: the propeller rotation per minute (RPM), the water vapor pressure ( $P_V$ ) and the mesh refinement (number of elements), while the performance parameters analyzed were: the thrust coefficient of the propeller ( $K_T$ ), the torque coefficient ( $K_Q$ ), the propellant efficiency ( $\eta$ ) and the cavitation index ( $C_{AVINDEX}$ ). The results showed that the parameters  $K_T$ ,  $K_Q$  and  $\eta$  are little affected by discretization uncertainties (5%) and that they are strongly affected by rotation (more than 94%), but not by water vapor pressure  $P_V$  (less than 1%). However, the cavitation index presents a different result, being strongly affected by discretization uncertainties (99%) and less affected (1%) by input uncertainties.

**Keywords:** Uncertainty quantification. Cavitation. Propeller. Numerical simulation. Duisburg propeller.

## LISTA DE FIGURAS

Figura 1 - Tipos de cavitação .....	24
Figura 2 - Representação das etapas para a obtenção das variáveis de saída. ....	28
Figura 3 - Fluxograma referente ao desenvolvimento do trabalho. ....	34
Figura 4 - Propulsor Duisburg P1570 e seu modelo CAD. ....	36
Figura 5 - Representação do experimento do propulsor Duisburg. ....	36
Figura 6 - Modelo do túnel utilizado na simulação numérica em milímetros. ....	37
Figura 7 - Diagrama de cavitação para 2500 RPM. ....	37
Figura 8 - Diagrama de águas abertas para o propulsor Duisburg a 2500 RPM. ....	38
Figura 9 - Visualização 3D do ambiente numérico de simulação. ....	39
Figura 10 - Etapas para a realização das simulações numéricas. ....	45
Figura 11 - Região de análise no diagrama de início de cavitação. ....	47
Figura 12 - Região de análise no diagrama de águas abertas. ....	47
Figura 13 – Módulos desenvolvidos e suas funções principais. ....	48
Figura 14 - Comparação do $KT_{mean}$ . ....	51
Figura 15 - Comparação do $KQ_{mean}$ . ....	51
Figura 16 - Comparação do $KQ_{s_{mean}}$ . ....	52
Figura 17 - Comparação do $\eta_{mean}$ . ....	52
Figura 18 - Comparação de P entre as malhas $f_{0,325}$ e $f_{0,700}$ para o caso 0. ....	54
Figura 19 - Comparação de P entre as malhas $f_{0,325}$ e $f_{0,700}$ para o caso 1. ....	54
Figura 20 - Comparação de P entre as malhas $f_{0,325}$ e $f_{0,700}$ para o caso 2. ....	54
Figura 21 - Comparação de P entre as malhas $f_{0,325}$ e $f_{0,700}$ para o caso 3. ....	55
Figura 22 - Comparação de P entre as malhas $f_{0,325}$ e $f_{0,700}$ para o caso 4. ....	55
Figura 23 - Comparação de $C_P$ entre as malhas $f_{0,325}$ e $f_{0,700}$ para o caso 0. ....	56
Figura 24 - Comparação de $C_P$ entre as malhas $f_{0,325}$ e $f_{0,700}$ para o caso 1. ....	57
Figura 25 - Comparação de $C_P$ entre as malhas $f_{0,325}$ e $f_{0,700}$ para o caso 2. ....	57
Figura 26 - Comparação de $C_P$ entre as malhas $f_{0,325}$ e $f_{0,700}$ para o caso 3. ....	57
Figura 27 - Comparação de $C_P$ entre as malhas $f_{0,325}$ e $f_{0,700}$ para o caso 4. ....	58
Figura 28 – Pressão atuante na superfície do hélice e início de cavitação. ....	60
Figura 29 - Sequência de análise utilizada. ....	64
Figura 30 - Índices de Sobol para $KT_{mean}$ . ....	65

Figura 31 - Índices de Sobol para KQ_mean. ....	65
Figura 32 - Índices de Sobol para eta_mean.....	66
Figura 33 - Índices de Sobol para CavINDEX.....	66
Figura 34 - União dos índices de Sobol e incertezas de discretização (KT_mean)...	67
Figura 35- União dos índices de Sobol e incertezas de discretização (KQ_mean) ...	67
Figura 36- União dos índices de Sobol e incertezas de discretização (eta_mean) ...	68
Figura 37- União dos índices de Sobol e incertezas de discretização (CavINDEX) .....	68
Figura 38 - Índices de Sobol combinados. ....	70
Figura 39 – CDF das variáveis de saída. ....	71
Figura 40 - CDF das variáveis de saída em porcentagem em relação à média. ....	71
Figura 41 - Modelo interpolador para K <sub>T</sub> .....	121
Figura 42 - Modelo interpolador ara K <sub>Q</sub> .....	121
Figura 43 - Modelo interpolador para η. ....	122
Figura 44 - Modelo interpolador para Cav <sub>index</sub> .....	122

## LISTA DE QUADROS

Quadro 1 – Tipos de instalação propulsora.....	20
--	----

## LISTA DE TABELAS

Tabela 1 - Parâmetros do propulsor Duisburg P1570. ....	35
Tabela 2 – Variáveis definidas com base no experimento de Golf (2018). ....	38
Tabela 3 – Variáveis de saída. ....	39
Tabela 4 – Número de elementos de cada malha estudada. ....	40
Tabela 5 – Simulações computacionais e seus respectivos casos. ....	42
Tabela 6 – Variáveis de saída para o caso 0. ....	61
Tabela 7 – Variáveis de saída para o caso 1. ....	61
Tabela 8 – Variáveis de saída para o caso 2. ....	61
Tabela 9 – Variáveis de saída para o caso 3. ....	62
Tabela 10 – Variáveis de saída para o caso 4. ....	62
Tabela 11 - Limite superior e inferior para $K_T$ e $C_{PNmédio}$ . ....	62
Tabela 12 – Arquivo csv utilizado para análise de incertezas numéricas. ....	63
Tabela 13 - Arquivo csv dos 150 pontos avaliados na simulação numérica. ....	123
Tabela 14 - Índices de Sobol combinados. ....	128

## LISTA DE ABREVIATURAS E SIGLAS

Re: Reynolds

CFD: Dinâmica do fluido computacional

$K_T$ : Coeficiente de empuxo do hélice

$K_Q$ : Coeficiente de torque

$\eta$ : Eficiência do propulsor

J: Coeficiente de avanço

T: Empuxo

Q: Torque

n: Velocidade de rotação por segundo

$V_A$ : Velocidade de avanço

N: Newtons

$\rho$ : Densidade do fluido

D: Diâmetro do propulsor

$\sigma$ : Número de cavitação

$P_\infty$ : Pressão antes de o fluido ser perturbado

$P_V$ : Pressão de vapor do líquido

$P_L$ : Pressão local do fluido

$V_{ENTRADA}$ : Variância do valor de entrada

$V_{SAÍDA}$ : Variância do valor de saída

$S_i$ : Índice de Sobol

HOT: Termos de alta ordem

$E(SAÍDA|ENTRADA)$ : Valor esperado de uma variável de saída dado uma variável de entrada

SST: Transporte de tensão cisalhante

$\kappa$ : Energia cinética turbulenta

$C_P$ : Coeficiente de pressão

$C_{P_{Nmin}}$ : Coeficiente de pressão mínimo

r: Raio avaliado

R: Raio total do propulsor

P: Passo do hélice

$P_{0,7}$ : Passo do hélice no comprimento de corda quando  $r/R$  0,7

$A_E$ : Área expandida do hélice  
 $A_0$ : Área real do hélice  
 $C_{0,7}$ : Comprimento de corda do hélice quando  $r/R$  0,7  
 $\theta_{eff}$ : Ângulo de skew efetivo  
 $h(-)$ : Número de elementos contido na malha  
 $P_{REF}$ : Pressão de referência  
AFM: Modelo de referência absoluta  
 $U_{VAL}$ : Incerteza de validação  
 $U_{NUM}$ : Incerteza numérica  
 $U_{EXP}$ : Incerteza experimental  
 $U_{PAR}$ : Incerteza paramétrica  
 $U_{ITER}$ : Incerteza de iteração  
 $U_{DIS}$ : Incerteza de discretização  
E: Erro de comparação de validação  
S: Predição numérica  
DE: Resultado experimental  
CDF: Função de distribuição acumulada  
IC: Intervalo de confiança  
CAD: Desenho auxiliado por computador  
SSH: Secure Shell  
VOF: Método de volume de fluido  
RANS: Reynolds-Average Navier-Stokes

## SUMÁRIO

<b>1. INTRODUÇÃO</b> .....	<b>17</b>
1.1. OBJETIVOS .....	18
<b>1.1.1. Objetivo Geral</b> .....	<b>19</b>
<b>1.1.2. Objetivos Específicos</b> .....	<b>19</b>
<b>2. FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>20</b>
2.1 TIPOS DE PROPULSÃO E O FENÔMENO DE CAVITAÇÃO .....	20
2.2 ESTUDO DE INCERTEZAS E VARIÁVEIS DE ANÁLISE .....	26
2.3 ÍNDICES DE SOBOL .....	29
2.4 OS MODELOS DE TRANSIÇÃO LAMINAR-TURBULENTO .....	30
2.5 O PROPULSOR DUISBURG P1570 .....	32
<b>3. METODOLOGIA</b> .....	<b>34</b>
3.1 MODELAGEM .....	35
<b>3.1.1 Modelo geométrico do propulsor Duisburg P1570</b> .....	<b>35</b>
3.2 VARIÁVEIS E CONDIÇÕES DE CONTORNO .....	37
3.3 TOPOLOGIA DA MALHA E CONFIGURAÇÃO NUMÉRICA .....	40
3.4 ANÁLISE DOS RESULTADOS .....	41
<b>3.4.1 Elaboração dos arquivos CSV para a análise de incertezas</b> .....	<b>42</b>
<b>3.4.2 Realização de simulações computacionais</b> .....	<b>42</b>
<b>3.4.3 Validação dos resultados</b> .....	<b>45</b>
3.5 INCERTEZAS DE ENTRADA .....	46
3.6 DESENVOLVIMENTO E VALIDAÇÃO DA ESTRUTURA NUMÉRICA .....	47
<b>4. RESULTADOS</b> .....	<b>50</b>
4.1 VERIFICAÇÃO E VALIDAÇÃO DOS MÓDULOS EM PYTHON .....	50
4.2 RESULTADOS NUMÉRICOS E INCERTEZAS DE DISCRETIZAÇÃO .....	53
4.3 AVALIAÇÃO E VERIFICAÇÃO DOS RESULTADOS CFD .....	59
4.4 VARIÁVEIS DE SAÍDA .....	63
4.5 AVALIAÇÃO DOS ÍNDICES DE SOBOL .....	64
4.6 INCERTEZAS DE ENTRADA E DE DISCRETIZAÇÃO .....	67
4.7 EFEITOS DA AMOSTRAGEM .....	70
<b>5. CONCLUSÃO</b> .....	<b>73</b>
<b>REFERÊNCIAS</b> .....	<b>75</b>

APÊNDICE A – ARQUIVO CONTROLS.XML.....	79
APÊNDICE B – CAV_INDEXCALCS.PY.....	95
APÊNDICE C – CONVERGÊNCIA DAS SIMULAÇÕES NUMÉRICAS.....	97
APÊNDICE D – INCERTEZAS DE DISCRETIZAÇÃO NO DESEMPENHO.....	102
APÊNDICE E – RELAÇÃO ENTRE $\Sigma$ E $C_{PN}$ .....	107
APÊNDICE F – DIAGRAMA DE INÍCIO DE CAVITAÇÃO.....	108
APÊNDICE G – RESPONSESURFACE.PY.....	111
APÊNDICE H – MODELOS INTERPOLADORES.....	121
APÊNDICE I – ARQUIVO CSV COM TODOS OS PONTOS ANALISADOS.....	123
APÊNDICE J – ÍNDICES DE SOBOL COMBINADOS.....	128
ANEXO A – MÓDULO PREPARECASES.PY.....	129
ANEXO B – MÓDULO PROCESSCASES.PY.....	133
ANEXO C – MÓDULO CAV_INDEXCALCS.PY.....	141
ANEXO D – RUN_NUA.PY.....	144
ANEXO E – POSTPROCESSOR_LHS.PY.....	147

## 1. INTRODUÇÃO

O transporte marítimo é um dos modais mais antigos da história, sendo atualmente responsável por mais de 90% do volume das importações e exportações nacionais (CONFEDERAÇÃO NACIONAL DOS TRANSPORTES - CNT, 2006) e mais de 80% de todo o volume do comércio mundial (CONFERÊNCIA DAS NAÇÕES UNIDAS SOBRE COMÉRCIO E DESENVOLVIMENTO – UNCTAD, 2019). Nesse contexto, as embarcações mercantes e seus subsistemas desempenham funções essenciais para a integração da economia global.

De acordo com Leme (2021), uma embarcação pode ser descrita como um sistema resultante da interação de diversos subsistemas, tais como: elétrico, de refrigeração, de iluminação, de controle, propulsivo, estrutural, entre outros. Uma vez que esses subsistemas apresentam sempre um certo grau de interdependência, o funcionamento eficiente de cada componente é vital para assegurar que os requisitos do cliente sejam alcançados.

Devido à grande influência no desempenho da embarcação (HERDZIK, 2013), em especial no consumo de combustível, o sistema propulsivo tem grande relevância no projeto de um navio. Idealmente, os propulsores devem produzir um alto valor de empuxo aliado a um baixo requisito de potência do maquinário (WATSON, 1998). Assim, estudos sobre o desempenho desses componentes são relevantes, uma vez que potencializam seleções eficientes, contribuem para melhor performance da embarcação e levam à um menor consumo de combustível.

Entre os diferentes fenômenos associados ao desempenho de um propulsor, a cavitação é, em muitos casos, um importante fator limitador para o projeto. Pois, além de reduzir a eficiência do propulsor, ainda ocasiona a erosão das pás, ruído e vibração. Nesse sentido, Wielgosz et al. (2019) afirmam:

O entendimento sobre o comportamento de cavitação de propulsores é de crítica importância para projetistas navais já que tal fenômeno dita diversos limites devido ao aumento do risco de erosão e aos níveis inaceitáveis de ruído e vibração que tal fenômeno pode causar[...] (WIELGOSZ et al., 2019, p. 1, tradução nossa).

Paradoxalmente, estudos nessa área ainda são raros e fortemente dependentes de resultados empíricos, cujas incertezas associadas a modelagem do fenômeno físico ainda permanecem como tópicos a serem estudados.

Visando aprofundar a discussão sobre o tema, este trabalho avalia os efeitos que as incertezas associadas às variáveis de entrada exercem nos resultados de uma análise de cavitação. Mais especificamente, este estudo se concentra na quantificação das incertezas presentes no início da cavitação de propulsores convencionais do tipo hélice operando totalmente submersos, avaliando a sensibilidade das predições realizadas por simulações computacionais (CFD).

Em vista da importância que a cavitação exerce na eficiência do propulsor e, conseqüentemente, na eficiência energética da embarcação, a quantificação de incertezas exerce grande influência na robustez e lucratividade do projeto. Ademais, dado que o nível de ruído irradiado durante a cavitação é uma ordem de magnitude superior ao nível de ruído de um fluxo não-cavitante (KUIPER, 2012), a cavitação é um tema especialmente importante para navios militares (SILVA, 2016).

Além disso, como afirma NASA (2014), o gerenciamento de erros e incertezas devido à falta de conhecimento em parâmetros de um determinado problema fluídico é uma importante tendência para estudos que envolvam simulações computacionais.

O desenvolvimento realizado é descrito em cinco capítulos, sendo o primeiro a introdução, a qual define o contexto e os objetivos da pesquisa; o segundo, apresenta o referencial teórico que fundamenta o trabalho; o terceiro contempla a metodologia e a modelagem realizada; o quarto apresenta e discute os resultados obtidos; enquanto no último capítulo são descritas as principais conclusões e as recomendações de trabalhos futuros.

## 1.1. OBJETIVOS

Para mitigar as lacunas existente em estudos referentes ao início de cavitação de propulsores navais, propõe-se os objetivos descritos nesta seção.

### **1.1.1. Objetivo Geral**

Quantificar as incertezas que os dados de entrada de uma simulação CFD destinada a avaliar o início da cavitação em propulsores navais do tipo hélice operando totalmente submersos exercem nos resultados da simulação.

### **1.1.2. Objetivos Específicos**

- Identificar e quantificar as incertezas dos parâmetros de uma análise CFD de um propulsor naval operando totalmente submerso;
- Desenvolver uma estrutura numérica capaz de quantificar a influência das incertezas associadas às variáveis de entrada nas variáveis de saída em um problema numérico;
- Quantificar as influências que as incertezas na entrada exercem nos resultados numa simulação CFD destinada a avaliar o início da cavitação em propulsores navais do tipo hélice operando totalmente submersos;
- Aplicar o procedimento desenvolvido para avaliar as incertezas associadas ao ensaio de cavitação do propulsor Duisburg P1570, estudando as cavitações dos tipos folha e nuvem.

## 2. FUNDAMENTAÇÃO TEÓRICA

Para embasar o desenvolvimento realizado, este capítulo aborda os conceitos essenciais associados à pesquisa, bem como a revisão bibliográfica sobre o tema. Inicialmente, são descritos os fundamentos do fenômeno físico estudado, além dos conceitos referentes ao estudo de incertezas presentes em modelos numéricos. São expostos, também, os estudos e modelos matemáticos destinados ao cálculo e validação da quantificação de incertezas. Por fim, são citados estudos já existentes sobre o propulsor Duisburg, juntamente com as características geométricas desse propulsor.

### 2.1 TIPOS DE PROPULSÃO E O FENÔMENO DE CAVITAÇÃO

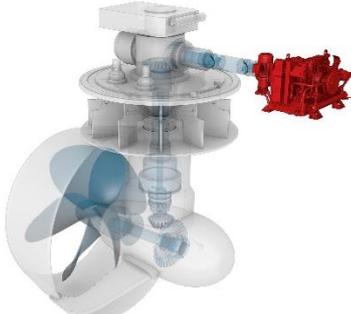
Em linhas gerais, a cavitação descreve a transição de uma região de um escoamento fluídico para o estado gasoso, induzindo a formação de bolhas ao longo do fluxo. Embora a cavitação seja similar ao cozimento, no cozimento ocorre o aumento da temperatura local do fluido, já na cavitação ocorre exclusivamente a redução da pressão local do fluido.

Já a pressão local do fluido, por sua vez, é fortemente afetada pelas características geométricas e condições de operação do propulsor. Dentre os diferentes tipos de instalações propulsoras (Quadro 1), Molland et al. (2011) destacam que o hélice é um importante elemento presente na grande maioria das instalações.

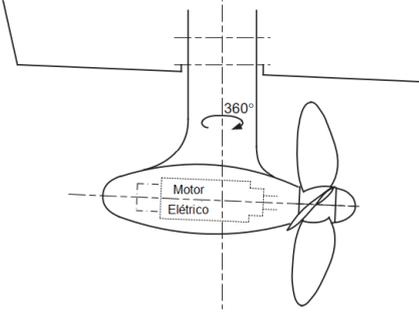
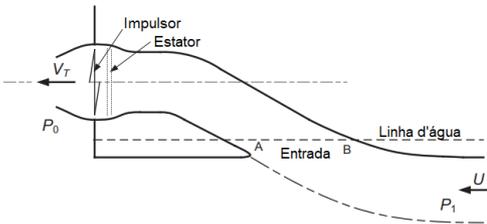
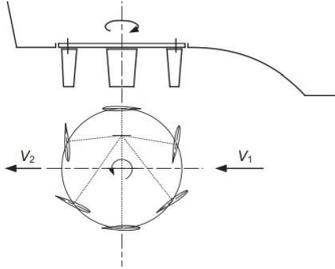
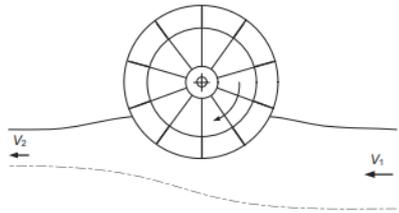
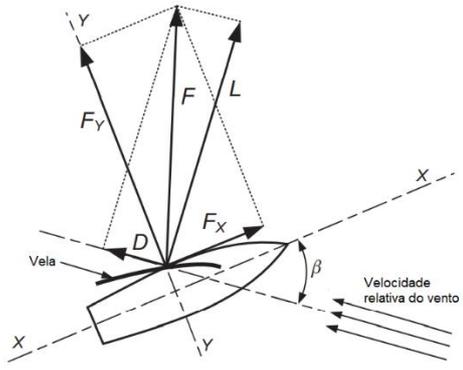
Quadro 1 – Tipos de instalação propulsora.

Tipos de instalações	Características principais
<p data-bbox="395 1648 671 1682"><b>Passo controlável</b></p>  <p data-bbox="395 2018 667 2051">Fonte: Wärtsilä (2022).</p>	<p data-bbox="850 1809 1417 1883">Permitem definir o passo do hélice para diferentes condições de uso.</p>

Continua (...)

Tipos de instalações	Características principais
<p data-bbox="360 230 703 304"><b>Propulsores em dutos</b></p>  <p data-bbox="395 636 663 667">Fonte: Oliveira (2013).</p>	<p data-bbox="847 360 1453 577">Podem ser separados em dutos aceleradores, desaceleradores, aro condutor e jato de bomba. O duto é utilizado para alterar o fluxo de água e suas características dependem do objetivo da embarcação.</p>
<p data-bbox="312 667 751 701"><b>Propulsores contra rotativos</b></p>  <p data-bbox="395 931 663 965">Fonte: Brunvoll (2022).</p>	<p data-bbox="847 741 1453 887">Possuem propulsores secundários menores que os propulsores principais com o objetivo de reduzir a resistência de esteira gerada.</p>
<p data-bbox="376 965 687 999"><b>Propulsores tandem</b></p>  <p data-bbox="336 1312 724 1346">Fonte: Koronowicz et al. (2010).</p>	<p data-bbox="847 1077 1453 1223">Mais de um propulsor é adicionado ao eixo principal. É utilizado quando um único propulsor não é capaz de fornecer o empuxo requerido pelo projeto.</p>
<p data-bbox="472 1346 592 1379"><b>Z-Drive</b></p>  <p data-bbox="376 1749 687 1783">Fonte: Ortlinghaus (2022).</p>	<p data-bbox="847 1514 1453 1615">A potência entre o motor e o propulsor é transmitida por engrenagens, sem a necessidade de eixos.</p>

Continua (...)

Tipos de instalações	Características principais
<p style="text-align: center;"><b>Azimutais</b></p>  <p style="text-align: center;">Fonte: Adaptado de Molland et al. (2011).</p>	<p>Alia propulsores de passo fixo e motores elétricos. O propulsor pode estar livre ou acoplado à um duto, sendo que o sistema pode rotacionar 360°.</p>
<p style="text-align: center;"><b>Jato d'água</b></p>  <p style="text-align: center;">Fonte: Adaptado de Molland et al. (2011).</p>	<p>Não possui hélice propulsor, pois a propulsão é dada por um jato d'água expelido à altas velocidades.</p>
<p style="text-align: center;"><b>Cicloidal</b></p>  <p style="text-align: center;">Fonte: Molland et al. (2011).</p>	<p>Propulsor vertical com pás que rotacionam em torno de um eixo vertical central. O empuxo pode ser direcionado em diferentes direções, não sendo necessária a adição de leme.</p>
<p style="text-align: center;"><b>Pedais</b></p>  <p style="text-align: center;">Fonte: Molland et al (2011).</p>	<p>Os pedais aceleram a superfície do fluido e podem ser alocados nas laterais ou na popa da embarcação, além de poderem conter pás fixas ou ajustáveis.</p>
<p style="text-align: center;"><b>Velas</b></p>  <p style="text-align: center;">Fonte: Adaptado de Molland (2011).</p>	<p>As velas atuam como aerofólios e utilizam o vento como principal fonte de movimento.</p>

Continua (...)

Tipos de instalações	Características principais
<p data-bbox="371 230 692 264"><b>Tipos de instalações</b></p> <p data-bbox="475 271 588 304"><b>Remos</b></p>  <p data-bbox="381 539 679 566">Fonte: Decathlon (2022).</p>	<p data-bbox="847 365 1449 472">Geralmente utilizados em pequenas embarcações e botes, resultam em baixos valores de potência.</p>
<p data-bbox="371 573 687 607"><b>Propulsores laterais</b></p>  <p data-bbox="384 891 676 918">Fonte: Kawasaki (2022).</p>	<p data-bbox="847 674 1449 815">Originalmente conhecidas como propulsores de proa, são utilizados em embarcações que requerem alto nível de manobrabilidade a baixas velocidades.</p>

Fonte: autoria própria (2022).

Embora o tipo de hélice varie a depender da embarcação e dos requisitos do projeto, Carlton (2007) afirma que, independentemente do modelo selecionado, as forças e os momentos gerados pelos propulsores podem ser expressos a partir dos coeficientes adimensionais mostrados nas Equações 1, 2 e 3, onde  $K_T$  representa o coeficiente de empuxo do hélice,  $K_Q$  o coeficiente de torque,  $J$  o coeficiente de avanço,  $T$  o empuxo gerado,  $Q$  o torque gerado,  $n$  a rotação do propulsor por segundo,  $D$  o diâmetro do hélice,  $V_A$  a velocidade de avanço e  $\rho$  a densidade do fluido (CARLTON, 2007).

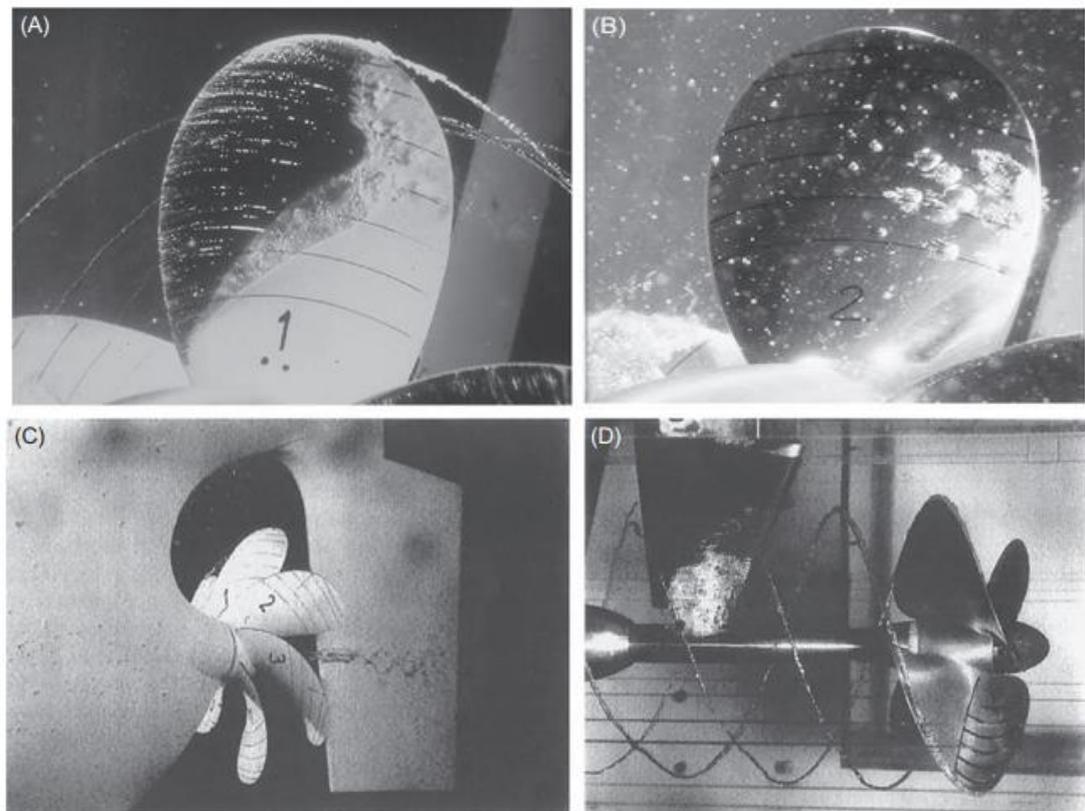
$$K_T = \frac{T}{\rho \times n^2 \times D^4} \quad (1)$$

$$K_Q = \frac{Q}{\rho \times n^2 \times D^5} \quad (2)$$

$$J = \frac{V_A}{n \times D} \quad (3)$$

Carlton (2007) ainda classifica o fenômeno da cavitação em quatro tipos principais: a cavitação do tipo folha, do tipo nuvem, do tipo bolha e do tipo vórtice (Figura 1).

Figura 1 - Tipos de cavitação



(A) Cavitação tipo folha e nuvem, com influência de cavitação tipo vórtice na ponta da pá. (B) Cavitação tipo bolha no meio do comprimento da pá. (C) Cavitação tipo vórtice no cubo do hélice. (D) Cavitação tipo vórtice junto da ponta das pás. Fonte: Carlton (2007, p. 225).

A cavitação do tipo folha ocorre devido ao ângulo de ataque presente no dorso das pás do propulsor. Esse tipo de cavitação depende diretamente da geometria das pás e é fortemente afetada pelas altas pressões de sucção existentes na região da meia-corda, além de crescer na direção do comprimento da pá.

Já a cavitação do tipo nuvem é caracterizada pela formação de pequenas bolhas, as quais provocam pequenos vórtices. As bolhas formadas durante esse processo podem apresentar dimensões micrométricas e, normalmente, são encontradas atrás de cavitações do tipo folha.

A cavitação do tipo bolha é resultado da queda de pressão decorrente da alta sucção existente na região central das pás (SGROTT, 2017). Esse tipo de cavitação é fortemente influenciado pela relação entre a espessura e a curvatura das pás.

Por fim, a cavitação do tipo vórtice ocorre devido a criação de regiões de baixa pressão geradas pela rotação do hélice. Normalmente, esse tipo de cavitação ocorre

na ponta das pás, porém, ao se estabilizar, o vértice pode se deslocar para o cubo do hélice.

A Conferência Internacional de Tanques de Reboque (ITTC, 2002) afirma que, independentemente do tipo de cavitação, fatores que alteram as características da água, como, por exemplo, a quantidade de gás presente no fluido, influenciam a ocorrência do fenômeno.

No entanto, segundo Carlton (2007), de maneira geral, o início da cavitação pode ser verificado usando a Equação 4, onde  $\sigma$  é o número de cavitação,  $P_\infty$  é a pressão do fluido no infinito (sem considerar a influência do sistema analisado) e  $P_V$  é uma constante que se refere a pressão de vapor do fluido.

$$\sigma = \frac{P_\infty - P_V}{\rho \times \frac{(n \times D)^2}{2}} \quad (4)$$

Sendo que o valor de  $P_\infty$  pode ser determinado usando a Equação 5. Onde  $P_{ATM}$  corresponde à pressão atmosférica,  $P_{VÁCUO}$  à pressão no vácuo e  $P_{HIDROSTÁTICA}$  à pressão decorrente da coluna de água acima do ponto analisado.

$$P_\infty = P_{ATM} - P_{VÁCUO} + P_{HIDROSTÁTICA} \quad (5)$$

Molland et al. (2011) também relacionam o início da cavitação com a igualdade  $\sigma = -C_{PNmin}$ , onde  $C_{PNmin}$  é o coeficiente de pressão mínimo atuante na superfície do hélice (Equação 6). Já na Equação 7 é descrito o coeficiente de pressão médio, o qual considera a pressão média atuante na superfície da pá do propulsor.

$$C_{PNmin} = \frac{P_{min} - P_\infty}{\rho \times \frac{(n \times D)^2}{2}} \quad (6)$$

$$C_{PNmédio} = \frac{P_{médio} - P_\infty}{0.5 \times \rho \times (n \times D)^2} \quad (7)$$

Onde o parâmetro  $P_{min}$  representa a pressão mínima atuante na superfície do hélice e  $P_{médio}$  a pressão média atuante na superfície do hélice.

Correlacionando  $\sigma$  e  $C_{PNmin}$  obtêm-se o índice de cavitação  $Cav_{INDEX}$  (Equação 8), o qual estabelece a relação entre a pressão mínima atuante na superfície do hélice ( $P_{MIN}$ ) e a pressão de vapor do fluido.

$$Cav_{INDEX} = \frac{P_{MIN} + P_V}{0,5 \times \rho \times (n \times D)^2} \quad (8)$$

Apesar dos trabalhos iniciais sobre o tema datarem do século XIX (REYNOLDS, 1894), ainda hoje existem incertezas sobre o assunto, as quais decorrem da complexidade física e computacional existente nos estudos com modelos em escala reduzida nas situações em que o escoamento apresenta transição laminar-turbulento. Menter et al. (2006) afirmam que a complexidade existente em simulações com baixos Re e a exigência de informações não locais são duas importantes limitações desses modelos.

Para mitigar tais dificuldades, em geral, os modelos de análise incluem fatores baseados em correlações experimentais. Por conta disso, autores como Walters e Cokljat (2008) e Menter et al. (2006) destacam a importância de se realizar a estimativa da influência que essa falta de informação (incerteza de entrada) exerce nos parâmetros simulados (incerteza na saída). Em outras palavras, dependendo do método de análise utilizado para avaliar a cavitação, incertezas de entrada podem distorcer os resultados numéricos.

## 2.2 ESTUDO DE INCERTEZAS E VARIÁVEIS DE ANÁLISE

Em virtude da complexidade do fenômeno, análises sobre o início da cavitação precisam considerar inúmeros fatores, que muitas vezes são difíceis de serem estimados. Sobre isso Kuiper (2012) escreve:

Mesmo a definição de início de cavitação sendo dada a partir da instabilidade de uma única bolha, a distinção entre cavitação gasosa e iniciação não pode ser realizada sem conhecer o conteúdo de núcleos do fluxo. Quando a distribuição de núcleo é considerada, a distribuição da densidade de núcleos é outro fator determinante para a frequência de início de cavitação. Além disso, a precisão de observação (o que é definido como uma bolha visível) é um fator que deve ser levado em consideração no início de cavitação. A determinação do início da cavitação é, portanto, difícil e a precisão da pressão de início observada de um propulsor não é alta. (KUIPER, 2012, p. 26, tradução nossa).

Birvalski et al. (2019) reforçam a ideia de que os princípios da cavitação são conhecidos há décadas, entretanto, certas características do fenômeno ainda hoje permanecem desconhecidas. Os autores apontam dificuldades como a obtenção de medições precisas do espectro de núcleos gasosos e líquidos e a dificuldade em se estimar sistematicamente a sensibilidade associada ao fenômeno.

Nesse caso, a análise de incertezas é um recurso importante para quantificar a sensibilidade da modelagem. Entretanto, enquanto a análise de incertezas foca na quantificação das incertezas nas variáveis de saída de um modelo, a análise de sensibilidade avalia como essas incertezas podem ser repartidas e relacionadas com as incertezas dos dados de entrada (SALTELLI et al., 2004).

As incertezas de entrada provêm de fatores físicos existentes e que não podem ser controlados, tais como a rugosidade da superfície das pás, os regimes turbulentos presentes no escoamento, as características do fluido, a rigidez do propulsor durante os experimentos, entre outros. Cada um desses fatores leva a respostas diferentes que podem ser previstas apenas com base em análises estatísticas.

Assim, uma análise de incertezas envolve dados de entrada e de saída de um modelo, os quais devem ser avaliados de acordo com a natureza do problema estudado. Nesse sentido, Saltelli et al. (2008, p. 5) descrevem duas abordagens para a análise de incertezas. Na primeira, chamada de modelo dedutivo ou prognóstico<sup>1</sup>, distinguem-se as equações utilizadas para entender um determinado sistema, daquelas usadas para prever o comportamento do sistema. Na segunda, chamada de modelo baseado em dados<sup>2</sup>, tenta-se ajustar as equações de modo a inferir o comportamento de um sistema utilizando apenas os dados observados.

Ainda segundo Saltelli et al. (2008), os dados de entrada referem-se a quaisquer informações que podem gerar uma variação nos dados de saída do modelo (Figura 2). Já Smith (2013) explica que, no contexto de quantificação de incertezas, o termo dados de entrada designa parâmetros, condições iniciais ou condições de contorno que podem apresentar variabilidade aleatória e/ou incerteza epistêmica.

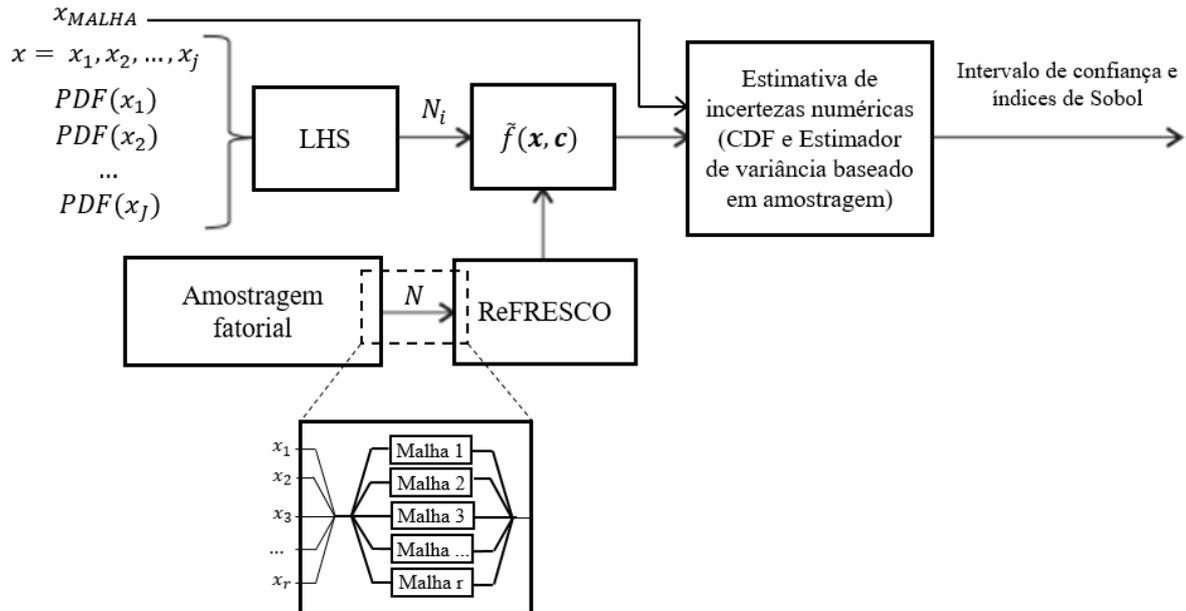
---

<sup>1</sup> Saltelli et al. (2008) utilizam como exemplo de modelos dedutivos tanto as leis físicas comprovadas quanto as hipóteses e especulações.

