



UNIVERSIDADE
ESTADUAL DE LONDRINA

GUSTAVO TAIJI NAOZUKA

GERAÇÃO E ANÁLISE DE QUALIDADE DE MALHAS
COMPUTACIONAIS EM COORDENADAS
GENERALIZADAS

Londrina
2018

GUSTAVO TAIJI NAOZUKA

GERAÇÃO E ANÁLISE DE QUALIDADE DE MALHAS
COMPUTACIONAIS EM COORDENADAS
GENERALIZADAS

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Alan Salvany Felinto

Londrina
2018

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UEL

Naozuka, Gustavo Taiji.

Geração e Análise de Qualidade de Malhas Computacionais em Coordenadas Generalizadas / Gustavo Taiji Naozuka. - Londrina, 2018.
193 f. : il.

Orientador: Alan Salvany Felinto.

Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual de Londrina, Centro de Ciências Exatas, Programa de Pós-Graduação em Ciência da Computação, 2018.

Inclui bibliografia.

1. Qualidade de malhas - Tese. 2. Métricas de qualidade - Tese. 3. Geração de malhas - Tese. 4. Malhas computacionais - Tese. I. Felinto, Alan Salvany. II. Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação. III. Título.

GUSTAVO TAIJI NAOZUKA

GERAÇÃO E ANÁLISE DE QUALIDADE DE MALHAS
COMPUTACIONAIS EM COORDENADAS GENERALIZADAS

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Mestre em Ciência da Computação.

BANCA EXAMINADORA

Orientador: Prof. Dr. Alan Salvany Felinto
Universidade Estadual de Londrina – UEL

Prof. Dr. André Campos de Moura
Universidade Estadual de Londrina – UEL

Profa. Dra. Tamara Angélica Baldo
Universidade Estadual do Norte do Paraná –
UENP

Londrina, 22 de fevereiro de 2018.

*Dedico este trabalho à minha família, à
minha namorada e aos meus amigos.*

AGRADECIMENTOS

Em um primeiro momento, gostaria de agradecer à minha família, em especial aos meus pais, Marcelo e Julia, e às minhas irmãs, Elaine e Juliana, por me auxiliarem em alguns trabalhos e estudos e sempre me apoiarem nas escolhas que fiz desde o início da minha vida acadêmica. Agradeço a eles por terem me ajudado a conquistar tudo o que já conquistei, além de outras conquistas que ainda estão por vir.

Agradeço também aos meus amigos de Ciência da Computação, principalmente àqueles que continuaram comigo desde o período da graduação, Ricardo e Saulo, por terem encarado as dificuldades juntos a mim. Ainda, gostaria de agradecer às novas amizades que realizei com os alunos da Pós-Graduação em Matemática Aplicada e Computacional (PGMAC) quando me matriculei em disciplinas ofertadas pelo departamento. Agradeço a todos eles por terem me proporcionado algo muito importante, o trabalho em grupo.

Um agradecimento bastante especial é dado a Bruna, aluna do PGMAC que conheci juntamente com os outros alunos, por todo o amor e carinho que me tem oferecido. Com certeza, sem a presença dela, os fatos que ocorreram seriam muito diferentes, uma vez que ter a conhecido me trouxe felicidade e ânimo para enfrentar a pesquisa e a escrita deste trabalho, assim como as disciplinas que estava realizando no momento. Novamente, agradeço-a por ela existir e ser essa pessoa como ela exatamente é.

Além disso, agradeço aos professores do Departamento de Matemática, Neyva e Eliandro, que me convidaram a fazer parte do grupo de pesquisa enquanto ainda era calouro e me forneceram importantes ensinamentos desde então. Com certeza, grande parte do que continuarei trabalhando estará embasado nesses ensinamentos. Agradeço também ao meu orientador do Departamento de Computação, Alan, o qual me concedeu seus conhecimentos sobre a dissertação de Mestrado.

Finalmente, gostaria de agradecer à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), por meio do convênio 04/2016, pelo suporte financeiro atrelado ao Programa de Pós-Graduação em Ciência da Computação, que me permitiu manter os estudos e apresentar um artigo em um congresso de âmbito internacional.

*“A gente não pode ter tudo
Qual seria a graça do mundo se fosse
assim?
Por isso eu prefiro sorrisos
E os presentes que a vida trouxe para perto
de mim”
(Ana Vilela)*

NAOZUKA, G. T. **Geração e análise de qualidade de malhas computacionais em coordenadas generalizadas**. 2018. 193 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina, 2018.

RESUMO

As equações diferenciais têm sido intensamente utilizadas como ferramentas de análise e descrição em estudos de fenômenos naturais. Por meio de modelos matemáticos, é possível transformar um problema do domínio físico para o domínio computacional. Neste contexto, este trabalho desenvolve um gerador de malhas bidimensionais em coordenadas generalizadas para o domínio, o qual aplica o método *spline* linear parametrizado para a construção do contorno da geometria e o método de diferenças finitas para a solução numérica das equações governantes. O gerador é automatizado e capaz de tratar domínios complexos e, assim, mais realísticos. Entretanto, mesmo tendo essa capacidade, existe a possibilidade de elementos de menor qualidade serem introduzidos na malha computacional. Por esse motivo, este trabalho também objetiva investigar métricas que identificam elementos considerados de menor qualidade. Esses critérios de qualidade e a metodologia empregada constituem o diferencial deste trabalho. Experimentos são efetuados para verificar a eficiência das métricas adotadas, considerando malhas com um bloco, como a geometria cuspide, e multi-blocos, como a geometria de um avião, do corpo humano e de lagos. De acordo com os experimentos, o trabalho permitiu a detecção de elementos de menor qualidade, contribuindo para a realização de uma modelagem adequada de geometrias.

Palavras-chave: Qualidade de malhas. Métricas de qualidade. Geração de malhas. Malhas computacionais. Coordenadas generalizadas.

NAOZUKA, G. T. **Generation and quality analysis of computational meshes in curvilinear coordinates**. 2018. 193 p. Dissertation (Master's Degree in Science in Computer Science) – Universidade Estadual de Londrina, Londrina, 2018.

ABSTRACT

Differential equations have been intensively used as tools of analysis and description in studies of natural phenomena. Through mathematical models, it is possible to turn a problem from the physical domain to the computational domain. In this context, we develop a two-dimensional mesh generator in curvilinear coordinates for the domain, which applies the parameterized linear spline method to construct the geometry contour and the finite difference method to numerically solve the governing equations. The generator is automated and able to handle complex and more realistic domains. However, even with this ability, there is possibility that elements of lower quality are introduced into the computational mesh. For this reason, we also aim to investigate metrics that identify elements of lower quality. These quality criteria and the methodology used are the differential of this work. Experiments are performed to verify the efficiency of the adopted metrics, considering meshes with a block, such as the cusp geometry, and multi-blocks, such as the geometry of an airplane, the human body and lakes. According to the experiments, the work allowed to detect elements of lower quality, contributing to accomplish a suitable modeling of geometries.

Keywords: Mesh quality. Quality metrics. Mesh generation. Computational meshes. Curvilinear coordinates.

LISTA DE ILUSTRAÇÕES

Figura 1 – Classificação de malhas quanto ao espaçamento entre os pontos	26
Figura 2 – Classificação de malhas quanto à distribuição espacial dos pontos.....	26
Figura 3 – Representação dos pontos utilizados na discretização pelo método de diferenças finitas	35
Figura 4 – Sistemas de coordenadas	36
Figura 5 – Representação gráfica de um sistema qualquer de coordenadas generalizadas	39
Figura 6 – Exemplo de geração de malha em um domínio qualquer	40
Figura 7 – Rotulagem de índices em malhas computacionais.....	41
Figura 8 – Exemplo de clusterização por <i>k-means</i>	50
Figura 9 – Nomenclaturas adotadas para a definição das arestas e ângulos de um elemento	59
Figura 10 – Legenda de cores associadas à qualidade dos elementos de uma malha computacional.....	67
Figura 11 – Malha semi-trapezoidal.....	72
Figura 12 – Diagrama de dispersão de ϵ_i por $ ZU_i $ da malha semi-trapezoidal.....	79
Figura 13 – Elementos de menor qualidade identificados para a malha semi- trapezoidal	80
Figura 14 – Pontos de fronteira da malha quadrada.....	81
Figura 15 – Malha computacional do quadrado em coordenadas generalizadas.....	81
Figura 16 – Nenhum elemento de menor qualidade identificado para a malha quadrada	82
Figura 17 – Pontos de fronteira da malha retangular.....	82
Figura 18 – Malha computacional do retângulo em coordenadas generalizadas.....	83

Figura 19 – Nenhum elemento de menor qualidade identificado para a malha retangular	84
Figura 20 – Pontos de fronteira da malha com obstáculo	84
Figura 21 – Malha computacional com obstáculo em coordenadas generalizadas	85
Figura 22 – Diagrama de dispersão de ϵ_i por $ ZU_i $ da malha com obstáculo	86
Figura 23 – Elementos de menor qualidade identificados para a malha com obstáculo	87
Figura 24 – Pontos de fronteira da placa de orifício.....	88
Figura 25 – Malha computacional da placa de orifício em coordenadas generalizadas.....	88
Figura 26 – Diagrama de dispersão de ϵ_i por $ ZU_i $ da placa de orifício	89
Figura 27 – Elementos de menor qualidade identificados para a placa de orifício.....	90
Figura 28 – Pontos de fronteira da forma de gancho	91
Figura 29 – Malha computacional da forma de gancho em coordenadas generalizadas.....	91
Figura 30 – Diagrama de dispersão de ϵ_i por $ ZU_i $ da forma de gancho	93
Figura 31 – Elementos de menor qualidade identificados para a forma de gancho	94
Figura 32 – Pontos de fronteira da geometria cúspide	95
Figura 33 – Malha computacional da geometria cúspide em coordenadas generalizadas	96
Figura 34 – Diagrama de dispersão de ϵ_i por $ ZU_i $ da geometria cúspide.....	97
Figura 35 – Elementos de menor qualidade identificados para a geometria cúspide.....	98
Figura 36 – Pontos de fronteira das placas paralelas com círculo imerso.....	99

Figura 37 – Malha computacional das placas paralelas com círculo imerso em coordenadas generalizadas.....	99
Figura 38 – Diagrama de dispersão de ϵ_i por $ \mathcal{E}U_i $ das placas paralelas com círculo imerso.....	101
Figura 39 – Elementos de menor qualidade identificados para as placas paralelas com círculo imerso.....	102
Figura 40 – Pontos de fronteira da malha hexagonal.....	103
Figura 41 – Malha computacional do hexágono em coordenadas generalizadas.....	103
Figura 42 – Diagrama de dispersão de ϵ_i por $ \mathcal{E}U_i $ da malha hexagonal.....	105
Figura 43 – Elementos de menor qualidade identificados para a malha hexagonal.....	105
Figura 44 – Pontos de fronteira do NACA.....	106
Figura 45 – Malha computacional do NACA em coordenadas generalizadas.....	107
Figura 46 – Diagrama de dispersão de ϵ_i por $ \mathcal{E}U_i $ do NACA.....	108
Figura 47 – Elementos de menor qualidade identificados para o NACA	109
Figura 48 – Pontos de fronteira da malha com obstáculo multi-blocos.....	110
Figura 49 – Malha computacional com obstáculo multi-blocos em coordenadas generalizadas	110
Figura 50 – Elementos de menor qualidade identificados para a malha com obstáculo multi-blocos.....	112
Figura 51 – Pontos de fronteira da placa de orifício multi-blocos.....	113
Figura 52 – Malha computacional da placa de orifício multi-blocos em coordenadas generalizadas.....	113
Figura 53 – Elementos de menor qualidade identificados para a placa de orifício multi-blocos	115
Figura 54 – Pontos de fronteira do Lago Igapó I.....	116

Figura 55 – Malha computacional do Lago Igapó I em coordenadas generalizadas .	117
Figura 56 – Diagrama de dispersão de ϵ_i por $ \mathcal{E}U_i $ do Lago Igapó I	118
Figura 57 – Elementos de menor qualidade identificados para o Lago Igapó I	119
Figura 58 – Pontos de fronteira do golfinho	120
Figura 59 – Malha computacional do golfinho em coordenadas generalizadas	121
Figura 60 – Diagrama de dispersão de ϵ_i por $ \mathcal{E}U_i $ do golfinho	123
Figura 61 – Elementos de menor qualidade identificados para o golfinho	124
Figura 62 – Pontos de fronteira da tartaruga	125
Figura 63 – Malha computacional da tartaruga em coordenadas generalizadas	126
Figura 64 – Diagrama de dispersão de ϵ_i por $ \mathcal{E}U_i $ da tartaruga	127
Figura 65 – Elementos de menor qualidade identificados para a tartaruga	129
Figura 66 – Pontos de fronteira do perfil de um rosto	130
Figura 67 – Malha computacional do perfil de um rosto em coordenadas generalizadas	131
Figura 68 – Diagrama de dispersão de ϵ_i por $ \mathcal{E}U_i $ do perfil de um rosto	133
Figura 69 – Elementos de menor qualidade identificados para o perfil de um rosto	134
Figura 70 – Pontos de fronteira do pássaro	135
Figura 71 – Malha computacional do pássaro em coordenadas generalizadas	136
Figura 72 – Diagrama de dispersão de ϵ_i por $ \mathcal{E}U_i $ do pássaro	137
Figura 73 – Elementos de menor qualidade identificados para o pássaro	138
Figura 74 – Pontos de fronteira do cachorro	139
Figura 75 – Malha computacional do cachorro em coordenadas generalizadas	140
Figura 76 – Diagrama de dispersão de ϵ_i por $ \mathcal{E}U_i $ do cachorro	142
Figura 77 – Elementos de menor qualidade identificados para o cachorro	143
Figura 78 – Pontos de fronteira do avião	144

Figura 79 – Malha computacional do avião em coordenadas generalizadas	145
Figura 80 – Diagrama de dispersão de ϵ_i por $ \mathcal{E}U_i $ do avião	147
Figura 81 – Elementos de menor qualidade identificados para o avião	148
Figura 82 – Pontos de fronteira do Lago Luruaco.....	149
Figura 83 – Malha computacional do Lago Luruaco em coordenadas generalizadas.....	150
Figura 84 – Diagrama de dispersão de ϵ_i por $ \mathcal{E}U_i $ do Lago Luruaco.....	152
Figura 85 – Elementos de menor qualidade identificados para o Lago Luruaco	153
Figura 86 – Pontos de fronteira do corpo humano	155
Figura 87 – Malha computacional do corpo humano em coordenadas generalizadas.....	156
Figura 88 – Diagrama de dispersão de ϵ_i por $ \mathcal{E}U_i $ do corpo humano.....	158
Figura 89 – Elementos de menor qualidade identificados para o corpo humano.....	159
Figura 90 – Pontos de fronteira do sapo.....	160
Figura 91 – Malha computacional do sapo em coordenadas generalizadas	162
Figura 92 – Diagrama de dispersão de ϵ_i por $ \mathcal{E}U_i $ do sapo	164
Figura 93 – Elementos de menor qualidade identificados para o sapo	165

LISTA DE TABELAS

Tabela 1 –	Coordenadas dos nós da malha semi-trapezoidal	72
Tabela 2 –	Tamanho das arestas, razões entre arestas e ângulos internos do elemento 1 da malha semi-trapezoidal	73
Tabela 3 –	Tamanho das arestas, razões entre arestas e ângulos internos do elemento 2 da malha semi-trapezoidal	73
Tabela 4 –	Tamanho das arestas, razões entre arestas e ângulos internos do elemento 3 da malha semi-trapezoidal	73
Tabela 5 –	Tamanho das arestas, razões entre arestas e ângulos internos do elemento 4 da malha semi-trapezoidal	74
Tabela 6 –	Tamanho das arestas, razões entre arestas e ângulos internos do elemento 5 da malha semi-trapezoidal	74
Tabela 7 –	Tamanho das arestas, razões entre arestas e ângulos internos do elemento 6 da malha semi-trapezoidal	74
Tabela 8 –	Tamanho das arestas, razões entre arestas e ângulos internos do elemento 7 da malha semi-trapezoidal	74
Tabela 9 –	Tamanho das arestas, razões entre arestas e ângulos internos do elemento 8 da malha semi-trapezoidal	74
Tabela 10 –	Tamanho das arestas, razões entre arestas e ângulos internos do elemento 9 da malha semi-trapezoidal	75
Tabela 11 –	Tamanho das arestas, razões entre arestas e ângulos internos do elemento 10 da malha semi-trapezoidal	75
Tabela 12 –	Tamanho das arestas, razões entre arestas e ângulos internos do elemento 11 da malha semi-trapezoidal	75
Tabela 13 –	Tamanho das arestas, razões entre arestas e ângulos internos do elemento 12 da malha semi-trapezoidal	75

Tabela 14 – Tamanho das arestas, razões entre arestas e ângulos internos do elemento 13 da malha semi-trapezoidal	75
Tabela 15 – Tamanho das arestas, razões entre arestas e ângulos internos do elemento 14 da malha semi-trapezoidal	76
Tabela 16 – Tamanho das arestas, razões entre arestas e ângulos internos do elemento 15 da malha semi-trapezoidal	76
Tabela 17 – Área, perímetro e coeficiente de compacidade dos elementos da malha semi-trapezoidal.....	76
Tabela 18 – Conjunto total de elementos e suas métricas de qualidade da malha semi-trapezoidal.....	77
Tabela 19 – Valores mínimo e máximo das métricas de qualidade da malha semitrapezoidal.....	77
Tabela 20 – Conjunto total de elementos e suas métricas de qualidade normalizadas da malha semi-trapezoidal.....	78
Tabela 21 – Conjunto total de elementos e valores de q da malha semi-trapezoidal.....	79
Tabela 22 – Quantidades de elementos dos resultados obtidos para a malha semitrapezoidal.....	80
Tabela 23 – Valores mínimo e máximo das métricas de qualidade da malha com obstáculo.....	85
Tabela 24 – Quantidades de elementos dos resultados obtidos para a malha com obstáculo.....	87
Tabela 25 – Valores mínimo e máximo das métricas de qualidade da placa de orifício	89
Tabela 26 – Quantidades de elementos dos resultados obtidos para a placa de orifício	90

Tabela 27 – Valores mínimo e máximo das métricas de qualidade da forma de gancho	92
Tabela 28 – Quantidades de elementos dos resultados obtidos para a forma de gancho	95
Tabela 29 – Valores mínimo e máximo das métricas de qualidade da geometria cúspide	96
Tabela 30 – Quantidades de elementos dos resultados obtidos para a geometria cúspide.....	98
Tabela 31 – Número de partições nas direções x e y para cada bloco das placas paralelas com círculo imerso.....	99
Tabela 32 – Número de partições nas direções ξ e η para cada bloco das placas paralelas com círculo imerso.....	99
Tabela 33 – Valores mínimo e máximo das métricas de qualidade das placas paralelas com círculo imerso.....	100
Tabela 34 – Quantidades de elementos dos resultados obtidos para as placas paralelas com círculo imerso.....	102
Tabela 35 – Número de partições nas direções x e y para cada bloco da malha hexagonal.....	103
Tabela 36 – Número de partições nas direções ξ e η para cada bloco da malha hexagonal.....	104
Tabela 37 – Valores mínimo e máximo das métricas de qualidade da malha hexagonal.....	104
Tabela 38 – Quantidades de elementos dos resultados obtidos para a malha hexagonal.....	106
Tabela 39 – Número de partições nas direções x e y para cada bloco do NACA.....	107

Tabela 40 – Número de partições nas direções ξ e η para cada bloco do NACA.....	107
Tabela 41 – Valores mínimo e máximo das métricas de qualidade do NACA.....	108
Tabela 42 – Quantidades de elementos dos resultados obtidos para o NACA	109
Tabela 43 – Número de partições nas direções x e y para cada bloco da malha com obstáculo multi-blocos	110
Tabela 44 – Número de partições nas direções ξ e η para cada bloco da malha com obstáculo multi-blocos	111
Tabela 45 – Valores mínimo e máximo das métricas de qualidade da malha com obstáculo multi-blocos	111
Tabela 46 – Quantidades de elementos dos resultados obtidos para a malha com obstáculo multi-blocos	112
Tabela 47 – Número de partições nas direções x e y para cada bloco da placa de orifício multi-blocos	113
Tabela 48 – Número de partições nas direções ξ e η para cada bloco da placa de orifício multi-blocos	114
Tabela 49 – Valores mínimo e máximo das métricas de qualidade da placa de orifício multi-blocos.....	114
Tabela 50 – Quantidades de elementos dos resultados obtidos para a placa de orifício multi-blocos.....	115
Tabela 51 – Número de partições nas direções x e y para cada bloco do Lago Igapó I.....	116
Tabela 52 – Número de partições nas direções ξ e η para cada bloco do Lago Igapó I.....	117
Tabela 53 – Valores mínimo e máximo das métricas de qualidade do Lago Igapó I.....	118

Tabela 54 – Quantidades de elementos dos resultados obtidos para o Lago Igapó I.....	120
Tabela 55 – Número de partições nas direções x e y para cada bloco do golfinho.....	121
Tabela 56 – Número de partições nas direções ξ e η para cada bloco do golfinho.....	121
Tabela 57 – Valores mínimo e máximo das métricas de qualidade do golfinho.....	122
Tabela 58 – Quantidades de elementos dos resultados obtidos para o.....	124
Tabela 59 – Número de partições nas direções x e y para cada bloco da tartaruga	125
Tabela 60 – Número de partições nas direções ξ e η para cada bloco da tartaruga	126
Tabela 61 – Valores mínimo e máximo das métricas de qualidade da tartaruga	127
Tabela 62 – Quantidades de elementos dos resultados obtidos para a tartaruga	130
Tabela 63 – Número de partições nas direções x e y para cada bloco do perfil de um rosto	131
Tabela 64 – Número de partições nas direções ξ e η para cada bloco do perfil de um rosto	132
Tabela 65 – Valores mínimo e máximo das métricas de qualidade do perfil de um rosto	132
Tabela 66 – Quantidades de elementos dos resultados obtidos para o perfil de um rosto	134
Tabela 67 – Número de partições nas direções x e y para cada bloco do pássaro.....	135
Tabela 68 – Número de partições nas direções ξ e η para cada bloco do pássaro.....	136

Tabela 69 – Valores mínimo e máximo das métricas de qualidade do pássaro.....	137
Tabela 70 – Quantidades de elementos dos resultados obtidos para o pássaro	139
Tabela 71 – Número de partições nas direções x e y para cada bloco do cachorro.....	140
Tabela 72 – Número de partições nas direções ξ e η para cada bloco do cachorro.....	141
Tabela 73 – Valores mínimo e máximo das métricas de qualidade do cachorro.....	141
Tabela 74 – Quantidades de elementos dos resultados obtidos para o cachorro	143
Tabela 75 – Número de partições nas direções x e y para cada bloco do avião	145
Tabela 76 – Número de partições nas direções ξ e η para cada bloco do avião.....	146
Tabela 77 – Valores mínimo e máximo das métricas de qualidade do avião.....	146
Tabela 78 – Quantidades de elementos dos resultados obtidos para o avião.....	149
Tabela 79 – Número de partições nas direções x e y para cada bloco do Lago Luruaco	150
Tabela 80 – Número de partições nas direções ξ e η para cada bloco do Lago Luruaco	151
Tabela 81 – Valores mínimo e máximo das métricas de qualidade do Lago Luruaco	151
Tabela 82 – Quantidades de elementos dos resultados obtidos para o Lago Luruaco	154
Tabela 83 – Número de partições nas direções x e y para cada bloco do corpo humano.....	155
Tabela 84 – Número de partições nas direções ξ e η para cada bloco do corpo humano.....	157
Tabela 85 – Valores mínimo e máximo das métricas de qualidade do corpo humano	158

Tabela 86 – Quantidades de elementos dos resultados obtidos para o corpo humano	159
Tabela 87 – Número de partições nas direções \mathcal{X} e \mathcal{Y} para cada bloco do sapo.....	161
Tabela 88 – Número de partições nas direções ξ e η para cada bloco do sapo.....	162
Tabela 89 – Valores mínimo e máximo das métricas de qualidade do sapo	163
Tabela 90 – Quantidades de elementos dos resultados obtidos para o sapo.....	165

LISTA DE ALGORITMOS

1	Método iterativo de Gauss-Jacobi.....	45
2	Método iterativo de Gauss-Seidel.....	45
3	Clusterização por <i>k-means</i>	51
4	Gerador de malhas bidimensionais em coordenadas generalizadas	57
5	Gerador de malhas bidimensionais em coordenadas generalizadas envolvendo multi-blocos	58
6	Cálculo das métricas de qualidade de malhas em coordenadas generalizadas	63
7	Identificação dos elementos de menor qualidade de malhas em coordenadas generalizadas.....	68
8	Identificação dos elementos de menor qualidade de malhas em coordenadas generalizadas - Parte	169
9	Identificação dos elementos de menor qualidade de malhas em coordenadas generalizadas - Parte.....	270

LISTA DE SÍMBOLOS

$S_p(x)$	<i>Spline</i> de grau p
$S_1(x)$	<i>Spline</i> linear
$SLP_x(t)$ e $SLP_y(t)$	<i>Spline</i> linear parametrizado
h	Espaçamento na direção x
k	Espaçamento na direção y
(x, y)	Sistema de coordenadas cartesianas
(ξ, η)	Sistema de coordenadas generalizadas
$\Delta\xi$	Dimensão dos volumes elementares na direção ξ
$\Delta\eta$	Dimensão dos volumes elementares na direção η
d^t	Diferencial no domínio transformado
d^f	Diferencial no domínio físico
A e B	Matrizes de transformação entre os domínios físico e transformado
J	Jacobiano da transformação
S	Área de um elemento no plano físico
Γ_1	Fronteira esquerda
Γ_2	Fronteira superior
Γ_3	Fronteira direita
Γ_4	Fronteira inferior
M	Número de partições na direção ξ
N	Número de partições na direção η
P e Q	Termos fontes das equações governantes para a geração de malhas
n_i	Número total de linhas na direção ξ
n_j	Número total de linhas na direção η
ϕ	Coordenada genérica

P	Ponto cardeal central
E	Ponto cardeal leste
W	Ponto cardeal oeste
N	Ponto cardeal norte
S	Ponto cardeal sul
NW	Ponto cardeal noroeste
SW	Ponto cardeal sudoeste
SE	Ponto cardeal sudeste
NE	Ponto cardeal nordeste
A	Matriz dos coeficientes
x	Vetor de variáveis
b	Vetor constante
$x^{(0)}$	Aproximação inicial
\bar{x}	Solução aproximada de um sistema linear
ε	Precisão
IT	Número máximo de iterações
X	Conjunto de dados
C	Centros
k	Quantidade de centros C
ϕ	Distância quadrática total
G	<i>Clusters</i>
$p_x^{\Gamma_1\Gamma_3}$	Peso sobre as fronteiras esquerda/direita em x
$p_x^{\Gamma_2\Gamma_4}$	Peso sobre as fronteiras superior/inferior em x
$p_y^{\Gamma_1\Gamma_3}$	Peso sobre as fronteiras esquerda/direita em y
$p_y^{\Gamma_2\Gamma_4}$	Peso sobre as fronteiras superior/inferior em y
$r_{i,j}$	Razões entre o tamanho das arestas de um elemento

r_{\square}	Razão entre duas arestas de um quadrado
d	Distância entre dois vértices
$A(x_A, y_A)$ e $B(x_B, y_B)$	Vértices conectados de um elemento
d_i	Tamanho das arestas de um elemento
r	Razão entre cada tamanho de aresta com os demais
α_i	Ângulos internos de um elemento
α_{\square}	Ângulo interno de um quadrado
\vec{u} e \vec{v}	Vetores de mesma origem representando duas arestas de um elemento
\overline{AB} e \overline{AC}	Arestas de um elemento
$\ \vec{u}\ $	Comprimento do vetor \vec{u}
$\ \vec{v}\ $	Comprimento do vetor \vec{v}
c	Coefficiente de compacidade de um elemento
P	Perímetro de um elemento
d_{\square}	Tamanho da aresta de um quadrado
c_{\square}	Coefficiente de compacidade de um quadrado
r	Razão entre arestas com maior disparidade em relação a $r_{\square} = 1$
α	Ângulo interno com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad
\bar{p}	Porcentagem desejada de elementos de menor qualidade
U	Conjunto total de elementos e suas métricas de qualidade
U'	Conjunto total de elementos cujas métricas de qualidade existem e são finitas
s_R^2	Variância das razões entre arestas
s_A^2	Variância dos ângulos internos
s_C^2	Variância dos coeficientes de compacidade
M	Representação de um dos conjuntos R , A ou C
$ M $	Quantidade de elementos pertencentes ao conjunto M

m_k	Valor da métrica de qualidade M
\bar{m}	Média dos valores da métrica de qualidade M
m_{\min}	Valor mínimo da métrica de qualidade M
m_{\max}	Valor máximo da métrica de qualidade M
m_{norm_k}	Valor normalizado da métrica de qualidade M
M_{norm}	Representação de um dos conjuntos normalizados R_{norm} , A_{norm} ou C_{norm}
m_{\square}	Valor das métricas avaliadas sobre um quadrado
$m_{\text{norm}_{\square}}$	Valor normalizado das métricas avaliadas sobre um quadrado
$s_{M_{\text{norm}}}^2$	Variância das métricas de qualidade normalizadas
ϵ_{\min}	Tolerância mínima
ϵ_{\max}	Tolerância máxima
ϵ	Tolerância
E_r	Elementos de menor qualidade em relação à razão entre arestas
E_{α}	Elementos de menor qualidade em relação ao ângulo interno
E_c	Elementos de menor qualidade em relação ao coeficiente de compactidade
E_m	Representação de um dos conjuntos E_r , E_{α} ou E_c
E_{\cup}	União entre os conjuntos E_r , E_{α} e E_c
$ E_{\cup} $	Quantidade de elementos pertencentes ao conjunto E_{\cup}
c_1 e c_2	Coefficientes da função de regressão não linear exponencial
$\bar{\epsilon}$	Tolerância desejada de elementos de menor qualidade
$ U' $	Quantidade de elementos pertencentes ao conjunto U'
$\overline{E_r}$	Elementos de menor qualidade em relação à razão entre arestas, considerando $\bar{\epsilon}$
$\overline{E_{\alpha}}$	Elementos de menor qualidade em relação ao ângulo interno, considerando $\bar{\epsilon}$
$\overline{E_c}$	Elementos de menor qualidade em relação ao coeficiente de compactidade, considerando $\bar{\epsilon}$

$\overline{E_{\cup}}$	União entre os conjuntos $\overline{E_r}$, $\overline{E_{\alpha}}$ e $\overline{E_c}$
$U' - \overline{E_{\cup}}$	Elementos de maior qualidade
$\overline{E_{r,\alpha}}$	Intersecção entre os conjuntos $\overline{E_r}$ e $\overline{E_{\alpha}}$
$\overline{E_{r,c}}$	Intersecção entre os conjuntos $\overline{E_r}$ e $\overline{E_c}$
$\overline{E_{\alpha,c}}$	Intersecção entre os conjuntos $\overline{E_{\alpha}}$ e $\overline{E_c}$
$\overline{E_{\cap}}$	Intersecção entre os conjuntos $\overline{E_r}$, $\overline{E_{\alpha}}$ e $\overline{E_c}$
q	Valor atribuído para cada categoria de elementos
$U - U'$	Conjunto de elementos em que $r = +\infty$ e $\alpha = \nexists$

LISTA DE ABREVIATURAS E SIGLAS

PGMAC	Pós-Graduação em Matemática Aplicada e Computacional
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
EDO	Equação Diferencial Ordinária
EDP	Equação Diferencial Parcial
ET	Erro de truncamento
IA	Inteligência Artificial
GA	<i>Genetic Algorithm</i>
ACO	<i>Ant Colony Optimization</i>
PSO	<i>Particle Swarm Optimization</i>

SUMÁRIO

1	INTRODUÇÃO.....	25
1.1	Objetivos.....	28
1.2	Organização do Trabalho.....	28
2	FUNDAMENTAÇÃO TEÓRICO-METODOLÓGICA.....	29
2.1	Método de Interpolação.....	29
2.1.1	Funções <i>Spline</i>	30
2.2	Equações Diferenciais.....	30
2.3	Solução Numérica de Equações Diferenciais.....	32
2.3.1	Método de Diferenças Finitas.....	33
2.4	Sistema de Coordenadas Generalizadas.....	35
2.4.1	Conceitos Básicos.....	36
2.4.2	Métricas de Transformação.....	37
2.4.3	Geração de Malhas Bidimensionais.....	39
2.5	Resolução de Sistemas Lineares.....	42
2.5.1	Método de Gauss-Seidel.....	43
2.6	Ajuste de Curvas.....	46
2.6.1	Método de Mínimos Quadrados.....	47
2.7	Algoritmo de Clusterização <i>k-means</i>	49
2.8	Trabalhos Correlatos.....	51
3	MATERIAIS E MÉTODOS.....	54
3.1	Aquisição de Pontos de Fronteira.....	54
3.2	Gerador de Malhas em Coordenadas Generalizadas.....	55
3.2.1	Descrição do Algoritmo Desenvolvido.....	56

3.2.2	Técnica Multi-blocos.....	57
3.3	Análise de Qualidade de Malhas em Coordenadas Generalizadas.....	58
3.3.1	Métricas de Qualidade.....	58
3.3.2	Descrição dos Algoritmos Desenvolvidos.....	60
3.3.2.1	Cálculo das Métricas de Qualidade.....	61
3.3.2.2	Identificação dos Elementos de Menor Qualidade.....	64
4	RESULTADOS E DISCUSSÕES.....	71
4.1	Malha Semi-trapezoidal.....	71
4.2	Malha Quadrada.....	80
4.3	Malha Retangular.....	82
4.4	Malha com Obstáculo.....	84
4.5	Placa de Orifício.....	88
4.6	Forma de Gancho.....	91
4.7	Geometria Cúspide.....	95
4.8	Placas Paralelas com Círculo Imerso.....	98
4.9	Malha Hexagonal.....	103
4.10	NACA 64A 010.....	106
4.11	Malha com Obstáculo Multi-blocos.....	110
4.12	Placa de Orifício Multi-blocos.....	113
4.13	Lago Igapó I.....	116
4.14	Golfinho.....	120
4.15	Tartaruga.....	125
4.16	Perfil de um Rosto 2D.....	130
4.17	Pássaro.....	135
4.18	Cachorro.....	139
4.19	Avião.....	144

4.20	Lago Luruaco.....	149
4.21	Corpo Humano.....	154
4.22	Sapo	160
4.23	Análise dos Resultados.....	166
5	CONCLUSÕES.....	168
	REFERÊNCIAS.....	170
	APÊNDICES	175
	APÊNDICE A – CÓDIGOS COMPUTACIONAIS.....	176
	A.1 Gerador de Malhas em Coordenadas Generalizadas	176
	A.2 Análise de Qualidade de Malhas em Coordenadas Generalizadas	180
	A.2.1 Cálculo das Métricas de Qualidade	180
	A.2.2 Identificação dos Elementos de Menor Qualidade.....	182
	Trabalhos Publicados pelo Autor.....	192
	Patentes Desenvolvidas pelo Autor	193

1 INTRODUÇÃO

A modelagem e simulação de fenômenos naturais por meio da manipulação de equações diferenciais constitui uma importante ferramenta para a Ciência. Contudo, devido ao fato de que a grande maioria das equações diferenciais não possui solução analítica, é necessária a aplicação de métodos numéricos para sua resolução. Uma das características mais importantes dos métodos numéricos é a necessidade de malhas computacionais para descrever a geometria do meio que está sendo investigado.

A malha consiste no domínio físico discretizado, ou seja, no domínio físico representado através de um conjunto de pontos [1, 2]. O processo de discretização constitui-se como um passo indispensável, pelo fato de que esses métodos de resolução não permitem a obtenção de uma solução para todos os pontos do domínio [3].

Uma malha computacional é formada por um agrupamento de células, limitadas por arestas denominadas de faces e contendo vértices chamados de nós [3, 4]. Dependendo do tipo de malha, essas células podem possuir diferentes formas, tais como triângulos ou quadriláteros quando se consideram duas dimensões, e tetraedros ou hexaedros, em três dimensões [2].

As malhas possuem determinadas classificações dependendo do parâmetro empregado, como segue [3]:

- de acordo com o espaçamento entre os pontos: malhas uniformes ou não uniformes;
- de acordo com a distribuição espacial dos pontos: malhas estruturadas ou não estruturadas.

A Figura 1 ilustra, por meio de um exemplo, as principais diferenças entre as malhas uniformes, Figura 1a, e as não uniformes, Figura 1b.

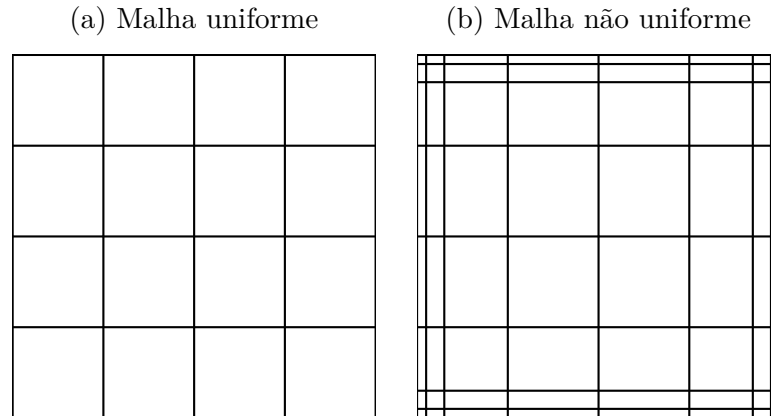


Figura 1 – Classificação de malhas quanto ao espaçamento entre os pontos

Analogamente, a Figura 2 demonstra as principais diferenças entre as malhas estruturadas, Figura 2a, e as não estruturadas, Figura 2b.

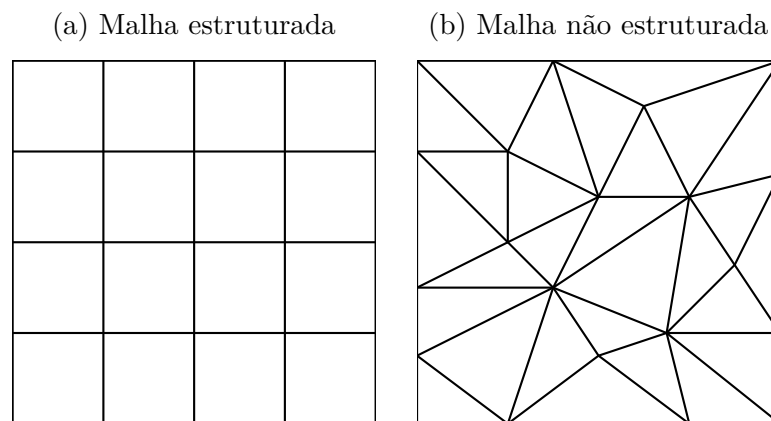


Figura 2 – Classificação de malhas quanto à distribuição espacial dos pontos

Existe também o conceito de malhas híbridas ou estruturadas em blocos, as quais consistem em malhas constituídas por uma pequena quantidade de elementos estruturados combinados através de um padrão não estruturado [2]. Entretanto, sua geração para quaisquer geometrias constitui ainda um obstáculo [2].

As malhas estruturadas podem ser definidas como aquelas cujos elementos podem ser facilmente ordenados, enquanto que as malhas não estruturadas não apresentam um padrão de ordenação. Ainda, as malhas não estruturadas possuem a característica de serem mais versáteis que as estruturadas [2, 5], o que permite a sua adaptabilidade às geometrias consideradas complexas. Em contrapartida, como apresentam dificuldade de ordenação dos volumes elementares, ou seja, as células da malha, implicam diretamente no consumo adicional de memória, devido ao aumento das dimensões da matriz computacional. Assim,

em virtude da facilidade de manipulação dos elementos da malha, considera-se, neste trabalho, a utilização de malhas estruturadas.

Para representar as estruturas físicas, ou seja, o domínio a ser estudado, em um ambiente computacional, a utilização das estruturas de organização de dados mais simples, tais como matrizes e vetores, tende a não ser capaz de descrever, de forma realística, o objeto de estudo. Dessa maneira, diferentes técnicas têm sido empregadas como alternativas para a representação de domínios, tais como malhas estruturadas e não estruturadas, generalizadas ou híbridas, possuindo vantagens e desvantagens quanto à flexibilidade e à capacidade representativa dos objetos em estudo [5, 6, 7, 8, 9, 10].

No que diz respeito ao sistema de coordenadas, o sistema de coordenadas cartesianas é comumente adotado para a representação do domínio físico do problema, devido a sua simplicidade. No entanto, para problemas que envolvem geometrias complexas, esse sistema de coordenadas acarreta uma má adequação da fronteira do problema, pelo fato de que o domínio físico não coincide com o domínio da malha computacional.

O sistema de coordenadas generalizadas, também denominado como sistema de coordenadas curvilíneas, cujas malhas produzidas são do tipo estruturadas, possibilita também a adaptabilidade presente em malhas não estruturadas, permitindo a representação computacional se adequar à geometria do problema, a qual pode ser complexa ou não [5, 6, 8, 10, 11]. Outro fator que se deve levar em consideração é a possibilidade de se concentrar pontos da malha conforme a necessidade, proporcionando um estudo mais aprofundado em uma certa localização dentro da geometria.

Assim, buscando uma forma representativa e simplificada quanto à estruturação computacional e partindo de uma mudança de sistema de coordenadas, é possível converter um domínio descrito de forma complexa em um conjunto de dados facilmente manipulável [5]. No âmbito de problemas matemáticos, pode-se transformar uma geometria qualquer, descrita em coordenadas cartesianas, em coordenadas generalizadas, permitindo melhor adequação na modelagem computacional [5, 6, 7, 8, 10, 11, 12, 13].

Entretanto, a geração de malhas em coordenadas generalizadas, assim como em qualquer outro sistema de coordenadas, pode introduzir elementos considerados de menor qualidade nas malhas, dependendo da complexidade da geometria e do refinamento empregado. Um elemento dessa categoria é aquele sobre o qual a resolução de alguma equação diferencial descrevendo um fenômeno qualquer produz resultados insatisfatórios [14, 15, 16], tais como instabilidade numérica e inadequação à realidade.

Dessa forma, pode-se enunciar que, quanto maior a regularidade da geometria do elemento, melhor será a qualidade da malha que o contém [17, 18]. Uma vez que os elementos pertencentes às malhas em coordenadas generalizadas são quadriláteros, é importante que a geometria do elemento se aproxime de um quadrado. Para tal, após o

processo de geração, é fundamental identificar os elementos de menor qualidade, a fim de que, posteriormente, possam ser restaurados.

O processo de identificação dos elementos de menor qualidade pode ser realizado por meio da aplicação de métricas de qualidade, as quais atribuem valores numéricos aos elementos, os ordenam de acordo com os valores calculados e os classificam entre maior e menor qualidade. Para este trabalho, três critérios são utilizados para a classificação de tais elementos, os quais permitem averiguar a similaridade a um quadrado.

1.1 Objetivos

Este trabalho tem como objetivo o desenvolvimento de um gerador de malhas bidimensionais, utilizando a teoria de métricas de transformação em coordenadas generalizadas e produzindo um domínio mapeado para manipulações matemáticas, o qual descreve um objeto físico qualquer, a partir de um conjunto finito de pontos.

Visando um melhor mapeamento da localização dos pontos pertencentes à malha, três critérios para análise de qualidade são propostos e aplicados: as razões entre o tamanho de cada aresta do elemento com os demais, os ângulos internos e a compacidade do elemento. Todos os critérios verificam a similaridade dos elementos da malha computacional a um quadrado, elemento considerado ideal, identificando assim aqueles considerados de menor qualidade.

1.2 Organização do Trabalho

Este trabalho encontra-se estruturado como apresentado a seguir.

- No Capítulo 2, explicam-se os principais conceitos associados aos processos de geração de malhas em coordenadas generalizadas e de análise de qualidade. Dedicam-se também uma seção para a revisão de trabalhos relacionados, apresentando as respectivas metodologias empregadas.
- No Capítulo 3, demonstram-se os procedimentos metodológicos aplicados nos processos de aquisição de pontos de fronteira, geração de malhas bidimensionais em coordenadas generalizadas e análise de qualidade das malhas produzidas.
- No Capítulo 4, executam-se diferentes experimentos envolvendo geometrias variadas e utilizando malhas computacionais com um bloco e multi-blocos, como forma de validação dos algoritmos desenvolvidos e dos procedimentos metodológicos aplicados. Ainda, discutem-se os resultados obtidos para cada experimento.
- No Capítulo 5, apontam-se as considerações finais, incluindo as contribuições esperadas deste trabalho e possíveis trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICO-METODOLÓGICA

Neste capítulo, apresentam-se os conceitos mais relevantes para facilitar o entendimento da metodologia adotada nos processos de geração de malhas em coordenadas generalizadas e de análise de qualidade.

Expondo uma visão geral dos conceitos associados à geração de malhas, na Seção 2.1, explica-se o método de interpolação para a construção das fronteiras da geometria em estudo; na Seção 2.2, define-se o conceito de equações diferenciais e suas classificações, como auxílio à compreensão das equações governantes para a geração de malhas; na Seção 2.3, esclarece-se a ideia de métodos numéricos para a solução de equações diferenciais, salientando o método de diferenças finitas; na Seção 2.4, introduz-se o sistema de coordenadas generalizadas e apresentam-se as métricas de transformação e o processo de geração de malhas bidimensionais; e, na Seção 2.5, aborda-se o método para a resolução de sistemas lineares, necessário para o desenvolvimento computacional das equações governantes.

No que se refere aos conceitos associados à análise de qualidade de malhas, na Seção 2.6, explica-se o problema do ajuste de curvas, destacando o método de mínimos quadrados; e, na Seção 2.7, define-se o conceito de clusterização, enfatizando o algoritmo *k-means*. Ambas as seções são importantes para os algoritmos desenvolvidos nesse processo.

Finalmente, na Seção 2.8, realiza-se uma revisão bibliográfica de trabalhos relacionados ao processo de análise de qualidade de malhas computacionais, acompanhados com as respectivas metodologias empregadas.

2.1 Método de Interpolação

O método de interpolação consiste em aproximar uma função $f(x)$ por meio de outra função $g(x)$, satisfazendo determinadas propriedades. Esse método é aplicável a diferentes situações, quando, por exemplo, é necessário encontrar valores da função em pontos não tabelados, utilizando somente os pontos conhecidos [19]. A esse conjunto de pontos conhecidos, denominam-se nós da interpolação.

Expressando o método matematicamente, dada uma função $f(x)$, são conhecidos os valores da função em $n + 1$ nós da interpolação: x_0, x_1, \dots, x_n . A função interpoladora

$g(x)$ que aproxima $f(x)$ [19] é tal que

$$\left\{ \begin{array}{l} g(x_0) = f(x_0) \\ g(x_1) = f(x_1) \\ \vdots \\ g(x_n) = f(x_n). \end{array} \right. \quad (2.1)$$

Em outras palavras, o Sistema de Equações (2.1) enuncia que os nós da interpolação $(x_i, f(x_i))$, em que $i = 0, \dots, n$, devem obrigatoriamente pertencer à função interpoladora $g(x)$.

A aproximação de uma função, em determinados casos, pode produzir resultados insatisfatórios, isto é, o erro existente entre a função interpoladora com respeito à função original pode ser suficientemente grande. Com a finalidade de minimizar esse erro, surge o conceito de funções *spline*, o qual é apresentado na Subseção 2.1.1.

2.1.1 Funções *Spline*

As funções *spline* consistem em uma alternativa, na qual a interpolação é realizada em grupos de poucos pontos. Essa interpolação deve obedecer condições, as quais estabelecem que a função de aproximação deve ser contínua, assim como suas derivadas, até uma certa ordem. Uma função $S_p(x)$, dita como *spline* de grau p , possui a função interpoladora $g(x)$ descrita como um polinômio de grau p [19].

Durante o processo de geração de malhas em coordenadas generalizadas, uma importante etapa necessita ser realizada, referindo-se à definição das fronteiras da geometria em estudo. Nessa etapa, a técnica de *spline* linear $S_1(x)$ é empregada, cuja função é representada na forma paramétrica, por meio das funções $SLP_x(t)$ e $SLP_y(t)$ dadas por [5, 11, 19]

$$SLP_x(t) = x_i \left(\frac{t_{i+1} - t}{t_{i+1} - t_i} \right) + x_{i+1} \left(\frac{t - t_i}{t_{i+1} - t_i} \right); \quad (2.2)$$

$$SLP_y(t) = y_i \left(\frac{t_{i+1} - t}{t_{i+1} - t_i} \right) + y_{i+1} \left(\frac{t - t_i}{t_{i+1} - t_i} \right); \quad \forall t \in [t_i, t_{i+1}], \quad (2.3)$$

em que $i = 0, \dots, n - 1$, levando em consideração que existem $n + 1$ nós da interpolação.

Com a finalidade de facilitar a compreensão das equações governantes para a geração de malhas em coordenadas generalizadas, a Seção 2.2 explica o conceito de equações diferenciais e suas classificações.

2.2 Equações Diferenciais

No contexto da Matemática, equação diferencial é a denominação dada às equações que contêm derivadas [20]. As equações diferenciais são extremamente importantes

em diferentes áreas do conhecimento pelo fato de que possibilitam determinar o comportamento de diversos fenômenos físicos, químicos, biológicos, entre outros. Em outras palavras, uma grande variedade de aplicações é proporcionada pelo estudo e resolução das equações diferenciais [20, 21]. Alguns exemplos de aplicações modeladas por essas equações são: problemas envolvendo o movimento de fluidos, o fluxo de corrente elétrica em circuitos, a dissipação de calor em objetos sólidos, a propagação e detecção de ondas sísmicas e a variação da taxa populacional [20].

Nas equações diferenciais, uma função constitui a incógnita do problema, o que permite classificá-las da seguinte maneira [20]:

- Equações Diferenciais Ordinárias (EDOs): equações compostas por derivadas simples, isto é, a função desconhecida depende de uma única variável independente;
- Equações Diferenciais Parciais (EDPs): equações compostas por derivadas parciais, ou seja, a função desconhecida depende de duas ou mais variáveis independentes.

Apesar de possuírem soluções que descrevem comportamentos bastante diferentes, uma EDO consiste basicamente em um caso especial de EDP [21].

Em muitas situações, outra questão que também deve ser levada em consideração está no fato de que podem existir mais de uma função desconhecida. Nesses casos, é necessário um sistema de equações diferenciais para a sua resolução [20].

Um conceito muito importante quando se trata de equações diferenciais é a ordem, a qual corresponde à ordem da maior derivada presente na equação [20, 22]. Esse conceito possibilita formular, de maneira matemática, as EDOs e as EDPs.

Uma EDO de ordem n para uma função desconhecida $y(x)$ é descrita por [21]

$$F(x, y(x), y'(x), \dots, y^{(n)}(x)) = 0. \quad (2.4)$$

Uma EDP, por exemplo, de segunda ordem para uma função desconhecida $u(x, y)$, isto é, dependente de duas variáveis, é expressa por [21]

$$F(x, y, u, u_x, u_y, u_{xx}, u_{xy}, u_{yy}) = 0. \quad (2.5)$$

As EDPs de ordem n com mais variáveis independentes possuem formulações análogas. Ainda, nas Equações (2.4) e (2.5), F é uma função conhecida, a qual representa a equação diferencial, dado o problema.

Outra classificação possível para as equações diferenciais pode ser realizada segundo a linearidade, ou seja, uma equação diferencial é dita linear se a função F é linear em relação à função desconhecida e suas derivadas [20, 21]. Atualmente, os métodos existentes para a resolução de equações lineares são bastante satisfatórios, porém ainda há

grande dificuldade em se resolver equações não lineares [20]. Por esse motivo, quando o fenômeno pode ser representado por uma equação linear, é comum realizar um processo denominado de linearização, consistindo na aproximação de uma equação não linear por uma linear [20].

De modo particular, as EDPs possuem também uma classificação específica. Considere reescrever a Equação (2.5) como combinação linear da função desconhecida $u(x, y)$ e de suas derivadas, de acordo com [23]

$$a \frac{\partial^2 u}{\partial x^2} + b \frac{\partial^2 u}{\partial x \partial y} + c \frac{\partial^2 u}{\partial y^2} + d \frac{\partial u}{\partial x} + e \frac{\partial u}{\partial y} + fu + g = 0; \quad (2.6)$$

em que a, b, c, d, e, f e g podem ser funções dependentes de x, y e u . Sabendo-se que $\Delta = b^2 - 4ac$, a EDP, representada na Equação (2.6), pode ser classificada como segue [23]:

- elíptica: quando $\Delta < 0$;
- parabólica: quando $\Delta = 0$;
- hiperbólica: quando $\Delta > 0$.

Tendo em vista que boa parte das equações diferenciais não possuem soluções analíticas, isto é, soluções exatas, a obtenção de uma solução numérica apropriada é fundamental. Para tal fim, é necessário o desenvolvimento de um modelo matemático capaz de representar o problema em questão [3]. Esse modelo é constituído pelas equações e pelo domínio físico do problema, os quais devem estar adequadamente expressos [3].

As equações diferenciais aproximadas são encontradas através de métodos numéricos de resolução, os quais são especificados na Seção 2.3, destacando o método de diferenças finitas. Além disso, o domínio físico, representado na forma de malhas computacionais descritas no sistema de coordenadas generalizadas, é definido de maneira mais detalhada na Seção 2.4.

2.3 Solução Numérica de Equações Diferenciais

Para se solucionar numericamente uma equação diferencial, assim como o domínio físico, também é necessário discretizá-la, realizando manipulações algébricas para expressá-la na forma de operações aritméticas facilmente interpretadas pelo computador [3, 23]. Essa discretização é realizada através de metodologias, dentre as quais as mais conhecidas atualmente são: diferenças finitas, elementos finitos, volumes finitos e os métodos espectrais [24].

Em virtude da sua simplicidade, aplica-se o método de diferenças finitas como estratégia de discretização das equações governantes para a geração de malhas bidimensionais em coordenadas generalizadas neste trabalho. Esse método e suas principais equações de aproximação são abordados na Subseção 2.3.1.

2.3.1 Método de Diferenças Finitas

O método numérico por diferenças finitas realiza o processo de discretização das equações diferenciais em função dos pontos (nós) da malha computacional, nos quais se deve representar a sua solução aproximada [23]. Essa aproximação pode ser encontrada por meio de técnicas, das quais as mais empregadas são: a interpolação polinomial e a expansão em série de Taylor [1]. Neste trabalho, adota-se a expansão em série de Taylor para as aproximações.

Tendo em vista que as equações governantes para a geração de malhas bidimensionais em coordenadas generalizadas são EDPs dependentes de duas variáveis, as equações de aproximação são desenvolvidas considerando uma função $u = u(x, y)$ contínua e com derivadas contínuas. Consideram-se também um espaçamento h na direção x e um espaçamento k na direção y . Dessa maneira, a expansão de u em série de Taylor na direção x é descrita por [23]

$$u(x + h, y) = u(x, y) + h \frac{\partial u(x, y)}{\partial x} + \frac{h^2}{2} \frac{\partial^2 u(x, y)}{\partial x^2} + \frac{h^3}{6} \frac{\partial^3 u(x, y)}{\partial x^3} + \dots \quad (2.7)$$

Da mesma forma,

$$u(x - h, y) = u(x, y) - h \frac{\partial u(x, y)}{\partial x} + \frac{h^2}{2} \frac{\partial^2 u(x, y)}{\partial x^2} - \frac{h^3}{6} \frac{\partial^3 u(x, y)}{\partial x^3} + \dots \quad (2.8)$$

A partir da Equação (2.7), desprezando-se a expressão $\frac{h^2}{2} \frac{\partial^2 u(x, y)}{\partial x^2} + \frac{h^3}{6} \frac{\partial^3 u(x, y)}{\partial x^3} + \dots$, tem-se que $u(x + h, y) \approx u(x, y) + h \frac{\partial u(x, y)}{\partial x}$. Dessa forma, a aproximação progressiva (ou avançada) para a primeira derivada na direção x é denotada por [23]

$$\frac{\partial u(x, y)}{\partial x} = \frac{u(x + h, y) - u(x, y)}{h} + O(h); \quad (2.9)$$

em que $O(h) = \frac{h}{2} \frac{\partial^2 u(x, y)}{\partial x^2} + \frac{h^2}{6} \frac{\partial^3 u(x, y)}{\partial x^3} + \dots$ é o erro de truncamento (ET) da série.

Analogamente, realizando o mesmo procedimento para a Equação (2.8), a aproximação regressiva (ou atrasada) para a primeira derivada na direção x é expressa por [23]

$$\frac{\partial u(x, y)}{\partial x} = \frac{u(x, y) - u(x - h, y)}{h} + O(h); \quad (2.10)$$

em que ET é $O(h) = \frac{h}{2} \frac{\partial^2 u(x, y)}{\partial x^2} - \frac{h^2}{6} \frac{\partial^3 u(x, y)}{\partial x^3} + \dots$

Além disso, subtraindo a Equação (2.7) de (2.8) e realizando algumas manipulações matemáticas, obtém-se a aproximação de diferença central para a primeira derivada na direção x , dada por [23]

$$\frac{\partial u(x, y)}{\partial x} = \frac{u(x + h, y) - u(x - h, y)}{2h} + O(h^2); \quad (2.11)$$

em que ET é $O(h^2) = \frac{h^2}{6} \frac{\partial^3 u(x, y)}{\partial x^3} + \frac{h^4}{120} \frac{\partial^5 u(x, y)}{\partial x^5} + \dots$

De maneira análoga, é possível calcular aproximações para derivadas de ordens superiores. Adicionando as Equações (2.7) e (2.8), obtém-se, por meio de alguns desenvolvimentos matemáticos, a aproximação de diferença central para a segunda derivada na direção x , expressada por [23]

$$\frac{\partial^2 u(x, y)}{\partial x^2} = \frac{u(x + h, y) - 2u(x, y) + u(x - h, y)}{h^2} + O(h^2); \quad (2.12)$$

em que a notação $O(h^2) = \frac{h^2}{12} \frac{\partial^4 u(x, y)}{\partial x^4} + \dots$ indica que ET é de segunda ordem de acurácia.

Vale ressaltar ainda que as aproximações demonstradas pelas Equações (2.9) - (2.12) são semelhantes às discretizações efetuadas na direção y , considerando o espaçamento k . Assim sendo, as Equações (2.9) - (2.12) e suas formulações derivadas englobam grande parte das formas de discretização que são utilizadas pelo método de diferenças finitas. Para que o computador possa interpretá-las, a implementação das expressões é comumente realizada pela manipulação de matrizes ou vetores.

A fim de facilitar a compreensão, considere, por exemplo, uma matriz contendo os pontos da malha pertencentes a um meio bidimensional e a função $u = u(x, y)$ previamente definida. Dessa forma, os valores de $u(x - h, y)$, $u(x, y)$ e $u(x + h, y)$ presentes nas aproximações correspondem respectivamente a $u_{i-1, j}$, $u_{i, j}$ e $u_{i+1, j}$, onde i, j equivale à posição na matriz sobre a qual a derivada de u está sendo aproximada, como se pode verificar na Figura 3.

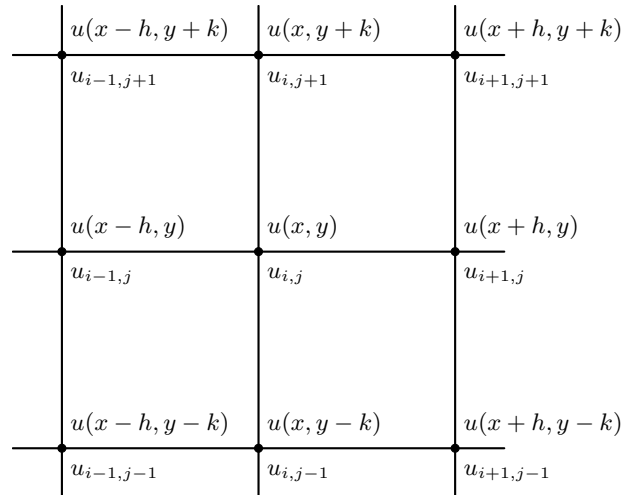


Figura 3 – Representação dos pontos utilizados na discretização pelo método de diferenças finitas

2.4 Sistema de Coordenadas Generalizadas

Para que uma metodologia computacional possa ser aplicada para determinar a solução numérica de equações diferenciais, é necessário discretizar o domínio [3], ou seja, construir uma malha computacional que possa representar a geometria estudada e, assim, obter os valores de interesse.

A discretização do domínio pode ser tratada segundo uma malha estruturada ou não estruturada. As malhas não estruturadas são mais adaptáveis que as estruturadas [2, 5], principalmente quando problemas que envolvem geometrias complexas são investigados. Contudo, a grande desvantagem está relacionada à dificuldade de ordenação dos elementos, o que implica na variação do tamanho das diagonais da matriz dos coeficientes e no custo adicional do uso de memória, dificultando a aplicação de métodos numéricos para o cálculo da solução de sistemas lineares. Sendo assim, a discretização do domínio neste trabalho é realizada utilizando malhas estruturadas.

No que diz respeito ao sistema de coordenadas, em geral, o domínio do problema é discretizado segundo o sistema de coordenadas cartesianas, por sua simplicidade. Porém, para problemas com geometrias complexas, convém adotar outro sistema, devido ao fato de que o sistema de coordenadas cartesianas leva a uma má adequação da fronteira, pois o domínio físico nem sempre coincide com o domínio da malha. Para contornar a má formação na fronteira de domínios complexos, é considerado o sistema de coordenadas generalizadas, no qual a malha computacional coincide com a geometria do problema, e o tratamento computacional torna-se mais adequado [5, 6, 8, 10, 11]. Outras razões que levam ao uso do sistema de coordenadas generalizadas no processo de discretização referem-se à menor dificuldade na programação de códigos computacionais para resolver problemas complexos e à facilidade em desenvolver metodologias genéricas.

