



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE CIÊNCIAS**  
**DEPARTAMENTO DE COMPUTAÇÃO**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**  
**DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO**

**ELIAS SARAIVA BARROSO**

**GERAÇÃO DE MALHAS DE ALTA ORDEM PARA ANÁLISE ISOGOMÉTRICA**  
**UTILIZANDO ELEMENTOS DE BÉZIER RACIONAIS**

**FORTALEZA**

**2022**

ELIAS SARAIVA BARROSO

GERAÇÃO DE MALHAS DE ALTA ORDEM PARA ANÁLISE ISOGEOMÉTRICA  
UTILIZANDO ELEMENTOS DE BÉZIER RACIONAIS

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Ciência da Computação. Área de Concentração: Ciência da Computação

Orientador: Prof. Joaquim Bento Cavalcante Neto, D.Sc

Coorientador: Prof. Creto Augusto Vidal, D.Sc

Coorientador: Prof. Evandro Parente Junior, D.Sc

FORTALEZA

2022

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Sistema de Bibliotecas  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

B285g Barroso, Elias Saraiva.  
Geração de malhas de alta ordem para análise isogeométrica utilizando elementos de Bézier racionais /  
Elias Saraiva Barroso. – 2022.  
205 f. : il. color.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em  
Ciência da Computação, Fortaleza, 2022.  
Orientação: Prof. Dr. Joaquim Bento Cavalcante Neto.  
Coorientação: Prof. Dr. Creto Augusto Vidal e Prof. Dr. Evandro Parente Junior.

1. Geração de malhas de alta ordem. 2. Análise isogeométrica. 3. Elementos de Bézier racionais. 4.  
Método dos elementos finitos. I. Título.

CDD 005

---

ELIAS SARAIVA BARROSO

GERAÇÃO DE MALHAS DE ALTA ORDEM PARA ANÁLISE ISOGEOMÉTRICA  
UTILIZANDO ELEMENTOS DE BÉZIER RACIONAIS

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Ciência da Computação. Área de Concentração: Ciência da Computação

Aprovada em: 26 de Agosto de 2022

BANCA EXAMINADORA

---

Prof. Joaquim Bento Cavalcante Neto, D.Sc (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Creto Augusto Vidal, D.Sc (Coorientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Evandro Parente Junior, D.Sc (Coorientador)  
Universidade Federal do Ceará (UFC)

---

Prof. João Batista Marques de Sousa Junior, D.Sc  
Universidade Federal do Ceará (UFC)

---

Prof. Luiz Fernando Campos Ramos Martha, D.Sc  
Pontifícia Universidade Católica do Rio de Janeiro (PUC-RJ)

A Deus, por ter me concedido a oportunidade de realizar este trabalho.

## AGRADECIMENTOS

A meus pais, Eliseu e Maria José, por todo o amor e dedicação que me deram durante toda minha vida.

À minha esposa Renata, pelo amor, carinho, ajuda e companheirismo que sempre me deu e que foram essenciais para conclusão deste trabalho.

Aos meus familiares, meus irmãos Eliseu Neto e Lígia Barroso, minha cunhada Iara Inete, minhas tias Eliana Barroso e Regina Saraiva, meu tio Elísio Barroso, pela grande amizade e ajuda que me deram.

A meu orientador Joaquim Bento e coorientador Creto Vidal, pelas contribuições realizadas neste trabalho e todos os ensinamentos adquiridos no decorrer deste doutorado.

A meu coorientador Evandro Parente, pela amizade e exemplo de dedicação, competência e honestidade, que testemunhei ao longo destes 13 anos que trabalhamos juntos.

Ao professor John Evans, pela excelente orientação e contribuições dadas durante o intercâmbio que realizei na University of Colorado at Boulder.

Aos professores Luiz Fernando Martha e João Batista, por fazerem parte de minha banca examinadora.

Aos colegas do CRAb, Suzana, Daniel Texeira, Paulo Bruno, Tiago Sombra, Anderson, Erison, pela amizade, ajuda e discussões realizadas durante este doutorado.

Aos colegas do LMCV Marina Maia, Leonardo Ribeiro, Carlos David, Jamires, Bergson, João Pedro, Juliana, Francisco David, Edson, pela amizade, discussões e bons momentos vividos no laboratório e congressos durante este doutorado.

A UFC e ao MDCC, pela oportunidade de realização deste doutorado e suporte financeiro dado.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001. Também foi concedida uma bolsa pelo programa PDSE (Programa de Doutorado Sanduíche no Exterior), processo 88881.190187/2018-01.

## RESUMO

A Análise Isogeométrica (AIG) é um método numérico que vem recebendo atenção na última década. A principal proposta da AIG é aproximar, nos sistemas CAE (*Computer Aided Engineering*), as etapas de modelagem geométrica e de simulação numérica. Para essa finalidade, na AIG, os dois componentes usam as mesmas representações geométricas, como curvas e superfícies de Bézier, NURBS e T-Splines. Entretanto, em muitas aplicações, especialmente no caso de sistemas CAE baseados no Método dos Elementos Finitos, os modelos utilizados na análise numérica não são fornecidos diretamente pelos sistemas CAD, pois apenas a fronteira do modelo é representada. Assim, uma parametrização do interior do modelo deve ser construída para realização da análise isogeométrica. As técnicas de parametrização do domínio utilizando NURBS e T-Splines não são suficientemente robustas em casos de geometrias complexas, como em modelos com múltiplos furos ou com regiões estreitas. Além disto, o uso de modelos com superfícies aparadas apresentam grande complexidade, seja na sua parametrização ou quando são empregados diretamente em simulações numéricas. Por outro lado, técnicas de geração de malhas de alta ordem compostas por triângulos de Bézier racionais são mais robustas nesses casos. Este trabalho apresenta um algoritmo de geração de malhas isogeométricas compostas por triângulos de Bézier racionais de grau arbitrário, e aplica-o em modelos planos representados por curvas NURBS. O algoritmo é capaz de gerar malhas de boa qualidade mesmo no caso de modelos com pouca discretização e elevada curvatura. O algoritmo apresenta desempenho superior em comparação com outra alternativa da literatura, além de apresentar bom desempenho em geometrias complexas, e de produzir malhas com qualidade comparáveis às produzidas via otimização. O algoritmo é generalizado para modelos de superfície no espaço 3D, onde a geometria é dada por superfícies NURBS aparadas. Também são apresentadas técnicas de geração de malhas estruturadas de triângulos e superfícies de Bézier. Além disto, foi desenvolvida uma formulação de elementos de Bézier racionais para análise de cascas. Os algoritmos desenvolvidos são utilizados em problemas planos de elasticidade e transferência de calor, e em problemas de análise estática e de vibração livre de cascas, utilizando os elementos desenvolvidos neste trabalho. O desempenho dos elementos é estudado em vários exemplos de aplicação, demonstrando sua capacidade de convergência com o refinamento do modelo.

**Palavras-chave:** análise isogeométrica; geração de malhas de alta ordem; triângulo de Bézier; superfície de Bézier; superfícies aparadas; elemento de casca degenerado.

## ABSTRACT

Isogeometric Analysis (IGA) is a numerical method that has been receiving increasing attention in the last decade. The main goal of IGA is to closely couple geometric modeling with numerical analysis in Computer-Aided Engineering (CAE) systems. To that end, in IGA both components use the same geometric representation, e.g., Bézier curves and surfaces, NURBS, and T-Splines. However, in many cases, especially in CAE systems based on the Finite Element Method, the geometric representation in a CAD system cannot be directly employed in numerical simulation, as only the boundary of the geometry is parametrized. In these cases, an interior parametrization must be constructed before performing isogeometric analysis. The domain parametrization techniques for NURBS and T-Splines are not enough robust to handle complex geometries, e.g. geometries with multiple holes and narrow regions. In addition, the use of trimmed surfaces in IGA has considerable complexity, either in the construction of their parameterization or when these models are employed directly in numerical simulations. On the other hand, the use of high-order unstructured mesh generator algorithms to construct domain parametrization is more robust in these cases. This work presents an algorithm for generation of unstructured geometrically exact meshes composed of rational Bézier triangles of arbitrary degree and applies it to plane models described by NURBS curves in a Boundary-Representation (B-Rep) scheme. The proposed algorithm is capable of generating high quality coarse meshes even when high curvature segments are considered. The proposed algorithm attains superior performance when compared to a well-known algorithm in the literature and produces meshes with similar quality in comparison to meshes obtained through quality optimization. The algorithm is generalized to surface models in 3D, where geometry is given by trimmed NURBS. Structured meshing algorithms for generation of rational Bézier triangles and quadrilaterals are also presented. Moreover, a shell formulation based on degenerated solid for rational Bézier elements is presented. The algorithms developed are used in elasticity and heat transfer problems, and static and free vibration analysis of shells, using the elements developed in this work. The performance of rational Bézier elements is assessed in several numerical examples, demonstrating its convergence under mesh refinement.

**Keywords:** isogeometric analysis; high-order mesh generation; Bézier triangle; Bézier surface; trimmed surfaces; degenerated shell element.

## LISTA DE FIGURAS

Figura 1 – Etapas realizadas em uma aplicação CAE. . . . .	27
Figura 2 – Exemplo de curvas de Bézier cúbicas. . . . .	28
Figura 3 – Definição de contorno de fontes utilizando representação de Bézier. . . . .	29
Figura 4 – Polinômios de Bernstein para vários graus. . . . .	30
Figura 5 – Algoritmo de <i>de Casteljau</i> aplicado a curva de Bézier cúbica. . . . .	31
Figura 6 – Subdivisão aplicada a uma curva de Bézier cúbica. . . . .	31
Figura 7 – Elevação de grau aplicada a uma curva de Bézier cúbica. . . . .	32
Figura 8 – Efeito da variação do peso. . . . .	33
Figura 9 – Representação de um arco de círculo utilizando Bézier Racional. . . . .	33
Figura 10 – Exemplo de superfície Bézier. . . . .	34
Figura 11 – Triângulo de Bézier cúbico. . . . .	35
Figura 12 – Código da avaliação das funções de base do triângulo de Bézier. . . . .	37
Figura 13 – Código da avaliação da primeira derivada dos polinômios de Bernstein biva- riados. . . . .	39
Figura 14 – Um oitavo de uma esfera representado por um triângulo de Bézier racional quártico. . . . .	40
Figura 15 – Funções base B-Spline quadráticas. . . . .	42
Figura 16 – Funções de base B-Spline cúbica. . . . .	43
Figura 17 – Funções de base B-Spline quadráticas com $\Xi = [0, 0, 0, 0.5, 0.5, 1, 1, 1]$ . . . . .	44
Figura 18 – Efeito da multiplicidade em uma B-Spline quadrática. . . . .	45
Figura 19 – Exemplo de inserção de <i>knot</i> em curva B-Spline ( $\bar{\xi} = 2$ ). . . . .	46
Figura 20 – Controle local obtido pelo inserção de <i>knots</i> . . . . .	47
Figura 21 – Elevação de grau aplicada a uma curva B-Spline quadrática. . . . .	47
Figura 22 – Aplicação da extração de Bézier em uma curva cúbica com quatro <i>knot spans</i> . . . . .	49
Figura 23 – Curva NURBS $\mathbf{c}(\xi)$ em $\mathbb{R}^2$ sendo obtida da projeção de uma curva B-Spline $\mathbf{c}^w(\xi)$ em $\mathbb{R}^3$ . . . . .	51
Figura 24 – Bases NURBS e B-Spline quadráticas com $\Xi = [0, 0, 0, 0.5, 1, 1, 1]$ . . . . .	52
Figura 25 – Semi circunferência construída com NURBS e um arco de 180° construído com B-Spline. . . . .	52
Figura 26 – Exemplo de um jarro modelado por uma superfície NURBS. . . . .	53
Figura 27 – Exemplo de superfícies construídas a partir das curvas do contorno. . . . .	54