<sup>2</sup> Modelos baseados em dados, também conhecido como modelos empíricos, descrevem a realidade com o mínimo de parâmetros ajustáveis (YOUNG et al., 1996, tradução nossa).

Portanto, é nos casos em que se objetiva quantificar os efeitos das incertezas dos dados de entrada é que se aplicam a análise de incertezas, já que essa visa obter a incerteza na predição dos valores da saída propagando as incertezas dos parâmetros de entrada ao longo da modelagem.

Figura 2 - Representação das etapas para a obtenção das variáveis de saída.



Fonte: autoria própria (2022).

Na parte inferior da Figura 2 são mostradas as etapas seguidas para a obtenção dos valores de entrada das simulações numéricas. Como já mencionado, os valores iniciais são, em muitos casos, definidos com base em experimentos práticos, sendo assumido um intervalo de incerteza e aplicada a amostragem fatorial para obter os valores de entrada utilizados na análise CFD.

Uma vez finalizadas as simulações numéricas e obtidos os resultados, é possível avaliar a incerteza referente ao refino da malha ( $x_{malha}$ ), para em seguida serem avaliadas a influência das variáveis de entrada nos resultados das simulações.

No âmbito de uma simulação CFD, as incertezas nas variáveis de entrada são propagadas no modelo a fim de avaliar as incertezas das variáveis de saída estudadas. Por exemplo, a dificuldade em entender e modelar a queda de turbulência na região a montante do objeto simulado contribui para as incertezas existentes nos resultados referentes aos coeficientes de força e pressão atuantes no corpo.

### 2.3 ÍNDICES DE SOBOL

Os índices de Sobol, também conhecidos como índices de sensibilidade baseados em variância, são utilizados em análises de sensibilidade de resposta. Sobol (2001) afirma que tais índices são utilizados para estimar a influência e os efeitos que cada uma das variáveis de entrada (ou grupo de variáveis) exerce nas variáveis de saída.

Os índices de Sobol de uma determinada variável de saída, associada à uma variável de entrada, correspondem as razões entre as variâncias da variável de entrada ( $V_{ENTRADA}$ ) e as variâncias da variável de saída ( $V_{SAÍDA}$ ). Como não é possível obter o valor real da variável de entrada devido às incertezas existentes nas medições, os índices de Sobol são calculados utilizando o valor esperado (Equação 9). Maiores detalhes sobre o cálculo desses coeficientes podem ser vistos em Saltelli et al. (2010).

$$S_i = \frac{V_{ENTRADA}(E(SAÍDA|ENTRADA))}{V_{SAÍDA}} \quad (9)$$

A variância de entrada ( $V_{ENTRADA}$ ) pode ser calculada utilizando a Equação 10, onde A e B são matrizes contendo os valores de entrada e  $A_B^{(i)}$  é a matriz em que os valores da coluna i vêm da matriz B e todas as outras colunas vêm da matriz A. Para mais detalhes sobre a estimação numérica dos índices de Sobol recomenda-se a leitura de Sobol (2001).

$$V_{ENTRADA}(E(SAÍDA|ENTRADA)) = \frac{1}{N_i} \sum_{k=1}^{N_i} [f(B)_k (f(A_B^{(i)})_k - f(A)_k)] \quad (10)$$

Uma importante característica dos índices de Sobol é que a soma de todas as parcelas de incerteza, associadas à uma dada variável de saída, deve ser igual a 1. Considerando um exemplo com duas variáveis de entrada, a soma dos índices de Sobol, para cada variável de saída analisada, é descrita pela Equação 11, onde  $S_1$  corresponde ao efeito que a variável de entrada 1 tem na resposta estudada,  $S_2$  o efeito da variável de entrada 2 e  $S_{12}$  o efeito cruzado das variáveis 1 e 2.

$$1 = S_1 + S_2 + S_{12} \quad (11)$$

## 2.4 OS MODELOS DE TRANSIÇÃO LAMINAR-TURBULENTO

Diversos pesquisadores, tais como Menter et al (2006), Walters e Cokljat (2008) e Coder e Maughmer (2015), desenvolveram modelos numéricos para analisar escoamentos transientes, os quais permitem avaliar o efeito da transição para o regime turbulento em escoamentos com baixos  $Re$ .

O modelo de transição desenvolvido por Menter et al. (2006)<sup>3</sup> se baseia na equação de transporte para uma variável generalizada não previsível e, além de ser formulado sem o uso de parâmetros integrais, utiliza o modelo de turbulência de transporte de tensão cisalhante (SST).

Walters e Cokljat (2008) também desenvolveram um modelo de turbulência, baseado em viscosidade turbulenta, o qual contempla três equações de transporte. Esse modelo utiliza a energia cinética turbulenta ( $\kappa$ ) e busca refinar o comportamento do escoamento em regiões nas quais ocorre a mudança para o regime turbulento. Os autores avaliaram o modelo proposto em simulações com diferentes números de  $Re$ , diferentes geometrias, diferentes gradientes de pressão, diferentes ângulos de ataque e diferentes condições de turbulência na região que antecede o objeto de estudo. Em todos os casos estudados, o modelo apresentou boas previsões, sendo capaz de reproduzir escoamentos de transição com bons níveis de acurácia.

Eça et al. (2016) e Baltazar et al. (2018) demonstram que os resultados obtidos a partir de simulações realizadas com modelos de transição são afetados pela intensidade turbulenta definida na região da entrada do domínio computacional, além da redução que essa intensidade sofre a montante do propulsor. Entretanto, tais valores são definidos a partir de resultados experimentais e transportar esses resultados até o início do corpo analisado é uma tarefa complicada que pode levar à dúvida na previsão em modelos de transição do regime laminar para o turbulento.

Ainda segundo Eça et al. (2016), as incertezas capazes de gerar diferenças entre resultados numéricos e experimentais podem ser divididas em três grupos. O primeiro, contempla a qualidade da malha, o nível de discretização da geometria e os erros gerados a partir da convergência iterativa. Já o segundo, contempla as

---

<sup>3</sup> Florian Menter (1994) desenvolveu o modelo matemático SST com base nos modelos para análise de turbulência  $\kappa$ - $\omega$  de Wilcox e  $\kappa$ - $\epsilon$  com o objetivo de melhorar as previsões de fluxos com gradientes de pressão. Menter et al. (2006) aplicaram o modelo matemático SST no desenvolvimento de um modelo de transição baseado em variáveis locais.

propriedades dos materiais, as condições de contorno e as condições iniciais da simulação. Por fim, o terceiro grupo inclui os modelos físicos utilizados na simulação, que podem apresentar incertezas por conta das hipóteses adotadas na modelagem.

A incerteza associada à discretização resulta das diferenças existentes entre a solução discreta e a solução exata (contínua) do sistema. Entre os trabalhos que discutem essa incerteza, destaca-se a Sociedade Americana de Engenheiros Mecânicos (ASME, 2009), que busca quantificá-la afirmando que essa decorre de três parcelas: erros devido às soluções numéricas das equações utilizadas, erros devido aos dados de entrada e erros devido aos dados experimentais.

Além disso, é importante ressaltar que alguns autores, como Eça et al. (2016) e Rakhimov, Visser e Komen (2018), tratam as incertezas paramétricas e as incertezas de discretização como fatores não-relacionados. No entanto, deve-se ter cuidado ao utilizar tal abordagem, já que podem existir efeitos cruzados significativos, tal como discutido por Eça et al. (2019).

Já Sullivan (2015) explica que o procedimento de análise de quantificação de incertezas<sup>4</sup> pode ser dividido em duas abordagens: intrusiva e não-intrusiva. Ele descreve o método estocástico de Galerkin (aplicado em análises computacionais) como uma abordagem polinomial intrusiva que expande as variáveis de entrada em uma base polinomial caótica<sup>5</sup> a fim de determinar as equações governantes para a solução do sistema. No entanto, apesar dessa abordagem reduzir os resíduos estocásticos, ela exige o desenvolvimento e a resolução de modelos e equações diferentes para cada combinação de dados de entrada estudada, o que pode resultar em complicações adicionais para a análise.

Por outro lado, os métodos não-intrusivos baseados em amostragem consideram a simulação como uma caixa-preta que não pode ser modificada. Autores como Jun e Du (2020) apontam que a principal desvantagem dessa abordagem reside no alto custo computacional do processo. Ademais, na maioria dos casos práticos, análises CFD apresentam alto custo computacional de per si, e uma abordagem baseada em amostras pode ser impraticável para a maioria dos projetos.

---

<sup>4</sup> Análises de quantificação de incertezas estudam as possíveis fontes de erros e incertezas, incluindo erros de medição sistemáticos e estocásticos, limitações dos modelos, limitações numéricas, computacionais e de precisão (U.S. DEPARTMENT OF ENERGY, 2009, p. 135, tradução nossa).

<sup>5</sup> Como explica Saltelli (2015), um polinômio caótico contém componentes independentes e pode ser generalizado a partir de uma variável qualquer.

Entretanto, apesar de existirem métodos que visam reduzir a dificuldade da utilização das projeções de Galerkin, a facilidade em avaliar a função como uma caixa-preta motiva a escolha dessa abordagem, já que não é necessário ter acesso ou modificar o modelo de análise utilizado.

## 2.5 O PROPULSOR DUISBURG P1570

A favor da concisão textual, omite-se neste texto a clássica descrição sobre as características geométricas de um propulsor do tipo hélice. No entanto, o leitor interessado poderá encontrá-la facilmente em diversas referências, entre elas Carlton (2007), Leme (2021) e Molland et al. (2011).

Em 2012, o Instituto de Tecnologia Naval, Engenharia Oceânica e Sistemas de Transporte (ISMT) desenvolveu o Duisburg Test Case (DTC), o qual se baseia no projeto típico de um navio contêiner com capacidade para 14.000 TEUs (EL MOCTAR, 2012). O objetivo do DTC era fornecer dados confiáveis a serem utilizados em novos projetos similares. Entre os ensaios realizados, avaliou-se o desempenho de um propulsor do tipo hélice com cinco pás de passo fixo e rotação para a direita.

Os ensaios com o propulsor foram realizados utilizando o método britânico para dedução de atrito descrito por El Moctar (2012). Foram consideradas diferentes velocidades de avanço, cada qual submetida a diferentes rotações. Entre os resultados medidos, destaca-se a força residual atuante no modelo<sup>6</sup> para cada uma das condições de operação ensaiadas.

Katsuno (2019) buscou avaliar e quantificar as incertezas contidas em simulações computacionais de baixo  $Re$  utilizando como modelo o propulsor Duisburg P1570. Ele propôs um método para estimar incertezas paramétricas e de discretização e combiná-las em um único valor. Sua análise considerou três variáveis de entrada: intensidade turbulenta, viscosidade turbulenta e um parâmetro adimensional utilizado para definir uma zona sem perturbação a montante do propulsor.

Já Wielgosz et al. (2019) realizaram ensaios com o propulsor Duisburg P1570 operando em águas abertas visando analisar quantitativa e qualitativamente os fluxos cavitantes nas diferentes condições de operação. Os resultados obtidos destinavam-

---

<sup>6</sup> Nesse caso, o termo força residual refere-se a diferença entre o empuxo produzido pelo propulsor operando em águas abertas e a resistência ao avanço do casco nu.

se a validação das predições de cavitação e de desempenho do propulsor obtidas em simulações em CFD. Além de variar as condições de operação do propulsor, as análises de Wielgosz et al. (2019) variaram a pressão estática do fluido.

Com base nos resultados experimentais do propulsor e equações empíricas, Golf (2018) e Wielgosz et al. (2019) correlacionaram as diferenças observadas à possíveis erros e incertezas que poderiam afetar o experimento, tais como, as incertezas nas medições de empuxo, torque e velocidade do propulsor, as diferenças decorrentes da geometria do propulsor, as possíveis canalizações do escoamento devido às paredes do tanque de provas, além de outras características de menor importância. Para uma explicação mais detalhada pode-se consultar Golf (2018), Wielgosz (2019) e El Moctar et al. (2012).

Por fim, Kinaci e Gokce (2015) também estudaram o propulsor Duisburg P1570 visando melhorar sua eficiência hidrodinâmica. A análise computacional realizada pelos autores utilizou o modelo de turbulência Reynolds-Average Navier-Stokes (RANS)  $\kappa$ - $\omega$  e, para acompanhar a região de início da superfície livre, foi utilizado o método de volume de fluido<sup>7</sup> (VOF). Além disso, a análise foi dividida em duas seções. Na primeira, foram avaliados os efeitos que o hélice provoca no avanço da embarcação. Já na segunda, foram avaliados os efeitos de superfície livre gerados pelo propulsor.

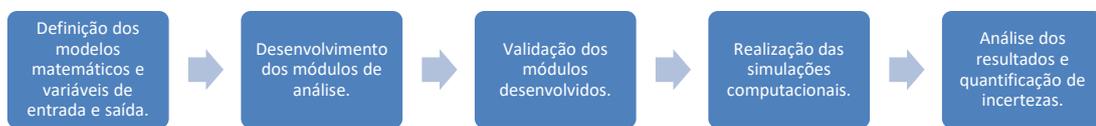
---

<sup>7</sup> Tradução nossa do termo “volume of fluid method”. VOF atua bem para inúmeros tipos de ondas de superfície livre as quais ocorrem devido à ação de forças gravitacionais e inerciais ao redor da interface ar-água (RHEE et al., 2005, tradução nossa).

### 3. METODOLOGIA

Este capítulo descreve as etapas, técnicas e ferramentas utilizadas no desenvolvimento deste trabalho. A metodologia proposta é fortemente baseada em Katsuno (2019), cujo procedimento de análise é esquematizado na Figura 3.

Figura 3 - Fluxograma referente ao desenvolvimento do trabalho.



Fonte: autoria própria (2022).

A partir da revisão bibliográfica realizada, foram determinadas as variáveis de entrada e de saída utilizadas nas simulações numéricas destinadas a avaliar o início da cavitação em propulsores do tipo hélice operando totalmente submersos.

Para a escolha das variáveis de saída foi realizado um estudo prévio que considerou os tipos de cavitação existentes, os fatores que afetam o início do fenômeno e os parâmetros da simulação que dependem da existência de resultados experimentais, tal como discutido em Golf (2018).

Dado o objetivo estabelecido para este trabalho, foram escolhidos os parâmetros  $K_T$ ,  $K_Q$ ,  $\eta$  e o índice de cavitação  $Ca_{INDEX}$  como variáveis de saída, as quais descrevem o desempenho do propulsor e a ocorrência de cavitação.

A seguir, foram identificadas as variáveis de entrada necessárias para realizar as simulações, ou seja, as variáveis que afetam diretamente os resultados obtidos. Com base no experimento realizado por Golf (2018), definiu-se os valores iniciais para a velocidade de rotação do propulsor (2.500 RPM) e para a pressão de vapor do fluido (2.300 Pa). As demais variáveis presentes nas Equações 1, 2 e 4 foram calculadas utilizando-se as respectivas equações.

### 3.1 MODELAGEM

Esta seção apresenta os dados de entrada e o modelo geométrico utilizado para as simulações computacionais. É importante observar que as imagens desta seção foram geradas utilizando os códigos em Python fornecidos por Golf (2018) e Wielgosz et al. (2019), sendo que os dados exibidos nas figuras se referem às condições avaliadas neste trabalho, as quais se baseiam nos resultados experimentais descritos em El Moctar (2012).

#### 3.1.1 Modelo geométrico do propulsor Duisburg P1570

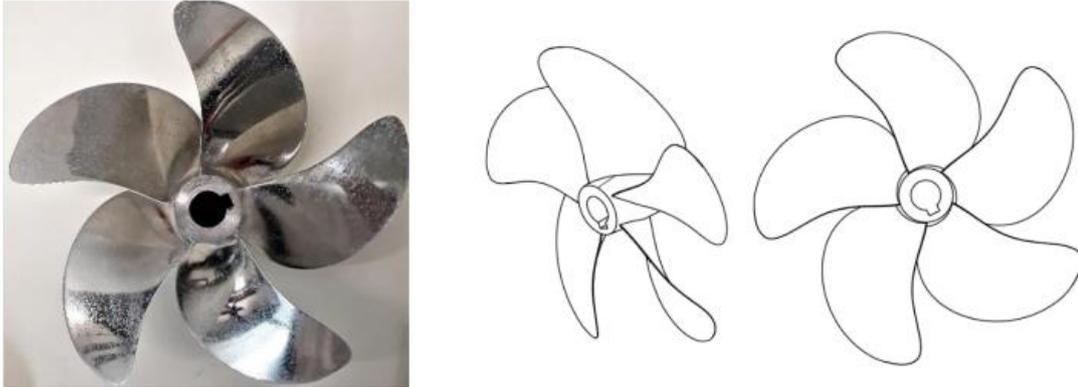
O modelo utilizado nas simulações foi o propulsor Duisburg P1570 desenvolvido pela Ship Model Basin Potsdam (SVA) e descrito por El Moctar e Shigunov (2014) e Golf (2018). Na Tabela 3 são apresentadas as características principais do propulsor, tanto para o modelo em escala reduzida quanto para o propulsor em escala real. Por fim, na Figura 4 é mostrada uma foto do propulsor e o modelo CAD desenvolvido por Golf (2018).

Tabela 1 - Parâmetros do propulsor Duisburg P1570.

Parâmetros	Modelo	Escala Real
D [m]	0,150	8,911
$P_{0.7}/D$ [-]	0,959	0,959
$A_E/A_0$ [-]	0,800	0,800
$C_{0.7}$ [m]	0,054	3,208
$\theta_{eff}$ [°]	31,970	31,970

Fonte: El Moctar et al. (2012, p. 51).

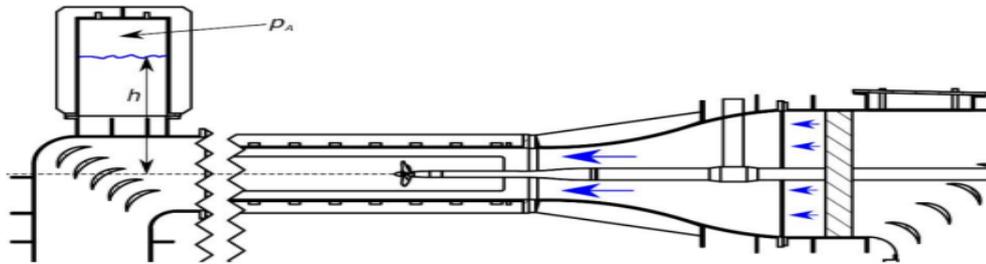
Figura 4 - Propulsor Duisburg P1570 e seu modelo CAD.



Fonte: Wielgosz (2019).

Os dados experimentais utilizados para comparação foram retirados de ensaios realizados no túnel de cavitação K23 do Instituto de Tecnologia Marítima, Engenharia Marítima e Sistemas de Transporte (ISTM) (Figura 5). Como descrito por Wielgosz (2019) e mostrado na Figura 6, o túnel possui seção transversal de 0,300 m por 0,300 m e o propulsor é conectado à um dinamômetro J19 Kempf & Remmers montado ao final do eixo rotativo<sup>8</sup>.

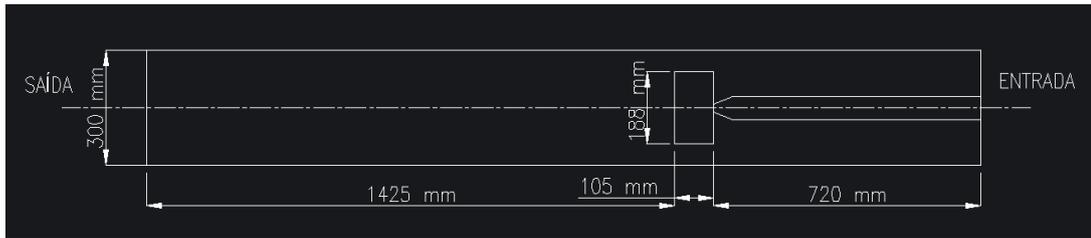
Figura 5 - Representação do experimento do propulsor Duisburg.



Fonte: Golf (2018).

<sup>8</sup> Embora durante um ensaio de águas abertas, para permitir que o escoamento atuante nas pás esteja livre de obstáculos, o propulsor seja fixado por um eixo localizado a jusante do hélice, nesse caso os autores optaram por fixar o eixo propulsor a montante do hélice, de forma a reproduzir mais fielmente a condição de operação de um propulsor naval.

Figura 6 - Modelo do túnel utilizado na simulação numérica em milímetros.



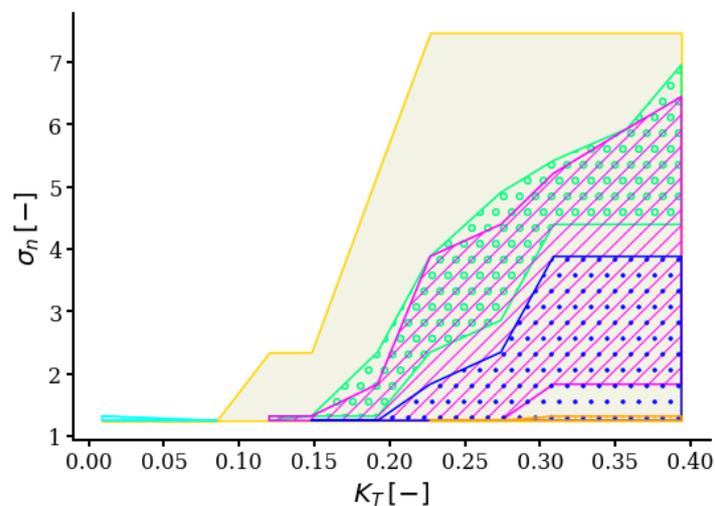
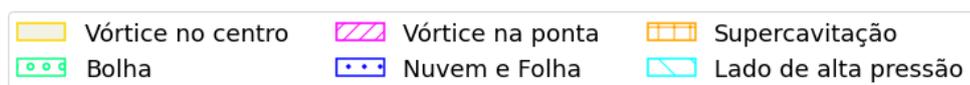
Fonte: autoria própria (2022).

Por fim é importante destacar que o arquivo IGES contendo a geometria simulada foi fornecido por Golf (2018), enquanto a malha desenvolvida seguiu a modelagem proposta por Katsuno (2019), o que permitiu a comparação assertiva dos resultados.

### 3.2 VARIÁVEIS E CONDIÇÕES DE CONTORNO

Para a seleção dos valores a serem utilizados no estudo das incertezas paramétricas, foram considerados os trabalhos de Golf (2018) e Wielgosz et al. (2019). Seguindo esses trabalhos, foi possível definir a região em que ocorre o início de cada um dos tipos de cavitação (Figura 7).

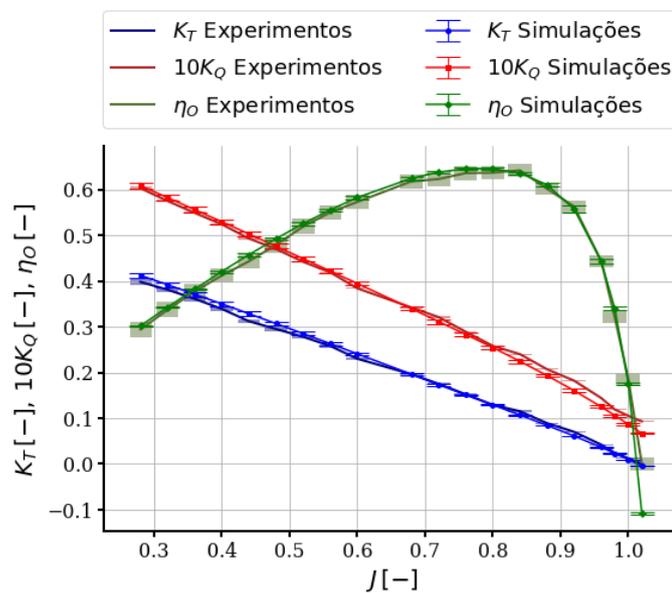
Figura 7 - Diagrama de cavitação para 2500 RPM.



Fonte: adaptado de Golf (2018) e Wielgosz et al. (2019).

Observando a Figura 7, é possível estimar a relação  $\sigma$  vs.  $K_T$  na qual ocorre o início da cavitação dos tipos folha e nuvem. Mapeada a região de interesse, foi arbitrado um valor de referência para  $\sigma$ , a partir do qual foi definido o valor de  $K_T$ . Utilizando o diagrama de águas abertas apresentado na Figura 8, os valores de  $K_Q$ ,  $\eta$  e  $J$  também foram definidos. Por fim, foram calculados os valores do empuxo ( $T$ ), da pressão de referência ( $P_{REF}$ ) e da velocidade de avanço ( $V_A$ ), os quais são sintetizados na Tabela 2.

Figura 8 - Diagrama de águas abertas para o propulsor Duisburg a 2500 RPM.



Fonte: adaptado de Golf (2018) e Wielgosz et al. (2019).

Tabela 2 – Variáveis definidas com base no experimento de Golf (2018).

Variável	Valor
RPM [RPM]	2.500,000
$P_V$ [Pa]	2.300,000
$T$ [N]	219,463
$P_{REF}$ [Pa]	43.266,406
$V_A$ [m/s]	3,625

Fonte: autoria própria (2022).

Com as variáveis de saída da simulação numérica definidas, cujos valores são sintetizados na Tabela 3, foram definidas as regiões de análise relacionadas a cada parâmetro. Inicialmente, foi utilizado o trabalho de Wielgosz et al. (2019) para a

definição do intervalo de avaliação, porém, a fim de ampliar o escopo do trabalho original, optou-se por aumentar o intervalo originalmente definido pelos autores (1,3%) para um intervalo de variação de 5%.

Tabela 3 – Variáveis de saída.

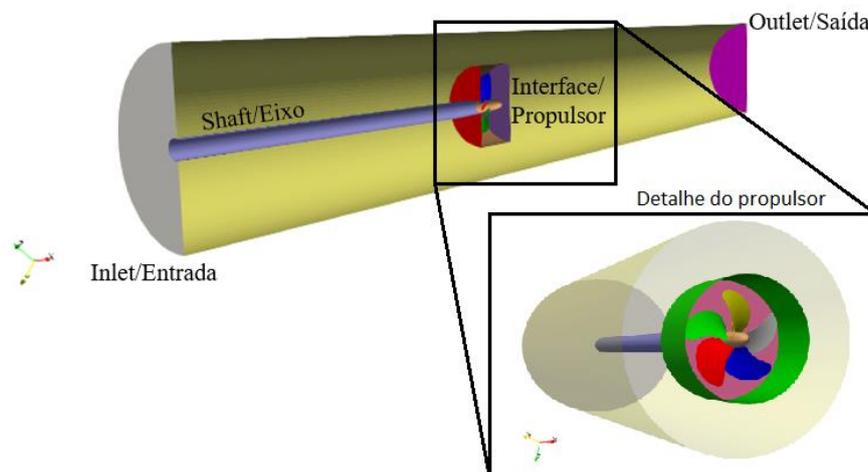
Variável	Valor
$K_T$ [-]	0,250
$K_Q$ [-]	0,040
$\eta$ [-]	0,569
$\sigma$ [-]	2,100

Fonte: autoria própria (2022).

Dado que os resultados da simulação devem ser comparados àqueles obtidos em ensaios reais, as condições de contorno aplicadas as simulações devem se assemelhar ao máximo àquelas observadas durante o experimento. Por conta disso, a realização das simulações desconsiderou a transferência de calor e a rugosidade das paredes do tanque. Já a velocidade do fluido no início do volume de controle foi definida como 3,625 m/s na direção paralela ao eixo do propulsor e 0,000 m/s nas demais direções. A intensidade turbulenta foi considerada igual a  $2,000 \times 10^{-2}$  (2%) e a pressão ao final do túnel foi considerada 0 Pa.

Na Figura 9 é mostrada a modelagem utilizada para a simulação numérica, onde é mostrado um detalhamento da região no entorno do propulsor.

Figura 9 - Visualização 3D do ambiente numérico de simulação.



Fonte: adaptado de Katsuno (2019).

As paredes do túnel foram consideradas fixas e, conforme já mencionado, não foram consideradas trocas de calor durante o escoamento e nem perdas por conta da rugosidade das paredes. Para a viscosidade do fluido foi considerado um valor de  $1,111 \times 10^{-3} \text{ N s/m}^2$ , enquanto para a densidade da água foi utilizado o valor de  $998,100 \text{ kg/m}^3$ . Já em relação as condições de contorno, para o eixo que antecede o propulsor considerou-se a condição do tipo escorregamento, enquanto para o propulsor foi utilizada a condição de contorno do tipo parede. É importante observar que o cilindro ao redor do hélice propulsor, mostrado na Figura 9, serve apenas para representar a região do rotor na qual foi aplicado o refinamento da malha.

Conforme já mencionado, seguindo o trabalho de Katsuno (2019), optou-se por utilizar o programa ReFRESKO para as simulações CFD e desenvolver um procedimento em Python para calcular o índice de cavitação para cada uma das condições de operação analisada.

### 3.3 TOPOLOGIA DA MALHA E CONFIGURAÇÃO NUMÉRICA

Para avaliar o efeito que a discretização exerce nas incertezas dos resultados numéricos, foram estudados diferentes graus de refino de malha (Tabela 4).

Tabela 4 – Número de elementos de cada malha estudada.

Refino da malha	Número de elementos	Razão de malha ( $\frac{h}{h(j700)}$ )
<i>f</i> 0,325	2.919.825	2,202
<i>f</i> 0,400	5.164.000	1,821
<i>f</i> 0,475	8.796.570	1,524
<i>f</i> 0,500	14.250.060	1,298
<i>f</i> 0,625	21.620.280	1,129
<i>f</i> 0,700	31.164.240	1,000

Fonte: autoria própria (2022).

Ainda seguindo a abordagem de Katsuno (2019), apenas a região no entorno do propulsor foi refinada, sendo que o comprimento característico médio dos elementos ( $h$ ) foi definido seguindo a Equação 12, onde  $h(j700)$  corresponde ao comprimento característico médio da malha *f*0,700.

$$\text{razao de malhas} = \sqrt[3]{\frac{\text{Número de elementos [f0,700]}}{\text{Número de elementos [malha]}}} = \frac{h(j700)}{h_{malha}} \quad (12)$$

Para a análise numérica, as equações governantes foram discretizadas utilizando a abordagem baseada em volumes finitos e foi considerado um regime permanente.

O acoplamento pressão-velocidade foi solucionado utilizando o algoritmo SIMPLE sendo calculado no centro do elemento. E, a fim de garantir a conservação da massa, uma equação de correção de pressão baseada no algoritmo SIMPLE foi utilizada.

Para as equações de transporte de momento, o solucionador PETSc-GMRES juntamente com o pré-condicionador JACOBI foi utilizado. O esquema de discretização escolhido foi o LIMITED QUICK com a correção de excentricidade ativada. Por fim, a turbulência foi modelada empregando-se o algoritmo  $\kappa$ -SKL e, para a resolução do modelo, foi utilizado o solucionador PETSc-GMRES junto com o Bloco de Jacobi.

Para simular a rotação, foi utilizado o modelo de referência absoluta (AFM)<sup>9</sup>, o qual mantém o propulsor fixo e rotaciona o fluido ao redor dele. Segundo Marin (2021), embora simplifique o procedimento, tal abordagem apresenta limitações quando se busca avaliar mais detalhadamente o fluxo em regiões distante das pás.

O número máximo de iterações utilizado foi de 20.000 e, para as equações de momento e pressão, foi considerada uma tolerância de convergência igual à  $10^{-2}$ .

### 3.4 ANÁLISE DOS RESULTADOS

Para a análise de resultados, primeiramente foram gerados seis arquivos em formato CSV contendo os resultados obtidos com as simulações executadas (Tabela 5), sendo um arquivo para cada refino de malha estudado.

---

<sup>9</sup> Método que resolve as equações RANS no referencial móvel, mas que as escrevem em termos de quantidades absolutas ou inerciais do referencial (MARIN, 2021).

### 3.4.1 Elaboração dos arquivos CSV para a análise de incertezas

Inicialmente, foi definido o valor de rotação do hélice igual a 2.500 RPM utilizando como base as condições estudadas por Golf (2018) durante os experimentos realizados. Conforme já mencionado, para definir a região de análise foi considerada uma incerteza de 5% nos dados experimentais e, aplicando-se a amostragem fatorial<sup>10</sup> para a seleção dos valores a serem analisados, obteve-se: 2474,062 RPM, 2505,781 RPM, 2537,500 RPM, 2569,219 RPM e 2600,937 RPM.

De maneira análoga, a partir do valor de pressão de vapor inicial de 2.300 Pa, observado por Golf (2018), foram definidos cinco valores dentro do intervalo de 5%, sendo eles: 2242,500 Pa, 2271,250 Pa, 2300,000 Pa, 2328,750 Pa e 2357,500 Pa.

Uma vez definidas as variáveis de entrada e as malhas a serem estudadas, foram executadas as simulações. Finalizadas as simulações, foi possível realizar o processamento de dados e calcular os valores das variáveis de saída  $K_T$ ,  $K_Q$ ,  $\eta$  (Anexo B) e do índice de cavitação ( $C_{aVINDEX}$ ), o qual foi calculado utilizando o código mostrado no Apêndice B. Assim, cada um dos seis arquivos CSV é constituído de sete colunas (RPM,  $P_V$ ,  $K_T$ ,  $K_Q$ ,  $\eta$  e  $C_{aVINDEX}$ ) e 25 linhas.

### 3.4.2 Realização de simulações computacionais

Uma vez que os valores de pressão de vapor foram corrigidos na fase de análise dos resultados, a rigor, foram realizadas 30 simulações, correspondendo a cinco valores de rotação e seis graus de refino de malha (Tabela 5).

Tabela 5 – Simulações computacionais e seus respectivos casos.

Refino da malha	RPM	Caso
$f_{0,325}$	2.474,063	0
	2.505,781	1
	2.537,500	2
	2.569,219	3
	2.600,937	4

Continua (...)

<sup>10</sup> A análise fatorial é um método estatístico usado para descrever a variabilidade entre variáveis correlacionadas (PEREIRA et al., 2019).

Refino da malha	RPM	Caso
<i>f</i> 0,400	2.474,063	0
	2.505,781	1
	2.537,500	2
	2.569,219	3
	2.600,937	4
<i>f</i> 0,475	2.474,063	0
	2.505,781	1
	2.537,500	2
	2.569,219	3
	2.600,937	4
<i>f</i> 0,550	2.474,063	0
	2.505,781	1
	2.537,500	2
	2.569,219	3
	2.600,937	4
<i>f</i> 0,625	2.474,063	0
	2.505,781	1
	2.537,500	2
	2.569,219	3
	2.600,937	4
<i>f</i> 0,700	2.474,063	0
	2.505,781	1
	2.537,500	2
	2.569,219	3
	2.600,937	4

Fonte: autoria própria (2022).

Para a simulação numérica foi utilizado o programa ReFRESKO<sup>11</sup> versão 2.7.0 trunk-rc1<sup>12</sup>, que, como explicado por Vaz, Jaouen e Hoekstra (2009), “é um

<sup>11</sup> [www.refresco.org](http://www.refresco.org).

<sup>12</sup> Pode também ser realizada utilizando a versão 2.8.0-t4 desenvolvida pelo MARIN.

código CFD cujos modelos de turbulência e cavitação permitem resolver escoamentos instáveis utilizando as equações de Navier-Stokes” (apud MARIN, 2021).

As simulações computacionais foram executadas utilizando os arquivos controls.xml contidos no Apêndice A, sendo que foi utilizado um arquivo diferente para cada uma das malhas estudadas (Tabela 5). Os arquivos xml foram desenvolvidos em 2021 e foram executados no servidor Linux do Instituto MARIN por meio de uma conexão Secure Shell (SSH).

No início do arquivo controls.xml estão descritos os dados gerais, especificados na seção <general>. Sendo que nessa seção estão localizados o nome e a descrição da simulação, juntamente com valores de referência utilizados para o cálculo dos parâmetros adimensionais, como Re, por exemplo.

O número de elementos utilizados na malha, o nome do arquivo que contém a malha (gridFileName) e o diretório em que ele se encontra (gridFilePath) são especificados na seção <grids>.

Já na seção <massMomentumSolver> é definido o método utilizado para resolver o sistema de equações lineares para os campos de velocidade e pressão. Vale ressaltar que as equações de transporte para as variáveis podem ser resolvidas tanto separadamente quanto de maneira acoplada, a depender do método utilizado.

Na seção <maxIteration> é definido o número máximo de iterações a serem realizadas, enquanto as condições de contorno são descritas em <boundaries>. Cada condição de contorno utilizada deve ser nomeada e definida de acordo com o arquivo CGNS (CFD General Notation System), o qual é usado para a manipulação dos dados de uma análise CFD. Mais detalhes sobre a formatação do arquivo CGNS podem ser vistos em NASA (2001).

Já na seção <equations> são enumeradas as equações de transporte resolvidas durante a simulação. No presente estudo, uma equação de momento, uma equação de pressão e uma equação de turbulência foram utilizadas. Para a equação de turbulência foi utilizado o modelo  $\kappa$ -SKL ( $\kappa - \sqrt{\kappa L}$ ) que apresenta melhor custo-benefício em relação a acurácia de resposta e o tempo de processamento (PINHO, 2020).

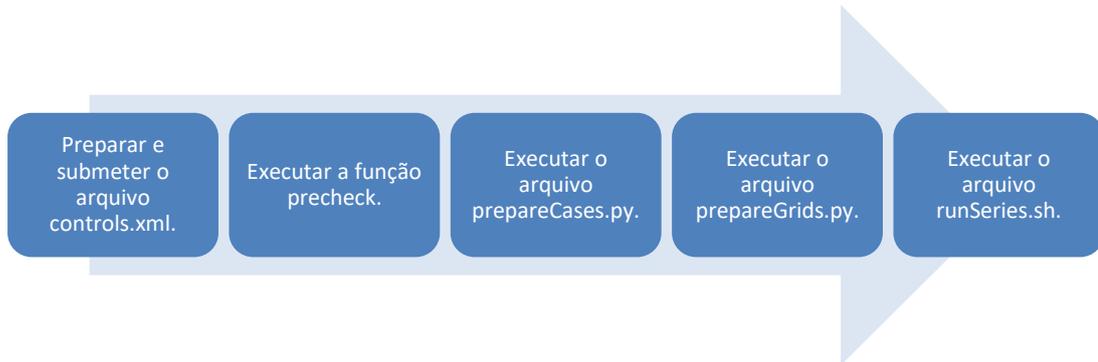
Por fim, as seções seguintes do arquivo (<materials> e <monitors>) são utilizadas para definir outros parâmetros de menor relevância neste trabalho.