Na Subseção 2.4.1, alguns conceitos básicos e importantes sobre a teoria de coordenadas generalizadas são introduzidos.

2.4.1 Conceitos Básicos

O mapeamento de geometrias irregulares ou regulares descritas em coordenadas cartesianas (x, y) , Figura 4a, pode ser realizado para geometrias regulares representadas em coordenadas generalizadas (ξ, η) , Figura 4b. As notações ξ e η indicam as linhas coordenadas que compõem o sistema de coordenadas generalizadas. Nesse procedimento, a transformação em geral é numérica entre os sistemas cartesiano e generalizado. O sistema cartesiano (x, y) é chamado de domínio físico, e o sistema generalizado (ξ, η) é chamado de domínio transformado ou computacional.

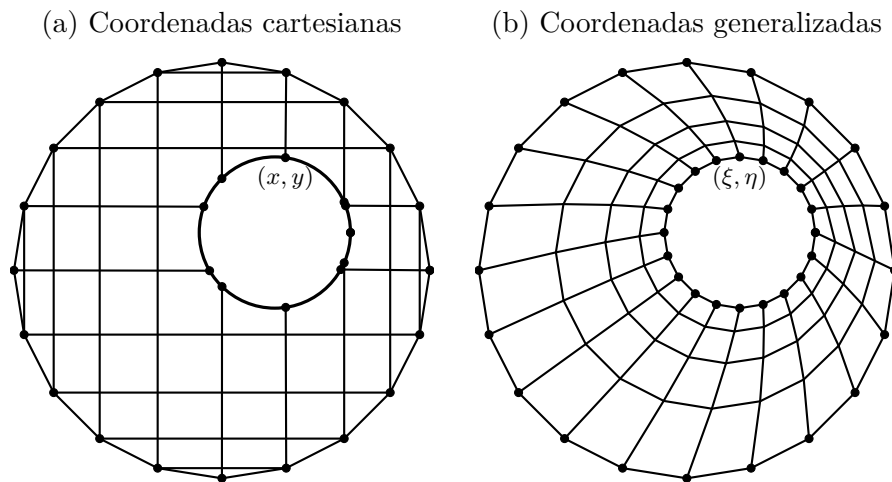


Figura 4 – Sistemas de coordenadas

Uma vez que o domínio transformado é regular, por conveniência, assume-se uma normalização dos volumes elementares com dimensões unitárias, ou seja, $\Delta\xi = \Delta\eta = 1$. Dessa forma, mesmo que no plano físico as linhas coordenadas assumam espaçamentos arbitrários, no plano computacional as dimensões são fixas.

As métricas de transformação são encarregadas de realizar as compensações decorrentes da mudança do sistema coordenado, tanto na malha computacional quanto nas equações diferenciais do modelo [5]. A questão essencial consiste em determinar os pontos (x_i, y_j) das intersecções de todas as linhas ξ e η , permitindo a geração da malha. Em posse desses dados, é possível obter as relações matemáticas da transformação numericamente. Assim, a Subseção 2.4.2 apresenta o desenvolvimento matemático para a definição das métricas de transformação.

2.4.2 Métricas de Transformação

A transformação de uma geometria não trivial descrita no sistema de coordenadas cartesianas para o sistema de coordenadas generalizadas envolve encontrar métricas, ou relações matemáticas, que possam representar os dados transformados de forma fiel ao plano original. Dessa maneira, para a obtenção do sistema (ξ, η) , utilizam-se equações geradoras do tipo [5]

$$\xi = \xi(x, y); \quad (2.13)$$

$$\eta = \eta(x, y). \quad (2.14)$$

As métricas de transformação, a partir de diferenciais das Equações (2.13) e (2.14), são

$$d\xi = \xi_x dx + \xi_y dy; \quad (2.15)$$

$$d\eta = \eta_x dx + \eta_y dy; \quad (2.16)$$

ou, na forma matricial,

$$\begin{pmatrix} d\xi \\ d\eta \end{pmatrix} = \begin{pmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{pmatrix} \cdot \begin{pmatrix} dx \\ dy \end{pmatrix}; \quad (2.17)$$

que pode também ser escrita como

$$d^t = Ad^f; \quad (2.18)$$

em que d^t e d^f representam, respectivamente, as diferenciais no domínio transformado e físico, enquanto A é a matriz de transformação entre os domínios.

Partindo do pressuposto de que é possível encontrar uma representação no sistema de coordenadas cartesianas para um modelo descrito em coordenadas generalizadas, admite-se a existência da inversa das Equações (2.13) e (2.14), logo [5]

$$x = x(\xi, \eta); \quad (2.19)$$

$$y = y(\xi, \eta); \quad (2.20)$$

onde se pode igualmente afirmar, a partir de diferenciais das Equações (2.19) e (2.20), que

$$dx = x_\xi d\xi + x_\eta d\eta; \quad (2.21)$$

$$dy = y_\xi d\xi + y_\eta d\eta; \quad (2.22)$$

ou, na forma matricial,

$$\begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{pmatrix} \cdot \begin{pmatrix} d\xi \\ d\eta \end{pmatrix}; \quad (2.23)$$

que pode também ser escrita como

$$d^f = B d^t; \quad (2.24)$$

sendo B a matriz de transformação entre os domínios físico e transformado.

Substituindo a Equação (2.18) em (2.24), obtém-se $d^f = B A d^f$, de modo que $BA = I$ ou equivalentemente $A = B^{-1}$. Dessa forma, a matriz A torna-se

$$\begin{pmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{pmatrix} = \begin{pmatrix} \frac{y_\eta}{x_\xi y_\eta - x_\eta y_\xi} & \frac{-x_\eta}{x_\xi y_\eta - x_\eta y_\xi} \\ \frac{-y_\xi}{x_\xi y_\eta - x_\eta y_\xi} & \frac{x_\xi}{x_\xi y_\eta - x_\eta y_\xi} \end{pmatrix}. \quad (2.25)$$

A partir da igualdade matricial dada na Equação (2.25), encontra-se

$$\begin{aligned} \xi_x &= J y_\eta; & \eta_x &= -J y_\xi; \\ \xi_y &= -J x_\eta; & \eta_y &= J x_\xi; \end{aligned} \quad (2.26)$$

onde $J = (x_\xi y_\eta - x_\eta y_\xi)^{-1}$ é denominado de Jacobiano da transformação [5, 10]. Logo, as métricas da função inversa, obtidas das relações dadas na Equação (2.26), são

$$\begin{aligned} x_\xi &= \frac{1}{J} \eta_y; & y_\xi &= -\frac{1}{J} \eta_x; \\ x_\eta &= -\frac{1}{J} \xi_y; & y_\eta &= \frac{1}{J} \xi_x. \end{aligned} \quad (2.27)$$

Pode-se ainda interpretar graficamente os resultados observando a Figura 5, na qual se verifica, da Equação (2.23), que $dx = x_\xi d\xi + x_\eta d\eta$ e $dy = y_\xi d\xi + y_\eta d\eta$, assim

$$\begin{aligned} a &= x_\xi \Delta\xi \quad (\text{ao longo de } \overline{AB} \quad d\eta \approx \Delta\eta = 0); \\ b &= y_\xi \Delta\xi \quad (\text{ao longo de } \overline{AB} \quad d\eta \approx \Delta\eta = 0); \\ c &= x_\eta \Delta\eta \quad (\text{ao longo de } \overline{BC} \quad d\xi \approx \Delta\xi = 0); \\ d &= y_\eta \Delta\eta \quad (\text{ao longo de } \overline{BC} \quad d\xi \approx \Delta\xi = 0). \end{aligned}$$

Dessa maneira,

$$dL_\xi = \sqrt{x_\xi^2 + y_\xi^2} \Delta\xi; \quad (2.28)$$

$$dL_\eta = \sqrt{x_\eta^2 + y_\eta^2} \Delta\eta; \quad (2.29)$$

que vetorialmente podem ser escritas como

$$\overrightarrow{dL_\xi} = x_\xi \Delta\xi \vec{i} + y_\xi \Delta\xi \vec{j}; \quad (2.30)$$

$$\overrightarrow{dL_\eta} = x_\eta \Delta\eta \vec{i} + y_\eta \Delta\eta \vec{j}. \quad (2.31)$$

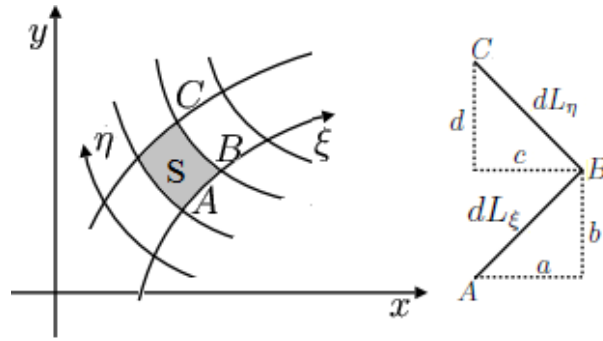


Figura 5 – Representação gráfica de um sistema qualquer de coordenadas generalizadas. (Fonte: 5)

A partir das relações dadas nas Equações (2.30) e (2.31), tem-se que a área formada pelos vetores \overrightarrow{dL}_ξ e \overrightarrow{dL}_η é calculada por $S = |\overrightarrow{dL}_\xi \times \overrightarrow{dL}_\eta| = (x_\xi y_\eta - x_\eta y_\xi) \Delta\xi \Delta\eta$, ou ainda $S = J^{-1} \Delta\xi \Delta\eta$, ou

$$S = \frac{1}{J}; \quad (2.32)$$

pois $\Delta\xi = \Delta\eta = 1$. Desse modo, tem-se que a área de um elemento no plano físico é o inverso do Jacobiano da transformação e, da Equação (2.32), obtém-se que a razão entre as áreas dos planos físico e transformado é $\frac{1}{J}$.

O resultado encontrado na Equação (2.32) possui uma implicação direta e importante no estudo da transformação do sistema de coordenadas: para a garantia de que o método numérico aplicado represente com precisão a solução numérica, é necessário que J seja não nulo, caso contrário a representação do plano físico não será possível [5, 8].

Apresentadas as métricas de transformação, a Subseção 2.4.3 demonstra a teoria necessária para a geração de malhas bidimensionais, imprescindível para o desenvolvimento do algoritmo proposto neste trabalho.

2.4.3 Geração de Malhas Bidimensionais

De acordo com a literatura, existem diferentes metodologias para a geração de malhas computacionais [5], tais como: elípticas, hiperbólicas, parabólicas, geométricas e algébricas. Neste trabalho, opta-se pela utilização de equações elípticas lineares como método de geração de malhas bidimensionais, uma vez que as soluções não geram Jacobiano nulo, e ainda duas linhas coordenadas ξ ou η nunca se interceptam [5]. Além disso, a simplicidade dessas equações foi um outro fator que motivou a escolha realizada.

Antes de enunciar as equações governantes para a geração de malhas bidimensionais, é importante ilustrar um exemplo de geração de malha em um domínio qualquer, o qual é ilustrado na Figura 6.

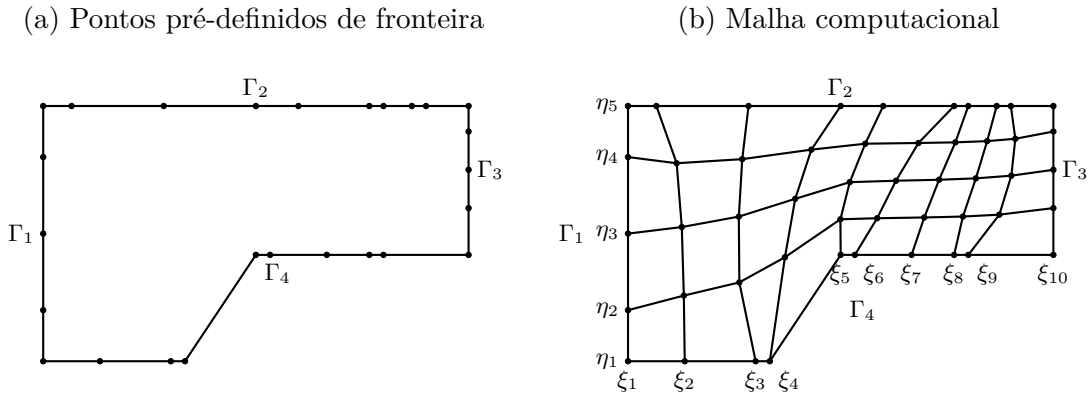


Figura 6 – Exemplo de geração de malha em um domínio qualquer. (Fonte: 5)

Assim, com base na Figura 6, escolhem-se as equações de Poisson como equações governantes para a geração de malhas bidimensionais em um domínio, ou seja,

$$\nabla^2 \xi = P(\xi, \eta); \quad (2.33)$$

$$\nabla^2 \eta = Q(\xi, \eta); \quad (2.34)$$

cujas condições de fronteira do tipo *Dirichlet* são expressadas por $\xi = \xi_1$ em Γ_1 (fronteira esquerda), $\xi = \xi_{M+1}$ em Γ_3 (fronteira direita); $\eta = \eta_1$ em Γ_4 (fronteira inferior) e $\eta = \eta_{N+1}$ em Γ_2 (fronteira superior). Particularmente, na Figura 6, utilizam-se $M = 9$ e $N = 4$, onde M é o número de partições na direção ξ , e N , o número de partições na direção η .

Nas Equações (2.33) e (2.34), P e Q são os termos fontes, os quais podem determinar a concentração de pontos de malha conforme a necessidade [5, 11]. Essas funções são escolhidas como [5, 11]

$$P(\xi, \eta) = - \sum_{j=1}^{n_j} a_j \text{sign}(\xi - \xi_j) e^{-c_j |\xi - \xi_j|} - \sum_{i=1}^{n_i} b_i \text{sign}(\xi - \xi_i) e^{-d_i \sqrt{(\xi - \xi_i)^2 + (\eta - \eta_i)^2}}; \quad (2.35)$$

$$Q(\xi, \eta) = - \sum_{j=1}^{n_j} a_j \text{sign}(\eta - \eta_j) e^{-c_j |\eta - \eta_j|} - \sum_{i=1}^{n_i} b_i \text{sign}(\eta - \eta_i) e^{-d_i \sqrt{(\xi - \xi_i)^2 + (\eta - \eta_i)^2}}; \quad (2.36)$$

onde n_i e n_j , presentes nos somatórios, são os números totais de linhas nas direções ξ e η , respectivamente. Os termos a_j , b_i , c_j e d_i são números reais ajustados por meio de experimentação numérica, com a finalidade de atrair linhas ξ e η em direção a ξ_i e η_i [11].

Solucionando as Equações (2.33) e (2.34), em relação a (x, y) , e utilizando as métricas de transformação do sistema (ξ, η) , dadas em (2.26) e (2.27), pode-se gerar as linhas coordenadas nas direções ξ e η , no interior da malha computacional da Figura 6b, como de qualquer outra geometria, por

$$\alpha x_{\xi\xi} + \gamma x_{\eta\eta} - 2\beta x_{\xi\eta} + \frac{1}{J^2} (Px_{\xi} + Qx_{\eta}) = 0; \quad (2.37)$$

$$\alpha y_{\xi\xi} + \gamma y_{\eta\eta} - 2\beta y_{\xi\eta} + \frac{1}{J^2} (Py_{\xi} + Qy_{\eta}) = 0; \quad (2.38)$$

onde $\alpha = x_\eta^2 + y_\eta^2$, $\beta = x_\xi x_\eta + y_\xi y_\eta$ e $\gamma = x_\xi^2 + y_\xi^2$, com x_ξ , x_η , y_ξ e y_η denotando derivadas parciais.

A solução numérica das equações elípticas lineares, (2.37) e (2.38), as quais consistem em um conjunto de equações de geração de malhas para as coordenadas físicas no plano transformado, sujeitas a condições iniciais e de contorno, fornece as linhas ξ e η , que superpostas, geram a malha computacional, como pode ser verificado no exemplo ilustrado na Figura 6b. O leitor interessado nos pormenores de como se obter essas equações, partindo de manipulações matemáticas das equações governantes para a geração de malhas bidimensionais, pode consultar as referências [5, 11, 25].

Verifica-se ainda que as Equações (2.37) e (2.38) podem ser escritas segundo uma coordenada genérica ϕ [5, 11, 25], como

$$\alpha\phi_{\xi\xi} + \gamma\phi_{\eta\eta} - 2\beta\phi_{\xi\eta} + \frac{1}{J^2}(P\phi_\xi + Q\phi_\eta) = 0. \quad (2.39)$$

Para aproximar as derivadas presentes na Equação (2.39), considerando que os nós da malha computacional sejam rotulados pelos pontos cardeais P (centro), E (leste), W (oeste), N (norte), S (sul), NW (noroeste), SW (sudoeste), SE (sudeste) e NE (nordeste), de acordo com a Figura 7, utiliza-se o método de diferenças finitas, explicado na Seção 2.3.1.

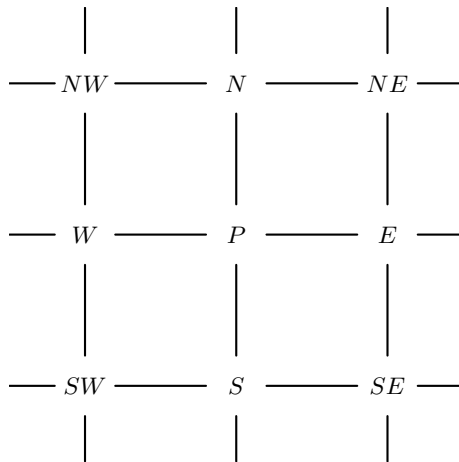


Figura 7 – Rotulagem de índices em malhas computacionais

Dessa maneira, o método de diferenças finitas aproxima os termos das derivadas por diferenças centrais para o ponto P , resultando, da Equação (2.39), em

$$\alpha \left(\frac{\phi|_E - 2\phi|_P + \phi|_W}{\Delta\xi^2} \right) + \gamma \left(\frac{\phi|_N - 2\phi|_P + \phi|_S}{\Delta\eta^2} \right) - 2\beta \left(\frac{\phi|_{NE} - \phi|_{NW} + \phi|_{SW} - \phi|_{SE}}{4\Delta\xi\Delta\eta} \right) + \frac{1}{J^2} \left(P \frac{\phi|_E - \phi|_W}{2\Delta\xi} + Q \frac{\phi|_N - \phi|_S}{2\Delta\eta} \right) = 0; \quad (2.40)$$

onde A é a matriz dos coeficientes, x é o vetor de variáveis, e b é o vetor constante.

Assim, a resolução de um sistema linear consiste em calcular o vetor de variáveis x que satisfaz as m equações do Sistema (2.44) simultaneamente. Os métodos numéricos de resolução podem ser divididos em dois grupos:

- os métodos diretos fornecem a solução exata do sistema linear, caso exista, após um número finito de operações;
- os métodos iterativos produzem, a partir de uma aproximação inicial $x^{(0)}$, uma sequência de vetores $\{x^{(k)}\}$ que converge para uma solução aproximada do sistema linear, caso exista.

Uma vez que o desenvolvimento computacional da solução numérica da equação governante para a geração de malhas em coordenadas generalizadas, Equação (2.42), aplica o método iterativo de Gauss-Seidel, a Subseção 2.5.1 apresenta uma introdução aos métodos iterativos e o funcionamento do método utilizado.

2.5.1 Método de Gauss-Seidel

Para a explicação dos métodos iterativos de resolução de sistemas lineares, considere um sistema linear como representado na Equação (2.44), em que o número de equações é equivalente ao número de variáveis, isto é, $m = n$. Isolando o vetor de variáveis x através da separação pela diagonal principal da matriz dos coeficientes A , esse sistema pode ser convertido em um sistema da forma

$$x = Cx + g \Leftrightarrow \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} 0 & -\frac{a_{12}}{a_{11}} & -\frac{a_{13}}{a_{11}} & \dots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & -\frac{a_{23}}{a_{22}} & \dots & -\frac{a_{2n}}{a_{22}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & -\frac{a_{n3}}{a_{nn}} & \dots & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} + \begin{pmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \\ \vdots \\ \frac{b_n}{a_{nn}} \end{pmatrix}; \quad (2.45)$$

onde C é a matriz $n \times n$, e g , o vetor $n \times 1$ definidos na Equação (2.45).

Utilizando a Equação (2.45), a partir de uma aproximação inicial $x^{(0)}$, obtêm-se consecutivamente os vetores $x^{(1)}, x^{(2)}, \dots$, de modo que a aproximação $x^{(k+1)}$ é calculada por

$$x^{(k+1)} = Cx^{(k)} + g; \quad (2.46)$$

sendo $k = 0, 1, \dots$. A relação recursiva presente na Equação (2.46) pode ser expressa por

um sistema linear

$$\begin{cases} x_1^{(k+1)} = \frac{1}{a_{11}} \left(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \dots - a_{1n}x_n^{(k)} \right); \\ x_2^{(k+1)} = \frac{1}{a_{22}} \left(b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)} - \dots - a_{2n}x_n^{(k)} \right); \\ x_3^{(k+1)} = \frac{1}{a_{33}} \left(b_3 - a_{31}x_1^{(k)} - a_{32}x_2^{(k)} - a_{34}x_4^{(k)} - \dots - a_{3n}x_n^{(k)} \right); \\ \vdots \\ x_n^{(k+1)} = \frac{1}{a_{nn}} \left(b_n - a_{n1}x_1^{(k)} - a_{n2}x_2^{(k)} - \dots - a_{n,n-1}x_{n-1}^{(k)} \right). \end{cases} \quad (2.47)$$

O Sistema de Equações (2.47) é conhecido como método iterativo de Gauss-Jacobi.

Similarmente ao método de Gauss-Jacobi, no método iterativo de Gauss-Seidel, a aproximação $x^{(k+1)}$, dada uma aproximação inicial $x^{(0)}$, é calculada por

$$\begin{cases} x_1^{(k+1)} = \frac{1}{a_{11}} \left(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \dots - a_{1n}x_n^{(k)} \right); \\ x_2^{(k+1)} = \frac{1}{a_{22}} \left(b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} - \dots - a_{2n}x_n^{(k)} \right); \\ x_3^{(k+1)} = \frac{1}{a_{33}} \left(b_3 - a_{31}x_1^{(k+1)} - a_{32}x_2^{(k+1)} - a_{34}x_4^{(k)} - \dots - a_{3n}x_n^{(k)} \right); \\ \vdots \\ x_n^{(k+1)} = \frac{1}{a_{nn}} \left(b_n - a_{n1}x_1^{(k+1)} - a_{n2}x_2^{(k+1)} - \dots - a_{n,n-1}x_{n-1}^{(k+1)} \right). \end{cases} \quad (2.48)$$

Assim, com o objetivo de aprimorar a convergência do método numérico, esse processo iterativo calcula o valor de $x_j^{(k+1)}$ utilizando todos os valores de $x_1^{(k+1)}, \dots, x_{j-1}^{(k+1)}$ previamente calculados e os valores de $x_{j+1}^{(k)}, \dots, x_n^{(k)}$ restantes.

Em ambos os métodos iterativos, se a sequência de aproximações $\{x^{(k)}\}$ é tal que $\lim_{k \rightarrow \infty} x^{(k)} = \bar{x}$, então \bar{x} é uma solução aproximada do sistema linear descrito na Equação (2.44). Em termos computacionais, a atribuição do vetor $x^{(k)}$ como \bar{x} ocorre no momento em que o vetor $x^{(k)}$ estiver suficientemente próximo do vetor $x^{(k-1)}$. Em outras palavras, por meio da distância $d^{(k)}$ entre os dois vetores, dada por

$$d^{(k)} = \max_{1 \leq j \leq n} |x_j^{(k)} - x_j^{(k-1)}|; \quad (2.49)$$

e de uma precisão ε , encontra-se a solução aproximada \bar{x} como sendo $x^{(k)}$ se $d^{(k)} < \varepsilon$, e interrompe-se o processo iterativo. Além disso, outro critério de parada comumente empregado consiste em um número máximo de iterações IT .

Utilizando os critérios de parada baseados em um número máximo de iterações IT e na distância $d^{(k)}$ entre os vetores $x^{(k)}$ e $x^{(k-1)}$, Equação (2.49), os Algoritmos 1 e 2 implementam, respectivamente, os métodos iterativos de Gauss-Jacobi e de Gauss-Seidel, representados nos Sistemas de Equações (2.47) e (2.48).

Algoritmo 1: Método iterativo de Gauss-Jacobi

Entrada: Matriz dos coeficientes A , vetor de constantes b , aproximação inicial $x^{(0)}$, precisão ε e número máximo de iterações IT .

Saída: Vetor de variáveis x .

```

1 início
2   para  $k \leftarrow 0$  até  $IT$  faça
3     para  $j \leftarrow 1$  até  $n$  faça
4        $x_j^{(k+1)} \leftarrow \frac{1}{a_{jj}} \left[ b_j - \sum_{\substack{i=1 \\ i \neq j}}^n a_{ji} x_i^{(k)} \right];$ 
5     fim
6     se  $\max_{1 \leq j \leq n} |x_j^{(k+1)} - x_j^{(k)}| < \varepsilon$  então
7       pare
8     fim
9   fim
10 fim
```

Algoritmo 2: Método iterativo de Gauss-Seidel

Entrada: Matriz dos coeficientes A , vetor de constantes b , aproximação inicial $x^{(0)}$, precisão ε e número máximo de iterações IT .

Saída: Vetor de variáveis x .

```

1 início
2   para  $k \leftarrow 0$  até  $IT$  faça
3     para  $j \leftarrow 1$  até  $n$  faça
4        $x_j^{(k+1)} \leftarrow \frac{1}{a_{jj}} \left[ b_j - \sum_{i=1}^{j-1} a_{ji} x_i^{(k+1)} - \sum_{i=j+1}^n a_{ji} x_i^{(k)} \right];$ 
5     fim
6     se  $\max_{1 \leq j \leq n} |x_j^{(k+1)} - x_j^{(k)}| < \varepsilon$  então
7       pare
8     fim
9   fim
10 fim
```

Portanto, a aplicação do método de Gauss-Seidel para a resolução numérica da equação governante para a geração de malhas em coordenadas generalizadas significa resolver iterativamente a Equação (2.42), verificando a convergência da solução, através

de uma expressão análoga à Equação (2.49) que calcula a distância entre as soluções da iteração atual e anterior.

Esclarecidos os conceitos mais importantes associados ao processo de geração de malhas em coordenadas generalizadas, a Seção 2.6 define o problema do ajuste de curvas, salientando o método de mínimos quadrados. Esse método é aplicado em uma das etapas do processo de análise de qualidade de malhas.

2.6 Ajuste de Curvas

Assim como a Seção 2.5, esta seção também foi escrita fortemente baseada em [19, 26]. Visto que uma das premissas dos métodos de interpolação consiste no fato de que os nós da interpolação devem pertencer à função interpoladora, como mencionado na Seção 2.1, existem casos em que esses métodos não são aconselháveis, como, por exemplo:

- quando é preciso calcular um valor aproximado da função em algum ponto fora do intervalo de tabelamento;
- quando os valores tabelados são resultados de algum experimento ou de alguma pesquisa, podendo conter erros inerentes ao estudo.

Dessa forma, surge a necessidade de ajustar uma função que seja uma aproximação satisfatória para os pontos conhecidos, permitindo extrapolar alguns pontos dentro de uma margem de segurança.

Considere uma tabela de pontos conhecidos $(x_i, f(x_i))$, com $x_i \in [a, b]$ e $i = 1, \dots, m$. O problema do ajuste de curvas aos pontos conhecidos constitui-se em, escolhidas funções $g_j(x)$ contínuas em $[a, b]$, $j = 1, \dots, n$, obter os coeficientes c_j tais que a função $\varphi(x)$, dada por

$$\varphi(x) = c_1g_1(x) + c_2g_2(x) + \dots + c_n g_n(x); \quad (2.50)$$

se aproxime ao máximo de $f(x)$. Nesse caso, a função $\varphi(x)$ é um modelo matemático linear, uma vez que os coeficientes c_j são lineares. Entretanto, é importante salientar que as funções $g_j(x)$ podem ser não lineares com respeito à x .

Uma importante questão associada ao problema do ajuste de curvas no caso discreto refere-se à escolha das funções $g_j(x)$. Essa escolha pode ser realizada com base em fundamentos teóricos do experimento que forneceu a tabela de pontos conhecidos ou na plotagem gráfica desses pontos. Por meio desse gráfico, conhecido como diagrama de dispersão, é possível visualizar a curva com melhor ajuste aos dados.

Escolhidas as funções $g_j(x)$, para o cálculo dos coeficientes c_j , é preciso definir o conceito de proximidade entre as funções $\varphi(x)$ e $f(x)$. Esse conceito pode ser estabelecido

pelo desvio d_k , expressado por

$$d_k = f(x_k) - \varphi(x_k); \quad (2.51)$$

em que $k = 1, \dots, m$. Assim, o desvio d_k necessita ser minimizado para todo k , e uma das maneiras de impor que os desvios sejam mínimos e conseqüentemente obter os coeficientes c_j pode ser dada pelo método de mínimos quadrados, descrito na Subseção 2.6.1.

2.6.1 Método de Mínimos Quadrados

O método de mínimos quadrados consiste em escolher os coeficientes c_j de forma que a soma dos quadrados dos desvios seja mínima. Dessa maneira, os coeficientes c_j que aproximam, ao máximo, $\varphi(x)$ de $f(x)$ são aqueles que minimizam a função $F(c_1, c_2, \dots, c_n)$, representada por

$$F(c_1, c_2, \dots, c_n) = \sum_{k=1}^m d_k^2 = \sum_{k=1}^m (f(x_k) - \varphi(x_k))^2. \quad (2.52)$$

Utilizando conceitos de Cálculo Diferencial, a obtenção de um ponto mínimo de $F(c_1, c_2, \dots, c_n)$ pode ser realizada pelo cálculo dos pontos críticos, isto é, os coeficientes c_j tais que

$$\left. \frac{\partial F}{\partial c_j} \right|_{(c_1, c_2, \dots, c_n)} = 0; \quad (2.53)$$

onde $j = 1, \dots, n$.

Calculando as derivadas parciais da função F em relação a cada coeficiente c_j e impondo a condição presente na Equação (2.53), encontra-se a expressão descrita por

$$\sum_{k=1}^m [f(x_k) - c_1 g_1(x_k) - \dots - c_n g_n(x_k)] \cdot [g_j(x_k)] = 0; \quad (2.54)$$

a qual, reescrita por um sistema linear e realizadas algumas manipulações matemáticas, é dada por

$$\begin{cases} \left[\sum_{k=1}^m g_1(x_k) g_1(x_k) \right] c_1 + \dots + \left[\sum_{k=1}^m g_n(x_k) g_1(x_k) \right] c_n = \sum_{k=1}^m f(x_k) g_1(x_k); \\ \left[\sum_{k=1}^m g_1(x_k) g_2(x_k) \right] c_1 + \dots + \left[\sum_{k=1}^m g_n(x_k) g_2(x_k) \right] c_n = \sum_{k=1}^m f(x_k) g_2(x_k); \\ \vdots \\ \left[\sum_{k=1}^m g_1(x_k) g_n(x_k) \right] c_1 + \dots + \left[\sum_{k=1}^m g_n(x_k) g_n(x_k) \right] c_n = \sum_{k=1}^m f(x_k) g_n(x_k); \end{cases} \quad (2.55)$$

possuindo n equações e n variáveis.

Sabendo que o produto escalar entre dois vetores \vec{u} e \vec{v} de dimensão m é calculado por

$$\langle \vec{u}, \vec{v} \rangle = \sum_{k=1}^m u_k v_k; \quad (2.56)$$

e denotando o Sistema de Equações (2.55) na forma matricial $Ac = b$, tem-se que

$$\begin{aligned} A &= (a_{ij}) = \langle \bar{g}_i, \bar{g}_j \rangle; \\ c &= (c_j); \\ b &= (b_i) = \langle \bar{f}, \bar{g}_i \rangle; \end{aligned} \quad (2.57)$$

em que $i = 1, \dots, n$ e $j = 1, \dots, n$. Na Equação (2.57), $\bar{f} = (f(x_1), f(x_2), \dots, f(x_m))$, $\bar{g}_i = (g_i(x_1), g_i(x_2), \dots, g_i(x_m))$, e, de modo análogo, define-se o vetor \bar{g}_j . Portanto, a solução do sistema linear $Ac = b$ utilizando a Equação (2.57) fornece os coeficientes c_j tais que a função $F(c_1, c_2, \dots, c_n)$ atinge seu valor mínimo, obtendo conseqüentemente a função $\varphi(x)$ representada na Equação (2.50).

No entanto, as funções escolhidas $g_j(x)$ podem ser não lineares nos parâmetros, como é o caso da função $\varphi(x)$ utilizada no decorrer do processo de análise de qualidade dos elementos das malhas. Essa função é expressada por

$$\varphi(x) = c_1 e^{c_2 x}; \quad (2.58)$$

sendo denominada como regressão não linear exponencial.

Para utilizar o método de mínimos quadrados considerando a Equação (2.58), é necessário efetuar uma linearização do problema, a qual pode ser realizada aplicando o logaritmo natural sobre os termos presentes na equação, ou seja,

$$\ln(\varphi(x)) = \ln(c_1 e^{c_2 x}). \quad (2.59)$$

Aplicando propriedades de logaritmos, a Equação (2.59) pode ser reescrita como

$$\ln(\varphi(x)) = \ln(c_1) + c_2 x. \quad (2.60)$$

Finalmente, a Equação (2.60) pode ser dada como

$$\zeta(x) = \gamma + \beta x; \quad (2.61)$$

em que $\zeta(x) = \ln(\varphi(x))$, $\gamma = \ln(c_1)$ e $\beta = c_2$. Assim, a Equação (2.61) constitui um problema linear nos parâmetros γ e β .

Além da linearização do problema, é preciso efetuar a linearização dos dados. Dado um conjunto de pontos conhecidos $(x_i, f(x_i))$, onde $i = 1, \dots, n$, a linearização dos dados aplica o logaritmo natural sobre cada valor de $f(x_i)$, isto é, obtém-se $z(x_i) = \ln(f(x_i))$ para todo i .

Sobre o problema e os dados linearizados, é possível aplicar a resolução pelo método de mínimos quadrados. Para tal, representa-se a Equação (2.61) por

$$\zeta(x) = \gamma g_1(x) + \beta g_2(x); \quad (2.62)$$

em que $g_1(x) = 1$ e $g_2(x) = x$. Dessa maneira, para calcular os coeficientes γ e β da função $\zeta(x)$, é necessário resolver o sistema linear dado por

$$\begin{bmatrix} \langle \bar{g}_1, \bar{g}_1 \rangle & \langle \bar{g}_1, \bar{g}_2 \rangle \\ \langle \bar{g}_2, \bar{g}_1 \rangle & \langle \bar{g}_2, \bar{g}_2 \rangle \end{bmatrix} \cdot \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = \begin{bmatrix} \langle \bar{g}_1, \bar{z} \rangle \\ \langle \bar{g}_2, \bar{z} \rangle \end{bmatrix}; \quad (2.63)$$

onde $\bar{g}_1 = (1 \dots 1)$, $\bar{g}_2 = (x_1 \dots x_n)$ e $\bar{z} = (z(x_1) \dots z(x_n))$ são vetores de n elementos.

Portanto, a partir dos coeficientes γ e β da função $\zeta(x)$ calculados, os coeficientes c_1 e c_2 da função $\varphi(x)$ podem ser encontrados, de acordo com

$$c_1 = e^\gamma; \quad (2.64)$$

$$c_2 = \beta; \quad (2.65)$$

e, assim, a função de regressão não linear exponencial, Equação (2.58), é definida.

Além do método de mínimos quadrados, o processo de análise de qualidade de malhas computacionais aplica o algoritmo de clusterização *k-means* para a classificação dos elementos. O conceito de clusterização e o método utilizado são apresentados na Seção 2.7.

2.7 Algoritmo de Clusterização *k-means*

Clusterização é o nome dado ao processo de particionar ou agrupar um determinado conjunto de pontos em *clusters* disjuntos [27, 28]. Pontos pertencentes a um mesmo *cluster* são semelhantes entre si, e pontos pertencentes a dois *clusters* distintos são diferentes. Esse processo tem sido amplamente estudado em uma variedade de aplicações, tais como redes neurais, inteligência artificial e estatística [27, 28].

Dentre os algoritmos propostos na literatura, o método *k-means* mostrou ser eficaz em produzir resultados satisfatórios de clusterização para diferentes problemas científicos e industriais, destacando-se pela velocidade e pela simplicidade [27, 29]. Esse algoritmo busca minimizar a distância quadrática média entre os pontos pertencentes a um mesmo *cluster* [29]. Assim, dado um inteiro k e um conjunto de dados X com n pontos, o objetivo constitui-se em escolher k centros C , de modo a minimizar a distância quadrática total ϕ , expressada por

$$\phi = \sum_{x \in X} \min_{c \in C} \|x - c\|^2; \quad (2.66)$$

entre cada ponto e seu centro mais próximo. A partir dos centros C , pode-se definir a clusterização como o agrupamento dos pontos, de acordo com o centro em que cada ponto é atribuído.

A Figura 8 ilustra um exemplo de clusterização aplicando o método *k-means*, dado um conjunto de pontos (x, y) e um inteiro $k = 3$. Esse procedimento resulta, então, no

particionamento do conjunto de pontos em três *clusters*, ilustrados na Figura 8 por cores distintas. Ainda, o centro de cada *cluster* é representado por um triângulo nas respectivas cores.

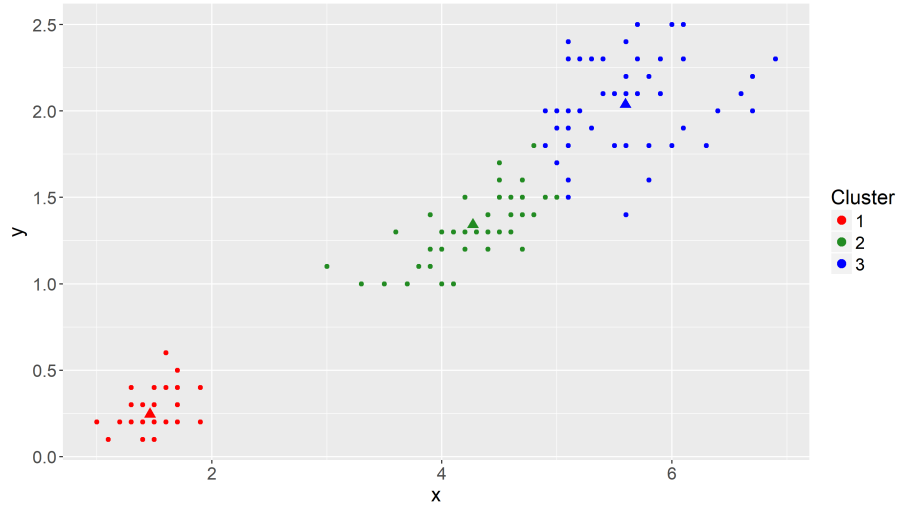


Figura 8 – Exemplo de clusterização por *k-means*

A resolução exata do problema de clusterização é considerada NP-difícil, mas Lloyd propôs uma heurística de busca local para esse problema [29]. Esse algoritmo inicializa k centros arbitrários, geralmente escolhidos de maneira aleatória e uniforme a partir dos pontos. Cada ponto é, então, atribuído ao centro mais próximo, e cada centro é recomputado como o centro de massa de todos os pontos atribuídos a ele. Esses dois últimos passos são repetidos até o processo estabilizar.

De modo formal, a técnica de clusterização *k-means* pode ser ilustrada pelo Algoritmo 3.

Algoritmo 3: Clusterização por *k-means*

Entrada: Inteiro k e conjunto de dados X com n pontos.

Saída: k centros C e *clusters* G .

```

1 início
2   Passo 1: Escolha arbitrariamente  $k$  centros iniciais  $C = \{c_1, c_2, \dots, c_k\}$ ;
3   repita
4     para  $i \leftarrow 1$  até  $k$  faça
5       Passo 2: Atribua ao cluster  $G_i$  o conjunto de pontos em  $X$  mais próxi-
6         mos ao centro  $c_i$ ;
7       fim
8     para  $i \leftarrow 1$  até  $k$  faça
9       Passo 3: Atribua ao centro  $c_i$  o centro de massa de todos os pontos
10        pertencentes ao cluster  $G_i$ ;
11      fim
12   até  $C$  não se alterar;
13 fim

```

No Passo 2 do Algoritmo 3, empates podem ser quebrados arbitrariamente. A ideia dos Passos 2 e 3 consiste na garantia de minimização da distância quadrática total ϕ , realizando melhorias locais sobre uma clusterização arbitrária até a estabilização do processo.

Esclarecidos os conceitos mais importantes associados ao processo de análise de qualidade de malhas computacionais, a Seção 2.8 é dedicada a uma revisão bibliográfica de trabalhos correlatos, levando em consideração as metodologias empregadas.

2.8 Trabalhos Correlatos

Nesta seção, um levantamento de material bibliográfico foi efetuado em bibliotecas digitais como *Google Acadêmico*¹, *ScienceDirect*² e *Springer*³, utilizando palavras-chave relevantes e associadas à qualidade de malhas computacionais. Para cada referência levantada, apresentam-se os procedimentos metodológicos aplicados para a solução do problema.

Tamura [30] produz malhas bidimensionais no sistema de coordenadas generalizadas, aplicando o método *spline* cúbico parametrizado e a adimensionalização das equações.

¹ <<https://scholar.google.com.br/>>

² <<http://www.sciencedirect.com/>>

³ <<http://www.springer.com/br/>>

Além disso, a dissertação deduz duas métricas de qualidade, nomeadas como comprimento de forma e diferença de área, com base em métricas fundamentais. A partir desses critérios de avaliação, diferentes simulações são executadas, envolvendo malhas computacionais com geometrias variadas. Os resultados mostraram que as métricas de qualidade e o modelo desenvolvido apontaram os elementos de menor qualidade.

Borouchaki e Frey [17] propõem um esquema para a conversão automática de malhas com elementos triangulares para malhas com elementos quadriláteros. A ideia principal dessa conversão consiste em emparelhar triângulos para formar quadriláteros, controlando a qualidade. O controle é realizado por meio da medida das qualidades dos ângulos internos dos elementos. De acordo com os autores, essa métrica é suficiente para identificar um quadrilátero com alta razão de aspecto, quando acoplada a um controle do comprimento da aresta. Os experimentos, em particular um problema de dinâmica dos fluidos computacional, demonstraram a eficiência do método aplicado.

Johnen, Ernst e Geuzaine [18] elaboram um algoritmo para a geração indireta de malhas quadrangulares, com base em técnicas de tomada de decisão para a busca por recombinações ótimas entre triângulos. O funcionamento desse algoritmo constitui-se em, a partir de uma malha triangular, desenvolver uma malha com elementos quadriláteros com a melhor qualidade de forma, através de recombinações. No artigo, a qualidade de forma também é baseada na métrica referente aos ângulos internos dos elementos, discutida em [17]. Por meio dos resultados obtidos de experimentos, os autores concluíram que o algoritmo é uma etapa inicial à resolução do problema tridimensional de recombinar tetraedros em hexaedros.

Koko [31] desenvolve um código na linguagem MATLAB para a geração de malhas triangulares não estruturadas em duas dimensões. O método tem como base o gerador de malhas `distmesh`, também implementado em MATLAB [32]. Uma vez que o artigo trabalha com malhas triangulares para aplicações de elementos finitos, a medida da qualidade da malha computacional consiste na razão entre o raio da maior circunferência inscrita e o raio da menor circunferência circunscrita do elemento. Os experimentos numéricos mostraram que diferentes malhas, com dimensões variadas, podem ser geradas com o código desenvolvido.

Shewchuk [16] explica as conexões matemáticas entre a geometria de malhas, erros de interpolação, erros de discretização e o condicionamento da matriz de rigidez. Essas relações são expressadas por limites de erros e medidas de qualidade dos elementos que determinam a aptidão de um triângulo ou tetraedro para a interpolação ou para o alcance de números baixos de condição. Segundo o estudo, as métricas de qualidade apresentadas são indicadores precisos de aptidão dos elementos. Ainda, verifica-se que existem circunstâncias em que os problemas são melhores solucionados por elementos com razões de aspecto ou orientações diferentes, em comparação a triângulos equiláteros.

Machado [33] estuda conceitos de Geometria Computacional voltados ao problema de geração de malhas Delaunay e implementa um método inovador para a geração de malhas. Além disso, o autor aborda algoritmos de refinamento e o processo de melhoria de qualidade de malhas computacionais. Com o propósito de analisar os algoritmos de refinamento de Delaunay, a dissertação aplica uma medida de qualidade descrita em [16]. Essa métrica refere-se à razão circunraio-menor aresta de um triângulo, ou seja, a razão entre o raio da circunferência de um triângulo e o tamanho da menor aresta do triângulo. Assim, simulações numéricas são efetuadas utilizando malhas triangulares, nas quais os métodos para a geração das malhas Delaunay obtiveram resultados satisfatórios.

Nguyen et al. [34] foca na triangulação de Delaunay uniforme de regiões planas e no modo para a seleção das posições dos vértices da triangulação. O artigo discute um método, baseado no conceito de tesselação de Voronoi centroidal, para efetuar as triangulações e desenvolve dois algoritmos para a geração de malhas computacionais. Ainda, definem-se diferentes medidas quantitativas da qualidade de malhas uniformes. Finalmente, geram-se triangulações sobre regiões planas, e conclui-se que malhas uniformes de alta qualidade foram produzidas.

Identificados os principais trabalhos relacionados ao processo de análise de qualidade de malhas, o Capítulo 3 explica os procedimentos metodológicos aplicados nas etapas de aquisição de pontos de fronteira, geração de malhas bidimensionais em coordenadas generalizadas e análise de qualidade das malhas produzidas.

3 MATERIAIS E MÉTODOS

Neste capítulo, apresentam-se os procedimentos metodológicos aplicados nos processos de aquisição de pontos de fronteira, geração de malhas em coordenadas generalizadas e análise de qualidade das malhas produzidas. Expondo uma visão geral desses procedimentos, na Seção 3.1, explicam-se os métodos para a aquisição de pontos necessários para a construção das fronteiras da geometria em estudo; na Seção 3.2, descreve-se o algoritmo desenvolvido para a geração de malhas com um bloco e multi-blocos no sistema de coordenadas generalizadas; finalmente, na Seção 3.3, definem-se as métricas de qualidade utilizadas e demonstram-se os algoritmos desenvolvidos para o cálculo dessas métricas e para a identificação dos elementos de menor qualidade em malhas computacionais.

3.1 Aquisição de Pontos de Fronteira

Para a definição e criação de uma geometria mapeada em coordenadas generalizadas, parte-se de um conjunto de pontos que descrevem o bordo, ou seja, o contorno físico. Nos experimentos apresentados no Capítulo 4, esse conjunto de pontos é obtido de diferentes formas, as quais variam conforme o experimento efetuado.

Apesar de ser mencionado na descrição de cada experimento, esta seção reúne todos os processos de aquisição de pontos de fronteira realizados. Assim, os seguintes processos de aquisição são elencados:

- manual: definição dos pontos de fronteira manualmente, por meio da visualização gráfica da localização dos pontos estipulados;
- baseado em trabalhos científicos: extração e adaptação de pontos de fronteira em trabalhos correlatos de geração de malhas e análise de qualidade [30, 35];
- automatizado: desenvolvimento de um código computacional para a aquisição dos pontos de fronteira, utilizando conceitos de geometria, tais como as equações da reta e da circunferência;
- via *Google*: obtenção de pontos de fronteira a partir de imagens de satélite fornecidas pelo aplicativo *Google Earth* e imagens de um determinado objeto ou geometria encontradas no *Google* [36, 37].

Vale salientar que, nos casos em que havia imagens da geometria sobre a qual o estudo da análise de qualidade seria efetuado, a ferramenta *WebPlotDigitizer* é empregada

[38]. Essa ferramenta permite, entre outras utilidades, extrair dados numéricos com base em plotagens gráficas, imagens e mapas, possibilitando então a aquisição dos pontos de fronteira da geometria sendo investigada.

Tendo em vista a equação governante para a geração de malhas em coordenadas generalizadas, descrita na Subseção 2.4.3, a Seção 3.2 apresenta minuciosamente o algoritmo que implementa a resolução numérica da Equação (2.42).

3.2 Gerador de Malhas em Coordenadas Generalizadas

Para a criação de um procedimento automatizado no mapeamento de uma geometria descrita em coordenadas cartesianas segundo o sistema de coordenadas generalizadas, opta-se por utilizar a linguagem de programação *Python* [39]. A motivação para tal decisão reside no fato de que *Python* é uma linguagem de propósito geral, livre, aberta e multiplataforma [39], o que permite a criação de extensões da aplicação desenvolvida, como, por exemplo, um módulo de interface gráfica, também desenvolvido em *Python*.

Em relação ao desempenho, faz-se necessário o uso de bibliotecas extras, específicas para aplicações em computação numérica, para o alcance de resultados satisfatórios no tempo de execução. Por ser uma linguagem de *script*, ou seja, interpretada, o desempenho de uma aplicação escrita em *Python* tende a ser inferior, quando comparado com uma linguagem compilada, como é o caso de *Fortran* [40]. Nativamente, as ferramentas disponíveis para a manipulação matricial e álgebra linear são limitadas, ou, em outras palavras, não otimizadas para o tipo de aplicação que se deseja desenvolver. Todavia, pelo fato de que uma das características da linguagem adotada é ser modular, é possível contornar esse problema com o uso de rotinas e bibliotecas de terceiros, escritas em linguagens compiladas, com o propósito de otimizar operações mais complexas, como as utilizadas para a transformação de domínios [40].

Para o desenvolvimento do gerador de malhas bidimensionais, as seguintes bibliotecas foram empregadas:

- *Numpy*: biblioteca voltada para operações matriciais e algébricas de forma geral [41, 42];
- *Matplotlib*: biblioteca utilizada para a plotagem de gráficos e manipulação de dados em forma gráfica [42, 43].

Os módulos adicionais empregados são também livres para uso e desenvolvidos em comunidade.

Na Subseção 3.2.1, uma descrição mais detalhada do algoritmo desenvolvido é apresentada.

3.2.1 Descrição do Algoritmo Desenvolvido

Considerando uma coleção de pontos discretos que definem o bordo, neste trabalho, aplica-se o método *spline* linear parametrizado [5, 11, 19], descrito na Subseção 2.1.1, para interpolar os pontos e obter as fronteiras da geometria em estudo. A partir do momento em que as curvas interpoladoras do bordo são conhecidas, Equações (2.2) e (2.3), define-se a quantidade de partições desejadas para cada fronteira. Vale salientar que se opta por manter um espaçamento uniforme entre os pontos de cada fronteira, observando que, se o número de partições do plano transformado não for divisível pelo número de partições do plano físico, os novos pontos definidos sofrerão um leve deslocamento em comparação aos pontos pertencentes aos dados de entrada.

Em seguida, calculam-se os pontos internos da geometria aproximada com base nos novos pontos de bordo definidos. Para tal, faz-se uso de uma abordagem de posicionamento de pontos interiores considerando uma média ponderada dos componentes dos bordos, representada por

$$x_{i,j} = p_x^{\Gamma_1\Gamma_3}(x_{0,j} + i\Delta_{j-1}^{x\xi}) + p_x^{\Gamma_2\Gamma_4}(x_{i,0} + j\Delta_{i-1}^{x\eta}); \quad (3.1)$$

$$y_{i,j} = p_y^{\Gamma_1\Gamma_3}(y_{0,j} + i\Delta_{j-1}^{y\xi}) + p_y^{\Gamma_2\Gamma_4}(y_{i,0} + j\Delta_{i-1}^{y\eta}); \quad (3.2)$$

em que $\Delta_j^{x\xi} = \frac{x_{\xi,j} - x_{0,j}}{\xi}$, $\Delta_i^{x\eta} = \frac{x_{i,\eta} - x_{i,0}}{\eta}$, $\Delta_j^{y\xi} = \frac{y_{\xi,j} - y_{0,j}}{\xi}$, $\Delta_i^{y\eta} = \frac{y_{i,\eta} - y_{i,0}}{\eta}$, $i = 1, \dots, \xi$ e $j = 1, \dots, \eta$. Os valores de $p_x^{\Gamma_1\Gamma_3}$, $p_x^{\Gamma_2\Gamma_4}$, $p_y^{\Gamma_1\Gamma_3}$ e $p_y^{\Gamma_2\Gamma_4}$ indicam os pesos sobre as fronteiras esquerda/direita em x , superior/inferior em x , esquerda/direita em y e superior/inferior em y , respectivamente.

Atribuindo pesos percentuais, através das Equações (3.1) e (3.2), pode-se melhor ajustar a distribuição dos pontos. Dessa maneira, a disposição dos pontos internos é influenciada em diferentes graus pelos bordos, conforme a configuração da geometria em estudo, o que permite maior liberdade e alcance representativo. Finalmente, sobre o conjunto de pares ordenados, aplica-se a resolução numérica da equação governante para a geração de malhas bidimensionais em coordenadas generalizadas, Equação (2.42), utilizando o método de Gauss-Seidel [19, 26], abordado na Subseção 2.5.1.

De maneira sintética, o Algoritmo 4 exhibe um pseudocódigo do gerador de malhas proposto e implementado.

Algoritmo 4: Gerador de malhas bidimensionais em coordenadas generalizadas

Entrada: Número de partições dos planos físico e transformado, pesos das fronteiras e pontos de contorno.

Saída: Coordenadas dos nós da malha computacional.

1 **início**

2 **Leitura dos dados de entrada;**

3 **Interpolação das fronteiras:** determine as curvas interpoladoras das fronteiras utilizando o método *spline* linear parametrizado;

4 **Cálculo dos pontos internos:** encontre os pontos internos por meio de uma média ponderada das fronteiras interpoladas;

5 **Resolução da equação diferencial:** resolva numericamente a equação governante para a geração de malhas em coordenadas generalizadas, aplicando o método iterativo de Gauss-Seidel;

6 **fim**

O Algoritmo 4 consiste em um gerador de malhas bidimensionais desenvolvido para geometrias constituídas por um bloco. Para geometrias bastante complexas, é conveniente utilizar a técnica multi-blocos [11, 25], a qual é descrita e implementada na Subseção 3.2.2.

3.2.2 Técnica Multi-blocos

Uma relevante parte das geometrias presentes no mundo real é constituída por estruturas complexas que não podem ser modeladas por um gerador simples de malhas em coordenadas generalizadas, exigindo técnicas mais aprimoradas para o processo de geração. Para contornar esse problema e permitir a adaptabilidade do gerador a diferentes geometrias, existe a técnica multi-blocos [11, 25].

Essa metodologia consiste em dividir a geometria estudada em partes e, sobre cada uma das partes, aplicar a geração das malhas em coordenadas generalizadas independentemente, o que possibilita captar melhor a complexidade da geometria em comparação com a estratégia empregada sobre um bloco.

O algoritmo de geração de malhas envolvendo multi-blocos utiliza o mesmo gerador de malhas definido no Algoritmo 4 e pode ser ilustrado, de maneira simplificada, pelo Algoritmo 5.

Algoritmo 5: Gerador de malhas bidimensionais em coordenadas generalizadas envolvendo multi-blocos

Entrada: Blocos contendo número de partições dos planos físico e transformado, pesos das fronteiras e pontos de contorno.

Saída: Gráfico e coordenadas dos nós de cada bloco da malha computacional.

```

1 início
2   para cada arquivo de entrada (bloco) faça
3     Geração da malha: gere a malha computacional do bloco executando o
      Algoritmo 4;
4     Plotagem da malha: plote a malha computacional do bloco utilizando os
      resultados da geração;
5     Escrita dos resultados;
6   fim
7 Exibição do gráfico: mostre as malhas computacionais geradas para todos os
      blocos em um gráfico;
8 fim

```

Sendo possível realizar a geração de malhas bidimensionais em coordenadas generalizadas, parte-se, nesse momento, para o próximo objetivo deste trabalho, o qual consiste em analisar a qualidade das malhas geradas. As métricas de qualidade e os algoritmos desenvolvidos são explicados detalhadamente na Seção 3.3.

3.3 Análise de Qualidade de Malhas em Coordenadas Generalizadas

Com a finalidade de analisar a qualidade dos elementos pertencentes às malhas geradas em coordenadas generalizadas, três características foram levadas em consideração. Um elemento considerado de maior qualidade é um quadrilátero que mais se aproxima de um quadrado [17, 18], ou seja, com todos os lados e ângulos iguais. Proveniente da área de Computação Gráfica, a terceira característica diz respeito ao coeficiente de compacidade do elemento [44, 45, 46]. Mais detalhes acerca das métricas de qualidade utilizadas no processo de análise são apresentados na Subseção 3.3.1.

3.3.1 Métricas de Qualidade

Antes de definir as métricas de qualidade de maneira detalhada, é importante ressaltar a padronização adotada para a localização das arestas e ângulos de um elemento

qualquer na malha computacional. A Figura 9 demonstra as nomenclaturas adotadas para a definição das arestas, Figura 9a, e ângulos, Figura 9b, do elemento, onde a numeração parte do nó i, j , contornando o elemento em sentido horário.

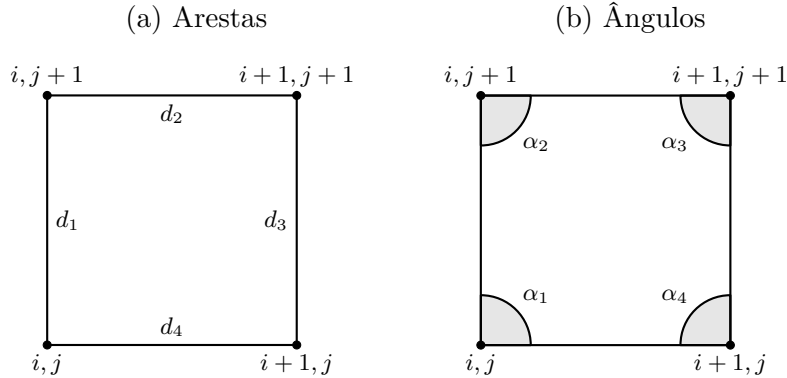


Figura 9 – Nomenclaturas adotadas para a definição das arestas e ângulos de um elemento

A imagem de um quadrado na Figura 9 é meramente ilustrativa, significando que os elementos podem ser quadriláteros de formatos quaisquer. Ainda, a notação i, j é uma simplificação do nó (x_i, y_j) , em que $i = 1, \dots, M$ e $j = 1, \dots, N$, lembrando que M é o número de partições na direção ξ , e N , o número de partições na direção η , como mencionados na Subseção 2.4.3.

Na abordagem empregada, executa-se a avaliação das malhas geradas, na qual, para cada elemento, calculam-se as razões $r_{i,j}$ entre o tamanho das arestas, de modo a verificar se os valores estão próximos a $r_{\square} = 1$, correspondendo à razão entre duas arestas de um quadrado. Em outras palavras, a partir do cálculo da distância d entre dois vértices $A(x_A, y_A)$ e $B(x_B, y_B)$ conectados do elemento, indicada por [47]

$$d = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}; \quad (3.3)$$

encontra-se o tamanho das arestas d_i , $i = 1, \dots, 4$, e, em seguida, efetua-se a razão $r_{i,j}$ de cada aresta com as demais, ou seja,

$$r_{i,j} = \frac{d_i}{d_j}; \quad (3.4)$$

em que $i = 1, \dots, 4$, $j = 1, \dots, 4$ e $i \neq j$. Vale salientar que o símbolo \square é utilizado com a finalidade de denotar os cálculos efetuados sobre um quadrado.

Além da razão entre o tamanho das arestas do quadrilátero, para garantir a propriedade citada de que um elemento de maior qualidade é um quadrilátero similar a um quadrado, é necessário também averiguar se os ângulos internos α_i , $i = 1, \dots, 4$, de cada elemento se aproximam a $\alpha_{\square} = \frac{\pi}{2}$ rad, utilizando [48]

$$\alpha_i = \arccos \left(\frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \cdot \|\vec{v}\|} \right); \quad (3.5)$$

em que $\vec{u} = \overrightarrow{AB} = \langle x_B - x_A, y_B - y_A \rangle$ e $\vec{v} = \overrightarrow{AC} = \langle x_C - x_A, y_C - y_A \rangle$ são vetores de origem $A(x_A, y_A)$ que representam, respectivamente, duas arestas \overline{AB} e \overline{AC} do elemento. As normas ou módulos $\|\vec{u}\|$ e $\|\vec{v}\|$ correspondem ao comprimento dos vetores \vec{u} e \vec{v} , respectivamente, podendo serem calculadas por meio da Equação (3.3) [48].

Ainda, uma terceira métrica de qualidade, empregada para verificar a similaridade entre os elementos pertencentes à malha computacional e um quadrado, diz respeito ao coeficiente de compacidade c . Essa métrica possui a característica de ser invariante à translação, à rotação e à mudança de escala e é calculada por [44, 45, 46]

$$c = \frac{P^2}{S}; \quad (3.6)$$

em que $P = \sum_{i=1}^4 d_i$ é o perímetro, e S é a área do elemento. A área S do quadrilátero é calculada por meio da Equação (2.32), na qual se utiliza o Jacobiano da transformação.

Considere d_{\square} o tamanho da aresta de um quadrado, então o coeficiente de compacidade c_{\square} do quadrado é dado por

$$c_{\square} = \frac{(4d_{\square})^2}{d_{\square}^2} = \frac{16d_{\square}^2}{d_{\square}^2} = 16. \quad (3.7)$$

Assim, o coeficiente de compacidade de um quadrado qualquer é $c_{\square} = 16$, e o coeficiente de compacidade c para quadriláteros com formatos diferentes apresentam valores mais altos, permitindo aplicá-lo no processo de análise de qualidade.

Portanto, a importância da aplicação das métricas de qualidade nos processos de análise e, posteriormente, de melhoria de malhas computacionais reside no fato de que a regularidade da geometria dos elementos possibilita aumentar a acurácia da solução das equações diferenciais investigadas, melhorar a estabilidade e a convergência do método numérico, além de reduzir o tempo de execução do algoritmo de resolução [14, 15, 16], trazendo benefícios para a simulação e, conseqüentemente, para o estudo realizado.