Figura 28 – Placa com furo representado por 8 superfícies de Bézier racionais. . . . .	55
Figura 29 – Superfície NURBS com aparos. . . . .	56
Figura 30 – Operação de subtração entre um toro e um subespaço definido por um plano. . . . .	57
Figura 31 – Placa com furos circulares e restrições. . . . .	60
Figura 32 – Modelo B-Rep com três regiões. . . . .	60
Figura 33 – Modelo B-Rep de uma placa quadrada com furo detalhado. . . . .	61
Figura 34 – Estrutura da dados baseadas em aresta. . . . .	62
Figura 35 – Exemplo de malhas em várias dimensões. . . . .	63
Figura 36 – Exemplo de aplicação da técnica de geração de malhas via decomposição espacial. . . . .	65
Figura 37 – Exemplo de aplicação do algoritmo de avanço de fronteira. . . . .	66
Figura 38 – Razão arco/corda de vários segmentos circulares. . . . .	70
Figura 39 – Aplicação do teste de proximidade. . . . .	72
Figura 40 – Detecção de cantos com ângulos pequenos. O segmento à direita interrompe a etapa de proximidade. . . . .	73
Figura 41 – Efeito da parametrização de arestas curvas sobre a variação do determinante da matriz Jacobiana em elementos de alta ordem. Os casos (a) e (b) apresentam curvas com boa parametrização, enquanto que os casos (c) e (d) apresentam curvas com parametrização ruim. . . . .	74
Figura 42 – Elemento finito T6 de boa qualidade construído sobre uma aresta com parametrização ruim (Figura 41d). . . . .	74
Figura 43 – Processo de criação da <i>quadtree</i> usada no critério de disparidade. . . . .	75
Figura 44 – Processo de suavização da discretização. Os vértices de cor laranja não são movidos. . . . .	76
Figura 45 – Efeito da aplicação da etapa de suavização. . . . .	77
Figura 46 – Exemplo de malhas de elementos lineares obtidas via mapeamento transfinito. . . . .	78
Figura 47 – Exemplo de malhas de elementos quadráticos obtidas via mapeamento transfinito. . . . .	78
Figura 48 – Mapeamento transfinito aplicado para geração de elementos quadrilaterais de Bézier racional. . . . .	80
Figura 49 – Exemplo de malhas de elementos cúbicos isogeométricos obtidos via mapeamento transfinito. . . . .	81

Figura 50 – Etapas do algoritmo proposto. . . . .	83
Figura 51 – $d_l$ medida em segmentos côncavos e convexos. . . . .	85
Figura 52 – Seleção de vértice ideal para criação de novo triângulo . . . . .	86
Figura 53 – Decomposição do domínio usando <i>KDTree</i> . . . . .	87
Figura 54 – Processamento de uma iteração do <i>Back-tracking</i> . . . . .	88
Figura 55 – Regiões candidatas avaliadas na etapa de <i>Back-tracking</i> : (a) Região inicial; (b) Triangulação; (c) Geração de malha. . . . .	89
Figura 56 – Restauração da fronteira realizada em um triângulo de Bézier cúbico. . . . .	89
Figura 57 – Procedimento de divisão em quatro elementos aplicado a elementos adjacentes a duas arestas curvas da entrada. . . . .	90
Figura 58 – Procedimento de divisão em três elementos aplicado a elementos adjacentes a três arestas curvas da entrada. . . . .	91
Figura 59 – Substituição do contorno de uma malha quártica: (a) com suavização por elasticidade e (b) sem suavização por elasticidade. . . . .	92
Figura 60 – Efeito da suavização dos pesos em $W$ . . . . .	92
Figura 61 – Efeito da suavização dos pesos em $ \mathbf{J} $ . . . . .	93
Figura 62 – Exemplo de aplicação da suavização por elasticidade e os dos deslocamentos obtidos a partir da análise linear. Apenas a vizinhança de arestas curvas apresentam deslocamentos relevantes. . . . .	94
Figura 63 – Etapas para encontrar o conjunto de elementos processados na SAO ( $\alpha = 2$ ). . . . .	95
Figura 64 – Diagrama de classes da estrutura de dados de sub-malha. . . . .	95
Figura 65 – Aplicação do algoritmo mostrado no Quadro 4 a um conjunto de 24 elementos, resultando em duas sub-malhas com 8 e 16 elementos. . . . .	97
Figura 66 – $J_{ts}$ dos triângulos com base reta e duas bases circulares. . . . .	100
Figura 67 – Valores de $J_{ts}$ por curvatura $\theta$ da base circular de triângulos. . . . .	101
Figura 68 – Efeito da suavização dos pesos em $Q_j^e$ . . . . .	101
Figura 69 – Efeito da suavização dos pesos em $J_{ts}^e$ . . . . .	102
Figura 70 – Geração de malhas em superfície aparadas . . . . .	102
Figura 71 – Rotação e escala de vetores no espaço Cartesiano da superfície. . . . .	106
Figura 72 – Malhas da AIG e MEF para um problema bidimensional. Os pontos de controle interpolados são destacados em azul. . . . .	116
Figura 73 – Modelo sólido degenerado. . . . .	120

Figura 74 – Deslocamento de um ponto arbitrário no modelo de sólido degenerado. . . . .	121
Figura 75 – Vetores diretores nos pontos de controle da casca. . . . .	122
Figura 76 – Esquema de cálculo dos vetores normais nos pontos de controle. . . . .	124
Figura 77 – Malhas geradas nos modelos do círculo e quarto de cilindro. . . . .	131
Figura 78 – Distribuição dos elementos por faixas de qualidade para os modelos do círculo e quarto de cilindro. . . . .	132
Figura 79 – Malhas geradas pelo PMGen nos modelos da placa com furo e bocal. . . . .	134
Figura 80 – Distribuição dos elementos por faixas de qualidade nos modelos da placa com furo e bocal. . . . .	134
Figura 81 – Malhas geradas pelo PMGen nos modelos da placa complexa e braço de torque. . . . .	135
Figura 82 – Distribuição dos elementos por faixas de qualidade nos modelos da placa complexa e braço de torque. . . . .	136
Figura 83 – Geometria dos problemas para aplicação da discretização automática do contorno. . . . .	136
Figura 84 – Malhas obtidas no modelo do violão. . . . .	137
Figura 85 – Malhas obtidas no modelo dos rotores. . . . .	138
Figura 86 – Malhas geradas pelo PMGen e Gmsh. . . . .	139
Figura 87 – Malhas quárticas da parte mecânica geradas pelo algorithm AFP com SAO. . . . .	143
Figura 88 – Histograma de qualidade dos elementos da parte mecânica. . . . .	144
Figura 89 – Descrição da superfície do modelo de um oitavo de esfera com corte. . . . .	145
Figura 90 – Malhas geradas para o modelo de um oitavo de esfera com corte. . . . .	146
Figura 91 – Descrição da superfície de foma livre. . . . .	147
Figura 92 – Curvaturas máximas absolutas no modelo da superfície de forma livre. . . . .	147
Figura 93 – Malhas geradas no exemplo da superfície de forma livre. . . . .	148
Figura 94 – Descrição do modelo da faixa curva. . . . .	149
Figura 95 – Malhas geradas para o modelo da faixa curva. . . . .	150
Figura 96 – Malhas da faixa curva no espaço paramétrico para $\Lambda = (300, 90^\circ, 150)$ . . . . .	151
Figura 97 – Descrição do modelo da faixa curva modificada. . . . .	151
Figura 98 – Malhas geradas no exemplo da faixa modificada. . . . .	152
Figura 99 – Descrição do modelo da casca esférica com furos. . . . .	153
Figura 100 – Malhas geradas no exemplo casca esférica com furos. . . . .	154
Figura 101 – Descrição do cilindro de parede espessa. . . . .	156

Figura 102–Malhas 1-4 utilizadas para modelar o cilindro com parede espessa. . . . .	157
Figura 103–Norma $L^2$ do campo de temperatura obtida no cilindro espesso. . . . .	158
Figura 104–Campo de temperatura $T$ no cilindro espesso. . . . .	158
Figura 105–Descrição do chapa infinita com furo circular. . . . .	159
Figura 106–Malhas utilizadas para modelar a chapa infinita com furo. . . . .	160
Figura 107–Particionamento do domínio necessário para aplicação do mapeamento trans- finito. . . . .	161
Figura 108–Norma $L^2$ do campo de tensões da chapa infinita com furo. . . . .	162
Figura 109–Campo das tensões- $xx$ da chapa infinita com furo. . . . .	162
Figura 110–Descrição do domínio de análise considerado para estudo da suavização dos pesos. . . . .	163
Figura 111–Malhas utilizadas no estudo da suavização dos pesos. . . . .	163
Figura 112–Norma $L^2$ do campo de tensões obtidas de malhas com e sem a suavização de pesos. . . . .	164
Figura 113–Descrição da casca de Scordelis-Lo. . . . .	164
Figura 114–Malhas usadas na casca Scordelis-Lo. . . . .	165
Figura 115–Convergência do deslocamento $w_A$ para casca de Scordelis-Lo. . . . .	166
Figura 116–Campo de deslocamento $w$ (escala $20\times$ ) do problema Scordelis-Lo <i>roof</i> . . .	167
Figura 117–Modos de vibração da casca Scordelis-Lo <i>roof</i> (diagrama de cores para resposta $w$ ). . . . .	168
Figura 118–Descrição do cilindro com carga pontual. . . . .	169
Figura 119–Malhas utilizadas para modelar o cilindro. . . . .	169
Figura 120–Convergência do deslocamento $w_P$ para o cilindro com carga pontual. . . . .	170
Figura 121–Campo de deslocamento $w$ (fator = $1e6$ ) para o cilindro com carga pontual. .	170
Figura 122–Descrição do hemisfério com furo. . . . .	171
Figura 123–Malhas utilizadas para modelar o hemisfério com furo. . . . .	172
Figura 124–Convergência do deslocamento $u_A$ para o hemisfério com furo. . . . .	173
Figura 125–Campo de deslocamento $u$ (fator de escala = 30) no hemisfério com furo. . .	173
Figura 126–Descrição da casca de forma livre. . . . .	174
Figura 127–Malhas utilizadas para modelar a casca com curvatura dupla. . . . .	175
Figura 128–Convergência do deslocamento $v$ para a casca com curvatura dupla. . . . .	176
Figura 129–Campo de deslocamento $v$ da casca de forma livre. . . . .	177

Figura 130–Descrição da casca de Scordelis-Lo com furo. . . . .	177
Figura 131–Particionamento do modelo considerado para geração de malhas estruturadas.	178
Figura 132–Malhas utilizadas na casca de Scordelis-Lo com furo. . . . .	179
Figura 133–Convergência do deslocamento $w$ para a casca de Scordelis-Lo com furo. . .	179
Figura 134–Campo de deslocamento $w$ (fator de escala = 8) da casca de forma livre. . .	180
Figura 135–Descrição da faixa curva apoiada. . . . .	180
Figura 136–Convergência da frequência natural para faixa curva. . . . .	181
Figura 137–Modos de vibração da faixa curva apoiada (diagrama de cores para resposta $w$ ).	182
Figura 138–Descrição da casca esférica com furos. . . . .	183
Figura 139–Convergência do deslocamento $w_p$ para a casca esférica com furos. . . . .	184
Figura 140–Campo de deslocamento $w$ (fator de escala = $2e4$ ) da casca esférica com furos.	184