Após o arquivo controls.xml ter sido preparado, ele é enviado ao servidor do MARIN, onde uma pré-verificação (função precheck) é realizada. A seguir a simulação

é iniciada por meio da rotina `prepareCases.py` (Anexo A), a qual servirá para criar os 30 casos simulados e suas respectivas pastas. Por fim, as rotinas `prepareGrids.sh` e `runSeries.sh` executam as simulações CFD contidas no arquivo `controls.xml`.

Na Figura 10 são mostrados os passos descritos nesta seção e que foram utilizados para a realização das simulações numéricas.

Figura 10 - Etapas para a realização das simulações numéricas.



Fonte: autoria própria (2022).

### 3.4.3 Validação dos resultados

Inicialmente foi realizada a validação dos resultados obtidos nas simulações CFD, os quais foram comparados com os resultados experimentais disponíveis em Golf (2018).

No entanto, visto que ambas as análises (numérica e experimental) possuem incertezas, a comparação dos resultados deve considerar ambas as faixas de incertezas. Assim, as Equações 13, 14 e 15 vistas em Katsuno (2019), foram utilizadas para calcular a incerteza de validação  $U_{VAL}$  e o erro de comparação de validação  $E$ .

$$U_{VAL}^2 = U_{NUM}^2 + U_{EXP}^2 + U_{PAR}^2 \quad (13)$$

$$U_{NUM}^2 = U_{ITER}^2 + U_{DIS}^2 \quad (14)$$

$$E = S - D_E \quad (15)$$

Onde  $U_{NUM}$  corresponde à incerteza numérica;  $U_{PAR}$  corresponde à incerteza paramétrica;  $U_{EXP}$  corresponde à incerteza experimental;  $S$  corresponde à predição numérica e  $D_E$  corresponde ao valor experimental. Sendo que, como visto na Equação

14, a incerteza numérica pode ser separada em dois termos: a incerteza de iteração ( $U_{ITER}$ ) e a incerteza de discretização ( $U_{DIS}$ ).

Neste trabalho, a incerteza paramétrica  $U_{PAR}$  representa a incerteza nos resultados (variáveis de saída), enquanto o termo  $U_{ENTRADA}$  se refere as incertezas das variáveis de entrada. Destaca-se que, a exemplo de Katsuno (2019), a incerteza de discretização ( $U_{DIS}$ ) também será considerada como uma incerteza de entrada.

A análise dos resultados visa avaliar as propriedades de uma análise CFD que causam incertezas na análise de cavitação de um propulsor. Tais incertezas foram definidas a partir das oscilações observadas nas variáveis de saída da simulação, expressas por meio da função de densidade acumulada (CDF) e do intervalo de confiança (IC) dos resultados.

Para cada malha estudada, os cinco valores de rotação foram combinados com os cinco valores de pressão de vapor, gerando 25 casos, a partir dos quais foi montada a CDF.

Já o IC entre dois níveis de probabilidade, em se tratando de uma análise de quantificação de incertezas, é considerado independente dos índices de Sobol e corresponde ao inverso da CDF.

As barras de erro mostradas neste trabalho consideram sempre um intervalo de confiança de 95%<sup>13</sup> e são calculadas utilizando a Equação 16.

$$IC_{TAMANHO} = CDF^{-1}(0,975) - CDF^{-1}(0,025). \quad (16)$$

### 3.5 INCERTEZAS DE ENTRADA

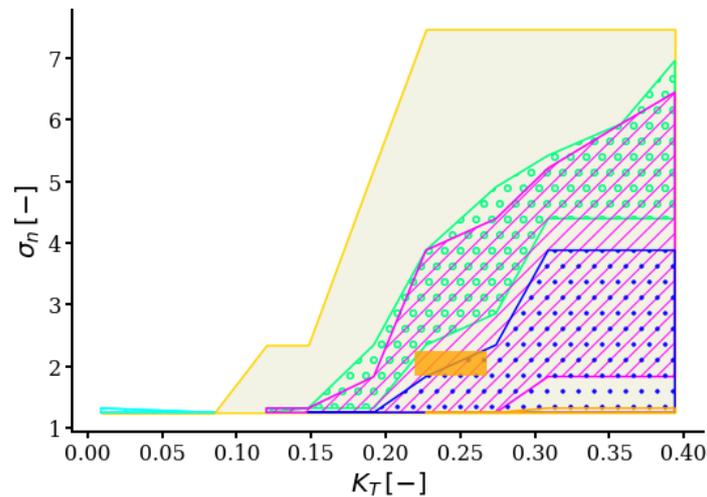
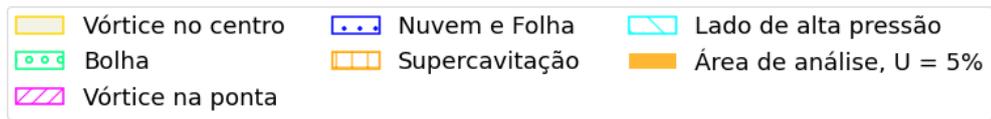
Uma vez que o objetivo do trabalho é avaliar o início da cavitação dos tipos folha e nuvem, a região de análise deve estar localizada na área onde predominam tais tipos de cavitação.

Conforme já mencionado, os cálculos do intervalo das variáveis de saída utilizaram como base os valores encontrados por Wielgosz et al. (2019), que obtiveram uma incerteza de 1,3% para o  $K_T$  operando no  $J$  igual a 0,8. Entretanto, para ampliar o escopo da análise, optou-se por considerar uma incerteza de 5%, o que define a região de análise destacada em amarelo nas Figuras 11 e 12.

---

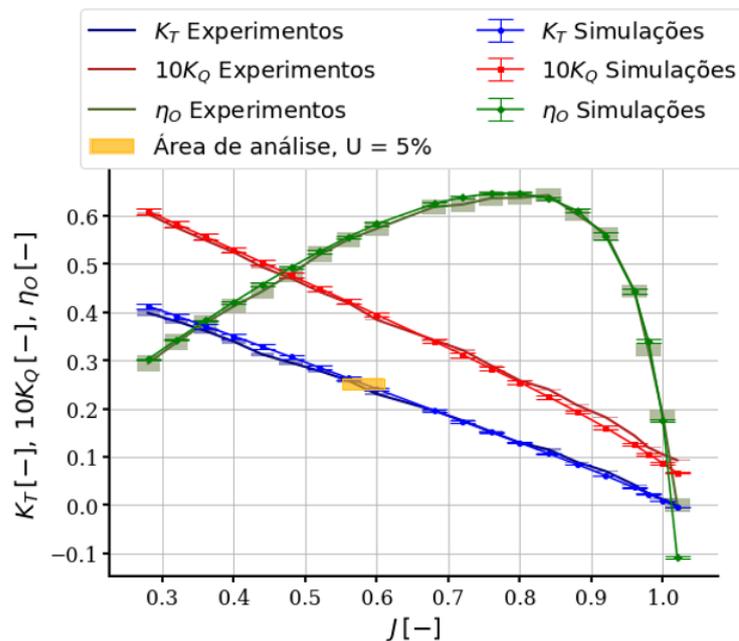
<sup>13</sup> Valor normalmente utilizado quando se trata de plotagens de erros (KATSUNO, 2019).

Figura 11 - Região de análise no diagrama de início de cavitação.



Fonte: adaptado de Golf (2018) e Wielgosz et al. (2019).

Figura 12 - Região de análise no diagrama de águas abertas.



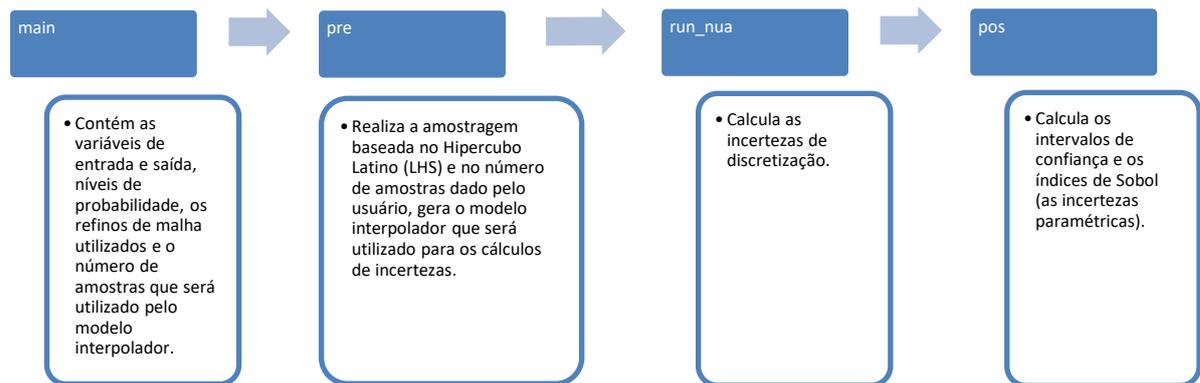
Fonte: adaptado de Golf (2018) e Wielgosz et al. (2019).

### 3.6 DESENVOLVIMENTO E VALIDAÇÃO DA ESTRUTURA NUMÉRICA

Após a definição dos casos a serem analisados e da modelagem CFD adotada, foi desenvolvido um procedimento numérico para a quantificação de

incertezas, o qual foi programado usando a linguagem Python e é composto pelas etapas mostradas na Figura 13. Essa separação visa generalizar o programa, permitindo que esse seja utilizado em diferentes tipos de análises de incertezas, não se limitando às análises de início de cavitação.

Figura 13 – Módulos desenvolvidos e suas funções principais.



Fonte: autoria própria (2022).

O programa desenvolvido em Python é capaz de realizar o pré-processamento dos dados de entrada e os cálculos dos índices de Sobol, além do cálculo das incertezas devido ao refino de malha.

O arquivo principal é denominado main.py e nele são definidas as entradas do programa, sendo elas: as variáveis de saída desejadas, os níveis de refino de malha estudados, a quantidade de elementos em cada refino de malha, os limites inferiores e superiores de cada uma das variáveis de entrada, os níveis de probabilidade avaliados e a quantidade de amostras avaliadas pelo modelo interpolador utilizado para o cálculo das incertezas.

Como o presente trabalho seguiu o modelo de análise de incertezas baseado em dados e utilizou a abordagem não-intrusiva baseada em amostragem de Galerkin, foi necessário desenvolver um modelo interpolador com base nos 25 valores (combinações entre RPM e  $P_v$ ) obtidos para cada um dos refinamentos de malha estudados.

O uso de um modelo interpolador permite aumentar o espaço amostral explorado sem a execução de novas simulações CFD, o que permite ampliar a análise de incertezas realizada.

Nessa abordagem, a função  $f$  (resultado da simulação CFD) é assumida como uma caixa-preta e é examinada diversas vezes buscando estabelecer a correlação entre as variáveis de entrada utilizadas e as variáveis de saída obtidas. O modelo interpolador foi criado utilizando a Amostragem por Hipercubo Latino (LHS) (SHIELDS e ZHANG, 2016), a qual se baseia na probabilidade de ocorrência da variável de entrada e correlaciona os dados de entrada selecionados e os dados de saída obtidos com as simulações.

Essa abordagem permite que regiões de alta probabilidade sejam melhor exploradas em detrimento de regiões com menores probabilidades de ocorrência. A implementação do modelo interpolador, bem como as rotinas de uso, foi programada no módulo `pre.py`.

Já o módulo `run_nua.py` realiza o cálculo da incerteza de discretização, utilizando a biblioteca denominada `numerical_uncertainty.exe`<sup>14</sup>, a qual calcula as incertezas contidas nas variáveis de saída para cada refino de malha.

Por fim, o módulo `pos.py` realiza o processamento dos resultados obtidos na simulação numérica e, utilizando a biblioteca de código aberto SALib (HERMAN, 2021), calcula os índices de Sobol e a função de distribuição acumulada (CDF), permitindo avaliar o efeito de cada uma das variáveis de entrada nos resultados obtidos.

É interessante observar que, uma vez que o programa desenvolvido permite avaliar os resultados da simulação para diferentes graus de refino da malha, pode-se também utilizá-lo para estudos de convergência e qualidade de malha.

Para a validação dos módulos desenvolvidos foram realizados dois procedimentos. Inicialmente, cada um dos módulos foi executado individualmente e os resultados foram comparados com aqueles obtidos por Katsuno (2019). Por fim, o programa completo foi executado e os resultados também foram comparados com aqueles obtidos no trabalho de referência.

---

<sup>14</sup> Esse programa foi desenvolvido por Eça e Hoekstra (2014) e objetiva estimar as incertezas numéricas a partir da resolução de equações que têm como base os níveis de refino das malhas utilizadas. Para mais detalhes recomenda-se a leitura de Eça e Hoekstra (2014).

## 4. RESULTADOS

Neste capítulo são apresentados e discutidos os resultados obtidos nas simulações numéricas, além das incertezas calculadas para o estudo de caso realizado.

### 4.1 VERIFICAÇÃO E VALIDAÇÃO DOS MÓDULOS EM PYTHON

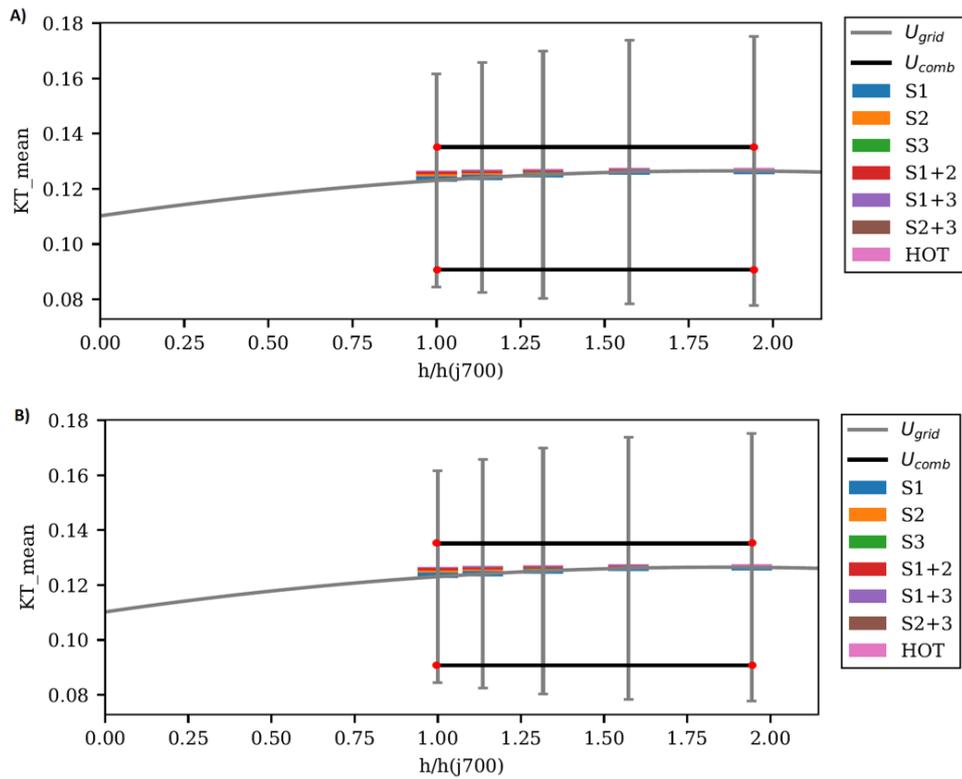
Antes de realizar as simulações referentes ao estudo de caso proposto, optou-se por realizar a verificação dos módulos Python desenvolvidos. Essa avaliação visa verificar o correto funcionamento dos módulos elaborados, aumentando assim a confiabilidade do trabalho realizado.

Esta seção descreve o processo de verificação, o qual se baseou nos resultados das simulações realizadas por Katsuno (2019). Nas Figuras 14, 15, 16 e 17 são apresentados os gráficos de incertezas obtidos com o código desenvolvido neste trabalho (Imagens A) e aqueles obtidos por Katsuno (2019) (Imagens B). O eixo x representa as razões de malha,  $S_1$ ,  $S_2$ ,  $S_3$ ,  $S_{1+2}$ ,  $S_{1+3}$  e  $S_{2+3}$  representam as incertezas de entrada, HOT representa os termos de alta ordem,  $U_{grid}$  representa as incertezas de discretização e  $U_{comb}$  representa as incertezas combinadas.

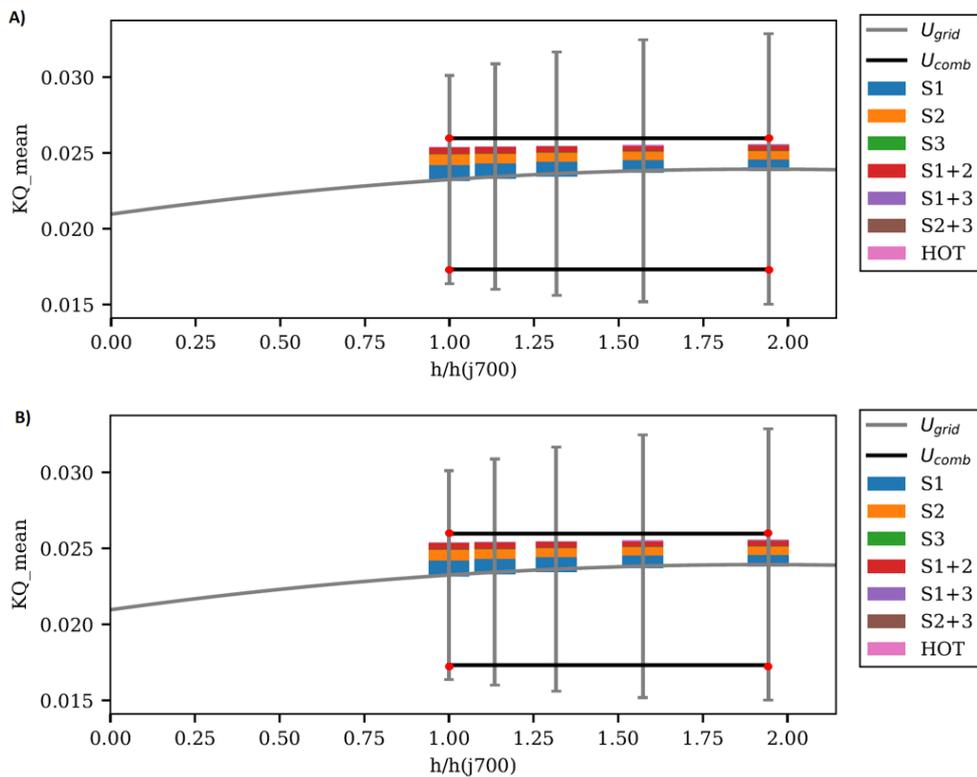
Além da análise visual, foi utilizado o aplicativo WebSiteDigitizer<sup>15</sup> para extrair pontos dos gráficos de Katsuno (2019) e realizar a comparação numérica dos valores. Assim, foram extraídas as coordenadas x e y dos pontos vermelhos (Figuras 14, 15, 16 e 17) posicionados sobre a linha de incerteza combinada ( $U_{COMB}$ ). Em todos os pontos analisados as diferenças observadas foram inferiores a 0,01%, confirmando que os resultados obtidos com a estrutura de códigos desenvolvida são significativamente próximos àqueles obtidos por Katsuno (2019).

---

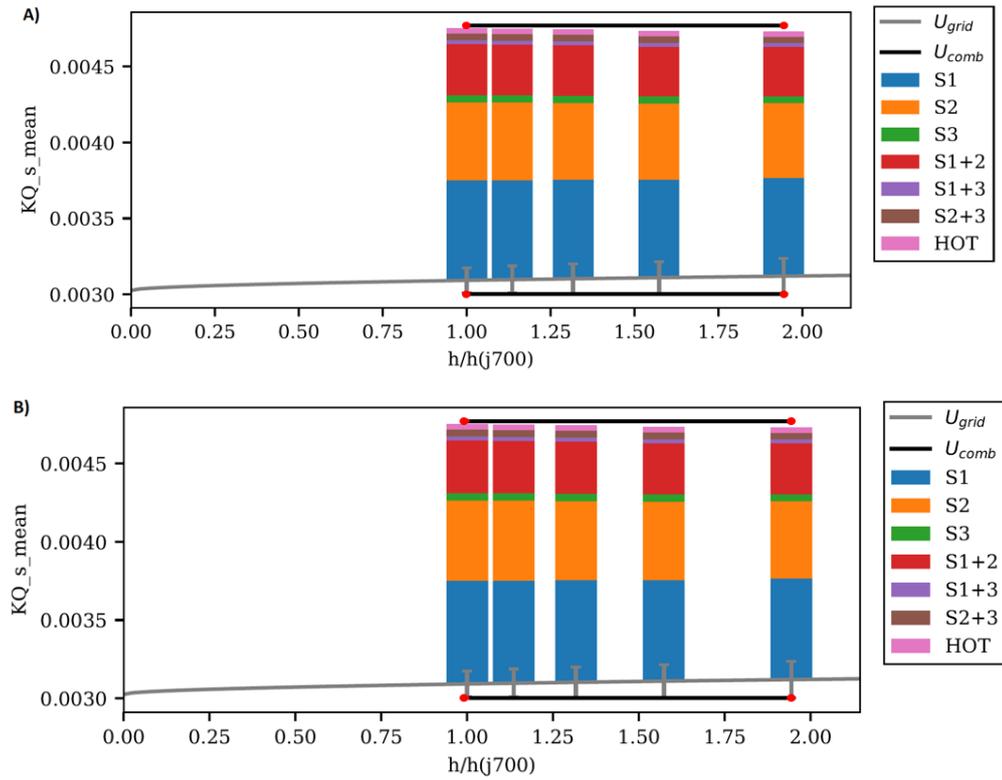
<sup>15</sup> <https://apps.automeris.io/wpd/>

Figura 14 - Comparação do  $KT_{mean}$ .

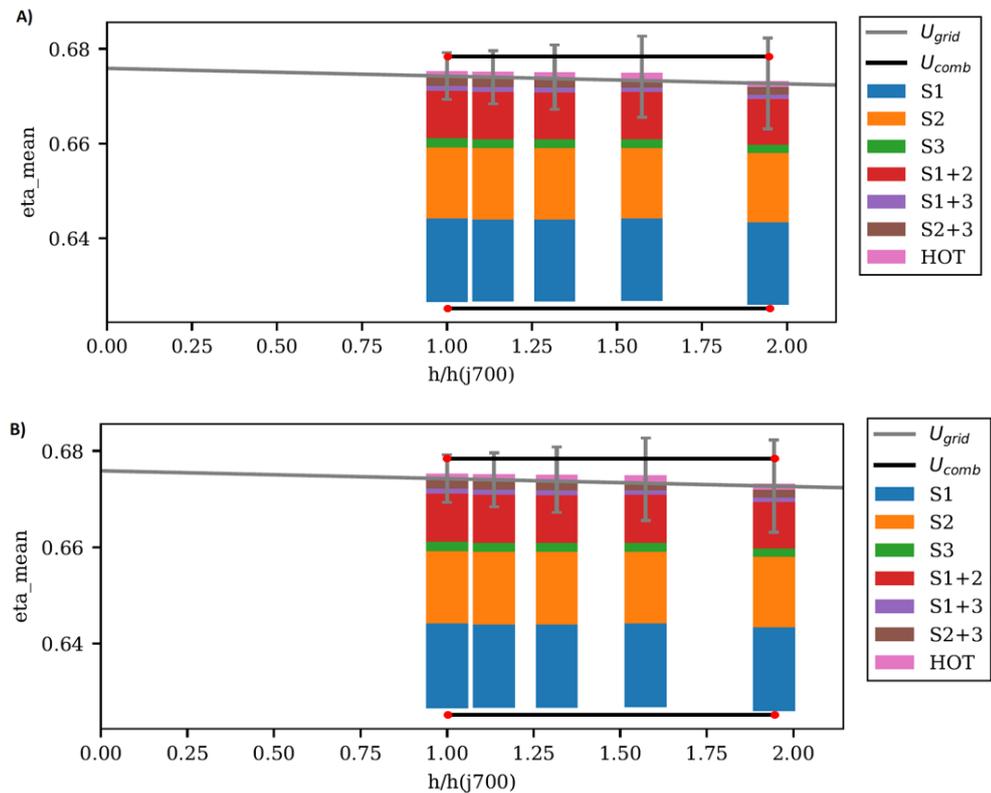
A) A autoria própria (2022). B) Katsuno (2019).

Figura 15 - Comparação do  $KQ_{mean}$ .

A) A autoria própria (2022). B) Katsuno (2019).

Figura 16 - Comparação do  $KQ\_s\_mean$ .

A) Autoria própria (2022). B) Katsuno (2019).

Figura 17 - Comparação do  $\eta_{a\_mean}$ .

A) Autoria própria (2022). B) Katsuno (2019).

## 4.2 RESULTADOS NUMÉRICOS E INCERTEZAS DE DISCRETIZAÇÃO

Após o fim das simulações, foram obtidos os arquivos contendo os resultados de forças, momentos e pressões atuantes nas pás, além dos resultados de forças e momentos atuantes no eixo do hélice (Apêndice C). Além desses resultados, também foi obtido o arquivo contendo os valores máximos de  $y^+$  e o arquivo `boundary.cgns`, o qual sintetiza as informações ao longo da geometria do hélice

A seguir, no Apêndice C são exibidos os gráficos de  $y^+$ , pressão mínima atuante na superfície do hélice e T para cada um dos casos analisados, o que permite verificar a convergência das simulações.

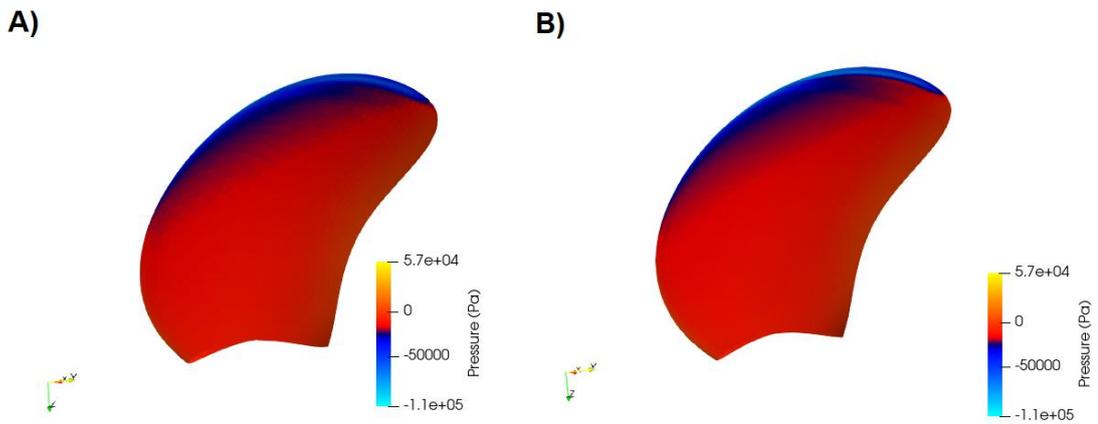
Analisando os gráficos do Apêndice C, observa-se que, segundo o critério proposto por Katsuno (2019), ocorreu a convergência das simulações antes do limite imposto de 20.000 iterações. A rigor, cerca de 10.000 iterações já seriam suficientes para a convergência dos resultados. No entanto, optou-se por manter o limite de iterações originalmente proposto visando evitar futuros problemas de convergência das simulações.

Utilizando o programa Paraview 5.10.0-RC1<sup>16</sup>, foram geradas imagens do campo de pressão atuante na superfície das pás do hélice. Nas Figuras 18, 19, 20, 21 e 22 são apresentadas a comparação gráfica entre as malhas  $f0,325$  (imagem A) e  $f0,700$  (imagem B).

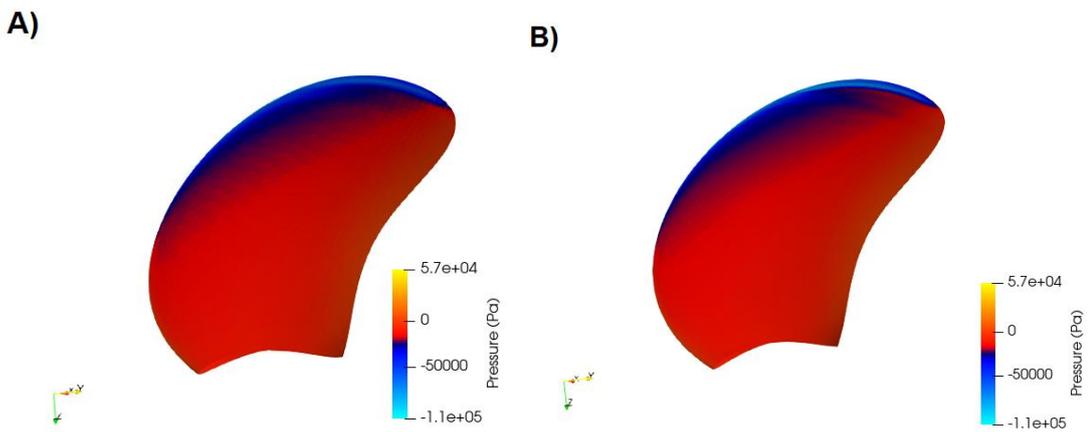
As imagens apresentadas permitem uma avaliação qualitativa da distribuição do campo de pressão nas superfícies das pás, permitindo avaliar o potencial de cavitação em cada região. Por fim destaca-se que, devido ao baixo processamento computacional disponível, apenas uma das cinco pás é mostrada.

---

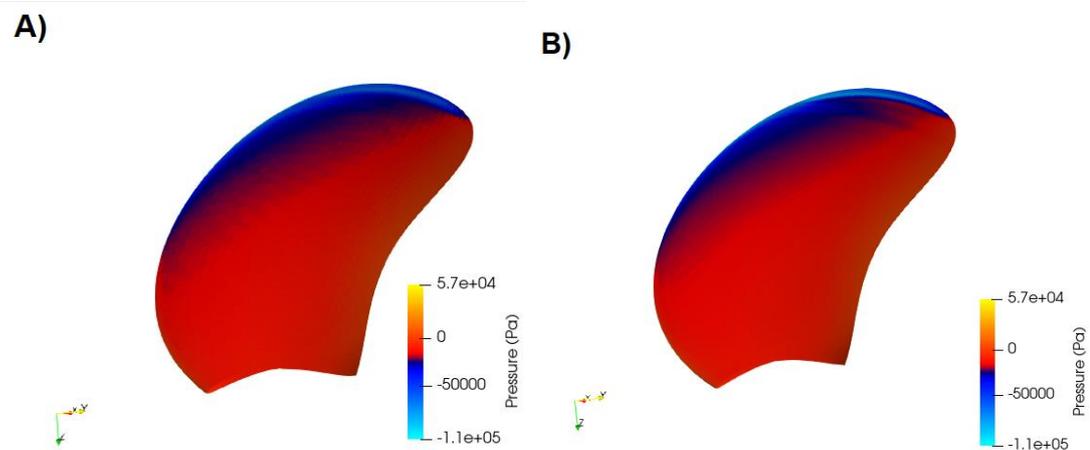
<sup>16</sup> <https://www.paraview.org>.

Figura 18 - Comparação de P entre as malhas  $f_{0,325}$  e  $f_{0,700}$  para o caso 0.

Fonte: autoria própria (2022).

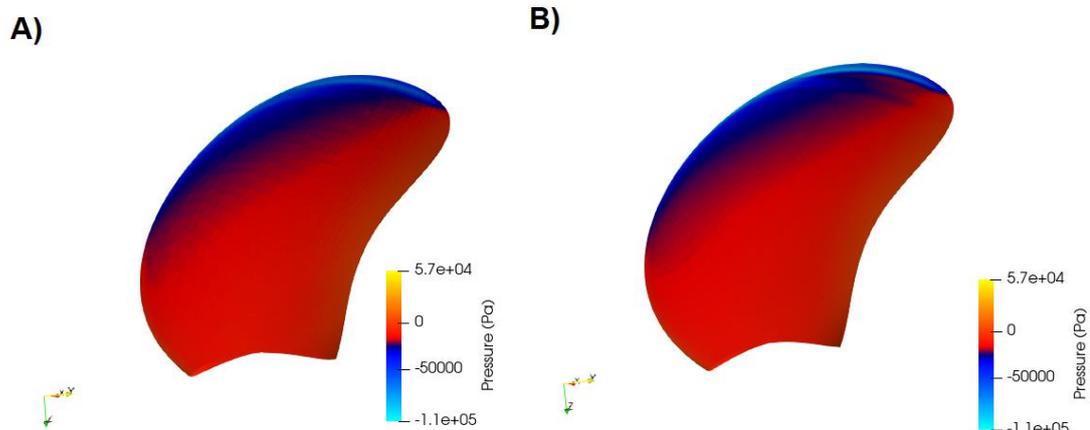
Figura 19 - Comparação de P entre as malhas  $f_{0,325}$  e  $f_{0,700}$  para o caso 1.

Fonte: autoria própria (2022).

Figura 20 - Comparação de P entre as malhas  $f_{0,325}$  e  $f_{0,700}$  para o caso 2.

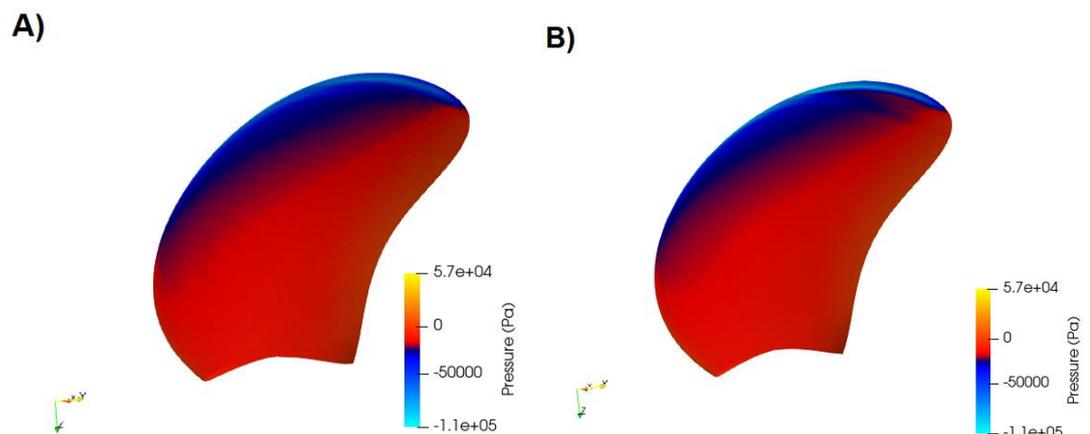
Fonte: autoria própria (2022).

Figura 21 - Comparação de P entre as malhas  $f0,325$  e  $f0,700$  para o caso 3.



Fonte: autoria própria (2022).

Figura 22 - Comparação de P entre as malhas  $f0,325$  e  $f0,700$  para o caso 4.



Fonte: autoria própria (2022).

Analisando cuidadosamente as imagens é possível observar que as malhas mais refinadas apresentam um maior nível de definição da superfície e uma maior discretização dos resultados quando comparadas às malhas menos refinadas.

Embora os gráficos plotados apresentem o mesmo intervalo de pressão, a distribuição do campo de pressões é afetada pela qualidade da malha, sendo importante considerar os valores de todos os elementos contidos na superfície da geometria, e não apenas os valores extremos. Ao longo do trabalho observou-se que valores extremos podem estar restritos a regiões de pouco interesse, ou ainda, estarem presentes em um número pequeno de elementos, o que levaria a conclusões errôneas sobre o início efetivo da cavitação.

Outro ponto importante é o efeito que a velocidade de rotação tem sobre o campo de pressões ao longo da superfície das pás. É possível observar nas figuras que o aumento da rotação do propulsor gera um aumento na região de baixa pressão do hélice, o que potencializa o início da cavitação.

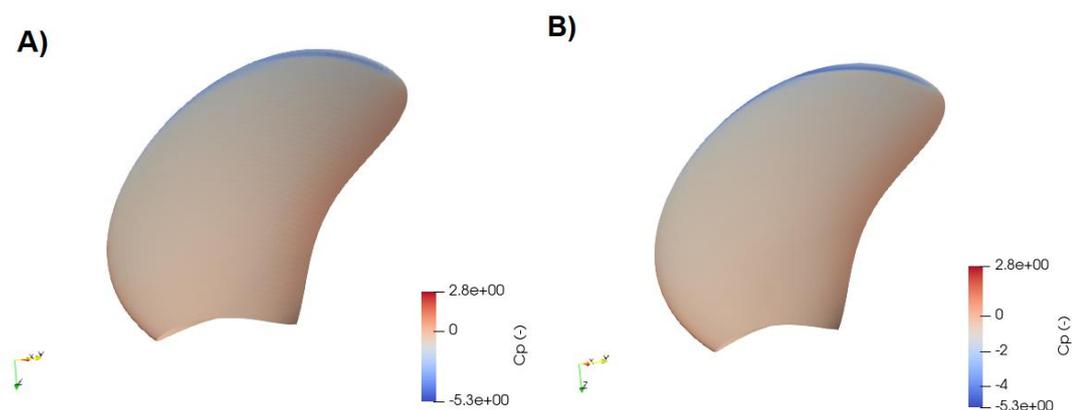
Utilizando o campo de pressão atuante na superfície do propulsor, foi possível adicionar um filtro ao Paraview e calcular a distribuição do coeficiente de pressão ( $C_P$ ) para cada um dos casos avaliados (Equação 17).

$$C_P(x, y, z) = \frac{Pressão(x, y, z) - P_\infty}{0,5 \times \rho \times (n \times D)^2} \quad (17)$$

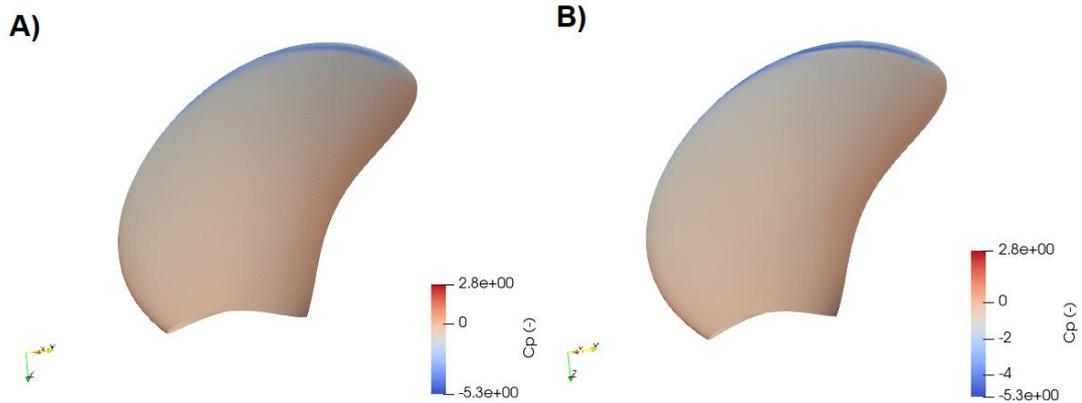
Onde  $P_\infty$  corresponde a pressão de referência utilizada na simulação numérica, sendo considerado igual a 0,000 Pa,  $\rho$  corresponde a densidade, sendo considerado igual a 998,100 kg/m<sup>3</sup>,  $n$  corresponde a velocidade de rotação (RPS),  $D$  corresponde ao diâmetro do propulsor, sendo considerado igual a 0,150 m e  $P_{x,y,z}$  corresponde a pressão atuante no elemento analisado.