Levando em consideração as métricas de qualidade, a Subseção 3.3.2 descreve os algoritmos desenvolvidos para a análise de qualidade de malhas computacionais em coordenadas generalizadas.

3.3.2 Descrição dos Algoritmos Desenvolvidos

O processo de desenvolvimento do algoritmo para análise de qualidade de malhas em coordenadas generalizadas se dividiu em dois módulos. O primeiro módulo constitui um algoritmo para o cálculo das métricas de qualidade, a partir das malhas geradas pelos Algoritmos 4 e 5. Assim como os geradores de malhas desenvolvidos, esse módulo também foi implementado na linguagem de programação *Python* [39] e é abordado na Subseção 3.3.2.1.

O segundo módulo implementado para análise de qualidade de malhas em coordenadas generalizadas realiza a identificação dos elementos de menor qualidade, de acordo com as métricas empregadas e calculadas pelo primeiro módulo desenvolvido. Esse módulo foi implementado na linguagem de programação R [49], devido à existência de bibliotecas constituídas por procedimentos estatísticos utilizados no decorrer desse algoritmo [49, 50]. Ainda, a biblioteca *ggplot2* foi utilizada para plotagens gráficas [51]. Mais detalhes sobre o segundo módulo são explicados na Subseção 3.3.2.2.

3.3.2.1 Cálculo das Métricas de Qualidade

Com base nas coordenadas dos nós das malhas geradas e no número de partições nas direções ξ e η , calcula-se no primeiro módulo, para cada elemento de cada bloco da malha computacional, o tamanho das arestas d_i , $i = 1, \dots, 4$, através da Equação (3.3). Posteriormente, calculam-se as razões $r_{i,j}$ entre cada tamanho de aresta com os demais, por meio da Equação (3.4), totalizando 12 combinações. Verifica-se e extrai-se, então, qual a combinação r com maior disparidade em comparação a $r_{\square} = 1$, equivalente à razão entre duas arestas de um quadrado.

Analogamente, calculam-se os ângulos internos α_i , $i = 1, \dots, 4$, de cada elemento, utilizando a Equação (3.5) e o tamanho das arestas d_i previamente calculado. Verifica-se e extrai-se, então, qual o ângulo interno α com maior disparidade em comparação a $\alpha_{\square} = \frac{\pi}{2}$ rad, correspondente ao ângulo interno de um quadrado.

Na sequência, a área S de cada elemento é calculada pela Equação (2.32), e o perímetro P , calculado pelo somatório do tamanho das arestas d_i previamente determinado. Dessa forma, calcula-se o coeficiente de compacidade c de cada elemento através da Equação (3.6).

É importante salientar que se realiza um tratamento adequado para os casos em que o elemento é constituído por dois pontos sobrepostos, isto é, possui um formato triangular. Esses casos podem acontecer sobre os elementos pertencentes às fronteiras da geometria em estudo, por conta dos pontos de fronteira adquiridos e da complexidade da geometria. Por esse motivo, o tamanho da aresta entre os pontos sobrepostos é nulo, ou seja, $d = 0$, e assim atribuem-se $r = +\infty$ e $\alpha = \#$ para o elemento. Uma vez que o tamanho da aresta nulo não influencia sobre o coeficiente de compacidade c causando indeterminação ou inexistência, essa métrica de qualidade continua a ser calculada da mesma forma.

Finalmente, o primeiro módulo produz, como saída, um arquivo *.csv* contendo o conjunto total de elementos e suas métricas de qualidade, com informações sobre o número do bloco em que o elemento pertence, as coordenadas dos nós, a razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, o ângulo interno α com maior disparidade em

relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e o coeficiente de compacidade c .

De maneira organizada, o Algoritmo 6 apresenta a sequência de passos empregada para o cálculo das métricas de qualidade de malhas em coordenadas generalizadas.

Algoritmo 6: Cálculo das métricas de qualidade de malhas em coordenadas generalizadas

Entrada: Malhas contendo número de partições do plano transformado e coordenadas dos nós.

Saída: Conjunto total de elementos e suas métricas de qualidade, contendo número do bloco em que o elemento pertence, coordenadas dos nós, razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e coeficiente de compacidade c .

1 **início**

2 **para cada** *arquivo de entrada (malha)* **faça**

3 **Leitura dos dados de entrada;**

4 **para cada** *elemento da malha* **faça**

5 **Cálculo do tamanho das arestas:** calcule o tamanho das arestas d_i , $i = 1, \dots, 4$, conectando os nós do elemento, através da Equação (3.3);

6 **Cálculo das razões entre arestas:** calcule as razões $r_{i,j}$ entre cada aresta do elemento com as demais, de acordo com a Equação (3.4);

7 **Extração da razão entre arestas:** extraia a razão entre arestas r com maior disparidade em comparação a $r_{\square} = 1$;

8 **Cálculo dos ângulos internos:** calcule os ângulos internos α_i , $i = 1, \dots, 4$, do elemento, por meio da Equação (3.5) e do tamanho das arestas d_i ;

9 **Extração do ângulo interno:** extraia o ângulo interno α com maior disparidade em comparação a $\alpha_{\square} = \frac{\pi}{2}$ rad;

10 **Cálculo da área:** calcule a área S do elemento, utilizando a Equação (2.32);

11 **Cálculo do perímetro:** calcule o perímetro P do elemento pelo somatório do tamanho das arestas d_i ;

12 **Cálculo do coeficiente de compacidade:** calcule o coeficiente de compacidade c do elemento, através da Equação (3.6);

13 **fim**

14 **fim**

15 **Escrita dos resultados;**

16 **fim**

3.3.2.2 Identificação dos Elementos de Menor Qualidade

A partir do arquivo *.csv* contendo o conjunto total de elementos e suas métricas de qualidade gerado pelo Algoritmo 6, identificam-se os elementos considerados de menor qualidade no segundo módulo. Nessa etapa, as informações acerca da razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, do ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e do coeficiente de compacidade c são essenciais para a classificação dos elementos. Além disso, é necessário também especificar o número de partições nas direções ξ e η para cada bloco da malha computacional e uma porcentagem desejada \bar{p} de elementos de menor qualidade a serem identificados.

Antes de explicar minuciosamente o algoritmo de identificação dos elementos de menor qualidade, é importante mencionar que o conjunto total de elementos e suas métricas de qualidade é denotado por U . Assim, o primeiro passo do segundo módulo para análise de qualidade é caracterizado pela retirada dos elementos em que $r = +\infty$ e $\alpha = \nexists$ do conjunto U . Dessa forma, os passos posteriores realizam operações utilizando apenas o conjunto de elementos cujas métricas de qualidade existem e são finitas. Esse novo conjunto de elementos e suas métricas de qualidade é denotado por U' .

Utilizando o conjunto U' , a variância das razões entre arestas s_R^2 , dos ângulos internos s_A^2 e dos coeficientes de compacidade s_C^2 é calculada por [52, 53, 54]

$$s_M^2 = \frac{\sum_{k=1}^{|M|} (m_k - \bar{m})^2}{|M| - 1}; \quad (3.8)$$

onde M é uma notação adotada para a representação de um dos conjuntos R , A ou C , dependendo da métrica de qualidade em que a variância é calculada. Ainda na Equação (3.8), $|M|$ representa a quantidade de elementos pertencentes ao conjunto M , $m_k \in M$ indica um valor da métrica de qualidade M , e \bar{m} é a média dos valores da métrica de qualidade M .

Com base nos valores das variâncias s_R^2 , s_A^2 e s_C^2 , é possível determinar se não existem elementos de menor qualidade no conjunto U' , uma vez que, quando todas as variâncias são nulas, todos os elementos da malha computacional são idênticos entre si. Dessa maneira, se $s_R^2 = s_A^2 = s_C^2 = 0$, verifica-se também se os valores das razões entre arestas r_k , dos ângulos internos α_k e dos coeficientes de compacidade c_k , $k = 1, \dots, |M|$, são equivalentes às métricas avaliadas sobre um quadrado. Em outras palavras, verifica-se se $r_k = 1$, $\alpha_k = \frac{\pi}{2}$ rad e $c_k = 16$ para todo k , permitindo afirmar se os elementos são iguais ou diferentes de um quadrado, interrompendo a execução do algoritmo.

Caso, pelo menos, uma das variâncias s_R^2 , s_A^2 ou s_C^2 seja não nula, o algoritmo continua sua execução, na qual se calculam os valores mínimo e máximo de cada métrica de qualidade, expressados, respectivamente, por m_{\min} e m_{\max} . No cálculo dessas variáveis, o valor das métricas avaliadas sobre um quadrado m_{\square} também é levado em consideração.

Em termos matemáticos, m_{\min} e m_{\max} são encontrados, respectivamente, por

$$\begin{aligned} m_{\min} &= \min\{m_k, m_{\square}\}; \\ m_{\max} &= \max\{m_k, m_{\square}\}; \end{aligned} \quad (3.9)$$

em que $k = 1, \dots, |M|$ indica um elemento pertencente ao conjunto M .

Utilizando os valores mínimo m_{\min} e máximo m_{\max} das métricas de qualidade, a próxima etapa consiste em normalizar os valores de cada métrica, assim como o valor das métricas avaliadas sobre um quadrado. A etapa de normalização possibilita manter todos os conjuntos de dados dentro de um mesmo intervalo, evitando-se trabalhar com dados cujos valores se sobressaem em relação a outros. Dessa forma, o cálculo das métricas de qualidade normalizadas m_{norm_k} , $k = 1, \dots, |M|$, é dado por

$$m_{\text{norm}_k} = (b - a) \frac{m_k - m_{\min}}{m_{\max} - m_{\min}} + a; \quad (3.10)$$

resultando em valores pertencentes ao intervalo $[a, b]$ e que formam o conjunto M_{norm} . Substituindo m_k por m_{\square} na Equação (3.10), o cálculo das métricas avaliadas sobre um quadrado normalizadas $m_{\text{norm}_{\square}}$ é realizado de modo análogo

$$m_{\text{norm}_{\square}} = (b - a) \frac{m_{\square} - m_{\min}}{m_{\max} - m_{\min}} + a. \quad (3.11)$$

Nos experimentos executados no Capítulo 4, padroniza-se a normalização dos valores das métricas de qualidade no intervalo $[0, 10]$.

Sobre as métricas de qualidade normalizadas m_{norm_k} , um novo cálculo das variâncias $s_{M_{\text{norm}}}^2$ é efetuado, através da Equação (3.8). A partir das variâncias $s_{M_{\text{norm}}}^2$, das métricas de qualidade normalizadas m_{norm_k} e das métricas avaliadas sobre um quadrado normalizadas $m_{\text{norm}_{\square}}$, o algoritmo calcula tolerâncias mínima ϵ_{\min} e máxima ϵ_{\max} .

Pode-se enunciar que a tolerância ϵ é definida como o critério de parada de um processo iterativo, em que se extraem os elementos de menor qualidade em relação à razão entre arestas, ao ângulo interno ou ao coeficiente de compacidade, formando os conjuntos E_r , E_{α} ou E_c , respectivamente. Para simplificar a explicação do processo iterativo, considere que E_m representa um dos conjuntos E_r , E_{α} ou E_c , o qual depende da métrica de qualidade empregada para a classificação dos elementos. Assim, cada iteração i consiste na obtenção de um subconjunto de E_m , denotado por E_{m_i} e constituído pelos elementos de menor qualidade identificados na iteração i e unidos com os identificados na iteração $i - 1$, sendo que $E_{m_0} = \emptyset$. Sabendo que M_i é uma das métricas avaliadas sobre os elementos de menor qualidade E_{m_i} , uma nova variância $s_{M_{\text{norm}}-M_i}^2$ também é calculada no decorrer da iteração. Para que o processo iterativo termine, é preciso que $|s_{M_{\text{norm}}-M_i}^2 - s_{M_{\text{norm}}-M_{i-1}}^2| \leq \epsilon$ ou, em outras palavras, é necessário que a diferença entre as variâncias da iteração atual i e da iteração anterior $i - 1$, em módulo, seja menor ou igual à tolerância ϵ .

Dessa maneira, a tolerância mínima ϵ_{\min} é o menor valor de tolerância em que o conjunto $E_U = E_r \cup E_\alpha \cup E_c$ é constituído pela máxima quantidade de elementos possível, e a tolerância máxima ϵ_{\max} é o maior valor de tolerância em que o conjunto E_U é constituído pela mínima quantidade de elementos possível. Dados os valores de ϵ_{\min} e ϵ_{\max} , itera-se no intervalo entre as duas tolerâncias com um incremento de $1 \cdot 10^{-5}$, identificando os elementos de menor qualidade em relação à razão entre arestas E_r , ao ângulo interno E_α e ao coeficiente de compacidade E_c , considerando a tolerância ϵ na iteração i , expressada por ϵ_i . Para cada ϵ_i , obtêm-se, então, o conjunto E_{U_i} , formado pela união dos elementos de menor qualidade em relação a cada uma das métricas, e conseqüentemente a sua quantidade de elementos $|E_{U_i}|$.

Plotando o gráfico da tolerância ϵ_i pela quantidade de elementos de menor qualidade $|E_{U_i}|$ em diferentes experimentos, observou-se a característica exponencial da curva formada pela união dos pontos discretos. Essa informação é mostrada com mais detalhes nos experimentos presentes no Capítulo 4. Assim, o passo subsequente do algoritmo objetiva encontrar uma função de regressão não linear exponencial que se aproxima do gráfico da tolerância ϵ_i pela quantidade de elementos de menor qualidade $|E_{U_i}|$. Adequando a Equação (2.58) ao problema, ou seja, substituindo x por ϵ e $\varphi(x)$ por $|E_U|$, essa função é descrita por [54, 55, 56]

$$|E_U| = c_1 e^{c_2 \epsilon}; \quad (3.12)$$

em que c_1 e c_2 são os coeficientes a serem determinados. Para a obtenção desses coeficientes, o método de mínimos quadrados é aplicado [26], cuja técnica de resolução é abordada na Subseção 2.6.1.

Com base nos coeficientes c_1 e c_2 da função de regressão e na porcentagem desejada \bar{p} de elementos de menor qualidade a serem identificados, é possível calcular uma tolerância desejada $\bar{\epsilon}$. Isolando ϵ na Equação (3.12), tem-se que

$$\epsilon = \frac{\ln(|E_U|) - \ln(c_1)}{c_2}. \quad (3.13)$$

Então, a tolerância desejada $\bar{\epsilon}$ pode ser encontrada por

$$\bar{\epsilon} = \frac{\ln(\bar{p} \cdot |U'|) - \ln(c_1)}{c_2}; \quad (3.14)$$

onde $|U'|$ é quantidade de elementos pertencentes ao conjunto U' .

Considerando o valor da tolerância desejada $\bar{\epsilon}$, o algoritmo identifica os elementos de menor qualidade em relação à razão entre arestas, ao ângulo interno e ao coeficiente de compacidade, compondo, respectivamente, os conjuntos \bar{E}_r , \bar{E}_α e \bar{E}_c . Além disso, o conjunto $\bar{E}_U = \bar{E}_r \cup \bar{E}_\alpha \cup \bar{E}_c$ de elementos de menor qualidade é encontrado, com a finalidade de classificar os elementos considerados de maior qualidade, isto é, os elementos pertencentes ao conjunto $U' - \bar{E}_U$. Finalmente, obtêm-se os conjuntos $\bar{E}_{r,\alpha} = \bar{E}_r \cap \bar{E}_\alpha$,

$\overline{E_{r,c}} = \overline{E_r} \cap \overline{E_c}$ e $\overline{E_{\alpha,c}} = \overline{E_\alpha} \cap \overline{E_c}$, formados pela intersecção entre dois dos conjuntos $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$, assim como se obtém o conjunto $\overline{E_\cap} = \overline{E_r} \cap \overline{E_\alpha} \cap \overline{E_c}$, constituído pela intersecção entre os três conjuntos. Essas intersecções são realizadas com o propósito de destacar os elementos pertencentes a esses conjuntos, expressando a menor qualidade quanto a duas ou três métricas de avaliação.

Para a visualização da classificação dos elementos pertencentes à malha computacional, define-se um número q para cada categoria de elementos, utilizando os conjuntos obtidos na etapa anterior. Ao conjunto $U' - \overline{E_\cap}$ de elementos de maior qualidade, atribui-se $q = 0$. Os conjuntos $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$ de elementos de menor qualidade em relação a uma das métricas de avaliação são categorizados, respectivamente, nos intervalos de valores $1 \leq q \leq 5$, $6 \leq q \leq 10$ e $11 \leq q \leq 15$. Para essa classificação, aplica-se o algoritmo de clusterização *k-means*, apresentado na Seção 2.7, o qual particiona cada um dos conjuntos em k *clusters*. A princípio, a quantidade de *clusters* é $k = 5$, podendo variar caso a quantidade de valores distintos das métricas de qualidade seja menor que k . Ainda, atribuem-se $q = 16, 17, 18$ e 20 , respectivamente, aos conjuntos formados pelas intersecções $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$ e $\overline{E_\cap}$. Finalmente, os elementos em que $r = +\infty$ e $\alpha = \#$, pertencentes ao conjunto $U - U'$, são rotulados com $q = 19$.

Portanto, o segundo módulo para análise de qualidade produz, como saída, um arquivo *.vtk* para cada bloco da malha computacional. Cada arquivo contém informações sobre o número de pontos nas direções ξ e η , as coordenadas dos nós e o número q associado a cada elemento pertencente ao bloco. Esses dados constituem o arquivo de entrada para o *software Paraview* [57, 58], no qual os resultados obtidos da análise de qualidade são visualizados.

Internamente a esse *software*, é importante mencionar também que uma legenda de cores é empregada para a visualização gráfica da classificação dos elementos, onde cada cor está relacionada a um valor de q atribuído. A Figura 10 ilustra essa legenda de cores e é frequentemente utilizada nos experimentos presentes no Capítulo 4.

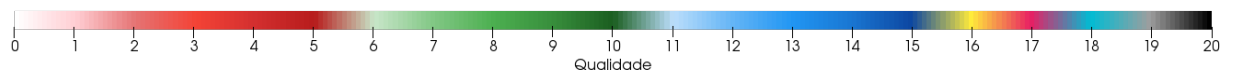


Figura 10 – Legenda de cores associadas à qualidade dos elementos de uma malha computacional

Na Figura 10, vale salientar que elementos com tonalidades mais escuras de vermelho, verde e azul indicam uma qualidade pior em comparação aos elementos com tonalidades mais claras dessas cores. Nota-se também que a combinação de duas das cores primárias do canal RGB (vermelho, verde e azul), relativas a duas métricas de qualidade, formam as cores dos elementos de menor qualidade pertencentes aos conjuntos resultantes das intersecções entre essas métricas.

Assim, de maneira sucinta, o Algoritmo 7 elenca a sequência de passos empregada para a identificação dos elementos de menor qualidade de malhas em coordenadas generalizadas. Por conta de sua extensão e com a finalidade de apresentá-lo de forma mais didática, esse algoritmo foi dividido em duas partes, executando os Algoritmos 8 e 9.

Algoritmo 7: Identificação dos elementos de menor qualidade de malhas em coordenadas generalizadas

Entrada: Conjunto total de elementos e suas métricas de qualidade, contendo número do bloco em que o elemento pertence, coordenadas dos nós, razão entre arestas r , ângulo interno α e coeficiente de compacidade c ; número de partições do plano transformado para cada bloco; e porcentagem desejada \bar{p} de elementos de menor qualidade.

Saída: Blocos contendo número de pontos nas direções ξ e η , coordenadas dos nós e valor de q associado a cada elemento pertencente ao bloco.

1 **início**
 2 | **Execução do Algoritmo 8;**
 3 | **Execução do Algoritmo 9;**
 4 **fim**

Algoritmo 8: Identificação dos elementos de menor qualidade de malhas em coordenadas generalizadas - Parte 1

```

1 início
2   Leitura dos dados de entrada: leia o número do bloco, as coordenadas dos
   nós, a razão entre arestas  $r$ , o ângulo interno  $\alpha$  e o coeficiente de compacidade
    $c$  de cada elemento;
3   Obtenção do conjunto de elementos  $U'$ : retire os elementos em que  $r =$ 
    $+\infty$  e  $\alpha = \nexists$  do conjunto de elementos  $U$ ;
4   Cálculo das variâncias: calcule a variância das razões entre arestas  $s_R^2$ , dos
   ângulos internos  $s_A^2$  e dos coeficientes de compacidade  $s_C^2$ , de acordo com a
   Equação (3.8);
5   se  $s_R^2 = 0$  e  $s_A^2 = 0$  e  $s_C^2 = 0$  então
6     se  $r_k = r_{\square}$  e  $\alpha_k = \alpha_{\square}$  e  $c_k = c_{\square}$  então
7       Todos os elementos da malha são idênticos e iguais a um quadrado;
8     senão
9       Todos os elementos da malha são idênticos, mas diferentes de um qua-
       drado;
10    fim
11    pare
12  fim
13  Cálculo dos valores mínimo e máximo: calcule os valores mínimo  $m_{\min}$  e
   máximo  $m_{\max}$  das métricas de qualidade, utilizando as Equações (3.9);
14  Normalização: normalize os valores das métricas de qualidade, assim como o
   valor das métricas avaliadas sobre um quadrado, por meio das Equações (3.10)
   e (3.11);
15  Cálculo das variâncias: calcule as variâncias  $s_{M_{\text{norm}}}^2$  das métricas de qualidade
   normalizadas, através da Equação (3.8);
16 fim

```

Algoritmo 9: Identificação dos elementos de menor qualidade de malhas em coordenadas generalizadas - Parte 2

1 **início**

2 **Cálculo das tolerâncias:** calcule as tolerâncias mínima ϵ_{\min} e máxima ϵ_{\max} ;

3 **para** $\epsilon_i \leftarrow \epsilon_{\min}$ **até** ϵ_{\max} **faça**

4 **Identificação dos elementos de menor qualidade:** identifique os elementos de menor qualidade em relação à razão entre arestas E_r , ao ângulo interno E_α e ao coeficiente de compacidade E_c , considerando a tolerância ϵ_i ;

5 **Obtenção do conjunto** E_{\cup_i} : obtenha o conjunto E_{\cup_i} , formado pela união dos conjuntos E_r , E_α e E_c , e sua quantidade de elementos $|E_{\cup_i}|$;

6 **fim**

7 **Obtenção da função de regressão não linear exponencial:** encontre a função de regressão não linear exponencial que se ajusta ao gráfico da tolerância ϵ_i pela quantidade de elementos de menor qualidade $|E_{\cup_i}|$, aplicando o método de mínimos quadrados;

8 **Cálculo da tolerância desejada:** calcule a tolerância desejada $\bar{\epsilon}$ a partir da função de regressão não linear exponencial e da porcentagem desejada \bar{p} ;

9 **Identificação dos elementos de menor qualidade:** identifique os elementos de menor qualidade em relação à razão entre arestas \bar{E}_r , ao ângulo interno \bar{E}_α e ao coeficiente de compacidade \bar{E}_c , considerando a tolerância desejada $\bar{\epsilon}$;

10 **Obtenção de demais conjuntos:** obtenha os conjuntos \bar{E}_{\cup} , $\bar{E}_{r,\alpha}$, $\bar{E}_{r,c}$, $\bar{E}_{\alpha,c}$ e \bar{E}_{\cap} , formados por operações envolvendo os conjuntos \bar{E}_r , \bar{E}_α e \bar{E}_c ;

11 **Atribuição do valor de** q : defina um número q para cada categoria de elementos, considerando os conjuntos obtidos;

12 **Escrita dos resultados;**

13 **fim**

Utilizando a razão entre arestas, o ângulo interno e o coeficiente de compacidade como métricas de qualidade, um conjunto de experimentos é executado no Capítulo 4 com a finalidade de validar os algoritmos desenvolvidos e os procedimentos metodológicos aplicados. Para cada experimento efetuado, abordam-se também os processos de aquisição de pontos de fronteira e de geração de malhas em coordenadas generalizadas, e discutem-se os resultados obtidos.

4 RESULTADOS E DISCUSSÕES

Para verificar a eficiência dos algoritmos desenvolvidos e dos procedimentos metodológicos empregados neste trabalho, diferentes experimentos são efetuados. Dentre os experimentos, apresentam-se malhas computacionais formadas por um bloco e malhas computacionais constituídas por multi-blocos.

No que diz respeito às malhas com um bloco, sete experimentos são realizados, envolvendo as seguintes estruturas: semi-trapezoidal, quadrada, retangular, com obstáculo, placa de orifício, forma de gancho e geometria cúspide. Respectivamente, essas malhas computacionais são apresentadas de maneira detalhada nas Seções 4.1 - 4.7.

Em relação às malhas multi-blocos, quinze experimentos são realizados, envolvendo as seguintes estruturas: placas paralelas com círculo imerso, hexagonal, NACA 64A 010, com obstáculo, placa de orifício, Lago Igapó I, golfinho, tartaruga, perfil de um rosto 2D, pássaro, cachorro, avião, Lago Luruaco, corpo humano e sapo. Respectivamente, essas malhas computacionais são apresentadas de maneira detalhada nas Seções 4.8 - 4.22 e estão ordenadas com base no número de blocos e na complexidade da geometria. Além disso, de acordo com as estruturas citadas, é importante notar que algumas malhas computacionais possuem geometrias reais.

Portanto, para cada experimento efetuado, demonstram-se e discutem-se os resultados obtidos referentes aos três processos explicados no Capítulo 3: a aquisição de pontos de fronteira, a geração da malha computacional em coordenadas generalizadas e a análise de qualidade da malha gerada. No último processo, utilizam-se as porcentagens desejadas $\bar{p} = 5, 10$ e 20% de elementos de menor qualidade em todos os experimentos. Vale salientar que “.” foi adotado como separador decimal neste trabalho.

Ainda, com o objetivo de facilitar a compreensão dos Algoritmos 6 e 7, a malha semi-trapezoidal, descrita na Seção 4.1, constitui um exemplo com foco no processo de análise de qualidade da malha computacional. Com exceção desse experimento, os resultados provenientes do Algoritmo 6 para o cálculo das métricas de qualidade não são apresentados, devido ao refinamento empregado no plano transformado das malhas dos demais experimentos. Contudo, apresentam-se os resultados do Algoritmo 7 para a identificação dos elementos de menor qualidade.

4.1 Malha Semi-trapezoidal

Nesta seção, apresenta-se um experimento como forma de auxílio ao entendimento do processo de análise de qualidade de malhas, aplicando os Algoritmos 6 e 7. A malha, nomeada como semi-trapezoidal, é ilustrada na Figura 11 e é constituída por 5 partições

na direção ξ e 3 partições na direção η , totalizando 24 nós e 15 elementos. A aquisição dessa malha computacional é baseada em um trabalho científico [30].

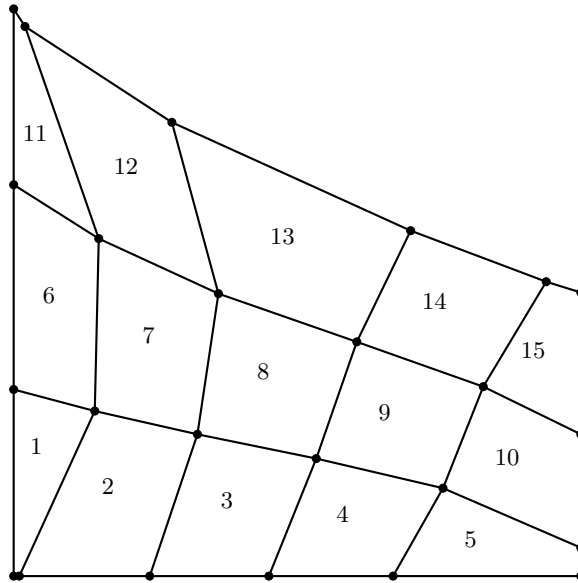


Figura 11 – Malha semi-trapezoidal. (Fonte: 30)

A Tabela 1 exibe as coordenadas dos nós de cada elemento da malha semi-trapezoidal [30], de acordo com as numerações dos elementos, presentes na Figura 11, e com as nomenclaturas adotadas, representadas na Figura 9.

Tabela 1 – Coordenadas dos nós da malha semi-trapezoidal

Elemento	i, j	$i, j + 1$	$i + 1, j + 1$	$i + 1, j$
1	(0.0000, 0.0000)	(0.0000, 3.2999)	(1.4359, 2.9164)	(0.1000, 0.0000)
2	(0.1000, 0.0000)	(1.4359, 2.9164)	(3.2422, 2.5007)	(2.4000, 0.0000)
3	(2.4000, 0.0000)	(3.2422, 2.5007)	(5.3370, 2.0776)	(4.5000, 0.0000)
4	(4.5000, 0.0000)	(5.3370, 2.0776)	(7.5773, 1.5565)	(6.6999, 0.0000)
5	(6.6999, 0.0000)	(7.5773, 1.5565)	(10.0000, 0.5000)	(10.0000, 0.0000)
6	(0.0000, 3.2999)	(0.0000, 6.9000)	(1.5184, 5.9551)	(1.4359, 2.9164)
7	(1.4359, 2.9164)	(1.5184, 5.9551)	(3.6106, 4.9824)	(3.2422, 2.5007)
8	(3.2422, 2.5007)	(3.6106, 4.9824)	(6.0566, 4.1341)	(5.3370, 2.0776)
9	(5.3370, 2.0776)	(6.0566, 4.1341)	(8.2839, 3.3448)	(7.5773, 1.5565)
10	(7.5773, 1.5565)	(8.2839, 3.3448)	(10.0000, 2.5000)	(10.0000, 0.5000)
11	(0.0000, 6.9000)	(0.0000, 10.0000)	(0.2000, 9.6999)	(1.5184, 5.9551)
12	(1.5184, 5.9551)	(0.2000, 9.6999)	(2.7999, 8.0000)	(3.6106, 4.9824)
13	(3.6106, 4.9824)	(2.7999, 8.0000)	(7.0000, 6.0999)	(6.0566, 4.1341)
14	(6.0566, 4.1341)	(7.0000, 6.0999)	(9.3999, 5.1999)	(8.2839, 3.3448)
15	(8.2839, 3.3448)	(9.3999, 5.1999)	(10.0000, 5.0000)	(10.0000, 2.5000)

Utilizando a malha semi-trapezoidal como parâmetro de entrada, executa-se o primeiro módulo para análise de qualidade, através do Algoritmo 6. Após o cálculo das métricas de qualidade para cada elemento da malha computacional, o tamanho das arestas

d_i , as razões $r_{i,j}$ entre cada aresta com as demais e os ângulos internos α_i são obtidos, produzindo as Tabelas 2 - 16.

Nas Tabelas 2 - 16, a tabela à esquerda apresenta os valores do tamanho das arestas d_i do elemento, calculados pela Equação (3.3). A tabela central descreve os valores das razões $r_{i,j}$ entre arestas, em que a linha corresponde ao numerador, e a coluna equivale ao denominador da Equação (3.4). Já a tabela à direita representa os valores dos ângulos internos α_i do elemento, encontrados pela Equação (3.5). Além disso, o valor destacado em vermelho indica a razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, e o valor destacado em verde, o ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad.

Tabela 2 – Tamanho das arestas, razões entre arestas e ângulos internos do elemento 1 da malha semi-trapezoidal

	d_i		d_1	d_2	d_3	d_4		α_i
d_1	3.2999	d_1		2.2203	1.0287	32.9990	α_1	1.5708
d_2	1.4862	d_2	0.4504		0.4633	14.8623	α_2	1.3098
d_3	3.2078	d_3	0.9721	2.1584		32.0781	α_3	1.4022
d_4	0.1000	d_4	0.0303	0.0673	0.0312		α_4	2.0003

Tabela 3 – Tamanho das arestas, razões entre arestas e ângulos internos do elemento 2 da malha semi-trapezoidal

	d_i		d_1	d_2	d_3	d_4		α_i
d_1	3.2078	d_1		1.7307	1.2157	1.3947	α_1	1.1413
d_2	1.8535	d_2	0.5778		0.7024	0.8059	α_2	1.7741
d_3	2.6387	d_3	0.8226	1.4236		1.1473	α_3	1.4721
d_4	2.3000	d_4	0.7170	1.2409	0.8716		α_4	1.8957

Tabela 4 – Tamanho das arestas, razões entre arestas e ângulos internos do elemento 3 da malha semi-trapezoidal

	d_i		d_1	d_2	d_3	d_4		α_i
d_1	2.6387	d_1		1.2347	1.1781	1.2565	α_1	1.2459
d_2	2.1371	d_2	0.8099		0.9541	1.0177	α_2	1.6964
d_3	2.2399	d_3	0.8488	1.0481		1.0666	α_3	1.3871
d_4	2.1000	d_4	0.7958	0.9826	0.9376		α_4	1.9538

Tabela 5 – Tamanho das arestas, razões entre arestas e ângulos internos do elemento 4 da malha semi-trapezoidal

	d_i		d_1	d_2	d_3	d_4		α_i
d_1	2.2399	d_1		0.9738	1.2536	1.0182	α_1	1.1878
d_2	2.3001	d_2	1.0269		1.2873	1.0456	α_2	1.7252
d_3	1.7868	d_3	0.7977	0.7768		0.8122	α_3	1.2860
d_4	2.1999	d_4	0.9822	0.9564	1.2312		α_4	2.0841

Tabela 6 – Tamanho das arestas, razões entre arestas e ângulos internos do elemento 5 da malha semi-trapezoidal

	d_i		d_1	d_2	d_3	d_4		α_i
d_1	1.7868	d_1		0.6760	3.5735	0.5414	α_1	1.0575
d_2	2.6430	d_2	1.4792		5.2861	0.8009	α_2	1.6729
d_3	0.5000	d_3	0.2798	0.1892		0.1515	α_3	1.9820
d_4	3.3001	d_4	1.8470	1.2486	6.6002		α_4	1.5708

Tabela 7 – Tamanho das arestas, razões entre arestas e ângulos internos do elemento 6 da malha semi-trapezoidal

	d_i		d_1	d_2	d_3	d_4		α_i
d_1	3.6001	d_1		2.0130	1.1843	2.4223	α_1	1.8318
d_2	1.7884	d_2	0.4968		0.5883	1.2033	α_2	1.0141
d_3	3.0398	d_3	0.8444	1.6997		2.0453	α_3	2.1003
d_4	1.4862	d_4	0.4128	0.8310	0.4889		α_4	1.3370

Tabela 8 – Tamanho das arestas, razões entre arestas e ângulos internos do elemento 7 da malha semi-trapezoidal

	d_i		d_1	d_2	d_3	d_4		α_i
d_1	3.0398	d_1		1.3175	1.2116	1.6400	α_1	1.7699
d_2	2.3073	d_2	0.7590		0.9196	1.2448	α_2	1.1627
d_3	2.5089	d_3	0.8253	1.0874		1.3536	α_3	1.8586
d_4	1.8535	d_4	0.6097	0.8033	0.7388		α_4	1.4920

Tabela 9 – Tamanho das arestas, razões entre arestas e ângulos internos do elemento 8 da malha semi-trapezoidal

	d_i		d_1	d_2	d_3	d_4		α_i
d_1	2.5089	d_1		0.9691	1.1515	1.1740	α_1	1.6227
d_2	2.5889	d_2	1.0319		1.1883	1.2114	α_2	1.3843
d_3	2.1788	d_3	0.8684	0.8416		1.0195	α_3	1.5680
d_4	2.1371	d_4	0.8518	0.8255	0.9809		α_4	1.7081

Tabela 10 – Tamanho das arestas, razões entre arestas e ângulos internos do elemento 9 da malha semi-trapezoidal

	d_i		d_1	d_2	d_3	d_4		α_i
d_1	2.1788	d_1		0.9220	1.1331	0.9472	α_1	1.4627
d_2	2.3630	d_2	1.0846		1.2289	1.0274	α_2	1.5668
d_3	1.9228	d_3	0.8825	0.8137		0.8360	α_3	1.5351
d_4	2.3001	d_4	1.0557	0.9734	1.1962		α_4	1.7186

Tabela 11 – Tamanho das arestas, razões entre arestas e ângulos internos do elemento 10 da malha semi-trapezoidal

	d_i		d_1	d_2	d_3	d_4		α_i
d_1	1.9228	d_1		1.0053	0.9614	0.7275	α_1	1.6057
d_2	1.9128	d_2	0.9948		0.9564	0.7237	α_2	1.4896
d_3	2.0000	d_3	1.0401	1.0456		0.7567	α_3	2.0282
d_4	2.6430	d_4	1.3746	1.3818	1.3215		α_4	1.1596

Tabela 12 – Tamanho das arestas, razões entre arestas e ângulos internos do elemento 11 da malha semi-trapezoidal

	d_i		d_1	d_2	d_3	d_4		α_i
d_1	3.1000	d_1		8.5959	0.7808	1.7334	α_1	2.1275
d_2	0.3606	d_2	0.1163		0.0908	0.2017	α_2	0.5878
d_3	3.9701	d_3	1.2807	11.0085		2.2199	α_3	2.8923
d_4	1.7884	d_4	0.5769	4.9590	0.4505		α_4	0.6756

Tabela 13 – Tamanho das arestas, razões entre arestas e ângulos internos do elemento 12 da malha semi-trapezoidal

	d_i		d_1	d_2	d_3	d_4		α_i
d_1	3.9701	d_1		1.2781	1.2706	1.7207	α_1	2.3445
d_2	3.1063	d_2	0.7824		0.9941	1.3463	α_2	0.6532
d_3	3.1246	d_3	0.7870	1.0059		1.3542	α_3	2.4123
d_4	2.3073	d_4	0.5812	0.7428	0.7384		α_4	0.8731

Tabela 14 – Tamanho das arestas, razões entre arestas e ângulos internos do elemento 13 da malha semi-trapezoidal

	d_i		d_1	d_2	d_3	d_4		α_i
d_1	3.1246	d_1		0.6778	1.4330	1.2069	α_1	2.1671
d_2	4.6099	d_2	1.4754		2.1142	1.7806	α_2	0.8835
d_3	2.1805	d_3	0.6978	0.4730		0.8422	α_3	1.5482
d_4	2.5889	d_4	0.8286	0.5616	1.1873		α_4	1.6844

Tabela 15 – Tamanho das arestas, razões entre arestas e ângulos internos do elemento 14 da malha semi-trapezoidal

	d_i		d_1	d_2	d_3	d_4		α_i
d_1	2.1805	d_1		0.8507	1.0072	0.9227	α_1	1.4639
d_2	2.5631	d_2	1.1755		1.1839	1.0847	α_2	1.6595
d_3	2.1649	d_3	0.9929	0.8446		0.9162	α_3	1.3880
d_4	2.3630	d_4	1.0837	0.9219	1.0915		α_4	1.7718

Tabela 16 – Tamanho das arestas, razões entre arestas e ângulos internos do elemento 15 da malha semi-trapezoidal

	d_i		d_1	d_2	d_3	d_4		α_i
d_1	2.1649	d_1		3.4227	0.8660	1.1318	α_1	1.4867
d_2	0.6325	d_2	0.2922		0.2530	0.3307	α_2	1.7908
d_3	2.5000	d_3	1.1548	3.9525		1.3070	α_3	1.8923
d_4	1.9128	d_4	0.8835	3.0241	0.7651		α_4	1.1133

Além dos resultados obtidos nas Tabelas 2 - 16, o Algoritmo 6 encontra também a área S , o perímetro P e o coeficiente de compacidade c de cada elemento da malha computacional, cujos valores são agrupados na Tabela 17. É importante lembrar que a área S , o perímetro P e o coeficiente de compacidade c são calculados, respectivamente, pela Equação (2.32), pelo somatório do tamanho das arestas d_i e pela Equação (3.6).

Tabela 17 – Área, perímetro e coeficiente de compacidade dos elementos da malha semi-trapezoidal

Elemento	S	P	c
1	2.5150	8.0939	26.0486
2	5.7874	10.0000	17.2790
3	4.9789	9.1157	16.6896
4	4.2574	8.5266	17.0771
5	3.1740	8.2299	21.3396
6	4.9306	9.9146	19.9362
7	5.5368	9.7095	17.0268
8	5.4976	9.4137	16.1193
9	4.7615	8.7647	16.1337
10	4.2556	8.4786	16.8924
11	2.5302	9.2191	33.5916
12	6.5099	12.5083	24.0337
13	8.3712	12.5039	18.6767
14	5.2898	9.2715	16.2504
15	2.8133	7.2102	18.4791

Finalmente, o Algoritmo 6 produz, como saída, um arquivo *.csv* contendo o conjunto total de elementos, associados à razão entre arestas r com maior disparidade em

relação a $r_{\square} = 1$, ao ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e ao coeficiente de compacidade c , como apresenta a Tabela 18.

Tabela 18 – Conjunto total de elementos e suas métricas de qualidade da malha semi-trapezoidal

Elemento	r	α	c
1	32.9990	2.0003	26.0486
2	1.7307	1.1413	17.2790
3	1.2565	1.9538	16.6896
4	1.2873	2.0841	17.0771
5	6.6002	1.0575	21.3396
6	2.4223	1.0141	19.9362
7	1.6400	1.1627	17.0268
8	1.2114	1.3843	16.1193
9	1.2289	1.7186	16.1337
10	1.3818	2.0282	16.8924
11	11.0085	2.8923	33.5916
12	1.7207	0.6532	24.0337
13	2.1142	0.8835	18.6767
14	1.1839	1.7718	16.2504
15	3.9525	1.1133	18.4791

Tendo posse do conjunto total de elementos e suas métricas de qualidade, assim como informações sobre o número de partições do plano transformado e as porcentagens desejadas \bar{p} , executa-se o segundo módulo para análise de qualidade, por meio do Algoritmo 7. Após a leitura dos dados de entrada, obtém-se que $|U'| = |U| = 15$ elementos, pois não há elementos em que $r = +\infty$ e $\alpha = \nexists$ no conjunto U . Em seguida, a variância de cada métrica de qualidade é calculada, resultando em $s_R^2 = 68.1947$, $s_A^2 = 0.3616$ e $s_C^2 = 23.6867$ para a malha semi-trapezoidal.

Uma vez que $s_R^2 \neq 0$, $s_A^2 \neq 0$ e $s_C^2 \neq 0$, calculam-se os valores mínimo m_{\min} e máximo m_{\max} das métricas de qualidade, cujos resultados estão presentes na Tabela 19.

Tabela 19 – Valores mínimo e máximo das métricas de qualidade da malha semi-trapezoidal

	r	α	c
min	1.0000	0.6532	16.0000
max	32.9990	2.8923	33.5916

Utilizando os valores encontrados na Tabela 19, normalizam-se os valores das métricas de qualidade no intervalo $[0, 10]$ através da Equação (3.10), transformando a Tabela 18 na Tabela 20.

Tabela 20 – Conjunto total de elementos e suas métricas de qualidade normalizadas da malha semi-trapezoidal

Elemento	r_{norm}	α_{norm}	c_{norm}
1	10.0000	6.0165	5.7122
2	0.2283	2.1797	0.7270
3	0.0802	5.8085	0.3920
4	0.0898	6.3906	0.6123
5	1.7501	1.8056	3.0353
6	0.4445	1.6119	2.2375
7	0.2000	2.2757	0.5837
8	0.0661	3.2653	0.0678
9	0.0715	4.7580	0.0760
10	0.1193	6.1412	0.5073
11	3.1278	10.0000	10.0000
12	0.2252	0.0000	4.5668
13	0.3482	1.0284	1.5216
14	0.0575	4.9959	0.1423
15	0.9227	2.0550	1.4092

Ainda, o valor das métricas avaliadas sobre um quadrado também é normalizado no intervalo $[0, 10]$ através da Equação (3.11), obtendo $r_{\text{norm}\square} = 0.0000$, $\alpha_{\text{norm}\square} = 4.0981$ e $c_{\text{norm}\square} = 0.0000$. Na sequência, calculam-se as variâncias $s_{M_{\text{norm}}}^2$ das métricas de qualidade normalizadas, resultando em $s_{R_{\text{norm}}}^2 = 6.6601$, $s_{A_{\text{norm}}}^2 = 7.2126$ e $s_{C_{\text{norm}}}^2 = 7.6541$.

Com as variâncias $s_{M_{\text{norm}}}^2$ calculadas, parte-se para o cálculo das tolerâncias mínima e máxima, obtendo $\epsilon_{\text{min}} = 0.0057$ e $\epsilon_{\text{max}} = 0.1987$. Iterando entre o intervalo de valores, constrói-se o diagrama de dispersão da tolerância ϵ_i pela quantidade de elementos $|E_{\cup_i}|$, o qual é ilustrado na Figura 12 pelos pontos em preto. A curva em verde representa a função de regressão não linear exponencial, encontrada na etapa posterior. No caso da malha semi-trapezoidal, o método de mínimos quadrados obteve, como resultado, os coeficientes $c_1 = 15.0875$ e $c_2 = -2.3533$.



Figura 12 – Diagrama de dispersão de ϵ_i por $|E_{U_i}|$ da malha semi-trapezoidal

Dados os coeficientes c_1 e c_2 da função de regressão não linear exponencial, calculam-se as tolerâncias desejadas $\bar{\epsilon}$, considerando $\bar{p} = 5, 10$ e 20% . As porcentagens utilizadas resultaram, respectivamente, em $\bar{\epsilon} = 1.2754, 0.9809$ e 0.6864 . Para cada valor de tolerância desejada $\bar{\epsilon}$, identificam-se os elementos de menor qualidade em relação à razão entre arestas $\overline{E_r}$, ao ângulo interno $\overline{E_\alpha}$ e ao coeficiente de compacidade $\overline{E_c}$. Além disso, por meio de operações envolvendo os conjuntos $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$, obtêm-se os conjuntos $\overline{E_U}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$ e $\overline{E_\cap}$. Finalmente, um número q é atribuído para cada categoria de elementos a partir dos conjuntos obtidos, como mostra a Tabela 21.

Tabela 21 – Conjunto total de elementos e valores de q da malha semi-trapezoidal

Elemento	q		
	$\bar{p} = 5\%$	$\bar{p} = 10\%$	$\bar{p} = 20\%$
1	17	17	17
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	11
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	0	0	0
11	20	20	20
12	18	18	18
13	0	0	0
14	0	0	0
15	0	0	0

Após a escrita dos resultados em um arquivo *.vtk*, a Figura 13 ilustra os elementos de menor qualidade identificados para a malha semi-trapezoidal, levando em consideração

$\bar{p} = 5, 10$ e 20% , respectivamente, nas Figuras 13a - 13c.

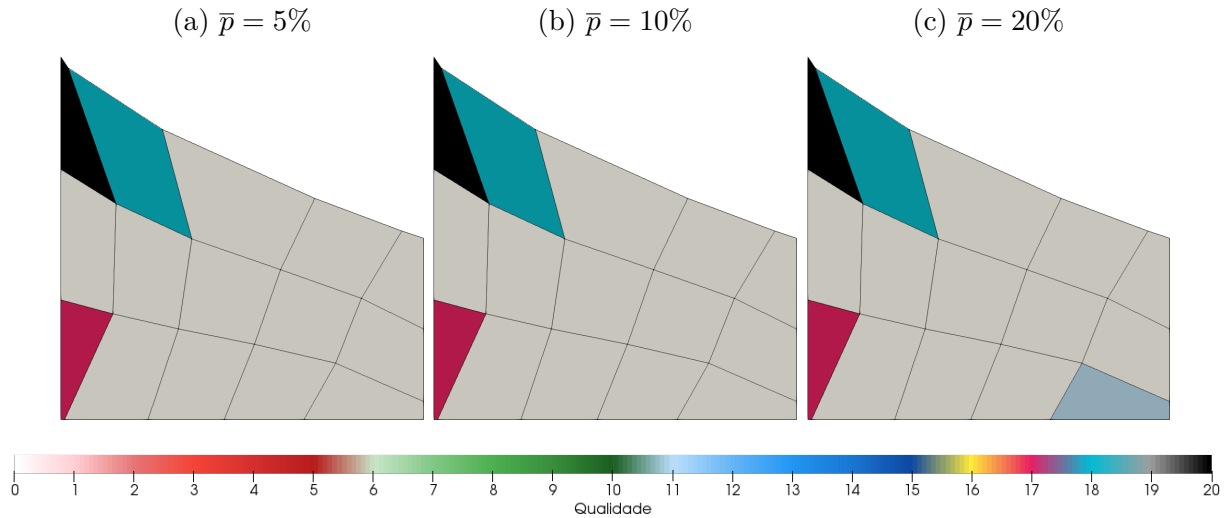


Figura 13 – Elementos de menor qualidade identificados para a malha semi-trapezoidal

Uma outra maneira para se observar os resultados encontrados da malha semi-trapezoidal pode ser representada na Tabela 22, na qual se apresentam as porcentagens desejadas \bar{p} e as tolerâncias desejadas $\bar{\epsilon}$ em relação às quantidades de elementos dos conjuntos $U' - \overline{E_U}$, $\overline{E_r}$, $\overline{E_\alpha}$, $\overline{E_c}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$, $U - U'$ e $\overline{E_\Omega}$.

Tabela 22 – Quantidades de elementos dos resultados obtidos para a malha semi-trapezoidal

\bar{p}	$\bar{\epsilon}$	$U' - \overline{E_U}$	$\overline{E_r}$	$\overline{E_\alpha}$	$\overline{E_c}$	$\overline{E_{r,\alpha}}$	$\overline{E_{r,c}}$	$\overline{E_{\alpha,c}}$	$ U - U' $	$\overline{E_\Omega}$
5%	1.2754	12	2	2	3	1	2	2	0	1
10%	0.9809	12	2	2	3	1	2	2	0	1
20%	0.6864	11	2	2	4	1	2	2	0	1

4.2 Malha Quadrada

Com o objetivo de verificar se o algoritmo possui possíveis inconsistências nos resultados, duas malhas simples são modeladas e analisadas: a primeira consiste em uma malha quadrada com elementos em formato quadrado e é apresentada nesta seção, e a segunda constitui uma malha retangular com elementos em formato retangular e é descrita na Seção 4.3.

Na malha quadrada, os pontos de fronteira são adquiridos manualmente, através da visualização gráfica da localização dos pontos no *software* Geogebra [59]. Esses pontos de fronteira são ilustrados na Figura 14 e são divididos em 1 partição na direção x e 1 partição na direção y , totalizando 4 nós.



Figura 14 – Pontos de fronteira da malha quadrada

Por meio do número de partições do plano físico e dos pontos de contorno, executa-se o algoritmo para geração da malha computacional em coordenadas generalizadas. A malha computacional do quadrado é ilustrada na Figura 15 e é constituída por 10 partições na direção ξ e 10 partições na direção η , totalizando 121 nós e 100 elementos.

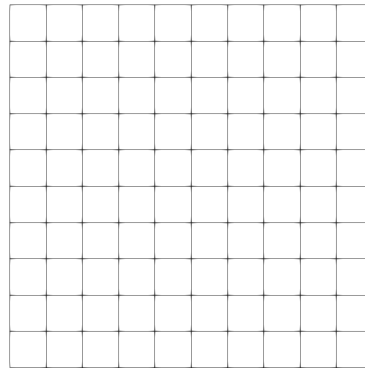


Figura 15 – Malha computacional do quadrado em coordenadas generalizadas

Utilizando a malha quadrada como parâmetro de entrada, executa-se o primeiro módulo para análise de qualidade, através do Algoritmo 6. Após o cálculo das métricas de qualidade para cada elemento da malha computacional, obtém-se o conjunto total de elementos, associados à razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, ao ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e ao coeficiente de compacidade c .

Tendo posse do conjunto total de elementos e suas métricas de qualidade, assim como informações sobre o número de partições do plano transformado e as porcentagens desejadas \bar{p} , executa-se o segundo módulo para análise de qualidade, por meio do Algoritmo 7. Após a leitura dos dados de entrada, obtém-se que $|U'| = |U| = 100$ elementos, pois não há elementos em que $r = +\infty$ e $\alpha = \frac{\pi}{2}$ no conjunto U . Em seguida, a variância de cada métrica de qualidade é calculada, resultando em $s_R^2 = 0.0000$, $s_A^2 = 0.0000$ e $s_C^2 = 0.0000$ para a malha quadrada.

Uma vez que $s_R^2 = s_A^2 = s_C^2 = 0$, verifica-se se $r_k = r_{\square}$, $\alpha_k = \alpha_{\square}$ e $c_k = c_{\square}$ para todo $k = 1, \dots, |M|$, obtendo que todos os elementos da malha computacional são idênticos entre si e iguais a um quadrado. Assim, a execução do Algoritmo 7 é interrompida, e os resultados são escritos em um arquivo *.vtk*. A Figura 16 ilustra que nenhum elemento de menor qualidade foi identificado para a malha quadrada, independente das porcentagens desejadas \bar{p} levadas em consideração.

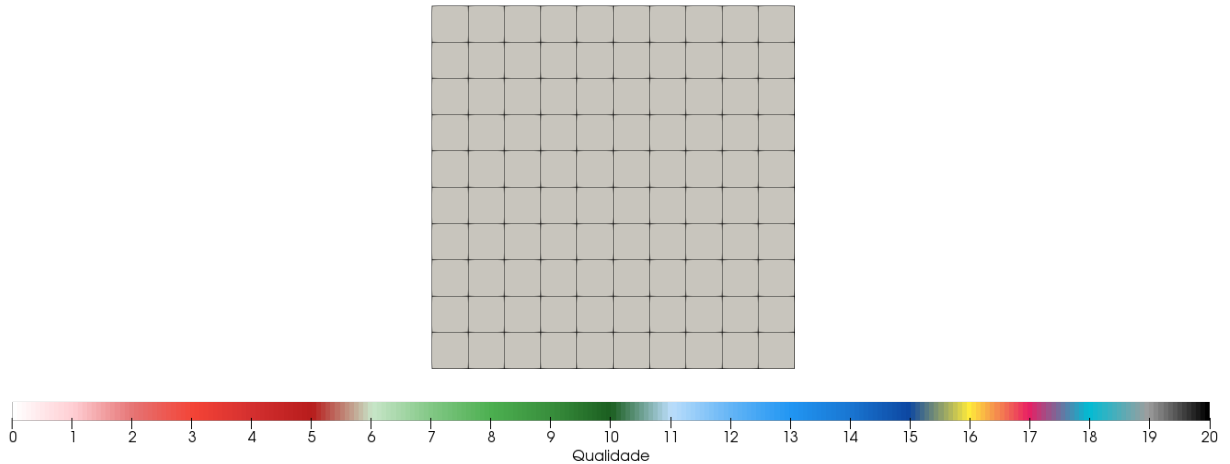


Figura 16 – Nenhum elemento de menor qualidade identificado para a malha quadrada

De acordo com a Figura 16, é possível observar que nenhum elemento da malha quadrada apresenta problemas de qualidade, como esperado. A razão para tal resultado reside no fato de que todos os elementos da malha computacional são idênticos entre si e possuem um formato equivalente ao de um quadrado.

4.3 Malha Retangular

Na malha retangular, os pontos de fronteira são adquiridos manualmente, através da visualização gráfica da localização dos pontos no *software* Geogebra [59]. Esses pontos de fronteira são ilustrados na Figura 17 e são divididos em 1 partição na direção x e 1 partição na direção y , totalizando 4 nós.



Figura 17 – Pontos de fronteira da malha retangular

Por meio do número de partições do plano físico e dos pontos de contorno, executa-se o algoritmo para geração da malha computacional em coordenadas generalizadas. A malha computacional do retângulo é ilustrada na Figura 18 e é constituída por 20 partições na direção ξ e 20 partições na direção η , totalizando 441 nós e 400 elementos.

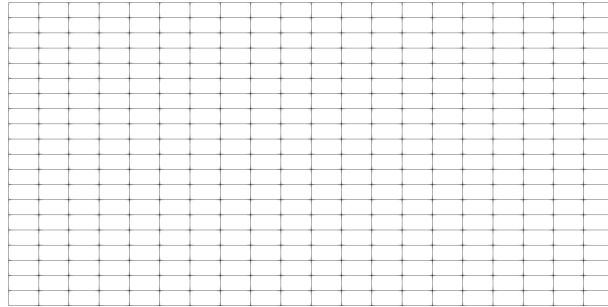


Figura 18 – Malha computacional do retângulo em coordenadas generalizadas

Utilizando a malha retangular como parâmetro de entrada, executa-se o primeiro módulo para análise de qualidade, através do Algoritmo 6. Após o cálculo das métricas de qualidade para cada elemento da malha computacional, obtém-se o conjunto total de elementos, associados à razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, ao ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e ao coeficiente de compacidade c .

Tendo posse do conjunto total de elementos e suas métricas de qualidade, assim como informações sobre o número de partições do plano transformado e as porcentagens desejadas \bar{p} , executa-se o segundo módulo para análise de qualidade, por meio do Algoritmo 7. Após a leitura dos dados de entrada, obtém-se que $|U'| = |U| = 400$ elementos, pois não há elementos em que $r = +\infty$ e $\alpha = \nexists$ no conjunto U . Em seguida, a variância de cada métrica de qualidade é calculada, resultando em $s_R^2 = 0.0000$, $s_A^2 = 0.0000$ e $s_C^2 = 0.0000$ para a malha retangular.

Uma vez que $s_R^2 = s_A^2 = s_C^2 = 0$, verifica-se se $r_k = r_{\square}$, $\alpha_k = \alpha_{\square}$ e $c_k = c_{\square}$ para todo $k = 1, \dots, |M|$, obtendo que todos os elementos da malha computacional são idênticos entre si e diferentes de um quadrado. Assim, a execução do Algoritmo 7 é interrompida, e os resultados são escritos em um arquivo *.vtk*. A Figura 19 ilustra que nenhum elemento de menor qualidade foi identificado para a malha retangular, independente das porcentagens desejadas \bar{p} levadas em consideração.

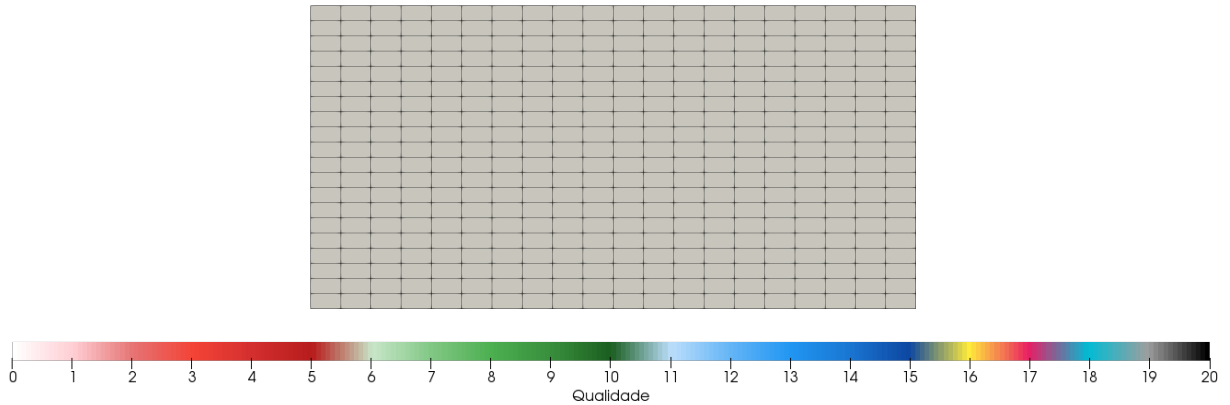


Figura 19 – Nenhum elemento de menor qualidade identificado para a malha retangular

De acordo com a Figura 19, é possível observar que nenhum elemento da malha retangular apresenta problemas de qualidade, como esperado. A razão para tal resultado reside no fato de que todos os elementos da malha computacional são idênticos entre si, apesar de possuírem um formato diferente de um quadrado, impossibilitando afirmar o nível de menor qualidade dos elementos.

4.4 Malha com Obstáculo

Nesta seção, modela-se e analisa-se uma malha retangular incluindo um obstáculo horizontalmente centralizado na parte inferior da geometria. Nesse experimento, a aquisição dos pontos de fronteira é baseada em um trabalho científico [30]. Esses pontos de fronteira são ilustrados na Figura 20 e são divididos em 5 partições na direção x e 1 partição na direção y , totalizando 12 nós.

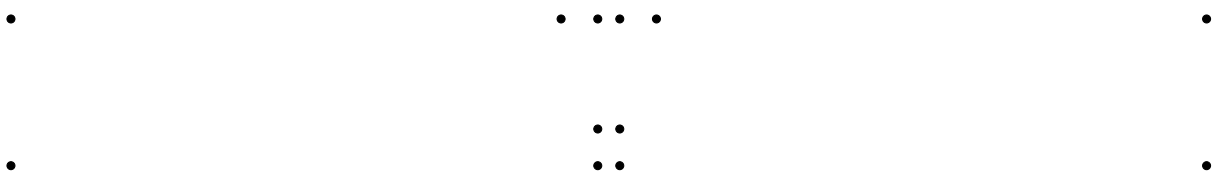


Figura 20 – Pontos de fronteira da malha com obstáculo. (Fonte: 30)

Por meio do número de partições do plano físico e dos pontos de contorno, executa-se o algoritmo para geração da malha computacional em coordenadas generalizadas. A malha computacional com obstáculo é ilustrada na Figura 21 e é constituída por 80 partições na direção ξ e 16 partições na direção η , totalizando 1377 nós e 1280 elementos.

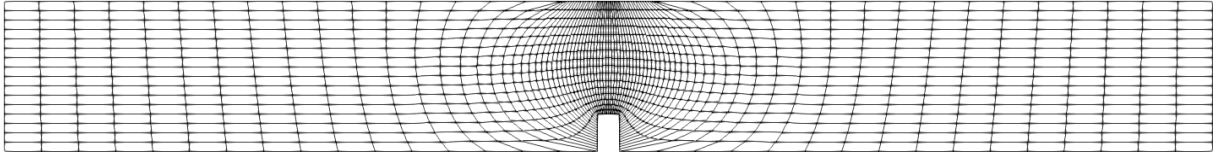


Figura 21 – Malha computacional com obstáculo em coordenadas generalizadas. (Fonte: 30)

Utilizando a malha com obstáculo como parâmetro de entrada, executa-se o primeiro módulo para análise de qualidade, através do Algoritmo 6. Após o cálculo das métricas de qualidade para cada elemento da malha computacional, obtém-se o conjunto total de elementos, associados à razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, ao ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e ao coeficiente de compacidade c .