## LISTA DE QUADROS

Quadro 1	–	Algoritmo para determinação da coordenada paramétrica que define um segmento com o comprimento fornecido. . . . .	69
Quadro 2	–	Pseudocódigo do operador <i>MakeSet</i> . . . . .	96
Quadro 3	–	Pseudocódigo do operador <i>Union</i> . . . . .	96
Quadro 4	–	Pseudocódigo do algoritmo para avaliar sub-malhas. . . . .	97

## LISTA DE TABELAS

Tabela 1 – Intervalo de contribuição das bases B-Spline para diferentes vetores $\Xi$ . . . . .	43
Tabela 2 – Qualidade das malhas obtidas nos modelos do círculo e quarto de cilindro. . . . .	132
Tabela 3 – Parâmetros usados para otimização das malhas. . . . .	133
Tabela 4 – Qualidade das malhas obtidas nos modelos da placa com furo e bocal. . . . .	134
Tabela 5 – Qualidade das malhas obtidas nos modelos da placa complexa e braço de torque. . . . .	135
Tabela 6 – Qualidade das malhas para diferentes níveis de discretização. . . . .	137
Tabela 7 – Parâmetros de discretização do contorno usados na comparação com o Gmsh. . . . .	139
Tabela 8 – Parâmetros do Gmsh (versão 4.4.1) usados para gerar as malhas do violão e rotores. . . . .	140
Tabela 9 – Tamanho e qualidade das malhas obtidas pelo PMGen e Gmsh. . . . .	140
Tabela 10 – Tempos de processamento obtidos pelo PMGen e Gmsh. . . . .	141
Tabela 11 – Dados referentes as sub-malhas do violão e rotores. . . . .	141
Tabela 12 – Métricas de qualidade e custo computacional da SAO em análises globais e locais. . . . .	141
Tabela 13 – Influência da SAO na qualidade das malhas e tempos de processamento. . . . .	142
Tabela 14 – Qualidade das malhas para o modelo de um oitavo de esfera com corte. . . . .	146
Tabela 15 – Qualidade das malhas para o modelo da superfície de forma livre. . . . .	149
Tabela 16 – Qualidade das malhas para o modelo da faixa curva. . . . .	150
Tabela 17 – Qualidade das malhas para o modelo da faixa modificada. . . . .	153
Tabela 18 – Qualidade das malhas para o modelo de casca esférica. . . . .	155
Tabela 19 – Frequências de vibração da casca Scordelis-Lo <i>roof</i> . . . . .	167
Tabela 20 – Parâmetros de discretização das malhas da casca de forma livre. . . . .	175
Tabela 21 – Parâmetros de discretização das malhas da casca de Scordelis-Lo com furo. . . . .	178
Tabela 22 – Parâmetros de discretização das malhas da faixa curva apoiada. . . . .	181
Tabela 23 – Frequências de vibração da faixa curva apoiada. . . . .	182
Tabela 24 – Parâmetros de discretização das malhas da casca esférica apoiada. . . . .	183
Tabela 25 – Pontos de controle e vetor de <i>knots</i> de um quarto da casca de Scordelos-Lo. . . . .	201
Tabela 26 – Pontos de controle e vetor de <i>knots</i> da casca de Scordelos-Lo. . . . .	201
Tabela 27 – Pontos de controle e vetor de <i>knots</i> do furo central da casca de Scordelos-Lo. . . . .	201
Tabela 28 – Pontos de controle e vetor de <i>knots</i> do oitavo de cilindro. . . . .	202

Tabela 29 – Pontos de controle e vetor de <i>knots</i> do oitavo de esfera com corte. . . . .	202
Tabela 30 – Pontos de controle e vetor de <i>knots</i> da superfície de forma livre. . . . .	202
Tabela 31 – Pontos de controle e vetor de <i>knots</i> da faixa curva. . . . .	203
Tabela 32 – Pontos de controle e vetor de <i>knots</i> da faixa curva modificada (furos adicionais).204	
Tabela 33 – Pontos de controle e vetor de <i>knots</i> da casca esférica (parte 1). . . . .	205
Tabela 34 – Pontos de controle e vetor de <i>knots</i> da casca esférica (parte 2). . . . .	206

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>20</b>
<b>1.1</b>	<b>Objetivos e Contribuições</b>	<b>21</b>
<b>1.2</b>	<b>Organização da Tese</b>	<b>24</b>
<b>2</b>	<b>MODELAGEM GEOMÉTRICA</b>	<b>26</b>
<b>2.1</b>	<b>Representações de curvas e superfícies de Bézier</b>	<b>28</b>
<b>2.1.1</b>	<i>Bézier Racional</i>	<b>32</b>
<b>2.1.2</b>	<i>Superfícies de Bézier</i>	<b>33</b>
<b>2.2</b>	<b>Triângulos de Bézier</b>	<b>35</b>
<b>2.2.1</b>	<i>Bases racionais</i>	<b>39</b>
<b>2.3</b>	<b>B-Splines</b>	<b>41</b>
<b>2.3.1</b>	<i>Inserção de knot e elevação de grau</i>	<b>45</b>
<b>2.3.2</b>	<i>Extração de Bézier</i>	<b>48</b>
<b>2.4</b>	<b>NURBS</b>	<b>49</b>
<b>2.4.1</b>	<i>Superfícies NURBS</i>	<b>53</b>
<b>2.5</b>	<b>Superfície de Coons</b>	<b>53</b>
<b>2.6</b>	<b>Superfícies aparadas</b>	<b>55</b>
<b>2.7</b>	<b>Ajuste de curvas e superfícies</b>	<b>57</b>
<b>2.8</b>	<b>Representação de Fronteira</b>	<b>59</b>
<b>3</b>	<b>GERAÇÃO DE MALHAS</b>	<b>63</b>
<b>3.1</b>	<b>Geração de malhas unidimensional</b>	<b>68</b>
<b>3.1.1</b>	<i>Discretização uniforme</i>	<b>69</b>
<b>3.1.2</b>	<i>Discretização recursiva</i>	<b>71</b>
<b>3.1.3</b>	<i>Detecção de regiões próximas</i>	<b>72</b>
<b>3.1.4</b>	<i>Variação do vetor tangente</i>	<b>73</b>
<b>3.1.5</b>	<i>Disparidade</i>	<b>75</b>
<b>3.1.6</b>	<i>Suavização</i>	<b>76</b>
<b>3.2</b>	<b>Geração de malhas estruturadas</b>	<b>77</b>
<b>3.2.1</b>	<i>Mapeamento Transfinito Bilinear</i>	<b>77</b>
<b>3.2.2</b>	<i>Mapeamento Transfinito na AIG</i>	<b>79</b>
<b>3.3</b>	<b>Geração de malhas não estruturadas para modelos planos</b>	<b>81</b>

3.3.1	<i>Geração de malha poligonal</i>	84
3.3.2	<i>Elevação de grau e restauração da fronteira</i>	89
3.3.3	<i>Tratamento de singularidades</i>	90
3.3.4	<i>Suavização de malhas de alta ordem</i>	91
3.3.4.1	<i>Análise Linear Localizada</i>	94
3.3.4.2	<i>Discussão sobre a implementação</i>	95
3.4	<b>Métricas de qualidade</b>	98
3.5	<b>Geração de malhas em superfícies aparadas</b>	102
3.5.1	<i>Adaptações para geração de malhas estruturadas</i>	104
3.5.2	<i>Geometria diferencial</i>	105
3.5.3	<i>Adaptações na função densidade</i>	106
3.5.4	<i>Adaptações na geração de malhas unidimensionais</i>	108
3.5.5	<i>Adaptações na geração de malhas não-estruturadas</i>	109
3.6	<b>Otimização de malhas</b>	110
3.6.1	<i>Otimização de malhas de superfície</i>	112
4	<b>ANÁLISE ISOGEOMÉTRICA</b>	114
4.1	<b>Estado Plano de Tensões</b>	116
4.1.1	<i>Deformações</i>	117
4.1.2	<i>Equações de Equilíbrio</i>	117
4.2	<b>Análise de Cascas</b>	120
4.2.1	<i>Cálculo dos vetores diretores</i>	123
4.2.2	<i>Deformações</i>	125
4.2.3	<i>Matriz constitutiva</i>	126
4.2.4	<i>Integração do elemento</i>	127
4.2.5	<i>Vibrações livres</i>	128
4.3	<b>Implementação computacional</b>	129
5	<b>EXEMPLOS NUMÉRICOS</b>	130
5.1	<b>Geração de Malhas</b>	130
5.1.1	<i>Desempenho do algoritmo AFP</i>	131
5.1.2	<i>Aplicação da discretização recursiva em problemas planos</i>	136
5.1.3	<i>Eficiência do algoritmo AFP em geometrias complexas</i>	138
5.1.4	<i>Efeito da SAO na qualidade das malhas</i>	142

5.1.5	<i>Geração de malhas de superfícies</i>	144
5.1.6	<i>Geração de malhas em superfícies aparadas</i>	149
5.2	<b>Análises Numéricas</b>	155
5.2.1	<i>Transferência de calor em cilindro de parede espessa</i>	156
5.2.2	<i>Chapa infinita tracionada com furo circular</i>	158
5.2.3	<i>Efeito da suavização dos pesos</i>	162
5.2.4	<i>Casca de Scordelis-Lo</i>	164
5.2.5	<i>Cilindro submetido a carga concentrada</i>	168
5.2.6	<i>Hemisfério submetido a carga concentrada</i>	171
5.2.7	<i>Casca de forma livre</i>	174
5.2.8	<i>Casca de Scordelis-Lo com furo</i>	177
5.2.9	<i>Faixa curva</i>	180
5.2.10	<i>Casca esférica com furos</i>	182
6	<b>CONCLUSÃO</b>	186
	<b>REFERÊNCIAS</b>	189
	<b>APÊNDICE A – DESCRIÇÃO DOS EXEMPLOS</b>	201

## 1 INTRODUÇÃO

Simulações numéricas são importantes em diversas áreas da Engenharia, como nas indústrias Automobilística, Naval e Aeroespacial, além de serem utilizadas em outras áreas, como em animações fisicamente realistas. Os sistemas CAE (*Computer Aided Engineering*), são sistemas que integram simulações numéricas a sistemas de modelagem de problemas e visualização de resultados. Nesses sistemas, a geometria do problema é modelada através de ferramentas CAD (*Computer Aided Design*), que são programas de computador especializados no projeto, modificação e otimização de modelos diversos.

Em sistemas CAE baseados no Método dos Elementos Finitos (MEF), o pré-processamento da simulação consiste nas etapas de modelagem geométrica, geração de malha (ou discretização) e definição de atributos da simulação. Na etapa de geração de malha, o modelo geométrico é representado por uma malha de elementos que define o modelo de análise. Características da malha, como tamanho, distribuição e qualidade dos elementos, influenciam diretamente na precisão e custo computacional da simulação numérica.

Atualmente o MEF é a opção mais utilizada na análise de estruturas. Na formulação isoparamétrica do MEF, tanto os deslocamentos quanto a geometria da estrutura são aproximados com o uso de funções polinomiais<sup>1</sup>. Assim, salvo os casos de geometrias simples, a resposta do MEF contém tanto erros devido à aproximação do campo de deslocamentos, quanto devido à aproximação da geometria do modelo. Ambos os erros são reduzidos à medida que a malha de elementos finitos é refinada, o que pode ser feito aumentando o número de elementos (refinamento- $h$ ) ou aumentando o grau dos polinômios utilizados (refinamento- $p$ ).

Uma alternativa ao MEF convencional é a Análise Isogeométrica (AIG)<sup>2,3</sup>. A principal vantagem da AIG é utilizar na análise numérica as mesmas funções utilizadas pelos sistemas CAD para Modelagem Geométrica, como as B-Splines e as NURBS. Dessa forma, a geometria do problema é representada de forma exata, independentemente do nível de discretização considerado, eliminando o erro na representação da geometria que existe no MEF. Adicionalmente, B-Splines e NURBS possuem grau de continuidade maior que os normalmente utilizados no MEF, fazendo com que a AIG convirja mais rapidamente em certos problemas.