Usando a Equação 17, foram plotadas as Figuras 23, 24, 25, 26 e 27, as quais permitem comparar qualitativamente os valores de  $C_P$  atuantes em cada um dos casos analisados.

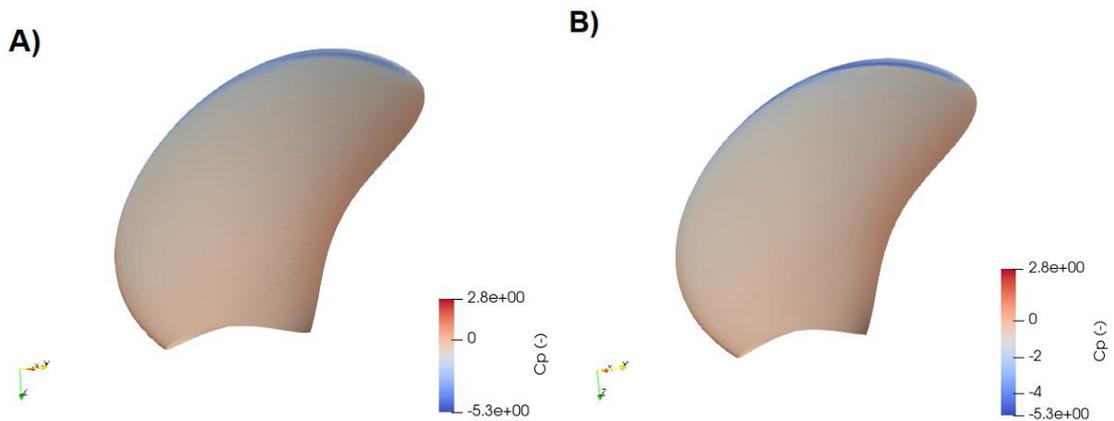
Figura 23 - Comparação de  $C_P$  entre as malhas  $f0,325$  e  $f0,700$  para o caso 0.



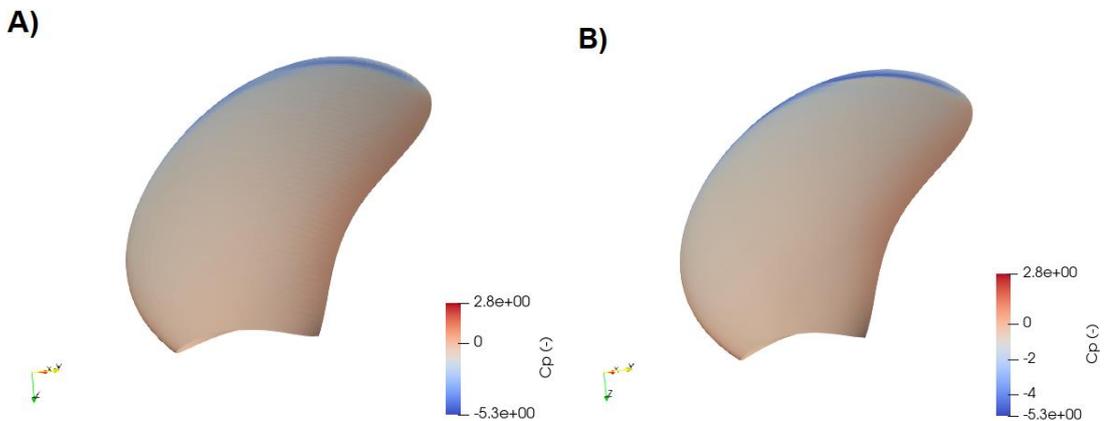
Fonte: autoria própria (2022).

Figura 24 - Comparação de  $C_p$  entre as malhas  $f_{0,325}$  e  $f_{0,700}$  para o caso 1.

Fonte: autoria própria (2022).

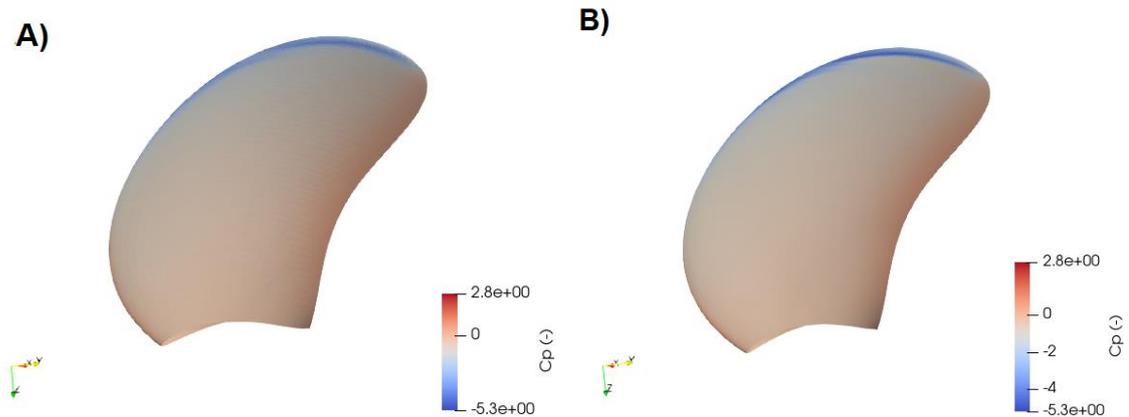
Figura 25 - Comparação de  $C_p$  entre as malhas  $f_{0,325}$  e  $f_{0,700}$  para o caso 2.

Fonte: autoria própria (2022).

Figura 26 - Comparação de  $C_p$  entre as malhas  $f_{0,325}$  e  $f_{0,700}$  para o caso 3.

Fonte: autoria própria (2022).

Figura 27 - Comparação de  $C_P$  entre as malhas  $f0,325$  e  $f0,700$  para o caso 4.



Fonte: autoria própria (2022).

Nas figuras acima, é possível observar as mesmas características vistas na análise dos campos de pressão. Ou seja, quanto maior o refino da malha, melhor a exatidão e o detalhamento da distribuição do  $C_P$ .

Por fim, observa-se que, na malha com refino igual a  $f0,700$ , os valores de  $C_P$  na região típica de início da cavitação dos tipos folha e nuvem ficaram mais próximos do limite ( $\sigma$ ) definido de  $-2,100$ , o que confirma que as condições simuladas estão na fronteira do início desses tipos de cavitação.

Após a plotagem dos valores de  $C_P$ , foram calculados os valores de  $K_T$ ,  $K_Q$  e  $\eta$  utilizando o código apresentado no Anexo B, bem como as incertezas de tais valores (Anexo D) devido ao refino de malhas para cada um dos cinco valores de RPM estudados. Por fim, no Apêndice D são mostrados os diagramas de águas abertas e as incertezas resultantes do refino de malha para todos os cinco valores de RPM estudados.

Analisando os gráficos de incerteza apresentados no Apêndice D, observa-se o efeito que o refino da malha exerce na análise de incertezas de cavitação e na simulação numérica em si. Já no gráfico de águas abertas, é possível observar que os resultados obtidos ficam dentro da região de análise considerada (retângulo laranja), e as incertezas de discretização obtidas são consideravelmente maiores para malhas menos refinadas, alcançando valores até três vezes maiores. Além disso, para a variável  $C_{AVINDEX}$ , foco deste trabalho, as incertezas de discretização são consideráveis, mostrando alta dependência com o refino da malha.

Em relação aos efeitos cruzados, observa-se que as incertezas de discretização das variáveis  $K_T$ ,  $K_Q$  e  $\eta$  são pouco afetadas pela variação da velocidade

de rotação do hélice. Já em relação a variável  $C_{aVINDEX}$ , observou-se uma redução das incertezas de discretização com o aumento da velocidade de rotação do propulsor.

É possível notar ainda que as malhas mais refinadas possuem menos oscilações em relação à linha de tendência obtida. Além disso, é possível observar que a incerteza de discretização de  $K_T$  para a malha  $f_{0,325}$  não segue a linha de tendência, o que pode indicar que esse refino de malha é excessivamente baixo, levando à resultados distorcidos.

Outro ponto importante é que, mesmo para as malhas menos refinadas, as incertezas obtidas para  $K_T$ ,  $K_Q$  e  $\eta$  se encontram abaixo de 10% e podem ser consideradas relativamente baixas, já que os resultados avaliados são pequenos e qualquer pequena variação numérica pode gerar grandes variações percentuais.

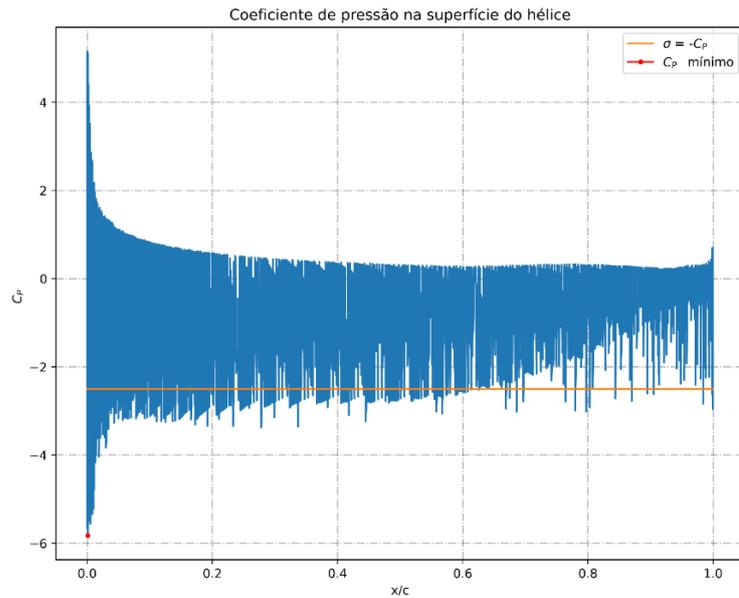
Tendo em vista ainda que na grande maioria dos casos analisados a incerteza segue uma linha de tendência e que as maiores incertezas estão relacionadas às malhas menos refinadas, considera-se que os resultados são consistentes. Para o índice de cavitação, as altas incertezas referentes ao refino de malha também podem ser consideradas coerentes, já que esse parâmetro é fortemente dependente da variação de pressão sofrida pelo hélice e, devido a região de transição laminar-turbulento, os resultados podem apresentar oscilações consideráveis a depender do refino de malha utilizado na discretização dessa região.

#### 4.3 AVALIAÇÃO E VERIFICAÇÃO DOS RESULTADOS CFD

Após a verificação das simulações numéricas, foram calculados os valores de  $K_T$  e  $C_{PNmédio}$  visando avaliar se esses se encontram dentro da região de análise, a qual representa o início da cavitação dos tipos folha e nuvem.

É importante ressaltar que, no presente estudo, apesar de possuírem significados físicos diferentes,  $\sigma$  e  $C_{PNmin}$  são relacionados pela diferença de pressão. Logo, um valor de  $-C_{PNmin}$  maior que o valor de  $\sigma$  significará que a pressão mínima atuante no hélice será maior que a pressão de vapor do fluido, fazendo com que o fenômeno de cavitação não ocorra (Figura 28).

Figura 28 – Pressão atuante na superfície do hélice e início de cavitação.



Fonte: autoria própria (2022)

Na Figura 28 o coeficiente de pressão é representado no eixo y, enquanto a razão  $x/C$  (posição avaliada/corda) está disposta no eixo x. Uma explicação mais detalhada dessa relação pode ser vista no Apêndice E e em Molland e Turnock (2007).

A seguir, utilizando a Equação 7, foram calculados e plotados os valores de  $C_{PNmédio}$  (Apêndice F), visando verificar a ocorrência do início da cavitação. Para isso, foi avaliado o valor de  $P_{\infty}$  necessário para que a diferença  $P_{\infty} - P_V$  se igualasse ao valor experimental. Com isso, utilizando a Equação 5,  $P_{\infty}$  foi definido como 49.240,156 Pa. Essa etapa é necessária pois a pressão relativa usada nas simulações foi de 0 Pa. Logo, para atingir os valores experimentais é necessário aplicar uma correção aos valores de pressão utilizados na simulação.

Assim,  $P_{ATM}$  foi considerado igual a 43.240,000 Pa,  $P_V$  igual a -6.000,000 Pa,  $P_{HIDROSTÁTICA}$  igual a 0 Pa. Sendo importante destacar que o valor de  $P_{HIDROSTÁTICA}$  pouco afeta os resultados, tendo em vista que, no caso de propulsores pequenos (modelos), esse valor é bastante reduzido. Por fim, a variação de  $P_{HIDROSTÁTICA}$  ao longo do diâmetro do propulsor também foi desconsiderada, já que a gravidade não foi considerada nas simulações.

Os valores das variáveis de saída e os limites de  $K_T$  e  $C_{PNmédio}$  são mostrados nas Tabelas 6, 7, 8, 9, 10 e 11, enquanto os gráficos podem ser vistos no Apêndice F.

Tabela 6 – Variáveis de saída para o caso 0.

Número de elementos	$K_T$	$K_Q$	eta	CPNmédio
2.919.825	0,251	0,404	0,572	2,184
5.164.000	0,251	0,403	0,574	2,175
8.796.570	0,251	0,402	0,575	2,171
14.250.060	0,251	0,401	0,575	2,168
21.620.280	0,250	0,400	0,575	2,165
31.164.240	0,250	0,400	0,576	2,163

Fonte: autoria própria (2022).

Tabela 7 – Variáveis de saída para o caso 1

Número de elementos	$K_T$	$K_Q$	eta	CPNmédio
2.919.825	0,255	0,409	0,567	2,117
5.164.000	0,256	0,408	0,569	2,108
8.796.570	0,255	0,407	0,570	2,103
14.250.060	0,255	0,406	0,570	2,100
21.620.280	0,254	0,405	0,571	2,097
31.164.240	0,254	0,405	0,571	2,095

Fonte: autoria própria (2022).

Tabela 8 – Variáveis de saída para o caso 2.

Número de elementos	$K_T$	$K_Q$	eta	CPNmédio
2.919.825	0,256	0,414	0,563	2,053
5.164.000	0,260	0,413	0,564	2,042
8.796.570	0,260	0,412	0,565	2,038
14.250.060	0,259	0,411	0,566	2,035
21.620.280	0,259	0,410	0,566	2,031
31.164.240	0,258	0,410	0,566	2,029

Fonte: autoria própria (2022).

Tabela 9 – Variáveis de saída para o caso 3.

Número de elementos	$K_T$	$K_Q$	eta	$CPN_{\text{médio}}$
2.919.825	0,264	0,419	0,558	1,990
5.164.000	0,264	0,418	0,559	1,981
8.796.570	0,264	0,417	0,560	1,975
14.250.060	0,263	0,416	0,561	1,971
21.620.280	0,263	0,415	0,561	1,968
31.164.240	0,263	0,414	0,561	1,966

Fonte: autoria própria (2022).

Tabela 10 – Variáveis de saída para o caso 4.

Número de elementos	$K_T$	$K_Q$	eta	$CPN_{\text{médio}}$
2.919.825	0,267	0,423	0,553	1,930
5.164.000	0,268	0,423	0,555	1,920
8.796.570	0,268	0,422	0,556	1,914
14.250.060	0,267	0,421	0,556	1,910
21.620.280	0,267	0,420	0,556	1,907
31.164.240	0,267	0,419	0,557	1,904

Fonte: autoria própria (2022).

Tabela 11 - Limite superior e inferior para  $K_T$  e  $CPN_{\text{médio}}$ .

Variáveis	Limite superior	Limite inferior
$K_T$	0,268	0,219
$CPN_{\text{médio}}$	2,251	1,843

Fonte: autoria própria (2022).

Avaliando os resultados acima, é possível notar que para o caso 4, malhas menos refinadas, como  $f_{0,400}$  e  $f_{0,475}$ , apresentam valores de  $K_T$  no limite da região estudada. Entretanto, deve-se considerar que o RPM para esse caso é o mais elevado e que os resultados obtidos com as malhas mais refinadas ( $f_{0,550}$ ,  $f_{0,625}$  e  $f_{0,700}$ ) se encontram dentro da região estudada. Logo, tais valores podem decorrer do uso de um refino de malha insuficiente para representar o alto valor de rotação simulado.

Já o valor de  $K_T$  para a malha  $f_{0,325}$  não ultrapassa os limites estipulados, porém, como visto anteriormente, o valor de  $K_T$  para a malha menos refinada tende a

ser um pouco menor e não seguir a linha de tendência, logo, para todos os efeitos, ele deve ser considerado como um resultado anômalo.

#### 4.4 VARIÁVEIS DE SAÍDA

Com os cinco valores de RPM e os cinco valores de  $P_v$  selecionados com base na amostragem fatorial, foi possível combinar os valores a fim de observar o efeito que ambas as variáveis de entrada exercem nas variáveis de saída. Para isso, foi desenvolvido um arquivo em formato CSV para cada malha contendo todas as possíveis combinações das variáveis de entrada e suas respectivas variáveis de saída.

Considerando que as simulações CFD utilizaram um valor de  $P_{REF}$  igual a 0 Pa, nessa etapa da análise, essa pressão foi corrigida visando ajustar os valores de  $P_v$  das simulações ( $P_{v,CFD}$ ). Na Tabela 12 são mostrados os valores combinados de RPM e  $P_v$  juntamente com as variáveis de saída calculadas utilizando as pressões de vapor corrigidas. O arquivo CSV para a malha  $f0,325$  é apresentado na Tabela 12, sendo que as demais malhas seguem o mesmo padrão.

Tabela 12 – Arquivo csv utilizado para análise de incertezas numéricas.

RPM	$P_v$	KT_mean	KQ_mean	eta_mean	Ca <sub>VINDEX</sub>
2474,063	2242,500	0,251	0,0404	0,572	-7,713
2474,063	2271,250	0,251	0,0404	0,572	-7,711
2474,063	2300,000	0,251	0,0404	0,572	-7,709
2474,063	2328,750	0,251	0,0404	0,572	-7,708
2474,063	2357,500	0,251	0,0404	0,572	-7,706
2505,781	2242,500	0,255	0,0409	0,567	-7,712
2505,781	2271,250	0,255	0,0409	0,567	-7,711
2505,781	2300,000	0,255	0,0409	0,567	-7,709
2505,781	2328,750	0,255	0,0409	0,567	-7,708
2505,781	2357,500	0,255	0,0409	0,567	-7,706
2537,500	2242,500	0,259	0,0414	0,562	-7,704
2537,500	2271,250	0,259	0,0414	0,562	-7,702
2537,500	2300,000	0,259	0,0414	0,562	-7,701

Continua (...)

RPM	Pv	KT_mean	KQ_mean	eta_mean	CaVINDEX
2537,500	2328,750	0,259	0,0414	0,562	-7,700
2537,500	2357,500	0,259	0,0414	0,562	-7,698
2569,219	2242,500	0,264	0,0419	0,558	-7,729
2569,219	2271,250	0,264	0,0419	0,558	-7,728
2569,219	2300,000	0,264	0,0419	0,558	-7,726
2569,219	2328,750	0,264	0,0419	0,558	-7,725
2569,219	2357,500	0,264	0,0419	0,558	-7,723
2600,938	2242,500	0,267	0,0423	0,553	-7,801
2600,938	2271,250	0,267	0,0423	0,553	-7,800
2600,938	2300,000	0,267	0,0423	0,553	-7,799
2600,938	2328,750	0,267	0,0423	0,553	-7,797
2600,938	2357,500	0,267	0,0423	0,553	-7,796

Fonte: autoria própria (2022).

#### 4.5 AVALIAÇÃO DOS ÍNDICES DE SOBOL

Com os arquivos CSV finalizados, foi utilizado o módulo main.py para analisar as incertezas e obter os índices de Sobol. As incertezas foram calculadas para cada uma das seis malhas estudadas, sendo que o número de amostras<sup>17</sup> avaliadas pelo modelo interpolador foi de 8.192 para cada uma das malhas estudadas.

Na Figura 29 é mostrada a sequência utilizada para a análise de incertezas considerando um refino de malha qualquer. A mesma sequência foi utilizada na análise de cada uma das seis malhas estudadas neste trabalho.

Figura 29 - Sequência de análise utilizada.



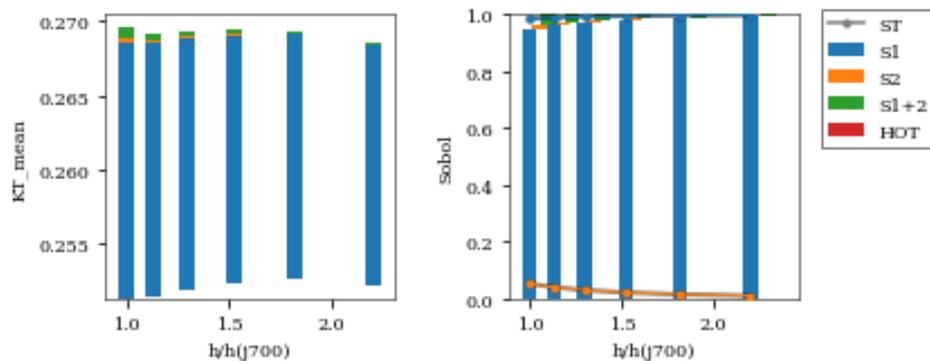
Fonte: autoria própria (2022).

<sup>17</sup> Valor correspondente a  $2^{13}$ , já que as propriedades de convergência dos índices de Sobol na análise são válidas apenas quando o número de amostras pode ser expresso como uma potência de base 2.

Utilizando o módulo `responseSurface.py` (Apêndice G) os modelos interpoladores foram gerados com base nas 25 análises realizadas para cada um dos refinamentos de malha estudados. As superfícies de resposta (modelos interpoladores) geradas para cada refinamento de malha podem ser vistas no Apêndice H.

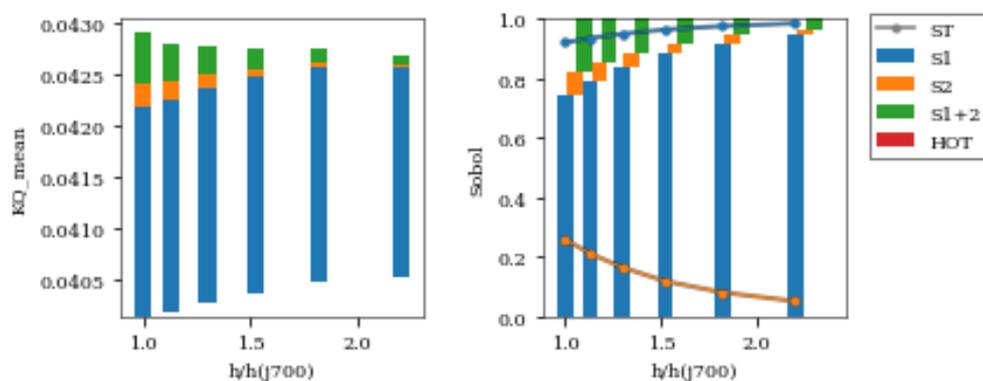
Por fim, foi utilizado o módulo `salib.py` para realizar a análise dos índices de Sobol das variáveis de saída levando em consideração as variáveis de entrada. Os gráficos obtidos nessa análise são mostrados nas Figuras 30, 31, 32 e 33. Onde  $S_1$  representa as incertezas referentes à RPM,  $S_2$  à  $P_V$ ,  $S_{1+2}$  às incertezas cruzadas de RPM e  $P_V$ , HQT representa os termos de alta ordem e  $S_T$  representa os índices de Sobol totais referentes à cada uma das variáveis.

Figura 30 - Índices de Sobol para  $KT_{mean}$ .



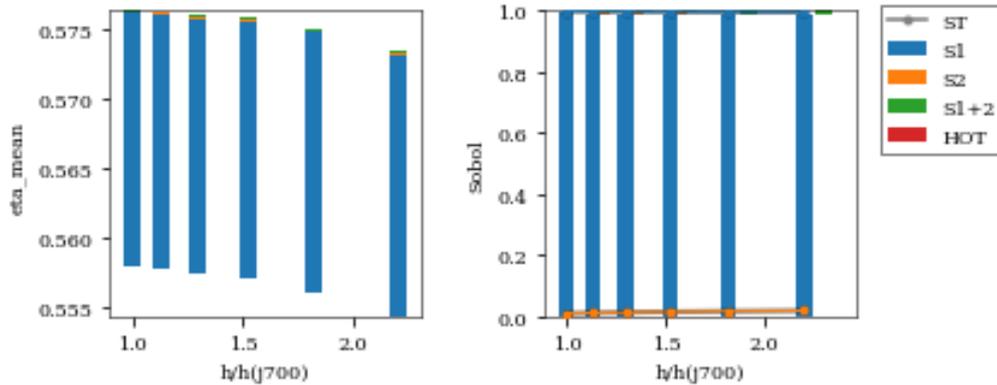
Fonte: autoria própria (2022).

Figura 31 - Índices de Sobol para  $KQ_{mean}$ .



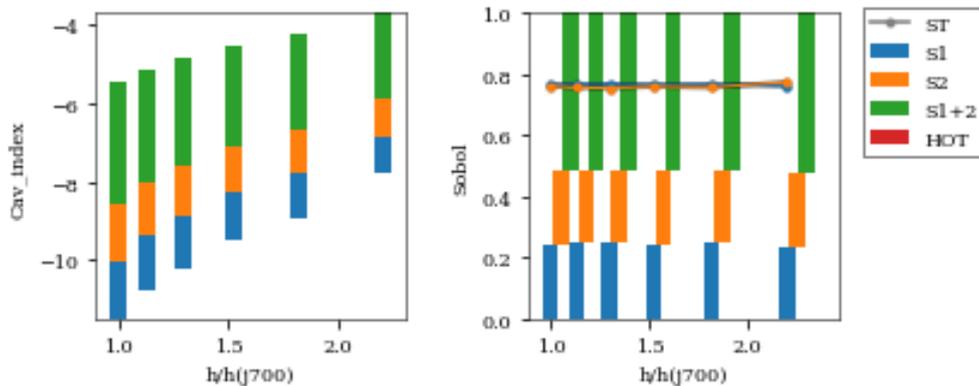
Fonte: autoria própria (2022).

Figura 32 - Índices de Sobol para eta\_mean.



Fonte: autoria própria (2022).

Figura 33 - Índices de Sobol para CavINDEX.



Fonte: autoria própria (2022).

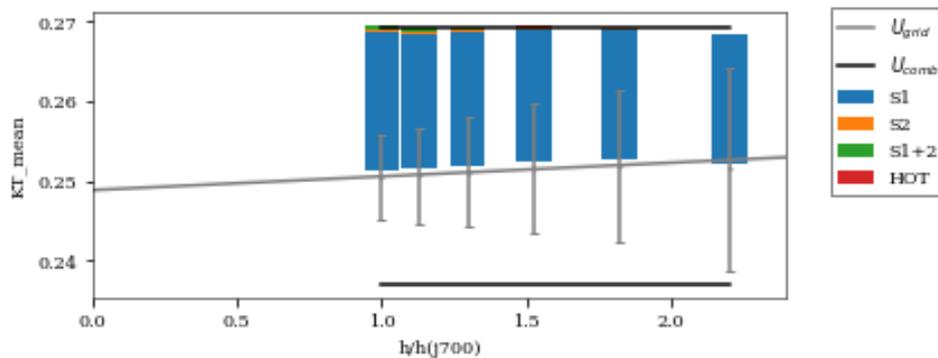
Observando as imagens acima, é possível afirmar que os resultados são coerentes, visto que os parâmetros  $K_T$  e  $K_Q$  apresentam baixas dependências em relação a  $P_V$ . Além disso, o gráfico de  $Cav_{INDEX}$  também segue o esperado, já que esse parâmetro depende de ambas as variáveis de entrada, logo é esperado que as incertezas de entrada de ambas as variáveis afetem os resultados obtidos.

Por fim, é possível observar que os termos de alta ordem (HOT) são nulos, o que é previsto por Sobol (2001) e confirma os resultados obtidos. Pois, como mostrado na Equação 7, estudos contendo apenas duas variáveis de entrada apresentam HOT nulos, já que a soma de todos os índices de Sobol deve ser igual a 1.

#### 4.6 INCERTEZAS DE ENTRADA E DE DISCRETIZAÇÃO

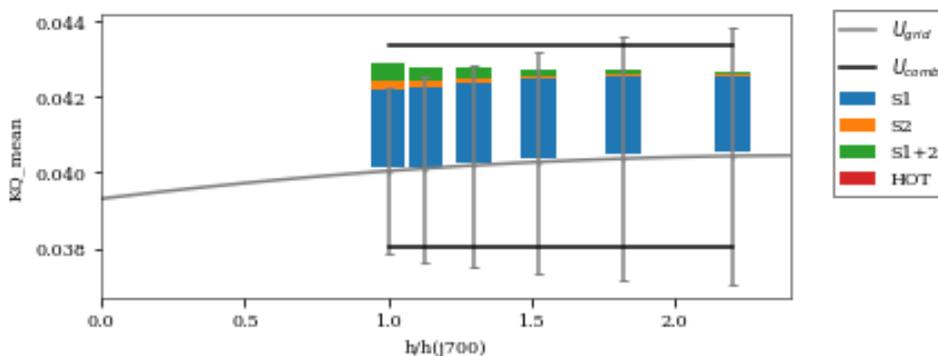
Ao final do processo, foi possível unir as incertezas relacionadas aos refinamentos de malha e aos índices de Sobol em um único gráfico para cada uma das variáveis de saída, resultando nos diagramas mostrados nas Figuras 34, 35, 36 e 37. Onde  $U_{grid}$  representa as incertezas referentes ao refino de malha,  $U_{comb}$  as incertezas combinadas de RPM,  $P_V$  e de discretização,  $S_1$  representa as incertezas referentes a RPM,  $S_2$  as incertezas referentes à  $P_V$ ,  $S_{1+2}$  as incertezas cruzadas de RPM e  $P_V$  e HOT os termos de alta ordem.

Figura 34 - União dos índices de Sobol e das incertezas de discretização (KT\_mean)



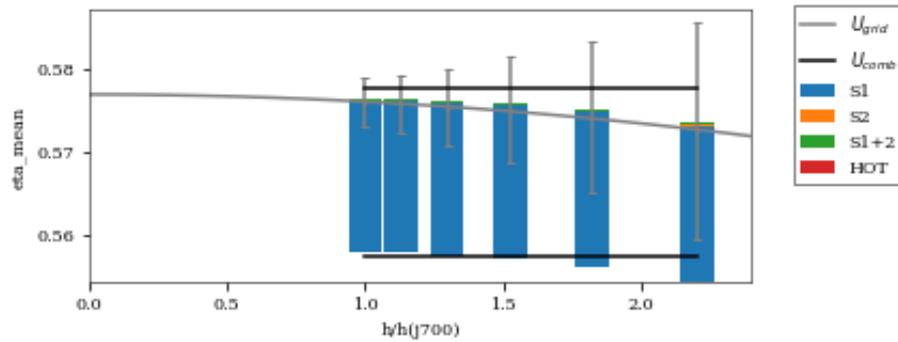
Fonte: autoria própria (2022).

Figura 35- União dos índices de Sobol e das incertezas de discretização (KQ\_mean)



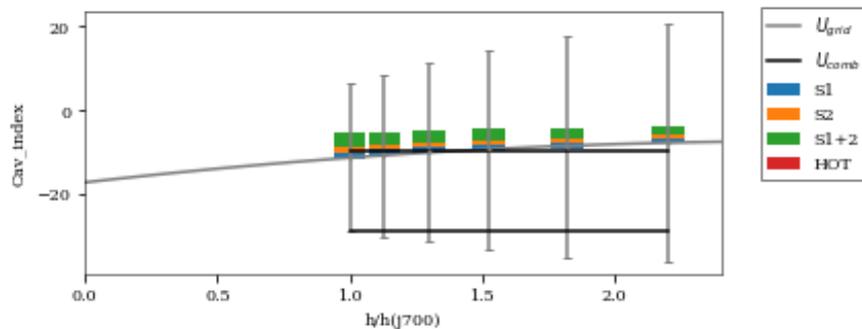
Fonte: autoria própria (2022).

Figura 36- União dos índices de Sobol e das incertezas de discretização (eta\_mean)



Fonte: autoria própria (2022).

Figura 37- União dos índices de Sobol e das incertezas de discretização (Cav\_INDEX)



Fonte: autoria própria (2022).

Nas figuras acima, os limites inferiores e superiores das barras coloridas representam os limites do intervalo de confiança de 95% de cada índice de Sobol. Essas barras consideram apenas as incertezas de entrada, sendo que os resultados referentes a cada refino de malha são mostrados ao longo do eixo x. Já as incertezas relacionadas apenas à discretização (considerando os valores esperados das variáveis de entrada) são representadas pelas barras de erro cinzas, sendo que  $U_{COMB}$  representa o IC do efeito combinado das incertezas de entrada juntamente com as incertezas de discretização.

É importante ressaltar que as incertezas combinadas ( $U_{COMB}$ ) podem ser menores que as incertezas de discretização ( $U_{GRID}$ ). Isso pode ocorrer porque as incertezas de discretização aqui apresentadas utilizam os valores esperados dos dados de entrada, o que pode resultar em valores mais conservadores.

Avaliando os valores de  $K_T$ , observa-se que as incertezas de refino de malha são baixas quando comparadas às incertezas de entrada. Esse resultado é coerente já que parte considerável do empuxo é função da velocidade de rotação do hélice,

sendo pouco afetada pelo fluxo do fluido. Logo, uma malha menos refinada e com baixa precisão não afeta significativamente os resultados. Esses gráficos corroboram os resultados apresentados no Apêndice D, os quais mostram a baixa influência da malha nos valores de  $K_T$ . Logo, para reduzir a incerteza referente ao coeficiente de empuxo do hélice, deve-se, primeiro, reduzir as incertezas referentes as variáveis de entrada e, em seguida, utilizar malhas mais densas com o objetivo de reduzir as incertezas de discretização.

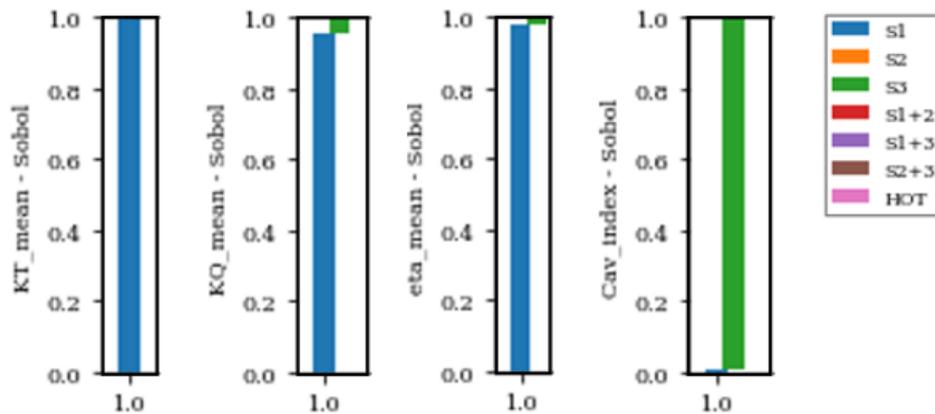
Os valores de  $K_Q$  e  $\eta$  seguem o mesmo comportamento dos valores de  $K_T$ , o que é bastante coerente, considerando que o torque também é fortemente afetado pela velocidade de rotação do hélice e a eficiência do propulsor depende diretamente de  $K_T$  e  $K_Q$ . Ademais, as incertezas relacionadas aos refinamentos de malha são baixas quando comparadas às incertezas referentes aos índices de Sobol. Para reduzir as incertezas dessas variáveis, deve-se também, inicialmente, reduzir as incertezas de entrada, para só então, reduzir as incertezas de discretização.

Para o índice de cavitação, as incertezas de discretização são notavelmente maiores que as incertezas de entrada, sugerindo que esse parâmetro tem baixa influência das incertezas referentes às variáveis de entrada e alta influência das incertezas referentes ao refino de malha, corroborando também com os resultados mostrado no Apêndice D, no qual observa-se que as incertezas de discretização reduzem consideravelmente com o refino de malha. Nesse caso, para reduzir as incertezas, sugere-se, inicialmente, buscar malhas mais refinadas, para, só então, buscar reduzir as incertezas associadas às variáveis de entrada.

A seguir todos os 150 casos (25 pontos de análise para cada uma das seis malhas) foram unificados em um arquivo CSV e uma coluna contendo a quantidade de elementos por malha foi adicionada (Apêndice I). Com esse arquivo foi possível obter os índices de Sobol combinados para as variáveis de saída considerando todas as fontes de incertezas (de entrada e de discretização).

A relação obtida para cada uma das variáveis é mostrada na Figura 38, onde  $S_1$  representa as incertezas referente ao RPM,  $S_2$  à  $P_V$ ,  $S_3$  ao número de elementos da malha,  $S_{1+2}$  aos efeitos cruzados de RPM e  $P_V$ ,  $S_{1+3}$  aos efeitos cruzados de RPM e do número de elementos da malha, e  $S_{2+3}$  aos efeitos cruzados de  $P_V$  e do número de elementos da malha. Os dados utilizados para a montagem do gráfico podem ser vistos no Apêndice J.

Figura 38 - Índices de Sobol combinados.



Fonte: autoria própria (2022).