Tendo posse do conjunto total de elementos e suas métricas de qualidade, assim como informações sobre o número de partições do plano transformado e as porcentagens desejadas \bar{p} , executa-se o segundo módulo para análise de qualidade, por meio do Algoritmo 7. Após a leitura dos dados de entrada, obtém-se que $|U'| = |U| = 1280$ elementos, pois não há elementos em que $r = +\infty$ e $\alpha = \nexists$ no conjunto U . Em seguida, a variância de cada métrica de qualidade é calculada, resultando em $s_R^2 = 2.7229$, $s_A^2 = 0.1300$ e $s_C^2 = 38.4913$ para a malha com obstáculo.

Uma vez que $s_R^2 \neq 0$, $s_A^2 \neq 0$ e $s_C^2 \neq 0$, calculam-se os valores mínimo m_{\min} e máximo m_{\max} das métricas de qualidade, cujos resultados estão presentes na Tabela 23.

Tabela 23 – Valores mínimo e máximo das métricas de qualidade da malha com obstáculo

	r	α	c
min	1.0000	0.1952	16.0000
max	23.9890	2.8845	83.1579

Utilizando os valores encontrados na Tabela 23, normalizam-se os valores das métricas de qualidade no intervalo $[0, 10]$. Ainda, o valor das métricas avaliadas sobre um quadrado também é normalizado no intervalo $[0, 10]$, obtendo $r_{\text{norm}\square} = 0.0000$, $\alpha_{\text{norm}\square} = 5.1151$ e $c_{\text{norm}\square} = 0.0000$. Na sequência, calculam-se as variâncias $s_{M_{\text{norm}}}^2$ das métricas de qualidade normalizadas, resultando em $s_{R_{\text{norm}}}^2 = 0.5152$, $s_{A_{\text{norm}}}^2 = 1.7976$ e $s_{C_{\text{norm}}}^2 = 0.8534$.

Com as variâncias $s_{M_{\text{norm}}}^2$ calculadas, parte-se para o cálculo das tolerâncias mínima e máxima, obtendo $\epsilon_{\min} = 0.0000$ e $\epsilon_{\max} = 0.0643$. Iterando entre o intervalo de valores, constrói-se o diagrama de dispersão de ϵ_i por $|E_{U_i}|$, o qual é ilustrado na Figura 22 pelos pontos em preto. A curva em verde representa a função de regressão não linear exponencial, encontrada na etapa posterior. No caso da malha com obstáculo, o método

de mínimos quadrados obteve, como resultado, os coeficientes $c_1 = 338.7436$ e $c_2 = -123.4856$.

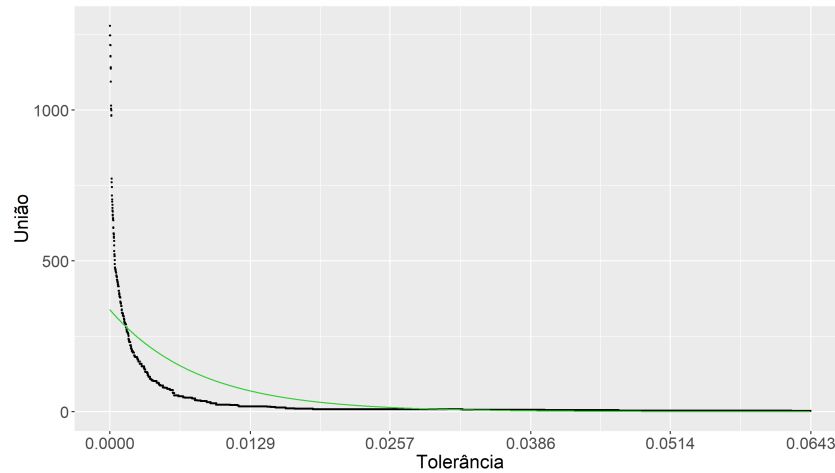


Figura 22 – Diagrama de dispersão de ϵ_i por $|E_{U_i}|$ da malha com obstáculo

Dados os coeficientes c_1 e c_2 da função de regressão não linear exponencial, calculam-se as tolerâncias desejadas $\bar{\epsilon}$, considerando $\bar{p} = 5, 10$ e 20% . As porcentagens utilizadas resultaram, respectivamente, em $\bar{\epsilon} = 0.0135, 0.0079$ e 0.0023 . Para cada valor de tolerância desejada $\bar{\epsilon}$, identificam-se os elementos de menor qualidade em relação à razão entre arestas $\overline{E_r}$, ao ângulo interno $\overline{E_\alpha}$ e ao coeficiente de compacidade $\overline{E_c}$. Além disso, por meio de operações envolvendo os conjuntos $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$, obtêm-se os conjuntos $\overline{E_U}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$ e $\overline{E_\cap}$. Finalmente, um número q é atribuído para cada categoria de elementos a partir dos conjuntos obtidos.

Após a escrita dos resultados em um arquivo *.vtk*, a Figura 23 ilustra os elementos de menor qualidade identificados para a malha com obstáculo, levando em consideração $\bar{p} = 5, 10$ e 20% , respectivamente, nas Figuras 23a - 23c.

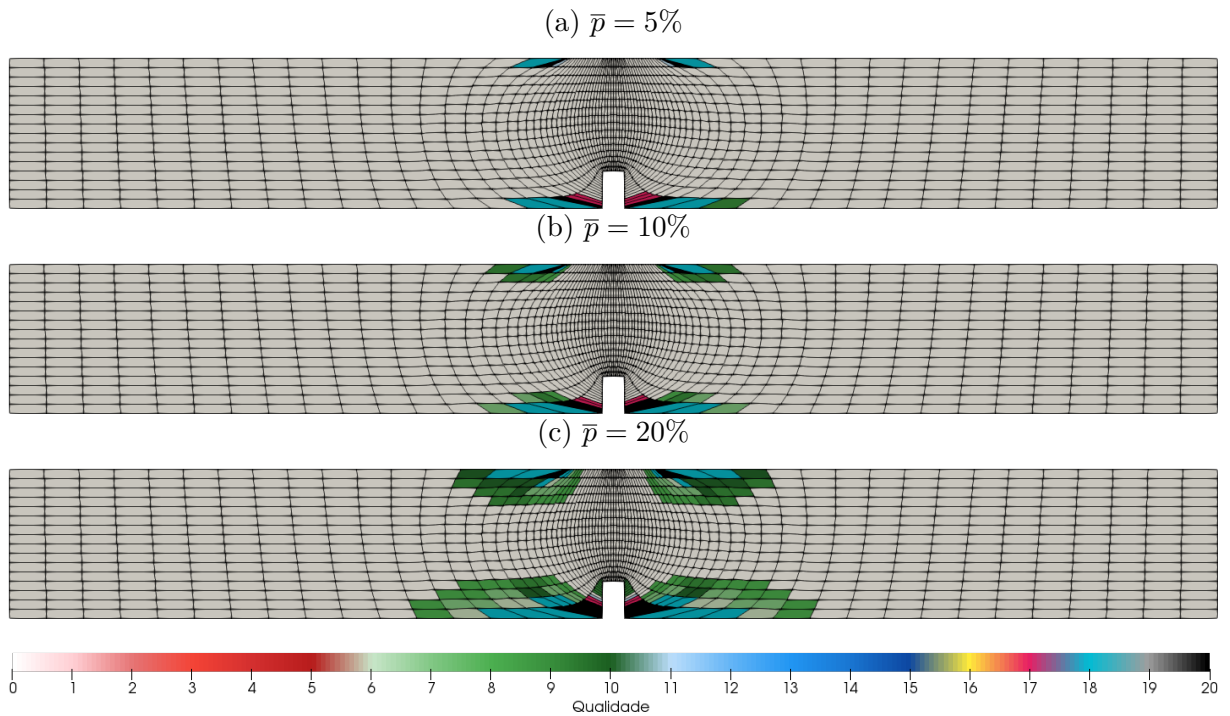


Figura 23 – Elementos de menor qualidade identificados para a malha com obstáculo

Uma outra maneira para se observar os resultados encontrados da malha com obstáculo pode ser representada na Tabela 24, na qual se apresentam as porcentagens desejadas \bar{p} e as tolerâncias desejadas $\bar{\epsilon}$ em relação às quantidades de elementos dos conjuntos $U' - \overline{E_U}$, $\overline{E_r}$, $\overline{E_\alpha}$, $\overline{E_c}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$, $U - U'$ e $\overline{E_\Omega}$.

Tabela 24 – Quantidades de elementos dos resultados obtidos para a malha com obstáculo

\bar{p}	$\bar{\epsilon}$	$U' - \overline{E_U}$	$\overline{E_r}$	$\overline{E_\alpha}$	$\overline{E_c}$	$\overline{E_{r,\alpha}}$	$\overline{E_{r,c}}$	$\overline{E_{\alpha,c}}$	$ U - U' $	$\overline{E_\Omega}$
5%	0.0135	1262	9	11	17	4	9	10	0	4
10%	0.0079	1241	11	35	19	7	11	15	0	7
20%	0.0023	1093	19	181	43	15	19	37	0	15

De acordo com a Figura 23 e com a Tabela 24, é possível observar que elementos de menor qualidade em relação a todas as métricas de avaliação foram identificados para a malha com obstáculo, considerando as porcentagens utilizadas \bar{p} . Além disso, destaca-se o número de elementos pertencentes aos conjuntos formados pelas intersecções entre $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$. Para $\bar{p} = 20\%$, percebem-se 181 elementos de menor qualidade em relação ao ângulo interno, quantidade em evidência quando comparada com as demais métricas de avaliação. Os elementos detectados localizam-se não apenas nas proximidades do obstáculo, como esperado, mas também na parte superior da geometria, a qual sofreu influência do obstáculo no processo de geração da malha computacional.

4.5 Placa de Orifício

Nesta seção, modela-se e analisa-se uma placa de orifício, isto é, uma malha retangular com dois obstáculos horizontalmente centralizados nas partes superior e inferior da geometria. Nesse experimento, a aquisição dos pontos de fronteira é baseada em um trabalho científico [30]. Esses pontos de fronteira são ilustrados na Figura 24 e são divididos em 5 partições na direção x e 1 partição na direção y , totalizando 12 nós.

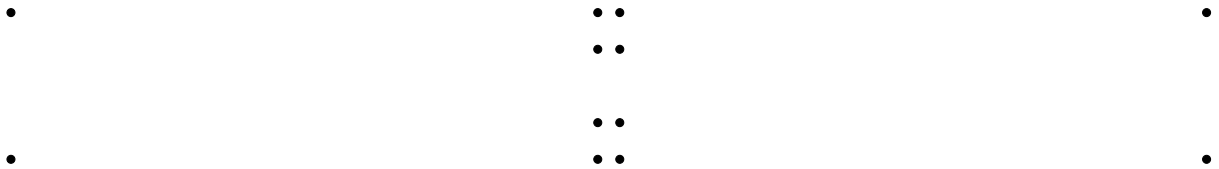


Figura 24 – Pontos de fronteira da placa de orifício. (Fonte: 30)

Por meio do número de partições do plano físico e dos pontos de contorno, executa-se o algoritmo para geração da malha computacional em coordenadas generalizadas. A malha computacional da placa de orifício é ilustrada na Figura 25 e é constituída por 80 partições na direção ξ e 16 partições na direção η , totalizando 1377 nós e 1280 elementos.

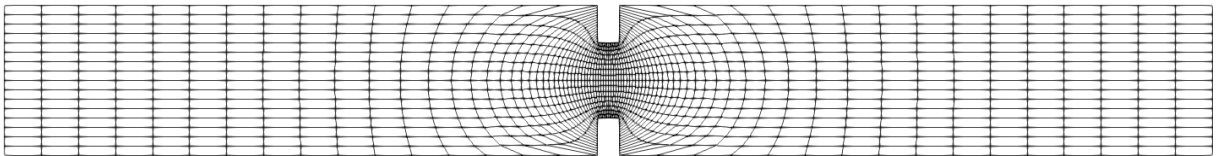


Figura 25 – Malha computacional da placa de orifício em coordenadas generalizadas. (Fonte: 30)

Utilizando a placa de orifício como parâmetro de entrada, executa-se o primeiro módulo para análise de qualidade, através do Algoritmo 6. Após o cálculo das métricas de qualidade para cada elemento da malha computacional, obtém-se o conjunto total de elementos, associados à razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, ao ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e ao coeficiente de compacidade c .

Tendo posse do conjunto total de elementos e suas métricas de qualidade, assim como informações sobre o número de partições do plano transformado e as porcentagens desejadas \bar{p} , executa-se o segundo módulo para análise de qualidade, por meio do Algoritmo 7. Após a leitura dos dados de entrada, obtém-se que $|U'| = |U| = 1280$ elementos, pois não há elementos em que $r = +\infty$ e $\alpha = \nexists$ no conjunto U . Em seguida, a variância de cada métrica de qualidade é calculada, resultando em $s_R^2 = 3.3765$, $s_A^2 = 0.1291$ e $s_C^2 = 45.7589$ para a placa de orifício.

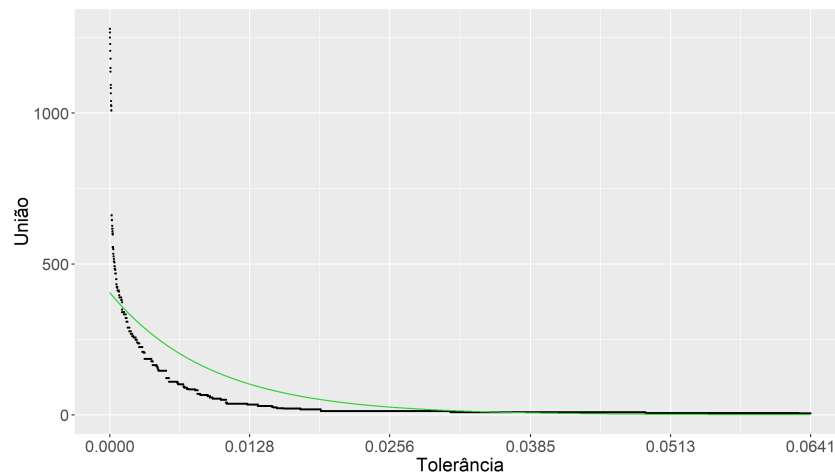
Uma vez que $s_R^2 \neq 0$, $s_A^2 \neq 0$ e $s_C^2 \neq 0$, calculam-se os valores mínimo m_{\min} e máximo m_{\max} das métricas de qualidade, cujos resultados estão presentes na Tabela 25.

Tabela 25 – Valores mínimo e máximo das métricas de qualidade da placa de orifício

	r	α	c
min	1.0000	0.1968	16.0000
max	22.4736	2.6689	79.5555

Utilizando os valores encontrados na Tabela 25, normalizam-se os valores das métricas de qualidade no intervalo $[0, 10]$. Ainda, o valor das métricas avaliadas sobre um quadrado também é normalizado no intervalo $[0, 10]$, obtendo $r_{\text{norm}\square} = 0.0000$, $\alpha_{\text{norm}\square} = 5.5580$ e $c_{\text{norm}\square} = 0.0000$. Na sequência, calculam-se as variâncias $s_{M_{\text{norm}}}^2$ das métricas de qualidade normalizadas, resultando em $s_{R_{\text{norm}}}^2 = 0.7322$, $s_{A_{\text{norm}}}^2 = 2.1131$ e $s_{C_{\text{norm}}}^2 = 1.1328$.

Com as variâncias $s_{M_{\text{norm}}}^2$ calculadas, parte-se para o cálculo das tolerâncias mínima e máxima, obtendo $\epsilon_{\text{min}} = 0.0000$ e $\epsilon_{\text{max}} = 0.0641$. Iterando entre o intervalo de valores, constrói-se o diagrama de dispersão de ϵ_i por $|E_{\cup_i}|$, o qual é ilustrado na Figura 26 pelos pontos em preto. A curva em verde representa a função de regressão não linear exponencial, encontrada na etapa posterior. No caso da placa de orifício, o método de mínimos quadrados obteve, como resultado, os coeficientes $c_1 = 405.3761$ e $c_2 = -107.9152$.

Figura 26 – Diagrama de dispersão de ϵ_i por $|E_{\cup_i}|$ da placa de orifício

Dados os coeficientes c_1 e c_2 da função de regressão não linear exponencial, calculam-se as tolerâncias desejadas $\bar{\epsilon}$, considerando $\bar{p} = 5, 10$ e 20% . As porcentagens utilizadas resultaram, respectivamente, em $\bar{\epsilon} = 0.0171, 0.0107$ e 0.0043 . Para cada valor de tolerância desejada $\bar{\epsilon}$, identificam-se os elementos de menor qualidade em relação à razão entre arestas $\overline{E_r}$, ao ângulo interno $\overline{E_\alpha}$ e ao coeficiente de compacidade $\overline{E_c}$. Além disso, por meio de operações envolvendo os conjuntos $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$, obtêm-se os conjuntos $\overline{E_{\cup}}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$ e $\overline{E_{\cap}}$. Finalmente, um número q é atribuído para cada categoria de elementos a partir dos conjuntos obtidos.

Após a escrita dos resultados em um arquivo *.vtk*, a Figura 27 ilustra os elementos de menor qualidade identificados para a placa de orifício, levando em consideração $\bar{p} = 5, 10$ e 20% , respectivamente, nas Figuras 27a - 27c.

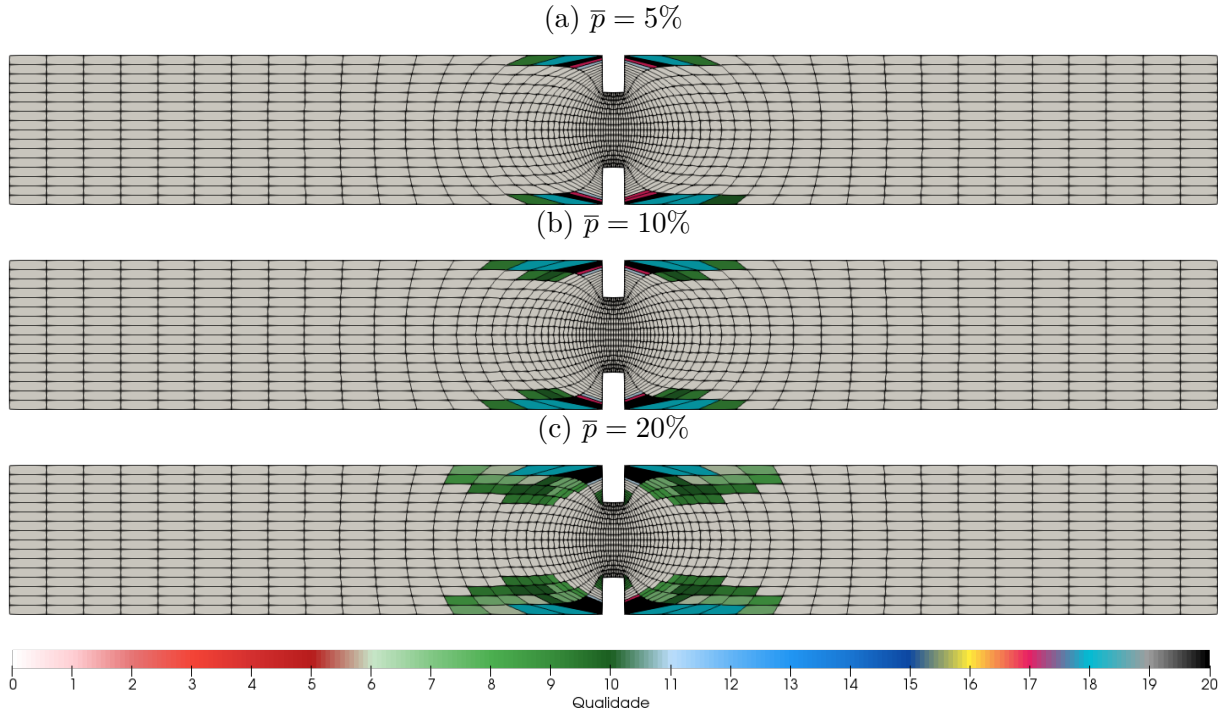


Figura 27 – Elementos de menor qualidade identificados para a placa de orifício

Uma outra maneira para se observar os resultados encontrados da placa de orifício pode ser representada na Tabela 26, na qual se apresentam as porcentagens desejadas \bar{p} e as tolerâncias desejadas $\bar{\epsilon}$ em relação às quantidades de elementos dos conjuntos $U' - \overline{E_U}$, $\overline{E_r}$, $\overline{E_\alpha}$, $\overline{E_c}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$, $U - U'$ e $\overline{E_\cap}$.

Tabela 26 – Quantidades de elementos dos resultados obtidos para a placa de orifício

\bar{p}	$\bar{\epsilon}$	$U' - \overline{E_U}$	$\overline{E_r}$	$\overline{E_\alpha}$	$\overline{E_c}$	$\overline{E_{r,\alpha}}$	$\overline{E_{r,c}}$	$\overline{E_{\alpha,c}}$	$ U - U' $	$\overline{E_\cap}$
5%	0.0171	1259	9	13	17	4	9	9	0	4
10%	0.0107	1239	13	33	25	9	13	17	0	9
20%	0.0043	1119	21	157	33	20	21	29	0	20

De acordo com a Figura 27 e com a Tabela 26, é possível observar que elementos de menor qualidade em relação a todas as métricas de avaliação foram identificados para a placa de orifício, considerando as porcentagens utilizadas \bar{p} . Além disso, para $\bar{p} = 10$ e 20% , destaca-se o número de elementos pertencentes aos conjuntos formados pelas intersecções entre $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$. Para $\bar{p} = 20\%$, percebem-se 157 elementos de menor qualidade em relação ao ângulo interno, quantidade em evidência quando comparada com as demais métricas de avaliação. Como esperado, os elementos detectados localizam-se nas proximidades dos dois obstáculos presentes na malha computacional.

4.6 Forma de Gancho

Nesta seção, modela-se e analisa-se a forma de um gancho, isto é, a região interior entre duas semicircunferências. As circunferências que formam as semicircunferências possuem raios e centros distintos, sendo que a circunferência de raio menor está contida na circunferência de raio maior. Nesse experimento, os pontos de fronteira são adquiridos de maneira automatizada, utilizando conceitos de geometria. Esses pontos de fronteira são ilustrados na Figura 28 e são divididos em 36 partições na direção x e 9 partições na direção y , totalizando 90 nós.

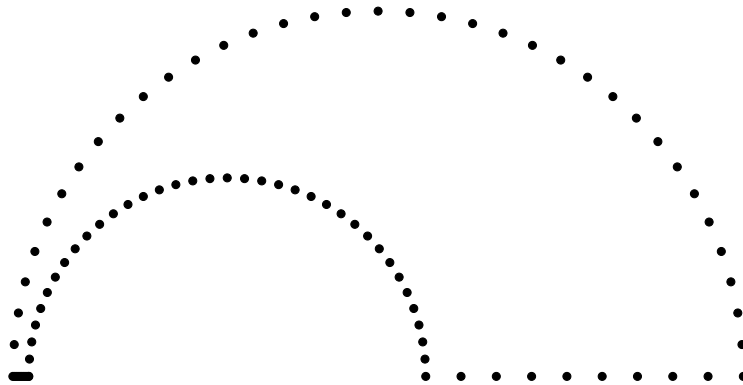


Figura 28 – Pontos de fronteira da forma de gancho

Por meio do número de partições do plano físico e dos pontos de contorno, executa-se o algoritmo para geração da malha computacional em coordenadas generalizadas. A malha computacional da forma de gancho é ilustrada na Figura 29 e é constituída por 36 partições na direção ξ e 9 partições na direção η , totalizando 370 nós e 324 elementos.

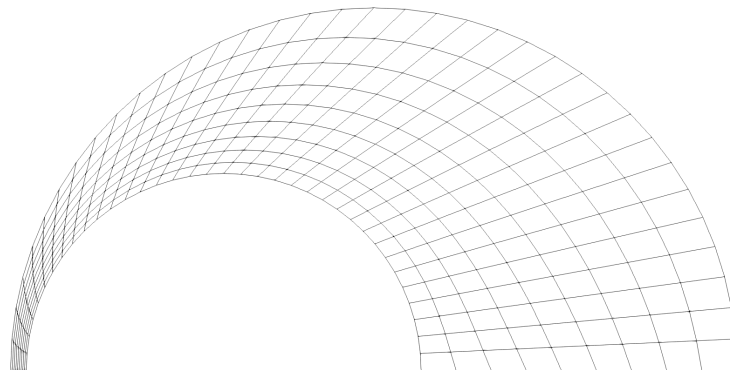


Figura 29 – Malha computacional da forma de gancho em coordenadas generalizadas

Utilizando a forma de gancho como parâmetro de entrada, executa-se o primeiro módulo para análise de qualidade, através do Algoritmo 6. Após o cálculo das métricas de qualidade para cada elemento da malha computacional, obtém-se o conjunto total de elementos, associados à razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$,

ao ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e ao coeficiente de compacidade c .

Tendo posse do conjunto total de elementos e suas métricas de qualidade, assim como informações sobre o número de partições do plano transformado e as porcentagens desejadas \bar{p} , executa-se o segundo módulo para análise de qualidade, por meio do Algoritmo 7. Após a leitura dos dados de entrada, obtém-se que $|U'| = |U| = 324$ elementos, pois não há elementos em que $r = +\infty$ e $\alpha = \nexists$ no conjunto U . Em seguida, a variância de cada métrica de qualidade é calculada, resultando em $s_R^2 = 7.0273$, $s_A^2 = 0.6155$ e $s_C^2 = 215.3621$ para a forma de gancho.

Uma vez que $s_R^2 \neq 0$, $s_A^2 \neq 0$ e $s_C^2 \neq 0$, calculam-se os valores mínimo m_{\min} e máximo m_{\max} das métricas de qualidade, cujos resultados estão presentes na Tabela 27.

Tabela 27 – Valores mínimo e máximo das métricas de qualidade da forma de gancho

	r	α	c
min	1.0000	0.3063	16.0000
max	18.0584	2.7551	79.7526

Utilizando os valores encontrados na Tabela 27, normalizam-se os valores das métricas de qualidade no intervalo $[0, 10]$. Ainda, o valor das métricas avaliadas sobre um quadrado também é normalizado no intervalo $[0, 10]$, obtendo $r_{\text{norm}\square} = 0.0000$, $\alpha_{\text{norm}\square} = 5.1638$ e $c_{\text{norm}\square} = 0.0000$. Na sequência, calculam-se as variâncias $s_{M_{\text{norm}}}^2$ das métricas de qualidade normalizadas, resultando em $s_{R_{\text{norm}}}^2 = 2.4150$, $s_{A_{\text{norm}}}^2 = 10.2647$ e $s_{C_{\text{norm}}}^2 = 5.2987$.

Com as variâncias $s_{M_{\text{norm}}}^2$ calculadas, parte-se para o cálculo das tolerâncias mínima e máxima, obtendo $\epsilon_{\min} = 0.0001$ e $\epsilon_{\max} = 0.1836$. Iterando entre o intervalo de valores, constrói-se o diagrama de dispersão de ϵ_i por $|E_{U_i}|$, o qual é ilustrado na Figura 30 pelos pontos em preto. A curva em verde representa a função de regressão não linear exponencial, encontrada na etapa posterior. No caso da forma de gancho, o método de mínimos quadrados obteve, como resultado, os coeficientes $c_1 = 246.8511$ e $c_2 = -27.5937$.

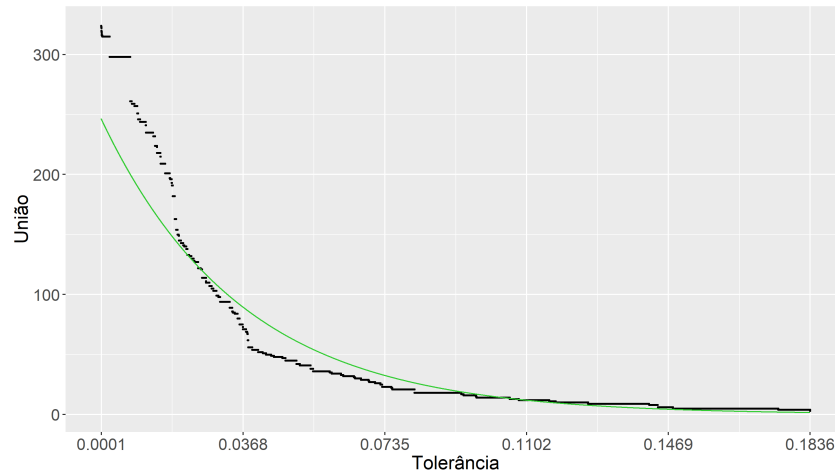


Figura 30 – Diagrama de dispersão de ϵ_i por $|E_{U_i}|$ da forma de gancho

Dados os coeficientes c_1 e c_2 da função de regressão não linear exponencial, calculam-se as tolerâncias desejadas $\bar{\epsilon}$, considerando $\bar{p} = 5, 10$ e 20% . As porcentagens utilizadas resultaram, respectivamente, em $\bar{\epsilon} = 0.0987, 0.0736$ e 0.0485 . Para cada valor de tolerância desejada $\bar{\epsilon}$, identificam-se os elementos de menor qualidade em relação à razão entre arestas $\overline{E_r}$, ao ângulo interno $\overline{E_\alpha}$ e ao coeficiente de compacidade $\overline{E_c}$. Além disso, por meio de operações envolvendo os conjuntos $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$, obtêm-se os conjuntos $\overline{E_U}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$ e $\overline{E_\Gamma}$. Finalmente, um número q é atribuído para cada categoria de elementos a partir dos conjuntos obtidos.

Após a escrita dos resultados em um arquivo *.vtk*, a Figura 31 ilustra os elementos de menor qualidade identificados para a forma de gancho, levando em consideração $\bar{p} = 5, 10$ e 20% , respectivamente, nas Figuras 31a - 31c.

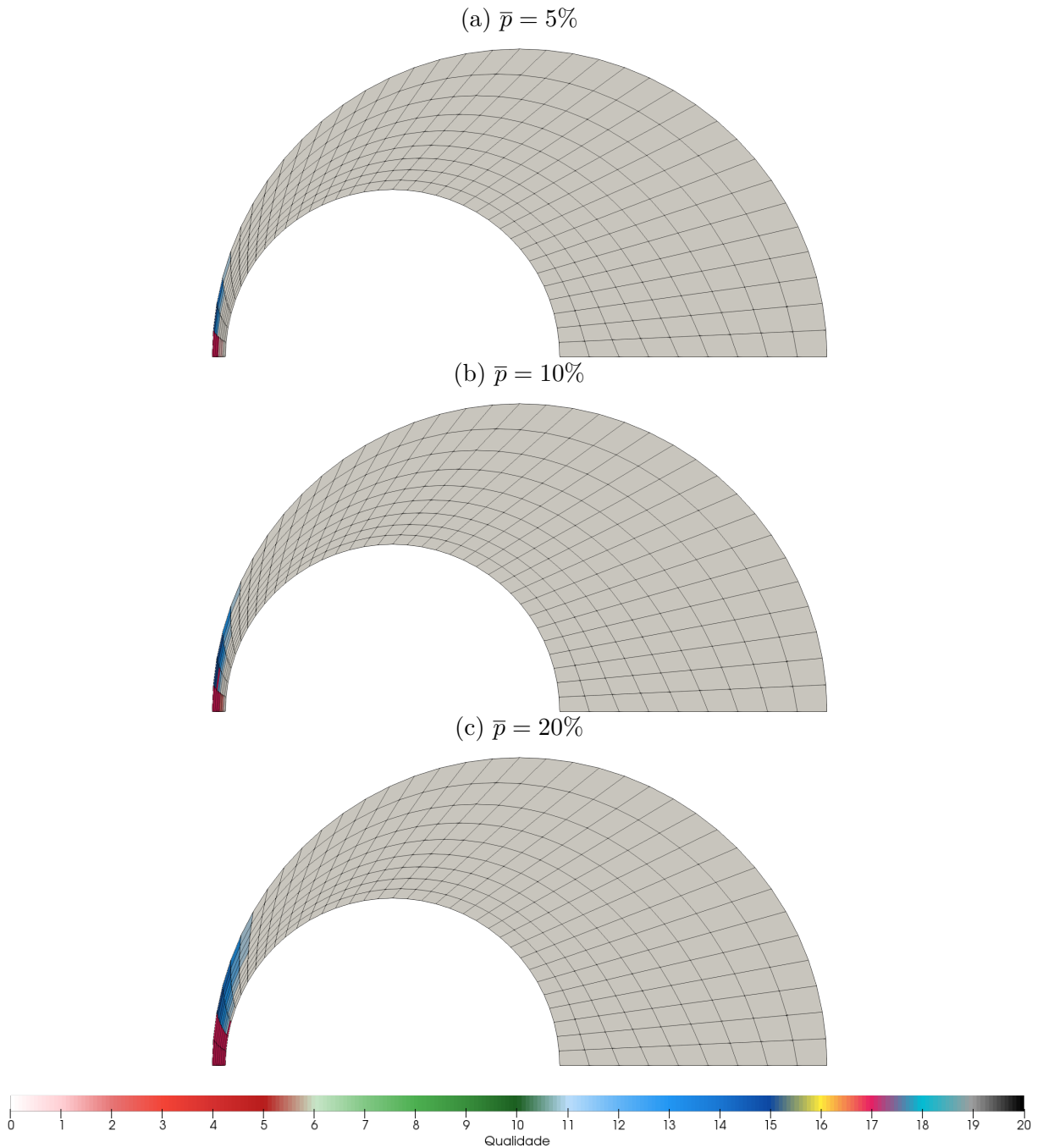


Figura 31 – Elementos de menor qualidade identificados para a forma de gancho

Uma outra maneira para se observar os resultados encontrados da forma de gancho pode ser representada na Tabela 28, na qual se apresentam as porcentagens desejadas \bar{p} e as tolerâncias desejadas $\bar{\epsilon}$ em relação às quantidades de elementos dos conjuntos $U' - \overline{E_U}$, $\overline{E_r}$, $\overline{E_\alpha}$, $\overline{E_c}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$, $U - U'$ e $\overline{E_\Omega}$.

Tabela 28 – Quantidades de elementos dos resultados obtidos para a forma de gancho

\bar{p}	$\bar{\epsilon}$	$U' - \overline{E_U}$	$\overline{E_r}$	$\overline{E_\alpha}$	$\overline{E_c}$	$\overline{E_{r,\alpha}}$	$\overline{E_{r,c}}$	$\overline{E_{\alpha,c}}$	$ U - U' $	$\overline{E_\Gamma}$
5%	0.0987	310	7	0	11	0	4	0	0	0
10%	0.0736	301	9	0	20	0	6	0	0	0
20%	0.0485	279	19	0	45	0	19	0	0	0

De acordo com a Figura 31 e com a Tabela 28, é possível observar que apenas elementos de menor qualidade em relação à razão entre arestas e ao coeficiente de compacidade foram identificados para a forma de gancho, considerando as porcentagens utilizadas \bar{p} . Além disso, para $\bar{p} = 20\%$, destaca-se o número de elementos pertencentes ao conjunto formado pela intersecção entre $\overline{E_r}$ e $\overline{E_c}$. Como esperado, os elementos detectados localizam-se na porção da malha computacional com maior densidade de elementos.

4.7 Geometria Cúspide

Nesta seção, modela-se e analisa-se uma geometria cúspide, isto é, uma malha construída por quatro circunferências, em que cada circunferência tangencia duas outras circunferências. A geometria consiste na região interior formada pelas circunferências. Nesse experimento, os pontos de fronteira são adquiridos de maneira automatizada, utilizando conceitos de geometria. Esses pontos de fronteira são ilustrados na Figura 32 e são divididos em 18 partições na direção x e 18 partições na direção y , totalizando 72 nós.

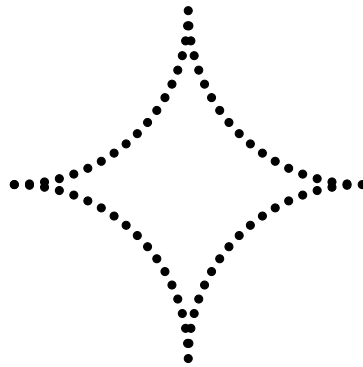


Figura 32 – Pontos de fronteira da geometria cúspide

Por meio do número de partições do plano físico e dos pontos de contorno, executa-se o algoritmo para geração da malha computacional em coordenadas generalizadas. A malha computacional da geometria cúspide é ilustrada na Figura 33 e é constituída por 18 partições na direção ξ e 18 partições na direção η , totalizando 361 nós e 324 elementos.

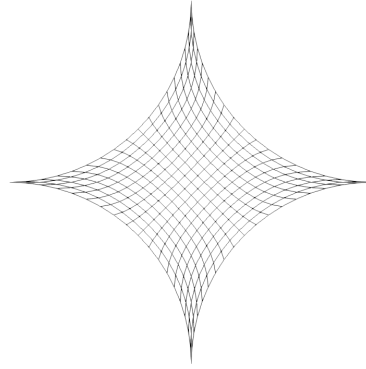


Figura 33 – Malha computacional da geometria cúspide em coordenadas generalizadas

Utilizando a geometria cúspide como parâmetro de entrada, executa-se o primeiro módulo para análise de qualidade, através do Algoritmo 6. Após o cálculo das métricas de qualidade para cada elemento da malha computacional, obtém-se o conjunto total de elementos, associados à razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, ao ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e ao coeficiente de compacidade c .

Tendo posse do conjunto total de elementos e suas métricas de qualidade, assim como informações sobre o número de partições do plano transformado e as porcentagens desejadas \bar{p} , executa-se o segundo módulo para análise de qualidade, por meio do Algoritmo 7. Após a leitura dos dados de entrada, obtém-se que $|U'| = |U| = 324$ elementos, pois não há elementos em que $r = +\infty$ e $\alpha = \nexists$ no conjunto U . Em seguida, a variância de cada métrica de qualidade é calculada, resultando em $s_R^2 = 0.0841$, $s_A^2 = 0.1759$ e $s_C^2 = 523.8016$ para a geometria cúspide.

Uma vez que $s_R^2 \neq 0$, $s_A^2 \neq 0$ e $s_C^2 \neq 0$, calculam-se os valores mínimo m_{\min} e máximo m_{\max} das métricas de qualidade, cujos resultados estão presentes na Tabela 29.

Tabela 29 – Valores mínimo e máximo das métricas de qualidade da geometria cúspide

	r	α	c
min	1.0000	0.0873	16.0000
max	2.2017	1.5708	183.1257

Utilizando os valores encontrados na Tabela 29, normalizam-se os valores das métricas de qualidade no intervalo $[0, 10]$. Ainda, o valor das métricas avaliadas sobre um quadrado também é normalizado no intervalo $[0, 10]$, obtendo $r_{\text{norm}\square} = 0.0000$, $\alpha_{\text{norm}\square} = 10.0000$ e $c_{\text{norm}\square} = 0.0000$. Na sequência, calculam-se as variâncias $s_{M_{\text{norm}}}^2$ das métricas de qualidade normalizadas, resultando em $s_{R_{\text{norm}}}^2 = 5.8231$, $s_{A_{\text{norm}}}^2 = 7.9905$ e $s_{C_{\text{norm}}}^2 = 1.8753$.

Com as variâncias $s_{M_{\text{norm}}}^2$ calculadas, parte-se para o cálculo das tolerâncias mínima e máxima, obtendo $\epsilon_{\min} = 0.0002$ e $\epsilon_{\max} = 0.1468$. Iterando entre o intervalo de valores,

constrói-se o diagrama de dispersão de ϵ_i por $|E_{U_i}|$, o qual é ilustrado na Figura 34 pelos pontos em preto. A curva em verde representa a função de regressão não linear exponencial, encontrada na etapa posterior. No caso da geometria cuspide, o método de mínimos quadrados obteve, como resultado, os coeficientes $c_1 = 300.8651$ e $c_2 = -26.9814$.

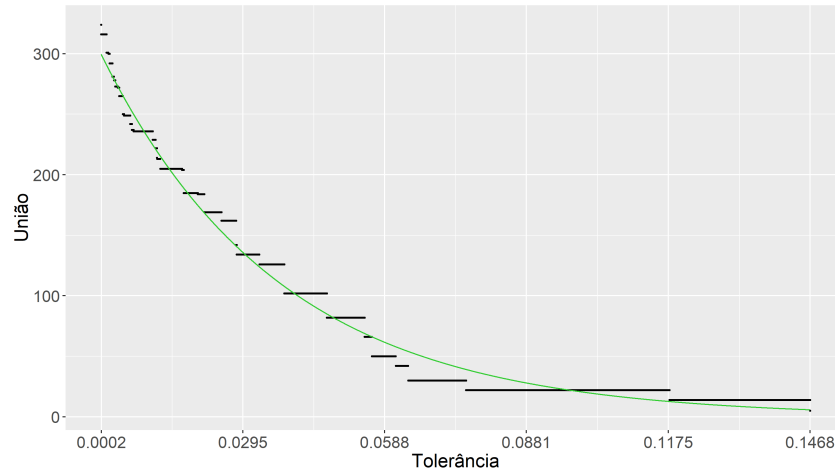


Figura 34 – Diagrama de dispersão de ϵ_i por $|E_{U_i}|$ da geometria cuspide

Dados os coeficientes c_1 e c_2 da função de regressão não linear exponencial, calculam-se as tolerâncias desejadas $\bar{\epsilon}$, considerando $\bar{p} = 5, 10$ e 20% . As porcentagens utilizadas resultaram, respectivamente, em $\bar{\epsilon} = 0.1083, 0.0826$ e 0.0569 . Para cada valor de tolerância desejada $\bar{\epsilon}$, identificam-se os elementos de menor qualidade em relação à razão entre arestas $\overline{E_r}$, ao ângulo interno $\overline{E_\alpha}$ e ao coeficiente de compacidade $\overline{E_c}$. Além disso, por meio de operações envolvendo os conjuntos $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$, obtêm-se os conjuntos $\overline{E_U}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$ e $\overline{E_\cap}$. Finalmente, um número q é atribuído para cada categoria de elementos a partir dos conjuntos obtidos.

Após a escrita dos resultados em um arquivo *.vtk*, a Figura 35 ilustra os elementos de menor qualidade identificados para a geometria cuspide, levando em consideração $\bar{p} = 5, 10$ e 20% , respectivamente, nas Figuras 35a - 35c.

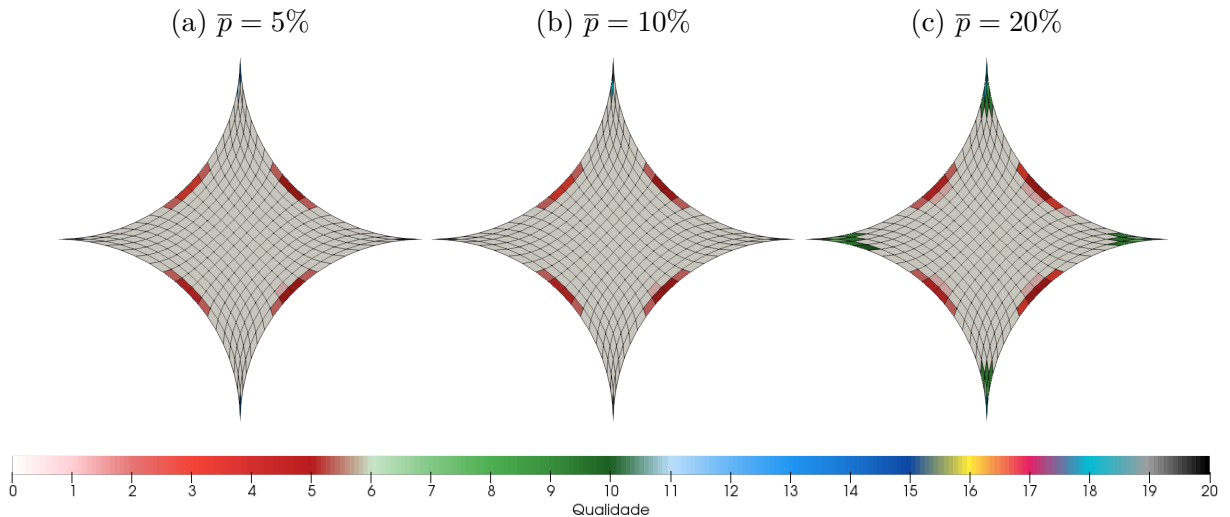


Figura 35 – Elementos de menor qualidade identificados para a geometria cúspide

Uma outra maneira para se observar os resultados encontrados da geometria cúspide pode ser representada na Tabela 30, na qual se apresentam as porcentagens desejadas \bar{p} e as tolerâncias desejadas $\bar{\epsilon}$ em relação às quantidades de elementos dos conjuntos $U' - \overline{E_U}$, $\overline{E_r}$, $\overline{E_\alpha}$, $\overline{E_c}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$, $U - U'$ e $\overline{E_\Gamma}$.

Tabela 30 – Quantidades de elementos dos resultados obtidos para a geometria cúspide

\bar{p}	$\bar{\epsilon}$	$U' - \overline{E_U}$	$\overline{E_r}$	$\overline{E_\alpha}$	$\overline{E_c}$	$\overline{E_{r,\alpha}}$	$\overline{E_{r,c}}$	$\overline{E_{\alpha,c}}$	$ U - U' $	$\overline{E_\Gamma}$
5%	0.1083	302	17	0	5	0	0	0	0	0
10%	0.0826	302	17	5	5	0	0	5	0	0
20%	0.0569	274	25	25	5	0	0	5	0	0

De acordo com a Figura 35 e com a Tabela 30, é possível observar que apenas elementos de menor qualidade em relação à razão entre arestas e ao coeficiente de compacidade foram identificados para a geometria cúspide, considerando $\bar{p} = 5\%$. Para as porcentagens utilizadas $\bar{p} = 10$ e 20% , elementos de menor qualidade em relação a todas as métricas de avaliação foram identificados. Como esperado, os elementos detectados quanto ao ângulo interno e ao coeficiente de compacidade localizam-se nas proximidades dos quatro vértices da malha computacional, e os elementos detectados quanto à razão entre arestas localizam-se na região central das curvas delimitadas pelos vértices da geometria.

4.8 Placas Paralelas com Círculo Imerso

Nesta seção, modelam-se e analisam-se placas paralelas com círculo imerso, ou seja, uma malha retangular com um círculo horizontalmente e verticalmente centralizado atuando como obstáculo. Nesse experimento, os pontos de fronteira são adquiridos de

maneira automatizada, utilizando conceitos de geometria. Esses pontos de fronteira são ilustrados na Figura 36 e são divididos em dois blocos, cujo número de partições nas direções x e y é mostrado na Tabela 31.

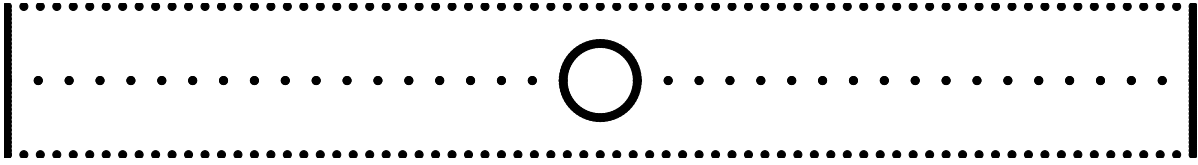


Figura 36 – Pontos de fronteira das placas paralelas com círculo imerso

Tabela 31 – Número de partições nas direções x e y para cada bloco das placas paralelas com círculo imerso

Bloco	x	y	Quantidade de nós
1	72	18	180
2	72	18	180
Total			360

Por meio do número de partições do plano físico e dos pontos de contorno, executa-se o algoritmo para geração da malha computacional em coordenadas generalizadas envolvendo multi-blocos. A malha computacional das placas paralelas com círculo imerso é ilustrada na Figura 37, apresentando também a distribuição dos blocos.

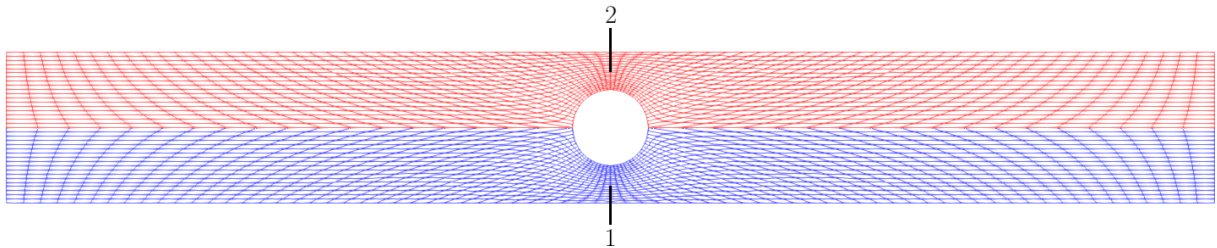


Figura 37 – Malha computacional das placas paralelas com círculo imerso em coordenadas generalizadas

Além disso, a Tabela 32 relaciona cada bloco das placas paralelas com círculo imerso com o número de partições nas direções ξ e η .

Tabela 32 – Número de partições nas direções ξ e η para cada bloco das placas paralelas com círculo imerso

Bloco	ξ	η	Quantidade de nós	Quantidade de elementos
1	72	18	1387	1296
2	72	18	1387	1296
Total			2774	2592

Utilizando as placas paralelas com círculo imerso como parâmetros de entrada, executa-se o primeiro módulo para análise de qualidade, através do Algoritmo 6. Após o cálculo das métricas de qualidade para cada elemento de cada bloco da malha computacional, obtém-se o conjunto total de elementos, associados à razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, ao ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e ao coeficiente de compacidade c .

Tendo posse do conjunto total de elementos e suas métricas de qualidade, assim como informações sobre o número de partições do plano transformado para cada bloco e as porcentagens desejadas \bar{p} , executa-se o segundo módulo para análise de qualidade, por meio do Algoritmo 7. Após a leitura dos dados de entrada, obtém-se que $|U'| = |U| = 2592$ elementos, pois não há elementos em que $r = +\infty$ e $\alpha = \nexists$ no conjunto U . Em seguida, a variância de cada métrica de qualidade é calculada, resultando em $s_R^2 = 1.6841$, $s_A^2 = 0.9103$ e $s_C^2 = 225.0062$ para as placas paralelas com círculo imerso.

Uma vez que $s_R^2 \neq 0$, $s_A^2 \neq 0$ e $s_C^2 \neq 0$, calculam-se os valores mínimo m_{\min} e máximo m_{\max} das métricas de qualidade, cujos resultados estão presentes na Tabela 33.

Tabela 33 – Valores mínimo e máximo das métricas de qualidade das placas paralelas com círculo imerso

	r	α	c
min	1.0000	0.1493	16.0000
max	9.7869	2.9512	99.8478

Utilizando os valores encontrados na Tabela 33, normalizam-se os valores das métricas de qualidade no intervalo $[0, 10]$. Ainda, o valor das métricas avaliadas sobre um quadrado também é normalizado no intervalo $[0, 10]$, obtendo $r_{\text{norm}\square} = 0.0000$, $\alpha_{\text{norm}\square} = 5.0734$ e $c_{\text{norm}\square} = 0.0000$. Na sequência, calculam-se as variâncias $s_{M_{\text{norm}}}^2$ das métricas de qualidade normalizadas, resultando em $s_{R_{\text{norm}}}^2 = 2.1812$, $s_{A_{\text{norm}}}^2 = 11.5958$ e $s_{C_{\text{norm}}}^2 = 3.2004$.

Com as variâncias $s_{M_{\text{norm}}}^2$ calculadas, parte-se para o cálculo das tolerâncias mínima e máxima, obtendo $\epsilon_{\min} = 0.0001$ e $\epsilon_{\max} = 0.0253$. Iterando entre o intervalo de valores, constrói-se o diagrama de dispersão de ϵ_i por $|E_{\cup_i}|$, o qual é ilustrado na Figura 38 pelos pontos em preto. A curva em verde representa a função de regressão não linear exponencial, encontrada na etapa posterior. No caso das placas paralelas com círculo imerso, o método de mínimos quadrados obteve, como resultado, os coeficientes $c_1 = 1716.0594$ e $c_2 = -391.0281$.

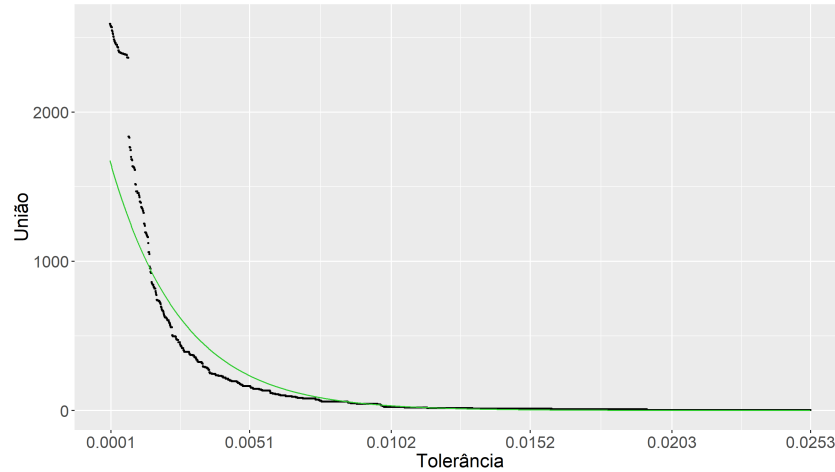


Figura 38 – Diagrama de dispersão de ϵ_i por $|E_{U_i}|$ das placas paralelas com círculo imerso

Dados os coeficientes c_1 e c_2 da função de regressão não linear exponencial, calculam-se as tolerâncias desejadas $\bar{\epsilon}$, considerando $\bar{p} = 5, 10$ e 20% . As porcentagens utilizadas resultaram, respectivamente, em $\bar{\epsilon} = 0.0066, 0.0048$ e 0.0031 . Para cada valor de tolerância desejada $\bar{\epsilon}$, identificam-se os elementos de menor qualidade em relação à razão entre arestas $\overline{E_r}$, ao ângulo interno $\overline{E_\alpha}$ e ao coeficiente de compacidade $\overline{E_c}$. Além disso, por meio de operações envolvendo os conjuntos $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$, obtêm-se os conjuntos $\overline{E_U}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$ e $\overline{E_\cap}$. Finalmente, um número q é atribuído para cada categoria de elementos a partir dos conjuntos obtidos.

Após a escrita dos resultados em um arquivo *.vtk* para cada bloco, a Figura 39 ilustra os elementos de menor qualidade identificados para as placas paralelas com círculo imerso, levando em consideração $\bar{p} = 5, 10$ e 20% , respectivamente, nas Figuras 39a - 39c.

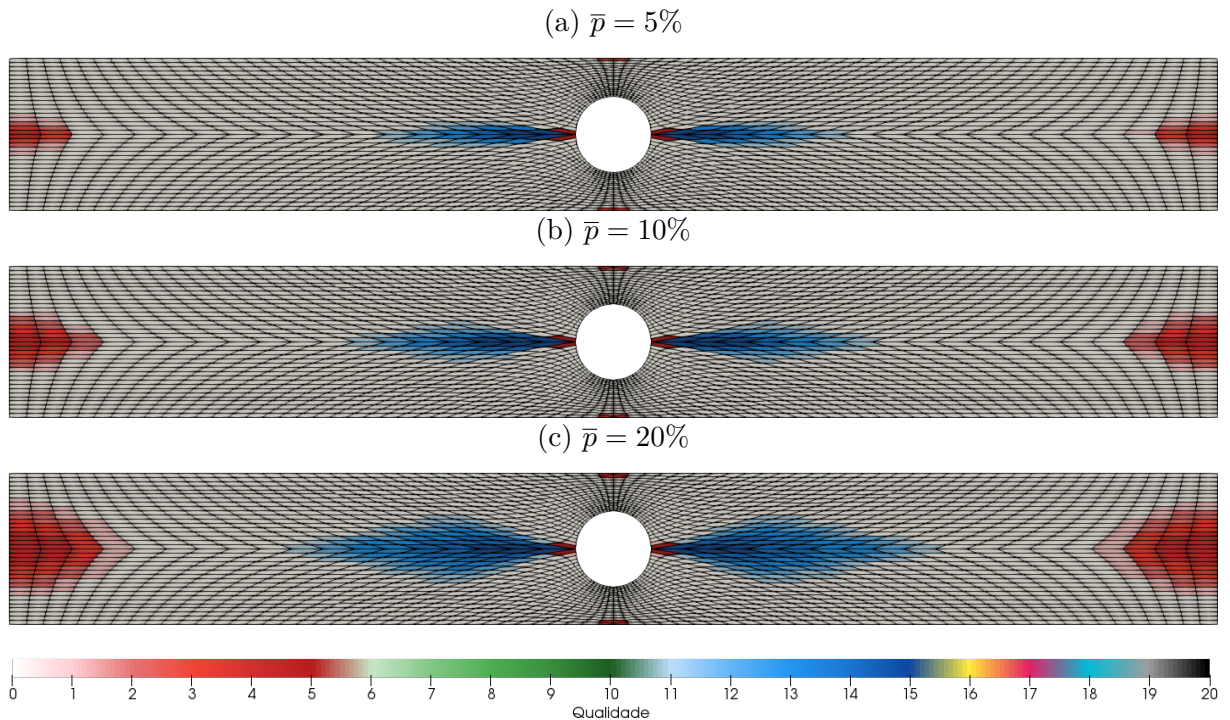


Figura 39 – Elementos de menor qualidade identificados para as placas paralelas com círculo imerso

Uma outra maneira para se observar os resultados encontrados das placas paralelas com círculo imerso pode ser representada na Tabela 34, na qual se apresentam as porcentagens desejadas \bar{p} e as tolerâncias desejadas $\bar{\epsilon}$ em relação às quantidades de elementos dos conjuntos $U' - \overline{E_U}$, $\overline{E_r}$, $\overline{E_\alpha}$, $\overline{E_c}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$, $U - U'$ e $\overline{E_\Gamma}$.

Tabela 34 – Quantidades de elementos dos resultados obtidos para as placas paralelas com círculo imerso

\bar{p}	$\bar{\epsilon}$	$U' - \overline{E_U}$	$\overline{E_r}$	$\overline{E_\alpha}$	$\overline{E_c}$	$\overline{E_{r,\alpha}}$	$\overline{E_{r,c}}$	$\overline{E_{\alpha,c}}$	$ U - U' $	$\overline{E_\Gamma}$
5%	0.0066	2494	45	0	53	0	0	0	0	0
10%	0.0048	2422	85	0	85	0	0	0	0	0
20%	0.0031	2222	177	0	193	0	0	0	0	0

De acordo com a Figura 39 e com a Tabela 34, é possível observar que apenas elementos de menor qualidade em relação à razão entre arestas e ao coeficiente de compacidade foram identificados para as placas paralelas com círculo imerso, considerando as porcentagens utilizadas \bar{p} . Além disso, destaca-se que não existem elementos pertencentes aos conjuntos formados pelas intersecções entre $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$. Os elementos detectados quanto ao coeficiente de compacidade localizam-se nas proximidades do círculo imerso, como esperado, e os elementos detectados quanto à razão entre arestas localizam-se, em grande parte, na região central das quatro fronteiras da malha retangular, as quais sofreram influência do círculo no processo de geração da malha computacional.

4.9 Malha Hexagonal

Nesta seção, modela-se e analisa-se uma malha hexagonal, ou seja, a região limitada por dois hexágonos regulares. Esses hexágonos possuem tamanho de arestas distinto e centros coincidentes, sendo que o hexágono menor está contido no hexágono maior. Ainda, o hexágono menor é rotacionado em $\frac{\pi}{6}$ rad em relação ao outro hexágono. Nesse experimento, os pontos de fronteira são adquiridos de maneira automatizada, utilizando conceitos de geometria. Esses pontos de fronteira são ilustrados na Figura 40 e são divididos em dois blocos, cujo número de partições nas direções x e y é mostrado na Tabela 35.

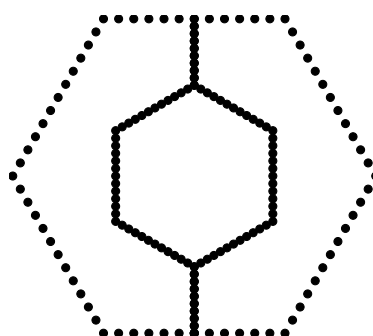


Figura 40 – Pontos de fronteira da malha hexagonal

Tabela 35 – Número de partições nas direções x e y para cada bloco da malha hexagonal

Bloco	x	y	Quantidade de nós
1	36	9	90
2	36	9	90
Total			180

Por meio do número de partições do plano físico e dos pontos de contorno, executa-se o algoritmo para geração da malha computacional em coordenadas generalizadas envolvendo multi-blocos. A malha computacional do hexágono é ilustrada na Figura 41, apresentando também a distribuição dos blocos.

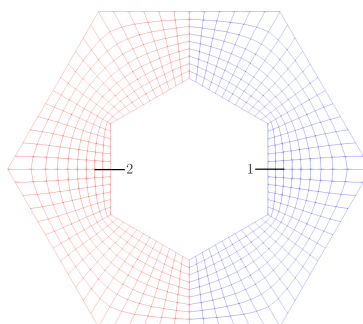


Figura 41 – Malha computacional do hexágono em coordenadas generalizadas

Além disso, a Tabela 36 relaciona cada bloco da malha hexagonal com o número de partições nas direções ξ e η .

Tabela 36 – Número de partições nas direções ξ e η para cada bloco da malha hexagonal

Bloco	ξ	η	Quantidade de nós	Quantidade de elementos
1	36	9	370	324
2	36	9	370	324
Total			740	648

Utilizando a malha hexagonal como parâmetro de entrada, executa-se o primeiro módulo para análise de qualidade, através do Algoritmo 6. Após o cálculo das métricas de qualidade para cada elemento de cada bloco da malha computacional, obtém-se o conjunto total de elementos, associados à razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, ao ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e ao coeficiente de compacidade c .