A discretização do modelo na AIG baseada em NURBS é obtida pela aplicação de algoritmos clássicos de Modelagem Geométrica, eliminando a etapa de geração de malha no processo de análise estrutural pelo MEF. Assim, o programa de análise trabalha diretamente com a geometria exata do modelo e pode gerenciar os procedimentos de discretização, possibilitando

que análises adaptativas sejam realizadas automaticamente ou semi-automaticamente, de maneira mais simples do que é feito no MEF<sup>2</sup>.

Apesar das vantagens da AIG, seu uso na prática é limitado pela forma como a modelagem da geometria do problema é realizada. Um exemplo disso é o caso da análise de modelos sólidos, onde são utilizadas representações tridimensionais, como NURBS trivariadas. Usualmente, os programas CAD representam sólidos por meio de suas fronteiras, ou B-Rep (*boundary representation*), onde apenas o contorno do modelo é representado. Assim, o modelo de análise não pode ser obtido diretamente do modelo geométrico. O mesmo ocorre no caso de modelos planos, onde o modelo de análise é dado por superfícies NURBS, mas o problema é modelado usando curvas NURBS. Outra questão relevante surge quando se consideram modelos de superfícies aparadas (*trimmed surfaces*). Neste caso, esquemas especiais de integração são necessários devido à complexa descrição da geometria dos modelos<sup>4</sup>.

Há trabalhos na literatura que abordam a geração de modelos NURBS e T-Splines para geometrias descritas usando B-Rep, tanto em casos 2D quanto em casos 3D. No entanto, essas abordagens não atendem a requisitos importantes<sup>5,6</sup>, tais como: ser aplicável a geometrias complexas, e.g., objetos com *genus* arbitrário ou objetos com grande variação de tamanho; ser automático, ou seja, não necessitar de intervenção do usuário; garantir a representação geométrica exata; e fornecer uma parametrização adequada para análise numérica. Os problemas acima são conhecidos na área de análise numérica no contexto do MEF, e são resolvidos com o uso de técnicas de geração de malhas.

Desta forma, o uso de malhas não estruturadas de triângulos e tetraedros de Bézier racionais tem sido explorado como uma alternativa para acoplar a Análise Isogeométrica e a representação de fronteira, onde algoritmos robustos de geração de malhas de alta ordem são empregados<sup>5,6</sup>. Além de garantir a representação exata da geometria, o uso desses elementos permite a aplicação de refinamentos *h* e *p* de modo similar ao realizado na AIG clássica baseada em NURBS.

## 1.1 Objetivos e Contribuições

O presente trabalho tem como objetivo principal o desenvolvimento de uma metodologia para análise isogeométrica baseada em elementos de Bézier, incluindo a geração de malhas de alta ordem e a formulação de elementos para análise de cascas.

Inicialmente, esta tese apresenta um algoritmo para parametrização automática de

modelos planos, descritos por uma B-Rep composta de curvas NURBS, em uma malha de triângulos de Bézier racionais de geometria exata e grau arbitrário. O algoritmo proposto respeita a parametrização de contorno do modelo e não requer refinamentos adicionais. Este requerimento é importante tanto quando se deseja manter a compatibilidade implícita das malhas em modelos com diversas regiões, permitindo inclusive que diferentes algoritmos de geração de malha sejam empregados com diferentes elementos (e.g., malhas de quadriláteros e triângulos de Bézier) quanto em situações que requerem o remalhamento, como em simulações de evolução de trincas. Esta característica também é importante para modelos de superfícies aparados, onde a parametrização de duas curvas de aparo (*trimming curves*), obtidas como resultado de uma interseção entre duas superfícies, deve ser compatível para garantir a estanqueidade do modelo.

Além disto, a função densidade utilizada no algoritmo permite obter boa transição em relação ao tamanho dos elementos no interior da malha, uma característica importante em problemas práticos onde há grande variação na discretização de entrada. O algoritmo também inclui etapas de suavização dos elementos de alta ordem, que buscam melhorar a qualidade da malha e evitar elementos inválidos e distorcidos. Usualmente, as etapas de suavização de alta ordem possuem alto custo computacional, sendo geralmente maior que o custo de geração da malha poligonal. Entretanto, no algoritmo proposto, as etapas de suavização da malha de alta ordem são realizadas de forma localizada e eficiente, objetivando manter a etapa de geração de malhas poligonal como a de custo computacional dominante do algoritmo.

A robustez do algoritmo proposto permite que ele seja aplicado em geometrias complexas com *genus* arbitrário ou em objetos com grande variação de tamanho. Além disso, ele apresenta eficácia e eficiência superiores a de outros algoritmos similares existentes na literatura<sup>5, 7, 8</sup>. Destaca-se, ainda, sua capacidade de gerar malhas de elementos racionais de alta qualidade, com garantia de geometria exata, o que contrasta com as abordagens tradicionais de geração de elementos de alta ordem no contexto do MEF.

Este trabalho avança o estado da arte atual sobre parametrização planar para AIG usando Triângulos de Bézier racionais<sup>9, 5</sup> nos seguintes aspectos. Os segmentos curvos são considerados na avaliação da função de densidade dos elementos usada durante a etapa de malha linear, em vez de usar segmentos retos. Modificações na topologia são realizadas para remover singularidades de elementos de alta ordem. Essas singularidades ocorrem eventualmente quando um elemento é adjacente a múltiplos segmentos curvos de entrada, e requerem a modificação da topologia da malha para serem removidas. Por fim, uma etapa de suavização de alta ordem é proposta para evitar elementos inválidos e melhorar a qualidade da malha. Esse procedimento é baseado em uma estratégia de deformação de malhas via análise elástica e suavização dos

pesos aplicada a modelos racionais. A aplicação dessas etapas em todos os elementos da malha tem um alto custo computacional. Para evitar isso, essas etapas são empregadas localmente em grupos de elementos disjuntos, mantendo a eficiência computacional do algoritmo de geração. Não se verificam trabalhos na literatura que tenham proposto uma estratégia de suavização local semelhante que atue nas coordenadas e pesos dos pontos de controle para evitar elementos inválidos e melhorar a qualidade dos elementos racionais. De fato, também não se verifica uma estratégia semelhante na literatura de geração de malhas de alta ordem. Este algoritmo foi publicado no artigo:

Barroso, E. S., Evans, J. A., Cavalcante-Neto, J. B., Vidal, C. A., e Parente, E., Jr. (2022). *An efficient automatic mesh generation algorithm for planar isogeometric analysis using high-order rational Bézier triangles*. Em *Engineering with Computers*. <https://doi.org/10.1007/s00366-022-01613-w>.

A implementação do algoritmo proposto está disponível no *PMGen* (Plane Mesh Generator), um programa de código aberto escrito em linguagem C++ e disponível em <<https://github.com/lmcv-ufc/PMGen>>. Sua interface gráfica de usuário possui funcionalidades para modelagem geométrica, discretização de modelos isogeométricos e visualização das malhas geradas e das métricas de qualidade de elementos discutidas neste trabalho. Além disso, *PMGen* é capaz de gerar malhas automaticamente usando procedimentos de subdivisão de curvas mesmo no caso de geometrias complexas. Assim, *PMGen* é uma contribuição para a comunidade acadêmica e técnica que pesquisa na área da AIG. O repositório contém todos os arquivos relacionados aos exemplos bidimensionais apresentados neste trabalho.

A tese também desenvolve uma versão do algoritmo proposto para geração de malhas de superfície em superfícies aparadas, onde o processo de geração da malha ocorre no espaço paramétrico de uma superfície NURBS com curvas de aparo. O algoritmo proposto é capaz de gerar malhas de alta ordem de elementos superfície em geometrias complexas, de alta curvatura e regiões estreitas. Finalmente, a tese apresenta algoritmos de geração de malhas estruturadas isogeométricas, possibilitando o uso de elementos quadrilaterais de Bézier racionais na análise estrutural de cascas.

Com relação à análise de estruturas, inicialmente foi implementada uma formulação isogeométrica baseada em superfícies e triângulos de Bézier para análise de problemas bidimensionais com modelos de estado plano de tensão. Ressalta-se que tais modelos são utilizados na etapa de suavização de alta ordem dos métodos de geração de malhas.

Finalmente, foi desenvolvida uma formulação isogeométrica para análise estática e de vibração livre de cascas. A formulação é baseada na teoria de Reissner-Mindlin e na

abordagem de sólidos degenerados, onde são considerados graus de liberdade de translação e rotação nos pontos de controle do elemento. Assim, uma estratégia para o cálculo dos vetores diretores nos pontos de controle é desenvolvida.

As técnicas de geração de malhas e as formulações isogeométricas desenvolvidas neste trabalho são aplicadas em vários exemplos, onde são avaliadas com relação a eficácia, precisão e eficiência.

## 1.2 Organização da Tese

A presente tese está dividida em 6 capítulos. No Capítulo 2, é realizada uma revisão bibliográfica na área de modelagem geométrica, abordando o paradigma de modelagem e as representações de curvas e superfícies utilizadas neste trabalho. Também são abordados algoritmos relacionados a construção e modificações de tais representações.

No Capítulo 3, são discutidos os algoritmos de geração de malhas de alta ordem desenvolvidos neste trabalho. Inicialmente, é feita uma revisão bibliográfica dos algoritmos clássicos de geração de malhas e dos algoritmos de geração de malhas de alta ordem relacionados a este trabalho. Em seguida, são apresentadas as técnicas de geração de malhas estruturadas e não-estruturadas desenvolvidas para problemas planos, assim como as métricas de qualidade de malhas adotadas no trabalho. A generalização destas técnicas para o caso de superfícies aparadas é apresentada em seguida. Por fim, são discutidos métodos de otimização de qualidade da malhas.

No Capítulo 4, são apresentadas as formulações isogeométricas utilizadas para solução dos problemas de engenharia tratados neste trabalho. Uma discussão sobre as formulações desenvolvidas neste trabalho e suas particularidades em relação a AIG clássica é realizada. Em seguida, é apresentada a formulação AIG para problemas planos de elasticidade, considerando modelo de estado plano de tensões, e do problema de transferência de calor. Na sequência, é abordado a formulação para análise estática e de vibrações livres de cascas.

No Capítulo 5, são apresentados exemplos de aplicação das técnicas de geração de malha e de análise isogeométrica propostas neste trabalho. Inicialmente, são avaliadas a eficácia e eficiência do algoritmo proposto para geração de malhas planas não estruturadas, comparando-o com alternativas da literatura e com malhas obtidas via otimização. A versão do algoritmo para geração de malhas de superfícies aparadas é avaliada na sequência. Em seguida, a precisão e a convergência dos elementos isogeométricos desenvolvidos neste trabalho são avaliadas em vários

problemas, incluindo *benchmarks* da literatura e problemas com solução analítica e manufaturada. Problemas em domínio de análise plano são apresentados primeiramente, e na sequência, são discutidos os exemplos de análise estática e de vibração livre de casca.

No Capítulo 6, são apresentadas as conclusões e os comentários finais do trabalho, bem como as sugestões para trabalhos futuros.

## 2 MODELAGEM GEOMÉTRICA

A Modelagem Geométrica é uma área da matemática aplicada que estuda a descrição de formas geométricas. Possui importante aplicação em sistemas de computação gráfica, como em sistemas CAD (*Computer Aided Design*). Estes são programas que auxiliam no projeto, na modificação e na otimização de modelos diversos, sendo utilizados em várias áreas, como na criação de filmes de animação, em jogos eletrônicos, e em projetos de diversos ramos da engenharia.