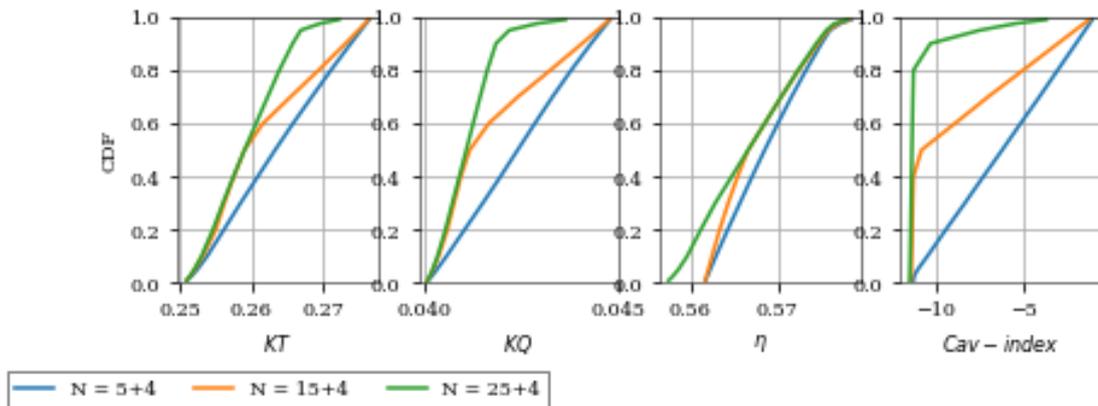
Em geral, as aplicações práticas se interessam por analisar os índices de cavitação na superfície do objeto simulado. Logo, o estudo de refino de malha pode ser uma boa solução quando se busca reduzir incertezas em simulações numéricas. Entretanto, deve-se tomar cuidado já que o regime de escoamento (laminar ou turbulento) é afetado pelas variáveis de entrada, e, portanto, incertezas de entrada podem apresentar grandes efeitos sobre os valores de  $K_T$ ,  $K_Q$  e  $\eta$ .

#### 4.7 EFEITOS DA AMOSTRAGEM

Para verificar se o número de casos analisado por malha foi suficiente para o desenvolvimento do modelo interpolador, três níveis de densidade amostral foram avaliados: 5, 15 e 25. Os níveis estudados incluem os valores contidos no arquivo csv desenvolvido e mais 4 amostras obtidas a partir do módulo `postProcessor_LHS.py` (Anexo E), o qual combina os valores máximos e mínimos das variáveis de entrada.

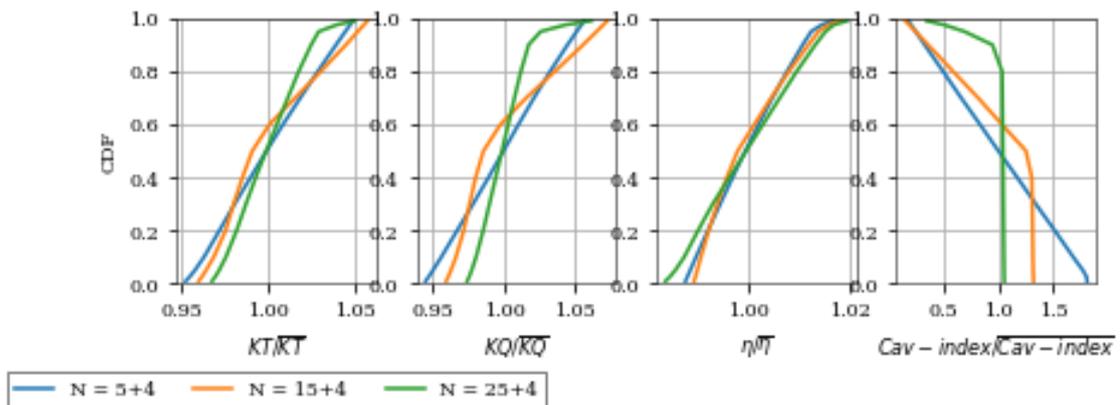
Na Figura 39 é apresentada a CDF das variáveis de saída para a malha  $f_{0,700}$ , enquanto na Figura 40 são apresentados os gráficos adimensionais usando para adimensionalização o valor médio de cada parâmetro representado.

Figura 39 – CDF das variáveis de saída.



Fonte: autoria própria (2022).

Figura 40 - CDF das variáveis de saída em porcentagem em relação à média.



Fonte: autoria própria (2022).

Analisando os parâmetros  $K_T$  e  $K_Q$ , nota-se que ambos apresentam a mesma tendência em relação as incertezas de entrada, e como  $\eta$  é dependente de ambos, o seu cálculo pode amplificar certas incertezas contidas nas variáveis, levando a incertezas ainda maiores.

Já o índice de cavitação apresenta alta dependência das incertezas associadas ao refino de malha. Dependendo da quantidade de amostras utilizadas, a curva CDF apresenta altas variações e, baseado em sua forma (Figura 40), o número de amostras utilizado não é o ideal, pois a curva para 25+4 amostras não se comporta de maneira suave e, quando calculado em função do valor médio, apresenta uma tendência oposta ao esperado para uma CDF.

Um fator importante para a análise das curvas CDF é o número de amostras utilizadas. Avaliando os gráficos, pode-se notar que as curvas apresentam quinas

pronunciadas nos três níveis de amostragem e variações abruptas de curvatura quando o nível de densidade amostral varia.

Apesar disso, a curva para 25+4 pontos apresenta uma maior suavidade do que as outras duas curvas e, por ser avaliado um intervalo pequeno de valores, um comportamento próximo ao linear é esperado, o que significa que esse é um número aceitável para a realização dos modelos interpoladores utilizados neste trabalho.

Pois, embora o aumento do número de amostras pudesse levar a uma superfície mais suave e precisa, essa abordagem resultaria em um maior custo computacional.

## 5. CONCLUSÃO

Apresentou-se neste trabalho uma análise quantitativa sobre a influência que as incertezas associadas às variáveis de entrada de uma simulação CFD exercem nos resultados associados a previsão de início de cavitação de um propulsor Duisburg P1570 operando totalmente submerso.

Entende-se que os objetivos propostos foram atingidos uma vez que foi possível estabelecer um procedimento capaz de quantificar as incertezas existentes na análise de um propulsor naval operando totalmente submerso, além de desenvolver uma estrutura numérica capaz de quantificar a influência das incertezas associadas às variáveis de entrada nas variáveis de saída da simulação.

Observou-se que a abordagem não-intrusiva se mostrou adequada para a análise proposta, tendo em vista que tal abordagem não exige acesso direto à modelagem, pois trata a simulação como uma caixa preta a ser executada. Já o uso do modelo interpolador, mostrou-se eficiente, pois reduziu drasticamente o número de simulações a serem realizadas, e, ainda sim, manteve a precisão e acurácia dos resultados.

Avaliando os resultados, verificou-se que até 2% das variações observadas nos valores de  $K_T$ ,  $K_Q$  e  $\eta$  se devem as incertezas de discretização, enquanto 98% dessas variações se devem as incertezas de entrada. Isso pode ser explicado pela relação direta que essas variáveis tem com o RPM, e pelas regiões de baixo  $Re$ , as quais são influenciadas pelas condições de contorno, cujo comportamento pode afetar o coeficiente de atrito, e, portanto, o torque gerado pelo propulsor.

Já para o índice de cavitação ( $C_{AVINDEX}$ ), as incertezas de discretização correspondem a 99% das variações observadas, enquanto as incertezas de entrada correspondem a apenas 1%. Muito provavelmente, isso ocorre devido ao fato da região de transição laminar-turbulento gerada na superfície das pás do propulsor ser fortemente afetada pela discretização da malha.

Analisando a CDF obtida, é possível observar que um maior número de dados resultaria em uma análise mais realista, pois um maior número de amostras resultaria em um modelo interpolador mais preciso. Por conta disso, para trabalhos futuros, sugere-se a simulação de um número maior de combinações dos valores de entrada,

visando aumentar a quantidade de pontos utilizados na elaboração dos modelos interpoladores.

Ademais, é incentivada a avaliação de outras variáveis de entrada, tais como a densidade do fluido e a rugosidade das superfícies das pás do propulsor, o que pode gerar resultados interessantes sobre o efeito que essas variáveis, não exploradas por este trabalho, exercem nos resultados associados ao início da cavitação. Também sugere-se utilizar malhas mais refinadas e intervalos maiores para RPM e  $P_v$ , sendo possível, inclusive, avaliar outros tipos de cavitação.

## REFERÊNCIAS

- BIRVALSKI, M. et al. Developments in micro-bubble measurement techniques for cavitation and PIV experiments. **Proceedings of the 6<sup>th</sup> INTERNATIONAL CONFERENCE ON ADVANCED MODEL MEASUREMENT TECHNOLOGY FOR THE MARITIME INDUSTRY (AMT'19)**, Rome, IT, out. 2019. Disponível em: [https://www.researchgate.net/publication/336718286\\_DEVELOPMENTS\\_IN\\_MICRO\\_BUBBLE\\_MEASUREMENT\\_TECHNIQUES\\_FOR\\_CAVITATION\\_AND\\_PIV\\_EXPERIMENTS](https://www.researchgate.net/publication/336718286_DEVELOPMENTS_IN_MICRO_BUBBLE_MEASUREMENT_TECHNIQUES_FOR_CAVITATION_AND_PIV_EXPERIMENTS). Acesso em 23 out. 2021.
- BRUNVOLL. **CRP (Contra rotating propellers)**. Disponível em: <https://www.brunvoll.no/products/crp-contra-rotating-propellers>. Acesso em: 02 mar. 2022.
- CODER, J; MAUGHMER, M. Application of the amplification factor Transport Transition Model to the Shear Stress Transport Model. **Proceeding of the 53<sup>rd</sup> AIAA AEROSPACE SCIENCES MEETING**, p. 1-16, jan. 2015. Disponível em: [https://www.researchgate.net/publication/306033225\\_Application\\_of\\_the\\_Amplification\\_Factor\\_Transport\\_Transition\\_Model\\_to\\_the\\_Shear\\_Stress\\_Transport\\_Model](https://www.researchgate.net/publication/306033225_Application_of_the_Amplification_Factor_Transport_Transition_Model_to_the_Shear_Stress_Transport_Model). Acesso em: 09 ago. 2021.
- CONFEDERAÇÃO NACIONAL DO TRANSPORTE. **Pesquisa Aquaviária CNT: portos marítimos longo curso e cabotagem**. Brasília: CNT, 2006. Disponível em: [https://issuu.com/transporteatual/docs/pesquisa\\_portos\\_mar\\_timos\\_-\\_longo\\_](https://issuu.com/transporteatual/docs/pesquisa_portos_mar_timos_-_longo_). Acesso em 16 ago. 2021.
- CONFERÊNCIA DAS NAÇÕES UNIDAS SOBRE COMÉRCIO E DESENVOLVIMENTO. **Review of maritime transport 2019**. 2019. Disponível em: [https://unctad.org/en/PublicationsLibrary/rmt2019\\_en.pdf](https://unctad.org/en/PublicationsLibrary/rmt2019_en.pdf). Acesso em: 14 ago. 2021.
- DECATLHON. **Remo para caiaque modelo rio grande** .Disponível em: <https://www.decathlon.com.br/-remo-caiaque-modelo-rio-grande-214/p>. Acesso em: 02 mar. 2022.
- DELFT UNIVERSITY OF TECHNOLOGY. **Cavitation in ship propulsion**. Mekelweg: TU Delft, 2012.
- EÇA, L., HOEKSTRA, M. A procedure for the estimation of the numerical uncertainty of CFD calculations based on grid refinement studies. **Journal of Computational Physics**, v. 262, p. 104-130, abr. 2014.
- EÇA, L. et al. Validation exercises of mathematical models for the prediction of transitional flows. **Proceedings of the 31<sup>st</sup> SYMPOSIUM ON NAVAL HYDRODYNAMICS**, Monterey, CA, p. 11-16, set. 2016. Disponível em: <https://www.marin.nl/en/publications/validation-exercises-of-mathematical-models-for-the-prediction-of-transitional-flows>. Acesso em: 16 ago. 2021.
- EL MOCTAR, O.; SHINGUNOV, V. Duisburg test case: Post-Panamax container ship for benchmarking. **Ship Technology Research**, v. 59, n. 3, p. 50-64, ago. 2012.

GOLF, R. **Kavitationsuntersuchung am DTC Propeller**. Dissertação de Graduação em Duisburg-Essen University, Alemanha, 1995. Disponível em: [https://www.uni-due.de/IST/ismt\\_bachelorarbeiten\\_2018.php](https://www.uni-due.de/IST/ismt_bachelorarbeiten_2018.php). Acesso em: 26 nov. 2021.

HERDZIK, J. Problems of propulsion systems and main engines choice for offshore support vessels. **Zeszyty Naukowe/Akademia Morska w Szczecinie**, Szczecin, v. 36, n. 2, p. 45-50, 2013.

HERMAN, J. **SALib - Sensitivity Analysis Library in Python**. Disponível em: <https://salib.readthedocs.io/en/latest/>. Acesso em: 07 fev. 2022.

HOMMA, T.; SALTELLI A. Use of Sobol's quasirandom sequence generator for integration of modified uncertainty importance measure. **Journal of nuclear science and technology**, v. 32, n. 11, p.1164-1173, nov. 1995.

KATSUNO, E et al. Estimating parameter and discretization uncertainties using a laminar–turbulent transition model. **Computers and Fluids**, v. 230, n. 105129, nov. 2021.

KATSUNO, E. Uncertainty quantification analysis of flat plate and model-scale propeller using laminar-turbulent transition model. 2019. Report n. **70060-3-RD**. MARIN. v. 2.1. Wageningen, 2019. Disponível em: [https://www.researchgate.net/publication/336564327\\_Parameter\\_Uncertainty\\_Quantification\\_applied\\_to\\_the\\_Duisburg\\_Propeller\\_Test\\_Case](https://www.researchgate.net/publication/336564327_Parameter_Uncertainty_Quantification_applied_to_the_Duisburg_Propeller_Test_Case). Acesso em: 26 nov. 2021.

KAWASAKI. **Propulsor lateral**. Disponível em: <https://global.kawasaki.com/br/mobility/marine/machinery/side.html>. Acesso em: 02 mar. 2022.

KINACI, O.; GOKCE, M. A computational hydrodynamic analysis of Duisburg test case with free surface and propeller. **Brodogradnja**, v. 66, n. 4, p. 23-38, dez. 2015.

KORONOVICZ, T. et al. A complete design of tandem co-rotating propellers using the new computer system. **Polish Maritime Research**, v. 17, n. 4, p 17-25, jan. 2010.

KUIPER, G. Cavitation in ship propulsion. **TU Delft OpenCourseWare**, jun. 2012. Disponível em: [https://ocw.tudelft.nl/wp-content/uploads/Chapter\\_4.pdf](https://ocw.tudelft.nl/wp-content/uploads/Chapter_4.pdf). Acesso em: 15 ago. 2021.

LEME, F. **Análise da influência do parâmetro corda/diâmetro na eficiência de hélices utilizando teoria de linha de sustentação**. Joinville: Universidade Federal de Santa Catarina, Trabalho de Conclusão de Graduação, SC. Disponível em: <https://repositorio.ufsc.br/handle/123456789/223273>. Acesso em: 10 out. 2021.

MARITIME RESEARCH INSTITUTE NETHERLANDS. **ReFRESCO Documentation: Tutorials**. Wageningen: MARIN, 2021.

MENTER, F. R. et al. A correlation-based transition model using local variables - Part I: model formulation. **Journal of Turbomachinery**, v. 128, p. 413-422, jul. 2006

MOLLAND, A; TURNOCK, S. **Marine rudders and control surfaces: Principles, data, design and applications**. Southampton: Elsevier, 2007.

MOLLAND, A et al. **Ship resistance and propeller**. Cambridge: Cambridge university press, 2011.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION. **CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences**. Hampton: NASA, 2014.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION. **A user's guide to CGNS**. Hampton: NASA, 2001.

OLIVEIRA, V. **Propulsores modernos e a inovação dos seus reparos subaquáticos**. Rio de Janeiro: Marinha do Brasil, Trabalho de Conclusão de Graduação, RJ. Disponível em: <http://www.redebim.dphdm.mar.mil.br/vinculos/000005/00000556.pdf>. Acesso em: 10 fev. 2022.

ORTLINGHAUS. **Z-drive in propulsion technology**. Disponível em: <https://www.ortlinghaus.com/english/industries/marine-technology/propulsion-technology/z-drive/z-drive.html>. Acesso em 02 mar. 2022.

PEREIRA, A. et al. **Apostila análise fatorial**. Brasil: Universidade de Passo Fundo (UPF), 2019.

PINHO, J. **Efeito da modelagem submalha em simulações de grandes escalas de jatos coaxiais turbulentos**. Porto Alegre: Universidade Federal do Rio Grande do Sul, Tese de Doutorado, RS. Disponível em: <https://repositorio.ifsc.edu.br/>. Acesso em: 02 fev. 2022.

RAKHIMOV, A. C.; VISSER, D. C.; KOMEN, E. M. Uncertainty quantification method for CFD applied to the turbulent mixing of two water layers. **Nuclear Engineering and Design**, v. 333, p. 1-15, abr. 2018.

RUMSEY, C. **Turbulence Modeling Resource**. Disponível em: <https://turbmodels.larc.nasa.gov/sst.html>. Acesso em: 06 fev. 2022.

SALTELLI, A. et al. **Sensitivity analysis in practice: a guide to assessing scientific models**. Itália: John Wiley & Sons, Ltd., 2004.

SALTELLI, A. et al. **Global sensitivity analysis**. Inglaterra: John Wiley & Sons, Ltd., 2008.

SALTELLI, A. et al. Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. **Computer Physics Communications**, v. 181, n. 2, p. 259-270, fev. 2010.

SGROTT, D. **Análise computacional (CFD) em hélice danificado por cavitação e bioincrustação**. 2021. Trabalho de Conclusão de Curso (Graduação em Engenharia Naval) – Centro Tecnológico de Joinville, Universidade Federal de Santa Catarina, Joinville, 2021.

SHIELDS, M; ZHANG, J. The generalization of Latin hypercube sampling. **Reliability Engineering and System Safety**, v. 148, p. 96-108, abr. 2016.

SILVA, V. **Análise CFD de um hélice da série b com diferentes ângulos de caimento**. 2016. Trabalho de Conclusão de Curso (Graduação em Engenharia Naval) – Centro Tecnológico de Joinville, Universidade Federal de Santa Catarina, Joinville, 2016.

SMITH, R. C. **Uncertainty quantification: theory, implementation, and applications**. Raleigh: SIAM, 2013.

SOBOL, I. Global sensitivity indices for the investigation of nonlinear mathematical models. **Mathematics and Computers in Simulation**, v. 55, p. 271-280, 2001.

SON, J.; DU, Y. Comparison of intrusive and nonintrusive polynomial chaos expansion-based approaches for high dimensional parametric uncertainty quantification and propagation. **Computers & Chemical Engineering**, v. 134, mar. 2020.

SULLIVAN, T. J. **Introduction to uncertainty quantification**. Coventry: Springer, 2015.

TASKAR, B. et al. The effect of waves on engine-propeller dynamics and propulsion performance of ships. **Ocean Engineering**, v. 122, p. 262-277, ago. 2016.

THE AMERICAN SOCIETY OF MECHANICAL ENGINEERS. **ASME V&V 20-2009**. Standard for verification and validation in computational fluid dynamics and heat transfer. Nova Iorque: ASME, 2009.

U.S. DEPARTMENT OF ENERGY. Uncertainty quantification and error analysis. **Scientific grand challenges for national security: the role of computing at the extreme scale**, p. 121-159, out. 2009.

WALTERS, K. D.; COKLJAT, D. A three-equation eddy-viscosity model for Reynolds-Averaged Navier–Stokes simulations of transitional flow. **Journal of Fluids Engineering**, v. 130, n. 12, p. 100-110, dez. 2008.

WÄRTSILÄ. **Controllable pitch propeller systems**. Disponível em: <https://www.wartsila.com/marine/build/propulsors-and-gears/propellers/wartsila-controllable-pitch-propeller-systems>. Acesso em: 02 mar. 2022.

WIELGOSZ, C. et al. Numerical and experimental study on the Duisburg Propeller Test Case. **NuTTS 2019**, Tomar, PT, out. 2019. Disponível em: [https://www.researchgate.net/publication/336564093\\_Numerical\\_and\\_experimental\\_study\\_on\\_the\\_Duisburg\\_Propeller\\_Test\\_Case](https://www.researchgate.net/publication/336564093_Numerical_and_experimental_study_on_the_Duisburg_Propeller_Test_Case). Acesso em: 07 nov. 2021.

## APÊNDICE A – Arquivo controls.xml

```

<?xml version="1.0"?>
-<controls>
-<general>
<codeVersion>2.8</codeVersion>
<name>DTCprop</name>
<description>DTCprop</description>
<caseid>DTCprop</caseid>
<material>WATER</material>
<referenceLength>0.150</referenceLength>
<referenceVelocity>3.625</referenceVelocity>
<referenceMaterial>WATER</referenceMaterial>
<referencePressure>0.0</referencePressure>
<outFileName>out</outFileName>
<outFilePath>data</outFilePath>
<nsave>-1</nsave>
<readCgnsDirectly>>true</readCgnsDirectly>
</general>
-<grids>
-<grid>
<gridFileName>grid_f0325</gridFileName>
<gridFilePath>../grids_refresco</gridFilePath>
-<subGrids>
-<subGrid name="Interior_1">
<moveGridApply>>false</moveGridApply>
<moveGridMethod>AFM</moveGridMethod>
</subGrid>
-<subGrid name="Interior_2">
<moveGridApply>>false</moveGridApply>
<moveGridMethod>AFM</moveGridMethod>
</subGrid>
</subGrids>

```

```

</grid>
</grids>
-<outerLoop>
<maxIteration>20000</maxIteration>
<convergenceToleranceLinf>1e-6</convergenceToleranceLinf>
<divergenceTolerance>1e+9</divergenceTolerance>
</outerLoop>
-<bodies>
-<body name="prop">
<families>BCWall_2 BCWallViscous_2 BCWallViscousIsothermal_2
BCWallViscousHeatFlux_2 BCWallInviscid_2 BCTunnelInflow_2</families>
<parentBody/>
<centerOfMass userCode="false">0 0 0</centerOfMass>
-<motions>
-<motion name="rotation">
-<IMPOSED>
<type>ROTATION</type>
<method>AFM</method>
<magnitude>62.83185307</magnitude>
<period>0.0</period>
<axis>-1.0 0.0 0.0</axis>
<origin>0.0 0.0 0.0</origin>
<referenceSystem>EARTH_FIXED</referenceSystem>
</IMPOSED>
</motion>
</motions>
</body>
</bodies>
-<boundaries>
-<family name="inlet_1">
-<BCInflow>
<velocity>3.625 0 0</velocity>
<referenceSystem>EARTH_FIXED</referenceSystem>
-<turbulence>

```

```

-<turbIntensity_eddyVisc>
<turbIntensity>2e-2</turbIntensity>
<eddyVisc>1.0</eddyVisc>
</turbIntensity_eddyVisc>
</turbulence>
</BCInflow>
</family>
-<family name="outer_1">
<BCSlipWall/>
</family>
-<family name="outlet_inner_1">
<BCOutflow/>
</family>
-<family name="outlet_1">
-<BCPressure>
<pressure>0.0</pressure>
</BCPressure>
</family>
-<family name="BCOutflow_2">
-<BCInterface>
-partnerName>interface_outlet_1</partnerName>
<interfaceType>STATIC</interfaceType>
<!-- NOTE these should all be static unless grid deformation is applied -->
<haloType>HALO_1</haloType>
-<interpolation>
<polyDegree>1</polyDegree>
<INVERSE_DISTANCE/>
</interpolation>
</BCInterface>
</family>
-<family name="BCInflow_2">
-<BCInterface>
<partnerName>interface_inlet_1</partnerName>
<interfaceType>STATIC</interfaceType>

```

```
<haloType>HALO_1</haloType>
-<interpolation>
<polyDegree>1</polyDegree>
<INVERSE_DISTANCE/>
</interpolation>
</BCInterface>
</family>
-<family name="BCFarfield_2">
-<BCInterface>
<partnerName>interface_outer_1</partnerName>
<interfaceType>STATIC</interfaceType>
<haloType>HALO_1</haloType>
-<interpolation>
<polyDegree>1</polyDegree>
<INVERSE_DISTANCE/>
</interpolation>
</BCInterface>
</family>
-<family name="interface_outlet_1">
-<BCInterface>
<partnerName>BCOutflow_2</partnerName>
<interfaceType>STATIC</interfaceType>
<haloType>HALO_1</haloType>
-<interpolation>
<polyDegree>1</polyDegree>
<INVERSE_DISTANCE/>
</interpolation>
</BCInterface>
</family>
-<family name="interface_inlet_1">
-<BCInterface>
<partnerName>BCInflow_2</partnerName>
<interfaceType>STATIC</interfaceType>
<haloType>HALO_1</haloType>
```

```

-<interpolation>
<polyDegree>1</polyDegree>
<INVERSE_DISTANCE/>
</interpolation>
</BCInterface>
</family>
-<family name="interface_outer_1">
-<BCInterface>
<partnerName>BCFarfield_2</partnerName>
<interfaceType>STATIC</interfaceType>
<haloType>HALO_1</haloType>
-<interpolation>
<polyDegree>1</polyDegree>
<INVERSE_DISTANCE/>
</interpolation>
</BCInterface>
</family>
-<family name="shaft_upstream_1">
-<BCWall>
<referenceSystem>EARTH_FIXED</referenceSystem>
<useWallFunction>NO</useWallFunction>
</BCWall>
</family>
-<family name="BCWall_2">
-<BCWall>
<referenceSystem>BODY_FIXED</referenceSystem>
<useWallFunction>NO</useWallFunction>
<bodyFixedReferenceFamily>BCWall_2</bodyFixedReferenceFamily>
<geometryFileName>../geometry_stl/geom_BCWall_20.stl_part_1_1</geometryFileN
ame>
<projection>>true</projection>
</BCWall>
</family>
-<family name="BCWallViscous_2">

```

```

-<BCWall>
<referenceSystem>BODY_FIXED</referenceSystem>
<useWallFunction>NO</useWallFunction>
<bodyFixedReferenceFamily>BCWall_2</bodyFixedReferenceFamily>
<geometryFileName>../geometry_stl/geom_BCWallViscous_20.stl_part__mod1_1</g
eometryFileName>
<projection>>true</projection>
</BCWall>
</family>
-<family name="BCWallViscousIsothermal_2">
-<BCWall>
<referenceSystem>BODY_FIXED</referenceSystem>
<useWallFunction>NO</useWallFunction>
<bodyFixedReferenceFamily>BCWall_2</bodyFixedReferenceFamily>
<geometryFileName>../geometry_stl/geom_BCWallViscousIsothermal_20.stl_part__
mod3_1</geometryFileName>
<projection>>true</projection>
</BCWall>
</family>
-<family name="BCWallInviscid_2">
-<BCWall>
<referenceSystem>BODY_FIXED</referenceSystem>
<useWallFunction>NO</useWallFunction>
<bodyFixedReferenceFamily>BCWall_2</bodyFixedReferenceFamily>
<geometryFileName>../geometry_stl/geom_BCWallInviscid_20.stl_part__mod2_1</g
eometryFileName>
<projection>>true</projection>
</BCWall>
</family>
-<family name="BCWallViscousHeatFlux_2">
-<BCWall>
<referenceSystem>BODY_FIXED</referenceSystem>
<useWallFunction>NO</useWallFunction>
<bodyFixedReferenceFamily>BCWall_2</bodyFixedReferenceFamily>

```

```

<geometryFileName>../geometry_stl/geom_BCWallViscousHeatFlux_20.stl_part__m
od4_1</geometryFileName>
<projection>>true</projection>
</BCWall>
</family>
-<family name="BCTunnelInflow_2">
-<BCWall>
<referenceSystem>BODY_FIXED</referenceSystem>
<useWallFunction>NO</useWallFunction>
<bodyFixedReferenceFamily>BCWall_2</bodyFixedReferenceFamily>
</BCWall>
</family>
</boundaries>
-<massMomentumSolver>
-<solverType name="SEGREGATED">
-<segregated>
<solver>FRESCO</solver>
</segregated>
</solverType>
</massMomentumSolver>
-<equations>
-<equation name="mom">
-<EQMomentum>
<solve_x>true</solve_x>
<solve_y>true</solve_y>
<solve_z>true</solve_z>
-<solver>
-<PETSC>
<solver>GMRES</solver>
<preconditioner>JACOBI</preconditioner>
</PETSC>
</solver>
<convergenceTolerance>1e-2</convergenceTolerance>
<maxIteration>200</maxIteration>

```

```

-<relaxationProcedure>
-<IMPLICIT_EXPLICIT>
<imp_relax_min>0.40</imp_relax_min>
<imp_relax_max>0.90</imp_relax_max>
<imp_relax_fac>200</imp_relax_fac>
<exp_relax>0.2</exp_relax>
</IMPLICIT_EXPLICIT>
</relaxationProcedure>
-<convectiveFluxDiscretisation>
-<TVD_SCHEME>
<schemeName>LIMITED_QUICK</schemeName>
</TVD_SCHEME>
</convectiveFluxDiscretisation>
-<gradientCalculation>
<GAUSS/>
</gradientCalculation>
<applyEccentricityCorrection>true</applyEccentricityCorrection>
<userSource>>false</userSource>
<saveResidual>true</saveResidual>
<saveChanges>>false</saveChanges>
<saveGradLimiter>>false</saveGradLimiter>
<saveCourantNo>>false</saveCourantNo>
<savePecletNo>>false</savePecletNo>
<saveSkinFriction>true</saveSkinFriction>
<saveYplus>true</saveYplus>
-<initialization>
-<FAMILY_BASED>
<familyName>inlet_1</familyName>
</FAMILY_BASED>
</initialization>
</EQMomentum>
</equation>
-<equation name="pres">
-<EQPressure>

```

```

-<solver>
-<PETSC>
<solver>CG</solver>
<preconditioner>BJACOBI</preconditioner>
</PETSC>
</solver>
<convergenceTolerance>1e-2</convergenceTolerance>
<maxIteration>500</maxIteration>
-<relaxationProcedure>
-<EXPLICIT>
<exp_relax>0.1</exp_relax>
</EXPLICIT>
</relaxationProcedure>
-<gradientCalculation>
<GAUSS/>
</gradientCalculation>
<applyEccentricityCorrection>true</applyEccentricityCorrection>
<userSource>>false</userSource>
<saveResidual>true</saveResidual>
<saveGradLimiter>>false</saveGradLimiter>
<saveHessian>>false</saveHessian>
<saveChanges>>false</saveChanges>
<initialPressure userCode="false">0.0</initialPressure>
</EQPressure>
</equation>
-<equation name="turb">
-<EQTurbulence>
-<turbulenceModel>
-<K_SKL>
<MODELTYPE>STANDARD</MODELTYPE>
</K_SKL>
<!-- <K_OMEGA><MODELTYPE>SST_2003</MODELTYPE></K_OMEGA> -->
</turbulenceModel>
-<solver>

```

```

-<PETSC>
<solver>GMRES</solver>
<preconditioner>BJACOBI</preconditioner>
</PETSC>
</solver>
<convergenceTolerance>5e-2</convergenceTolerance>
<maxIteration>200</maxIteration>
-<relaxationProcedure>
-<IMPLICIT_EXPLICIT>
<imp_relax_min>0.40</imp_relax_min>
<imp_relax_max>0.90</imp_relax_max>
<imp_relax_fac>200</imp_relax_fac>
<exp_relax>0.2</exp_relax>
</IMPLICIT_EXPLICIT>
</relaxationProcedure>
-<convectiveFluxDiscretisation>
-<FOU_CDS_BLEND>
<blendingFactor>0.0</blendingFactor>
</FOU_CDS_BLEND>
</convectiveFluxDiscretisation>
-<gradientCalculation>
<GAUSS/>
</gradientCalculation>
<applyEccentricityCorrection>true</applyEccentricityCorrection>
<saveTurbulenceIntensity>true</saveTurbulenceIntensity>
<userSource>false</userSource>
<saveResidual>true</saveResidual>
<saveChanges>false</saveChanges>
<saveGradLimiter>false</saveGradLimiter>
-<initialization>
-<turbIntensity_eddyVisc>
<turbIntensity>2e-2</turbIntensity>
<eddyVisc>1.0</eddyVisc>
</turbIntensity_eddyVisc>

```

```

</initialization>
-<decayControl>
-<XFREEZE>
<x2freezeTurb>-0.25</x2freezeTurb>
</XFREEZE>
</decayControl>
</EQTurbulence>
</equation>
<!--equation
name="trans"><EQTransition><transitionModel><LCTM><MODELTYPE>GAMMA_
RETHETA</MODELTYPE></LCTM></transitionModel><solver><PETSC><solver>G
MRES</solver><preconditioner>BJACOBI</preconditioner></PETSC></solver><con
vergenceTolerance>0.01</convergenceTolerance><maxIteration>200</maxIteration
><relaxationProcedure><IMPLICIT_EXPLICIT><imp_relax_min>0.25</imp_relax_mi
n><imp_relax_max>0.95</imp_relax_max><imp_relax_fac>100</imp_relax_fac><ex
p_relax>0.25</exp_relax></IMPLICIT_EXPLICIT></relaxationProcedure><convectiv
eFluxDiscretisation><FOU_CDS_BLEND><blendingFactor>0.0</blendingFactor></F
OU_CDS_BLEND></convectiveFluxDiscretisation><gradientCalculation><GAUSS/>
</gradientCalculation><userSource>>false</userSource><applyEccentricityCorrec
tion>>false</applyEccentricityCorrec
tion><toleranceApplyGamma>10</toleranceApplyGamma><relaxGamma>0.1</relaxGamma><saveResidual>>true</saveResidual><save
GradLimiter>>false</saveGradLimiter><saveFluxLimiter>>false</saveFluxLimiter><sav
eChanges>>false</saveChanges></EQTransition></equation-->
</equations>
-<extraQuantities>
<saveVorticity>>true</saveVorticity>
<saveQCriterion>>true</saveQCriterion>
<saveQ>>true</saveQ>
<saveReynoldsStress>>true</saveReynoldsStress>
<saveRefinementLevel>>true</saveRefinementLevel>
</extraQuantities>
-<coprocessing>
-<setup>
<outerIterations>10</outerIterations>

```

```

<coprocFile>../../coprocessing/pipeline.py</coprocFile>
<grid>>true</grid>
<ghostCellsApply>>true</ghostCellsApply>
<!--families>BCWall_2 BCWallViscous_2 BCWallViscousIsothermal_2
BCWallViscousHeatFlux_2 BCWallInviscid_2 BCTunnelInflow_2</families-->
<families>BCWall_2</families>
<fields>Pressure SkinFriction</fields>
</setup>
</coprocessing>
-<materials>
-<material name="WATER">
-<fluid>
<viscosityMolecular>1.111e-3</viscosityMolecular>
<density>998.10</density>
</fluid>
</material>
</materials>
<developer> </developer>
-<monitors>
-<monitor name="forces_blades">
-<MO_Force>
<fileName>data/forces_blades</fileName>
<saveFrequency>1</saveFrequency>
<families>BCWall_2 BCWallViscous_2 BCWallViscousIsothermal_2
BCWallViscousHeatFlux_2 BCWallInviscid_2</families>
<forTecplot>>true</forTecplot>
</MO_Force>
</monitor>
-<monitor name="forces_hub">
-<MO_Force>
<fileName>data/forces_hub</fileName>
<saveFrequency>1</saveFrequency>
<families>BCTunnelInflow_2</families>
<forTecplot>>true</forTecplot>

```

```

</MO_Force>
</monitor>
-<monitor name="moments_blades">
-<MO_Moment>
<fileName>data/moments_blades</fileName>
<saveFrequency>1</saveFrequency>
<families>BCWall_2 BCWallViscous_2 BCWallViscousIsothermal_2
BCWallViscousHeatFlux_2 BCWallInviscid_2</families>
<referencePoint>0 0 0</referencePoint>
<forTecplot>true</forTecplot>
<referenceSystem>EARTH_FIXED</referenceSystem>
</MO_Moment>
</monitor>
-<monitor name="moments_hub">
-<MO_Moment>
<fileName>data/moments_hub</fileName>
<saveFrequency>1</saveFrequency>
<families>BCTunnelInflow_2</families>
<referencePoint>0 0 0</referencePoint>
<forTecplot>true</forTecplot>
<referenceSystem>EARTH_FIXED</referenceSystem>
</MO_Moment>
</monitor>
-<monitor name="yPlus">
-<MO_Yplus>
<fileName>data/yPlus</fileName>
<saveFrequency>1</saveFrequency>
<families>BCTunnelInflow_2 BCWall_2 BCWallViscous_2
BCWallViscousIsothermal_2 BCWallViscousHeatFlux_2 BCWallInviscid_2</families>
<forTecplot>true</forTecplot>
</MO_Yplus>
</monitor>
-<monitor name="Global">
-<MO_Global>

```

```

<fileName>./data/scalar_nIntCells</fileName>
<saveFrequency>1</saveFrequency>
<label/>
<fields>nIntCells</fields>
<forTecplot>true</forTecplot>
</MO_Global>
</monitor>
-<monitor name="PressureStats">
-<MO_Scalar>
<fileName>./data/scalar_pressure</fileName>
<saveFrequency>1</saveFrequency>
<label/>
<fields>Pressure</fields>
<forTecplot>true</forTecplot>
</MO_Scalar>
</monitor>
<!-- <monitor name="CourantNumber"> For
unsteady<MO_Scalar><fileName>./data/scalar_CourantNumber</fileName><saveFr
equency>1</saveFrequency><label></label><fields>CourantNo</fields><forTecplot>
true</forTecplot></MO_Scalar></monitor> -->
</monitors>
<!--adaptiveGrid><adaptiveGridSetup
name="general"><general><adaptivity>true</adaptivity><isotropic>>false</isotropic><
projection>true</projection><maxAdapt>100</maxAdapt><nInitialSteps>0</nInitialSt
eps><adaptFrequency>500</adaptFrequency><minVolume>1e-
24</minVolume><maxAspectRatio>50</maxAspectRatio><coarseningFactor>-
1</coarseningFactor></general></adaptiveGridSetup><adaptiveGridSetup
name="boxWake"><box><active>>false</active><behaviour>OR</behaviour><centre
>0 0 0</centre><dimensions>0.3 0.3
0.3</dimensions><maxRefinementLevel>3</maxRefinementLevel><splitDirection>1
1
1</splitDirection><referenceSystem>EARTH_FIXED</referenceSystem></box></ad
aptiveGridSetup><adaptiveGridSetup
name="IsotropicVorticity"><IsotropicCriteria><refineTarget>1e24</refineTarget><ma