Tendo posse do conjunto total de elementos e suas métricas de qualidade, assim como informações sobre o número de partições do plano transformado para cada bloco e as porcentagens desejadas \bar{p} , executa-se o segundo módulo para análise de qualidade, por meio do Algoritmo 7. Após a leitura dos dados de entrada, obtém-se que $|U'| = |U| = 648$ elementos, pois não há elementos em que $r = +\infty$ e $\alpha = \nexists$ no conjunto U . Em seguida, a variância de cada métrica de qualidade é calculada, resultando em $s_R^2 = 0.0431$, $s_A^2 = 0.0486$ e $s_C^2 = 0.2013$ para a malha hexagonal.

Uma vez que $s_R^2 \neq 0$, $s_A^2 \neq 0$ e $s_C^2 \neq 0$, calculam-se os valores mínimo m_{\min} e máximo m_{\max} das métricas de qualidade, cujos resultados estão presentes na Tabela 37.

Tabela 37 – Valores mínimo e máximo das métricas de qualidade da malha hexagonal

	r	α	c
min	1.0000	1.0470	16.0000
max	2.0490	2.0944	17.9618

Utilizando os valores encontrados na Tabela 37, normalizam-se os valores das métricas de qualidade no intervalo $[0, 10]$. Ainda, o valor das métricas avaliadas sobre um quadrado também é normalizado no intervalo $[0, 10]$, obtendo $r_{\text{norm}\square} = 0.0000$, $\alpha_{\text{norm}\square} = 5.0009$ e $c_{\text{norm}\square} = 0.0000$. Na sequência, calculam-se as variâncias $s_{M_{\text{norm}}}^2$ das métricas de qualidade normalizadas, resultando em $s_{R_{\text{norm}}}^2 = 3.9201$, $s_{A_{\text{norm}}}^2 = 4.4293$ e $s_{C_{\text{norm}}}^2 = 5.2291$.

Com as variâncias $s_{M_{\text{norm}}}^2$ calculadas, parte-se para o cálculo das tolerâncias mínima e máxima, obtendo $\epsilon_{\min} = 0.0000$ e $\epsilon_{\max} = 0.1795$. Iterando entre o intervalo de valores, constrói-se o diagrama de dispersão de ϵ_i por $|E_{U_i}|$, o qual é ilustrado na Figura 42

pelos pontos em preto. A curva em verde representa a função de regressão não linear exponencial, encontrada na etapa posterior. No caso da malha hexagonal, o método de mínimos quadrados obteve, como resultado, os coeficientes $c_1 = 385.6094$ e $c_2 = -24.4200$.

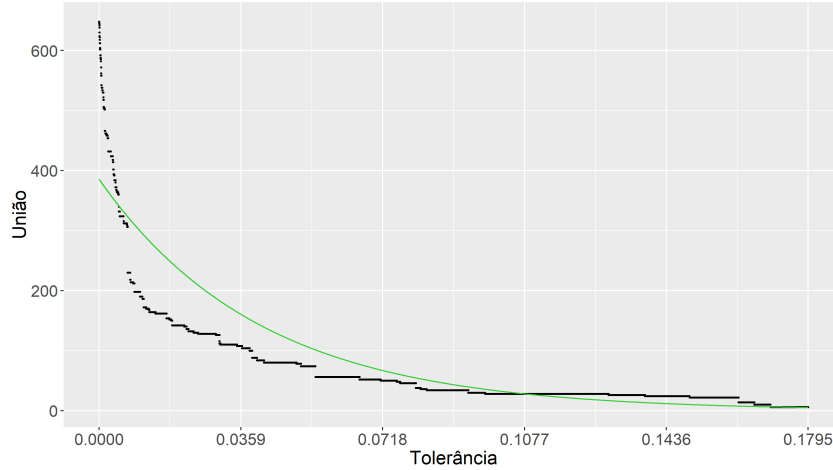


Figura 42 – Diagrama de dispersão de ϵ_i por $|E_{U_i}|$ da malha hexagonal

Dados os coeficientes c_1 e c_2 da função de regressão não linear exponencial, calculam-se as tolerâncias desejadas $\bar{\epsilon}$, considerando $\bar{p} = 5, 10$ e 20% . As porcentagens utilizadas resultaram, respectivamente, em $\bar{\epsilon} = 0.1014, 0.0730$ e 0.0447 . Para cada valor de tolerância desejada $\bar{\epsilon}$, identificam-se os elementos de menor qualidade em relação à razão entre arestas $\overline{E_r}$, ao ângulo interno $\overline{E_\alpha}$ e ao coeficiente de compacidade $\overline{E_c}$. Além disso, por meio de operações envolvendo os conjuntos $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$, obtêm-se os conjuntos $\overline{E_U}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$ e $\overline{E_\cap}$. Finalmente, um número q é atribuído para cada categoria de elementos a partir dos conjuntos obtidos.

Após a escrita dos resultados em um arquivo *.vtk* para cada bloco, a Figura 43 ilustra os elementos de menor qualidade identificados para a malha hexagonal, levando em consideração $\bar{p} = 5, 10$ e 20% , respectivamente, nas Figuras 43a - 43c.

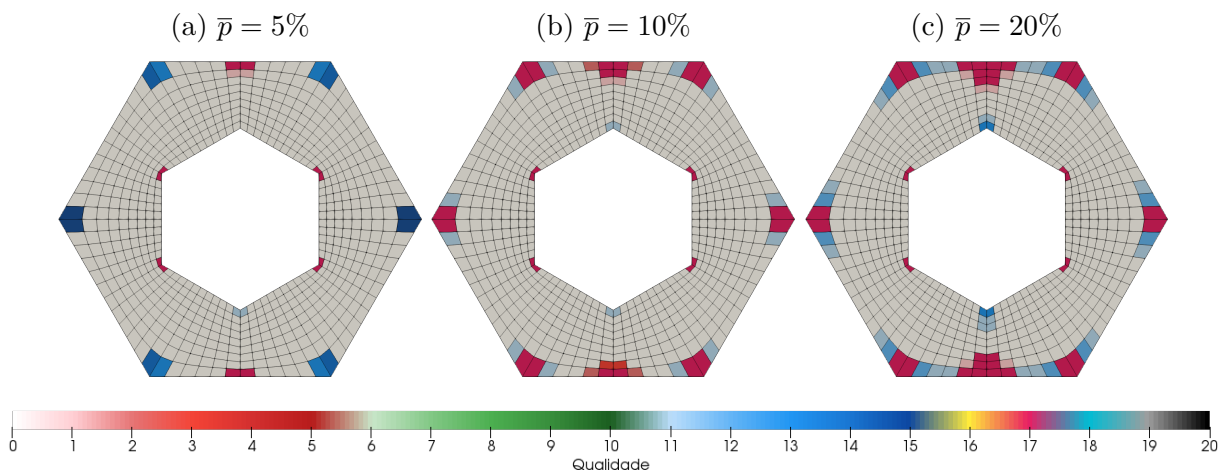


Figura 43 – Elementos de menor qualidade identificados para a malha hexagonal

Uma outra maneira para se observar os resultados encontrados da malha hexagonal pode ser representada na Tabela 38, na qual se apresentam as porcentagens desejadas \bar{p} e as tolerâncias desejadas $\bar{\epsilon}$ em relação às quantidades de elementos dos conjuntos $U' - \overline{E_U}$, $\overline{E_r}$, $\overline{E_\alpha}$, $\overline{E_c}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$, $U - U'$ e $\overline{E_\cap}$.

Tabela 38 – Quantidades de elementos dos resultados obtidos para a malha hexagonal

\bar{p}	$\bar{\epsilon}$	$U' - \overline{E_U}$	$\overline{E_r}$	$\overline{E_\alpha}$	$\overline{E_c}$	$\overline{E_{r,\alpha}}$	$\overline{E_{r,c}}$	$\overline{E_{\alpha,c}}$	$ U - U' $	$\overline{E_\cap}$
5%	0.1014	620	14	0	26	0	12	0	0	0
10%	0.0730	598	34	0	42	0	26	0	0	0
20%	0.0447	568	42	0	74	0	36	0	0	0

De acordo com a Figura 43 e com a Tabela 38, é possível observar que apenas elementos de menor qualidade em relação à razão entre arestas e ao coeficiente de compacidade foram identificados para a malha hexagonal, considerando as porcentagens utilizadas \bar{p} . Além disso, destaca-se o número de elementos pertencentes ao conjunto formado pela intersecção entre $\overline{E_r}$ e $\overline{E_c}$. Os elementos detectados localizam-se não apenas nas proximidades dos vértices de ambos os hexágonos regulares, como esperado, mas também nas partes superior e inferior do hexágono maior.

4.10 NACA 64A 010

Nesta seção, modela-se e analisa-se a malha da parte interna de um NACA 64A 010. Os aerofólios NACA constituem-se em formas para asas de aeronaves [60]. Essas formas são descritas pela série de dígitos que acompanha a palavra “NACA” [60]. Nesse experimento, a aquisição dos pontos de fronteira é baseada em um trabalho científico [30]. Esses pontos de fronteira são ilustrados na Figura 44 e são divididos em dois blocos, cujo número de partições nas direções x e y é mostrado na Tabela 39.

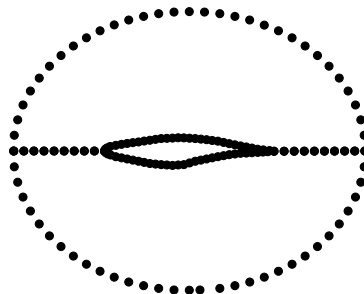


Figura 44 – Pontos de fronteira do NACA. (Fonte: 30)

Tabela 39 – Número de partições nas direções x e y para cada bloco do NACA

Bloco	x	y	Quantidade de nós
1	32	9	82
2	32	9	82
Total			164

Por meio do número de partições do plano físico e dos pontos de contorno, executa-se o algoritmo para geração da malha computacional em coordenadas generalizadas envolvendo multi-blocos. A malha computacional do NACA é ilustrada na Figura 45, apresentando também a distribuição dos blocos.

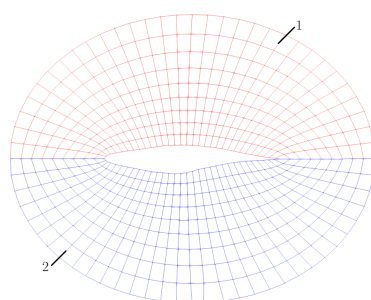


Figura 45 – Malha computacional do NACA em coordenadas generalizadas. (Fonte: 30)

Além disso, a Tabela 40 relaciona cada bloco do NACA com o número de partições nas direções ξ e η .

Tabela 40 – Número de partições nas direções ξ e η para cada bloco do NACA

Bloco	ξ	η	Quantidade de nós	Quantidade de elementos
1	32	9	330	288
2	32	9	330	288
Total			660	576

Utilizando o NACA como parâmetro de entrada, executa-se o primeiro módulo para análise de qualidade, através do Algoritmo 6. Após o cálculo das métricas de qualidade para cada elemento de cada bloco da malha computacional, obtém-se o conjunto total de elementos, associados à razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, ao ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e ao coeficiente de compacidade c .

Tendo posse do conjunto total de elementos e suas métricas de qualidade, assim como informações sobre o número de partições do plano transformado para cada bloco e as porcentagens desejadas \bar{p} , executa-se o segundo módulo para análise de qualidade, por meio do Algoritmo 7. Após a leitura dos dados de entrada, obtém-se que $|U'| = |U| = 576$ elementos, pois não há elementos em que $r = +\infty$ e $\alpha = \frac{\pi}{2}$ no conjunto U . Em seguida,

a variância de cada métrica de qualidade é calculada, resultando em $s_R^2 = 0.0985$, $s_A^2 = 0.0558$ e $s_C^2 = 2.1345$ para o NACA.

Uma vez que $s_R^2 \neq 0$, $s_A^2 \neq 0$ e $s_C^2 \neq 0$, calculam-se os valores mínimo m_{\min} e máximo m_{\max} das métricas de qualidade, cujos resultados estão presentes na Tabela 41.

Tabela 41 – Valores mínimo e máximo das métricas de qualidade do NACA

	r	α	c
min	1.0000	0.7744	16.0000
max	4.6274	3.0715	33.4479

Utilizando os valores encontrados na Tabela 41, normalizam-se os valores das métricas de qualidade no intervalo $[0, 10]$. Ainda, o valor das métricas avaliadas sobre um quadrado também é normalizado no intervalo $[0, 10]$, obtendo $r_{\text{norm}\square} = 0.0000$, $\alpha_{\text{norm}\square} = 3.4669$ e $c_{\text{norm}\square} = 0.0000$. Na sequência, calculam-se as variâncias $s_{M_{\text{norm}}}^2$ das métricas de qualidade normalizadas, resultando em $s_{R_{\text{norm}}}^2 = 0.7484$, $s_{A_{\text{norm}}}^2 = 1.0569$ e $s_{C_{\text{norm}}}^2 = 0.7012$.

Com as variâncias $s_{M_{\text{norm}}}^2$ calculadas, parte-se para o cálculo das tolerâncias mínima e máxima, obtendo $\epsilon_{\min} = 0.0000$ e $\epsilon_{\max} = 0.1486$. Iterando entre o intervalo de valores, constrói-se o diagrama de dispersão de ϵ_i por $|E_{U_i}|$, o qual é ilustrado na Figura 46 pelos pontos em preto. A curva em verde representa a função de regressão não linear exponencial, encontrada na etapa posterior. No caso do NACA, o método de mínimos quadrados obteve, como resultado, os coeficientes $c_1 = 152.7410$ e $c_2 = -47.1517$.

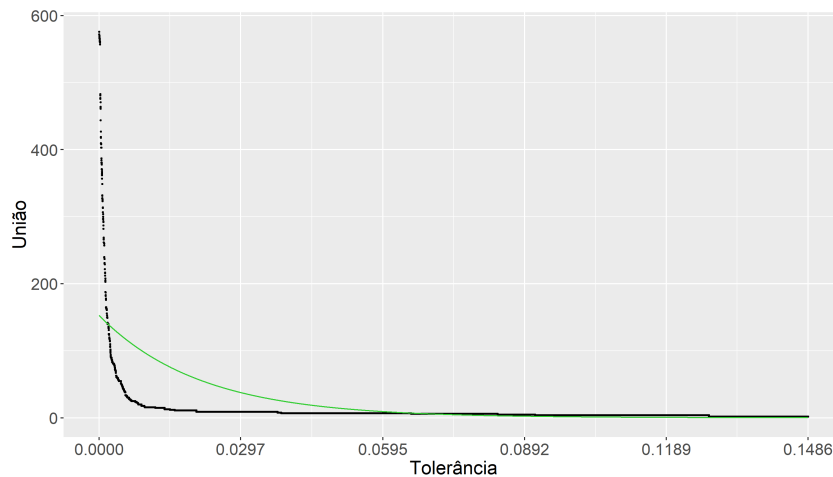


Figura 46 – Diagrama de dispersão de ϵ_i por $|E_{U_i}|$ do NACA

Dados os coeficientes c_1 e c_2 da função de regressão não linear exponencial, calculam-se as tolerâncias desejadas $\bar{\epsilon}$, considerando $\bar{p} = 5, 10$ e 20% . As porcentagens utilizadas resultaram, respectivamente, em $\bar{\epsilon} = 0.0354, 0.0207$ e 0.0060 . Para cada valor de tolerância desejada $\bar{\epsilon}$, identificam-se os elementos de menor qualidade em relação à razão

entre arestas \overline{E}_r , ao ângulo interno \overline{E}_α e ao coeficiente de compacidade \overline{E}_c . Além disso, por meio de operações envolvendo os conjuntos \overline{E}_r , \overline{E}_α e \overline{E}_c , obtêm-se os conjuntos \overline{E}_U , $\overline{E}_{r,\alpha}$, $\overline{E}_{r,c}$, $\overline{E}_{\alpha,c}$ e \overline{E}_\cap . Finalmente, um número q é atribuído para cada categoria de elementos a partir dos conjuntos obtidos.

Após a escrita dos resultados em um arquivo *.vtk* para cada bloco, a Figura 47 ilustra os elementos de menor qualidade identificados para o NACA, levando em consideração $\bar{p} = 5, 10$ e 20% , respectivamente, nas Figuras 47a - 47c.

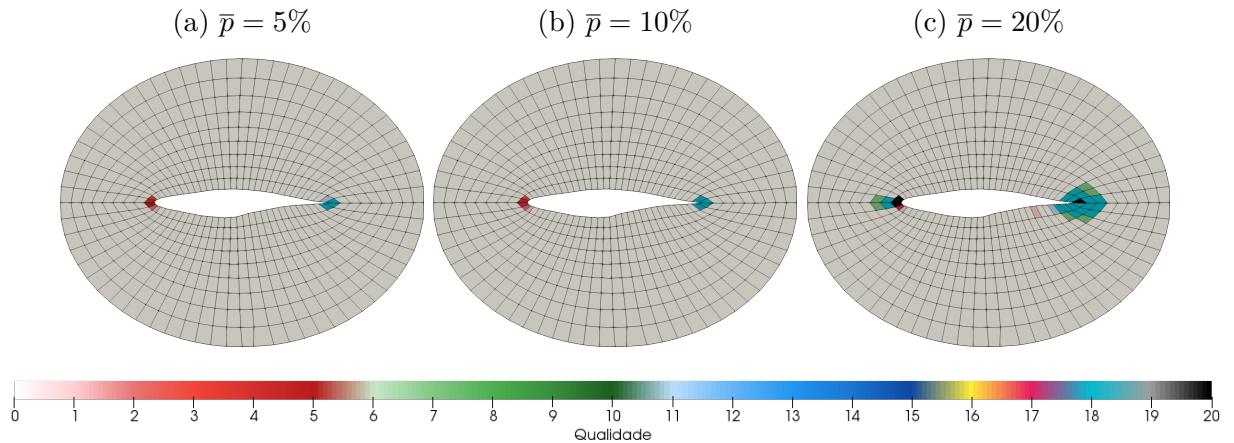


Figura 47 – Elementos de menor qualidade identificados para o NACA

Uma outra maneira para se observar os resultados encontrados do NACA pode ser representada na Tabela 42, na qual se apresentam as porcentagens desejadas \bar{p} e as tolerâncias desejadas $\bar{\epsilon}$ em relação às quantidades de elementos dos conjuntos $U' - \overline{E}_U$, \overline{E}_r , \overline{E}_α , \overline{E}_c , $\overline{E}_{r,\alpha}$, $\overline{E}_{r,c}$, $\overline{E}_{\alpha,c}$, $U - U'$ e \overline{E}_\cap .

Tabela 42 – Quantidades de elementos dos resultados obtidos para o NACA

\bar{p}	$\bar{\epsilon}$	$U' - \overline{E}_U$	\overline{E}_r	\overline{E}_α	\overline{E}_c	$\overline{E}_{r,\alpha}$	$\overline{E}_{r,c}$	$\overline{E}_{\alpha,c}$	$ U - U' $	\overline{E}_\cap
5%	0.0354	567	5	3	4	0	0	3	0	0
10%	0.0207	567	5	4	5	0	1	4	0	0
20%	0.0060	545	7	27	23	4	5	21	0	4

De acordo com a Figura 47 e com a Tabela 42, é possível observar que elementos de menor qualidade em relação a todas as métricas de avaliação foram identificados para o NACA 64A 010, considerando as porcentagens utilizadas \bar{p} . Além disso, para $\bar{p} = 20\%$, destaca-se o número de elementos pertencentes ao conjunto formado pela intersecção entre \overline{E}_α e \overline{E}_c . Como esperado, os elementos detectados localizam-se nas proximidades do aerofólio presente na malha computacional.

4.11 Malha com Obstáculo Multi-blocos

Nesta seção, modela-se e analisa-se a malha com obstáculo, descrita na Seção 4.4, dividida em blocos. Nesse experimento, a aquisição dos pontos de fronteira é baseada em um trabalho científico [30]. Esses pontos de fronteira são ilustrados na Figura 48 e são divididos em três blocos, cujo número de partições nas direções x e y é mostrado na Tabela 43.

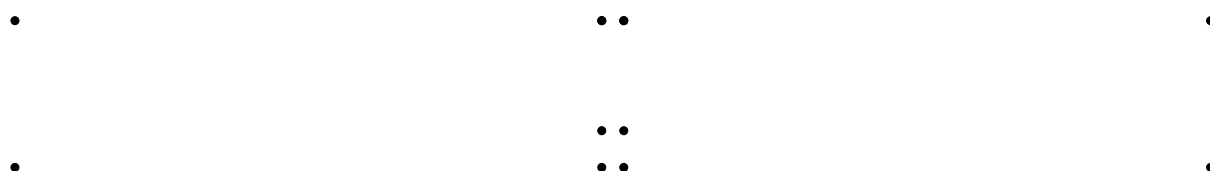


Figura 48 – Pontos de fronteira da malha com obstáculo multi-blocos. (Fonte: 30)

Tabela 43 – Número de partições nas direções x e y para cada bloco da malha com obstáculo multi-blocos

Bloco	x	y	Quantidade de nós
1	1	1	4
2	1	1	4
3	1	1	4
Total			12

Por meio do número de partições do plano físico e dos pontos de contorno, executa-se o algoritmo para geração da malha computacional em coordenadas generalizadas envolvendo multi-blocos. A malha computacional com obstáculo multi-blocos é ilustrada na Figura 49, apresentando também a distribuição dos blocos.

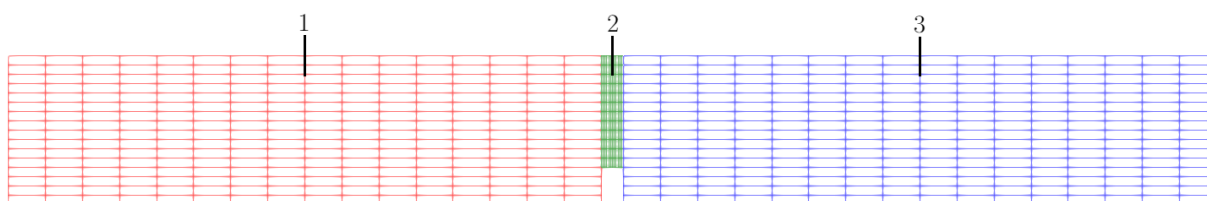


Figura 49 – Malha computacional com obstáculo multi-blocos em coordenadas generalizadas

Além disso, a Tabela 44 relaciona cada bloco da malha com obstáculo multi-blocos com o número de partições nas direções ξ e η .

Tabela 44 – Número de partições nas direções ξ e η para cada bloco da malha com obstáculo multi-blocos

Bloco	ξ	η	Quantidade de nós	Quantidade de elementos
1	16	16	289	256
2	12	12	169	144
3	16	16	289	256
Total			747	656

Utilizando a malha com obstáculo multi-blocos como parâmetro de entrada, executa-se o primeiro módulo para análise de qualidade, através do Algoritmo 6. Após o cálculo das métricas de qualidade para cada elemento de cada bloco da malha computacional, obtém-se o conjunto total de elementos, associados à razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, ao ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e ao coeficiente de compacidade c .

Tendo posse do conjunto total de elementos e suas métricas de qualidade, assim como informações sobre o número de partições do plano transformado para cada bloco e as porcentagens desejadas \bar{p} , executa-se o segundo módulo para análise de qualidade, por meio do Algoritmo 7. Após a leitura dos dados de entrada, obtém-se que $|U'| = |U| = 656$ elementos, pois não há elementos em que $r = +\infty$ e $\alpha = \nexists$ no conjunto U . Em seguida, a variância de cada métrica de qualidade é calculada, resultando em $s_R^2 = 0.1716$, $s_A^2 = 0.0000$ e $s_C^2 = 2.4777$ para a malha com obstáculo multi-blocos.

Uma vez que $s_R^2 \neq 0$ e $s_C^2 \neq 0$, calculam-se os valores mínimo m_{\min} e máximo m_{\max} das métricas de qualidade, cujos resultados estão presentes na Tabela 45.

Tabela 45 – Valores mínimo e máximo das métricas de qualidade da malha com obstáculo multi-blocos

	r	α	c
min	1.0000	1.5708	16.0000
max	5.0000	1.5708	28.8000

Utilizando os valores encontrados na Tabela 45, normalizam-se os valores das métricas de qualidade no intervalo $[0, 10]$. Ainda, o valor das métricas avaliadas sobre um quadrado também é normalizado no intervalo $[0, 10]$, obtendo $r_{\text{norm}\square} = 0.0000$, $\alpha_{\text{norm}\square} = 10.0000$ e $c_{\text{norm}\square} = 0.0000$. Na sequência, calculam-se as variâncias $s_{M_{\text{norm}}}^2$ das métricas de qualidade normalizadas, resultando em $s_{R_{\text{norm}}}^2 = 1.0724$, $s_{A_{\text{norm}}}^2 = 0.0000$ e $s_{C_{\text{norm}}}^2 = 1.5123$.

Com as variâncias $s_{M_{\text{norm}}}^2$ calculadas, parte-se para o cálculo das tolerâncias mínima ϵ_{\min} e máxima ϵ_{\max} . Contudo, visto que a malha com obstáculo multi-blocos possui apenas dois elementos distintos, o conjunto E_{\cup} é constituído pela mesma quantidade de elementos

para ambas as tolerâncias. Assim, iterando entre o intervalo de valores, o diagrama de dispersão de ϵ_i por $|E_{U_i}|$ é ajustado por uma função constante. Por esse motivo, a função de regressão não linear exponencial não é encontrada nesse experimento.

Considerando o diagrama de dispersão da malha com obstáculo multi-blocos, atribui-se à tolerância desejada $\bar{\epsilon}$ a média aritmética das tolerâncias mínima ϵ_{\min} e máxima ϵ_{\max} , independente das porcentagens desejadas \bar{p} . Dessa forma, para todas as porcentagens utilizadas, resultou-se em $\bar{\epsilon} = 0.1000$, para o qual se identificam os elementos de menor qualidade em relação à razão entre arestas $\overline{E_r}$, ao ângulo interno $\overline{E_\alpha}$ e ao coeficiente de compacidade $\overline{E_c}$. Além disso, por meio de operações envolvendo os conjuntos $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$, obtêm-se os conjuntos $\overline{E_U}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$ e $\overline{E_\cap}$. Finalmente, um número q é atribuído para cada categoria de elementos a partir dos conjuntos obtidos.

Após a escrita dos resultados em um arquivo *.vtk* para cada bloco, a Figura 50 ilustra os elementos de menor qualidade identificados para a malha com obstáculo multi-blocos, independente das porcentagens desejadas \bar{p} levadas em consideração.

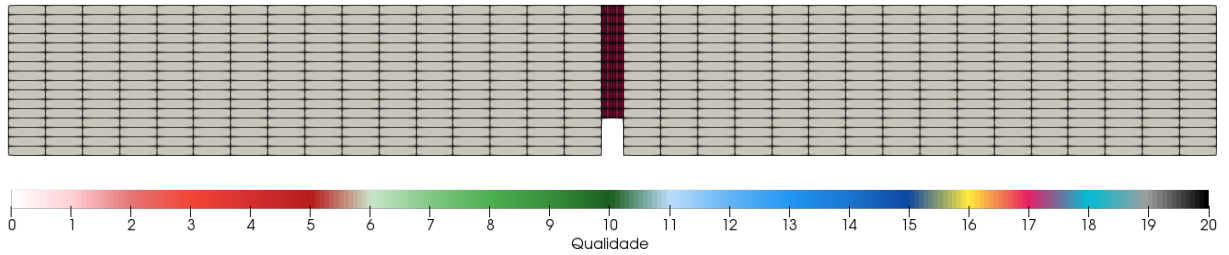


Figura 50 – Elementos de menor qualidade identificados para a malha com obstáculo multi-blocos

Utilizando a tolerância desejada $\bar{\epsilon}$, uma outra maneira para se observar os resultados encontrados da malha com obstáculo multi-blocos pode ser representada na Tabela 46, na qual se apresentam as quantidades de elementos dos conjuntos $U' - \overline{E_U}$, $\overline{E_r}$, $\overline{E_\alpha}$, $\overline{E_c}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$, $U - U'$ e $\overline{E_\cap}$.

Tabela 46 – Quantidades de elementos dos resultados obtidos para a malha com obstáculo multi-blocos

$U' - \overline{E_U}$	$\overline{E_r}$	$\overline{E_\alpha}$	$\overline{E_c}$	$\overline{E_{r,\alpha}}$	$\overline{E_{r,c}}$	$\overline{E_{\alpha,c}}$	$ U - U' $	$\overline{E_\cap}$
512	144	0	144	0	144	0	0	0

De acordo com a Figura 50 e com a Tabela 46, é possível observar que apenas elementos de menor qualidade em relação à razão entre arestas e ao coeficiente de compacidade foram identificados para a malha com obstáculo multi-blocos, considerando as porcentagens utilizadas \bar{p} . Além disso, esses elementos também pertencem ao conjunto formado pela intersecção entre $\overline{E_r}$ e $\overline{E_c}$. Como esperado, os elementos detectados localizam-se no bloco da malha computacional com maior densidade de elementos. Em comparação à

malha com obstáculo descrita na Seção 4.4, pode-se perceber que, apesar da concentração de elementos detectados em um bloco, a técnica multi-blocos permitiu a melhoria dos elementos de menor qualidade em relação ao ângulo interno, visto que todos os elementos da malha computacional são retangulares.

4.12 Placa de Orifício Multi-blocos

Nesta seção, modela-se e analisa-se a placa de orifício, descrita na Seção 4.5, dividida em blocos. Nesse experimento, a aquisição dos pontos de fronteira é baseada em um trabalho científico [30]. Esses pontos de fronteira são ilustrados na Figura 51 e são divididos em três blocos, cujo número de partições nas direções x e y é mostrado na Tabela 47.

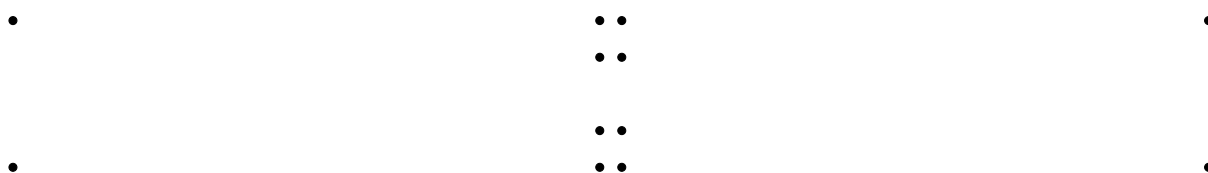


Figura 51 – Pontos de fronteira da placa de orifício multi-blocos. (Fonte: 30)

Tabela 47 – Número de partições nas direções x e y para cada bloco da placa de orifício multi-blocos

Bloco	x	y	Quantidade de nós
1	1	1	4
2	1	1	4
3	1	1	4
Total			12

Por meio do número de partições do plano físico e dos pontos de contorno, executa-se o algoritmo para geração da malha computacional em coordenadas generalizadas envolvendo multi-blocos. A malha computacional da placa de orifício multi-blocos é ilustrada na Figura 52, apresentando também a distribuição dos blocos.

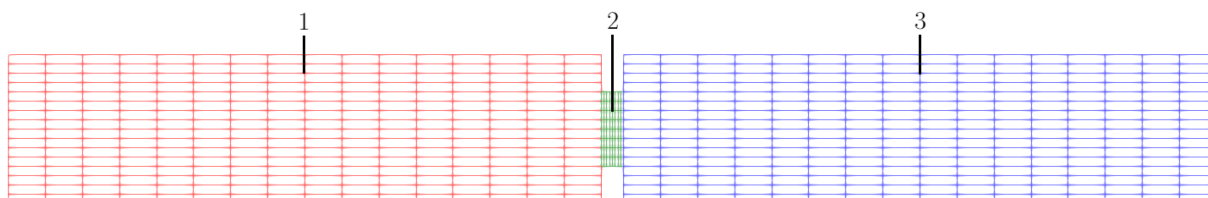


Figura 52 – Malha computacional da placa de orifício multi-blocos em coordenadas generalizadas

Além disso, a Tabela 48 relaciona cada bloco da placa de orifício multi-blocos com o número de partições nas direções ξ e η .

Tabela 48 – Número de partições nas direções ξ e η para cada bloco da placa de orifício multi-blocos

Bloco	ξ	η	Quantidade de nós	Quantidade de elementos
1	16	16	289	256
2	8	8	81	64
3	16	16	289	256
Total			659	576

Utilizando a placa de orifício multi-blocos como parâmetro de entrada, executa-se o primeiro módulo para análise de qualidade, através do Algoritmo 6. Após o cálculo das métricas de qualidade para cada elemento de cada bloco da malha computacional, obtém-se o conjunto total de elementos, associados à razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, ao ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e ao coeficiente de compacidade c .

Tendo posse do conjunto total de elementos e suas métricas de qualidade, assim como informações sobre o número de partições do plano transformado para cada bloco e as porcentagens desejadas \bar{p} , executa-se o segundo módulo para análise de qualidade, por meio do Algoritmo 7. Após a leitura dos dados de entrada, obtém-se que $|U'| = |U| = 576$ elementos, pois não há elementos em que $r = +\infty$ e $\alpha = \nexists$ no conjunto U . Em seguida, a variância de cada métrica de qualidade é calculada, resultando em $s_R^2 = 0.0440$, $s_A^2 = 0.0000$ e $s_C^2 = 0.6020$ para a placa de orifício multi-blocos.

Uma vez que $s_R^2 \neq 0$ e $s_C^2 \neq 0$, calculam-se os valores mínimo m_{\min} e máximo m_{\max} das métricas de qualidade, cujos resultados estão presentes na Tabela 49.

Tabela 49 – Valores mínimo e máximo das métricas de qualidade da placa de orifício multi-blocos

	r	α	c
min	1.0000	1.5708	16.0000
max	4.0000	1.5708	25.0000

Utilizando os valores encontrados na Tabela 49, normalizam-se os valores das métricas de qualidade no intervalo $[0, 10]$. Ainda, o valor das métricas avaliadas sobre um quadrado também é normalizado no intervalo $[0, 10]$, obtendo $r_{\text{norm}\square} = 0.0000$, $\alpha_{\text{norm}\square} = 10.0000$ e $c_{\text{norm}\square} = 0.0000$. Na sequência, calculam-se as variâncias $s_{M_{\text{norm}}}^2$ das métricas de qualidade normalizadas, resultando em $s_{R_{\text{norm}}}^2 = 0.4886$, $s_{A_{\text{norm}}}^2 = 0.0000$ e $s_{C_{\text{norm}}}^2 = 0.7432$.

Com as variâncias $s_{M_{\text{norm}}}^2$ calculadas, parte-se para o cálculo das tolerâncias mínima ϵ_{\min} e máxima ϵ_{\max} . Contudo, visto que a placa de orifício multi-blocos possui apenas dois elementos distintos, o conjunto E_{\cup} é constituído pela mesma quantidade de elementos

para ambas as tolerâncias. Assim, iterando entre o intervalo de valores, o diagrama de dispersão de ϵ_i por $|E_{U_i}|$ é ajustado por uma função constante. Por esse motivo, a função de regressão não linear exponencial não é encontrada nesse experimento.

Considerando o diagrama de dispersão da placa de orifício multi-blocos, atribui-se à tolerância desejada $\bar{\epsilon}$ a média aritmética das tolerâncias mínima ϵ_{\min} e máxima ϵ_{\max} , independente das porcentagens desejadas \bar{p} . Dessa forma, para todas as porcentagens utilizadas, resultou-se em $\bar{\epsilon} = 0.1000$, para o qual se identificam os elementos de menor qualidade em relação à razão entre arestas $\overline{E_r}$, ao ângulo interno $\overline{E_\alpha}$ e ao coeficiente de compacidade $\overline{E_c}$. Além disso, por meio de operações envolvendo os conjuntos $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$, obtêm-se os conjuntos $\overline{E_U}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$ e $\overline{E_\cap}$. Finalmente, um número q é atribuído para cada categoria de elementos a partir dos conjuntos obtidos.

Após a escrita dos resultados em um arquivo *.vtk* para cada bloco, a Figura 53 ilustra os elementos de menor qualidade identificados para a placa de orifício multi-blocos, independente das porcentagens desejadas \bar{p} levadas em consideração.

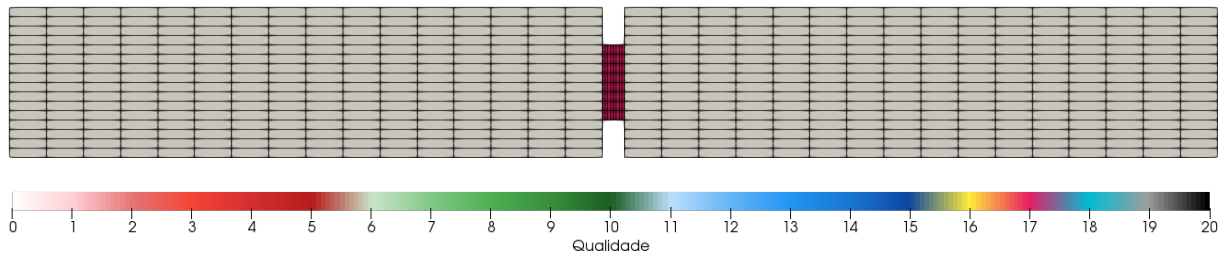


Figura 53 – Elementos de menor qualidade identificados para a placa de orifício multi-blocos

Utilizando a tolerância desejada $\bar{\epsilon}$, uma outra maneira para se observar os resultados encontrados da placa de orifício multi-blocos pode ser representada na Tabela 50, na qual se apresentam as quantidades de elementos dos conjuntos $U' - \overline{E_U}$, $\overline{E_r}$, $\overline{E_\alpha}$, $\overline{E_c}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$, $U - U'$ e $\overline{E_\cap}$.

Tabela 50 – Quantidades de elementos dos resultados obtidos para a placa de orifício multi-blocos

$U' - \overline{E_U}$	$\overline{E_r}$	$\overline{E_\alpha}$	$\overline{E_c}$	$\overline{E_{r,\alpha}}$	$\overline{E_{r,c}}$	$\overline{E_{\alpha,c}}$	$ U - U' $	$\overline{E_\cap}$
512	64	0	64	0	64	0	0	0

De acordo com a Figura 53 e com a Tabela 50, é possível observar que apenas elementos de menor qualidade em relação à razão entre arestas e ao coeficiente de compacidade foram identificados para a placa de orifício multi-blocos, considerando as porcentagens utilizadas \bar{p} . Além disso, esses elementos também pertencem ao conjunto formado pela intersecção entre $\overline{E_r}$ e $\overline{E_c}$. Como esperado, os elementos detectados localizam-se no bloco da malha computacional com maior densidade de elementos. Em comparação à

placa de orifício descrita na Seção 4.5, pode-se perceber que, apesar da concentração de elementos detectados em um bloco, a técnica multi-blocos permitiu a melhoria dos elementos de menor qualidade em relação ao ângulo interno, visto que todos os elementos da malha computacional são retangulares.

4.13 Lago Igapó I

Com o propósito de mostrar um exemplo de aplicação mais real, sobre o qual será possível realizar futuras pesquisas, foram coletados pontos de fronteira do Lago Igapó I, localizado em Londrina-PR, utilizando imagens de satélite oferecidas pelo aplicativo *Google Earth* e com o auxílio da ferramenta *WebPlotDigitizer* [36, 38]. Esses pontos de fronteira são ilustrados na Figura 54 e são divididos em três blocos, cujo número de partições nas direções x e y é mostrado na Tabela 51.



Figura 54 – Pontos de fronteira do Lago Igapó I. (Fonte: 36)

Tabela 51 – Número de partições nas direções x e y para cada bloco do Lago Igapó I

Bloco	x	y	Quantidade de nós
1	6	70	152
2	12	2	28
3	16	5	42
Total			222

Por meio do número de partições do plano físico e dos pontos de contorno, executa-se o algoritmo para geração da malha computacional em coordenadas generalizadas envolvendo multi-blocos. A malha computacional do Lago Igapó I é ilustrada na Figura 55, apresentando também a distribuição dos blocos.

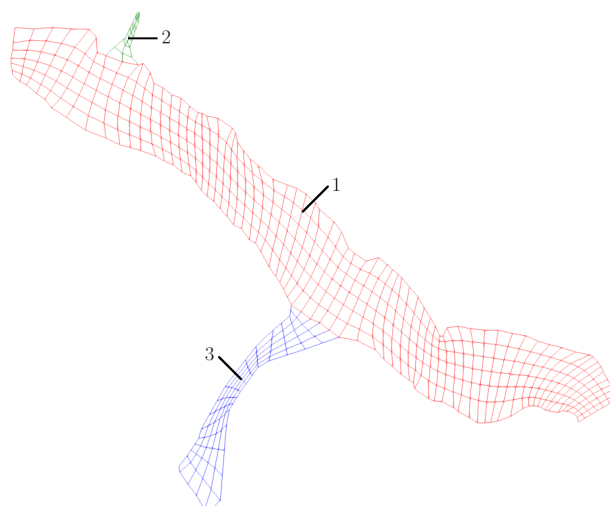


Figura 55 – Malha computacional do Lago Igapó I em coordenadas generalizadas

Na Figura 55, a malha gerada possui a mesma escala que o modelo no domínio físico, constituída pelos pontos $(x_0, y_0) = (482913, 7418777)$ e $(x_n, y_n) = (484447, 7420170)$ em coordenadas planas, os quais delimitam respectivamente o seu início e fim. Além disso, a Tabela 52 relaciona cada bloco do Lago Igapó I com o número de partições nas direções ξ e η .

Tabela 52 – Número de partições nas direções ξ e η para cada bloco do Lago Igapó I

Bloco	ξ	η	Quantidade de nós	Quantidade de elementos
1	6	70	497	420
2	12	2	39	24
3	16	5	102	80
Total			638	524

Utilizando o Lago Igapó I como parâmetro de entrada, executa-se o primeiro módulo para análise de qualidade, através do Algoritmo 6. Após o cálculo das métricas de qualidade para cada elemento de cada bloco da malha computacional, obtém-se o conjunto total de elementos, associados à razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, ao ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e ao coeficiente de compacidade c .

Tendo posse do conjunto total de elementos e suas métricas de qualidade, assim como informações sobre o número de partições do plano transformado para cada bloco e as porcentagens desejadas \bar{p} , executa-se o segundo módulo para análise de qualidade, por meio do Algoritmo 7. Após a leitura dos dados de entrada, obtém-se que $|U'| = |U| = 524$ elementos, pois não há elementos em que $r = +\infty$ e $\alpha = \frac{\pi}{2}$ no conjunto U . Em seguida, a variância de cada métrica de qualidade é calculada, resultando em $s_R^2 = 1.1438$, $s_A^2 = 0.3118$ e $s_C^2 = 13.6566$ para o Lago Igapó I.

Uma vez que $s_R^2 \neq 0$, $s_A^2 \neq 0$ e $s_C^2 \neq 0$, calculam-se os valores mínimo m_{\min} e máximo m_{\max} das métricas de qualidade, cujos resultados estão presentes na Tabela 53.

Tabela 53 – Valores mínimo e máximo das métricas de qualidade do Lago Igapó I

	r	α	c
min	1.0000	0.5113	16.0000
max	17.7670	2.9354	36.3437

Utilizando os valores encontrados na Tabela 53, normalizam-se os valores das métricas de qualidade no intervalo $[0, 10]$. Ainda, o valor das métricas avaliadas sobre um quadrado também é normalizado no intervalo $[0, 10]$, obtendo $r_{\text{norm}_{\square}} = 0.0000$, $\alpha_{\text{norm}_{\square}} = 4.3707$ e $c_{\text{norm}_{\square}} = 0.0000$. Na sequência, calculam-se as variâncias $s_{M_{\text{norm}}}^2$ das métricas de qualidade normalizadas, resultando em $s_{R_{\text{norm}}}^2 = 0.4069$, $s_{A_{\text{norm}}}^2 = 5.3059$ e $s_{C_{\text{norm}}}^2 = 3.2998$.

Com as variâncias $s_{M_{\text{norm}}}^2$ calculadas, parte-se para o cálculo das tolerâncias mínima e máxima, obtendo $\epsilon_{\min} = 0.0001$ e $\epsilon_{\max} = 0.1732$. Iterando entre o intervalo de valores, constrói-se o diagrama de dispersão de ϵ_i por $|E_{\cup_i}|$, o qual é ilustrado na Figura 56 pelos pontos em preto. A curva em verde representa a função de regressão não linear exponencial, encontrada na etapa posterior. No caso do Lago Igapó I, o método de mínimos quadrados obteve, como resultado, os coeficientes $c_1 = 170.9680$ e $c_2 = -39.7771$.

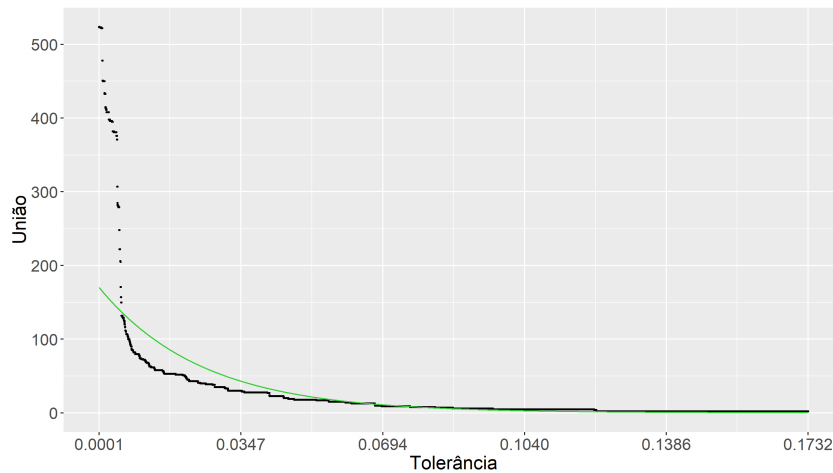


Figura 56 – Diagrama de dispersão de ϵ_i por $|E_{\cup_i}|$ do Lago Igapó I

Dados os coeficientes c_1 e c_2 da função de regressão não linear exponencial, calculam-se as tolerâncias desejadas $\bar{\epsilon}$, considerando $\bar{p} = 5, 10$ e 20% . As porcentagens utilizadas resultaram, respectivamente, em $\bar{\epsilon} = 0.0472, 0.0297$ e 0.0123 . Para cada valor de tolerância desejada $\bar{\epsilon}$, identificam-se os elementos de menor qualidade em relação à razão entre arestas $\overline{E_r}$, ao ângulo interno $\overline{E_\alpha}$ e ao coeficiente de compacidade $\overline{E_c}$. Além disso, por meio de operações envolvendo os conjuntos $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$, obtêm-se os conjuntos $\overline{E_{\cup}}$, $\overline{E_{r,\alpha}}$,

$\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$ e $\overline{E_{\Gamma}}$. Finalmente, um número q é atribuído para cada categoria de elementos a partir dos conjuntos obtidos.

Após a escrita dos resultados em um arquivo *.vtk* para cada bloco, a Figura 57 ilustra os elementos de menor qualidade identificados para o Lago Igapó I, levando em consideração $\bar{p} = 5, 10$ e 20% , respectivamente, nas Figuras 57a - 57c.

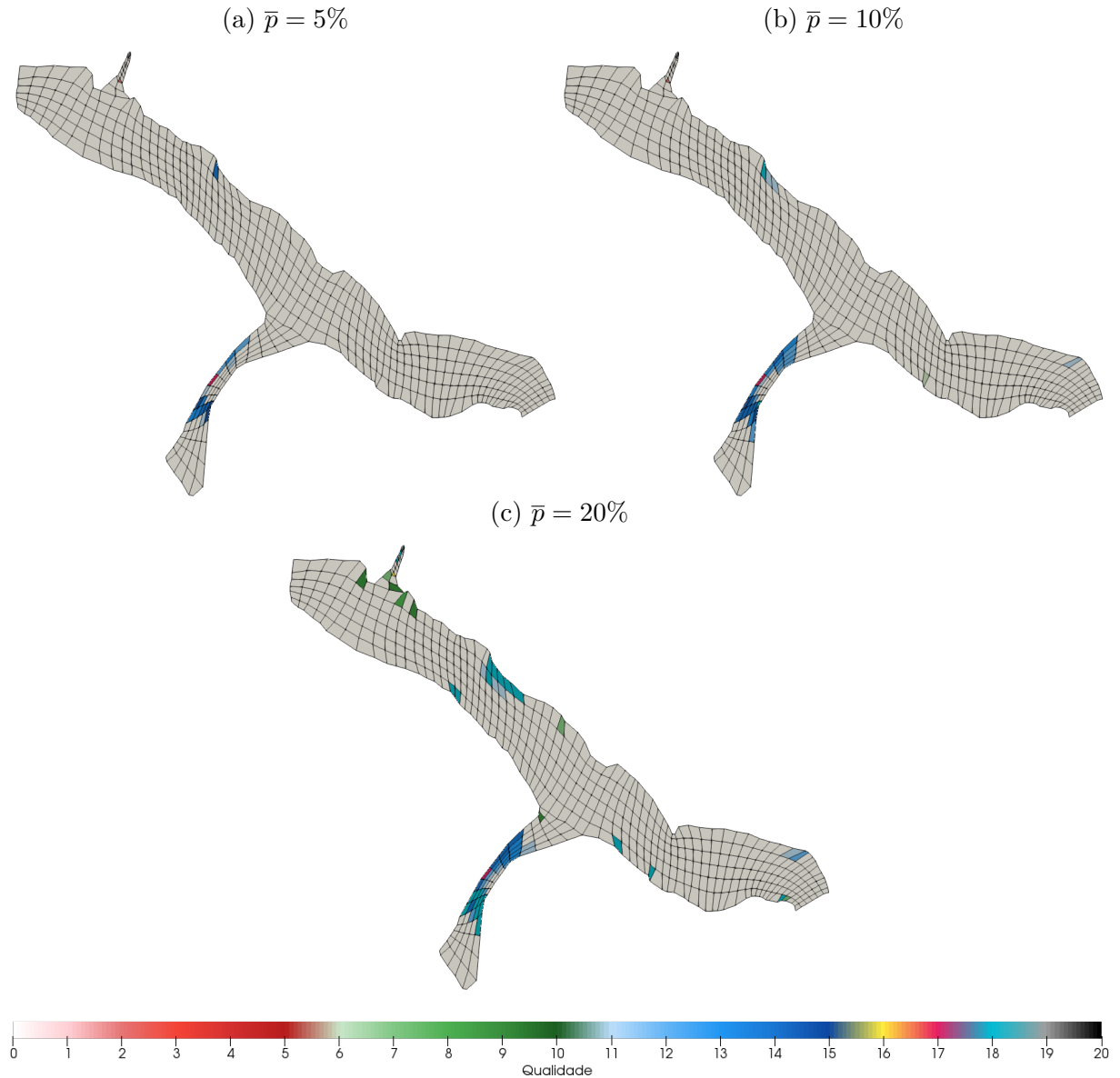


Figura 57 – Elementos de menor qualidade identificados para o Lago Igapó I

Uma outra maneira para se observar os resultados encontrados do Lago Igapó I pode ser representada na Tabela 54, na qual se apresentam as porcentagens desejadas \bar{p} e as tolerâncias desejadas $\bar{\epsilon}$ em relação às quantidades de elementos dos conjuntos $U' - \overline{E_U}$, $\overline{E_r}$, $\overline{E_\alpha}$, $\overline{E_c}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$, $U - U'$ e $\overline{E_\Gamma}$.

Tabela 54 – Quantidades de elementos dos resultados obtidos para o Lago Igapó I

\bar{p}	$\bar{\epsilon}$	$U' - \overline{E_U}$	$\overline{E_r}$	$\overline{E_\alpha}$	$\overline{E_c}$	$\overline{E_{r,\alpha}}$	$\overline{E_{r,c}}$	$\overline{E_{\alpha,c}}$	$ U - U' $	$\overline{E_\Gamma}$
5%	0.0472	505	2	0	18	0	1	0	0	0
10%	0.0297	489	2	4	33	0	1	3	0	0
20%	0.0123	459	2	35	55	1	1	25	0	0

De acordo com a Figura 57 e com a Tabela 54, é possível observar que apenas elementos de menor qualidade em relação à razão entre arestas e ao coeficiente de compacidade foram identificados para o Lago Igapó I, considerando $\bar{p} = 5\%$. Para as porcentagens utilizadas $\bar{p} = 10$ e 20% , elementos de menor qualidade em relação a todas as métricas de avaliação foram identificados. Além disso, para $\bar{p} = 20\%$, destaca-se o número de elementos pertencentes ao conjunto formado pela intersecção entre $\overline{E_\alpha}$ e $\overline{E_c}$. Por conta da complexidade da geometria, não existe um padrão na localização dos elementos detectados. No entanto, pode-se perceber uma evidente concentração de elementos de menor qualidade quanto ao coeficiente de compacidade no bloco 3, devido ao refinamento empregado na direção η , para todas as porcentagens. Para $\bar{p} = 20\%$, surgem elementos de menor qualidade nos demais blocos da malha computacional.

4.14 Golfinho

Nesta seção, modela-se e analisa-se a malha computacional de um golfinho. Nesse experimento, a aquisição dos pontos de fronteira é baseada em um trabalho científico e tem, como auxílio, a ferramenta *WebPlotDigitizer* [35, 38]. Esses pontos de fronteira são ilustrados na Figura 58 e são divididos em seis blocos, cujo número de partições nas direções x e y é mostrado na Tabela 55.



Figura 58 – Pontos de fronteira do golfinho. (Fonte: 35)

Tabela 55 – Número de partições nas direções x e y para cada bloco do golfinho

Bloco	x	y	Quantidade de nós
1	36	9	90
2	4	10	28
3	5	10	30
4	5	8	26
5	6	7	26
6	7	9	32
Total			232

Por meio do número de partições do plano físico e dos pontos de contorno, executa-se o algoritmo para geração da malha computacional em coordenadas generalizadas envolvendo multi-blocos. A malha computacional do golfinho é ilustrada na Figura 59, apresentando também a distribuição dos blocos.

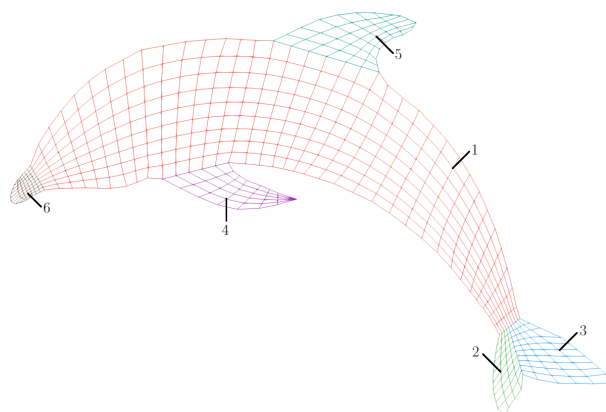


Figura 59 – Malha computacional do golfinho em coordenadas generalizadas

Além disso, a Tabela 56 relaciona cada bloco do golfinho com o número de partições nas direções ξ e η .

Tabela 56 – Número de partições nas direções ξ e η para cada bloco do golfinho

Bloco	ξ	η	Quantidade de nós	Quantidade de elementos
1	36	9	370	324
2	4	10	55	40
3	5	10	66	50
4	5	8	54	40
5	6	7	56	42
6	7	9	80	63
Total			681	559

Utilizando o golfinho como parâmetro de entrada, executa-se o primeiro módulo para análise de qualidade, através do Algoritmo 6. Após o cálculo das métricas de qualidade para cada elemento de cada bloco da malha computacional, obtém-se o conjunto

total de elementos, associados à razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, ao ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e ao coeficiente de compacidade c .

Tendo posse do conjunto total de elementos e suas métricas de qualidade, assim como informações sobre o número de partições do plano transformado para cada bloco e as porcentagens desejadas \bar{p} , executa-se o segundo módulo para análise de qualidade, por meio do Algoritmo 7. Após a leitura dos dados de entrada, obtém-se que $|U'| = 554$ elementos, pois há 5 elementos em que $r = +\infty$ e $\alpha = \nexists$ no conjunto U . Em seguida, a variância de cada métrica de qualidade é calculada, resultando em $s_R^2 = 3.1846$, $s_A^2 = 0.2835$ e $s_C^2 = 21.3201$ para o golfinho.

Uma vez que $s_R^2 \neq 0$, $s_A^2 \neq 0$ e $s_C^2 \neq 0$, calculam-se os valores mínimo m_{\min} e máximo m_{\max} das métricas de qualidade, cujos resultados estão presentes na Tabela 57.

Tabela 57 – Valores mínimo e máximo das métricas de qualidade do golfinho

	r	α	c
min	1.0000	0.3698	16.0000
max	19.8383	3.1316	43.4956

Utilizando os valores encontrados na Tabela 57, normalizam-se os valores das métricas de qualidade no intervalo $[0, 10]$. Ainda, o valor das métricas avaliadas sobre um quadrado também é normalizado no intervalo $[0, 10]$, obtendo $r_{\text{norm}\square} = 0.0000$, $\alpha_{\text{norm}\square} = 4.3485$ e $c_{\text{norm}\square} = 0.0000$. Na sequência, calculam-se as variâncias $s_{M_{\text{norm}}}^2$ das métricas de qualidade normalizadas, resultando em $s_{R_{\text{norm}}}^2 = 0.8974$, $s_{A_{\text{norm}}}^2 = 3.7174$ e $s_{C_{\text{norm}}}^2 = 2.8201$.

Com as variâncias $s_{M_{\text{norm}}}^2$ calculadas, parte-se para o cálculo das tolerâncias mínima e máxima, obtendo $\epsilon_{\min} = 0.0001$ e $\epsilon_{\max} = 0.1591$. Iterando entre o intervalo de valores, constrói-se o diagrama de dispersão de ϵ_i por $|E_{U_i}|$, o qual é ilustrado na Figura 60 pelos pontos em preto. A curva em verde representa a função de regressão não linear exponencial, encontrada na etapa posterior. No caso do golfinho, o método de mínimos quadrados obteve, como resultado, os coeficientes $c_1 = 222.0300$ e $c_2 = -45.6958$.

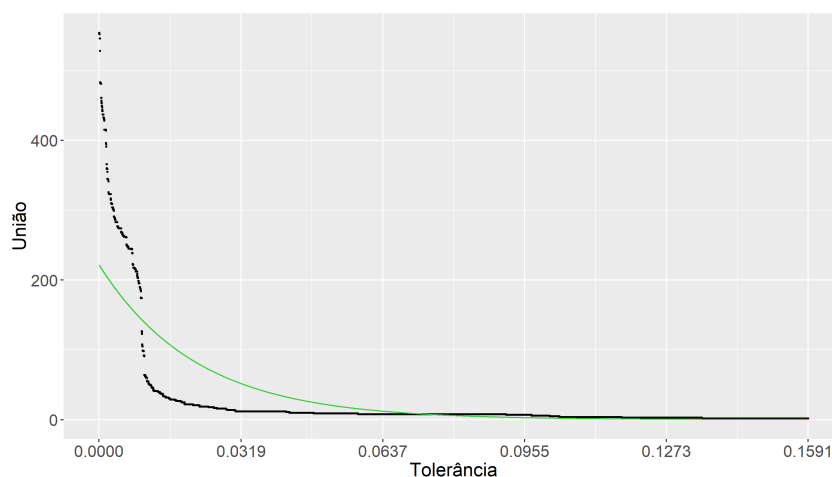


Figura 60 – Diagrama de dispersão de ϵ_i por $|E_{U_i}|$ do golfinho

Dados os coeficientes c_1 e c_2 da função de regressão não linear exponencial, calculam-se as tolerâncias desejadas $\bar{\epsilon}$, considerando $\bar{p} = 5, 10$ e 20% . As porcentagens utilizadas resultaram, respectivamente, em $\bar{\epsilon} = 0.0455, 0.0304$ e 0.0152 . Para cada valor de tolerância desejada $\bar{\epsilon}$, identificam-se os elementos de menor qualidade em relação à razão entre arestas $\overline{E_r}$, ao ângulo interno $\overline{E_\alpha}$ e ao coeficiente de compacidade $\overline{E_c}$. Além disso, por meio de operações envolvendo os conjuntos $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$, obtêm-se os conjuntos $\overline{E_U}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$ e $\overline{E_\cap}$. Finalmente, um número q é atribuído para cada categoria de elementos a partir dos conjuntos obtidos.

Após a escrita dos resultados em um arquivo *.vtk* para cada bloco, a Figura 61 ilustra os elementos de menor qualidade identificados para o golfinho, levando em consideração $\bar{p} = 5, 10$ e 20% , respectivamente, nas Figuras 61a - 61c.