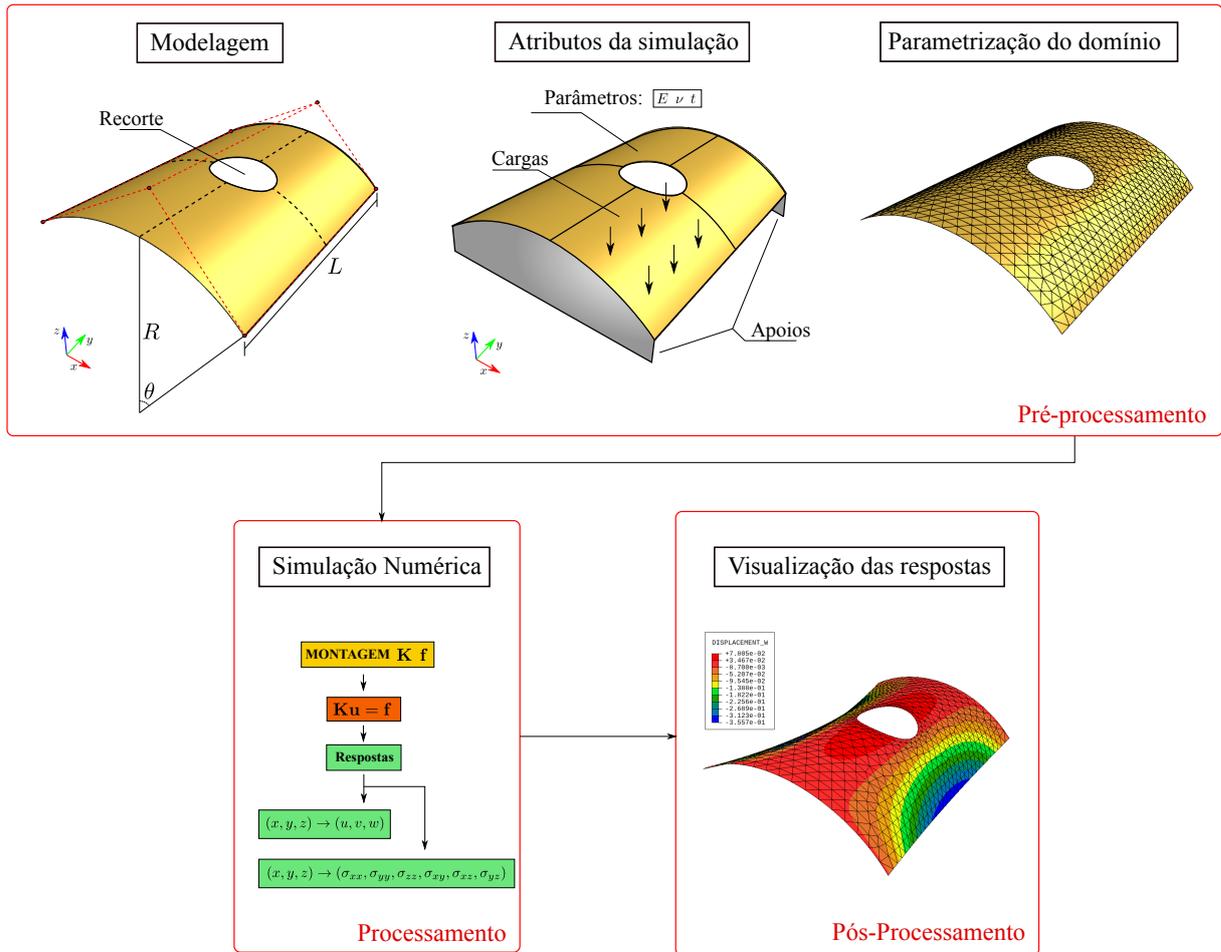
Os sistemas CAD utilizam a Modelagem Geométrica como base das representações envolvidas no desenho gráfico, tanto para modelos planos quanto para modelos tridimensionais. Na modelagem de sólidos, uma das representações mais usadas por boa parte dos sistemas CAD é a representação de fronteira ou B-Rep (*Boundary Representation*). Nessa representação, os objetos são representados pelo seu contorno, que consiste em um conjunto fechado de superfícies. No caso de modelos bidimensionais, os modelos B-Rep são definidos por um conjunto de curvas. Vale ressaltar que também existem sistemas gráficos baseados em outros paradigmas de modelagem<sup>11</sup>, como os modelos de decomposição espacial<sup>12</sup> e os modelos de construção<sup>13</sup>.

As representações matemáticas utilizadas por sistemas CAD foram desenvolvidas ao longo do tempo, principalmente no Século XX, com contribuições de diversos trabalhos. No área da modelagem de curvas e superfícies de forma livre, destacam-se as representações de Bézier<sup>14</sup>, Coons<sup>15</sup>, B-Splines<sup>16</sup>, NURBS<sup>17</sup> e T-Splines<sup>18, 19</sup>. As superfícies de subdivisão são superfícies suaves geradas a partir de malhas facetadas de triângulos ou quadriláteros, e são empregadas em várias aplicações, como, por exemplo, na área da animação computadorizada<sup>20, 21</sup>. Há também trabalhos relacionados a superfícies definidas em domínio triangulares, como os triângulos de Bézier<sup>14</sup> e generalizações triangulares de B-Splines<sup>22</sup>. Além destas, outras representações foram propostas recentemente, como as LR B-Splines<sup>23, 24, 25</sup>, PHT-Splines<sup>26, 27</sup> e THB-Splines<sup>28, 29</sup>. Vale destacar que, além de modelar geometrias complexas, como quádricas e superfície cúbicas, tais representações também são capazes de considerar furos e recortes utilizando curvas de aparo (*trimming curves*).

Do ponto de vista de aplicações ligadas à engenharia, os sistemas CAD são utilizados em conjunto com sistemas CAE (*Computer Aided Engineering*), que integram os modeladores geométricos a métodos computacionais, como o Método dos Elementos Finitos, para realização de simulações numéricas. Neste contexto, os sistemas CAD são empregados em uma etapa de pré-processamento, onde a geometria do problema é definida. A modelagem do problema

também inclui a definição de atributos ligados a aplicação numérica e a parametrização do domínio de análise. Esta última etapa é realizada por técnicas de geração de malhas, que são discutidas no Capítulo 3. As demais etapas realizadas nos sistemas CAE são o processamento, onde é realizada a análise numérica do problema em questão, e o pós-processamento, onde são visualizados os resultados obtidos. A Figura 1 ilustra as etapas descritas.

Figura 1 – Etapas realizadas em uma aplicação CAE.



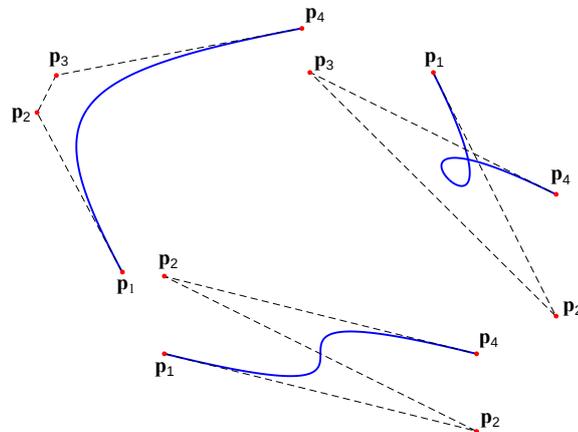
Fonte: Elaborada pelo autor.

Na análise isogeométrica, busca-se utilizar as mesmas representações empregadas no sistemas CAD em simulações numéricas, facilitando a integração entre a modelagem e a simulação dos problemas. No presente trabalho, foram desenvolvidas ferramentas para análise estrutural utilizando formulações isogeométricas em problemas onde a geometria é modelada por superfícies planas ou no espaço tridimensional. Logo, destaca-se que os conceitos de modelagem geométrica descritos neste capítulo são aplicados não apenas para descrição geométrica dos problemas, mas também para definição da forma dos elementos finitos utilizados e dos campos de resposta tratados na simulação.

## 2.1 Representações de curvas e superfícies de Bézier

A curva de Bézier foi criada pelo engenheiro francês Paul De Casteljaou, quando ainda era funcionário da Citroën, uma fabricante de automóveis, e foi popularizada pelo engenheiro francês Pierre Bézier. As primeiras publicações sobre essa representação foram apresentadas entre o final da década de 1950 e início da década de 1960<sup>30, 31</sup>. Embora tenha sido criada para projeto de automóveis, existem diversas aplicações da curva de Bézier em áreas relacionadas à computação gráfica, como em programas de manipulação de imagens (GIMP<sup>32</sup> e Photoshop<sup>33</sup>) e programas de desenho vetorial (Inkscape<sup>34</sup> e CorelDraw<sup>35</sup>).

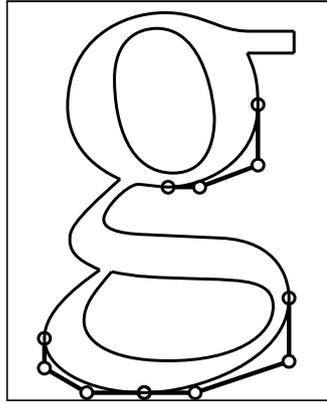
Figura 2 – Exemplo de curvas de Bézier cúbicas.



Fonte: Elaborada pelo autor.

A curva é definida em função de um conjunto de pontos discretos, denominados pontos de controle, e sua geometria pode ser facilmente alterada movendo os pontos de controle. A Figura 2 mostra exemplos de curvas de Bézier cúbicas (linha espessa) com o polígono formado pelos seus pontos de controle (linha tracejada). É importante notar que apenas os pontos de controle extremos pertencem à curva. Portanto, os pontos de controle não são pontos de interpolação. Pode-se notar que a curva é tangente ao polígono nas extremidades, possibilitando que geometrias complexas possam ser obtidas pela união de várias curvas mantendo a tangente contínua (continuidade  $C^1$ ). A Figura 3 mostra o contorno da letra "g" definido pela fonte PostScript com detalhes dos pontos de controle das curvas de Bézier utilizadas.

Figura 3 – Definição de contorno de fontes utilizando representação de Bézier.



Fonte: Sederberg (2012).

Uma curva de Bézier de grau  $p$  é construída por uma combinação linear dos pontos de controle  $\mathbf{p}_i$ :

$$\mathbf{c}(\xi) = \sum_{i=1}^{p+1} B_{i,p}(\xi) \mathbf{p}_i \quad (2.1)$$

onde  $B_{i,p}$  são os polinômios de Bernstein de grau  $p$  e  $\xi$  é a coordenada paramétrica. Esta expressão mostra que cada curva de grau  $p$  possui  $p + 1$  pontos de controle. Os polinômios de Bernstein<sup>36</sup> foram estudados por Bernstein para provar o Teorema de Stone-Weierstrass.