```

```

xRefinementLevel>1</maxRefinementLevel><save>true</save><fields>Vorticity</fields><geometricalConstraints>boxWake</geometricalConstraints></IsotropicCriteria>
</adaptiveGridSetup><adaptiveGridSetup
name="DirectionalJumpQ"><DirectionalCriteria><refineTarget>1e24</refineTarget><maxRefinementLevel>1</maxRefinementLevel><save>true</save><method>Jump</method><fields>Q</fields><nSmooth>0</nSmooth><nStandardBufferLayers>2</nStandardBufferLayers><nFractionalBufferLayers>2</nFractionalBufferLayers><geometricalConstraints>boxWake</geometricalConstraints></DirectionalCriteria></adaptiveGridSetup><adaptiveGridSetup
name="hessianVelocity"><DirectionalCriteria><refineTarget>1e24</refineTarget><maxRefinementLevel>1</maxRefinementLevel><save>true</save><method>Hessian</method><useFieldsSeparately>false</useFieldsSeparately><fields>Velocity</fields><nSmooth>0</nSmooth><hessian_nSmoothGradients>4</hessian_nSmoothGradients><hessian_nSmooth>4</hessian_nSmooth><nStandardBufferLayers>2</nStandardBufferLayers><nFractionalBufferLayers>2</nFractionalBufferLayers><geometricalConstraints>boxWake</geometricalConstraints></DirectionalCriteria></adaptiveGridSetup><adaptiveGridSetup
name="DirectionalHessianPressure"><DirectionalCriteria><refineTarget>1e24</refineTarget><maxRefinementLevel>1</maxRefinementLevel><save>true</save><method>Hessian</method><useFieldsSeparately>false</useFieldsSeparately><fields>Pressure</fields><nSmooth>0</nSmooth><hessian_nSmoothGradients>4</hessian_nSmoothGradients><hessian_nSmooth>4</hessian_nSmooth><nStandardBufferLayers>2</nStandardBufferLayers><nFractionalBufferLayers>2</nFractionalBufferLayers><geometricalConstraints>boxWake</geometricalConstraints></DirectionalCriteria></adaptiveGridSetup><adaptiveGridSetup
name="DirectionalJumpPressure"><DirectionalCriteria><refineTarget>1e24</refineTarget><maxRefinementLevel>1</maxRefinementLevel><save>true</save><method>Jump</method><fields>Pressure</fields><geometricalConstraints>boxWake</geometricalConstraints></DirectionalCriteria></adaptiveGridSetup><adaptiveGridSetup
name="IsotropicPressure"><IsotropicCriteria><refineTarget>1e24</refineTarget><maxRefinementLevel>1</maxRefinementLevel><save>true</save><fields>Pressure</fields><geometricalConstraints>boxWake</geometricalConstraints></IsotropicCriteria></adaptiveGridSetup>
</adaptiveGrid-->

```

```
<!--deformGrid><deformGridSetup
name="general"><general><deformGridApply>true</deformGridApply><exactWallDistanceFrequencyInTimeLoop>10</exactWallDistanceFrequencyInTimeLoop><updateFrequency>1</updateFrequency></general></deformGridSetup><deformGridSetup
name="method1"><deformMethod_RBF><supportRadius>0.25</supportRadius><preconditioner>LU</preconditioner><reuseGreedyDataInOuterLoop>false</reuseGreedyDataInOuterLoop><resetGreedyFrequencyInTimeLoop>10</resetGreedyFrequencyInTimeLoop><exactNearWallCorrFrequencyInTimeLoop>10</exactNearWallCorrFrequencyInTimeLoop></deformMethod_RBF></deformGridSetup><deformGridSetup
name="method2"><deformMethod_IDW><interpolType>2</interpolType><rotation>false</rotation><powerExponent>3</powerExponent><powerExponent2>5</powerExponent2><eta>5</eta></deformMethod_IDW></deformGridSetup> </deformGrid-->
</controls>
```

## APÊNDICE B – cav\_indexCalcs.py

```

# -*- coding: utf-8 -*-
"""
Created on Mon Dec 27 19:58:13 2021
@author: atilio
"""

import numpy as np
import h5py
import re
import os

def getKeyword(key, controlFileString):
    """ Retrive a keyword from an xml document. """
    return re.findall('<{ }\s?name="]?.*>'.format(key, key),
controlFileString)[0].split(">")[1].split("<")[0]
D = 0.15
rho_exp = 998.16 # kg/m3
Va_exp = 3.625 # m/s
pv_exp_list = [0.022425*1e5, 0.0227125*1e5, 0.02300*1e5, 0.0232875*1e5,
0.023575*1e5] # Pa
cases_list = ['case_0', 'case_1', 'case_2', 'case_3', 'case_4']
# Example data point from the inception diagram.
p_inf_exp = (0.43240156 + 0.06) * 1e5 # Pa, at hub level i.e. excluding hydrostatic
# component (small for a small prop anyway)
for cases in cases_list:
    print("-----" + cases + "-----")
    if cases == "case_0": n_exp = 2474.06250/60
    if cases == "case_1": n_exp = 2505.78125/60
    if cases == "case_2": n_exp = 2537.50000/60
    if cases == "case_3": n_exp = 2569.21875/60
    if cases == "case_4": n_exp = 2600.93750/60
    for pv_exp in pv_exp_list:

```

```

J_exp = Va_exp / (D * n_exp)
sigma_exp = (p_inf_exp - pv_exp) / (0.5 * rho_exp * (n_exp*D)**2.)
print("J_exp = {:.4f}".format(J_exp))
print("sigma_exp = {:.4f}".format(sigma_exp))

case = os.getcwd() + "/" + cases + "/case_AFM"

with open(os.path.join(case, "controls.xml"), "r") as infile:
    controls = infile.read()

omega = float(getKeyword("magnitude", controls))
Va = float(getKeyword("referenceVelocity", controls))

n = omega / (2*np.pi)
J = Va / (n * D)
print("J_cfd = {:.4f}".format(J))

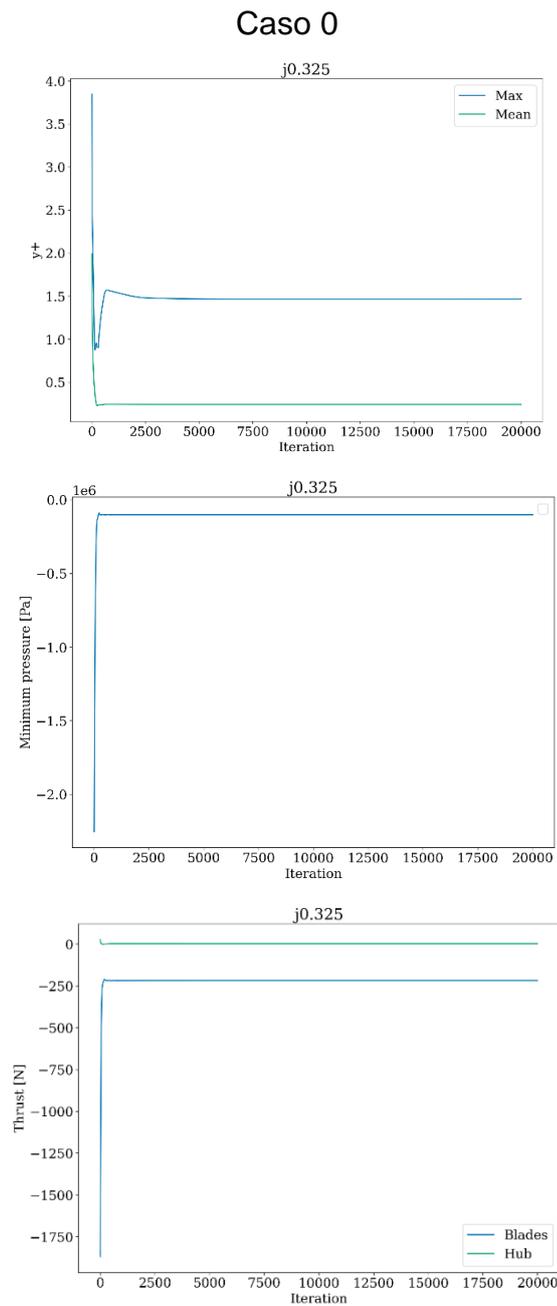
pv = - sigma_exp * (0.5*rho_exp*(n*D)**2)
print("pv_cfd = {:.1f} kPa".format(pv*1e-3))

# Read boundary pressures.
surfaceCgnsFile = h5py.File(os.path.join(case,
"data_coprocessing/boundary.cgns"), mode="r")
surfPressure =
surfaceCgnsFile["Base_Surface_Elements"]["BCWall_2"]["CellData"]["Pressure"]["
data"][:]
inception_index = (np.min(surfPressure) + pv) / (0.5*rho_exp*(n*D)**2)
inception_avg = (np.mean(surfPressure) + pv) / (0.5*rho_exp*(n*D)**2)
inception_max = (np.max(surfPressure) + pv) / (0.5*rho_exp*(n*D)**2)
print("Min: {:.4f}".format(inception_index))
print("Average: {:.4f}".format(inception_avg))
print("Max: {:.4f}".format(inception_max))
print("\n")

```

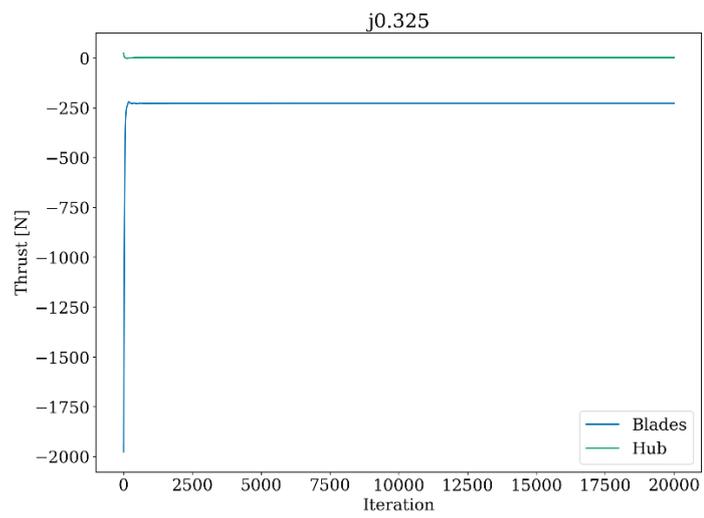
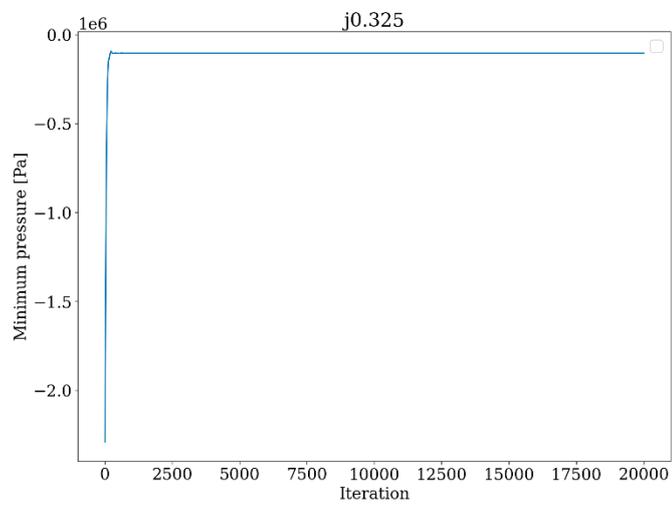
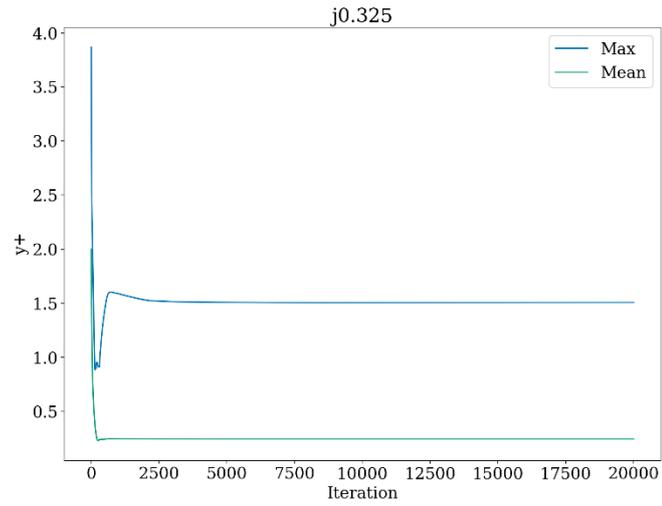
## APÊNDICE C – Convergência das simulações numéricas

São mostrados os gráficos de estabilização para as variáveis  $y^+$ , pressão mínima atuante na superfície do hélice e empuxo (T) para os casos 0, 1, 2, 3 e 4, para a malha  $f0.325$ , as outras malhas seguem a mesma tendência.



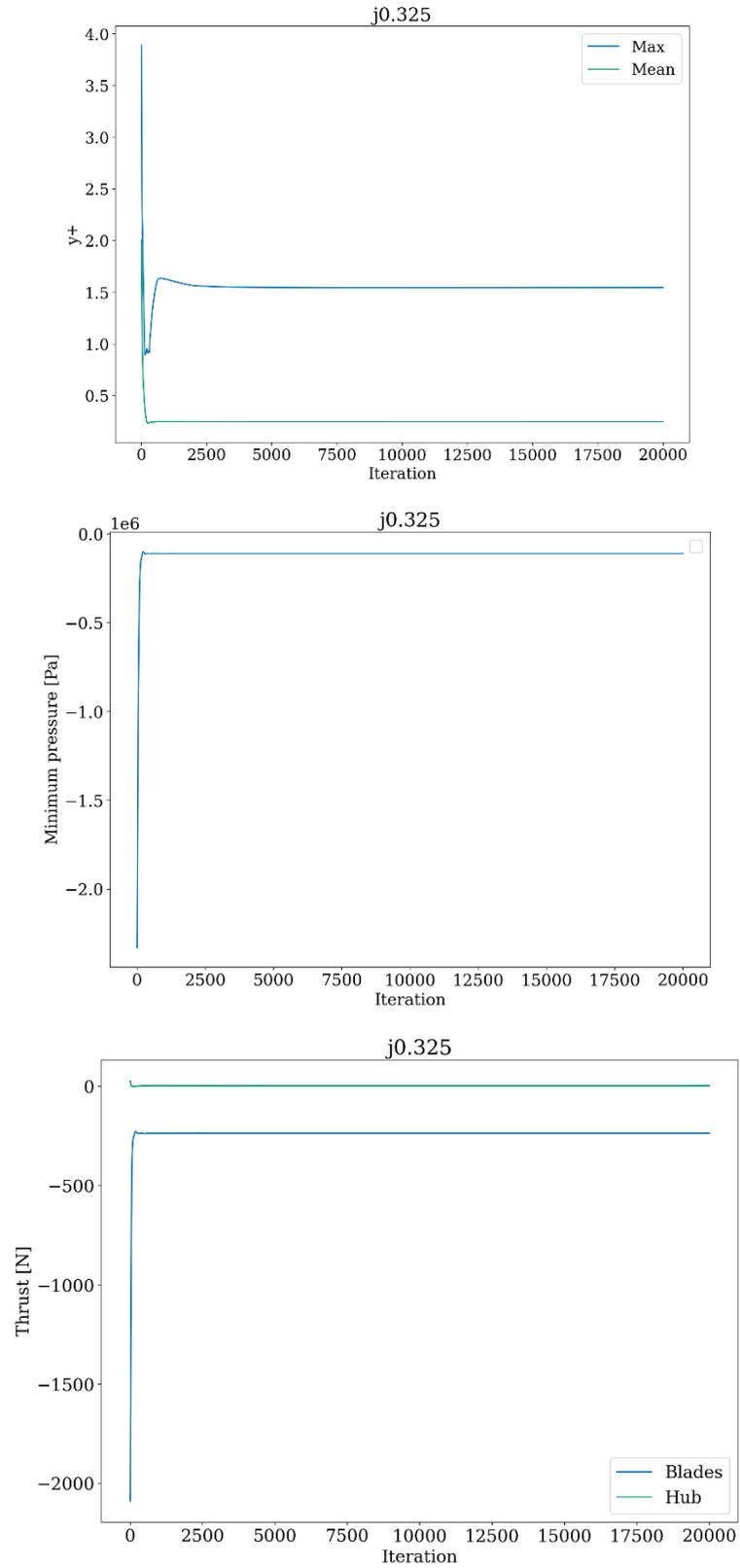
Fonte: autoria própria (2022).

### Caso 1



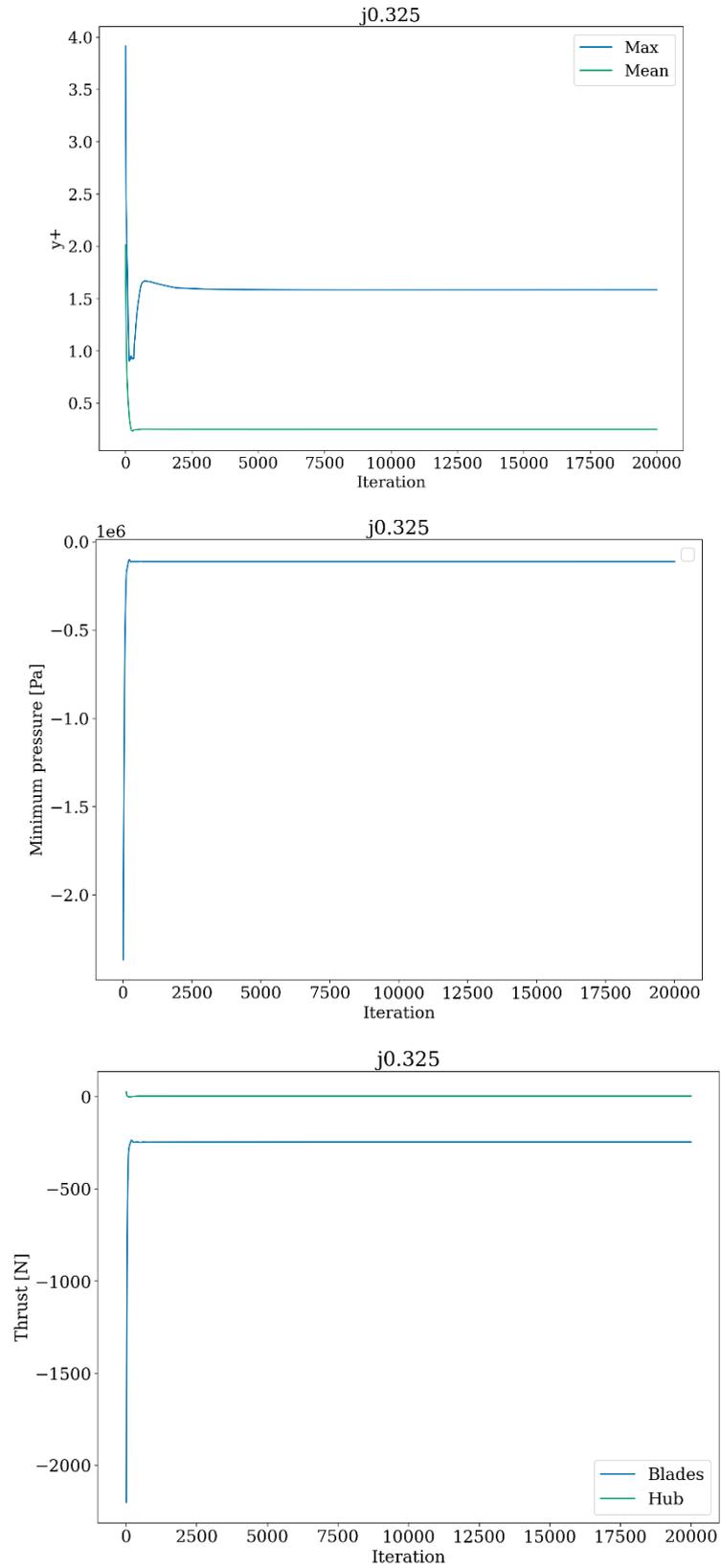
Fonte: autoria própria (2022).

## Caso 2



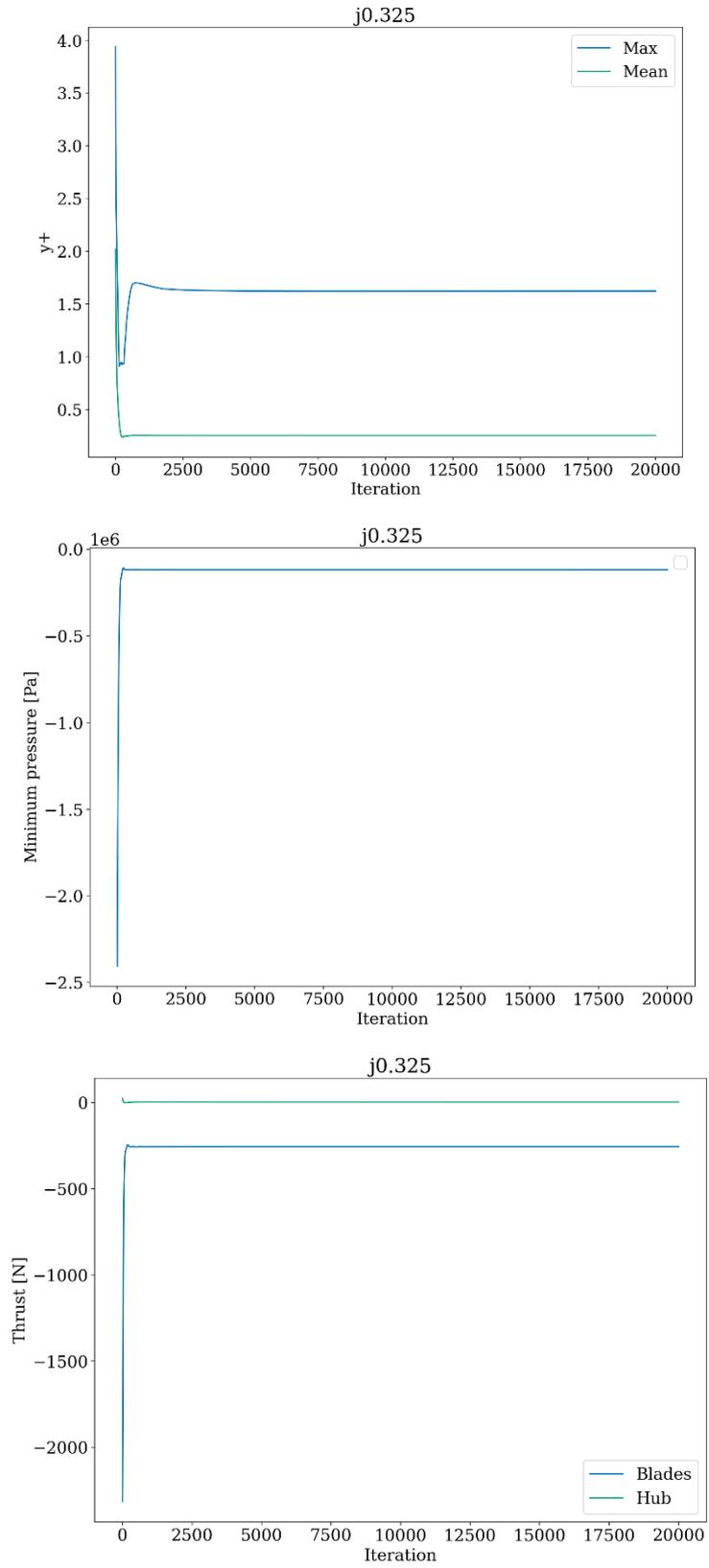
Fonte: autoria própria (2022).

## Caso 3



Fonte: autoria própria (2022).

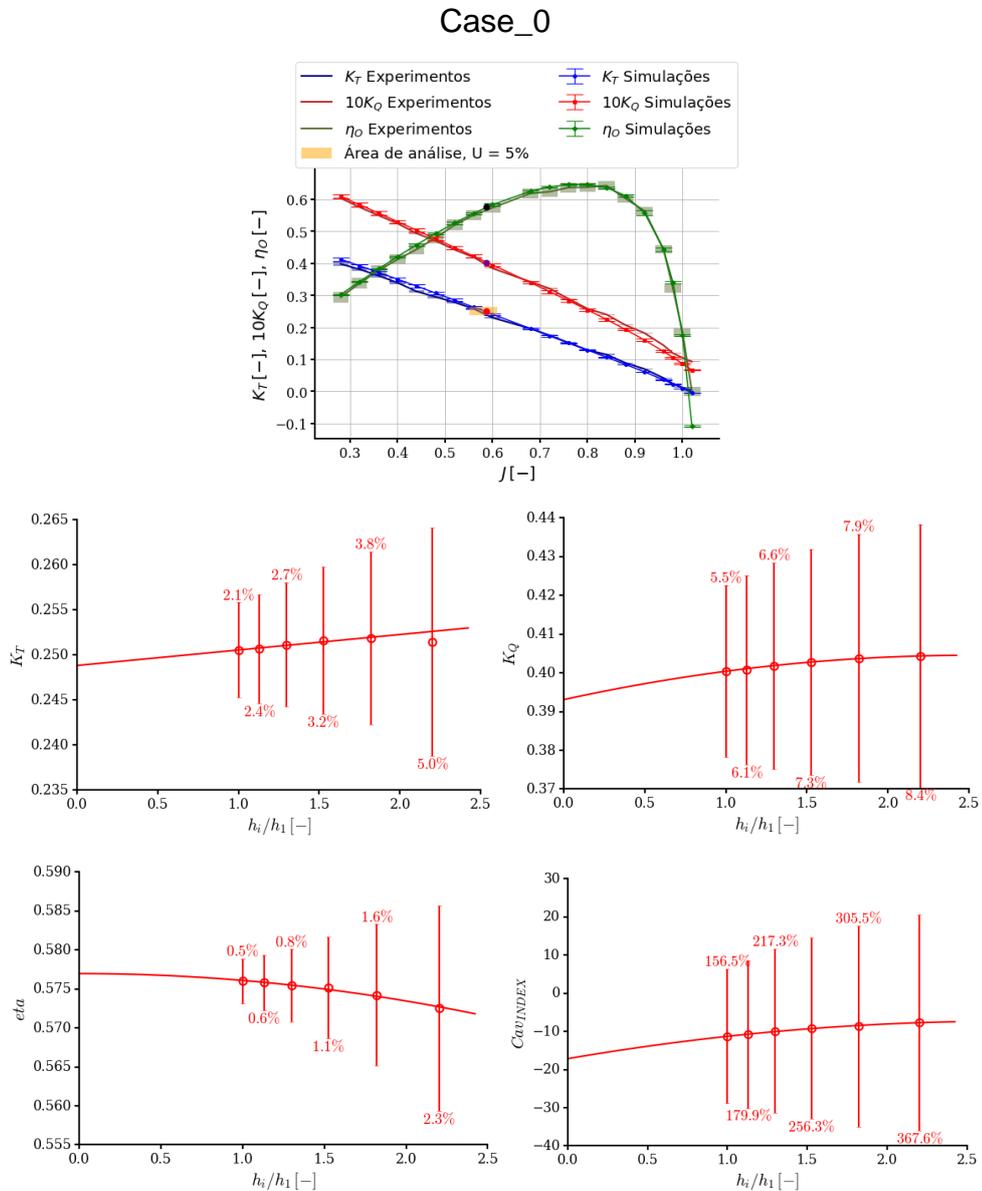
## Caso 4



Fonte: autoria própria (2022).

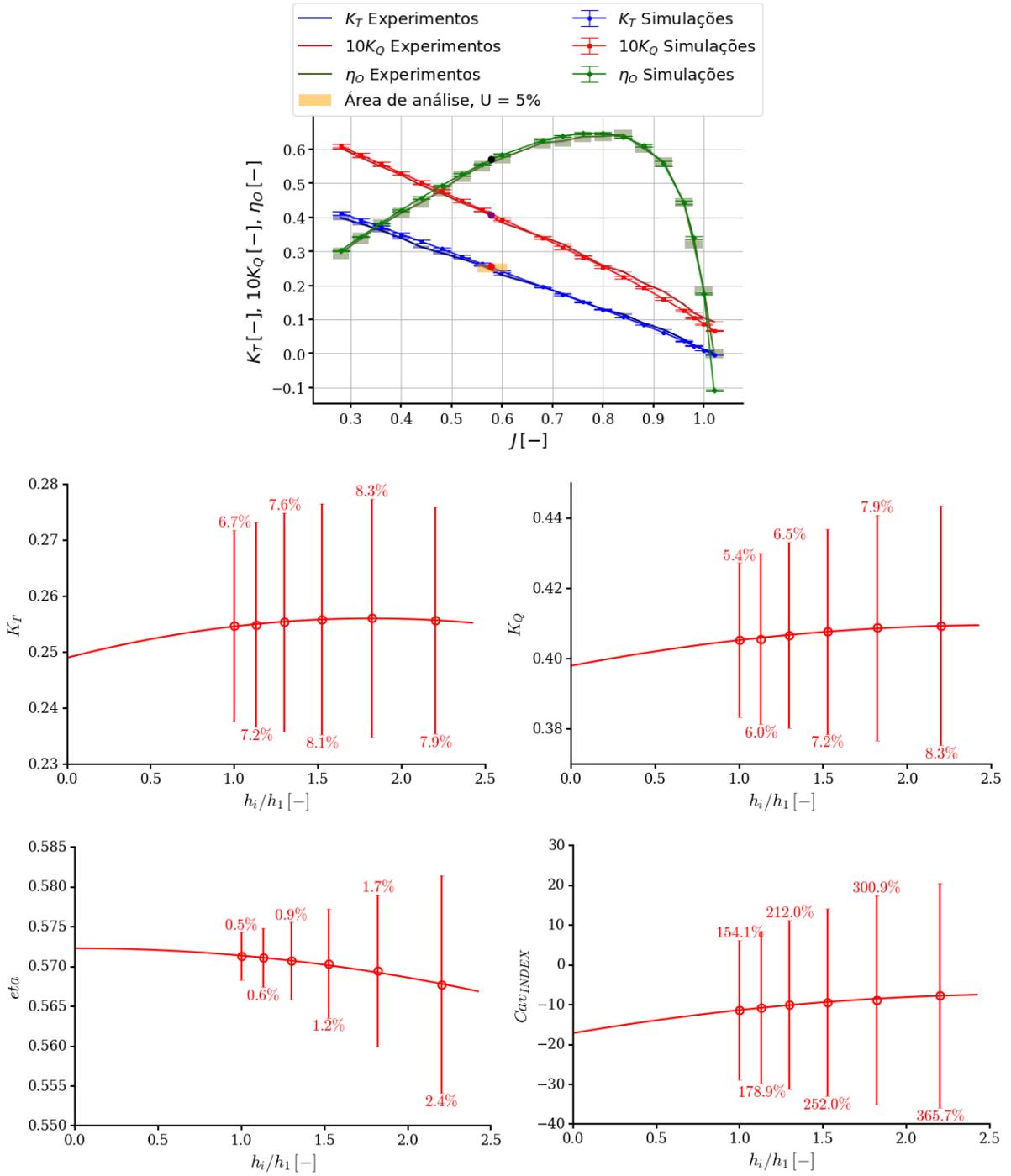
## APÊNDICE D – Incertezas de discretização no desempenho

Para os gráficos de incertezas aqui apresentados, o eixo y apresenta as incertezas referentes às variáveis analisadas e o eixo x apresenta a razão de malhas. Para os diagramas de águas abertas, as variáveis  $K_T$ ,  $K_Q$  e  $\eta$  apresentadas nas legendas se referem aos trabalhos de Golf (2018) e Wielgosz et al. (2019), enquanto os pontos apresentados neles (vermelho, roxo e preto) representam os valores de  $K_T$ ,  $K_Q$  e  $\eta$  obtidos com as simulações CFD realizadas.



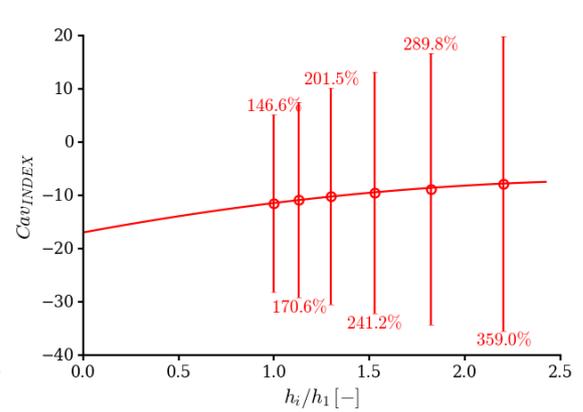
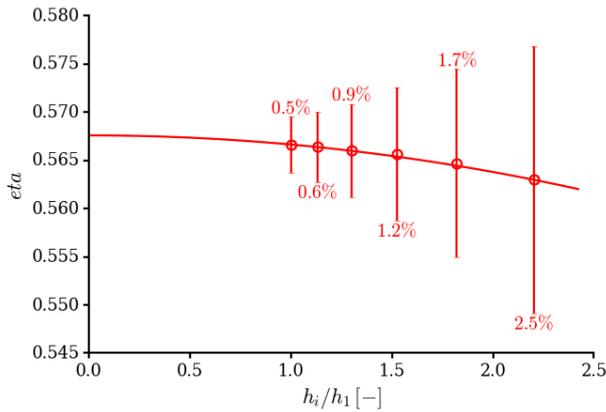
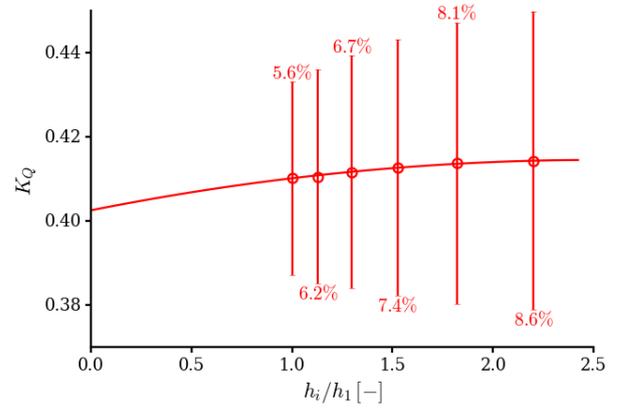
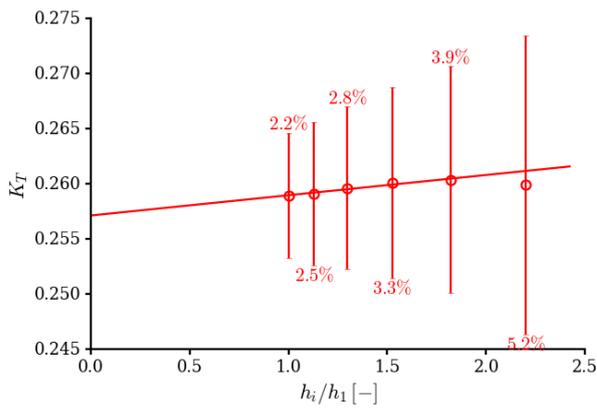
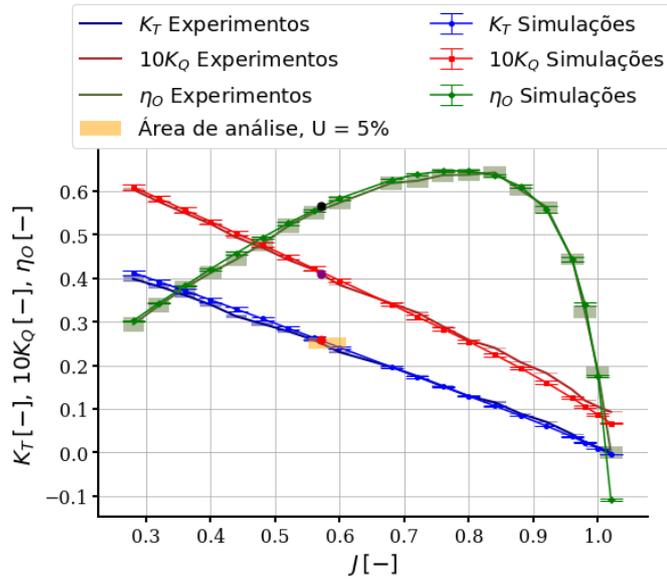
Fonte: autoria própria (2022).

Caso 1



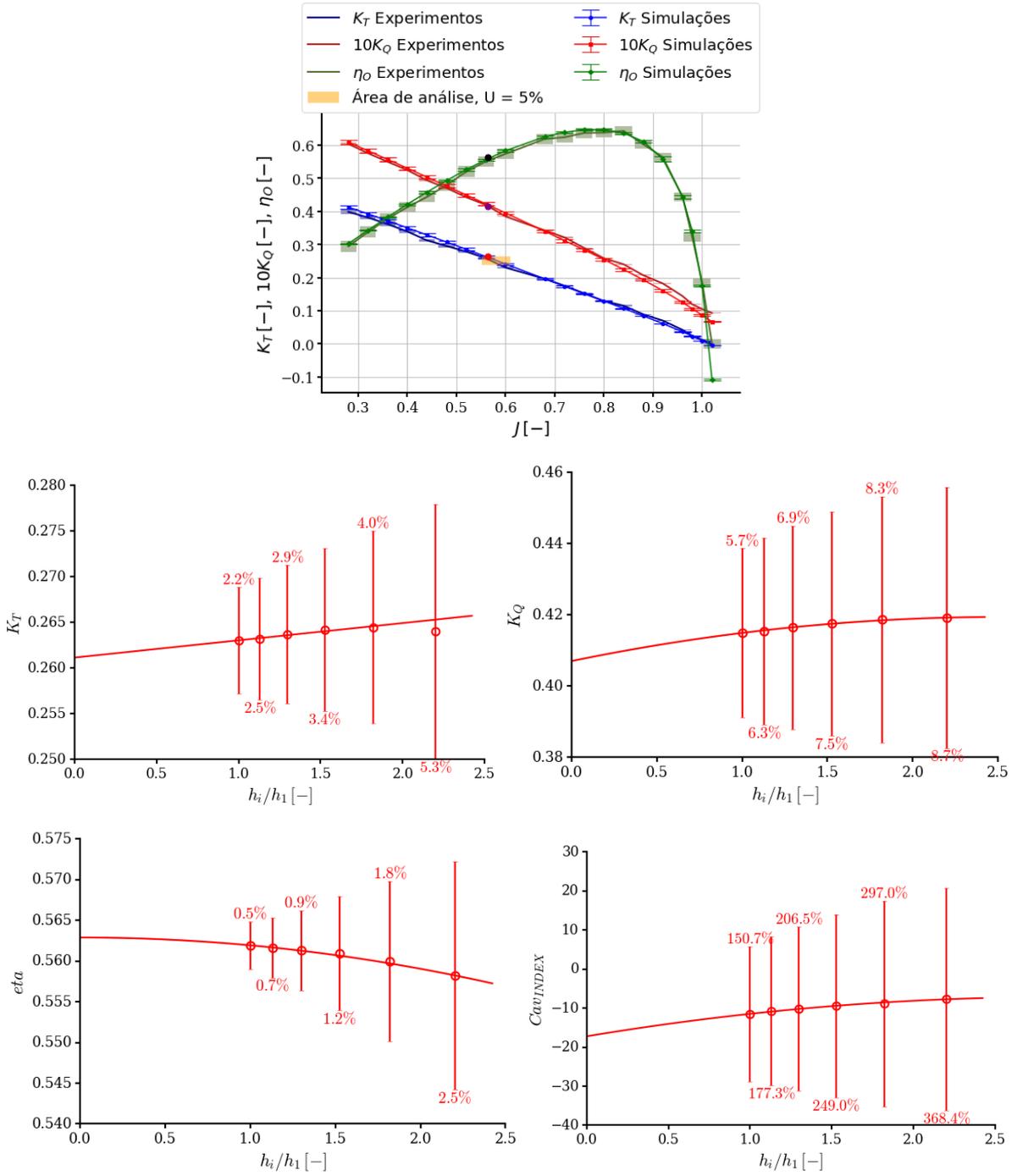
Fonte: autoria própria (2022).