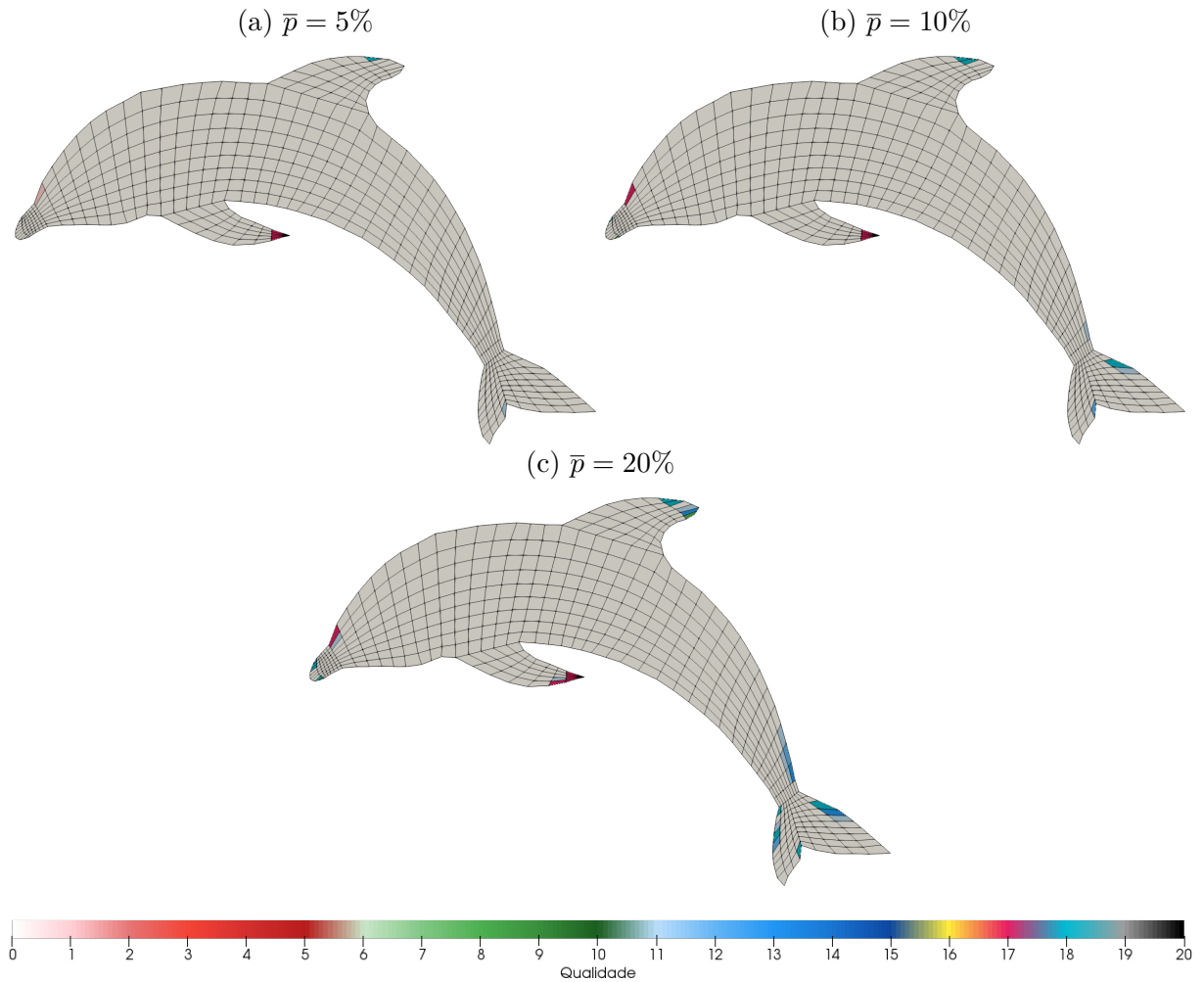


Figura 61 – Elementos de menor qualidade identificados para o golfinho

Uma outra maneira para se observar os resultados encontrados do golfinho pode ser representada na Tabela 58, na qual se apresentam as porcentagens desejadas \bar{p} e as tolerâncias desejadas $\bar{\epsilon}$ em relação às quantidades de elementos dos conjuntos $U' - \overline{E_U}$, $\overline{E_r}$, $\overline{E_\alpha}$, $\overline{E_c}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$, $U - U'$ e $\overline{E_\cap}$.

Tabela 58 – Quantidades de elementos dos resultados obtidos para o golfinho

\bar{p}	$\bar{\epsilon}$	$U' - \overline{E_U}$	$\overline{E_r}$	$\overline{E_\alpha}$	$\overline{E_c}$	$\overline{E_{r,\alpha}}$	$\overline{E_{r,c}}$	$\overline{E_{\alpha,c}}$	$ U - U' $	$\overline{E_\cap}$
5%	0.0455	544	6	2	8	0	5	1	5	0
10%	0.0304	540	6	5	13	0	6	4	5	0
20%	0.0152	522	7	12	30	0	7	10	5	0

De acordo com a Figura 61 e com a Tabela 58, é possível observar que elementos de menor qualidade em relação a todas as métricas de avaliação foram identificados para o golfinho, considerando as porcentagens utilizadas \bar{p} . Além disso, para $\bar{p} = 20\%$, destaca-se o número de elementos pertencentes ao conjunto formado pela intersecção entre $\overline{E_\alpha}$ e $\overline{E_c}$. No bloco 4, existem 5 elementos em que $r = +\infty$ e $\alpha = \nexists$, em virtude dos pontos de

fronteira adquiridos. Por conta da complexidade da geometria, não existe um padrão na localização dos elementos detectados. No entanto, pode-se perceber uma concentração de elementos de menor qualidade no bloco 4, para todas as porcentagens. Para $\bar{p} = 20\%$, surgem elementos de menor qualidade nos demais blocos da malha computacional.

4.15 Tartaruga

Nesta seção, modela-se e analisa-se a malha computacional de uma tartaruga. Nesse experimento, os pontos de fronteira foram coletados utilizando imagens encontradas no *Google* e com o auxílio da ferramenta *WebPlotDigitizer* [37, 38]. Esses pontos de fronteira são ilustrados na Figura 62 e são divididos em seis blocos, cujo número de partições nas direções x e y é mostrado na Tabela 59.

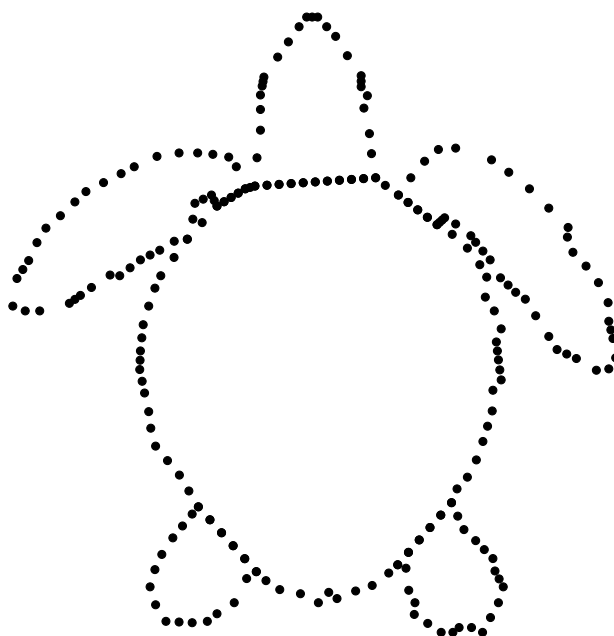


Figura 62 – Pontos de fronteira da tartaruga. (Fonte: 37)

Tabela 59 – Número de partições nas direções x e y para cada bloco da tartaruga

Bloco	x	y	Quantidade de nós
1	10	7	34
2	10	31	82
3	4	17	42
4	4	17	42
5	5	5	20
6	5	8	26
Total			246

Por meio do número de partições do plano físico e dos pontos de contorno, executa-se o algoritmo para geração da malha computacional em coordenadas generalizadas en-

volvendo multi-blocos. A malha computacional da tartaruga é ilustrada na Figura 63, apresentando também a distribuição dos blocos.

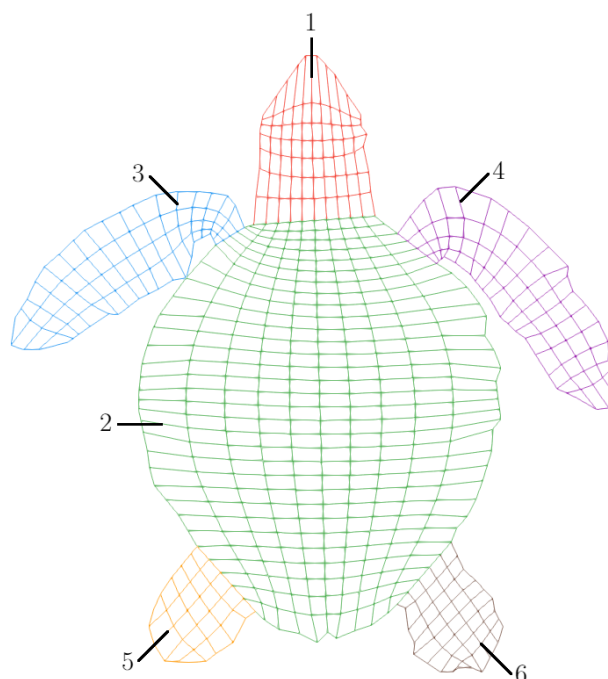


Figura 63 – Malha computacional da tartaruga em coordenadas generalizadas

Além disso, a Tabela 60 relaciona cada bloco da tartaruga com o número de partições nas direções ξ e η .

Tabela 60 – Número de partições nas direções ξ e η para cada bloco da tartaruga

Bloco	ξ	η	Quantidade de nós	Quantidade de elementos
1	10	7	88	70
2	10	31	352	310
3	4	17	90	68
4	4	17	90	68
5	5	5	36	25
6	5	8	54	40
Total			710	581

Utilizando a tartaruga como parâmetro de entrada, executa-se o primeiro módulo para análise de qualidade, através do Algoritmo 6. Após o cálculo das métricas de qualidade para cada elemento de cada bloco da malha computacional, obtém-se o conjunto total de elementos, associados à razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, ao ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e ao coeficiente de compacidade c .

Tendo posse do conjunto total de elementos e suas métricas de qualidade, assim como informações sobre o número de partições do plano transformado para cada bloco e

as porcentagens desejadas \bar{p} , executa-se o segundo módulo para análise de qualidade, por meio do Algoritmo 7. Após a leitura dos dados de entrada, obtém-se que $|U'| = |U| = 581$ elementos, pois não há elementos em que $r = +\infty$ e $\alpha = \#$ no conjunto U . Em seguida, a variância de cada métrica de qualidade é calculada, resultando em $s_R^2 = 0.6661$, $s_A^2 = 0.1676$ e $s_C^2 = 9.5647$ para a tartaruga.

Uma vez que $s_R^2 \neq 0$, $s_A^2 \neq 0$ e $s_C^2 \neq 0$, calculam-se os valores mínimo m_{\min} e máximo m_{\max} das métricas de qualidade, cujos resultados estão presentes na Tabela 61.

Tabela 61 – Valores mínimo e máximo das métricas de qualidade da tartaruga

	r	α	c
min	1.0000	0.2053	16.0000
max	8.8443	3.0485	42.3076

Utilizando os valores encontrados na Tabela 61, normalizam-se os valores das métricas de qualidade no intervalo $[0, 10]$. Ainda, o valor das métricas avaliadas sobre um quadrado também é normalizado no intervalo $[0, 10]$, obtendo $r_{\text{norm}\square} = 0.0000$, $\alpha_{\text{norm}\square} = 4.8028$ e $c_{\text{norm}\square} = 0.0000$. Na sequência, calculam-se as variâncias $s_{M_{\text{norm}}}^2$ das métricas de qualidade normalizadas, resultando em $s_{R_{\text{norm}}}^2 = 1.0826$, $s_{A_{\text{norm}}}^2 = 2.0727$ e $s_{C_{\text{norm}}}^2 = 1.3820$.

Com as variâncias $s_{M_{\text{norm}}}^2$ calculadas, parte-se para o cálculo das tolerâncias mínima e máxima, obtendo $\epsilon_{\min} = 0.0001$ e $\epsilon_{\max} = 0.1369$. Iterando entre o intervalo de valores, constrói-se o diagrama de dispersão de ϵ_i por $|E_{U_i}|$, o qual é ilustrado na Figura 64 pelos pontos em preto. A curva em verde representa a função de regressão não linear exponencial, encontrada na etapa posterior. No caso da tartaruga, o método de mínimos quadrados obteve, como resultado, os coeficientes $c_1 = 216.0572$ e $c_2 = -53.8616$.

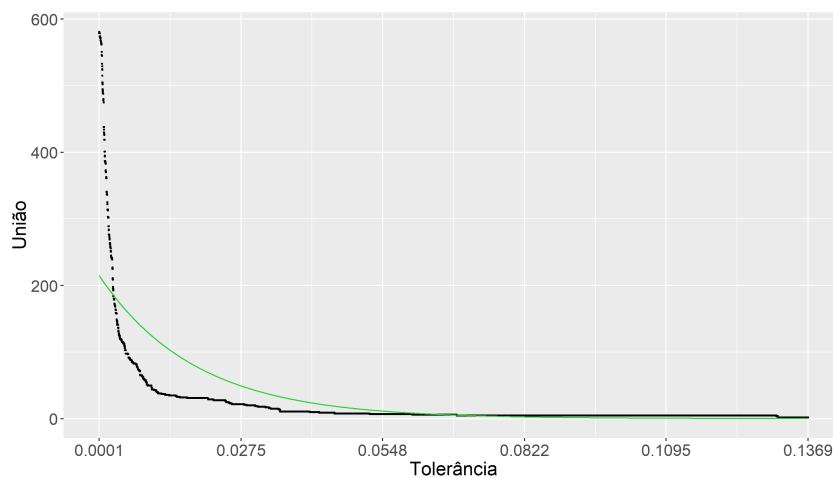


Figura 64 – Diagrama de dispersão de ϵ_i por $|E_{U_i}|$ da tartaruga

Dados os coeficientes c_1 e c_2 da função de regressão não linear exponencial, calcu-

lam-se as tolerâncias desejadas $\bar{\epsilon}$, considerando $\bar{p} = 5, 10$ e 20% . As porcentagens utilizadas resultaram, respectivamente, em $\bar{\epsilon} = 0.0373, 0.0244$ e 0.0115 . Para cada valor de tolerância desejada $\bar{\epsilon}$, identificam-se os elementos de menor qualidade em relação à razão entre arestas $\overline{E_r}$, ao ângulo interno $\overline{E_\alpha}$ e ao coeficiente de compacidade $\overline{E_c}$. Além disso, por meio de operações envolvendo os conjuntos $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$, obtêm-se conjuntos $\overline{E_\cup}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$ e $\overline{E_\cap}$. Finalmente, um número q é atribuído para cada categoria de elementos a partir dos conjuntos obtidos.

Após a escrita dos resultados em um arquivo *.vtk* para cada bloco, a Figura 65 ilustra os elementos de menor qualidade identificados para a tartaruga, levando em consideração $\bar{p} = 5, 10$ e 20% , respectivamente, nas Figuras 65a - 65c.

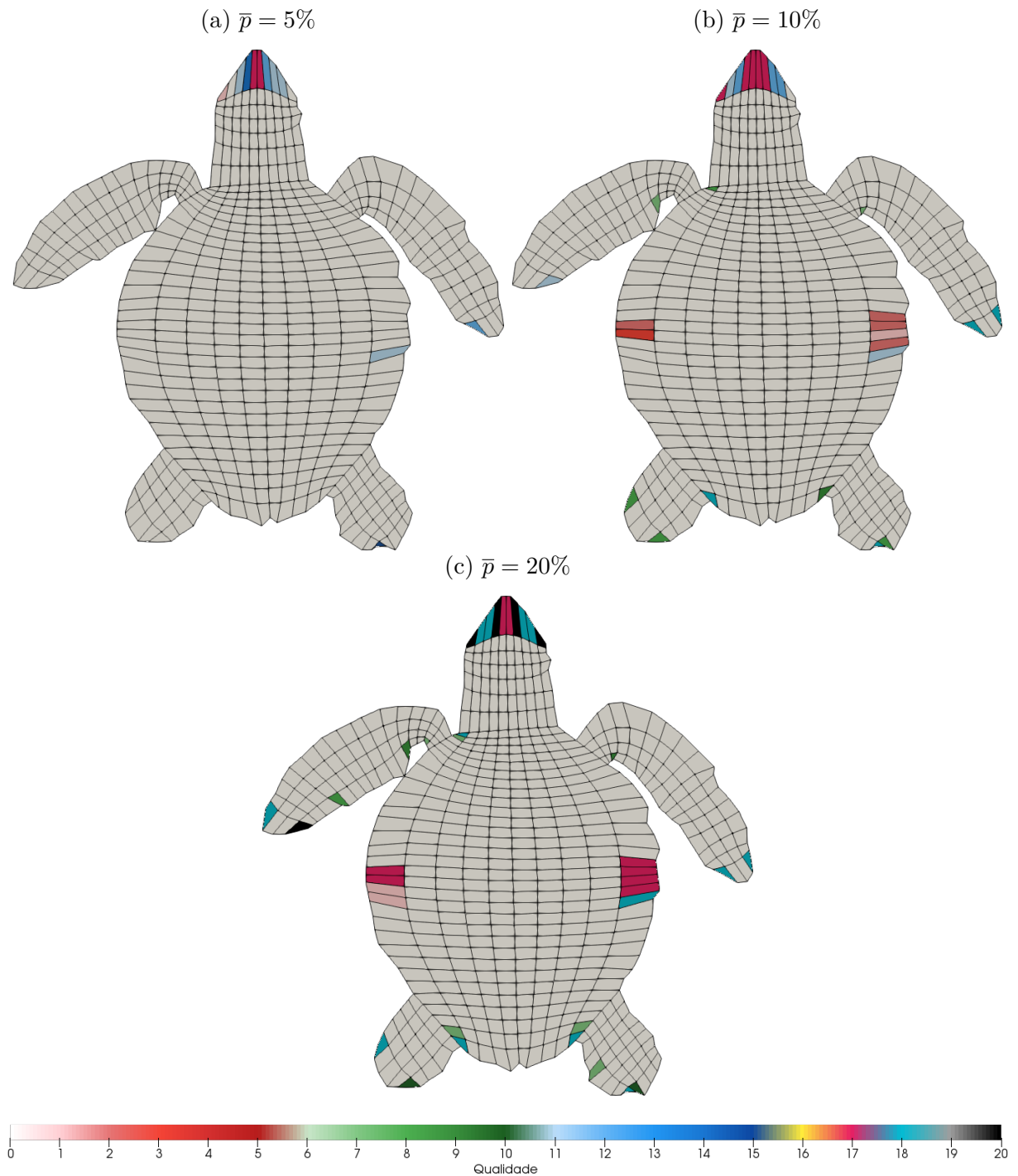


Figura 65 – Elementos de menor qualidade identificados para a tartaruga

Uma outra maneira para se observar os resultados encontrados da tartaruga pode ser representada na Tabela 62, na qual se apresentam as porcentagens desejadas \bar{p} e as tolerâncias desejadas $\bar{\epsilon}$ em relação às quantidades de elementos dos conjuntos $U' - \overline{E_U}$, $\overline{E_r}$, $\overline{E_\alpha}$, $\overline{E_c}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$, $U - U'$ e $\overline{E_\Omega}$.

Tabela 62 – Quantidades de elementos dos resultados obtidos para a tartaruga

\bar{p}	$\bar{\epsilon}$	$U' - \overline{E_U}$	$\overline{E_r}$	$\overline{E_\alpha}$	$\overline{E_c}$	$\overline{E_{r,\alpha}}$	$\overline{E_{r,c}}$	$\overline{E_{\alpha,c}}$	$ U - U' $	$\overline{E_\Gamma}$
5%	0.0373	570	3	0	10	0	2	0	0	0
10%	0.0244	553	11	11	15	0	5	4	0	0
20%	0.0115	543	15	28	26	5	13	18	0	5

De acordo com a Figura 65 e com a Tabela 62, é possível observar que apenas elementos de menor qualidade em relação à razão entre arestas e ao coeficiente de compatibilidade foram identificados para a tartaruga, considerando $\bar{p} = 5\%$. Para as porcentagens utilizadas $\bar{p} = 10$ e 20% , elementos de menor qualidade em relação a todas as métricas de avaliação foram identificados. Além disso, para $\bar{p} = 20\%$, destaca-se o número de elementos pertencentes aos conjuntos $\overline{E_{r,c}}$ e $\overline{E_{\alpha,c}}$. Por conta da complexidade da geometria, não existe um padrão na localização dos elementos detectados. No entanto, pode-se perceber uma evidente concentração de elementos de menor qualidade no bloco 1, para todas as porcentagens. Para $\bar{p} = 10$ e 20% , surgem elementos de menor qualidade na região central das fronteiras esquerda e direita do bloco 2 da malha computacional.

4.16 Perfil de um Rosto 2D

Nesta seção, modela-se e analisa-se a malha computacional do perfil de um rosto bidimensional. Nesse experimento, os pontos de fronteira foram coletados utilizando imagens encontradas no *Google* e com o auxílio da ferramenta *WebPlotDigitizer* [37, 38]. Esses pontos de fronteira são ilustrados na Figura 66 e são divididos em sete blocos, cujo número de partições nas direções x e y é mostrado na Tabela 63.

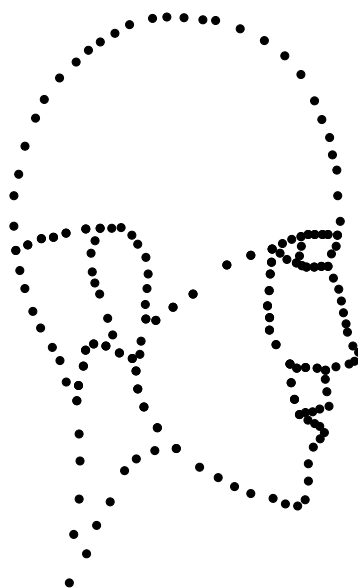


Figura 66 – Pontos de fronteira do perfil de um rosto. (Fonte: 37)

Tabela 63 – Número de partições nas direções x e y para cada bloco do perfil de um rosto

Bloco	x	y	Quantidade de nós
1	3	4	14
2	15	6	42
3	8	8	32
4	3	5	16
5	8	6	28
6	15	14	58
7	5	7	24
Total			214

Por meio do número de partições do plano físico e dos pontos de contorno, executa-se o algoritmo para geração da malha computacional em coordenadas generalizadas envolvendo multi-blocos. A malha computacional do perfil de um rosto é ilustrada na Figura 67, apresentando também a distribuição dos blocos.

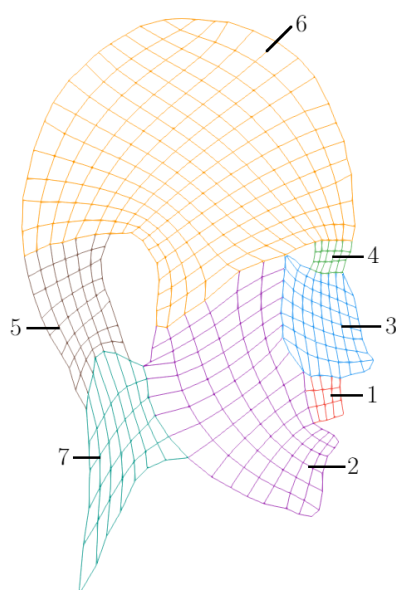


Figura 67 – Malha computacional do perfil de um rosto em coordenadas generalizadas

Além disso, a Tabela 64 relaciona cada bloco do perfil de um rosto com o número de partições nas direções ξ e η .

Tabela 64 – Número de partições nas direções ξ e η para cada bloco do perfil de um rosto

Bloco	ξ	η	Quantidade de nós	Quantidade de elementos
1	3	4	20	12
2	15	6	112	90
3	8	8	81	64
4	3	5	24	15
5	8	6	63	48
6	15	14	240	210
7	5	7	48	35
Total			588	474

Utilizando o perfil de um rosto como parâmetro de entrada, executa-se o primeiro módulo para análise de qualidade, através do Algoritmo 6. Após o cálculo das métricas de qualidade para cada elemento de cada bloco da malha computacional, obtém-se o conjunto total de elementos, associados à razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, ao ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e ao coeficiente de compacidade c .

Tendo posse do conjunto total de elementos e suas métricas de qualidade, assim como informações sobre o número de partições do plano transformado para cada bloco e as porcentagens desejadas \bar{p} , executa-se o segundo módulo para análise de qualidade, por meio do Algoritmo 7. Após a leitura dos dados de entrada, obtém-se que $|U'| = |U| = 474$ elementos, pois não há elementos em que $r = +\infty$ e $\alpha = \frac{\pi}{2}$ no conjunto U . Em seguida, a variância de cada métrica de qualidade é calculada, resultando em $s_R^2 = 0.2179$, $s_A^2 = 0.2224$ e $s_C^2 = 16.1769$ para o perfil de um rosto.

Uma vez que $s_R^2 \neq 0$, $s_A^2 \neq 0$ e $s_C^2 \neq 0$, calculam-se os valores mínimo m_{\min} e máximo m_{\max} das métricas de qualidade, cujos resultados estão presentes na Tabela 65.

Tabela 65 – Valores mínimo e máximo das métricas de qualidade do perfil de um rosto

	r	α	c
min	1.0000	0.3705	16.0000
max	4.6627	3.1200	48.3569

Utilizando os valores encontrados na Tabela 65, normalizam-se os valores das métricas de qualidade no intervalo $[0, 10]$. Ainda, o valor das métricas avaliadas sobre um quadrado também é normalizado no intervalo $[0, 10]$, obtendo $r_{\text{norm}_{\square}} = 0.0000$, $\alpha_{\text{norm}_{\square}} = 4.3655$ e $c_{\text{norm}_{\square}} = 0.0000$. Na sequência, calculam-se as variâncias $s_{M_{\text{norm}}}^2$ das métricas de qualidade normalizadas, resultando em $s_{R_{\text{norm}}}^2 = 1.6241$, $s_{A_{\text{norm}}}^2 = 2.9415$ e $s_{C_{\text{norm}}}^2 = 1.5451$.

Com as variâncias $s_{M_{\text{norm}}}^2$ calculadas, parte-se para o cálculo das tolerâncias mínima e máxima, obtendo $\epsilon_{\min} = 0.0001$ e $\epsilon_{\max} = 0.1787$. Iterando entre o intervalo de valores,

constrói-se o diagrama de dispersão de ϵ_i por $|E_{U_i}|$, o qual é ilustrado na Figura 68 pelos pontos em preto. A curva em verde representa a função de regressão não linear exponencial, encontrada na etapa posterior. No caso do perfil de um rosto, o método de mínimos quadrados obteve, como resultado, os coeficientes $c_1 = 176.1278$ e $c_2 = -37.6157$.

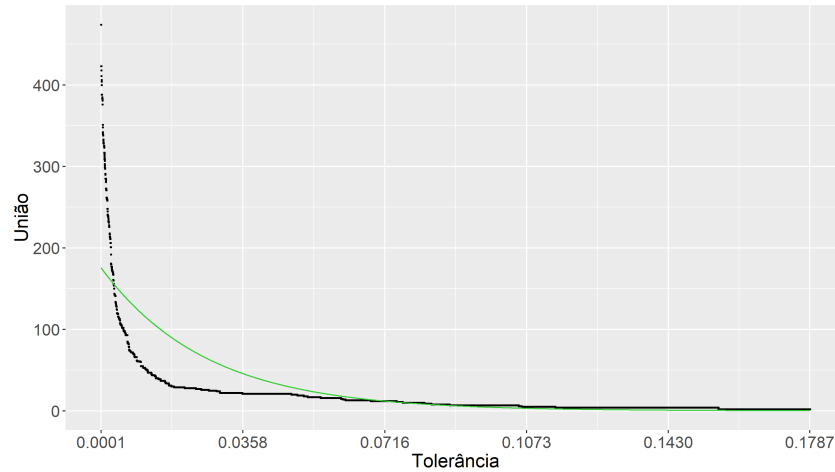


Figura 68 – Diagrama de dispersão de ϵ_i por $|E_{U_i}|$ do perfil de um rosto

Dados os coeficientes c_1 e c_2 da função de regressão não linear exponencial, calculam-se as tolerâncias desejadas $\bar{\epsilon}$, considerando $\bar{p} = 5, 10$ e 20% . As porcentagens utilizadas resultaram, respectivamente, em $\bar{\epsilon} = 0.0533, 0.0349$ e 0.0165 . Para cada valor de tolerância desejada $\bar{\epsilon}$, identificam-se os elementos de menor qualidade em relação à razão entre arestas $\overline{E_r}$, ao ângulo interno $\overline{E_\alpha}$ e ao coeficiente de compacidade $\overline{E_c}$. Além disso, por meio de operações envolvendo os conjuntos $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$, obtêm-se os conjuntos $\overline{E_U}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$ e $\overline{E_\cap}$. Finalmente, um número q é atribuído para cada categoria de elementos a partir dos conjuntos obtidos.

Após a escrita dos resultados em um arquivo *.vtk* para cada bloco, a Figura 69 ilustra os elementos de menor qualidade identificados para o perfil de um rosto, levando em consideração $\bar{p} = 5, 10$ e 20% , respectivamente, nas Figuras 69a - 69c.

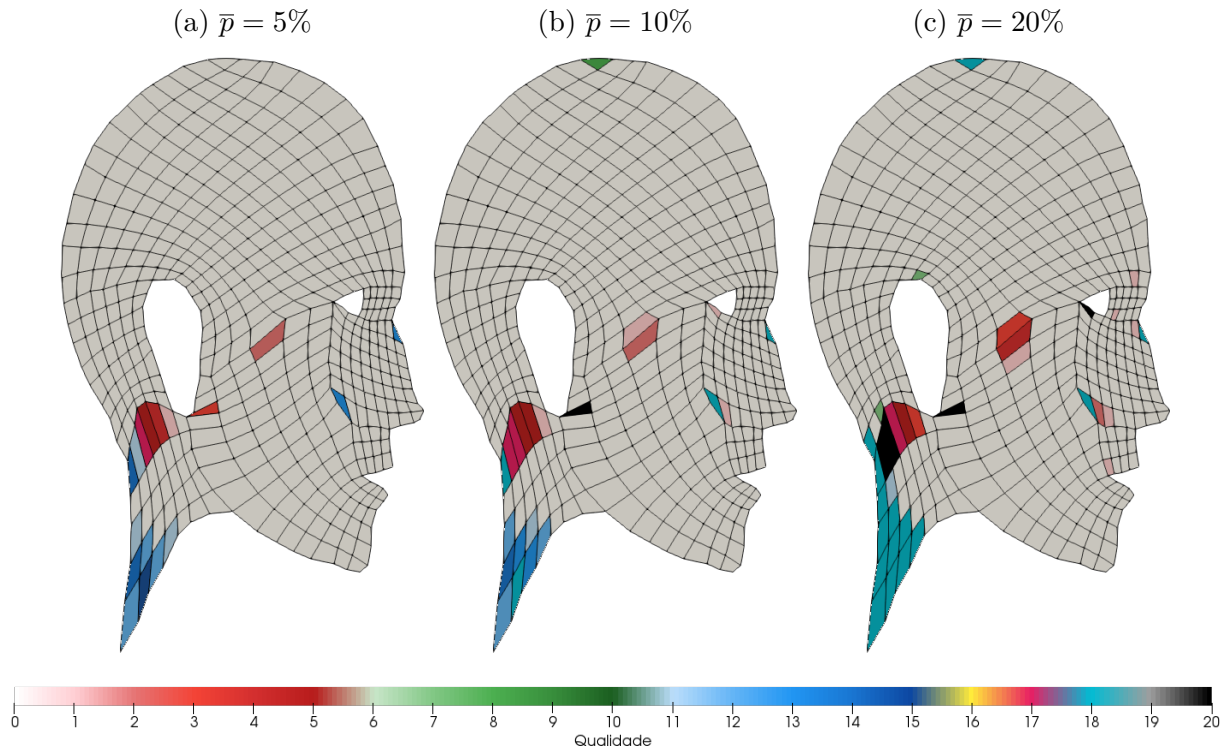


Figura 69 – Elementos de menor qualidade identificados para o perfil de um rosto

Uma outra maneira para se observar os resultados encontrados do perfil de um rosto pode ser representada na Tabela 66, na qual se apresentam as porcentagens desejadas \bar{p} e as tolerâncias desejadas $\bar{\epsilon}$ em relação às quantidades de elementos dos conjuntos $U' - \overline{E_U}$, $\overline{E_r}$, $\overline{E_\alpha}$, $\overline{E_c}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$, $U - U'$ e $\overline{E_\cap}$.

Tabela 66 – Quantidades de elementos dos resultados obtidos para o perfil de um rosto

\bar{p}	$\bar{\epsilon}$	$U' - \overline{E_U}$	$\overline{E_r}$	$\overline{E_\alpha}$	$\overline{E_c}$	$\overline{E_{r,\alpha}}$	$\overline{E_{r,c}}$	$\overline{E_{\alpha,c}}$	$ U - U' $	$\overline{E_\cap}$
5%	0.0533	457	6	0	12	0	1	0	0	0
10%	0.0349	452	10	6	14	1	3	5	0	1
20%	0.0165	440	16	21	21	4	5	19	0	4

De acordo com a Figura 69 e com a Tabela 66, é possível observar que apenas elementos de menor qualidade em relação à razão entre arestas e ao coeficiente de compacidade foram identificados para o perfil de um rosto, considerando $\bar{p} = 5\%$. Para as porcentagens utilizadas $\bar{p} = 10$ e 20% , elementos de menor qualidade em relação a todas as métricas de avaliação foram identificados. Além disso, para $\bar{p} = 20\%$, destaca-se o número de elementos pertencentes ao conjunto formado pela intersecção entre $\overline{E_\alpha}$ e $\overline{E_c}$. Por conta da complexidade da geometria, não existe um padrão na localização dos elementos detectados. No entanto, pode-se perceber uma evidente concentração de elementos de menor qualidade no bloco 7 da malha computacional, para todas as porcentagens.

4.17 Pássaro

Nesta seção, modela-se e analisa-se a malha computacional de um pássaro. Nesse experimento, a aquisição dos pontos de fronteira é baseada em um trabalho científico e tem, como auxílio, a ferramenta *WebPlotDigitizer* [35, 38]. Esses pontos de fronteira são ilustrados na Figura 70 e são divididos em sete blocos, cujo número de partições nas direções x e y é mostrado na Tabela 67.

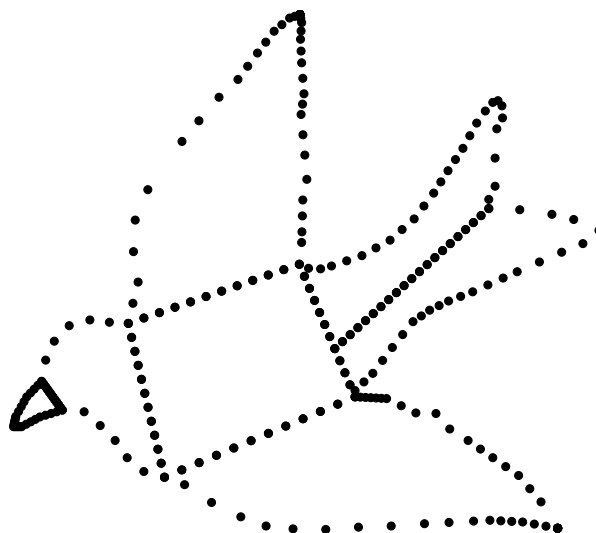


Figura 70 – Pontos de fronteira do pássaro. (Fonte: 35)

Tabela 67 – Número de partições nas direções x e y para cada bloco do pássaro

Bloco	x	y	Quantidade de nós
1	6	11	34
2	11	11	44
3	20	4	48
4	20	7	54
5	11	18	58
6	18	11	58
7	8	11	38
Total			334

Por meio do número de partições do plano físico e dos pontos de contorno, executa-se o algoritmo para geração da malha computacional em coordenadas generalizadas envolvendo multi-blocos. A malha computacional do pássaro é ilustrada na Figura 71, apresentando também a distribuição dos blocos.

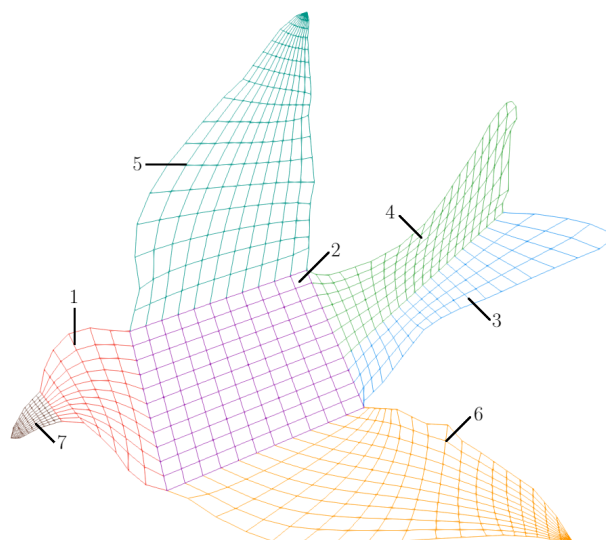


Figura 71 – Malha computacional do pássaro em coordenadas generalizadas

Além disso, a Tabela 68 relaciona cada bloco do pássaro com o número de partições nas direções ξ e η .

Tabela 68 – Número de partições nas direções ξ e η para cada bloco do pássaro

Bloco	ξ	η	Quantidade de nós	Quantidade de elementos
1	6	11	84	66
2	11	11	144	121
3	20	4	105	80
4	20	7	168	140
5	11	18	228	198
6	18	11	228	198
7	8	11	108	88
Total			1065	891

Utilizando o pássaro como parâmetro de entrada, executa-se o primeiro módulo para análise de qualidade, através do Algoritmo 6. Após o cálculo das métricas de qualidade para cada elemento de cada bloco da malha computacional, obtém-se o conjunto total de elementos, associados à razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, ao ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e ao coeficiente de compacidade c .

Tendo posse do conjunto total de elementos e suas métricas de qualidade, assim como informações sobre o número de partições do plano transformado para cada bloco e as porcentagens desejadas \bar{p} , executa-se o segundo módulo para análise de qualidade, por meio do Algoritmo 7. Após a leitura dos dados de entrada, obtém-se que $|U'| = 869$ elementos, pois há 22 elementos em que $r = +\infty$ e $\alpha = \nexists$ no conjunto U . Em seguida, a variância de cada métrica de qualidade é calculada, resultando em $s_R^2 = 2.6559$, $s_A^2 = 0.2930$ e $s_C^2 = 38.3952$ para o pássaro.

Uma vez que $s_R^2 \neq 0$, $s_A^2 \neq 0$ e $s_C^2 \neq 0$, calculam-se os valores mínimo m_{\min} e máximo m_{\max} das métricas de qualidade, cujos resultados estão presentes na Tabela 69.

Tabela 69 – Valores mínimo e máximo das métricas de qualidade do pássaro

	r	α	c
min	1.0000	0.3241	16.0000
max	11.7254	3.1035	67.9769

Utilizando os valores encontrados na Tabela 69, normalizam-se os valores das métricas de qualidade no intervalo $[0, 10]$. Ainda, o valor das métricas avaliadas sobre um quadrado também é normalizado no intervalo $[0, 10]$, obtendo $r_{\text{norm}\square} = 0.0000$, $\alpha_{\text{norm}\square} = 4.4855$ e $c_{\text{norm}\square} = 0.0000$. Na sequência, calculam-se as variâncias $s_{M_{\text{norm}}}^2$ das métricas de qualidade normalizadas, resultando em $s_{R_{\text{norm}}}^2 = 2.3088$, $s_{A_{\text{norm}}}^2 = 3.7924$ e $s_{C_{\text{norm}}}^2 = 1.4212$.

Com as variâncias $s_{M_{\text{norm}}}^2$ calculadas, parte-se para o cálculo das tolerâncias mínima e máxima, obtendo $\epsilon_{\min} = 0.0000$ e $\epsilon_{\max} = 0.0903$. Iterando entre o intervalo de valores, constrói-se o diagrama de dispersão de ϵ_i por $|E_{U_i}|$, o qual é ilustrado na Figura 72 pelos pontos em preto. A curva em verde representa a função de regressão não linear exponencial, encontrada na etapa posterior. No caso do pássaro, o método de mínimos quadrados obteve, como resultado, os coeficientes $c_1 = 506.3437$ e $c_2 = -90.0219$.

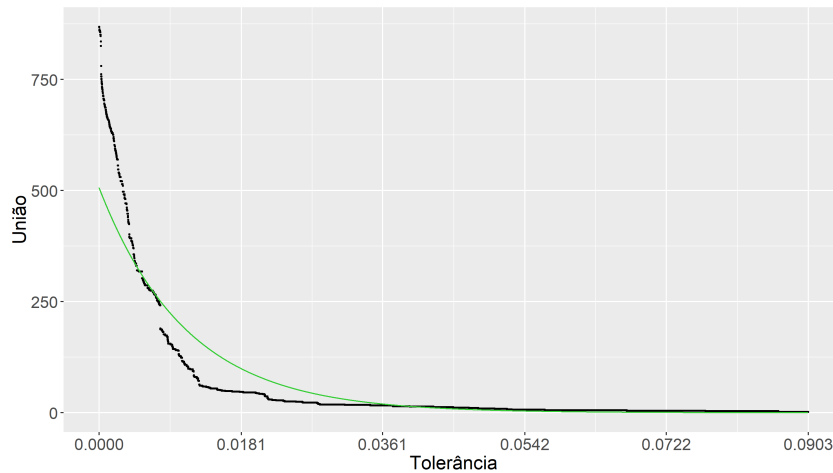


Figura 72 – Diagrama de dispersão de ϵ_i por $|E_{U_i}|$ do pássaro

Dados os coeficientes c_1 e c_2 da função de regressão não linear exponencial, calculam-se as tolerâncias desejadas $\bar{\epsilon}$, considerando $\bar{p} = 5, 10$ e 20% . As porcentagens utilizadas resultaram, respectivamente, em $\bar{\epsilon} = 0.0273, 0.0196$ e 0.0119 . Para cada valor de tolerância desejada $\bar{\epsilon}$, identificam-se os elementos de menor qualidade em relação à razão entre arestas $\overline{E_r}$, ao ângulo interno $\overline{E_\alpha}$ e ao coeficiente de compacidade $\overline{E_c}$. Além disso, por meio de operações envolvendo os conjuntos $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$, obtêm-se os conjuntos $\overline{E_U}$, $\overline{E_{r,\alpha}}$,

$\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$ e $\overline{E_{\Gamma}}$. Finalmente, um número q é atribuído para cada categoria de elementos a partir dos conjuntos obtidos.

Após a escrita dos resultados em um arquivo *.vtk* para cada bloco, a Figura 73 ilustra os elementos de menor qualidade identificados para o pássaro, levando em consideração $\bar{p} = 5, 10$ e 20% , respectivamente, nas Figuras 73a - 73c.

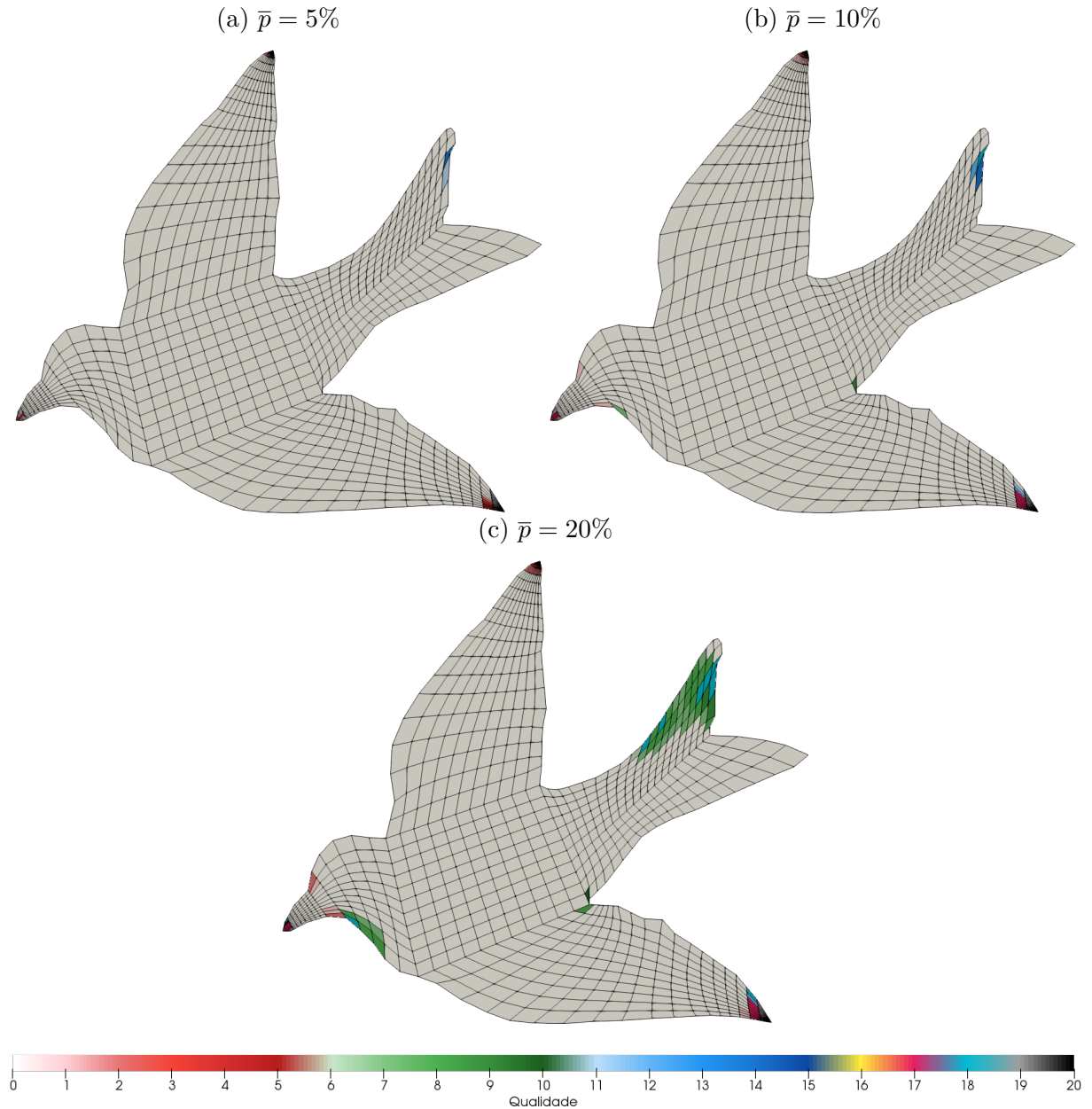


Figura 73 – Elementos de menor qualidade identificados para o pássaro

Uma outra maneira para se observar os resultados encontrados do pássaro pode ser representada na Tabela 70, na qual se apresentam as porcentagens desejadas \bar{p} e as tolerâncias desejadas $\bar{\epsilon}$ em relação às quantidades de elementos dos conjuntos $U' - \overline{E_U}$, $\overline{E_r}$, $\overline{E_\alpha}$, $\overline{E_c}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$, $U - U'$ e $\overline{E_\Gamma}$.

Tabela 70 – Quantidades de elementos dos resultados obtidos para o pássaro

\bar{p}	$\bar{\epsilon}$	$U' - \overline{E_U}$	$\overline{E_r}$	$\overline{E_\alpha}$	$\overline{E_c}$	$\overline{E_{r,\alpha}}$	$\overline{E_{r,c}}$	$\overline{E_{\alpha,c}}$	$ U - U' $	$\overline{E_\Gamma}$
5%	0.0273	846	21	0	4	0	2	0	22	0
10%	0.0196	823	38	4	18	0	13	1	22	0
20%	0.0119	771	48	50	27	1	16	11	22	1

De acordo com a Figura 73 e com a Tabela 70, é possível observar que apenas elementos de menor qualidade em relação à razão entre arestas e ao coeficiente de compacidade foram identificados para o pássaro, considerando $\bar{p} = 5\%$. Para as porcentagens utilizadas $\bar{p} = 10$ e 20% , elementos de menor qualidade em relação a todas as métricas de avaliação foram identificados. Além disso, para essas porcentagens, destaca-se o número de elementos pertencentes aos conjuntos $\overline{E_{r,c}}$ e $\overline{E_{\alpha,c}}$. Nos blocos 5 e 6, totalizam-se 22 elementos em que $r = +\infty$ e $\alpha = \#$, em virtude dos pontos de fronteira adquiridos. Por conta da complexidade da geometria, não existe um padrão na localização dos elementos detectados. No entanto, pode-se perceber uma concentração de elementos de menor qualidade nos blocos 5 e 6, para todas as porcentagens. Para $\bar{p} = 10$ e 20% , surgem elementos de menor qualidade no bloco 7 da malha computacional. Finalmente, apenas para $\bar{p} = 20\%$, surgem elementos de menor qualidade nos blocos 1 e 4, relacionados principalmente ao ângulo interno.

4.18 Cachorro

Nesta seção, modela-se e analisa-se a malha computacional de um cachorro. Nesse experimento, os pontos de fronteira foram coletados utilizando imagens encontradas no *Google* e com o auxílio da ferramenta *WebPlotDigitizer* [37, 38]. Esses pontos de fronteira são ilustrados na Figura 74 e são divididos em oito blocos, cujo número de partições nas direções x e y é mostrado na Tabela 71.

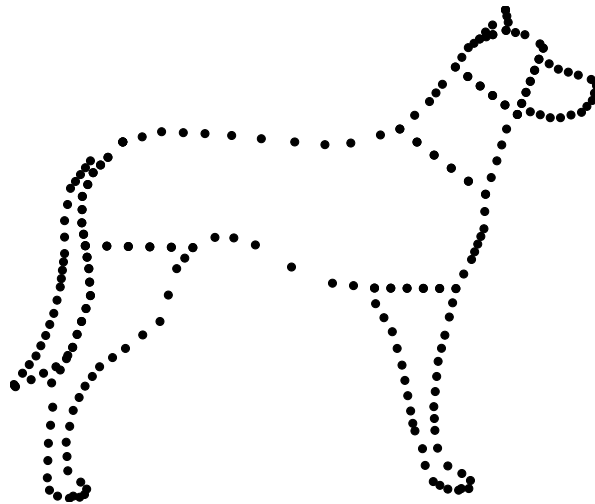


Figura 74 – Pontos de fronteira do cachorro. (Fonte: 37)

Tabela 71 – Número de partições nas direções x e y para cada bloco do cachorro

Bloco	x	y	Quantidade de nós
1	5	6	22
2	6	6	24
3	2	3	10
4	5	5	20
5	17	8	50
6	2	22	48
7	5	14	38
8	5	18	46
Total			258

Por meio do número de partições do plano físico e dos pontos de contorno, executa-se o algoritmo para geração da malha computacional em coordenadas generalizadas envolvendo multi-blocos. A malha computacional do cachorro é ilustrada na Figura 75, apresentando também a distribuição dos blocos.

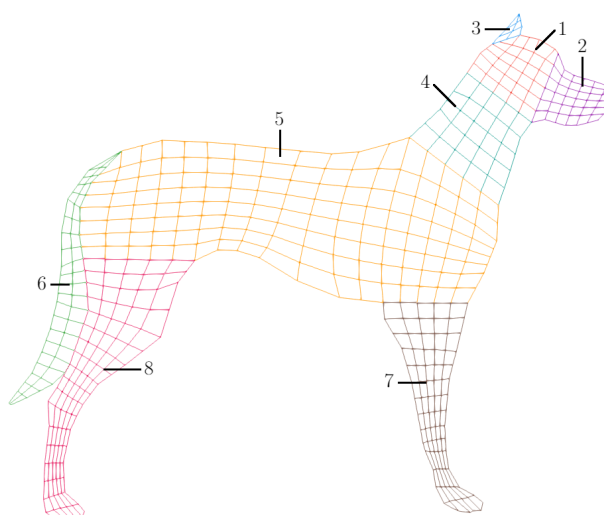


Figura 75 – Malha computacional do cachorro em coordenadas generalizadas

Além disso, a Tabela 72 relaciona cada bloco do cachorro com o número de partições nas direções ξ e η .

Tabela 72 – Número de partições nas direções ξ e η para cada bloco do cachorro

Bloco	ξ	η	Quantidade de nós	Quantidade de elementos
1	5	6	42	30
2	6	6	49	36
3	2	3	12	6
4	5	5	36	25
5	17	8	162	136
6	2	22	69	44
7	5	14	90	70
8	5	18	114	90
Total			574	437

Utilizando o cachorro como parâmetro de entrada, executa-se o primeiro módulo para análise de qualidade, através do Algoritmo 6. Após o cálculo das métricas de qualidade para cada elemento de cada bloco da malha computacional, obtém-se o conjunto total de elementos, associados à razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, ao ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e ao coeficiente de compacidade c .

Tendo posse do conjunto total de elementos e suas métricas de qualidade, assim como informações sobre o número de partições do plano transformado para cada bloco e as porcentagens desejadas \bar{p} , executa-se o segundo módulo para análise de qualidade, por meio do Algoritmo 7. Após a leitura dos dados de entrada, obtém-se que $|U'| = 433$ elementos, pois há 4 elementos em que $r = +\infty$ e $\alpha = \nexists$ no conjunto U . Em seguida, a variância de cada métrica de qualidade é calculada, resultando em $s_R^2 = 1.6786$, $s_A^2 = 0.1467$ e $s_C^2 = 14.4926$ para o cachorro.

Uma vez que $s_R^2 \neq 0$, $s_A^2 \neq 0$ e $s_C^2 \neq 0$, calculam-se os valores mínimo m_{\min} e máximo m_{\max} das métricas de qualidade, cujos resultados estão presentes na Tabela 73.

Tabela 73 – Valores mínimo e máximo das métricas de qualidade do cachorro

	r	α	c
min	1.0000	0.5134	16.0000
max	12.7124	3.1416	36.1177

Utilizando os valores encontrados na Tabela 73, normalizam-se os valores das métricas de qualidade no intervalo $[0, 10]$. Ainda, o valor das métricas avaliadas sobre um quadrado também é normalizado no intervalo $[0, 10]$, obtendo $r_{\text{norm}_{\square}} = 0.0000$, $\alpha_{\text{norm}_{\square}} = 4.0234$ e $c_{\text{norm}_{\square}} = 0.0000$. Na sequência, calculam-se as variâncias $s_{M_{\text{norm}}}^2$ das métricas de qualidade normalizadas, resultando em $s_{R_{\text{norm}}}^2 = 1.2237$, $s_{A_{\text{norm}}}^2 = 2.1237$ e $s_{C_{\text{norm}}}^2 = 3.5809$.

Com as variâncias $s_{M_{\text{norm}}}^2$ calculadas, parte-se para o cálculo das tolerâncias mínima e máxima, obtendo $\epsilon_{\text{min}} = 0.0000$ e $\epsilon_{\text{max}} = 0.1879$. Iterando entre o intervalo de valores, constrói-se o diagrama de dispersão de ϵ_i por $|E_{U_i}|$, o qual é ilustrado na Figura 76 pelos pontos em preto. A curva em verde representa a função de regressão não linear exponencial, encontrada na etapa posterior. No caso do cachorro, o método de mínimos quadrados obteve, como resultado, os coeficientes $c_1 = 202.0961$ e $c_2 = -36.1107$.

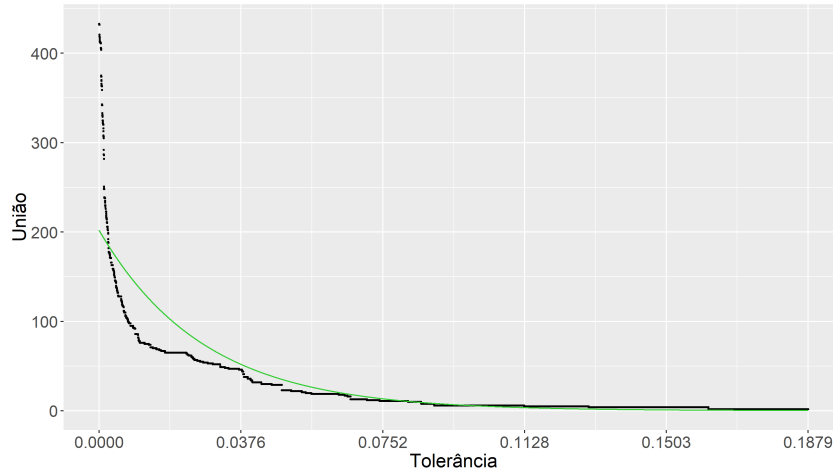


Figura 76 – Diagrama de dispersão de ϵ_i por $|E_{U_i}|$ do cachorro

Dados os coeficientes c_1 e c_2 da função de regressão não linear exponencial, calculam-se as tolerâncias desejadas $\bar{\epsilon}$, considerando $\bar{p} = 5, 10$ e 20% . As porcentagens utilizadas resultaram, respectivamente, em $\bar{\epsilon} = 0.0619, 0.0427$ e 0.0235 . Para cada valor de tolerância desejada $\bar{\epsilon}$, identificam-se os elementos de menor qualidade em relação à razão entre arestas $\overline{E_r}$, ao ângulo interno $\overline{E_\alpha}$ e ao coeficiente de compacidade $\overline{E_c}$. Além disso, por meio de operações envolvendo os conjuntos $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$, obtêm-se os conjuntos $\overline{E_U}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$ e $\overline{E_\cap}$. Finalmente, um número q é atribuído para cada categoria de elementos a partir dos conjuntos obtidos.

Após a escrita dos resultados em um arquivo *.vtk* para cada bloco, a Figura 77 ilustra os elementos de menor qualidade identificados para o cachorro, levando em consideração $\bar{p} = 5, 10$ e 20% , respectivamente, nas Figuras 77a - 77c.

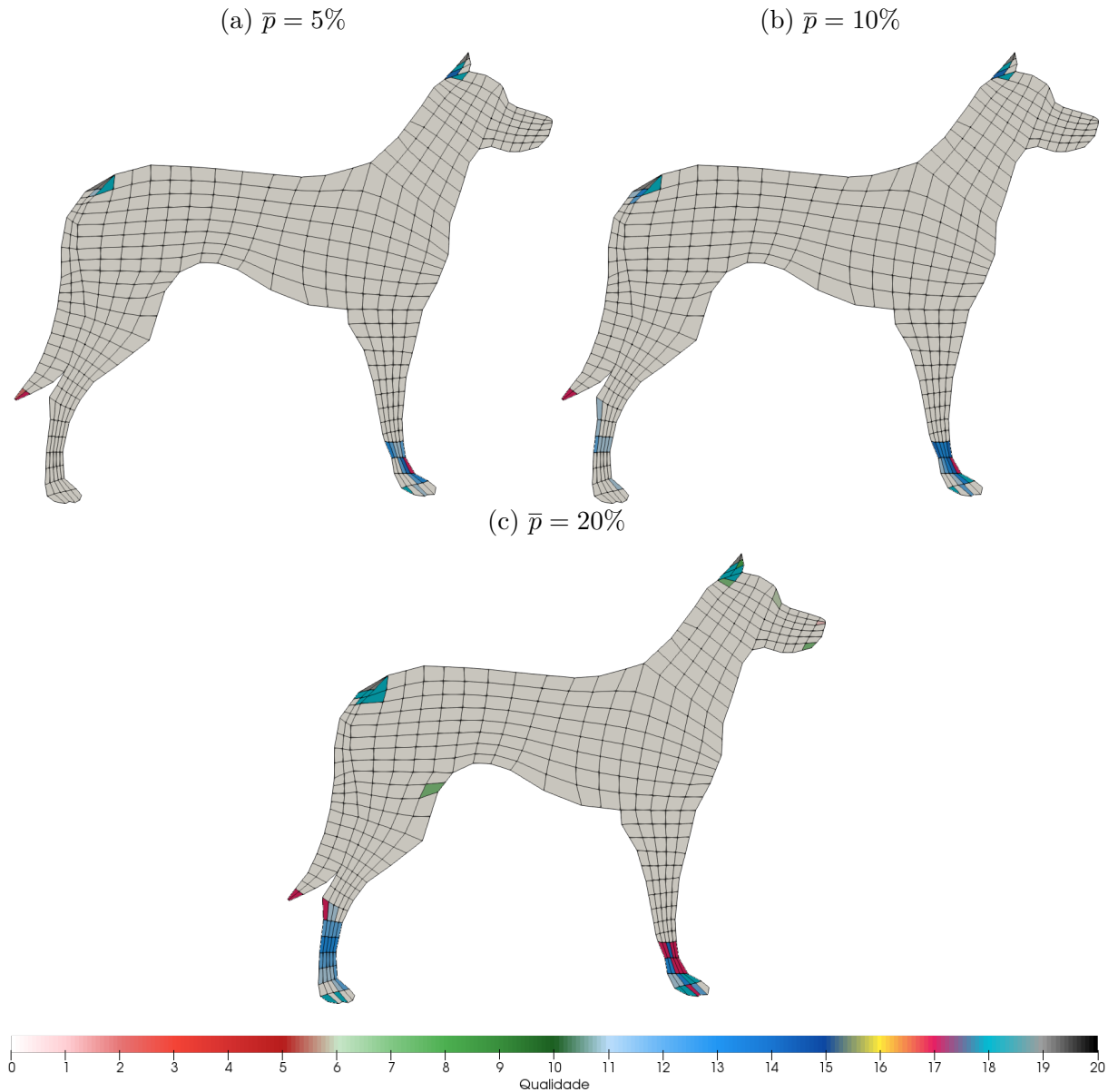


Figura 77 – Elementos de menor qualidade identificados para o cachorro

Uma outra maneira para se observar os resultados encontrados do cachorro pode ser representada na Tabela 74, na qual se apresentam as porcentagens desejadas \bar{p} e as tolerâncias desejadas $\bar{\epsilon}$ em relação às quantidades de elementos dos conjuntos $U' - \overline{E_U}$, $\overline{E_r}$, $\overline{E_\alpha}$, $\overline{E_c}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$, $U - U'$ e $\overline{E_\cap}$.

Tabela 74 – Quantidades de elementos dos resultados obtidos para o cachorro

\bar{p}	$\bar{\epsilon}$	$U' - \overline{E_U}$	$\overline{E_r}$	$\overline{E_\alpha}$	$\overline{E_c}$	$\overline{E_{r,\alpha}}$	$\overline{E_{r,c}}$	$\overline{E_{\alpha,c}}$	$ U - U' $	$\overline{E_\cap}$
5%	0.0619	414	3	4	18	0	2	4	4	0
10%	0.0427	401	3	5	32	0	3	5	4	0
20%	0.0235	370	12	20	57	0	11	15	4	0

De acordo com a Figura 77 e com a Tabela 74, é possível observar que elemen-

tos de menor qualidade em relação a todas as métricas de avaliação foram identificados para o cachorro, considerando as porcentagens utilizadas \bar{p} . Além disso, para $\bar{p} = 20\%$, destaca-se o número de elementos pertencentes aos conjuntos $\overline{E_{r,c}}$ e $\overline{E_{\alpha,c}}$. Nos blocos 3 e 6, totalizam-se 4 elementos em que $r = +\infty$ e $\alpha = \#$, em virtude dos pontos de fronteira adquiridos. Por conta da complexidade da geometria, não existe um padrão na localização dos elementos detectados. No entanto, pode-se perceber uma evidente concentração de elementos de menor qualidade nos blocos 3 e 7, para todas as porcentagens. Para $\bar{p} = 10$ e 20% , surgem elementos de menor qualidade no bloco 8 da malha computacional, relacionados principalmente ao coeficiente de compacidade.