Considerando um intervalo paramétrico  $[0, 1]$ , as funções de base Bernstein podem ser calculadas pela expressão<sup>37</sup>:

$$B_{i,p}(\xi) = \binom{p}{i} (1 - \xi)^{p-i} \xi^i, \quad i = 0, 1, \dots, p$$

$$\text{com } \binom{p}{i} = \frac{p!}{i!(p-i)!} \quad (2.2)$$

Outra forma de cálculo dos polinômios de Bernstein é através da fórmula recursiva<sup>38</sup>:

$$B_{i,p}(\xi) = (1 - \xi)B_{i,p-1}(\xi) + \xi B_{i-1,p-1}(\xi) \quad (2.3)$$

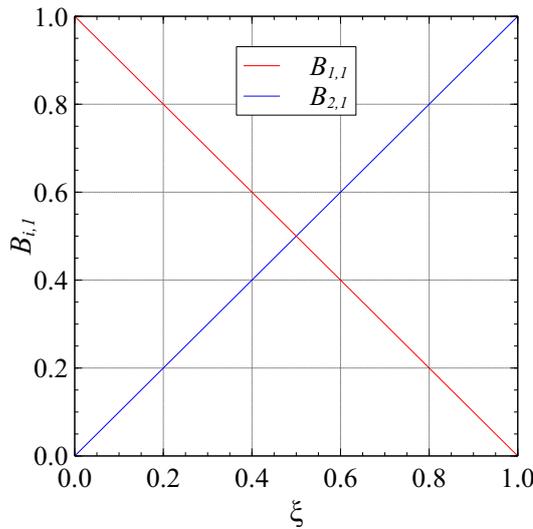
$$B_{i,p}(\xi) = 0 \quad \text{para } \xi < 0 \quad \text{ou } \xi > 1, \quad \text{e } B_{0,0}(\xi) = 1$$

A primeira derivada pode ser calculada por:

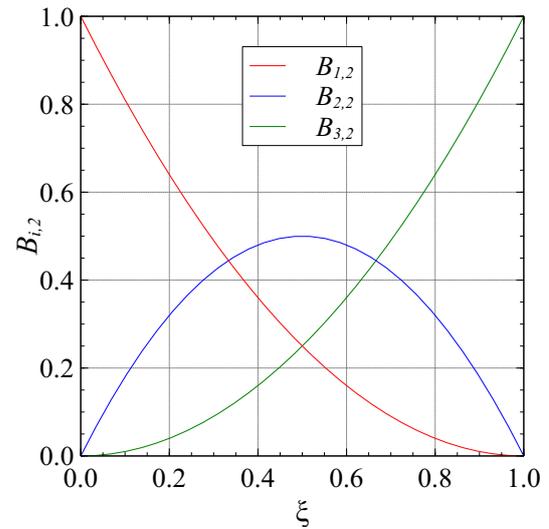
$$\frac{d}{d\xi} B_{i,p}(\xi) = p(B_{i-1,p-1}(\xi) - B_{i,p-1}(\xi)) \quad (2.4)$$

A Figura 4 mostra polinômios de Bernstein de diferentes graus.

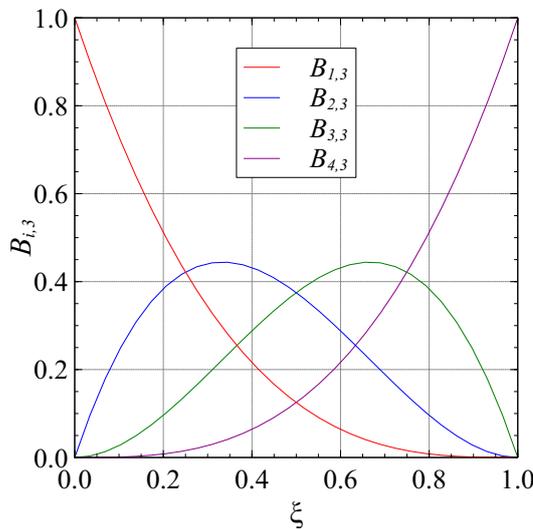
Figura 4 – Polinômios de Bernstein para vários graus.



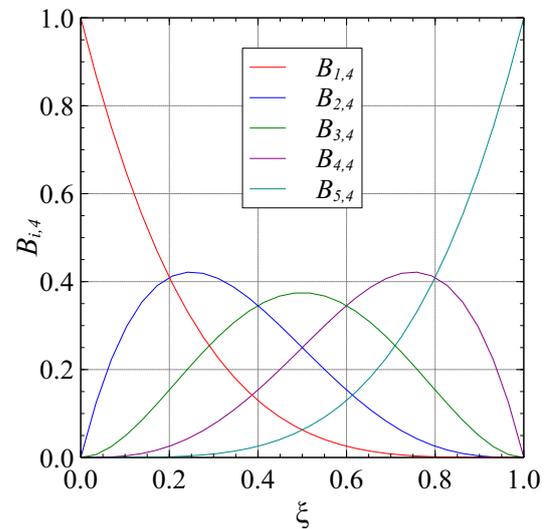
(a) Bases lineares.



(b) Bases quadráticas.



(c) Bases cúbicas.



(d) Bases quárticas.

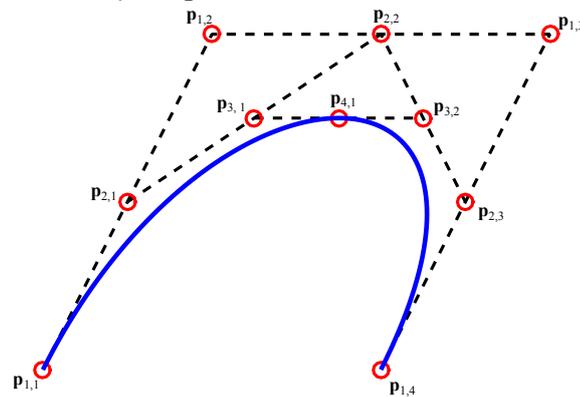
Fonte: Elaborada pelo autor.

Os polinômios de Bernstein  $B_{i,p}$  possuem importantes propriedades, como independência linear, não-negatividade ( $B_{i,p}(\xi) \geq 0$ ), partição da unidade ( $\sum_{i=1}^n B_{i,p}(\xi) = 1$ ) e simetria ( $B_{i,p}(\xi) = B_{p-i,p}(1 - \xi)$ ). Além disso, permitem o controle pseudo-localizado sobre a forma de curvas e superfícies de Bézier. Se um ponto de controle de uma representação de Bézier for movido, então as mudanças da forma da representação ocorrerão de forma dominante na vizinhança deste.

Uma maneira mais eficiente e numericamente estável de calcular os pontos da curva de Bézier do que pela Equação (2.1) é através do algoritmo de *de Casteljau*<sup>38</sup>. Neste algoritmo, cada segmento do polígono de controle é dividido de acordo com o ponto paramétrico

$\xi$  calculado, sendo traçadas novas retas com os pontos obtidos, reduzindo em um o número de retas. O processo é repetido até que apenas uma reta seja desenhada, e o ponto paramétrico  $\xi$  desta reta resultante é o valor da curva de Bézier. O processo é ilustrado na Figura 5. Uma abordagem mais detalhada do algoritmo pode ser encontrada na literatura<sup>38, 37</sup>.

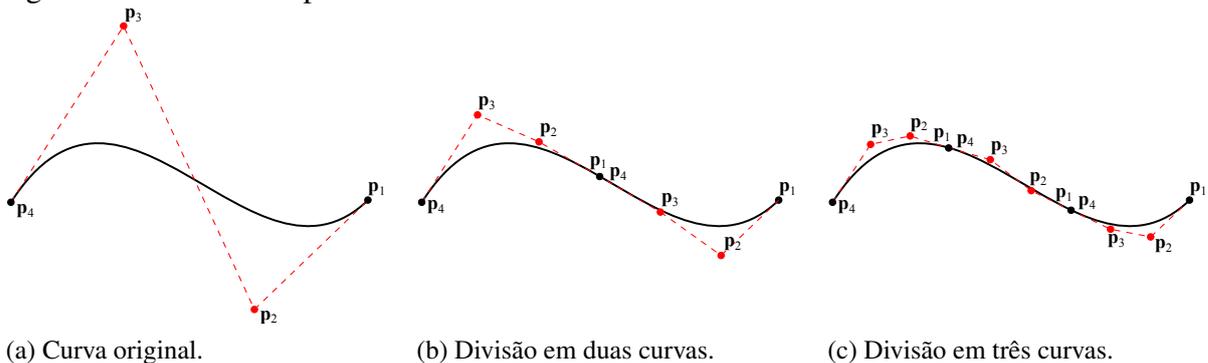
Figura 5 – Algoritmo de *de Casteljau* aplicado a curva de Bézier cúbica.



Fonte: Elaborada pelo autor.

A *subdivisão* é um procedimento que divide uma curva de Bézier em uma sequência de curvas de Bézier de mesmo grau conectadas nos extremos, de modo a preservar a geometria da curva original, como ilustrado na Figura 6. O procedimento também pode se aplicado em superfícies e triângulos de Bézier. Os algoritmos da *subdivisão* são apresentados na literatura<sup>38, 14</sup>.

Figura 6 – Subdivisão aplicada a uma curva de Bézier cúbica.



(a) Curva original.

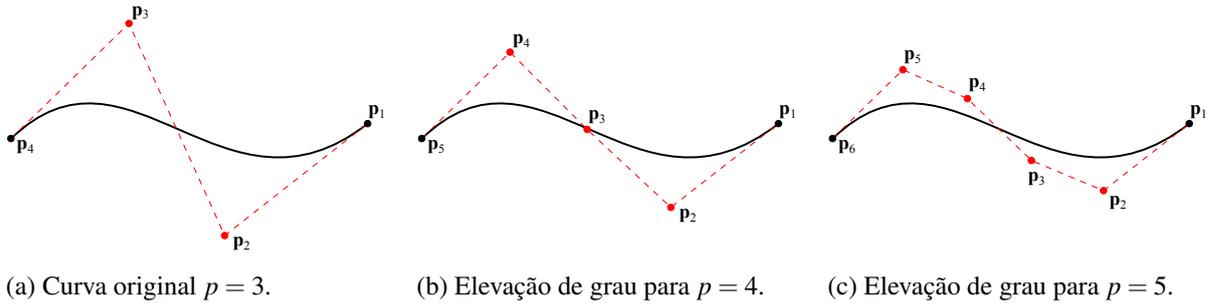
(b) Divisão em duas curvas.

(c) Divisão em três curvas.

Fonte: Elaborada pelo autor.

A *elevação de grau* é um procedimento que aumenta o grau de uma curva de Bézier, preservando sua geometria e parametrização. O mesmo procedimento existe também para superfícies e triângulos de Bézier. A Figura 7 mostra a elevação de grau sendo aplicada a uma curva de Bézier quadrática. Os algoritmos de *elevação de grau* para as curvas, superfícies e triângulos de Bézier são detalhados na literatura<sup>38, 14</sup>.

Figura 7 – Elevação de grau aplicada a uma curva de Bézier cúbica.



Fonte: Elaborada pelo autor.

Para a Análise Isogeométrica, a *elevação de grau* melhora a solução numérica, uma vez que se utiliza funções de forma de ordem maior, enquanto que a *subdivisão* melhora a solução numérica por aumentar o número de funções de aproximação utilizadas no modelo numérico.