Caso 2



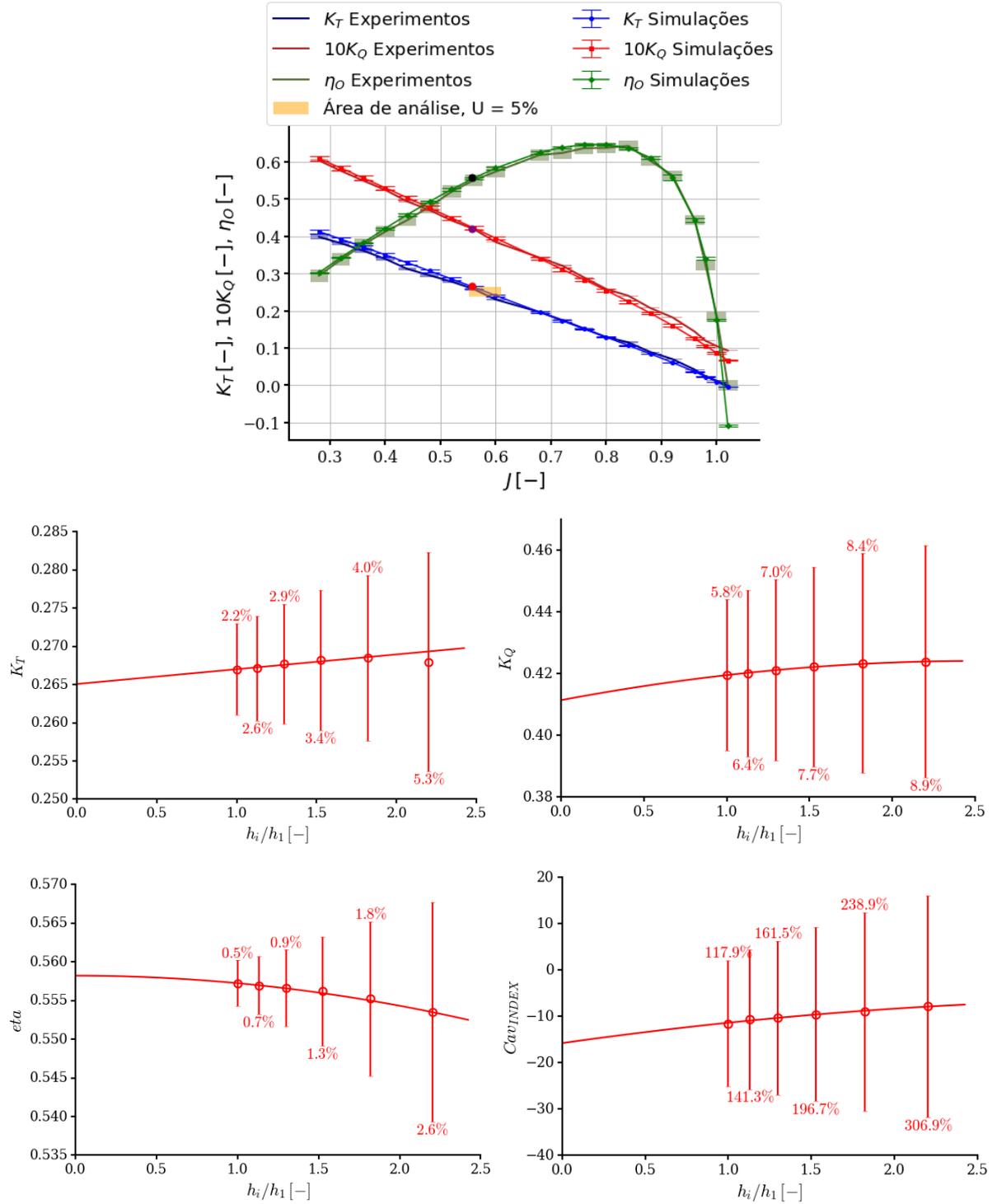
Fonte: autoria própria (2022).

Caso 3



Fonte: autoria própria (2022).

Caso 4



Fonte: autoria própria (2022).

## APÊNDICE E – Relação entre $\sigma$ e $C_{PN}$

Golf (2018) utilizou a Equação 15 para calcular  $\sigma$  e considerou  $P_\infty$  como mostrado pela Equação 16.

$$\sigma_{EXPERIMENTO} = \frac{P_\infty - P_{V,EXPERIMENTO}}{0,5 \times \rho \times (n \times D)^2} \quad (16)$$

$$P_\infty = P_{ATM} - P_{VÁCUO} + P_{HIDROSTÁTICA} \quad (17)$$

Para simulações executadas, a pressão de referência ( $P_\infty$ ) foi definida igual a 0 Pa. Então, com o objetivo de igualar as simulações CFD executadas com o experimento realizado por Golf (2018), calculou-se a  $P_{V,CFD}$  com base no  $\sigma$  estipulado (2,100). Para isso, foi utilizada a Equação 17.

$$P_{V,CFD} = -\sigma_{EXPERIMENTO} \times (0,5 \times \rho \times (n \times D)^2) \quad (18)$$

$$\therefore P_{V,CFD} = P_{V,EXPERIMENTO} - P_\infty$$

E, sabendo que para que ocorra cavitação  $-C_{PNmín} < \sigma$  (Equação 18).

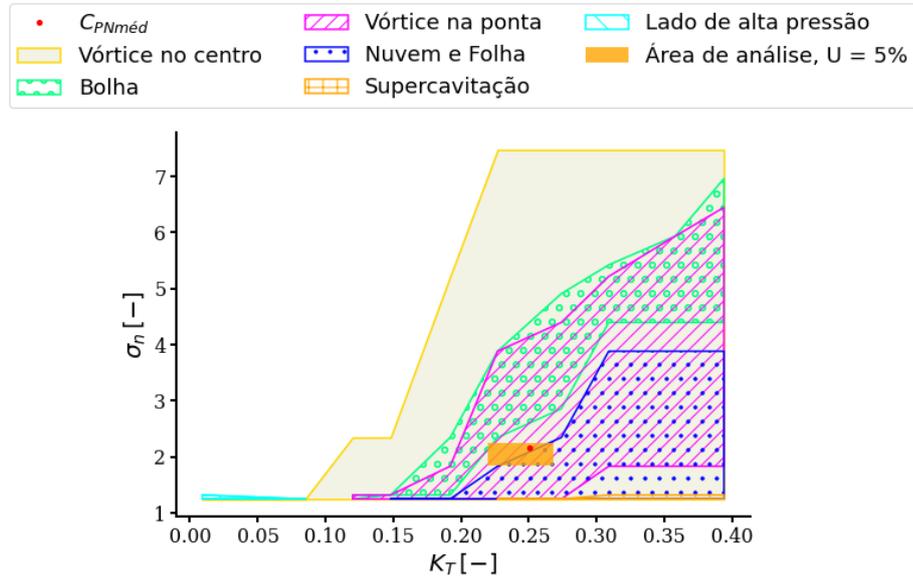
$$-C_{PNmín} < \sigma_{CFD} \therefore -P_L < P_{V,CFD} \quad (19)$$

$$\therefore P_L + P_{V,EXPERIMENTO} < 0$$

Esses cálculos foram realizados para compensar a pressão de referência utilizada nas simulações numéricas e obter o ponto de início de cavitação correto ( $-C_{PN} = \sigma$ ) para as simulações computacionais. Os cálculos visam igualar  $P_\infty - P_V$  das simulações CFD com e dos experimentos realizados por Golf (2018).

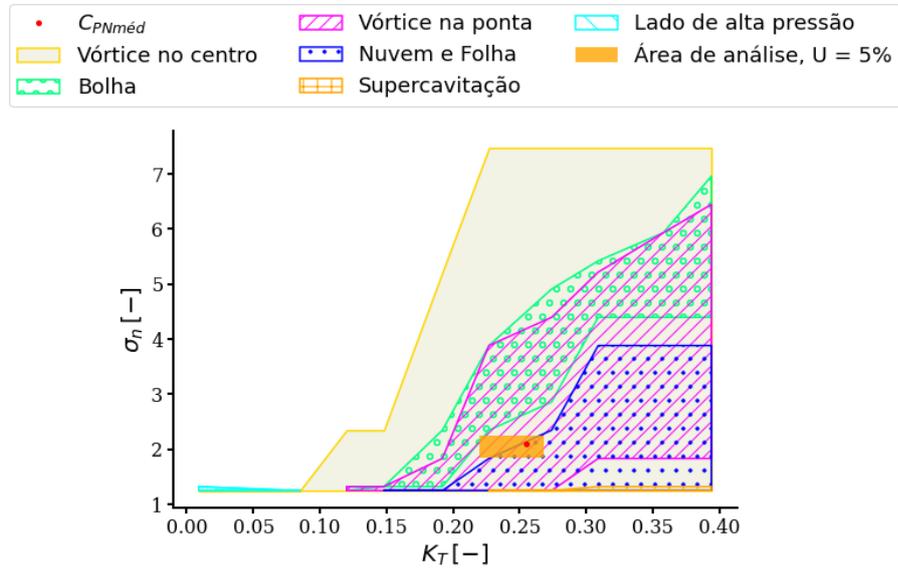
## APÊNDICE F – Diagrama de início de cavitação

### Caso 0



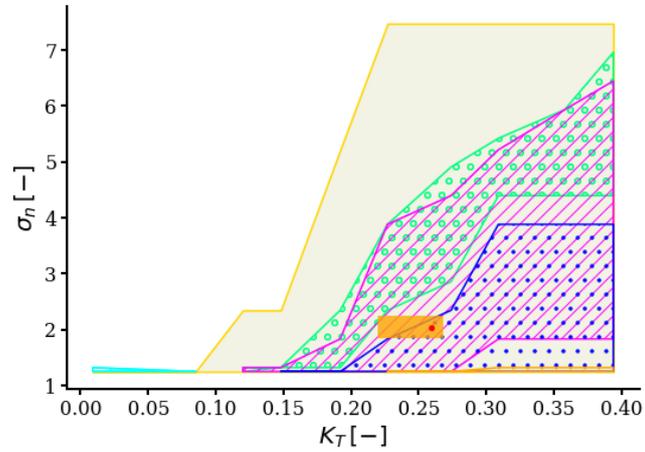
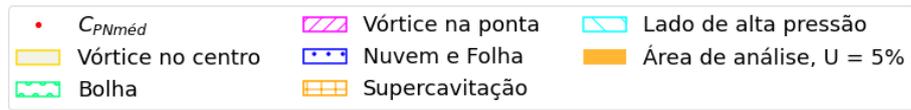
Fonte: adaptado de Golf (2018) e Wielgosz et al. (2019).

### Caso 1



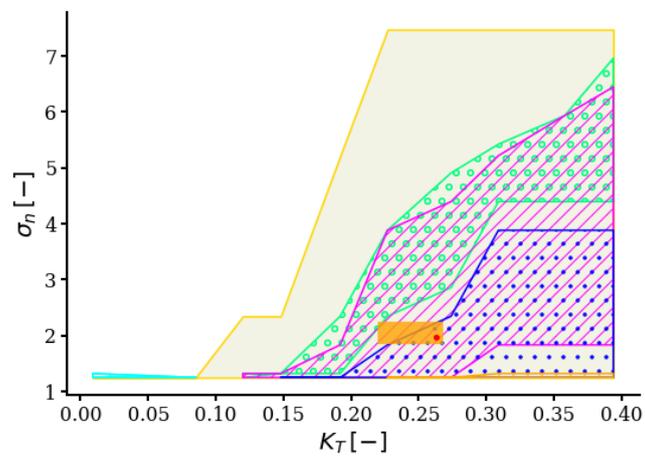
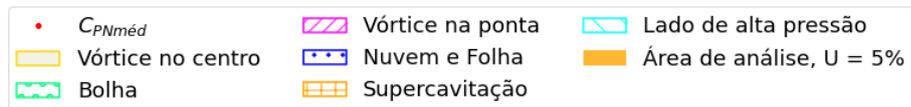
Fonte: adaptado de Golf (2018) e Wielgosz et al. (2019).

## Caso 2



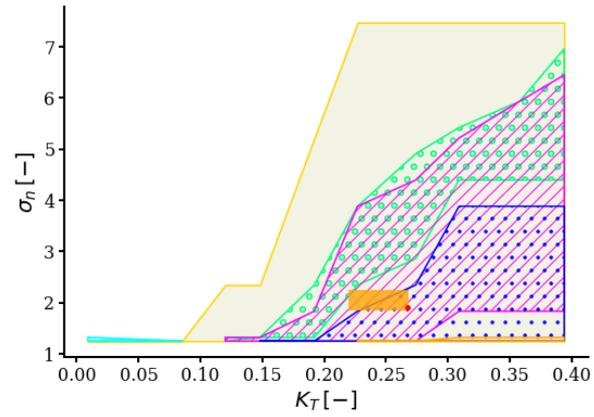
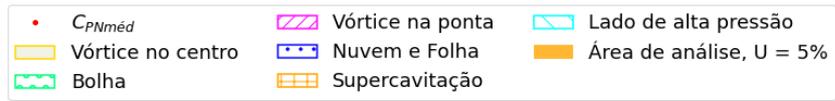
Fonte: adaptado de Golf (2018) e Wielgosz et al. (2019).

## Caso 3



Fonte: adaptado de Golf (2018) e Wielgosz et al. (2019).

## Caso 4



Fonte: adaptado de Golf (2018) e Wielgosz et al. (2019).

## APÊNDICE G – responseSurface.py

```

# -*- coding: utf-8 -*-
"""

Created on Sun Oct 17 23:17:58 2021
@author: atilio.scussel
"""

import datetime
from SALib.analyze import sobol
from SALib.sample import latin
from SALib.sample import saltelli
from scipy.interpolate import griddata
from matplotlib import cm
import matplotlib.pyplot as plt
import getopt, math, os, re, sys, h5py
import pickle
import pandas
import numpy as np
sys.path.append('.././../python')
sys.path.append('.././../opt/SALib-1.3.7/src')
from pymatt import dataReaders
# Fixing random state for reproducibility
np.random.seed(19680801)
def randrange(n, vmin, vmax):
    """
    Helper function to make an array of random numbers having shape (n, )
    with each number distributed Uniform(vmin, vmax).
    """
    return (vmax - vmin)*np.random.rand(n) + vmin
def getKeyword(key, controlFileString):
    """ Retrive a keyword from an xml document. """
    return re.findall('<{s?name="}?.*?.*</{s?>'.format(key, key),
controlFileString)[0].split(">")[1].split("</")[0]

```

```

def analyze_SALib_pre(data, paramDic):
    meshCode = data['meshCode']
    jobNameMeshRefList = data['jobNameMeshRefList']
    outputs = data['outputs']
    names = data['names']
    samples = paramDic['samples']
    bounds = data['realBounds']
    seed = data['seed'] if 'seed' in list(data) else 20
    dataPostRef_minLine = paramDic['dataPostRef_minLine'] if 'dataPostRef_minLine'
in list(
    paramDic) else 0
    dataPostRef_maxLine = paramDic['dataPostRef_maxLine'] if
'dataPostRef_maxLine' in list(
    paramDic) else 41
    typeOfStudy = paramDic['typeOfStudy'] if 'typeOfStudy' in paramDic else 'lhsi'
    verbose = paramDic['verbose'] if 'verbose' in paramDic else False
    nuaOutput = ['phi_' + meshName for meshName in meshCode] + ['u_' +
meshName for meshName in meshCode] + \
    ['alpha_1', 'alpha_2', 'p', 'phi_0', 'phi_gridRefCDF', 'a_1', 'a_2']
    np.random.seed(seed)
    if typeOfStudy == 'lhsvi':
        localInput = saltelli.sample(data, samples)
    if typeOfStudy == 'lhsi' :
        localInput = latin.sample(data, samples)
    data['intervalValid'] = np.full(localInput.shape[0], True)
    inputs = names[:-1] # remove gridRefCDFLike
    for keyOutput in outputs:
        data[keyOutput]['columnNames'] = ['index'] + names + nuaOutput
        data[keyOutput]['keyOutput'] = keyOutput
        data[keyOutput]['meshCode'] = meshCode
        data[keyOutput]['tabularResults'] = np.empty(
            [localInput.shape[0], 1 + len(names) + len(nuaOutput)])
        data[keyOutput]['tabularResults'][:] = np.nan

```

```

for i_name in range(0, len(names)):
    data[keyOutput]['tabularResults'][:, data[keyOutput]
        ['columnNames'].index(names[i_name])] = localInput[:,
i_name]
    data[keyOutput]['tabularResults'][:, data[keyOutput]
        ['columnNames'].index('index')] = np.arange(0,
localInput.shape[0])
for i in range(0, len(jobNameMeshRefList)):
    dataDirInterpList = [os.getcwd() + '/cases/' + case +
        '.pickle' for case in jobNameMeshRefList[i]]
    dataInterpList = []
    for dataDirInterp in dataDirInterpList:
        if os.path.exists(dataDirInterp):
            dataInterpList.append(pickle.load(open(dataDirInterp, 'rb')))
            if verbose: sys.stdout.write('\t\t\t[analyze_SALib_pre] Starting ' +
dataDirInterp + '\n')
        else:
            if verbose: sys.stdout.write('\t\t\t[analyze_SALib_pre] .pickle file '
+ dataDirInterp + ' not found. Exiting!\n')
            exit(-1)
    points_i = []
    for dataInterp in dataInterpList:
        indexInput = [ dataInterp['columnNames'].index(input) for input in inputs ]
        if len(points_i) == 0:
            points_i =
dataInterp['tabularResults'][dataPostRef_minLine:dataPostRef_maxLine, indexInput ]
        else:
            points_i = np.concatenate((points_i,
dataInterp['tabularResults'][dataPostRef_minLine:dataPostRef_maxLine, indexInput
]))
    for b in range(len(bounds)-1):
        points_i[:, b] = (points_i[:, b] - bounds[b]
            [0]) / (bounds[b][1] - bounds[b][0])
    if verbose:

```

```

        sys.stdout.write(
            '\t\t\t[analyze_SALib_pre] Total number of points_i: ' +
str(points_i.shape[0]) + '\n')
    if verbose:
        if verbose:
            sys.stdout.write('\t\t\t[analyze_SALib_pre] Minmax after norm: ' + str(
                np.min(points_i)) + ';' + str(np.max(points_i)) + '\n')
            if (np.min(points_i) >= 0.0) and (np.max(points_i) <= 1.0):
                sys.stdout.write(
                    '\t\t\t[analyze_SALib_pre] norm is expected\n')
        points_o = []
        for dataInterp in dataInterpList:
            if len(points_o) == 0:
                points_o =
dataInterp['tabularResults'][dataPostRef_minLine:dataPostRef_maxLine,
dataInterp['columnNames'].index(keyOutput)]
            else:
                points_o = np.concatenate( ( points_o,
dataInterp['tabularResults'][dataPostRef_minLine:dataPostRef_maxLine,
dataInterp['columnNames'].index(keyOutput)] ))

        interpData = griddata(points_i, points_o,
            localInput[:, :-1], method='cubic')
        interpData_nearest = griddata(
            points_i, points_o, localInput[:, :-1], method='nearest')
        data[keyOutput]['p_extrap'] = sum(
            np.isnan(interpData))/len(interpData)
        interpData[np.isnan(interpData)
            ] = interpData_nearest[np.isnan(interpData)]
        data[keyOutput]['tabularResults'][:, data[keyOutput]
            ['columnNames'].index('phi_' + meshCode[i])] =
interpData
    return data

```

```

if __name__ == '__main__':
    #####
    keysOutput = ['KT_mean', 'KQ_mean', 'eta_mean', 'Cav_index']
    var = 'eta_mean' # Change the variable in order to get the response for the
variable wanted
    jobnamebase = 'dtc_v03'
    nproc = 8
    ndiv = 1
    cdiv = 0
    advanceRatio = 'J80'
    myCurrentFolder = os.getcwd()
    isamples = 8192
    meshMinMax = ['j325', 'j700']
    keysInput = ['RPM', 'pv', 'gridRefCDF']
    meshNelem = [31164240, 21620280, 14250060, 8796570, 5164000, 2919825]
    meshCode = ['j700', 'j625', 'j550', 'j475', 'j400', 'j325']
    # meshCode = ['j700']
    os.system('cls')
    try:
        opts, args = getopt.getopt(sys.argv[1:], "n:c:N:j:S:", [
            "ndiv=", "cdiv=", "nproc=", "jobname=", "samples="])
    except getopt.GetoptError:
        print('Error: invalid argument.')
        exit(1)
    for opt, arg in opts:
        if opt in ('-n', "--ndiv"):
            ndiv = int(arg)
        if opt in ('-c', "--cdiv"):
            cdiv = int(arg)
        if opt in ('-N', "--nproc"):
            nproc = int(arg)
        if opt in ('-S', "--samples"):
            isamples = int(arg)
        if opt in ('-j', "--jobname"):

```

```

    jobnamebase = arg
    keysOutput = [keysOutput[i]
                  for i in range(len(keysOutput)) if (i % ndiv) == cdiv]
    jobName = jobnamebase + advanceRatio + '_v6Qi_' + str(
        isamples) + '_nua' + "{:03d}".format(cdiv) + "_" + "{:03d}".format(ndiv)

    nuaOutput = ['phi_' + meshName for meshName in meshCode] + ['u_' +
                                                                meshName for meshName in meshCode] +
    ['alpha_1', 'alpha_2', 'p', 'phi_0']
    jobNameMeshRefList = [[meshCode[r]] for r in range(len(meshCode))]

    probabilityLevels =
    [0.01,0.025,0.05,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,0.95,0.975,0.99]

    data = {'names': keysInput,
           'probabilityLevels': probabilityLevels,
           'bounds': [[0.0, 1.0], [0.0, 1.0], [0.0, 1.0]],
           'realBounds': [[2474.0625, 2600.93750],
                          [2242.5, 2357.5],
                          [0.0, 1.0]],
           'num_vars': len(keysInput),
           'outputs': keysOutput,
           'meshNelem': meshNelem,
           'meshCode': meshCode,
           'meshToWork': 0,
           'jobNameMeshRefList': jobNameMeshRefList
          }

    for samples in [isamples]:
        for keyOutput in keysOutput:
            data['outputs'] = [keyOutput]
            if not keyOutput in list(data):
                data[keyOutput] = {}
            for typeOfStudy in ['lhsvi', 'lhsi']:
                paramDic = {'samples': samples,
                           'nuaWorkDir': 'nuaWorkDir/' + jobName,

```

```

        'basename': 'nua',
            'deleteTabularResults': True,
            'nDivision': nproc,
            'gridStepsizeMethod': 3,
            'typeOfStudy': typeOfStudy,
            'nuaDir': myCurrentFolder,
            'jobName': jobName
        }
    data = analyze_SALib_pre(data, paramDic)
for mesh in meshCode:
    if mesh == 'j325':
        x325 = data[var]['tabularResults'][:, data[var]['columnNames'].index('RPM')]
        y325 = data[var]['tabularResults'][:, data[var]['columnNames'].index('pv')]
        z325 = data[var]['tabularResults'][:, data[var]['columnNames'].index('phi_' +
mesh)]
    if mesh == 'j400':
        x400 = data[var]['tabularResults'][:, data[var]['columnNames'].index('RPM')]
        y400 = data[var]['tabularResults'][:, data[var]['columnNames'].index('pv')]
        z400 = data[var]['tabularResults'][:, data[var]['columnNames'].index('phi_' +
mesh)]
    if mesh == 'j475':
        x475 = data[var]['tabularResults'][:, data[var]['columnNames'].index('RPM')]
        y475 = data[var]['tabularResults'][:, data[var]['columnNames'].index('pv')]
        z475 = data[var]['tabularResults'][:, data[var]['columnNames'].index('phi_' +
mesh)]
    if mesh == 'j550':
        x550 = data[var]['tabularResults'][:, data[var]['columnNames'].index('RPM')]
        y550 = data[var]['tabularResults'][:, data[var]['columnNames'].index('pv')]
        z550 = data[var]['tabularResults'][:, data[var]['columnNames'].index('phi_' +
mesh)]
    if mesh == 'j625':
        x625 = data[var]['tabularResults'][:, data[var]['columnNames'].index('RPM')]
        y625 = data[var]['tabularResults'][:, data[var]['columnNames'].index('pv')]

```

```

        z625 = data[var]['tabularResults'][:, data[var]['columnNames'].index('phi_' +
mesh)]
    if mesh == 'j700':
        x700 = data[var]['tabularResults'][:, data[var]['columnNames'].index('RPM')]
        y700 = data[var]['tabularResults'][:, data[var]['columnNames'].index('pv')]
        z700 = data[var]['tabularResults'][:, data[var]['columnNames'].index('phi_' +
mesh)]
    x1_csv = []
    y1_csv = []
    x2_csv = []
    y2_csv = []
    x3_csv = []
    y3_csv = []
    x4_csv = []
    y4_csv = []
    x5_csv = []
    y5_csv = []
    x6_csv = []
    y6_csv = []
    for mesh in meshCode:
        inputData1 = pandas.read_csv(os.getcwd() + "/cases/" + mesh + ".csv")
        # For each set of style and range settings, plot n random points in the box
        # defined by x in [23, 32], y in [0, 100], z in [zlow, zhigh].
        for m in range(len(inputData1['RPM'])):
            if mesh == 'j325':
                xs1 = inputData1['RPM']
                ys1 = inputData1['pv']
                kt1 = inputData1[var]
            if mesh == 'j400':
                xs2 = inputData1['RPM']
                ys2 = inputData1['pv']
                kt2 = inputData1[var]
            if mesh == 'j475':
                xs3 = inputData1['RPM']

```

```

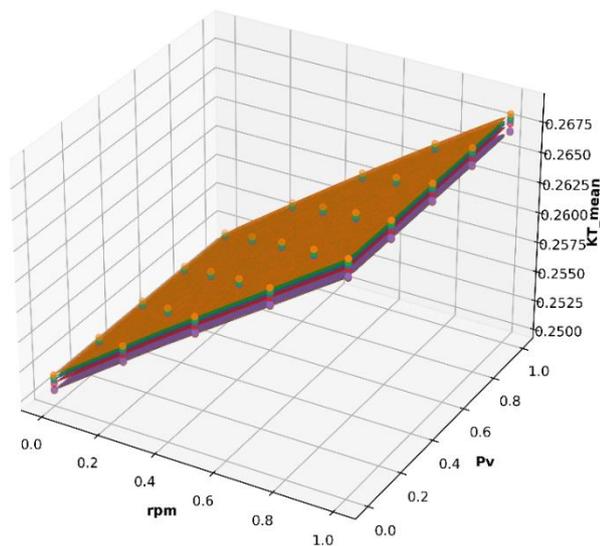
        ys3 = inputData1['pv']
        kt3 = inputData1[var]
    if mesh == 'j550':
        xs4 = inputData1['RPM']
        ys4 = inputData1['pv']
        kt4 = inputData1[var]
    if mesh == 'j625':
        xs5 = inputData1['RPM']
        ys5 = inputData1['pv']
        kt5 = inputData1[var]
    if mesh == 'j700':
        xs6 = inputData1['RPM']
        ys6 = inputData1['pv']
        kt6 = inputData1[var]
for i in range(len(inputData1['RPM'])):
    x1_csv.append((xs1[i] - np.min(xs1))/(np.max(xs1) - np.min(xs1)))
    y1_csv.append((ys1[i] - np.min(ys1))/(np.max(ys1) - np.min(ys1)))
    x2_csv.append((xs2[i] - np.min(xs2))/(np.max(xs2) - np.min(xs2)))
    y2_csv.append((ys2[i] - np.min(ys2))/(np.max(ys2) - np.min(ys2)))
    x3_csv.append((xs3[i] - np.min(xs3))/(np.max(xs3) - np.min(xs3)))
    y3_csv.append((ys3[i] - np.min(ys3))/(np.max(ys3) - np.min(ys3)))
    x4_csv.append((xs4[i] - np.min(xs4))/(np.max(xs4) - np.min(xs4)))
    y4_csv.append((ys4[i] - np.min(ys4))/(np.max(ys4) - np.min(ys4)))
    x5_csv.append((xs5[i] - np.min(xs5))/(np.max(xs5) - np.min(xs5)))
    y5_csv.append((ys5[i] - np.min(ys5))/(np.max(ys5) - np.min(ys5)))
    x6_csv.append((xs6[i] - np.min(xs6))/(np.max(xs6) - np.min(xs6)))
    y6_csv.append((ys6[i] - np.min(ys6))/(np.max(ys6) - np.min(ys6)))
fig = plt.figure(dpi = 500, figsize = (10,8))
ax = plt.axes(projection='3d')
ax.scatter(x1_csv, y1_csv, kt1)
ax.scatter(x2_csv, y2_csv, kt2)
ax.scatter(x3_csv, y3_csv, kt3)
ax.scatter(x4_csv, y4_csv, kt4)
ax.scatter(x5_csv, y5_csv, kt5)

```

```
ax.scatter(x6_csv, y6_csv, kt6)
ax.plot_trisurf(x325, y325, z325, linewidth = 0.2, antialiased = True)
ax.plot_trisurf(x400, y400, z400, linewidth = 0.2, antialiased = True)
ax.plot_trisurf(x475, y475, z475, linewidth = 0.2, antialiased = True)
ax.plot_trisurf(x550, y550, z550, linewidth = 0.2, antialiased = True)
ax.plot_trisurf(x625, y625, z625, linewidth = 0.2, antialiased = True)
ax.plot_trisurf(x700, y700, z700, linewidth = 0.2, antialiased = True)
ax.set_title("Response surface", size=18) #Add y=1.6 for scatter plot
ax.set_xlabel("RPM", fontweight = "bold")
ax.set_ylabel("Pv", fontweight = "bold")
ax.set_zlabel("%s" %var, fontweight = "bold")
print('[nua_v6Qi] End \t\t Date: ' + str(datetime.datetime.now()))
```

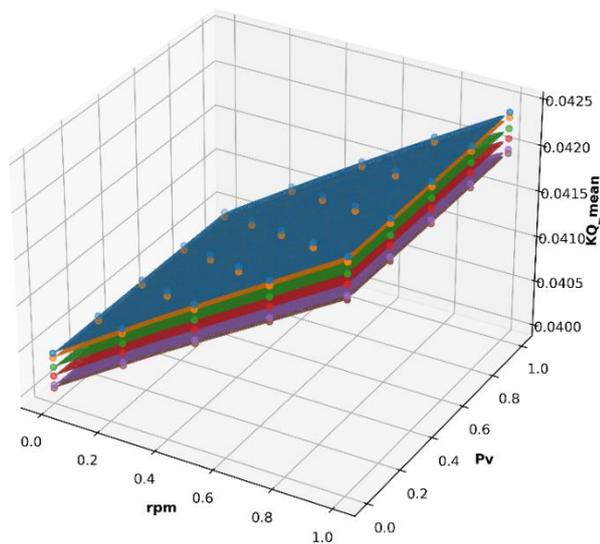
## APÊNDICE H – Modelos interpoladores

Figura 41 - Modelo interpolador para  $K_T$ .

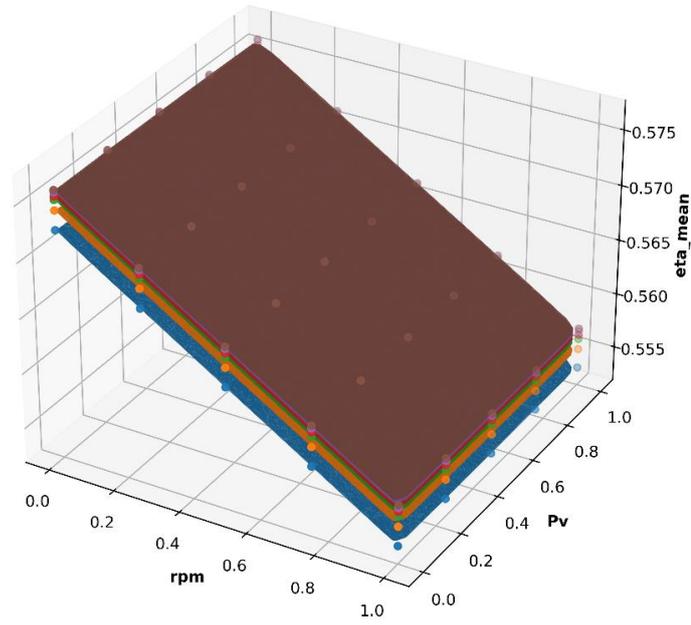


Fonte: autoria própria (2022).

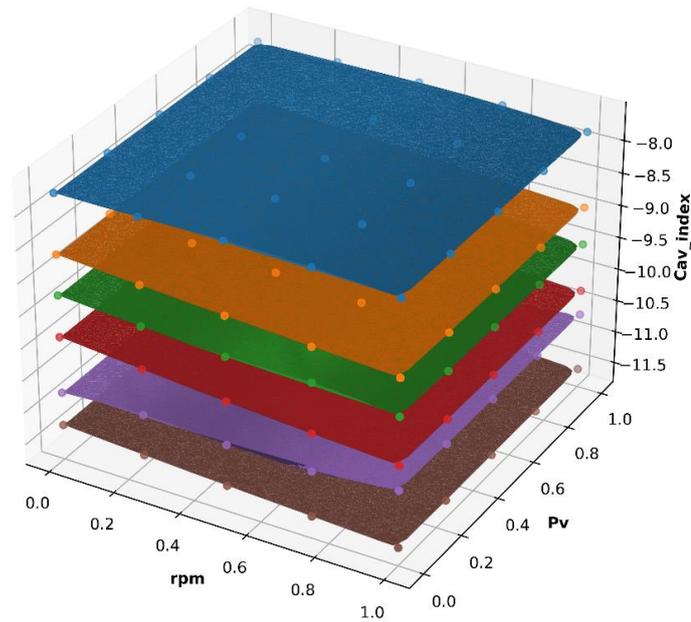
Figura 42 - Modelo interpolador para  $K_Q$ .



Fonte: autoria própria (2022).

Figura 43 - Modelo interpolador para  $\eta$ .

Fonte: autoria própria (2022).

Figura 44 - Modelo interpolador para  $Cav_{index}$ 

Fonte: autoria própria (2022).

## APÊNDICE I – Arquivo csv com todos os pontos analisados

Tabela 13 - Arquivo csv dos 150 pontos avaliados na simulação numérica.

RPM	P <sub>v</sub>	nElem	KT_mean	KQ_mean	eta_mean	Cav_index
2474,063	2242,500	2919825	0,251	0,040	0,573	-7,713
2474,063	2271,250	2919825	0,251	0,040	0,573	-7,712
2474,063	2300,000	2919825	0,251	0,040	0,573	-7,710
2474,063	2328,750	2919825	0,251	0,040	0,573	-7,708
2474,063	2357,500	2919825	0,251	0,040	0,573	-7,707
2505,781	2242,500	2919825	0,256	0,041	0,568	-7,713
2505,781	2271,250	2919825	0,256	0,041	0,568	-7,711
2505,781	2300,000	2919825	0,256	0,041	0,568	-7,710
2505,781	2328,750	2919825	0,256	0,041	0,568	-7,708
2505,781	2357,500	2919825	0,256	0,041	0,568	-7,707
2537,500	2242,500	2919825	0,260	0,041	0,563	-7,704
2537,500	2271,250	2919825	0,260	0,041	0,563	-7,703
2537,500	2300,000	2919825	0,260	0,041	0,563	-7,702
2537,500	2328,750	2919825	0,260	0,041	0,563	-7,700
2537,500	2357,500	2919825	0,260	0,041	0,563	-7,699
2569,219	2242,500	2919825	0,264	0,042	0,558	-7,729
2569,219	2271,250	2919825	0,264	0,042	0,558	-7,728
2569,219	2300,000	2919825	0,264	0,042	0,558	-7,727
2569,219	2328,750	2919825	0,264	0,042	0,558	-7,725
2569,219	2357,500	2919825	0,264	0,042	0,558	-7,724
2600,938	2242,500	2919825	0,268	0,042	0,554	-7,802
2600,938	2271,250	2919825	0,268	0,042	0,554	-7,801
2600,938	2300,000	2919825	0,268	0,042	0,554	-7,799
2600,938	2328,750	2919825	0,268	0,042	0,554	-7,798
2600,938	2357,500	2919825	0,268	0,042	0,554	-7,797
2474,063	2242,500	5164000	0,252	0,040	0,574	-8,632

Continua(...)