4.19 Avião

Nesta seção, modela-se e analisa-se a malha computacional de um avião. Nesse experimento, a aquisição dos pontos de fronteira é baseada em um trabalho científico e tem, como auxílio, a ferramenta *WebPlotDigitizer* [35, 38]. Esses pontos de fronteira são ilustrados na Figura 78 e são divididos em oito blocos, cujo número de partições nas direções x e y é mostrado na Tabela 75.

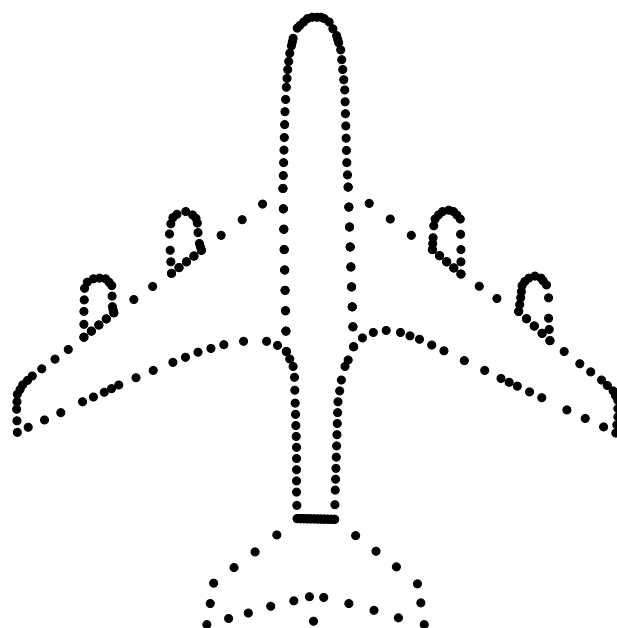


Figura 78 – Pontos de fronteira do avião. (Fonte: 35)

Tabela 75 – Número de partições nas direções x e y para cada bloco do avião

Bloco	x	y	Quantidade de nós
1	11	38	98
2	11	6	34
3	19	8	54
4	19	8	54
5	4	5	18
6	4	5	18
7	4	5	18
8	4	5	18
Total			312

Por meio do número de partições do plano físico e dos pontos de contorno, executa-se o algoritmo para geração da malha computacional em coordenadas generalizadas envolvendo multi-blocos. A malha computacional do avião é ilustrada na Figura 79, apresentando também a distribuição dos blocos.

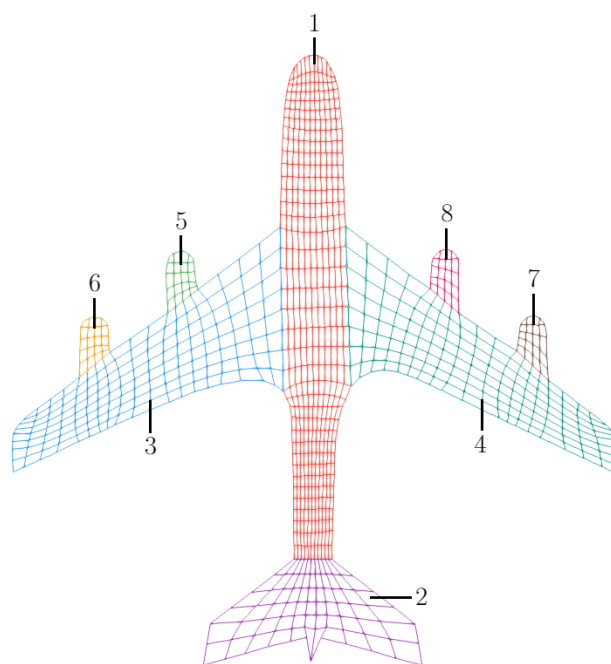


Figura 79 – Malha computacional do avião em coordenadas generalizadas

Além disso, a Tabela 76 relaciona cada bloco do avião com o número de partições nas direções ξ e η .

Tabela 76 – Número de partições nas direções ξ e η para cada bloco do avião

Bloco	ξ	η	Quantidade de nós	Quantidade de elementos
1	11	38	468	418
2	11	6	84	66
3	19	8	180	152
4	19	8	180	152
5	4	5	30	20
6	4	5	30	20
7	4	5	30	20
8	4	5	30	20
Total			1032	868

Utilizando o avião como parâmetro de entrada, executa-se o primeiro módulo para análise de qualidade, através do Algoritmo 6. Após o cálculo das métricas de qualidade para cada elemento de cada bloco da malha computacional, obtém-se o conjunto total de elementos, associados à razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, ao ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e ao coeficiente de compacidade c .

Tendo posse do conjunto total de elementos e suas métricas de qualidade, assim como informações sobre o número de partições do plano transformado para cada bloco e as porcentagens desejadas \bar{p} , executa-se o segundo módulo para análise de qualidade, por meio do Algoritmo 7. Após a leitura dos dados de entrada, obtém-se que $|U'| = |U| = 868$ elementos, pois não há elementos em que $r = +\infty$ e $\alpha = \nexists$ no conjunto U . Em seguida, a variância de cada métrica de qualidade é calculada, resultando em $s_R^2 = 0.6991$, $s_A^2 = 0.1784$ e $s_C^2 = 8.7439$ para o avião.

Uma vez que $s_R^2 \neq 0$, $s_A^2 \neq 0$ e $s_C^2 \neq 0$, calculam-se os valores mínimo m_{\min} e máximo m_{\max} das métricas de qualidade, cujos resultados estão presentes na Tabela 77.

Tabela 77 – Valores mínimo e máximo das métricas de qualidade do avião

	r	α	c
min	1.0000	0.5565	16.0000
max	7.8396	3.1396	40.6403

Utilizando os valores encontrados na Tabela 77, normalizam-se os valores das métricas de qualidade no intervalo $[0, 10]$. Ainda, o valor das métricas avaliadas sobre um quadrado também é normalizado no intervalo $[0, 10]$, obtendo $r_{\text{norm}\square} = 0.0000$, $\alpha_{\text{norm}\square} = 3.9268$ e $c_{\text{norm}\square} = 0.0000$. Na sequência, calculam-se as variâncias $s_{M_{\text{norm}}}^2$ das métricas de qualidade normalizadas, resultando em $s_{R_{\text{norm}}}^2 = 1.4944$, $s_{A_{\text{norm}}}^2 = 2.6740$ e $s_{C_{\text{norm}}}^2 = 1.4402$.

Com as variâncias $s_{M_{\text{norm}}}^2$ calculadas, parte-se para o cálculo das tolerâncias mínima e máxima, obtendo $\epsilon_{\text{min}} = 0.0005$ e $\epsilon_{\text{max}} = 0.0797$. Iterando entre o intervalo de valores, constrói-se o diagrama de dispersão de ϵ_i por $|E_{U_i}|$, o qual é ilustrado na Figura 80 pelos pontos em preto. A curva em verde representa a função de regressão não linear exponencial, encontrada na etapa posterior. No caso do avião, o método de mínimos quadrados obteve, como resultado, os coeficientes $c_1 = 419.6449$ e $c_2 = -102.5524$.

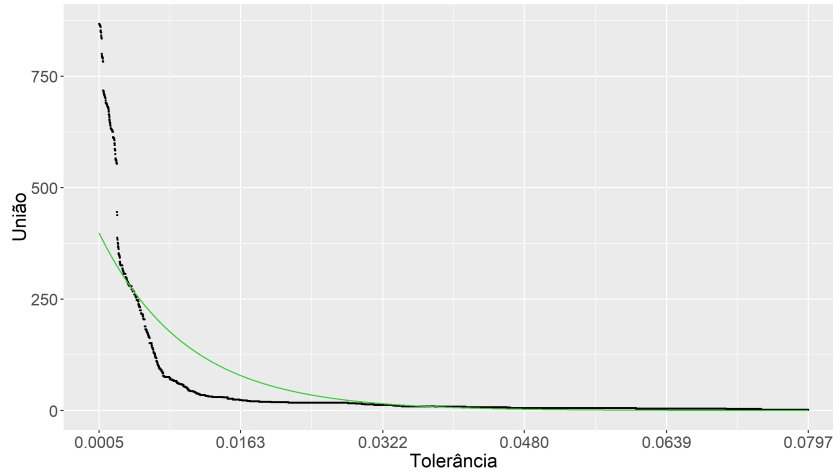


Figura 80 – Diagrama de dispersão de ϵ_i por $|E_{U_i}|$ do avião

Dados os coeficientes c_1 e c_2 da função de regressão não linear exponencial, calculam-se as tolerâncias desejadas $\bar{\epsilon}$, considerando $\bar{p} = 5, 10$ e 20% . As porcentagens utilizadas resultaram, respectivamente, em $\bar{\epsilon} = 0.0221, 0.0154$ e 0.0086 . Para cada valor de tolerância desejada $\bar{\epsilon}$, identificam-se os elementos de menor qualidade em relação à razão entre arestas $\overline{E_r}$, ao ângulo interno $\overline{E_\alpha}$ e ao coeficiente de compacidade $\overline{E_c}$. Além disso, por meio de operações envolvendo os conjuntos $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$, obtêm-se os conjuntos $\overline{E_U}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$ e $\overline{E_\cap}$. Finalmente, um número q é atribuído para cada categoria de elementos a partir dos conjuntos obtidos.

Após a escrita dos resultados em um arquivo *.vtk* para cada bloco, a Figura 81 ilustra os elementos de menor qualidade identificados para o avião, levando em consideração $\bar{p} = 5, 10$ e 20% , respectivamente, nas Figuras 81a - 81c.

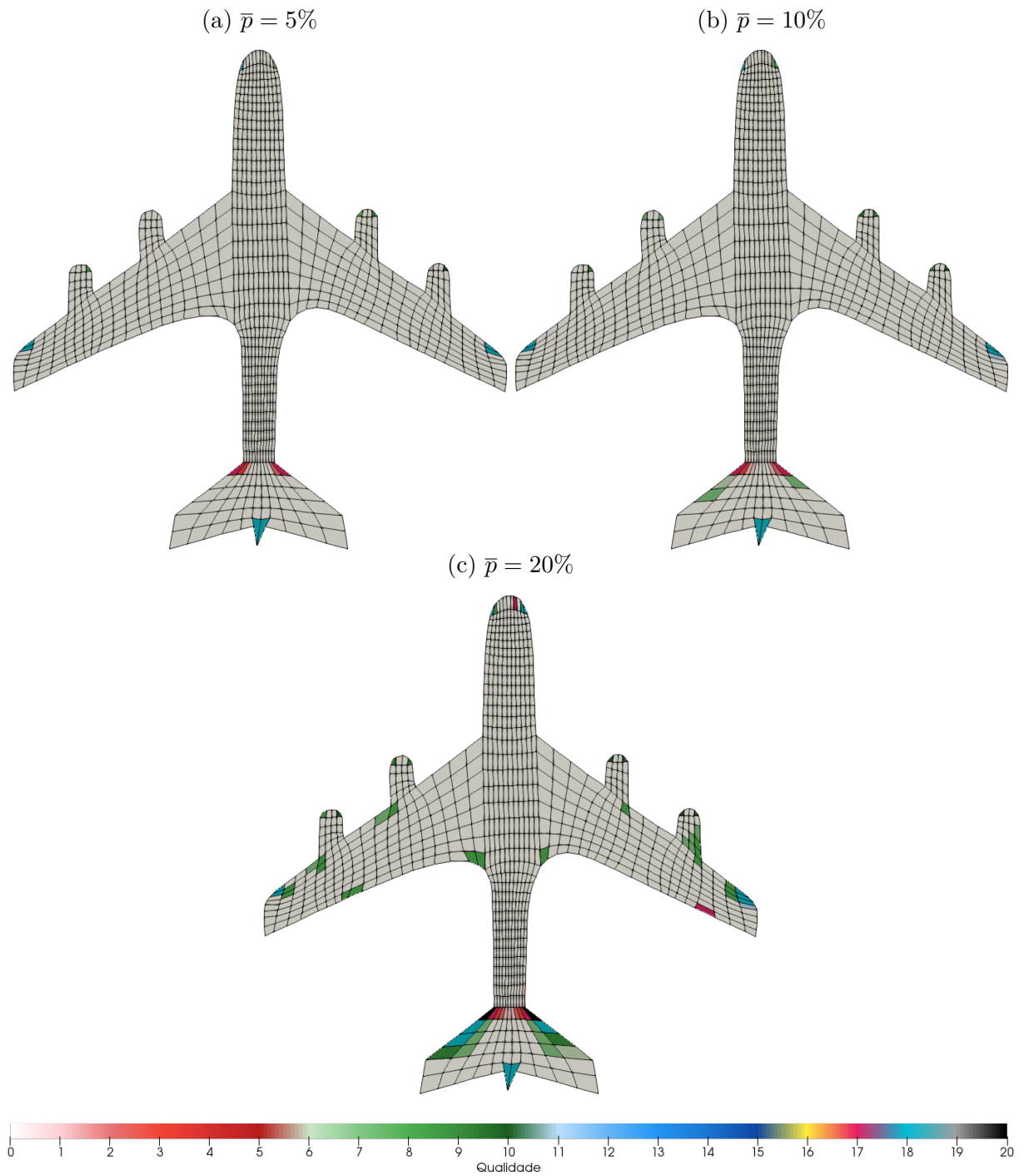


Figura 81 – Elementos de menor qualidade identificados para o avião

Uma outra maneira para se observar os resultados encontrados do avião pode ser representada na Tabela 78, na qual se apresentam as porcentagens desejadas \bar{p} e as tolerâncias desejadas $\bar{\epsilon}$ em relação às quantidades de elementos dos conjuntos $U' - \overline{E_U}$, $\overline{E_r}$, $\overline{E_\alpha}$, $\overline{E_c}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$, $U - U'$ e $\overline{E_\cap}$.

Tabela 78 – Quantidades de elementos dos resultados obtidos para o avião

\bar{p}	$\bar{\epsilon}$	$U' - \overline{E_U}$	$\overline{E_r}$	$\overline{E_\alpha}$	$\overline{E_c}$	$\overline{E_{r,\alpha}}$	$\overline{E_{r,c}}$	$\overline{E_{\alpha,c}}$	$ U - U' $	$\overline{E_\cap}$
5%	0.0221	850	5	13	9	0	2	7	0	0
10%	0.0154	841	7	19	10	0	2	7	0	0
20%	0.0086	798	15	56	20	2	6	15	0	2

De acordo com a Figura 81 e com a Tabela 78, é possível observar que elementos de menor qualidade em relação a todas as métricas de avaliação foram identificados para o avião, considerando as porcentagens utilizadas \bar{p} . Além disso, para $\bar{p} = 20\%$, destaca-se o número de elementos pertencentes ao conjunto formado pela intersecção entre $\overline{E_\alpha}$ e $\overline{E_c}$. Por conta da complexidade da geometria, não existe um padrão na localização dos elementos detectados. No entanto, pode-se perceber uma evidente concentração de elementos de menor qualidade no bloco 2, para todas as porcentagens. Para $\bar{p} = 20\%$, surgem elementos de menor qualidade nos blocos 3, 4 e 7 da malha computacional, relacionados principalmente ao ângulo interno.

4.20 Lago Luruaco

Similarmente ao estudo realizado sobre o Lago Igapó I na Seção 4.13, esta seção modela e analisa a malha computacional do Lago Luruaco, localizado na Colômbia. Nesse experimento, a aquisição dos pontos de fronteira é baseada em um trabalho científico [30]. Esses pontos de fronteira são ilustrados na Figura 82 e são divididos em treze blocos, cujo número de partições nas direções x e y é mostrado na Tabela 79.

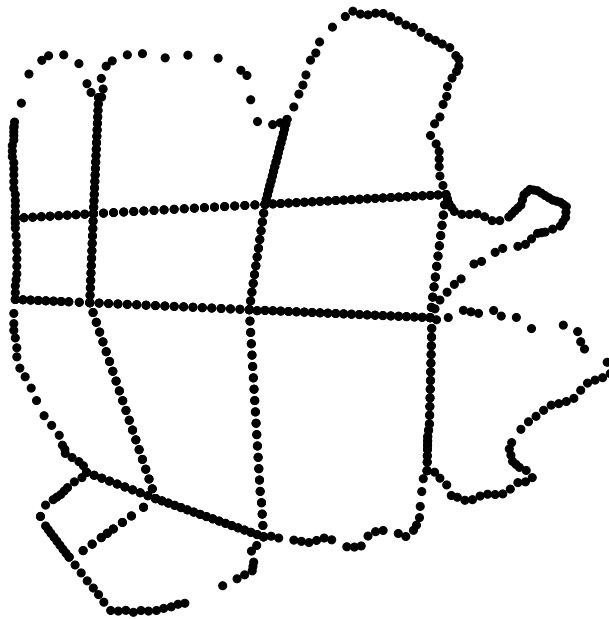


Figura 82 – Pontos de fronteira do Lago Luruaco. (Fonte: 30)

Tabela 79 – Número de partições nas direções x e y para cada bloco do Lago Luruaco

Bloco	x	y	Quantidade de nós
1	9	9	36
2	17	9	52
3	15	20	70
4	19	12	62
5	23	18	82
6	17	18	70
7	9	18	54
8	9	12	42
9	17	12	58
10	23	12	70
11	9	20	58
12	17	20	74
13	23	20	86
Total			814

Por meio do número de partições do plano físico e dos pontos de contorno, executa-se o algoritmo para geração da malha computacional em coordenadas generalizadas envolvendo multi-blocos. A malha computacional do Lago Luruaco é ilustrada na Figura 83, apresentando também a distribuição dos blocos.

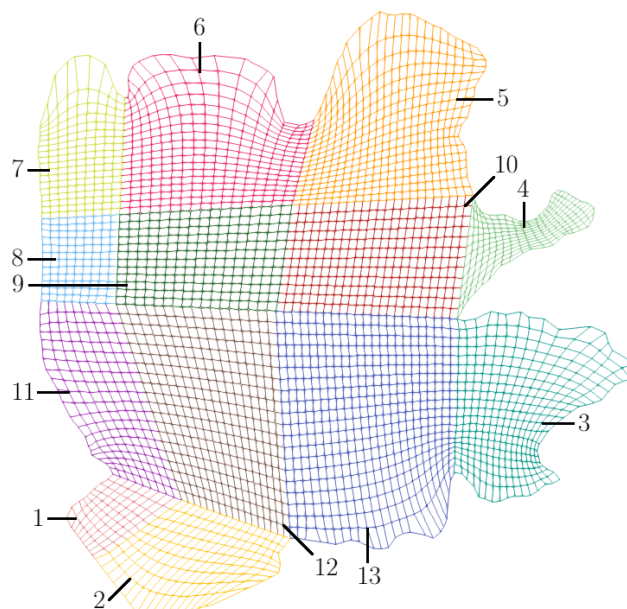


Figura 83 – Malha computacional do Lago Luruaco em coordenadas generalizadas. (Fonte: 30)

Além disso, a Tabela 80 relaciona cada bloco do Lago Luruaco com o número de partições nas direções ξ e η .

Tabela 80 – Número de partições nas direções ξ e η para cada bloco do Lago Luruaco

Bloco	ξ	η	Quantidade de nós	Quantidade de elementos
1	9	9	100	81
2	17	9	180	153
3	15	20	336	300
4	19	12	260	228
5	23	18	456	414
6	17	18	342	306
7	9	18	190	162
8	9	12	130	108
9	17	12	234	204
10	23	12	312	276
11	9	20	210	180
12	17	20	378	340
13	23	20	504	460
Total			3632	3212

Utilizando o Lago Luruaco como parâmetro de entrada, executa-se o primeiro módulo para análise de qualidade, através do Algoritmo 6. Após o cálculo das métricas de qualidade para cada elemento de cada bloco da malha computacional, obtém-se o conjunto total de elementos, associados à razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, ao ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e ao coeficiente de compacidade c .

Tendo posse do conjunto total de elementos e suas métricas de qualidade, assim como informações sobre o número de partições do plano transformado para cada bloco e as porcentagens desejadas \bar{p} , executa-se o segundo módulo para análise de qualidade, por meio do Algoritmo 7. Após a leitura dos dados de entrada, obtém-se que $|U'| = |U| = 3212$ elementos, pois não há elementos em que $r = +\infty$ e $\alpha = \nexists$ no conjunto U . Em seguida, a variância de cada métrica de qualidade é calculada, resultando em $s_R^2 = 0.2572$, $s_A^2 = 0.1227$ e $s_C^2 = 8.3238$ para o Lago Luruaco.

Uma vez que $s_R^2 \neq 0$, $s_A^2 \neq 0$ e $s_C^2 \neq 0$, calculam-se os valores mínimo m_{\min} e máximo m_{\max} das métricas de qualidade, cujos resultados estão presentes na Tabela 81.

Tabela 81 – Valores mínimo e máximo das métricas de qualidade do Lago Luruaco

	r	α	c
min	1.0000	0.2644	16.0000
max	8.0667	3.0894	59.9825

Utilizando os valores encontrados na Tabela 81, normalizam-se os valores das métricas de qualidade no intervalo $[0, 10]$. Ainda, o valor das métricas avaliadas sobre um quadrado também é normalizado no intervalo $[0, 10]$, obtendo $r_{\text{norm}_{\square}} = 0.0000$,

$\alpha_{\text{norm}\square} = 4.6244$ e $c_{\text{norm}\square} = 0.0000$. Na sequência, calculam-se as variâncias $s_{M_{\text{norm}}}^2$ das métricas de qualidade normalizadas, resultando em $s_{R_{\text{norm}}}^2 = 0.5151$, $s_{A_{\text{norm}}}^2 = 1.5378$ e $s_{C_{\text{norm}}}^2 = 0.4303$.

Com as variâncias $s_{M_{\text{norm}}}^2$ calculadas, parte-se para o cálculo das tolerâncias mínima e máxima, obtendo $\epsilon_{\text{min}} = 0.0000$ e $\epsilon_{\text{max}} = 0.0286$. Iterando entre o intervalo de valores, constrói-se o diagrama de dispersão de ϵ_i por $|E_{U_i}|$, o qual é ilustrado na Figura 84 pelos pontos em preto. A curva em verde representa a função de regressão não linear exponencial, encontrada na etapa posterior. No caso do Lago Luruaco, o método de mínimos quadrados obteve, como resultado, os coeficientes $c_1 = 468.1103$ e $c_2 = -355.4354$.

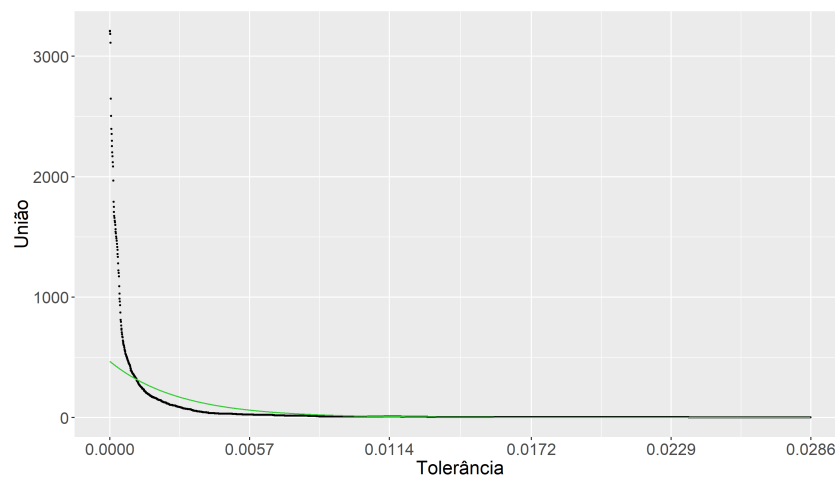


Figura 84 – Diagrama de dispersão de ϵ_i por $|E_{U_i}|$ do Lago Luruaco

Dados os coeficientes c_1 e c_2 da função de regressão não linear exponencial, calculam-se as tolerâncias desejadas $\bar{\epsilon}$, considerando $\bar{p} = 5, 10$ e 20% . As porcentagens utilizadas resultaram, respectivamente, em $\bar{\epsilon} = 0.0030, 0.0011$ e -0.0009 . Para cada valor de tolerância desejada $\bar{\epsilon}$, identificam-se os elementos de menor qualidade em relação à razão entre arestas $\overline{E_r}$, ao ângulo interno $\overline{E_\alpha}$ e ao coeficiente de compacidade $\overline{E_c}$. Além disso, por meio de operações envolvendo os conjuntos $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$, obtêm-se os conjuntos $\overline{E_U}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$ e $\overline{E_\cap}$. Finalmente, um número q é atribuído para cada categoria de elementos a partir dos conjuntos obtidos.

Após a escrita dos resultados em um arquivo *.vtk* para cada bloco, a Figura 85 ilustra os elementos de menor qualidade identificados para o Lago Luruaco, levando em consideração $\bar{p} = 5, 10$ e 20% , respectivamente, nas Figuras 85a - 85c.

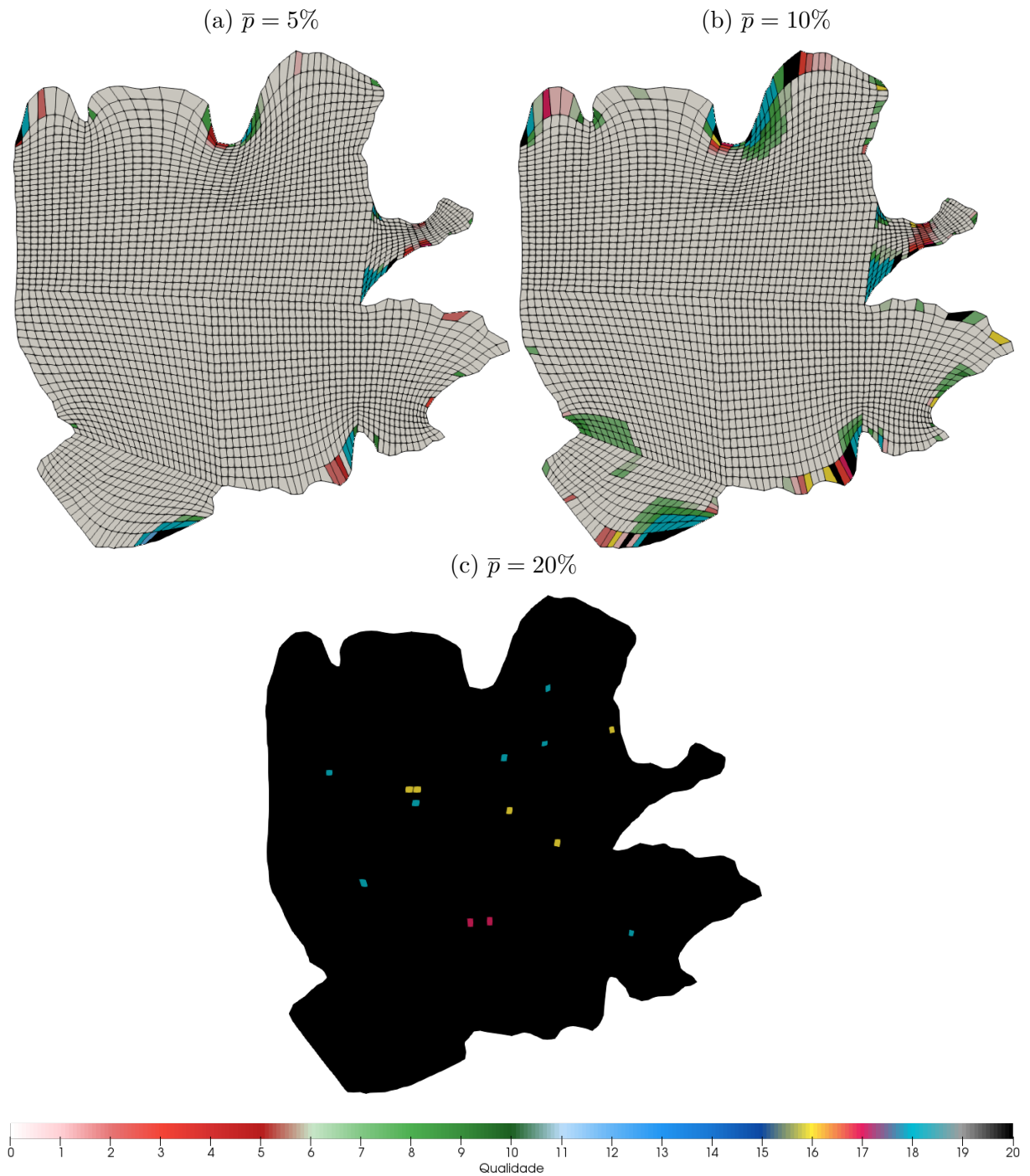


Figura 85 – Elementos de menor qualidade identificados para o Lago Luruaco

Uma outra maneira para se observar os resultados encontrados do Lago Luruaco pode ser representada na Tabela 82, na qual se apresentam as porcentagens desejadas \bar{p} e as tolerâncias desejadas $\bar{\epsilon}$ em relação às quantidades de elementos dos conjuntos $U' - \overline{E_U}$, $\overline{E_r}$, $\overline{E_\alpha}$, $\overline{E_c}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$, $U - U'$ e $\overline{E_\Gamma}$.

Tabela 82 – Quantidades de elementos dos resultados obtidos para o Lago Luruaco

\bar{p}	$\bar{\epsilon}$	$U' - \overline{E_{\cup}}$	$\overline{E_r}$	$\overline{E_{\alpha}}$	$\overline{E_c}$	$\overline{E_{r,\alpha}}$	$\overline{E_{r,c}}$	$\overline{E_{\alpha,c}}$	$ U - U' $	$\overline{E_{\cap}}$
5%	0.0030	3133	32	50	30	4	5	28	0	4
10%	0.0011	2885	99	256	77	28	24	72	0	19
20%	-0.0009	0	3205	3210	3207	3203	3200	3205	0	3198

De acordo com a Figura 85 e com a Tabela 82, é possível observar que elementos de menor qualidade em relação a todas as métricas de avaliação foram identificados para o Lago Luruaco, considerando as porcentagens utilizadas \bar{p} . Além disso, destaca-se o número de elementos pertencentes aos conjuntos formados pelas intersecções entre $\overline{E_r}$, $\overline{E_{\alpha}}$ e $\overline{E_c}$, salientando o conjunto $\overline{E_{\alpha,c}}$. Para $\bar{p} = 10\%$, percebem-se 256 elementos de menor qualidade em relação ao ângulo interno, quantidade em evidência quando comparada com as demais métricas de avaliação. Para $\bar{p} = 20\%$, todos os elementos da malha computacional foram classificados como sendo de menor qualidade. A razão para tal resultado reside no fato de que a tolerância desejada $\bar{\epsilon} = -0.0009 < 0$, assim o conjunto $\overline{E_{\cup}}$ é constituído pela máxima quantidade de elementos possível. Sobre as demais porcentagens, por conta da complexidade da geometria, não existe um padrão na localização dos elementos detectados. No entanto, pode-se verificar uma evidente concentração de elementos de menor qualidade nos blocos 2, 4 - 6 e 13 da malha computacional.

4.21 Corpo Humano

Nesta seção, modela-se e analisa-se a malha computacional do corpo humano. Nesse experimento, os pontos de fronteira foram coletados utilizando imagens encontradas no *Google* e com o auxílio da ferramenta *WebPlotDigitizer* [37, 38]. Esses pontos de fronteira são ilustrados na Figura 86 e são divididos em vinte blocos, cujo número de partições nas direções x e y é mostrado na Tabela 83.

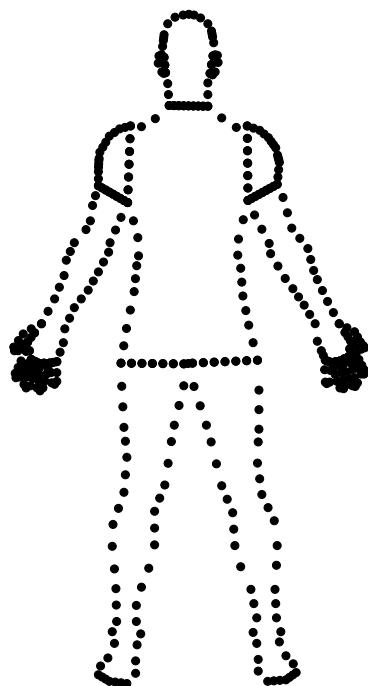


Figura 86 – Pontos de fronteira do corpo humano. (Fonte: 37)

Tabela 83 – Número de partições nas direções x e y para cada bloco do corpo humano

Bloco	x	y	Quantidade de nós
1	8	9	34
2	14	16	60
3	6	19	50
4	6	19	50
5	8	6	28
6	8	6	28
7	8	15	46
8	8	15	46
9	2	4	12
10	2	5	14
11	2	4	12
12	2	4	12
13	2	4	12
14	2	4	12
15	2	4	12
16	2	4	12
17	2	2	8
18	2	2	8
19	2	2	8
20	2	2	8
Total			472

Por meio do número de partições do plano físico e dos pontos de contorno, executa-se o algoritmo para geração da malha computacional em coordenadas generalizadas en-

volvendo multi-blocos. A malha computacional do corpo humano é ilustrada na Figura 87, apresentando também a distribuição dos blocos.

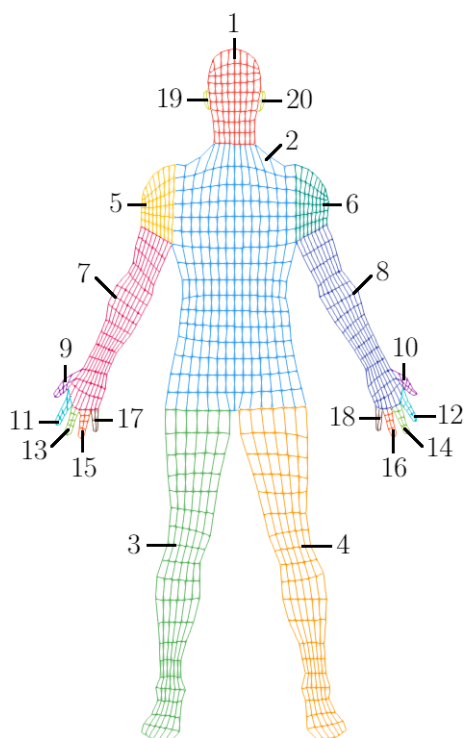


Figura 87 – Malha computacional do corpo humano em coordenadas generalizadas

Além disso, a Tabela 84 relaciona cada bloco do corpo humano com o número de partições nas direções ξ e η .

Tabela 84 – Número de partições nas direções ξ e η para cada bloco do corpo humano

Bloco	ξ	η	Quantidade de nós	Quantidade de elementos
1	8	9	90	72
2	14	16	255	224
3	6	19	140	114
4	6	19	140	114
5	8	6	63	48
6	8	6	63	48
7	8	15	144	120
8	8	15	144	120
9	2	4	15	8
10	2	5	18	10
11	2	4	15	8
12	2	4	15	8
13	2	4	15	8
14	2	4	15	8
15	2	4	15	8
16	2	4	15	8
17	2	2	9	4
18	2	2	9	4
19	2	2	9	4
20	2	2	9	4
Total			1198	942

Utilizando o corpo humano como parâmetro de entrada, executa-se o primeiro módulo para análise de qualidade, através do Algoritmo 6. Após o cálculo das métricas de qualidade para cada elemento de cada bloco da malha computacional, obtém-se o conjunto total de elementos, associados à razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, ao ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e ao coeficiente de compacidade c .

Tendo posse do conjunto total de elementos e suas métricas de qualidade, assim como informações sobre o número de partições do plano transformado para cada bloco e as porcentagens desejadas \bar{p} , executa-se o segundo módulo para análise de qualidade, por meio do Algoritmo 7. Após a leitura dos dados de entrada, obtém-se que $|U'| = |U| = 942$ elementos, pois não há elementos em que $r = +\infty$ e $\alpha = \nexists$ no conjunto U . Em seguida, a variância de cada métrica de qualidade é calculada, resultando em $s_R^2 = 1.0358$, $s_A^2 = 0.1322$ e $s_C^2 = 12.2421$ para o corpo humano.

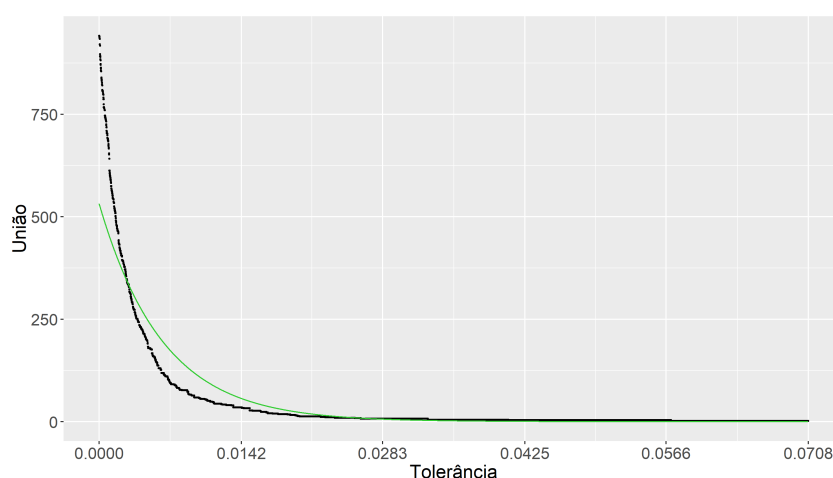
Uma vez que $s_R^2 \neq 0$, $s_A^2 \neq 0$ e $s_C^2 \neq 0$, calculam-se os valores mínimo m_{\min} e máximo m_{\max} das métricas de qualidade, cujos resultados estão presentes na Tabela 85.

Tabela 85 – Valores mínimo e máximo das métricas de qualidade do corpo humano

	r	α	c
min	1.0000	0.4590	16.0000
max	8.1404	3.1416	43.9065

Utilizando os valores encontrados na Tabela 85, normalizam-se os valores das métricas de qualidade no intervalo $[0, 10]$. Ainda, o valor das métricas avaliadas sobre um quadrado também é normalizado no intervalo $[0, 10]$, obtendo $r_{\text{norm}\square} = 0.0000$, $\alpha_{\text{norm}\square} = 4.1444$ e $c_{\text{norm}\square} = 0.0000$. Na sequência, calculam-se as variâncias $s_{M_{\text{norm}}}^2$ das métricas de qualidade normalizadas, resultando em $s_{R_{\text{norm}}}^2 = 2.0317$, $s_{A_{\text{norm}}}^2 = 1.8372$ e $s_{C_{\text{norm}}}^2 = 1.5720$.

Com as variâncias $s_{M_{\text{norm}}}^2$ calculadas, parte-se para o cálculo das tolerâncias mínima e máxima, obtendo $\epsilon_{\text{min}} = 0.0000$ e $\epsilon_{\text{max}} = 0.0708$. Iterando entre o intervalo de valores, constrói-se o diagrama de dispersão de ϵ_i por $|E_{U_i}|$, o qual é ilustrado na Figura 88 pelos pontos em preto. A curva em verde representa a função de regressão não linear exponencial, encontrada na etapa posterior. No caso do corpo humano, o método de mínimos quadrados obteve, como resultado, os coeficientes $c_1 = 532.0441$ e $c_2 = -157.5904$.

Figura 88 – Diagrama de dispersão de ϵ_i por $|E_{U_i}|$ do corpo humano

Dados os coeficientes c_1 e c_2 da função de regressão não linear exponencial, calculam-se as tolerâncias desejadas $\bar{\epsilon}$, considerando $\bar{p} = 5, 10$ e 20% . As porcentagens utilizadas resultaram, respectivamente, em $\bar{\epsilon} = 0.0154, 0.0110$ e 0.0066 . Para cada valor de tolerância desejada $\bar{\epsilon}$, identificam-se os elementos de menor qualidade em relação à razão entre arestas $\overline{E_r}$, ao ângulo interno $\overline{E_\alpha}$ e ao coeficiente de compacidade $\overline{E_c}$. Além disso, por meio de operações envolvendo os conjuntos $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$, obtêm-se os conjuntos $\overline{E_U}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$ e $\overline{E_\cap}$. Finalmente, um número q é atribuído para cada categoria de elementos a partir dos conjuntos obtidos.

Após a escrita dos resultados em um arquivo *.vtk* para cada bloco, a Figura 89 ilustra os elementos de menor qualidade identificados para o corpo humano, levando em consideração $\bar{p} = 5, 10$ e 20% , respectivamente, nas Figuras 89a - 89c.

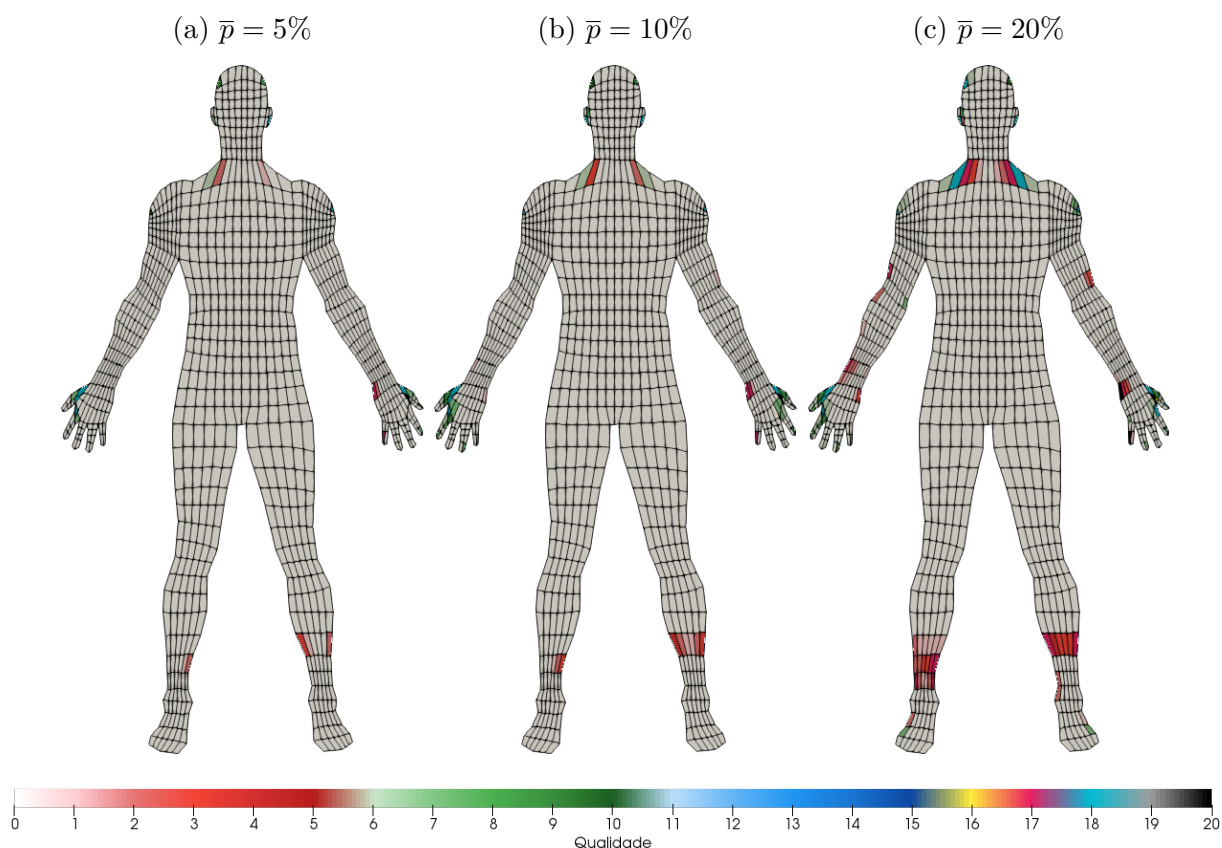


Figura 89 – Elementos de menor qualidade identificados para o corpo humano

Uma outra maneira para se observar os resultados encontrados do corpo humano pode ser representada na Tabela 86, na qual se apresentam as porcentagens desejadas \bar{p} e as tolerâncias desejadas $\bar{\epsilon}$ em relação às quantidades de elementos dos conjuntos $U' - \overline{E_U}$, $\overline{E_r}$, $\overline{E_\alpha}$, $\overline{E_c}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$, $U - U'$ e $\overline{E_\cap}$.

Tabela 86 – Quantidades de elementos dos resultados obtidos para o corpo humano

\bar{p}	$\bar{\epsilon}$	$U' - \overline{E_U}$	$\overline{E_r}$	$\overline{E_\alpha}$	$\overline{E_c}$	$\overline{E_{r,\alpha}}$	$\overline{E_{r,c}}$	$\overline{E_{\alpha,c}}$	$ U - U' $	$\overline{E_\cap}$
5%	0.0154	915	11	17	9	1	3	7	0	1
10%	0.0110	891	19	33	12	1	4	9	0	1
20%	0.0066	830	60	56	27	4	14	16	0	3

De acordo com a Figura 89 e com a Tabela 86, é possível observar que elementos de menor qualidade em relação a todas as métricas de avaliação foram identificados para o corpo humano, considerando as porcentagens utilizadas \bar{p} . Além disso, para $\bar{p} = 20\%$, destaca-se o número de elementos pertencentes aos conjuntos $\overline{E_{r,c}}$ e $\overline{E_{\alpha,c}}$. Por conta da

complexidade da geometria, não existe um padrão na localização dos elementos detectados. No entanto, pode-se perceber uma concentração de elementos de menor qualidade nos blocos 9 e 10, para todas as porcentagens. Para $\bar{p} = 10$ e 20%, surgem elementos de menor qualidade nos blocos 4, 11, 12 e 19. Finalmente, apenas para $\bar{p} = 20\%$, surgem elementos de menor qualidade nos blocos 2, 3, 5 - 8 e 18 da malha computacional.

4.22 Sapo

Nesta seção, modela-se e analisa-se a malha computacional de um sapo. Nesse experimento, a aquisição dos pontos de fronteira é baseada em um artigo científico e tem, como auxílio, a ferramenta *WebPlotDigitizer* [35, 38]. Esses pontos de fronteira são ilustrados na Figura 90 e são divididos em 25 blocos, cujo número de partições nas direções x e y é mostrado na Tabela 87.

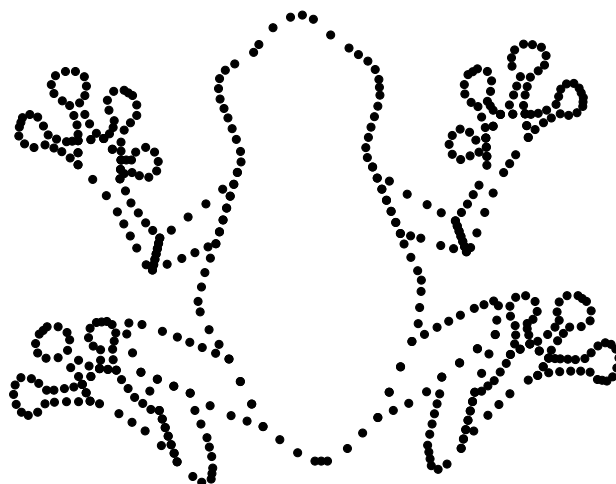


Figura 90 – Pontos de fronteira do sapo. (Fonte: 35)

Tabela 87 – Número de partições nas direções x e y para cada bloco do sapo

Bloco	x	y	Quantidade de nós
1	8	29	74
2	5	6	22
3	5	6	22
4	8	6	28
5	6	8	28
6	7	3	20
7	7	3	20
8	3	8	22
9	3	9	24
10	6	6	24
11	6	6	24
12	2	6	16
13	2	7	18
14	2	8	20
15	2	10	24
16	2	8	20
17	2	6	16
18	2	9	22
19	2	8	20
20	2	7	18
21	2	4	12
22	2	5	14
23	2	7	18
24	2	8	20
25	2	8	20
Total			566

Por meio do número de partições do plano físico e dos pontos de contorno, executa-se o algoritmo para geração da malha computacional em coordenadas generalizadas envolvendo multi-blocos. A malha computacional do sapo é ilustrada na Figura 91, apresentando também a distribuição dos blocos.

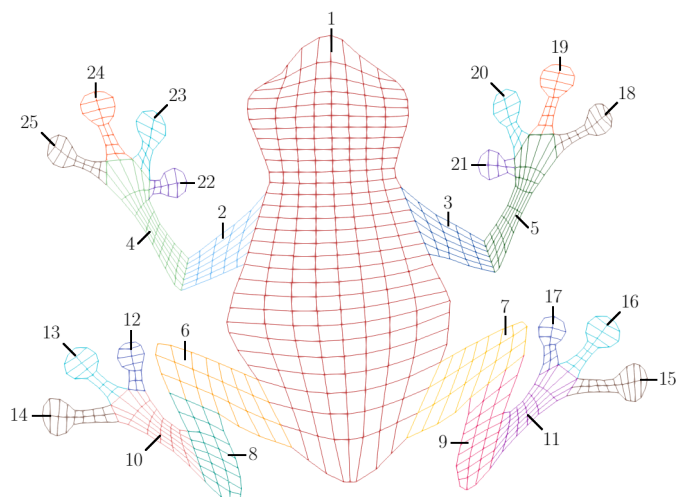


Figura 91 – Malha computacional do sapo em coordenadas generalizadas

Além disso, a Tabela 88 relaciona cada bloco do sapo com o número de partições nas direções ξ e η .

Tabela 88 – Número de partições nas direções ξ e η para cada bloco do sapo

Bloco	ξ	η	Quantidade de nós	Quantidade de elementos
1	8	29	270	232
2	5	6	42	30
3	5	6	42	30
4	8	6	63	48
5	6	8	63	48
6	7	3	32	21
7	7	3	32	21
8	3	8	36	24
9	3	9	40	27
10	6	6	49	36
11	6	6	49	36
12	2	6	21	12
13	2	7	24	14
14	2	8	27	16
15	2	10	33	20
16	2	8	27	16
17	2	6	21	12
18	2	9	30	18
19	2	8	27	16
20	2	7	24	14
21	2	4	15	8
22	2	5	18	10
23	2	7	24	14
24	2	8	27	16
25	2	8	27	16
Total			1063	755

Utilizando o sapo como parâmetro de entrada, executa-se o primeiro módulo para análise de qualidade, através do Algoritmo 6. Após o cálculo das métricas de qualidade para cada elemento de cada bloco da malha computacional, obtém-se o conjunto total de elementos, associados à razão entre arestas r com maior disparidade em relação a $r_{\square} = 1$, ao ângulo interno α com maior disparidade em relação a $\alpha_{\square} = \frac{\pi}{2}$ rad e ao coeficiente de compacidade c .

Tendo posse do conjunto total de elementos e suas métricas de qualidade, assim como informações sobre o número de partições do plano transformado para cada bloco e as porcentagens desejadas \bar{p} , executa-se o segundo módulo para análise de qualidade, por meio do Algoritmo 7. Após a leitura dos dados de entrada, obtém-se que $|U'| = |U| = 755$ elementos, pois não há elementos em que $r = +\infty$ e $\alpha = \nexists$ no conjunto U . Em seguida, a variância de cada métrica de qualidade é calculada, resultando em $s_R^2 = 1.4401$, $s_A^2 = 0.4106$ e $s_C^2 = 31.8725$ para o sapo.

Uma vez que $s_R^2 \neq 0$, $s_A^2 \neq 0$ e $s_C^2 \neq 0$, calculam-se os valores mínimo m_{\min} e máximo m_{\max} das métricas de qualidade, cujos resultados estão presentes na Tabela 89.

Tabela 89 – Valores mínimo e máximo das métricas de qualidade do sapo

	r	α	c
min	1.0000	0.2625	16.0000
max	8.2926	3.1416	50.5773

Utilizando os valores encontrados na Tabela 89, normalizam-se os valores das métricas de qualidade no intervalo $[0, 10]$. Ainda, o valor das métricas avaliadas sobre um quadrado também é normalizado no intervalo $[0, 10]$, obtendo $r_{\text{norm}\square} = 0.0000$, $\alpha_{\text{norm}\square} = 4.5441$ e $c_{\text{norm}\square} = 0.0000$. Na sequência, calculam-se as variâncias $s_{M_{\text{norm}}}^2$ das métricas de qualidade normalizadas, resultando em $s_{R_{\text{norm}}}^2 = 2.7079$, $s_{A_{\text{norm}}}^2 = 4.9536$ e $s_{C_{\text{norm}}}^2 = 2.6658$.

Com as variâncias $s_{M_{\text{norm}}}^2$ calculadas, parte-se para o cálculo das tolerâncias mínima e máxima, obtendo $\epsilon_{\min} = 0.0000$ e $\epsilon_{\max} = 0.0885$. Iterando entre o intervalo de valores, constrói-se o diagrama de dispersão de ϵ_i por $|E_{U_i}|$, o qual é ilustrado na Figura 92 pelos pontos em preto. A curva em verde representa a função de regressão não linear exponencial, encontrada na etapa posterior. No caso do sapo, o método de mínimos quadrados obteve, como resultado, os coeficientes $c_1 = 445.2167$ e $c_2 = -65.9164$.

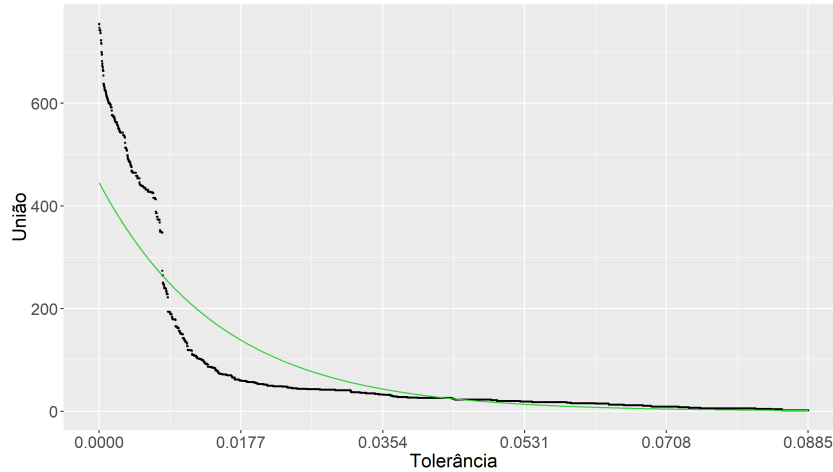


Figura 92 – Diagrama de dispersão de ϵ_i por $|E_{U_i}|$ do sapo

Dados os coeficientes c_1 e c_2 da função de regressão não linear exponencial, calculam-se as tolerâncias desejadas $\bar{\epsilon}$, considerando $\bar{p} = 5, 10$ e 20% . As porcentagens utilizadas resultaram, respectivamente, em $\bar{\epsilon} = 0.0374, 0.0269$ e 0.0164 . Para cada valor de tolerância desejada $\bar{\epsilon}$, identificam-se os elementos de menor qualidade em relação à razão entre arestas $\overline{E_r}$, ao ângulo interno $\overline{E_\alpha}$ e ao coeficiente de compacidade $\overline{E_c}$. Além disso, por meio de operações envolvendo os conjuntos $\overline{E_r}$, $\overline{E_\alpha}$ e $\overline{E_c}$, obtêm-se os conjuntos $\overline{E_U}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$ e $\overline{E_\cap}$. Finalmente, um número q é atribuído para cada categoria de elementos a partir dos conjuntos obtidos.

Após a escrita dos resultados em um arquivo *.vtk* para cada bloco, a Figura 93 ilustra os elementos de menor qualidade identificados para o sapo, levando em consideração $\bar{p} = 5, 10$ e 20% , respectivamente, nas Figuras 93a - 93c.

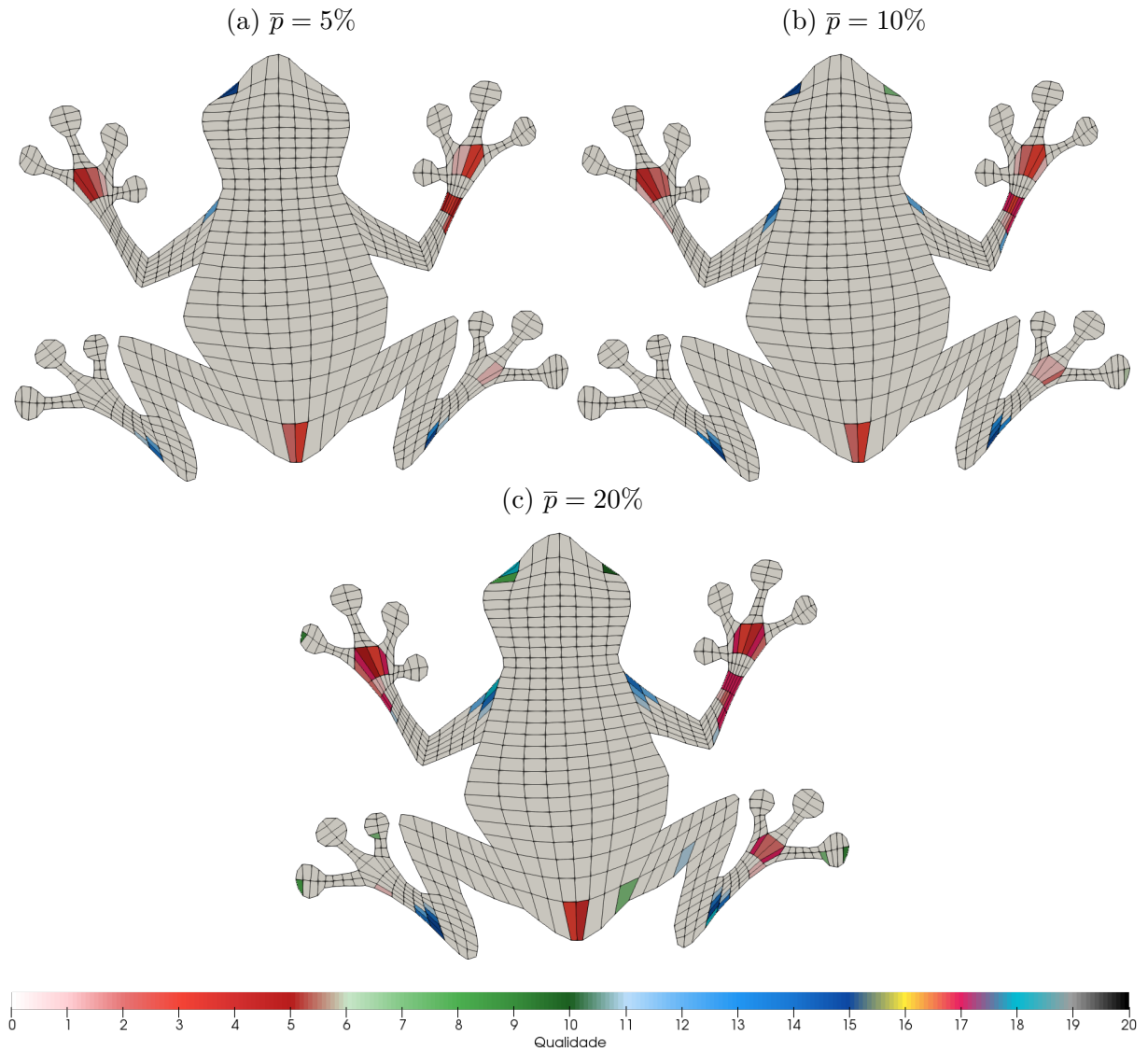


Figura 93 – Elementos de menor qualidade identificados para o sapo

Uma outra maneira para se observar os resultados encontrados do sapo pode ser representada na Tabela 90, na qual se apresentam as porcentagens desejadas \bar{p} e as tolerâncias desejadas $\bar{\epsilon}$ em relação às quantidades de elementos dos conjuntos $U' - \overline{E_{\cup}}$, $\overline{E_r}$, $\overline{E_{\alpha}}$, $\overline{E_c}$, $\overline{E_{r,\alpha}}$, $\overline{E_{r,c}}$, $\overline{E_{\alpha,c}}$, $U - U'$ e $\overline{E_{\cap}}$.

Tabela 90 – Quantidades de elementos dos resultados obtidos para o sapo

\bar{p}	$\bar{\epsilon}$	$U' - \overline{E_{\cup}}$	$\overline{E_r}$	$\overline{E_{\alpha}}$	$\overline{E_c}$	$\overline{E_{r,\alpha}}$	$\overline{E_{r,c}}$	$\overline{E_{\alpha,c}}$	$ U - U' $	$\overline{E_{\cap}}$
5%	0.0374	727	20	0	8	0	0	0	0	0
10%	0.0269	712	28	2	18	0	5	0	0	0
20%	0.0164	685	38	11	42	0	18	3	0	0

De acordo com a Figura 93 e com a Tabela 90, é possível observar que apenas elementos de menor qualidade em relação à razão entre arestas e ao coeficiente de compacidade foram identificados para o sapo, considerando $\bar{p} = 5\%$. Para as porcentagens

utilizadas $\bar{p} = 10$ e 20%, elementos de menor qualidade em relação a todas as métricas de avaliação foram identificados. Além disso, para $\bar{p} = 20\%$, destaca-se o número de elementos pertencentes ao conjunto formado pela intersecção entre $\overline{E_r}$ e $\overline{E_c}$. Por conta da complexidade da geometria, não existe um padrão na localização dos elementos detectados. No entanto, pode-se perceber uma evidente concentração de elementos de menor qualidade quanto à razão entre arestas nos blocos 4, 5 e 11, para todas as porcentagens. Para $\bar{p} = 20\%$, surgem elementos de menor qualidade nos blocos 2, 3 e 10 da malha computacional, relacionados principalmente ao coeficiente de compacidade.

A Seção 4.23 apresenta uma análise geral dos resultados obtidos para as malhas com um bloco e multi-blocos, enfatizando a importância da técnica multi-blocos no processo de geração da malha computacional.