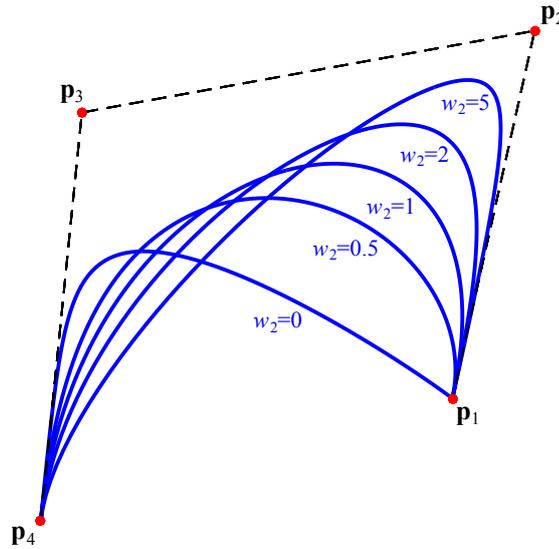
### 2.1.1 Bézier Racional

Uma das desvantagens da curva de Bézier apresentada anteriormente é que esta não é capaz de representar curvas cônicas, como circunferências e elipses, de forma exata, pois estas curvas não são descritas por funções polinomiais. Contudo, a representação exata é possível utilizando funções racionais. Para este fim são acrescentados pesos aos pontos de controle da curva. Desta forma, a curva de Bézier passa a ser calculada pela expressão:

$$\mathbf{c}(\xi) = \frac{\sum_{i=1}^{p+1} w_i B_{i,p}(\xi) \mathbf{p}_i}{\sum_{i=1}^{p+1} w_i \hat{B}_{i,p}(\xi)} \quad (2.5)$$

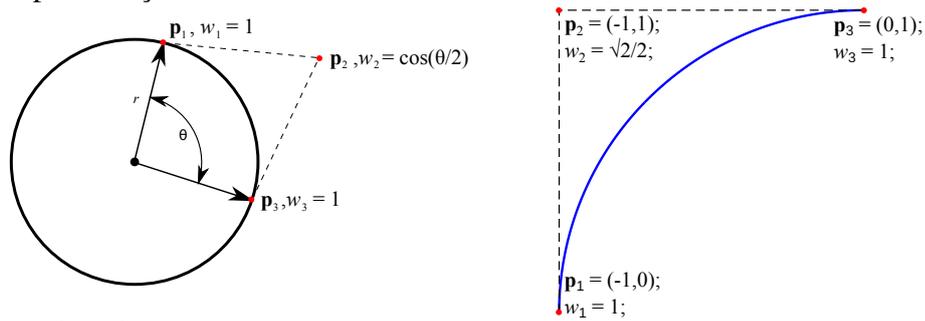
onde  $w_i$  é o peso associado a cada ponto de controle  $\mathbf{p}_i$ . O efeito da variação do peso pode ser observado na Figure 8, onde o peso do ponto  $\mathbf{p}_2$  é modificado entre os valores de 0 e 5. Observe que caso todos os pesos sejam iguais a 1, a equação racional se torna igual a equação tradicional. Um caso particular interessante de uma curva de Bézier racional é o arco de circunferência. Este pode ser obtido utilizando os pontos de controle e pesos ilustrados na Figura 9.

Figura 8 – Efeito da variação do peso.



Fonte: Elaborada pelo autor.

Figura 9 – Representação de um arco de círculo utilizando Bézier Racional.



Fonte: Elaborada pelo autor.

### 2.1.2 Superfícies de Bézier

Uma superfície de Bézier construída por produto tensorial é obtida a partir do produto de dois polinômios de Bernstein univariantes. Considerando uma superfície de Bézier racional de grau  $p$  na direção  $\xi$  e  $q$  na direção  $\eta$ , esta pode ser calculada por:

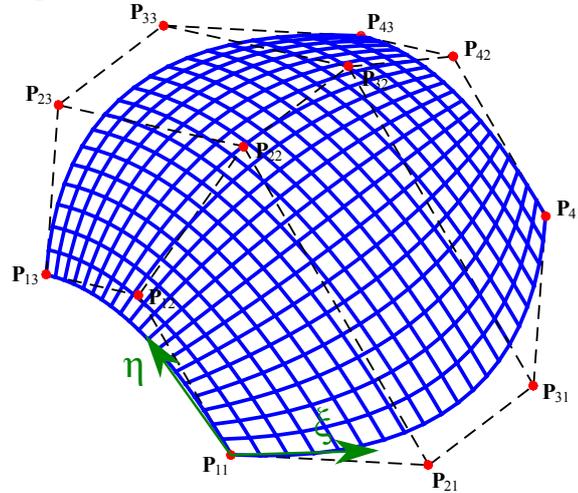
$$\mathbf{s}(\xi, \eta) = \frac{\sum_{i=1}^{p+1} \sum_{j=1}^{q+1} w_{ij} B_{i,p}(\xi) B_{j,q}(\eta) \mathbf{P}_{ij}}{\sum_{\hat{i}=1}^{p+1} \sum_{\hat{j}=1}^{q+1} w_{\hat{i}\hat{j}} B_{\hat{i},p}(\xi) B_{\hat{j},q}(\eta)} \quad (2.6)$$

onde  $B_{i,p}(\xi)$  e  $B_{j,q}(\eta)$  são os polinômios de Bernstein univariantes e  $\mathbf{P}$  é uma matriz de pontos de controle de tamanho  $(q+1) \times (p+1)$ .

É importante notar que a Equação (2.6) considera o caso geral de superfícies racionais. O caso particular de superfícies não-rationais é obtido quando todos os pesos são iguais a 1. A

Figura 10 mostra uma de superfície de Bézier de grau  $3 \times 2$ .

Figura 10 – Exemplo de superfície Bézier.



Fonte: Elaborada pelo autor.

A Equação (2.6) pode ser escrita considerando as bases racionais:

$$\mathbf{s}(\xi, \eta) = \sum_{i=1}^{p+1} \sum_{j=1}^{q+1} R(\xi, \eta)_{ij} \mathbf{P}_{ij} \quad (2.7)$$

onde  $R(\xi, \eta)_{ij}$  é a função de base racional associada ao ponto de controle  $\mathbf{P}_{ij}$ , e dada por:

$$R(\xi, \eta)_{ij} = \frac{w_{ij} B_{i,p}(\xi) B_{j,q}(\eta)}{W(\xi, \eta)}, \quad (2.8)$$

e  $W(\xi, \eta)$  é a função peso:

$$W(\xi, \eta) = \sum_{\hat{i}=1}^{p+1} \sum_{\hat{j}=1}^{q+1} w_{\hat{i}\hat{j}} B_{\hat{i},p}(\xi) B_{\hat{j},q}(\eta) \quad (2.9)$$

As derivadas parciais das bases racionais são necessárias na formulação de elementos isogeométricos, como na montagem da matriz de rigidez em problemas de elasticidade ou da matriz de condutividade térmica em problemas de transferência de calor. As derivadas parciais de  $W(\xi, \eta)$  são expressas por:

$$\begin{aligned} \frac{\partial}{\partial \xi} W(\xi, \eta) &= \sum_{\hat{i}=1}^{p+1} \sum_{\hat{j}=1}^{q+1} w_{\hat{i}\hat{j}} \frac{\partial}{\partial \xi} B_{\hat{i},p}(\xi) B_{\hat{j},q}(\eta) \\ \frac{\partial}{\partial \eta} W(\xi, \eta) &= \sum_{\hat{i}=1}^{p+1} \sum_{\hat{j}=1}^{q+1} w_{\hat{i}\hat{j}} B_{\hat{i},p}(\xi) \frac{\partial}{\partial \eta} B_{\hat{j},q}(\eta) \end{aligned} \quad (2.10)$$

Assim, as derivadas parciais das funções de base são calculadas por:

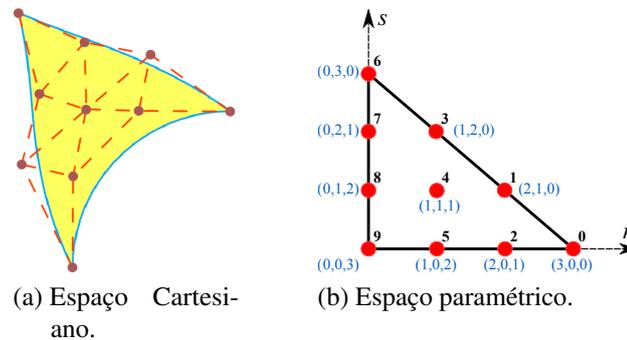
$$\frac{\partial}{\partial \xi} R_{ij}(\xi, \eta) = w_{ij} \frac{W(\xi, \eta) \frac{\partial}{\partial \xi} B_{i,p}(\xi) B_{j,q}(\eta) - \frac{\partial}{\partial \xi} W(\xi, \eta) B_{i,p}(\xi) B_{j,q}(\eta)}{W^2(\xi, \eta)} \quad (2.11)$$

$$\frac{\partial}{\partial \eta} R_{ij}(\xi, \eta) = w_{ij} \frac{W(\xi, \eta) B_{i,p}(\xi) \frac{\partial}{\partial \eta} B_{j,q}(\eta) - \frac{\partial}{\partial \eta} W(\xi, \eta) B_{i,p}(\xi) B_{j,q}(\eta)}{W^2(\xi, \eta)}$$

## 2.2 Triângulos de Bézier

Os triângulos de Bézier são superfícies bivariadas diferentes das superfícies de Bézier construídas por produto tensorial. Nos triângulos de Bézier, a malha de controle é definida por uma estrutura triangular, como ilustrado na Figura 11. Os pontos da superfície são avaliados em função de coordenadas baricêntricas  $\lambda = (\lambda_1, \lambda_2, \lambda_3)$ . Os pontos  $\mathbf{t}(1, 0, 0)$ ,  $\mathbf{t}(0, 1, 0)$ ,  $\mathbf{t}(0, 0, 1)$  da superfície são pontos extremos, vértices do triângulo. As arestas do triângulo de Bézier definem curvas de Bézier, portanto os pontos de controle extremos são interpoladores.

Figura 11 – Triângulo de Bézier cúbico.



Fonte: Elaborada pelo autor.

Cada ponto de controle possui três índices não negativos, cuja soma é igual ao grau da superfície. Este índice triplo é apresentado na Figura 11b em azul. Logo, para um ponto  $\mathbf{p}_{ijk}$ , os índices  $i, j, k$  são tais que  $i + j + k = p$ , onde  $p$  é o grau da superfície. Uma notação utilizada por Farin<sup>14</sup> abrevia o termo  $i, j, k$  para  $\mathbf{i}$ , e a condição  $i + j + k = p$  para  $|\mathbf{i}| = p$ . Também são utilizadas as abreviações  $\mathbf{e}_1 = (1, 0, 0)$ ,  $\mathbf{e}_2 = (0, 1, 0)$ ,  $\mathbf{e}_3 = (0, 0, 1)$  e  $\mathbf{0} = (0, 0, 0)$ . Assim, o número de pontos de controle  $N_{cp}$  para uma dada superfície de grau  $p$  é dada por:

$$N_{cp} = \frac{(p+1)(p+2)}{2}. \quad (2.12)$$

Uma maneira eficiente de avaliar um triângulo de Bézier é utilizar o algoritmo de *de*

*Casteljau*, que funciona pela aplicação repetida da formula recursiva dada por<sup>14</sup>:

$$b_{\mathbf{i}}^r(\lambda) = \lambda_1 b_{\mathbf{i}+\mathbf{e}_1}^{r-1}(\lambda) + \lambda_2 b_{\mathbf{i}+\mathbf{e}_2}^{r-1}(\lambda) + \lambda_3 b_{\mathbf{i}+\mathbf{e}_3}^{r-1}(\lambda), \quad (2.13)$$

onde  $b_{\mathbf{i}}^0(\lambda) = \mathbf{p}_{\mathbf{i}}$ . Considerando uma superfície de grau  $p$ , o ponto de coordenada baricêntrica  $\lambda$  é calculado por  $b_{\mathbf{0}}^p(\lambda)$  na expressão recursiva.