RPM	P <sub>v</sub>	nElem	KT_mean	KQ_mean	eta_mean	Cav_index
2474,063	2271,250	5164000	0,252	0,040	0,574	-8,630
2474,063	2300,000	5164000	0,252	0,040	0,574	-8,629
2474,063	2328,750	5164000	0,252	0,040	0,574	-8,627
2474,063	2357,500	5164000	0,252	0,040	0,574	-8,626
2505,781	2242,500	5164000	0,256	0,041	0,569	-8,724
2505,781	2271,250	5164000	0,256	0,041	0,569	-8,722
2505,781	2300,000	5164000	0,256	0,041	0,569	-8,721
2505,781	2328,750	5164000	0,256	0,041	0,569	-8,720
2505,781	2357,500	5164000	0,256	0,041	0,569	-8,718
2537,500	2242,500	5164000	0,260	0,041	0,565	-8,809
2537,500	2271,250	5164000	0,260	0,041	0,565	-8,807
2537,500	2300,000	5164000	0,260	0,041	0,565	-8,806
2537,500	2328,750	5164000	0,260	0,041	0,565	-8,805
2537,500	2357,500	5164000	0,260	0,041	0,565	-8,803
2569,219	2242,500	5164000	0,264	0,042	0,560	-8,889
2569,219	2271,250	5164000	0,264	0,042	0,560	-8,888
2569,219	2300,000	5164000	0,264	0,042	0,560	-8,887
2569,219	2328,750	5164000	0,264	0,042	0,560	-8,885
2569,219	2357,500	5164000	0,264	0,042	0,560	-8,884
2600,938	2242,500	5164000	0,268	0,042	0,555	-8,960
2600,938	2271,250	5164000	0,268	0,042	0,555	-8,959
2600,938	2300,000	5164000	0,268	0,042	0,555	-8,958
2600,938	2328,750	5164000	0,268	0,042	0,555	-8,956
2600,938	2357,500	5164000	0,268	0,042	0,555	-8,955
2474,063	2242,500	8796570	0,252	0,040	0,575	-9,251
2474,063	2271,250	8796570	0,252	0,040	0,575	-9,249
2474,063	2300,000	8796570	0,252	0,040	0,575	-9,248
2474,063	2328,750	8796570	0,252	0,040	0,575	-9,246
2474,063	2357,500	8796570	0,252	0,040	0,575	-9,245
2505,781	2242,500	8796570	0,256	0,041	0,570	-9,347
2505,781	2271,250	8796570	0,256	0,041	0,570	-9,346

Continua (...)

RPM	P <sub>v</sub>	nElem	KT_mean	KQ_mean	eta_mean	Cav_index
2505,781	2300,000	8796570	0,256	0,041	0,570	-9,344
2505,781	2328,750	8796570	0,256	0,041	0,570	-9,343
2505,781	2357,500	8796570	0,256	0,041	0,570	-9,341
2537,500	2242,500	8796570	0,260	0,041	0,566	-9,423
2537,500	2271,250	8796570	0,260	0,041	0,566	-9,422
2537,500	2300,000	8796570	0,260	0,041	0,566	-9,420
2537,500	2328,750	8796570	0,260	0,041	0,566	-9,419
2537,500	2357,500	8796570	0,260	0,041	0,566	-9,417
2569,219	2242,500	8796570	0,264	0,042	0,561	-9,423
2569,219	2271,250	8796570	0,264	0,042	0,561	-9,422
2569,219	2300,000	8796570	0,264	0,042	0,561	-9,420
2569,219	2328,750	8796570	0,264	0,042	0,561	-9,419
2569,219	2357,500	8796570	0,264	0,042	0,561	-9,417
2600,938	2242,500	8796570	0,268	0,042	0,556	-9,536
2600,938	2271,250	8796570	0,268	0,042	0,556	-9,535
2600,938	2300,000	8796570	0,268	0,042	0,556	-9,533
2600,938	2328,750	8796570	0,268	0,042	0,556	-9,532
2600,938	2357,500	8796570	0,268	0,042	0,556	-9,530
2474,063	2242,500	14250060	0,251	0,040	0,575	-9,890
2474,063	2271,250	14250060	0,251	0,040	0,575	-9,888
2474,063	2300,000	14250060	0,251	0,040	0,575	-9,887
2474,063	2328,750	14250060	0,251	0,040	0,575	-9,885
2474,063	2357,500	14250060	0,251	0,040	0,575	-9,884
2505,781	2242,500	14250060	0,255	0,041	0,571	-10,003
2505,781	2271,250	14250060	0,255	0,041	0,571	-10,002
2505,781	2300,000	14250060	0,255	0,041	0,571	-10,000
2505,781	2328,750	14250060	0,255	0,041	0,571	-9,999
2505,781	2357,500	14250060	0,255	0,041	0,571	-9,997
2537,500	2242,500	14250060	0,260	0,041	0,566	-10,105
2537,500	2271,250	14250060	0,260	0,041	0,566	-10,105
2537,500	2300,000	14250060	0,260	0,041	0,566	-10,102

Continua (...)

RPM	P <sub>v</sub>	nElem	KT_mean	KQ_mean	eta_mean	Cav_index
2537,500	2328,750	14250060	0,260	0,041	0,566	-10,101
2537,500	2357,500	14250060	0,260	0,041	0,566	-10,099
2569,219	2242,500	14250060	0,264	0,042	0,561	-10,191
2569,219	2271,250	14250060	0,264	0,042	0,561	-10,190
2569,219	2300,000	14250060	0,264	0,042	0,561	-10,188
2569,219	2328,750	14250060	0,264	0,042	0,561	-10,187
2569,219	2357,500	14250060	0,264	0,042	0,561	-10,185
2600,938	2242,500	14250060	0,268	0,042	0,557	-10,270
2600,938	2271,250	14250060	0,268	0,042	0,557	-10,268
2600,938	2300,000	14250060	0,268	0,042	0,557	-10,267
2600,938	2328,750	14250060	0,268	0,042	0,557	-10,266
2600,938	2357,500	14250060	0,268	0,042	0,557	-10,264
2474,063	2242,500	21620280	0,251	0,040	0,576	-10,758
2474,063	2271,250	21620280	0,251	0,040	0,576	-10,757
2474,063	2300,000	21620280	0,251	0,040	0,576	-10,755
2474,063	2328,750	21620280	0,251	0,040	0,576	-10,754
2474,063	2357,500	21620280	0,251	0,040	0,576	-10,752
2505,781	2242,500	21620280	0,255	0,041	0,571	-10,711
2505,781	2271,250	21620280	0,255	0,041	0,571	-10,709
2505,781	2300,000	21620280	0,255	0,041	0,571	-10,708
2505,781	2328,750	21620280	0,255	0,041	0,571	-10,706
2505,781	2357,500	21620280	0,255	0,041	0,571	-10,705
2537,500	2242,500	21620280	0,259	0,041	0,566	-10,772
2537,500	2271,250	21620280	0,259	0,041	0,566	-10,771
2537,500	2300,000	21620280	0,259	0,041	0,566	-10,769
2537,500	2328,750	21620280	0,259	0,041	0,566	-10,768
2537,500	2357,500	21620280	0,259	0,041	0,566	-10,766
2569,219	2242,500	21620280	0,263	0,042	0,562	-10,773
2569,219	2271,250	21620280	0,263	0,042	0,562	-10,772
2569,219	2300,000	21620280	0,263	0,042	0,562	-10,771
2569,219	2328,750	21620280	0,263	0,042	0,562	-10,769

Continua (...)

RPM	P <sub>v</sub>	nElem	KT_mean	KQ_mean	eta_mean	Cav_index
2569,219	2357,500	21620280	0,263	0,042	0,562	-10,768
2600,938	2242,500	21620280	0,267	0,042	0,557	-10,650
2600,938	2271,250	21620280	0,267	0,042	0,557	-10,649
2600,938	2300,000	21620280	0,267	0,042	0,557	-10,648
2600,938	2328,750	21620280	0,267	0,042	0,557	-10,646
2600,938	2357,500	21620280	0,267	0,042	0,557	-10,645
2474,063	2242,500	31164240	0,251	0,040	0,576	-11,267
2474,063	2271,250	31164240	0,251	0,040	0,576	-11,266
2474,063	2300,000	31164240	0,251	0,040	0,576	-11,264
2474,063	2328,750	31164240	0,251	0,040	0,576	-11,263
2474,063	2357,500	31164240	0,251	0,040	0,576	-11,261
2505,781	2242,500	31164240	0,255	0,041	0,571	-11,341
2505,781	2271,250	31164240	0,255	0,041	0,571	-11,340
2505,781	2300,000	31164240	0,255	0,041	0,571	-11,338
2505,781	2328,750	31164240	0,255	0,041	0,571	-11,337
2505,781	2357,500	31164240	0,255	0,041	0,571	-11,335
2537,500	2242,500	31164240	0,259	0,041	0,567	-11,409
2537,500	2271,250	31164240	0,259	0,041	0,567	-11,408
2537,500	2300,000	31164240	0,259	0,041	0,567	-11,406
2537,500	2328,750	31164240	0,259	0,041	0,567	-11,405
2537,500	2357,500	31164240	0,259	0,041	0,567	-11,404
2569,219	2242,500	31164240	0,263	0,041	0,562	-11,469
2569,219	2271,250	31164240	0,263	0,041	0,562	-11,467
2569,219	2300,000	31164240	0,263	0,041	0,562	-11,466
2569,219	2328,750	31164240	0,263	0,041	0,562	-11,465
2569,219	2357,500	31164240	0,263	0,041	0,562	-11,463
2600,938	2242,500	31164240	0,267	0,042	0,557	-11,530
2600,938	2271,250	31164240	0,267	0,042	0,557	-11,529
2600,938	2300,000	31164240	0,267	0,042	0,557	-11,528
2600,938	2328,750	31164240	0,267	0,042	0,557	-11,526
2600,938	2357,500	31164240	0,267	0,042	0,557	-11,525

Fonte: autoria própria (2022).

**APÊNDICE J – Índices de Sobol combinados**

Todos os valores aqui apresentados são expressões em termos de porcentagem.

Tabela 14 - Índices de Sobol combinados.

	KT_mean	KQ_mean	eta_mean	Cav_index
RPM	99,000	95,300	98,100	0,300
P <sub>v</sub>	0,100	0,100	0,100	0,300
nElem	0,900	4,600	1,800	99,400

Fonte: autoria própria (2022).

## ANEXO A – Módulo prepareCases.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Oct 5 21:58:36 2021
@author: artur
"""

import pandas
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import yaml
import pandas
import re
import os

from everydaytools import scripting
font = {"family": "serif",
        "weight": "normal",
        "size": 18}
matplotlib.rc("font", **font)
matplotlib.rcParams["figure.figsize"] = (12, 9)
matplotlib.style.use("seaborn-colorblind")
# %% Helper functions.
def sub(source, tag, newValue):
    found = re.findall("<{}>.*</{}>".format(tag, tag), source)
    if len(found) == 0:
        raise ValueError("Haven't found tag {}".format(tag))
    return re.sub("<{}>.*</{}>".format(tag, tag), "<{}>{}</{}>".format(tag, newValue, tag),
source)
# %% Constants.
seriesName = "series0"
# Cluster settings for computing a decent no. nodes.
```

```

# Assuming 100k cells per processor and 20 cores per node (Marclus 4). Cap at 10.
# Ideally would want to go as low as 10-20k cells for good scaling but then one job
would take up the entire user limit.
nCoresPerNode = 20
nCellsPerCoreDesired = 100e3
maxNodesPerJob = 10
# Common settings for all cases.
maxWallTime = 120. # hours
# TODO: AL: I interpolated Kt to for value of 0.25 to get this. Make sure it all adds up.
J = 0.5714285714285715
# Advance velocity.
Va = 3.625
# Diameter.
D = 0.15
# Grid levels to use - should match cgns names.
# nCells = [31.2, 21.6, 14.3, 8.8, 5.2, 2.0] # In million. Rough estimates assuming
this is the same as in Katsuno et al. (2021).
# NOTE: AL: I added one coarser grid compared to what Eduardo used.
# gridLevels = ["f0700", "f0625", "f0550", "f0475", "f0400", "f0325"]
nCells = [2.0]
gridLevels = ["f0325"]
# Compute central RPM.
n = Va / (J*D)
RPM = n*60
# Compute RPM range.
# TODO check these values.
RPMRange = np.linspace(RPM*0.975, RPM*1.025, 5)
omegaRange = 2*np.pi*(RPMRange/60)
baseCase = "baseCase_AFM"
targetDir = "batchData_UQ_{}".format(seriesName)
# %% Prepare intermediate settings.
# Compute a reasonable no. nodes for each grid.
nNodes = [max(1, min(maxNodesPerJob, int(np.ceil(nCells[i]*1e6 /
(nCoresPerNode*nCellsPerCoreDesired)))))) for i in range(len(gridLevels))]

```

```

# Assemble cases - full factorial of RPM and grid refinement levels.
cases = pandas.DataFrame(
    data=[(RPMRange[i], omegaRange[i], gridLevels[iGrid], nNodes[iGrid],
nNodes[iGrid]*nCoresPerNode)
        for i in range(len(RPMRange)) for iGrid in range(len(gridLevels))],
    columns=["RPM", "omega", "gridLevel", "nNodes", "nCores"])
cases.to_csv("samplingPlan_{}.csv".format(seriesName), index=False)
# %% Make cases.
for iCase in cases.index:
    case = "case_{:d}".format(iCase)
    target = os.path.join(targetDir, case)
    # Clone the source case.
    cmds = [
        "rm -rf {}".format(os.path.join(targetDir, case)),
        "mkdir -p {}".format(target),
        "cp -r {}/* {}".format(baseCase, target),
    ]
    scripting.runBatch(cmds, pwd=".", log=False)
    # Adjust controls.
    with open(os.path.join(target, "controls.xml"), "r") as infile:
        controls = infile.read()
        controls = sub(controls, "gridFileName", "grid_{}".format(cases.loc[iCase,
"gridLevel"]))
        controls = sub(controls, "gridFilePath", "../grids_refresco")
        controls = sub(controls, "magnitude", "{:.6e}".format(cases.loc[iCase, "omega"]))
        controls = sub(controls, "referenceVelocity", "{:.6e}".format(Va))
        controls = sub(controls, "velocity", "{:.6e} 0 0".format(Va))
        controls = sub(controls, "velocity", "{:.6e} 0 0".format(Va))
    with open(os.path.join(target, "controls.xml"), "w") as outfile:
        outfile.write(controls)
    # Adjust the jobfile.
    with open(os.path.join(target, "Allrun"), "r") as infile:
        jobfile = infile.read()

```

```
jobfile = re.sub("--nodes=[0-9]+", "--nodes={:d}".format(cases.loc[iCase,  
"nNodes"]), jobfile)  
jobfile = re.sub("--time=[0-9:]+", "--time={:.0f}:00:00".format(maxWallTime), jobfile)  
with open(os.path.join(target, "Allrun"), "w") as outfile:  
    outfile.write(jobfile)
```

## ANEXO B – Módulo processCases.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Sep 29 22:55:03 2021
@author: artur
"""
import pandas
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import os
import re
import h5py
import pickle
from pymatt import dataReaders
import ingesInterface

# %% Main data processing routines.
def getKeyword(key, controlFileString):
    """ Retrive a keyword from an xml document. """
    return re.findall('<{ }\s?name="]?.*>.*</{ }>'.format(key, key),
controlFileString)[0].split(">")[1].split("</")[0]

def readCaseData(casePath, unsteady=False, nAv=100):
    data = {}
    # Read outputs.
    data["counters"] = dataReaders.read_tec_plot(os.path.join(casePath,
"counters.dat"))
    data["residuals"] = dataReaders.read_tec_plot(os.path.join(casePath,
"residuals.dat"))
```

```

data["y+"] = dataReaders.read_tec_plot(os.path.join(casePath, "data",
"yPlus.dat"))
data["pressure"] = dataReaders.read_tec_plot(os.path.join(casePath, "data",
"scalar_pressure.dat"))
for comp in ["blades", "hub"]:
    data["forces_{}".format(comp)] =
dataReaders.read_tec_plot(os.path.join(casePath, "data",
"forces_{}.dat".format(comp)))
    data["moments_{}".format(comp)] =
dataReaders.read_tec_plot(os.path.join(casePath, "data",
"moments_{}.dat".format(comp)))
    if unsteady:
        data["motion"] = dataReaders.read_tec_plot(os.path.join(casePath,
"body_ship_motions.dat"))
        data["residuals_unstd"] = dataReaders.read_tec_plot(os.path.join(casePath,
"residuals_unstd.dat"))
        data["Co"] = dataReaders.read_tec_plot(os.path.join(casePath, "data",
"scalar_CourantNumber.dat"))
# Read controls and extract important values.
with open(os.path.join(casePath, "controls.xml"), "r") as infile:
    controls = infile.read()
data["omega"] = float(getKeyword("magnitude", controls))
data["rps"] = 1. / ((2*np.pi) / data["omega"])
data["D"] = float(getKeyword("referenceLength", controls))
data["Uinf"] = float(getKeyword("referenceVelocity", controls))
data["rhoInf"] = float(getKeyword("density", controls))
data["muInf"] = float(getKeyword("viscosityMolecular", controls))
data["nuInf"] = data["muInf"] / data["rhoInf"]
with open(os.path.join(casePath, "report_general.xml"), "r") as infile:
    report = infile.read()
data["wallTime"] = float(getKeyword("computation_time", report))
# Compute integral quantities.
iAv = data["counters"].shape[0] - nAv

```

```

    data["Kt"] = np.mean(data["forces_blades"]["ForceTotalX"].values[iAv:] /
(data["rhoInf"] * data["rps"]**2 * data["D"]**4))
    data["Kq"] = np.mean(-1*data["moments_blades"]["MomentTotalX"].values[iAv:] /
(data["rhoInf"] * data["rps"]**2 * data["D"]**5))
    Uref = np.sqrt(data["Uinf"]**2 + (data["omega"]*0.7*data["D"]/2)**2)
    data["CpMin"] = np.mean(data["pressure"]["PressureMin"].values[iAv:] /
(0.5*data["rhoInf"]*Uref**2))
    data["CpnMin"] = np.mean(data["pressure"]["PressureMin"].values[iAv:] /
(0.5*data["rhoInf"]*(data["rps"]*data["D"])**2))
    return data

def saveCaseData(data, filename):
    with open(filename, "wb") as outfile:
        pickle.dump(data, outfile, protocol=pickle.HIGHEST_PROTOCOL)

def loadCaseData(filename):
    with open(filename, "rb") as infile:
        return pickle.load(infile)

def readBladeCgns(filename):
    cgnsBaseName = "BCWall_2" # Note that this will be different if BC names are
changed or iso-contour is used.
    surfaceCgnsFile = h5py.File(filename, mode="r")
    # nSurfVertices = surfaceCgnsFile["Base_Surface_Elements"][cgnsBaseName][
data"][0][0]
    nSurfFaces = surfaceCgnsFile["Base_Surface_Elements"][cgnsBaseName][
data"][1][0]
    # nSurfBndVertices =
surfaceCgnsFile["Base_Surface_Elements"][cgnsBaseName][
data"][2][0]
    # For quads, connectivity is ordered.
    surfConn =
surfaceCgnsFile["Base_Surface_Elements"][cgnsBaseName][
"Elem_Quads"]["ElementConnectivity"][
data][:]
    surfFaces = surfConn.reshape((nSurfFaces, 4)) - 1

```

```

# Get vertices.
surfVertices = np.vstack([
surfaceCgnsFile["Base_Surface_Elements"][cgnsBaseName]["GridCoordinates"]["Co
ordinateX"][" data"][:],
surfaceCgnsFile["Base_Surface_Elements"][cgnsBaseName]["GridCoordinates"]["Co
ordinateY"][" data"][:],
surfaceCgnsFile["Base_Surface_Elements"][cgnsBaseName]["GridCoordinates"]["Co
ordinateZ"][" data"][:],
]).T
# Get pressure.
surfPressure =
surfaceCgnsFile["Base_Surface_Elements"][cgnsBaseName]["CellData"]["Pressure"]
" data"][:]
# Compute centres, areas, and normals.
surfCentres = np.mean(surfVertices[surfFaces, :], axis=1)
surfAreas = 0.5 * np.linalg.norm(
    np.cross(surfVertices[surfFaces[:, 1], :] - surfVertices[surfFaces[:, 0], :],
            surfVertices[surfFaces[:, 2], :] - surfVertices[surfFaces[:, 1], :]) +
    np.cross(surfVertices[surfFaces[:, 3], :] - surfVertices[surfFaces[:, 2], :],
            surfVertices[surfFaces[:, 0], :] - surfVertices[surfFaces[:, 3], :]),
    axis=1)
surfNormals = np.cross(surfVertices[surfFaces[:, 1], :] - surfVertices[surfFaces[:,
0], :],
                    surfVertices[surfFaces[:, 2], :] - surfVertices[surfFaces[:, 1], :])
surfNormals /= np.linalg.norm(surfNormals, axis=1)[:, np.newaxis]
# Assemble into a data frame.
surCDFta = pandas.DataFrame(data=np.hstack([surfCentres, surfAreas[:,
np.newaxis], surfNormals, surfPressure[:, np.newaxis]]),
                            columns=["CoordinateX", "CoordinateY", "CoordinateZ", "area",
"n_i", "n_j", "n_k", "pressure"])
return surCDFta

# %% Analysis pipeline..
if __name__ == "__main__":

```

```

# TODO this is what will need to be done for each case.
# ---
# Constants.
# TODO wrap in a function or big loop, or put at the top somewhere.
case = os.getcwd() + "/case_AFM"
D = 0.15
# TODO may need to adjust path depending on your folder structure.
propFilename = os.getcwd() + "/geometry/DTCPpropeller_adj.ppg"
chordwiseOffsets = np.arange(0, 1.000000001, 1e-3) # For making the prop grid
used for computing x/c.
# ---
# ===
# PART 1: ReFRESKO data files.
# Read the data.
data = readCaseData(case)
# Save processed CFD data for future use and test loading.
saveCaseData(data, os.path.join(case, "data_refrescoOutputs.pkl"))
# ===
# PART 2: Blade surface data.
# Read coprocessing output.
filename = os.path.join(case, "data_coprocessing", "boundary.cgns")
surCDFta = readBladeCgns(filename)
# Compute prop-specific info.
surCDFta["r/R"] = np.sqrt(np.sum(surCDFta[["CoordinateY",
"CoordinateZ"]].values**2, axis=1)) / (D/2.)
# Kind of a bad way to do this but I don't have a reverse formula for prop
geometry.
# Get unique radii to some tolerance.
tol = 1e-3
uniqueRadii =
np.sort(surCDFta["r/R"].values[~(np.triu(np.abs(surCDFta["r/R"].values[:, None] -
surCDFta["r/R"].values) <= tol, 1)).any(0)])
radialOffsets = np.linspace(surCDFta["r/R"].min(), 1.0, len(uniqueRadii))
# Create propeller geometry.

```

```

testProp = ingesInterface.PropGeom()
testProp.createFromPpg(propFilename)
testProp.changeDiameter(D)
testProp.setNewRadialOffsets(radialOffsets)
testProp.setNewChordwiseOffsets(chordwiseOffsets)
propGrid, xcGrid, rRGrid = ingesInterface.getPropGrid(testProp)
# Need to rotate to match the grid.
propGrid[:, :, 2] *= -1.
propGrid[:, :, 0] *= -1.
# Flatten data.
propGridFlat = propGrid.reshape((propGrid.shape[0]*propGrid.shape[1], 3))
xcGridFlat = xcGrid.reshape((propGrid.shape[0]*propGrid.shape[1]))
# Match grid points to the created geometry to work out x/c.
surCDFta["x/c"] = 0.
for j in surCDFta.index:
    dist = np.linalg.norm(surCDFta[["CoordinateX", "CoordinateY",
"CoordinateZ"]].values[j, :] - propGridFlat, axis=1)
    surCDFta.loc[j, "x/c"] = xcGridFlat[np.argmin(dist)]
# Save processed surface data.
surCDFta.to_csv(os.path.join(case, "data_bladeSurface.csv"), index=False)
# %% Only use this for testing.
saveFigs = True
font = {"family": "serif",
        "weight": "normal",
        "size": 20}
matplotlib.rc("font", **font)
matplotlib.rcParams["figure.figsize"] = (12, 9)
matplotlib.style.use("seaborn-colorblind")
# Test the loading function.
# data2 = loadCaseData(os.path.join(case, "testDataDump.pkl"))
# Test saving prop geom.
ingesInterface.propToVtk(os.getcwd() + "/geometry/propCentrelineGrid.vtk",
propGrid, rRGrid, xcGrid)
# ===

```

```

fig, ax = plt.subplots()
ax.set_ylabel("y+")
ax.set_xlabel("Iteration")
plt.plot(data["counters"]["TotalIter"].values, data["y+"]["y_plusMax"].values,
label="Max")
plt.plot(data["counters"]["TotalIter"].values, data["y+"]["y_plusAverage"].values,
label="Mean")
ax.legend()
if saveFigs:
    plt.savefig(os.getcwd() + "/Figures/test_maxYplus.png", dpi=200,
bbox_inches="tight")
# ===
fig, ax = plt.subplots()
ax.set_ylabel("Moment [Nm]")
ax.set_xlabel("Iteration")
plt.plot(data["counters"]["TotalIter"].values,
data["moments_blades"]["MomentTotalX"].values, label="Blades")
plt.plot(data["counters"]["TotalIter"].values,
data["moments_hub"]["MomentTotalX"].values, label="Hub")
if saveFigs:
    plt.savefig(os.getcwd() + "/Figures/test_moment.png", dpi=200,
bbox_inches="tight")
# ===
fig, ax = plt.subplots()
ax.set_ylabel("Minimum pressure [Pa]")
ax.set_xlabel("Iteration")
plt.plot(data["counters"]["TotalIter"].values, data["pressure"]["PressureMin"].values)
if saveFigs:
    plt.savefig(os.getcwd() + "/Figures/test_minPressure.png", dpi=200,
bbox_inches="tight")
# ===
fig, ax = plt.subplots()
ax.set_ylabel("Thrust [N]")
ax.set_xlabel("Iteration")

```

```

plt.plot(data["counters"]["TotalIter"].values,
data["forces_blades"]["ForceTotalX"].values, label="Blades")
plt.plot(data["counters"]["TotalIter"].values,
data["forces_hub"]["ForceTotalX"].values, label="Hub")
if saveFigs:
    plt.savefig(os.getcwd() + "/Figures/test_moment.png", dpi=200,
bbox_inches="tight")
# ===
for norm in ["2", "inf"]:
    # ---
    fig, ax = plt.subplots()
    ax.set_ylabel("Residual L{}".format(norm))
    ax.set_yscale("log")
    for f in data["residuals"]:
        if f.startswith("L{}".format(norm)):
            lab = f.split("_")[1]
            try:
                plt.plot(data["counters"]["TotalIter"].values, data["residuals"][f].values,
label=lab)
                ax.set_xlabel("TotalIter")
            except ValueError:
                plt.plot(data["residuals"][f].values, label=lab)
                ax.set_xlabel("Time step")
    ax.legend(loc="upper left", bbox_to_anchor=(1.0, 1.0))
    if saveFigs:
        plt.savefig(os.getcwd() + "/Figures/test_residuals_{}.png".format(norm),
dpi=200, bbox_inches="tight")

```

## ANEXO C – Módulo cav\_indexCalcs.py

```

# -*- coding: utf-8 -*-
"""
Created on Mon Dec 27 19:58:13 2021
@author: artur
"""

import numpy as np
import h5py
import re
import os
import pandas

def getKeyword(key, controlFileString):
    """ Retrive a keyword from an xml document. """
    return re.findall('<{s?name="}?.*?.*</{>'.format(key, key),
controlFileString)[0].split(">")[1].split("</")[0]

D = 0.15
rho_exp = 998.16 # kg/m3
Va_exp = 3.625 # m/s
pv_exp_list = [0.022425*1e5, 0.0227125*1e5, 0.02300*1e5, 0.0232875*1e5,
0.023575*1e5] # Pa
cases_list = ['case_0', 'case_1', 'case_2', 'case_3', 'case_4']
# cases_list = ['case_0']

# Example data point from the inception diagram.
p_inf_exp = (0.43240156 + 0.06) * 1e5 # Pa, at hub level i.e. excluding hydrostatic
# component (small for a small prop anyway)

for cases in cases_list:
    print("-----" + cases + "-----")
    if cases == "case_0": n_exp = 2474.06250/60

```

```

if cases == "case_1": n_exp = 2505.78125/60
if cases == "case_2": n_exp = 2537.50000/60
if cases == "case_3": n_exp = 2569.21875/60
if cases == "case_4": n_exp = 2600.93750/60
for pv_exp in pv_exp_list:
    J_exp = Va_exp / (D * n_exp)
    sigma_exp = (p_inf_exp - pv_exp) / (0.5 * rho_exp * (n_exp*D)**2.)
    print("J_exp = {:.4f}".format(J_exp))
    print("sigma_exp = {:.4f}".format(sigma_exp))

    case = os.getcwd() + "/" + cases + "/case_AFM"

    with open(os.path.join(case, "controls.xml"), "r") as infile:
        controls = infile.read()

    omega = float(getKeyword("magnitude", controls))
    Va = float(getKeyword("referenceVelocity", controls))

    n = omega / (2*np.pi)
    J = Va / (n * D)
    print("J_cfd = {:.4f}".format(J))

    pv = - sigma_exp * (0.5*rho_exp*(n*D)**2)
    print("pv_cfd = {:.1f} kPa".format(pv*1e-3))

    # Read boundary pressures.
    surfaceCgnsFile = h5py.File(os.path.join(case,
"data_coprocessing/boundary.cgns"), mode="r")
    # surfPressure =
surfaceCgnsFile["Base_Surface_Elements"]["BCWall_2"]["CellData"]["Pressure"]["
data"][:]
    surfPressureTotal = pandas.read_csv(os.getcwd() + "/" + cases +
"/case_AFM/data_bladeSurface.csv")['pressure']
    surfPressure = np.array_split(surfPressureTotal,3)

```

```
inception_index = (np.min(surfPressure[2]) + pv) / (0.5*rho_exp*(n*D)**2)
inception_avg = (np.mean(surfPressure[2]) + pv) / (0.5*rho_exp*(n*D)**2)
inception_max = (np.max(surfPressure[2]) + pv) / (0.5*rho_exp*(n*D)**2)
print("Min: {:.4f}".format(inception_index))
print("Average: {:.4f}".format(inception_avg))
print("Max: {:.4f}".format(inception_max))
print("\n")
```

**ANEXO D – run\_nua.py**

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import pandas
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import os
import shutil

from everydaytools import tecPlotIO
from everydaytools import nicePlots

font = {"family": "serif",
        "weight": "normal",
        "size": 16}

matplotlib.rc("font", **font)
matplotlib.rcParams["figure.figsize"] = (9, 6)
matplotlib.style.use("seaborn-colorblind")

# %% Do stuff.
baseCaseDir = "./nuaInputs"
exeName = os.getcwd() + "/nuaInputs/numerical_uncertainty" # TODO replace with
path to .exe on Windows
cases = ['case_0', 'case_1', 'case_2', 'case_3', 'case_4']

# Make a copy of the base case for demonstration purposes. Note how it is placed in
a different subdir
# not to clutter the top level folder.
dest = os.path.join("nuaWorkDir", "nuaCase")
```

try:

```
os.mkdir(dest)
```

except FileExistsError:

```
shutil.rmtree(dest)
```

```
os.mkdir(dest)
```

for case in cases:

```
# Copy files.
```

```
# TODO replace inputs with what comes out of your code here.
```

```
shutil.copy(os.path.join(baseCaseDir, "input_nua_" + case + ".ini"), dest)
```

```
shutil.copy(os.path.join(baseCaseDir, "gridUncertaintyInput_" + case + ".dat"), dest)
```

```
# Run nua.
```

```
currentPath = os.path.abspath(".")
```

```
destPathAbs = os.path.join(currentPath, dest)
```

```
os.chdir(destPathAbs)
```

```
output = os.popen("{} {}/input_nua_{}.ini".format(exeName, destPathAbs,
case)).read()
```

```
os.chdir(currentPath)
```

# This contains the output of the program in the form of a string. Can be checked for errors or what not.

```
print(output)
```

```
# %% Plot results.
```

```
xStr = "h<sub>i</sub>/h<sub>1</sub>"
```

```
gridUncFields = ["KT_mean", "KQ_mean", "eta_mean", "Cpn_avr"]
```

```
# read data
```

```
udata = {}
```

```
for f in gridUncFields:
```

```
    fname = "tec_gridStudy_RSM_{}".format(f)
```

```
    fname = fname[:38] # Truncate file name length because lazy Fortran.
```

```
    udata[f] = tecPlotIO.readTecPlot(os.path.join(destPathAbs, fname+".dat"))
```

```

# Plot each field
fieldLabels = dict(zip(gridUncFields, ["K_T", "K_Q", "eta", "C_{PNaverage}"]))

for f in gridUncFields:

    # ---
    # Without a separate axes for uncertainty
    fig, ax = nicePlots.niceFig([], [], "h_i/h_1 [-]", "{}".format(fieldLabels[f]), f)
    lns = []
    lns += ax.plot(udata[f]["Data"][xStr], udata[f]["Data"].values[:, 1], "o", mew=2,
mfc="None", ms=9, mec="r")
    ax.plot(udata[f]["Fit"][xStr], udata[f]["Fit"].values[:, 1], "-", lw=2, c="r")
    ax.errorbar(udata[f]["Data"][xStr], udata[f]["Data"].values[:, 1], yerr=udata[f]["Error
bar"]["U"],
                lw=2, ecolor="r", fmt="None")

    for i in udata[f]["Data"].index:
        if (-1.)**i > 0:
            va = "bottom"
        else:
            va = "top"
        ax.text(udata[f]["Data"].loc[i, xStr], udata[f]["Data"].values[i, 1] + udata[f]["Error
bar"].loc[i, "U"]*(-1.)**i,
                "${:.1f}\\%$".format(np.abs(udata[f]["Error
bar"].loc[i,
"U"]/udata[f]["Data"].values[i, 1]*100.)),
                size=18, ha="center", va=va, color="r")

    if len(lns) > 1:
        ax.legend(lns, [l.get_label() for l in lns], prop={"size": 18}, loc="upper center",
ncol=10)

```

## ANEXO E – postProcessor\_LHS.py

```

#!/usr/bin/python
# VERSION 1.0
import os,sys, pickle, math, pandas
import numpy as np
from scipy.interpolate import interp1d
sys.path.append('../././python')
import generateData as gdata

def getOutput_dtc_v031(dirMonitors):
    Ns = 40000 #number of samples
    Dp = 0.15
    rho = 998.16
    advanceVelocity = 3.625
    params = gdata.getParameters(dirMonitors + '/params.in')
    n = params['[[RPM]]']/60.
    pv = params['[[PV]]']
    J = advanceVelocity/(n*Dp)
    arr_KT = np.nan;
    arr_KQ = np.nan;
    arr_KQ_s = np.nan;
    arr_Cav_index = np.nan;
    dataF = gdata.getData(dirMonitors + "/monitors/forces_blades.dat",
headerLine=1, startColumn=1)
    dataM = gdata.getData(dirMonitors + "/monitors/moments_blades.dat",
headerLine=1, startColumn=1)
    dataP = gdata.getData(dirMonitors + "/monitors/scalar_pressure.dat",
headerLine=1, startColumn=1)
    if (np.isnan(arr_KT)).all(): arr_KT = -
1.0/(rho*n**2.0*Dp**4.0)*np.array(dataF['ForceTotalX'][-Ns:])
    else: arr_KT += -1.0/(rho*n**2.0*Dp**4.0)*np.array(dataF['ForceTotalX'][-Ns:])

```

```

        if (np.isnan(arr_KQ)).all(): arr_KQ =
1.0/(rho*n**2.0*Dp**5.0)*np.array(dataM['MomentTotalX'][-Ns:])
        else: arr_KQ += 1.0/(rho*n**2.0*Dp**5.0)*np.array(dataM['MomentTotalX'][-
Ns:])
        if (np.isnan(arr_KQ_s)).all(): arr_KQ_s =
1.0/(rho*n**2.0*Dp**5.0)*np.array(dataM['MomentFrictX'][-Ns:])
        else: arr_KQ_s += 1.0/(rho*n**2.0*Dp**5.0)*np.array(dataM['MomentFrictX'][-
Ns:])
        if (np.isnan(arr_Cav_index)).all(): arr_Cav_index =
1.0/(0.5*rho*(n**Dp)**2.0)*(np.array(dataP['PressureAverage'][-Ns:]) + pv)
        else: arr_Cav_index +=
1.0/(0.5*rho*(n**Dp)**2.0)*(np.array(dataP['PressureAverage'][-Ns:]) + pv)
        arr_eta = J*np.divide(arr_KT,arr_KQ)/(2*math.pi)
        dataToOutput = {'KT_mean' : np.mean(arr_KT),
                        'KQ_mean' : np.mean(arr_KQ),
                        'eta_mean' : np.mean(arr_eta),
                        'Cav_index' : np.mean(arr_Cav_index),
                        }
        return dataToOutput
#####
inputs = ['RPM', 'pv']
outputs = ['KT_mean', 'KQ_mean', 'eta_mean', 'Cav_index'];
meshAnalysis = ['j325', 'j400', 'j475', 'j550', 'j625', 'j700']
for mesh in meshAnalysis:
    print("-----" + mesh + "-----")
    case = 'dtc_v03' + mesh + 'J80_S02'
    listDir = os.listdir(os.getcwd() + '/cases/' + case)
    validDir = [x for _,x in sorted(zip([int(nameFolder.split('.')[1]) for nameFolder in listDir
if nameFolder.split('.')[0] == 'run'],[listPureDir for listPureDir in listDir if
listPureDir.split('.')[0] == 'run']))];
    sys.stdout.write('[postProcessor] ' + case + ' : number of runs=' + str(len(validDir)) +
\n')
    data = {}
    tabularResults = np.empty([len(validDir), 1 + len(inputs) + len(outputs)])

```

```

tabularResults[:] = np.nan
for i in range(0,len(validDir)):
    sys.stdout.write('[postProcessor] ' + case + ' : ' + validDir[i] + '\t(' + str(i+1) + '/' +
str(len(validDir)) + ')\n')
    tabularResults[i,0] = validDir[i].split('.')[0]
    inputDic = gdata.getParameters(os.getcwd() + '/cases/' + case + "/" + validDir[i]
+ '/params.in')
    for j in range(0,len(inputs)): tabularResults[i,j+1] = inputDic[inputs[j]]
    outputDic = getOutput_dtc_v031(os.getcwd() + '/cases/' + case + "/" + validDir[i]
)
    for j in range(0,len(outputs)):
        if outputDic == {}:
            sys.stdout.write('[postProcessor] ' + case + ' : ' + validDir[i] + '\t(' +
str(i+1) + '/' + str(len(validDir)) + '): WARNING: Monitors not found.\n')
        else:
            tabularResults[i,1+len(inputs)+j] = outputDic[outputs[j]]
    data = {'case' : case,
           'inputs': inputs,
           'outputs' : outputs,
           'columnNames': ['index'] + inputs + outputs,
           'tabularResults' : tabularResults
    }
    if not os.path.isdir("cases/"): os.mkdir("cases")
    pickle.dump(data, open('cases/' + case + '.pickle','wb'))
    sys.stdout.write('\n[postProcessor] END. saved in ' + case + '.pickle\n\n')

```