4.23 Análise dos Resultados

De uma maneira geral, a metodologia empregada e as métricas de qualidade adotadas foram capazes de identificar os elementos de menor qualidade presentes tanto nas malhas com um bloco quanto nas malhas multi-blocos. Com relação às malhas com um bloco, nenhum elemento de menor qualidade foi identificado nas malhas quadrada e retangular. Na malha com obstáculo e na placa de orifício, elementos de menor qualidade foram identificados, principalmente, nas proximidades dos obstáculos. Na forma de gancho, houve uma concentração de elementos de menor qualidade na porção da malha com maior densidade de elementos. Finalmente, na geometria cúspide, elementos de menor qualidade foram identificados, principalmente, nas proximidades dos vértices da malha.

Quanto às malhas multi-blocos com padrão na localização dos elementos detectados, elementos de menor qualidade foram identificados, principalmente, nas proximidades do círculo imerso nas placas paralelas. Na malha hexagonal, houve uma concentração de elementos de menor qualidade nas proximidades dos vértices da malha, de forma semelhante à geometria cúspide. No experimento do NACA 64A 010, elementos de menor qualidade foram identificados nas proximidades do aerofólio. Finalmente, a malha com obstáculo multi-blocos e a placa de orifício multi-blocos serviram como meio de comparação com os experimentos envolvendo um bloco, ressaltando a importância da utilização da técnica multi-blocos no processo de geração da malha computacional.

A geração de algumas malhas multi-blocos utilizando um bloco não altera consideravelmente a quantidade de elementos de menor qualidade identificados, dadas as porcentagens desejadas \bar{p} . Contudo, a qualidade da malha, como um todo, diminui significativamente, visto que as linhas pertencentes a um bloco sofrem influência de blocos adjacentes na etapa de geração, o que torna importante a aplicação da técnica multi-blocos para tais geometrias.

Assim, a partir dos resultados encontrados, pode-se verificar que é possível aplicar o algoritmo para análise de qualidade a situações reais, permitindo identificar problemas na modelagem da geometria do meio que está sendo investigado. Apresentados e discutidos os resultados obtidos, o Capítulo 5 encerra este trabalho, apontando algumas considerações finais e abordando as contribuições esperadas e possíveis trabalhos futuros.

5 CONCLUSÕES

A modelagem de objetos com geometrias complexas para posteriores estudos envolvendo equações diferenciais tem sido uma grande dificuldade nos dias de hoje. Entretanto, os avanços científicos, em particular na Matemática e na Ciência da Computação, já permitem soluções para vários problemas. Neste trabalho, a partir da teoria de coordenadas generalizadas [5, 6, 8, 10, 11], um gerador de malhas bidimensionais foi desenvolvido, utilizando a linguagem de programação *Python* [39]. Esse gerador automatizado possibilitou a reprodução de geometrias complexas, como foi observado por meio dos experimentos executados.

Especificamente, como contribuição original deste trabalho em relação à metodologia do processo de geração de malhas computacionais, destaca-se a aplicação do método *spline* linear parametrizado para a interpolação das fronteiras [5, 11, 19], devido a sua facilidade, e o cálculo da média ponderada das fronteiras para a obtenção dos pontos internos da malha, permitindo maior rapidez na convergência da resolução numérica da equação governante. Ainda, a linguagem de programação escolhida e as bibliotecas utilizadas possibilitam, além da otimização do código, precisão dos resultados e praticidade na visualização das plotagens, a portabilidade da aplicação, sendo capaz de ser interpretada em diferentes sistemas operacionais [39, 40].

Tendo em vista que, muitas vezes, a resolução das equações governantes para a geração de malhas bidimensionais em coordenadas generalizadas pode acarretar na introdução de elementos de menor qualidade, dependendo da complexidade da geometria e do refinamento empregado, é fundamental também analisar a qualidade das malhas produzidas. Dessa maneira, o segundo foco deste trabalho referiu-se ao processo de análise de qualidade de malhas computacionais em coordenadas generalizadas, constituindo outra importante contribuição deste trabalho.

Para a realização de tal análise, três critérios de avaliação foram aplicados: as razões entre o tamanho de cada aresta com os demais, os ângulos internos e o coeficiente de compacidade do elemento. Uma vez que os elementos pertencentes às malhas geradas no sistema de coordenadas adotado são quadriláteros, verificou-se a similaridade dos elementos a um quadrado, considerado como ideal. Assim, utilizando as linguagens de programação *Python* e R [39, 49], desenvolveram-se algoritmos com as funções de calcular as métricas de qualidade e de classificar os elementos entre maior ou menor qualidade. Segundo os experimentos efetuados, obtiveram-se resultados satisfatórios de uma maneira geral, nos quais os algoritmos implementados captaram com precisão os elementos de menor qualidade.

No que diz respeito aos trabalhos futuros, o processo de geração de malhas pode ser aprimorado por meio da aplicação da técnica *spline* cúbica parametrizada para a interpolação das fronteiras da geometria em estudo. Essa técnica permite obter resultados melhores para as funções interpoladoras, apesar de o algoritmo possuir uma complexidade maior quando comparado ao método *spline* linear parametrizado. Por esse motivo, objetiva-se também a automatização dessa etapa, selecionando a abordagem mais adequada para a interpolação das fronteiras, dependendo do problema em questão.

Com relação aos possíveis trabalhos futuros para o processo de análise de qualidade de malhas, é possível elencar: o estudo e o desenvolvimento de novas métricas de qualidade; a automatização da etapa de identificação dos elementos de menor qualidade através da eliminação do parâmetro referente à porcentagem desejada \bar{p} ; e o aperfeiçoamento da função de regressão para um melhor ajuste ao diagrama de dispersão. As tarefas citadas tornarão o processo de análise mais eficiente e preciso no sentido de classificação dos elementos da malha computacional.

Além disso, pretende-se cumprir uma outra etapa essencial, a qual consiste na melhoria da qualidade de malhas em coordenadas generalizadas. Levando em consideração que, até o presente momento, os algoritmos desenvolvidos realizam a geração de malhas bidimensionais em coordenadas generalizadas e a análise de qualidade dos elementos, é necessário melhorar ou amenizar a menor qualidade das malhas produzidas. Para tal, sabe-se que, no âmbito computacional, muitos problemas do mundo real têm sido solucionados empregando técnicas de Inteligência Artificial (IA) [61]. Dessa forma, tem-se a intenção de aplicar os principais algoritmos de IA, especificamente da área de otimização, como, por exemplo, algoritmos genéticos (GA - *Genetic Algorithm*) [62, 63], otimização por colônia de formigas (ACO - *Ant Colony Optimization*) [64] e otimização por enxame de partículas (PSO - *Particle Swarm Optimization*) [65], para a melhoria dos elementos de menor qualidade.

Finalmente, outros trabalhos futuros podem ser listados, tais como a utilização de técnicas de paralelização dos algoritmos e o desenvolvimento de um modelo em três dimensões. A execução de forma paralela dos algoritmos permite a obtenção de resultados em um tempo de processamento reduzido. Ainda, um modelo 3D possibilita reproduzir uma gama ainda maior de geometrias, tornando a modelagem mais realística.

REFERÊNCIAS

- [1] FORTUNA, A. de O. *Técnicas computacionais para dinâmica dos fluidos: conceitos básicos e aplicações*. São Paulo, SP: Editora da Universidade de São Paulo, 2000.
- [2] BERN, M.; PLASSMANN, P. Mesh generation. In: *Handbook of computational geometry*. North Holland: Elsevier Science, 2000. p. 291–332.
- [3] BARBA, A. N. D. *Estudo e implementação de esquema upwind na resolução de um modelo de dinâmica dos fluidos computacional em coordenadas generalizadas*. Dissertação (Mestrado) — Universidade Estadual de Londrina (UEL), Londrina, PR, 2015.
- [4] MARQUES, A. C. *Desenvolvimento de modelo numérico utilizando o método dos volumes finitos em malhas não-estruturadas*. Dissertação (Mestrado) — Universidade Federal do Espírito Santo (UFES), Vitória, ES, 2005.
- [5] MALISKA, C. R. *Transferência de calor e mecânica dos fluidos computacional*. 2. ed. São Paulo, SP: LTC Editora, 1995.
- [6] KOOMULLIL, R.; SONI, B.; SINGH, R. A comprehensive generalized mesh system for CFD applications. *Mathematics and Computers in Simulation*, v. 78, n. 5–6, p. 605–617, 2008. ISSN 0378-4754. Applied scientific computing: numerical grid generation and field simulation. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0378475408001626>>.
- [7] THOMPSON, J. F.; SONI, B. K.; WEATHERILL, N. P. *Handbook of grid generation*. CRC Press, 1998. ISBN 9781420050349. Disponível em: <<https://books.google.com.br/books?id=ImaDT6ijKq4C>>.
- [8] THOMPSON, J. F.; THAMES, F. C.; MASTIN, C. W. TOMCAT — a code for numerical generation of boundary-fitted curvilinear coordinate systems on fields containing any number of arbitrary two-dimensional bodies. *Journal of Computational Physics*, v. 24, n. 3, p. 274–302, 1977. ISSN 0021-9991. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0021999177900389>>.
- [9] ZHANG, L. et al. A 3D hybrid grid generation technique and a multigrid/parallel algorithm based on anisotropic agglomeration approach. *Chinese Journal of Aeronautics*, v. 26, n. 1, p. 47–62, 2013. ISSN 1000-9361. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1000936112000106>>.
- [10] THOMPSON, J. F.; WARSI, Z. U. A.; MASTIN, C. W. *Numerical grid generation: foundations and applications*. Mississippi, United States: Elsevier Science Publishing Co., 1985.
- [11] CIRILO, E. R.; BORTOLI Álvaro L. Geração da malha da traquéia e dos tubos bronquiais por splines cúbico. *Semina: Ciências Exatas e Tecnológicas*, v. 27, n. 2, p. 147–155, 2006.
- [12] ROMEIRO, N. M. L. et al. Local calibration of coliforms parameters of water quality problem at Igapó I lake, Londrina, Paraná, Brazil. *Ecological Modelling*, v. 222, n. 11, p. 1888–1896, 2011. ISSN 0304-3800. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0304380011001414>>.

- [13] PARDO, S. R. et al. A transport modeling of the carbon-nitrogen cycle at Igapó I lake - Londrina, Paraná state, Brazil. *Acta Scientiarum. Technology*, 2012.
- [14] PARK, J.; SHONTZ, S. M. Two derivative-free optimization algorithms for mesh quality improvement. *Procedia Computer Science*, v. 1, n. 1, p. 387–396, May 2010. ISSN 18770509. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S1877050910000438>>.
- [15] GARIMELLA, R. V.; SHASHKOV, M. J.; KNUPP, P. M. Triangular and quadrilateral surface mesh quality optimization using local parametrization. *Computer Methods in Applied Mechanics and Engineering*, v. 193, n. 9, p. 913–928, 2004. ISSN 00457825. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0045782503005942>>.
- [16] SHEWCHUK, J. What is a good linear finite element? Interpolation, conditioning, anisotropy, and quality measures. *University of California at Berkeley*, v. 73, 2002.
- [17] BOROUCHEKI, H.; FREY, P. J. Adaptive triangular–quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, John Wiley & Sons, Ltd, v. 41, n. 5, p. 915–934, 1998. ISSN 1097-0207. Disponível em: <[http://dx.doi.org/10.1002/\(SICI\)1097-0207\(19980315\)41:5<915::AID-NME318>3.0.CO;2-Y](http://dx.doi.org/10.1002/(SICI)1097-0207(19980315)41:5<915::AID-NME318>3.0.CO;2-Y)>.
- [18] JOHNEN, A.; ERNST, D.; GEUZAINÉ, C. Sequential decision-making approach for quadrangular mesh generation. *Engineering with Computers*, v. 31, n. 4, p. 729–735, Oct 2015. ISSN 1435-5663. Disponível em: <<http://dx.doi.org/10.1007/s00366-014-0383-9>>.
- [19] RUGGIERO, M. A. G.; LOPES, V. L. R. *Cálculo numérico: aspectos teóricos e computacionais*. 2. ed. São Paulo, SP: Makron Books, 1996.
- [20] BOYCE, W. E.; DIPRIMA, R. C. *Equações diferenciais elementares e problemas de valores de contorno*. eighth. São Paulo, SP: LTC Editora, 2002.
- [21] MIERSEMANN, E. *Partial differential equations*. CreateSpace Independent Publishing Platform, 2014. ISBN 9781502910967. Disponível em: <<https://books.google.com.br/books?id=ntb2oQEACAAJ>>.
- [22] MENDELSON, E.; AYRES, F. *Teoria e problemas de cálculo*. Bookman Companhia Ed, 2007. ISBN 9788560031092. Disponível em: <<https://books.google.com.br/books?id=4bsoo853eEgC>>.
- [23] CORRÊA, L.; LIMA, G. A. B. de; FERREIRA, V. G. Solução numérica de equações diferenciais parciais via o método das diferenças finitas. In: *II Colóquio de Matemática do Centro-Oeste*. Cuiabá, MT: Editora SBM, 2011.
- [24] FERZIGER, J.; PERIĆ, M. *Computational methods for fluid dynamics*. Berlin, Germany: Springer, 2002. ISBN 978-3-540-42074-3. Disponível em: <<http://opac.inria.fr/record=b1135325>>.
- [25] BORTOLI, Á. L. *Introdução à dinâmica de fluidos computational*. Porto Alegre, RS: Editora da UFRGS, 2000. ISBN 9788570255457. Disponível em: <<https://books.google.com.br/books?id=dUrJAAAACAAJ>>.
- [26] BURDEN, R. L.; FAIRES, J. D. *Numerical analysis*. Boston, United States: Cengage Learning, 2010. ISBN 9780538733519. Disponível em: <<https://books.google.com.br/books?id=zXnSxY9G2JgC>>.

- [27] ALSABTI, K. An efficient k-means clustering algorithm. In: *In Proceedings of IPPS/SPDP Workshop on High Performance Data Mining*. Orlando, Florida: [s.n.], 1998.
- [28] KANUNGO, T. et al. An efficient k-means clustering algorithm: analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24, n. 7, p. 881–892, Jul 2002. ISSN 0162-8828.
- [29] ARTHUR, D.; VASSILVITSKII, S. K-means++: the advantages of careful seeding. In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2007. (SODA '07), p. 1027–1035. ISBN 978-0-898716-24-5. Disponível em: <<http://dl.acm.org/citation.cfm?id=1283383.1283494>>.
- [30] TAMURA, C. H. *Resolução das equações adimensionais de geração de malhas 2D com modelagem de parâmetros de qualidade*. Dissertação (Mestrado) — Universidade Estadual de Londrina (UEL), Londrina, PR, 2017.
- [31] KOKO, J. A MATLAB mesh generator for the two-dimensional finite element method. *Applied Mathematics and Computation*, v. 250, p. 650 – 664, 2015. ISSN 0096-3003. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0096300314015148>>.
- [32] PERSSON, P.; STRANG, G. A simple mesh generator in MATLAB. *SIAM Review*, v. 46, p. 2004, 2004.
- [33] MACHADO, L. G. P. *Malhas adaptativas em domínios definidos por fronteiras curvas*. Dissertação (Mestrado) — Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, São Carlos - SP, aug 2007.
- [34] NGUYEN, H. et al. Constrained CVT meshes and a comparison of triangular mesh generators. *Computational Geometry*, v. 42, n. 1, p. 1 – 19, 2009. ISSN 0925-7721. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0925772108000394>>.
- [35] VERMA, C. S.; SURESH, K. A robust combinatorial approach to reduce singularities in quadrilateral meshes. *Procedia Engineering*, v. 124, n. Supplement C, p. 252 – 264, 2015. ISSN 1877-7058. 24th International Meshing Roundtable. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1877705815032385>>.
- [36] GOOGLE. *Google Earth*. 2017. Disponível em: <<https://www.google.com/earth/>>.
- [37] GOOGLE. *Google Images*. 2017. Disponível em: <<https://images.google.com/>>.
- [38] ROHATGI, A. *WebPlotDigitizer Version 4.0*. 2017. Disponível em: <<https://automeris.io/WebPlotDigitizer>>.
- [39] ROSSUM, G. *Python reference manual*. Amsterdam, The Netherlands, 1995.
- [40] CAI, X.; LANGTANGEN, H. P.; MOE, H. On the performance of the Python programming language for serial and parallel scientific computations. *Scientific Programming*, IOS Press, Amsterdam, The Netherlands, v. 13, n. 1, p. 31–56, jan. 2005. ISSN 1058-9244. Disponível em: <<http://dx.doi.org/10.1155/2005/619804>>.
- [41] WALT, S. van der; COLBERT, S. C.; VAROQUAUX, G. The NumPy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, v. 13, n. 2, p. 22–30, 2011. Disponível em: <<http://aip.scitation.org/doi/abs/10.1109/MCSE.2011.37>>.

- [42] JONES, E. et al. *SciPy: open source scientific tools for Python*. 2001. [Online; accessed <today>]. Disponível em: <<http://www.scipy.org/>>.
- [43] HUNTER, J. D. Matplotlib: a 2D graphics environment. *Computing in Science & Engineering*, v. 9, n. 3, p. 90–95, 2007. Disponível em: <<http://aip.scitation.org/doi/abs/10.1109/MCSE.2007.55>>.
- [44] OLIVEIRA, J. d. A. et al. *Extração de atributos de forma para classificação de imagens de alta resolução do satélite*. Natal, RN, 2013.
- [45] GOSE, E.; JOHNSONBAUGH, R.; JOST, S. *Pattern recognition and image analysis*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996. ISBN 0-13-236415-8.
- [46] GONZALEZ, R.; WOODS, R. *Processamento digital de imagens*. Addison Wesley Pearson Brasil, 2011. ISBN 9788576054016. Disponível em: <<https://books.google.com.br/books?id=r5f0RgAACAAJ>>.
- [47] DEZA, M. M.; DEZA, E. *Encyclopedia of distances*. Heidelberg, Germany: Springer Berlin Heidelberg, 2009.
- [48] COELHO, F. U.; LOURENCO, M. L. *Curso de álgebra linear*. 1. ed. São Paulo, SP: Editora da Universidade de São Paulo, 2001. v. 34. ISBN 9788531405945. Disponível em: <<https://books.google.com.br/books?id=aiLoWF-9AaoC>>.
- [49] R Core Team. *R: a language and environment for statistical computing*. Vienna, Austria, 2016. Disponível em: <<https://www.R-project.org/>>.
- [50] KOMSTA, L. *outliers: tests for outliers*. Vienna, Austria, 2011. R package version 0.14. Disponível em: <<https://CRAN.R-project.org/package=outliers>>.
- [51] WICKHAM, H. *ggplot2: elegant graphics for data analysis*. Springer-Verlag New York, 2009. ISBN 978-0-387-98140-6. Disponível em: <<http://ggplot2.org>>.
- [52] KENDALL, M. G. A new measure of rank correlation. *Biometrika*, [Oxford University Press, Biometrika Trust], v. 30, n. 1/2, p. 81–93, 1938. ISSN 00063444. Disponível em: <<http://www.jstor.org/stable/2332226>>.
- [53] MAGALHÃES, M. N.; LIMA, A. C. P. de. *Noções de probabilidade e estatística*. São Paulo, SP: IME-USP, 2000.
- [54] DEVORE, J. L. *Probability and statistics for engineering and the sciences*. 8th. ed. Pacific Grove, United States: Brooks/Cole, 2011. ISBN-13: 978-0-538-73352-6.
- [55] HYAMS, D. *CurveExpert Professional: documentation*. 2011.
- [56] CASELLA, G.; BERGER, R. *Statistical inference*. United States: Duxbury Resource Center, 2001. Textbook Binding. ISBN 0534243126.
- [57] AHRENS, J.; GEVECI, B.; LAW, C. Paraview: an end-user tool for large data visualization. *The Visualization Handbook*, p. 717–731, 2005.
- [58] AYACHIT, U. *The Paraview guide: a parallel visualization application*. United States: Kitware, Inc., 2015. ISBN 1930934300, 9781930934306.

- [59] HOHENWARTER, M. *GeoGebra: ein softwaresystem für dynamische geometrie und algebra der ebene*. Dissertação (Mestrado) — Paris Lodron University, Salzburg, Austria, fev. 2002. (In German.).
- [60] MORAN, J. *An introduction to theoretical and computational aerodynamics*. Dover Publications, 1984. (Dover Books on Aeronautical Engineering). ISBN 9780486428796. Disponível em: <<https://books.google.com.br/books?id=4eVP3yWZ1LgC>>.
- [61] RUSSELL, S. J.; NORVIG, P. *Artificial intelligence: a modern approach*. 2. ed. Upper Saddle River, United States: Pearson Education, 2003. ISBN 0137903952.
- [62] MITCHELL, M. *An introduction to genetic algorithms*. Cambridge, United States: Bradford Books, 1999. (A Bradford book). ISBN 9780262631853. Disponível em: <<https://books.google.com.br/books?id=0eznlz0TF-IC>>.
- [63] MAN, K.-F.; TANG, K.-S.; KWONG, S. Genetic algorithms: concepts and applications. *IEEE transactions on Industrial Electronics*, IEEE, v. 43, n. 5, p. 519–534, 1996.
- [64] DORIGO, M. et al. *Ant colony optimization and swarm intelligence*. Brussels, Belgium: Springer, 2008. v. 5217.
- [65] KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: IEEE. *IEEE International Conference on Neural Networks*. Perth, Australia, 1995. v. 4, p. 1942–1948.

Apêndices

APÊNDICE A – CÓDIGOS COMPUTACIONAIS

A.1 Gerador de Malhas em Coordenadas Generalizadas

```

1 class Gerador:
2
3     def __init__(self, filename, times):
4
5         def linear_spline(gama, i_straight):
6             den = 1.0 / (t[i_straight] - t[i_straight-1])
7             spline_x = (x[i_straight-1] * den * (t[i_straight] - gama)) + (x[i_straight] * den * (
8                 gama - t[i_straight-1]))
9             spline_y = (y[i_straight-1] * den * (t[i_straight] - gama)) + (y[i_straight] * den * (
10                 gama - t[i_straight-1]))
11             return (spline_x, spline_y)
12
13         def solver(X, Y):
14             for it in range(1, 1000):
15                 bigger_diff1 = 0.0
16                 bigger_diff2 = bigger_diff1
17                 for i in range(1, csi):
18                     for j in range(1, eta):
19                         diff_x_csi = (X[i+1][j] - X[i-1][j]) * 0.5
20                         diff_x_eta = (X[i][j+1] - X[i][j-1]) * 0.5
21                         diff_y_csi = (Y[i+1][j] - Y[i-1][j]) * 0.5
22                         diff_y_eta = (Y[i][j+1] - Y[i][j-1]) * 0.5
23
24                         a = diff_y_eta ** 2.0 + diff_x_eta ** 2.0
25                         b = diff_y_csi * diff_y_eta + diff_x_csi * diff_x_eta
26                         c = diff_y_csi ** 2.0 + diff_x_csi ** 2.0
27
28                         jacob = 1.0 / (diff_x_csi * diff_y_eta - diff_x_eta * diff_y_csi)
29
30                         source_p = 0.0
31                         source_q = 0.0
32
33                         ap = 2.0 * a + 2.0 * c
34                         ae = a + source_p / (2.0 * (jacob ** 2.0))
35                         aw = a - source_p / (2.0 * (jacob ** 2.0))
36                         an = c + source_q / (2.0 * (jacob ** 2.0))
37                         ast = c - source_q / (2.0 * (jacob ** 2.0))
38                         ane = -b / 2.0
39                         anw = b / 2.0
40                         ase = anw
41                         asw = ane
42
43                         x_new = (1.0 / ap) * (ae * X[i+1][j] + aw * X[i-1][j] + an * X[i][j+1] + ast * X
44                             [i][j-1] \
45                             + ane * X[i+1][j+1] + anw * X[i-1][j+1] + ase * X[i+1][j-1] + asw * X[i-1][j
46                                 -1])
47
48                         y_new = (1.0 / ap) * (ae * Y[i+1][j] + aw * Y[i-1][j] + an * Y[i][j+1] + ast * Y
49                             [i][j-1] \

```

```

45         + ane * Y[i+1][j+1] + anw * Y[i-1][j+1] + ase * Y[i+1][j-1] + asw * Y[i-1][j
46             -1])
47
48         max1 = abs(x_new - X[i][j])
49         max2 = abs(y_new - Y[i][j])
50
51         if (max1 > bigger_diff1) :
52             bigger_diff1 = max1
53
54         if (max2 > bigger_diff2) :
55             bigger_diff2 = max2
56
57         X[i][j] = x_new
58         Y[i][j] = y_new
59
60         if (bigger_diff1 <= 0.00001 and bigger_diff2 <= 0.00001) :
61             break
62
63         if (it == 1000) :
64             print('Solution not found')
65             sys.exit()
66
67         file = open(filename, 'r')
68
69         _aux1 = file.readline()
70         _aux1 = _aux1.replace('M', '').replace('=', '').replace(' ', '').replace('#', '')
71         M = int(_aux1)
72
73         _aux1 = file.readline()
74         _aux1 = _aux1.replace('N', '').replace('=', '').replace(' ', '').replace('#', '')
75         N = int(_aux1)
76
77         _aux1 = file.readline()
78         if times == 0:
79             _aux1 = _aux1.replace('ksi', '').replace('=', '').replace(' ', '').replace('#', '')
80             csi = int(_aux1)
81         else:
82             csi = M * times
83
84         _aux1 = file.readline()
85         if times == 0:
86             _aux1 = _aux1.replace('etha', '').replace('=', '').replace(' ', '').replace('#', '')
87             eta = int(_aux1)
88         else:
89             eta = N * times
90
91         lr_weight_X = 0.5
92         ul_weight_X = 0.5
93         lr_weight_Y = 0.5
94         ul_weight_Y = 0.5
95
96         while 1:
97             _aux1 = file.readline()
98             if _aux1 == '\n':
99                 continue
100            if _aux1.find('x y') != -1:
                break

```

```

101     if _aux1.find('# entradasaidax') != -1:
102         _aux1 = _aux1.replace('# entradasaidax', '').replace('=', '').replace(' ', '')
103         lr_weight_X = float(_aux1)
104         ul_weight_X = 1.0 - lr_weight_X
105     if _aux1.find('# entradasaiday') != -1:
106         _aux1 = _aux1.replace('# entradasaiday', '').replace('=', '').replace(' ', '')
107         lr_weight_Y = float(_aux1)
108         ul_weight_Y = 1.0 - lr_weight_Y
109     if _aux1.find('# superiorinferiorx') != -1:
110         _aux1 = _aux1.replace('# superiorinferiorx', '').replace('=', '').replace(' ', '')
111         ul_weight_X = float(_aux1)
112         lr_weight_X = 1.0 - ul_weight_X
113     if _aux1.find('# superiorinferiory') != -1:
114         _aux1 = _aux1.replace('# superiorinferiory', '').replace('=', '').replace(' ', '')
115         ul_weight_Y = float(_aux1)
116         lr_weight_Y = 1.0 - ul_weight_Y
117
118     x = []
119     y = []
120
121     while 1:
122         _aux1 = file.readline()
123         if _aux1 == '':
124             break
125         if _aux1 == '\n':
126             continue
127
128         _aux2 = _aux1.split(' ')
129         x += [float(_aux2[0])]
130         y += [float(_aux2[1])]
131
132     x += [x[0]]
133     y += [y[0]]
134
135     file.close()
136
137     start = time.time()
138
139     t = []
140     lenXY = len(x)
141     for i in range(lenXY) :
142         t += [float(i)]
143
144     self.X = zeros((csi+1, eta+1))
145     self.Y = zeros_like(self.X)
146
147     delta_t_ul = float(M) / float(csi)
148     delta_t_lr = float(N) / float(eta)
149
150     k = 0.0
151     j = 0
152
153     for i in range(csi+1):
154         self.X[i][0], self.Y[i][0] = linear_spline(k, j+1)
155         k += delta_t_ul
156         if(k >= j+1):
157             j += 1

```

```

158
159     for i in range(1, eta+1):
160         self.X[csi][i], self.Y[csi][i] = linear_spline(k, j+1)
161         k += delta_t_lr
162         if(k >= j+1):
163             j += 1
164
165     for i in range(csi-1, -1, -1):
166         self.X[i][eta], self.Y[i][eta] = linear_spline(k, j+1)
167         k += delta_t_ul
168         if(k >= j+1):
169             j += 1
170
171     for i in range(eta-1, 0, -1):
172         self.X[0][i], self.Y[0][i] = linear_spline(k, j+1)
173         k += delta_t_lr
174         if(k >= j+1):
175             j += 1
176
177     delta_x_csi = []
178     delta_y_csi = []
179     delta_x_eta = []
180     delta_y_eta = []
181
182     for j in range(1, eta):
183         delta_x_csi += [(self.X[csi][j] - self.X[0][j]) / csi]
184         delta_y_csi += [(self.Y[csi][j] - self.Y[0][j]) / csi]
185     for i in range(1, csi):
186         delta_x_eta += [(self.X[i][eta] - self.X[i][0]) / eta]
187         delta_y_eta += [(self.Y[i][eta] - self.Y[i][0]) / eta]
188     for i in range(1, csi):
189         for j in range(1, eta):
190             self.X[i][j] = ul_weight_X * (self.X[i][0] + j * delta_x_eta[i-1]) + lr_weight_X * (
191                 self.X[0][j] + i * delta_x_csi[j-1])
192             self.Y[i][j] = ul_weight_Y * (self.Y[i][0] + j * delta_y_eta[i-1]) + lr_weight_Y * (
193                 self.Y[0][j] + i * delta_y_csi[j-1])
194
195     solver(self.X, self.Y)
196
197     print('Runtime: ' + str(time.time() - start) + ' seconds')

```

Algoritmo A.1 – Gerador de malhas bidimensionais em coordenadas generalizadas

```

1 matplotlib.rc('xtick', labelsizesize = 20)
2 matplotlib.rc('ytick', labelsizesize = 20)
3
4 if len(sys.argv) < 3:
5     print('Pass at least 2 arguments: INPUT FILENAME(S) - OUTPUT FILENAME')
6     sys.exit()
7
8 fig, ax = plt.subplots()
9 y_formatter = matplotlib.ticker.ScalarFormatter(useOffset = False)
10 ax.yaxis.set_major_formatter(y_formatter)
11
12 out = open(sys.argv[len(sys.argv)-1], 'w')
13 out.write('NODE\tX\tY\n')
14

```

```

15 i_label = 0
16 for k in range(1, len(sys.argv)-1):
17     g = Gerador(sys.argv[k], 0)
18
19     x_plot1 = asarray(g.X)
20     x_plot2 = asarray(g.X.transpose())
21     y_plot1 = asarray(g.Y)
22     y_plot2 = asarray(g.Y.transpose())
23
24     ax.plot(x_plot1, y_plot1, x_plot2, y_plot2, 'k.-', color = 'black')
25
26     labels = [i for i in range(i_label, i_label + x_plot1.size)]
27     for l, i, j in zip(labels, nditer(x_plot1), nditer(y_plot1)):
28         out.write(str(l) + '\t' + str(i) + '\t' + str(j) + '\n')
29     i_label += x_plot1.size
30
31 out.close()
32
33 ax.grid(False)
34 plt.xticks(rotation = 270)
35 plt.show()

```

Algoritmo A.2 – Gerador de malhas bidimensionais em coordenadas generalizadas envolvendo multi-blocos

A.2 Análise de Qualidade de Malhas em Coordenadas Generalizadas

A.2.1 Cálculo das Métricas de Qualidade

```

1 class Metric:
2
3     def __init__(self, x, y):
4
5         def distance(p1, p2):
6             return math.sqrt((p1[0] - p2[0]) ** 2.0 + (p1[1] - p2[1]) ** 2.0)
7
8         def vector(p1, p2):
9             return (p2[0] - p1[0], p2[1] - p1[1])
10
11        def scalar_product(u, v):
12            return u[0] * v[0] + u[1] * v[1]
13
14        def angle(p1, p2, p3, d1, d2):
15            u = vector(p1, p2)
16            v = vector(p1, p3)
17            return math.acos(scalar_product(u,v) / (d1 * d2))
18
19        csi = len(x) - 1
20        eta = len(x[0]) - 1
21
22        self.ratios = zeros((csi, eta))
23        self.angles = zeros((csi, eta))
24

```

```

25 self.perimeters = zeros((csi, eta))
26 self.areas = zeros((csi, eta))
27 self.compacities = zeros((csi, eta))
28
29 r_ideal = 1.0
30 a_ideal = math.pi * 0.5
31 c_ideal = 16.0
32
33 for i in range(csi):
34     for j in range(eta):
35         d = [distance((x[i][j], y[i][j]), (x[i][j+1], y[i][j+1])), \
36                 distance((x[i][j+1], y[i][j+1]), (x[i+1][j+1], y[i+1][j+1])), \
37                 distance((x[i+1][j], y[i+1][j]), (x[i+1][j+1], y[i+1][j+1])), \
38                 distance((x[i][j], y[i][j]), (x[i+1][j], y[i+1][j])))]
39
40         if (d[0] == 0.0 or d[1] == 0.0 or d[2] == 0.0 or d[3] == 0.0):
41             r_value = inf
42             a_value = inf
43         else:
44             r = [d[0] / d[1], d[0] / d[2], d[0] / d[3], \
45                 d[1] / d[0], d[1] / d[2], d[1] / d[3], \
46                 d[2] / d[0], d[2] / d[1], d[2] / d[3], \
47                 d[3] / d[0], d[3] / d[1], d[3] / d[2]]
48
49             r_value = r_ideal
50             r_bigger_diff = 0.0
51             for k in range(len(r)):
52                 diff = abs(r_ideal - r[k])
53                 if (diff > r_bigger_diff):
54                     r_value = r[k]
55                     r_bigger_diff = diff
56
57             a = [angle((x[i][j], y[i][j]), (x[i][j+1], y[i][j+1]), (x[i+1][j], y[i+1][j]), d
58                   [0], d[3]), \
59                   angle((x[i][j+1], y[i][j+1]), (x[i+1][j+1], y[i+1][j+1]), (x[i][j], y[i][j]), d
60                   [0], d[1]), \
61                   angle((x[i+1][j+1], y[i+1][j+1]), (x[i+1][j], y[i+1][j]), (x[i][j+1], y[i][j+1])
62                   , d[1], d[2]), \
63                   angle((x[i+1][j], y[i+1][j]), (x[i][j], y[i][j]), (x[i+1][j+1], y[i+1][j+1]), d
64                   [2], d[3])]
65
66             a_value = a_ideal
67             a_bigger_diff = 0.0
68             for k in range(len(a)):
69                 diff = abs(a_ideal - a[k])
70                 if (diff > a_bigger_diff):
71                     a_value = a[k]
72                     a_bigger_diff = diff
73
74             self.ratios[i][j] = r_value
75             self.angles[i][j] = a_value
76
77             diff_x_csi = ((x[i+1][j] + x[i+1][j+1]) * 0.5) - ((x[i][j] + x[i][j+1]) * 0.5)
78             diff_x_eta = ((x[i][j+1] + x[i+1][j+1]) * 0.5) - ((x[i][j] + x[i+1][j]) * 0.5)
79             diff_y_csi = ((y[i+1][j] + y[i+1][j+1]) * 0.5) - ((y[i][j] + y[i][j+1]) * 0.5)
80             diff_y_eta = ((y[i][j+1] + y[i+1][j+1]) * 0.5) - ((y[i][j] + y[i+1][j]) * 0.5)

```

```

78 self.areas[i][j] = abs(diff_x_csi * diff_y_eta - diff_x_eta * diff_y_csi)
79 self.perimeters[i][j] = sum(d)
80 self.compacities[i][j] = pow(self.perimeters[i][j], 2.0) / self.areas[i][j]

```

Algoritmo A.3 – Cálculo das métricas de qualidade de malhas em coordenadas generalizadas - Parte 1

```

1  if len(sys.argv) < 2:
2      print('Pass at least 1 argument: INPUT FILENAME(S)')
3      sys.exit()
4
5  out = open("metrics.csv", 'w')
6  out.write("block, x1, y1, x2, y2, x3, y3, x4, y4, ratio, angle, compacity\n")
7
8  for k in range(1, len(sys.argv)):
9      g = Gerador(sys.argv[k], 0)
10     m = Metric(g.X, g.Y)
11
12     csi = len(g.X) - 1
13     eta = len(g.X[0]) - 1
14
15     for j in range(eta):
16         for i in range(csi):
17             out.write(str(k) + ",")
18             out.write(str(g.X[i][j]) + "," + str(g.Y[i][j]) + ",")
19             out.write(str(g.X[i][j+1]) + "," + str(g.Y[i][j+1]) + ",")
20             out.write(str(g.X[i+1][j+1]) + "," + str(g.Y[i+1][j+1]) + ",")
21             out.write(str(g.X[i+1][j]) + "," + str(g.Y[i+1][j]) + ",")
22             out.write(str(m.ratios[i][j]) + "," + str(m.angles[i][j]) + "," + str(m.compacities[i][j]
23                 ]) + "\n")
24 out.close()

```

Algoritmo A.4 – Cálculo das métricas de qualidade de malhas em coordenadas generalizadas - Parte 2

A.2.2 Identificação dos Elementos de Menor Qualidade

```

1  normalize <- function(metrics, metric.lab, min.max.metrics) {
2      return((norm["b"] - norm["a"]) * (metrics - min.max.metrics["min",metric.lab]) / (min.max.
3          metrics["max",metric.lab] - min.max.metrics["min",metric.lab]) + norm["a"])
4  }
5
6  calculate.distance <- function(p1, p2) {
7      return(sqrt(sum((p1 - p2) ^ 2.0)))
8  }
9
10 write.analysis.vtk <- function(filenamees, metric.title, mesh.results) {
11     for (block in blocks) {
12         file <- file(filenamees[block])
13
14         lines <- c(
15             "# vtk DataFile Version 2.0",
16             metric.title,
17             "ASCII",

```

```

17     "DATASET STRUCTURED_GRID",
18     paste("DIMENSIONS", csi[block] + 1, eta[block] + 1, 1),
19     paste("POINTS", (csi[block] + 1) * (eta[block] + 1), "FLOAT")
20 )
21
22 block.results <- mesh.results[which(mesh.results$block == block),]
23 for (j in 1:eta[block]) {
24     for (i in 1:csi[block]) {
25         lines <- c(lines,
26                 paste(
27                     format(block.results[(j - 1) * csi[block] + i, "x1"], nsmall = 10),
28                     format(block.results[(j - 1) * csi[block] + i, "y1"], nsmall = 10),
29                     format(0, nsmall = 10)
30                 ))
31     }
32     lines <- c(lines,
33             paste(
34                 format(block.results[j * csi[block], "x4"], nsmall = 10),
35                 format(block.results[j * csi[block], "y4"], nsmall = 10),
36                 format(0, nsmall = 10)
37             ))
38 }
39
40 for (i in 1:csi[block]) {
41     lines <- c(lines,
42             paste(
43                 format(block.results[(eta[block] - 1) * csi[block] + i, "x2"], nsmall = 10),
44                 format(block.results[(eta[block] - 1) * csi[block] + i, "y2"], nsmall = 10),
45                 format(0, nsmall = 10)
46             ))
47 }
48 lines <- c(lines,
49         paste(
50             format(block.results[csi[block] * eta[block], "x3"], nsmall = 10),
51             format(block.results[csi[block] * eta[block], "y3"], nsmall = 10),
52             format(0, nsmall = 10)
53         ),
54     paste("CELL_DATA", csi[block] * eta[block]),
55     paste("SCALARS", metric.title, "FLOAT"),
56     "LOOKUP_TABLE default"
57 )
58
59 for (j in 1:eta[block]) {
60     for (i in 1:csi[block]) {
61         lines <- c(lines,
62                 toString(format(block.results[(j - 1) * csi[block] + i, "scalar"], nsmall = 10)
63                 ))
64     }
65 }
66
67 writeLines(lines, file)
68
69 close(file)
70 }
71
72 find.outliers.rc <- function(metrics, epsilon, previous.var, metric.lab) {

```

```

73 outlier <- outlier(metrics[, metric.lab])
74 outliers.pos <- which(metrics[, metric.lab] == outlier)
75
76 current.var <- var(metrics[-outliers.pos, metric.lab])
77 if (abs(current.var - previous.var) <= epsilon)
78   return(NULL)
79
80 while (current.var > min.precision &&
81        abs(current.var - previous.var) > epsilon &&
82        length(outliers.pos) < (nrow(metrics) - 1)) {
83   outlier <- outlier(metrics[-outliers.pos, metric.lab])
84   outliers.pos <- c(outliers.pos, which(metrics[, metric.lab] == outlier))
85
86   previous.var <- current.var
87   current.var <- var(metrics[-outliers.pos, metric.lab])
88 }
89
90 return(rownames(metrics[unique(outliers.pos),]))
91 }
92
93 find.outliers.angle <- function(metrics, epsilon, previous.var, norm.square.metrics) {
94   outlier <- outlier(metrics$angle)
95   outlier.op <- outlier(metrics$angle, opposite = TRUE)
96   diff.outlier <- abs(outlier - norm.square.metrics["angle.b"])
97   diff.outlier.op <- abs(outlier.op - norm.square.metrics["angle.b"])
98   outliers.pos <- ifelse(diff.outlier >= diff.outlier.op,
99                          which(metrics$angle == outlier),
100                         which(metrics$angle == outlier.op)
101 )
102
103 current.var <- var(metrics[-outliers.pos, "angle"])
104 if (abs(current.var - previous.var) <= epsilon)
105   return(NULL)
106
107 while (current.var > min.precision &&
108        abs(current.var - previous.var) > epsilon &&
109        length(outliers.pos) < (nrow(metrics) - 2)) {
110   outlier <- outlier(metrics[-outliers.pos, "angle"])
111   outlier.op <- outlier(metrics[-outliers.pos, "angle"], opposite = TRUE)
112   diff.outlier <- abs(outlier - norm.square.metrics["angle.b"])
113   diff.outlier.op <- abs(outlier.op - norm.square.metrics["angle.b"])
114   outliers.pos <- c(outliers.pos, ifelse(diff.outlier >= diff.outlier.op,
115                                         which(metrics$angle == outlier),
116                                         which(metrics$angle == outlier.op)
117   ))
118
119   previous.var <- current.var
120   current.var <- var(metrics[-outliers.pos, "angle"])
121 }
122
123 return(rownames(metrics[unique(outliers.pos),]))
124 }
125
126 find.min.epsilon <- function(metrics, variances, norm.square.metrics) {
127   epsilon <- min.epsilon
128

```

```

129 outliers.ratio <- find.outliers.rc(metrics, epsilon + inc.epsilon, variances["ratio"], "ratio"
130 )
131 outliers.angle <- find.outliers.angle(metrics, epsilon + inc.epsilon, variances["angle"], norm
132 .square.metrics)
133 outliers.compacity <- find.outliers.rc(metrics, epsilon + inc.epsilon, variances["compacity"],
134 "compacity")
135 first.outliers.union <- union(union(outliers.ratio, outliers.angle), outliers.compacity)
136 outliers.union <- first.outliers.union
137 while (length(first.outliers.union) == length(outliers.union) &&
138 epsilon < max.epsilon) {
139 epsilon <- epsilon + inc.epsilon
140
141 outliers.ratio <- find.outliers.rc(metrics, epsilon + inc.epsilon, variances["ratio"], "
142 ratio")
143 outliers.angle <- find.outliers.angle(metrics, epsilon + inc.epsilon, variances["angle"],
144 norm.square.metrics)
145 outliers.compacity <- find.outliers.rc(metrics, epsilon + inc.epsilon, variances["compacity"
146 ], "compacity")
147 outliers.union <- union(union(outliers.ratio, outliers.angle), outliers.compacity)
148 }
149
150 return(epsilon - inc.epsilon)
151 }
152
153 find.max.epsilon <- function(metrics, variances, norm.square.metrics) {
154 epsilon <- max.epsilon
155
156 outliers.ratio <- find.outliers.rc(metrics, epsilon - inc.epsilon, variances["ratio"], "ratio"
157 )
158 outliers.angle <- find.outliers.angle(metrics, epsilon - inc.epsilon, variances["angle"], norm
159 .square.metrics)
160 outliers.compacity <- find.outliers.rc(metrics, epsilon - inc.epsilon, variances["compacity"],
161 "compacity")
162
163 first.outliers.union <- union(union(outliers.ratio, outliers.angle), outliers.compacity)
164 outliers.union <- first.outliers.union
165 while (length(first.outliers.union) == length(outliers.union) &&
166 epsilon > min.epsilon) {
167 epsilon <- epsilon - inc.epsilon
168
169 outliers.ratio <- find.outliers.rc(metrics, epsilon - inc.epsilon, variances["ratio"], "
170 ratio")
171 outliers.angle <- find.outliers.angle(metrics, epsilon - inc.epsilon, variances["angle"],
172 norm.square.metrics)
173 outliers.compacity <- find.outliers.rc(metrics, epsilon - inc.epsilon, variances["compacity"
174 ], "compacity")
175 outliers.union <- union(union(outliers.ratio, outliers.angle), outliers.compacity)
176 }
177
178 return(epsilon + inc.epsilon)
179 }
180
181 find.outliers.quantities <- function(metrics, variances, mesh.min.epsilon, mesh.max.epsilon,
182 norm.square.metrics) {
183 epsilon <- seq(mesh.min.epsilon, mesh.max.epsilon, inc.epsilon)
184 union <- vector("numeric", length(epsilon))

```

```

173 for (i in 1:length(epsilon)) {
174   outliers.ratio <- find.outliers.rc(metrics, epsilon[i], variances["ratio"], "ratio")
175   outliers.angle <- find.outliers.angle(metrics, epsilon[i], variances["angle"], norm.square.
      metrics)
176   outliers.compacity <- find.outliers.rc(metrics, epsilon[i], variances["compacity"], "
      compacity")
177   outliers.union <- union(union(outliers.ratio, outliers.angle), outliers.compacity)
178
179   union[i] <- length(outliers.union)
180 }
181 quantities <- data.frame(epsilon, union)
182 quantities <- quantities[which(quantities$union != 0),]
183 return(quantities)
184 }
185
186 least.squares <- function(quantities) {
187   linear.quantities <- data.frame(epsilon2 = quantities$epsilon, linear.union = log(quantities$
      union))
188   linear.quantities <- aggregate(epsilon2 ~ linear.union, linear.quantities, median)
189   linear.quantities <- cbind(linear.quantities, epsilon1 = rep(1.0, nrow(linear.quantities)))
190
191   matrix.a <- c(
192     linear.quantities$epsilon1 %% linear.quantities$epsilon1,
193     linear.quantities$epsilon1 %% linear.quantities$epsilon2,
194     linear.quantities$epsilon2 %% linear.quantities$epsilon1,
195     linear.quantities$epsilon2 %% linear.quantities$epsilon2
196   )
197   vector.b <- c(
198     linear.quantities$epsilon1 %% linear.quantities$linear.union,
199     linear.quantities$epsilon2 %% linear.quantities$linear.union
200   )
201
202   beta <- (matrix.a[1] * vector.b[2] - matrix.a[3] * vector.b[1]) /
203     (matrix.a[1] * matrix.a[4] - matrix.a[3] * matrix.a[2])
204   alpha <- (vector.b[1] - matrix.a[2] * beta) / matrix.a[1]
205
206   regression <- c("a" = exp(alpha), "b" = beta)
207   return(regression)
208 }
209
210 plot.outliers.quantities <- function(quantities, regression, mesh.min.epsilon, mesh.max.epsilon)
  {
211   gg.union <- ggplot(data = quantities, aes(x = epsilon)) +
212     geom_point(aes(y = union), colour = "black", size = 0.5) +
213     geom_line(aes(y = regression["a"] * exp(regression["b"] * epsilon)), colour = "limegreen") +
214     xlab("Tolerancia") + ylab("Uniao") +
215     theme(axis.text = element_text(size = 14), axis.title = element_text(size = 16)) +
216     scale_x_continuous(breaks = round(seq(mesh.min.epsilon, mesh.max.epsilon, (mesh.max.epsilon
      - mesh.min.epsilon) * 0.2), 4))
217   ggsave(paste0(in.folder, unions.file), gg.union, width = 10.02, height = 5.64)
218 }
219
220 clustering <- function(outliers, centers, quality, folder = NULL, metric.lab = NULL) {
221   set.seed(1)
222   kmeans.result <- kmeans(outliers, centers, iter.max = 100, algorithm = c("Lloyd"))
223   ordered.centers <- order(kmeans.result$centers)
224

```

```

225   if (!is.null(folder)) {
226     kmeans.centers.df <- data.frame("index" = 1:center, "centers" = kmeans.result$centers[
        ordered.centers])
227     write.csv(kmeans.centers.df, file = paste0(folder, metric.lab, ".csv"), row.names = FALSE)
228   }
229
230   adj.clusters <- rep(0, length(kmeans.result$cluster))
231   for (i in 1:center)
232     adj.clusters[which(kmeans.result$cluster == ordered.centers[i])] <- i + (quality - 1) *
        kmeans.centers
233   return(adj.clusters)
234 }
235
236 analyze.mesh <- function(all.metrics, in.folder, out.folder, percentage, kmeans.folder = NULL) {
237   metrics <- all.metrics[which(is.finite(all.metrics$ratio)),]
238
239   variances <- c("ratio" = var(metrics$ratio), "angle" = var(metrics$angle), "compacity" = var(
        metrics$compacity))
240   if (variances["ratio"] < min.precision && variances["angle"] < min.precision && variances["
        compacity"] < min.precision) {
241     if (metrics[1, "ratio"] == square.metrics["ratio"] &&
242         abs(metrics[1, "angle"] - square.metrics["angle"]) < min.precision &&
243         metrics[1, "compacity"] == square.metrics["compacity"])
244       print("Todos os elementos da malha sao identicos e iguais a um quadrado.")
245     else
246       print("Todos os elementos da malha sao identicos, mas diferentes de um quadrado.")
247
248     all.metrics$scalar <- 0
249     write.analysis.vtk(paste0(in.folder, mesh.files), "QUALITY", all.metrics)
250     return()
251   }
252
253   min.max.metrics <- matrix(c(min(c(metrics$ratio, square.metrics["ratio"])),
254                               max(c(metrics$ratio, square.metrics["ratio"])),
255                               min(c(metrics$angle, square.metrics["angle"])),
256                               max(c(metrics$angle, square.metrics["angle"])),
257                               min(c(metrics$compacity, square.metrics["compacity"])),
258                               max(c(metrics$compacity, square.metrics["compacity"]))),
259                             nrow = 2, ncol = 3,
260                             dimnames = list(c("min", "max"), names(square.metrics)))
261   metrics[,names(square.metrics)] <- sapply(names(square.metrics), function(metric.lab) {
262     return(normalize(metrics[,metric.lab], metric.lab, min.max.metrics))
263   })
264   norm.square.metrics <- c("ratio" = normalize(square.metrics["ratio"], "ratio", min.max.metrics
        ),
265                            "angle" = normalize(square.metrics["angle"], "angle", min.max.metrics),
266                            "compacity" = normalize(square.metrics["compacity"], "compacity", min.
        max.metrics))
267
268   variances <- c("ratio" = var(metrics$ratio), "angle" = var(metrics$angle), "compacity" = var(
        metrics$compacity))
269   mesh.min.epsilon <- find.min.epsilon(metrics, variances, norm.square.metrics)
270   mesh.max.epsilon <- find.max.epsilon(metrics, variances, norm.square.metrics)
271   if (mesh.min.epsilon > mesh.max.epsilon) {
272     epsilon <- rep(mean(c(mesh.min.epsilon, mesh.max.epsilon)), length(percentage))
273   } else {

```

```

274 quantities <- find.outliers.quantities(metrics, variances, mesh.min.epsilon, mesh.max.
      epsilon, norm.square.metrics)
275 regression <- least.squares(quantities)
276 plot.outliers.quantities(quantities, regression, mesh.min.epsilon, mesh.max.epsilon)
277
278 epsilon <- (log(percentage * nrow(metrics)) - log(regression["a"])) / regression["b"]
279 }
280
281 all.metrics <- lapply(1:length(percentage), function(i) {
282   outliers.ratio <- find.outliers.rc(metrics, epsilon[i], variances["ratio"], "ratio")
283   outliers.angle <- find.outliers.angle(metrics, epsilon[i], variances["angle"], norm.square.
      metrics)
284   outliers.compacity <- find.outliers.rc(metrics, epsilon[i], variances["compacity"], "
      compacity")
285
286   outliers.union <- union(union(outliers.ratio, outliers.angle), outliers.compacity)
287   outliers.intersect.ra <- intersect(outliers.ratio, outliers.angle)
288   outliers.intersect.rc <- intersect(outliers.ratio, outliers.compacity)
289   outliers.intersect.ac <- intersect(outliers.angle, outliers.compacity)
290   outliers.intersect.all <- intersect(outliers.intersect.ra, outliers.compacity)
291
292   all.metrics[setdiff(rownames(metrics), outliers.union), "scalar"] <- quality["good"]
293
294   kmeans.folder <- paste0(out.folder[i], "k-means", "/")
295   dir.create(kmeans.folder, showWarnings = FALSE, recursive = TRUE)
296
297   if (length(outliers.ratio) != 0) {
298     unique.ratio <- length(unique(all.metrics[outliers.ratio, "ratio"]))
299     min.centers <- min(unique.ratio, kmeans.centers)
300     all.metrics[outliers.ratio, "scalar"] <- clustering(all.metrics[outliers.ratio, "ratio"],
      min.centers, quality["ratio"], kmeans.folder, "ratio")
301   }
302
303   if (length(outliers.angle) != 0) {
304     unique.angle <- length(unique(all.metrics[outliers.angle, "angle"]))
305     min.centers <- min(unique.angle, kmeans.centers)
306     all.metrics[outliers.angle, "scalar"] <- clustering(all.metrics[outliers.angle, "angle"],
      min.centers, quality["angle"], kmeans.folder, "angle")
307   }
308
309   if (length(outliers.compacity) != 0) {
310     unique.compacity <- length(unique(all.metrics[outliers.compacity, "compacity"]))
311     min.centers <- min(unique.compacity, kmeans.centers)
312     all.metrics[outliers.compacity, "scalar"] <- clustering(all.metrics[outliers.compacity, "
      compacity"], min.centers, quality["compacity"], kmeans.folder, "compacity")
313   }
314
315   all.metrics[outliers.intersect.ra, "scalar"] <- quality["ra"]
316   all.metrics[outliers.intersect.rc, "scalar"] <- quality["rc"]
317   all.metrics[outliers.intersect.ac, "scalar"] <- quality["ac"]
318   all.metrics[outliers.intersect.all, "scalar"] <- quality["all"]
319   all.metrics[which(is.infinite(all.metrics$ratio)), "scalar"] <- quality["inf"]
320
321   write.analysis.vtk(paste0(out.folder[i], mesh.files), "QUALITY", all.metrics)
322   save.image(file = paste0(out.folder[i], rdata.file))
323   write.csv(all.metrics, file = paste0(out.folder[i], metrics.file), row.names = FALSE)
324

```

```
325     return(all.metrics)
326   })
327 }
328
329 mesh <- "Semi-trapezoidal"
330 csi <- c(5)
331 eta <- c(3)
332
333 # mesh <- "Quadrado"
334 # csi <- c(10)
335 # eta <- c(10)
336
337 # mesh <- "Retangulo"
338 # csi <- c(20)
339 # eta <- c(20)
340
341 # mesh <- "Obstaculo"
342 # csi <- c(80)
343 # eta <- c(16)
344
345 # mesh <- "Placa_de_Orificio"
346 # csi <- c(80)
347 # eta <- c(16)
348
349 # mesh <- "Forma_de_Gancho"
350 # csi <- c(36)
351 # eta <- c(9)
352
353 # mesh <- "Cuspide"
354 # csi <- c(18)
355 # eta <- c(18)
356
357 # mesh <- "Placas_Paralelas_com_Circulo_Imerso"
358 # csi <- c(72, 72)
359 # eta <- c(18, 18)
360
361 # mesh <- "Hexagono"
362 # csi <- c(36, 36)
363 # eta <- c(9, 9)
364
365 # mesh <- "Naca"
366 # csi <- c(32, 32)
367 # eta <- c(9, 9)
368
369 # mesh <- "Obstaculo_Multi-blocos"
370 # csi <- c(16, 12, 16)
371 # eta <- c(16, 12, 16)
372
373 # mesh <- "Placa_de_Orificio_Multi-blocos"
374 # csi <- c(16, 8, 16)
375 # eta <- c(16, 8, 16)
376
377 # mesh <- "Lago_Igapo_I"
378 # csi <- c(6, 12, 16)
379 # eta <- c(70, 2, 5)
380
381 # mesh <- "Golfinho"
```

```

382 # csi <- c(36, 4, 5, 5, 6, 7)
383 # eta <- c(9, 10, 10, 8, 7, 9)
384
385 # mesh <- "Tartaruga"
386 # csi <- c(10, 10, 4, 4, 5, 5)
387 # eta <- c(7, 31, 17, 17, 5, 8)
388
389 # mesh <- "Perfil_de_um_Rosto"
390 # csi <- c(3, 15, 8, 3, 8, 15, 5)
391 # eta <- c(4, 6, 8, 5, 6, 14, 7)
392
393 # mesh <- "Passaro"
394 # csi <- c(6, 11, 20, 20, 11, 18, 8)
395 # eta <- c(11, 11, 4, 7, 18, 11, 11)
396
397 # mesh <- "Cachorro"
398 # csi <- c(5, 6, 2, 5, 17, 2, 5, 5)
399 # eta <- c(6, 6, 3, 5, 8, 22, 14, 18)
400
401 # mesh <- "Aviao"
402 # csi <- c(11, 11, 19, 19, 4, 4, 4, 4)
403 # eta <- c(38, 6, 8, 8, 5, 5, 5, 5)
404
405 # mesh <- "Lago_Luruaco"
406 # csi <- c(9, 17, 15, 19, 23, 17, 9, 9, 17, 23, 9, 17, 23)
407 # eta <- c(9, 9, 20, 12, 18, 18, 18, 12, 12, 12, 20, 20, 20)
408
409 # mesh <- "Corpo_Humano"
410 # csi <- c(8, 14, 6, 6, 8, 8, 8, 8, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)
411 # eta <- c(9, 16, 19, 19, 6, 6, 15, 15, 4, 5, 4, 4, 4, 4, 4, 4, 2, 2, 2, 2)
412
413 # mesh <- "Sapo"
414 # csi <- c(8, 5, 5, 8, 6, 7, 7, 3, 3, 6, 6, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2)
415 # eta <- c(29, 6, 6, 6, 8, 3, 3, 8, 9, 6, 6, 6, 7, 8, 10, 8, 6, 9, 8, 7, 4, 5, 7, 8, 8)
416
417 blocks <- 1:length(csi)
418 square.metrics <- c("ratio" = 1.0, "angle" = pi * 0.5, "compacity" = 16.0)
419 norm <- c("a" = 0.0, "b" = 10.0)
420
421 analysis.runtimes <- list()
422
423 main.folder <- paste0("Saidas", "/", mesh, "/")
424 metrics.file <- "metrics.csv"
425 mesh.files <- paste0("mesh-", blocks, ".vtk")
426 unions.file <- "unions.png"
427 rdata.file <- "rdata.RData"
428
429 min.precision <- 0.00000001
430 min.epsilon <- 0.0
431 inc.epsilon <- 0.00001
432 max.epsilon <- 0.2
433 percentage <- c("0.05" = 0.05, "0.1" = 0.1, "0.2" = 0.2)
434 kmeans.centers <- 5
435 quality <- c("good" = 0, "ratio" = 1, "angle" = 2, "compacity" = 3, "ra" = 16, "rc" = 17, "ac" =
18, "inf" = 19, "all" = 20)
436
437 print("Iniciando Analise de Qualidade")

```

```
438 current.start <- proc.time()
439
440 in.folder <- main.folder
441 out.folder <- paste0(main.folder, names(percentage), "/", "Analise", "/")
442 analyze.mesh(read.csv(paste0(in.folder, metrics.file)), in.folder, out.folder, percentage)
443
444 analysis.runtimes[[length(analysis.runtimes)+1]] <- proc.time() - current.start
445 print("Finalizando Analise de Qualidade")
446
447 save.image(file = paste0(main.folder, rdata.file))
```

Algoritmo A.5 – Identificação dos elementos de menor qualidade de malhas em coordenadas generalizadas

TRABALHOS PUBLICADOS PELO AUTOR

Trabalhos publicados pelo autor durante o programa.

Publicações principais do trabalho.

1. G. T. Naozuka, A. S. Felinto, N. M. L. Romeiro, E. R. Cirilo, P. L. Natti, **Development of an Android application to generate meshes in curvilinear coordinates from contour points acquired by GPS**, 2016 35th International Conference of the Chilean Computer Science Society (SCCC), Oct/2016, 1-10, (Qualis CC 2016, B3)

Publicações complementares.

1. A. I. S. d. Abreu, G. T. Naozuka, N. M. L. Romeiro, **Simulação numérica da dinâmica do crescimento de tumores com controle por radioterapia**, I Simpósio Paranaense de Equações Diferenciais (SPED), Nov/2017, 1-3, ISBN 978-85-7846-476-9
2. L. M. D. Okamoto, G. T. Naozuka, N. M. L. Romeiro, E. R. Cirilo, P. L. Natti, **Numerical simulation of pollutant dispersion in the atmosphere**, 24th ABCM International Congress of Mechanical Engineering (COBEM), Dec/2017, 1-9
3. R. P. Silva, G. T. Naozuka, S. M. Mastelini, A. S. Felinto, **Automatic luminous reflections detector using Global Threshold with increased luminosity contrast in images**, Journal of Electronic Imaging, Feb/2018, 1-36, (Qualis CC 2016, A2)

PATENTES DESENVOLVIDAS PELO AUTOR

Patentes desenvolvidas pelo autor durante o programa.

1. G. T. Naozuka, A. S. Felinto, E. R. Cirilo, N. M. L. Romeiro, P. L. Natti, **BorderMap**, 2017, Programa de Computador, BR512017001108-7, INPI - Instituto Nacional da Propriedade Industrial