Uma maneira alternativa de avaliação da superfície é utilizando os polinômios de Bernstein bivariados ( $\hat{B}_{\mathbf{i}}^p$ ), por, i.e.,

$$\mathbf{t}(\lambda) = \sum_{|\mathbf{i}|=p} \hat{B}_{\mathbf{i}}^p(\lambda) \mathbf{p}_{\mathbf{i}}. \quad (2.14)$$

Nesta equação, o somatório é definido ao longo de todos os índices  $\mathbf{i}$  possíveis para o grau  $p$ , sendo o número total de índices dado pela Equação (2.12). Pode-se definir um índice único para cada índice triplo  $i, j, k$ , como ilustrado na Figura 11b, em cor preta. Tais índices são definidos priorizando as variações dos índices  $k, j$  e  $i$ , respectivamente, e atribuindo à primeira posição ao ponto  $\mathbf{p}_0 = \mathbf{p}_{p,0,0}$  (o mais à direita). De fato, a numeração dos pontos de controle é arbitrária, entretanto, ao utilizar a numeração proposta é possível converter o índice triplo  $(i, j, k)$  para o índice único  $a$  por:

$$a = p - i - j + 1 + (p - i + 1)(p - i)/2. \quad (2.15)$$

Assim, pode-se alocar as bases em um vetor de tamanho  $N_{cp}$  e avaliar a superfície pela expressão:

$$\mathbf{t}(\lambda) = \sum_{a=0}^{N_{cp}-1} \hat{B}_a^p(\lambda) \mathbf{p}_a. \quad (2.16)$$

Os polinômios bivariados de Bernstein são definidos por:

$$\hat{B}_{\mathbf{i}}^p(\lambda) = \frac{p!}{i!j!k!} \lambda_1^i \lambda_2^j \lambda_3^k. \quad (2.17)$$

Além disto, as bases  $\hat{B}_{i,j,k}^p$  satisfazem a recursão:

$$\hat{B}_{\mathbf{i}}^p(\lambda) = \lambda_1 \hat{B}_{\mathbf{i}-\mathbf{e}_1}^{p-1}(\lambda) + \lambda_2 \hat{B}_{\mathbf{i}-\mathbf{e}_2}^{p-1}(\lambda) + \lambda_3 \hat{B}_{\mathbf{i}-\mathbf{e}_3}^{p-1}(\lambda); \quad |\mathbf{i}| = p, \quad (2.18)$$

que assume  $B_{0,0,0}^0 = 1$  e  $i, j, k < 0 \implies B_{i,j,k}^p = 0$ . Tal definição recursiva é similar ao algoritmo de *de Casteljau* para avaliação dos triângulos de Bézier, que é numericamente estável<sup>14, 39</sup>.

O conceito de *programação dinâmica* é usado na definição de um algoritmo eficiente para avaliação de todas as funções de base, de acordo com a Equação (2.18), evitando desperdício

de recursos. O esquema de indexação do *array* usado para armazenamento das funções de base é o mesmo ilustrado na Figura 11 b. Durante a avaliação da base  $\hat{B}_{i,j,k}^p$ , os termos  $\hat{B}_{i-1,j,k}^{p-1}$ ,  $\hat{B}_{i,j-1,k}^{p-1}$  e  $\hat{B}_{i,j,k-1}^{p-1}$  zeram, respectivamente, quando  $i = 0$ ,  $j = 0$  e  $k = 0$ . Esses termos estão localizados na  $i$ -ésima coluna, onde  $\hat{B}_{i-1,j,k}^{p-1}$  está na mesma posição do elemento atual, e  $i + 1$ , onde  $\hat{B}_{i,j,k-1}^{p-1}$  e  $\hat{B}_{i,j-1,k}^{p-1}$  estão à direita do elemento. Assim, para cada grau intermediário, avaliar as bases na ordem reversa ( $i$  de 0 para 1,  $a$  de  $N_{cp} - 1$  para 0) não requer o uso de memória auxiliar. Tais considerações guiam a elaboração do algoritmo para avaliação das funções de base apresentado na Figura 12. A entrada é o grau do triângulo  $p$ , as coordenada baricêntricas  $r$  e  $s$  ( $r = \xi$  e  $s = \eta$ ), e a saída são as bases armazenadas no *array* **b**.

Figura 12 – Código da avaliação das funções de base do triângulo de Bézier.

```

1 Basis (p, r, s, b)
2 {
3     t      = 1 - r - s;
4     b[0]  = 1;
5     for(int d = 1; d <= p; d++)
6     {
7         n = (d+1)*(d+2)/2;
8
9         // Avaliação das bases na coluna i = 0.
10        b[n-d-1] = s * b[n-2*d-1];
11        for(q = n-d; q < n-1; ++q)
12            b[q] = s * b[q-d] + t * b[q-d-1];
13        b[n-1]   = t * b[n-d-2];
14
15        // Percorre cada coluna i.
16        for(q = n-2-d, i = 1; i < d; ++i, --q)
17        {
18            b[q] = r * b[q] + t * b[q-d+i-1];
19            for(j = 1, --q; j < d-i; ++j, --q)
20                b[q] = r * b[q] + s * b[q-d+i] +
21                    t * b[q-d+i-1];
22            b[q] = r * b[q] + s * b[q-d+i];
23        }
24        b[0] = r * b[0];
25    }
26 }

```

Fonte: Elaborada pelo autor.

Os polinômios de Bernstein bivariados também possuem as propriedades de independência linear, não-negatividade e partição de unidade<sup>14</sup>.

A primeira derivada parcial dos polinômio de Bernstein bivariados em relação a  $\xi$  é obtida por:

$$\frac{\partial \hat{B}_i^p}{\partial \xi}(\lambda) = \frac{p!}{i!j!k!} i \xi^{i-1} \eta^j (1 - \xi - \eta)^k - \frac{p!}{i!j!k!} \xi^i \eta^j k (1 - \xi - \eta)^{k-1}.$$

Simplificando e colocando  $p$  em evidência, obtemos:

$$\frac{\partial \hat{B}_{\mathbf{i}}^p}{\partial \xi}(\lambda) = p \left( \frac{(p-1)!}{(i-1)!j!k!} \xi^{i-1} \eta^j (1-\xi-\eta)^k - \frac{(p-1)!}{i!j!(k-1)!} \xi^i \eta^j (1-\xi-\eta)^{k-1} \right).$$

Pode-se verificar que os termos  $\frac{(p-1)!}{(i-1)!j!k!} \xi^{i-1} \eta^j (1-\xi-\eta)^k = \hat{B}_{\mathbf{i}-\mathbf{e}_1}^{p-1}$  e  $\frac{(p-1)!}{i!j!(k-1)!} \xi^i \eta^j (1-\xi-\eta)^{k-1} = \hat{B}_{\mathbf{i}-\mathbf{e}_3}^{p-1}$ . Assim:

$$\frac{\partial}{\partial \xi} \hat{B}_{\mathbf{i}}^p(\lambda) = p(\hat{B}_{\mathbf{i}-\mathbf{e}_1}^{p-1} - \hat{B}_{\mathbf{i}-\mathbf{e}_3}^{p-1}). \quad (2.19)$$

De modo análogo, a primeira derivada parcial em relação a  $\eta$  é dada por:

$$\frac{\partial}{\partial \eta} \hat{B}_{\mathbf{i}}^p(\lambda) = p(\hat{B}_{\mathbf{i}-\mathbf{e}_2}^{p-1} - \hat{B}_{\mathbf{i}-\mathbf{e}_3}^{p-1}). \quad (2.20)$$

Logo, as derivadas parciais podem ser calculadas utilizando a Equação 2.18 para avaliar os polinômios  $\hat{B}_{\mathbf{i}}^{p-1}$ , e, em seguida, todas as derivadas parciais.

O algoritmo para avaliação das primeiras derivadas dos polinômios de Bernstein bivariados é apresentado na Figure 13, onde as derivadas são armazenadas nos *arrays*  $dr$  e  $ds$  (com  $dr = d\xi$  e  $ds = d\eta$ ). As mesmas considerações usadas para construção do algoritmo de avaliação das funções base são usadas neste algoritmo. Vale notar que a mesma memória usada para armazenar as derivadas parciais na direção  $s$  ( $ds$ ) pode ser usada para armazenar as funções de base  $\mathbf{b}$ .

Figura 13 – Código da avaliação da primeira derivada dos polinômios de Bernstein bivariados.

```

1 FirstDerv(p,r,s,dr,ds)
2 {
3 // Avalia as p-1 funções de base.
4 Basis(p-1,r,s,b);
5 n = (p+1)*(p+2)/2;
6
7 // Avalia as derivadas na coluna i = 0.
8 dr[n-p-1] = 0.0;
9 ds[n-p-1] = p * b[n-2*p-1];
10 for(q = n-p; q < n-1; ++q)
11 {
12     dr[q] = -p * b[q-p-1];
13     ds[q] = p * (b[q-p] - b[q-p-1]);
14 }
15 dr[n-1] = ds[n-1] = -p * b[n-p-2];
16
17 // Percorre cada coluna i.
18 for(q = n-2-p, i = 1; i < p; ++i, --q)
19 {
20     dr[q] = p * (b[q] - b[q-p+i-1]);
21     ds[q] = -p * b[q-p+i-1];
22     for(j = 1, --q; j < (p-i); ++j, --q)
23     {
24         dr[q] = p * (b[q] - b[q-p+i-1]);
25         ds[q] = p * (b[q-p+i] - b[q-p+i-1]);
26     }
27     dr[q] = p * b[q];
28     ds[q] = p * b[q-p+i];
29 }
30 dr[0] = p * b[0];
31 ds[0] = 0;
32 }

```

Fonte: Elaborada pelo autor.

Finalmente, é importante notar que triângulos de Bézier podem representar superfícies de Bézier quadrilaterais de forma exata. Goldman e Filip<sup>40</sup> desenvolveram uma fórmula numericamente estável para representar uma superfície de Bézier de grau  $p \times q$  usando dois triângulos de Bézier de grau  $(p+q)$ . Vale notar que o mesmo se estende para superfícies NURBS, uma vez que estas podem ser representadas por um conjunto de superfícies de Bézier<sup>6</sup>.

### 2.2.1 Bases racionais

A versão racional do triângulo de Bézier pode ser obtida considerando pesos em cada ponto de controle, de modo análogo às superfícies de Bézier racionais. Assim, a base Bernstein

bivariada racional é definida por:

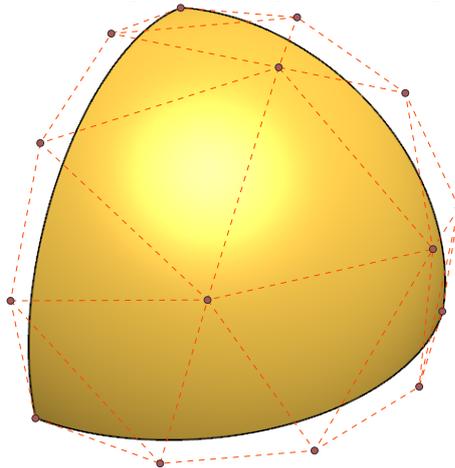
$$\hat{R}_a^p(\lambda) = \frac{\hat{B}_a^p(\lambda) w_a}{\sum_{\hat{a}=0}^{N_{cp}} \hat{B}_{\hat{a}}^p(\lambda) w_{\hat{a}}}, \quad (2.21)$$

e o triângulo de Bézier racional é obtido por:

$$\mathbf{t}(\lambda) = \sum_{a=0}^{N_{cp}} \hat{R}_a^p(\lambda) \mathbf{p}_a. \quad (2.22)$$

Os triângulo de Bézier racionais são importantes pois permitem representar quádricas exatamente, como o octante de uma esfera ilustrado na Figura 14. Pode-se notar que as arestas curvas do octante são quartos de circunferência definidas por curvas de Bézier racionais de quarto grau.

Figura 14 – Um oitavo de uma esfera representado por um triângulo de Bézier racional quártico.



Fonte: Elaborada pelo autor.

As derivadas parciais das bases racionais são avaliadas utilizando a regra de derivação do quociente:

$$\frac{\partial}{\partial \xi} \hat{R}_a^p(\lambda) = w_a \frac{\hat{W}(\lambda) \frac{\partial}{\partial \xi} \hat{B}_a^p(\lambda) - \frac{\partial}{\partial \xi} \hat{W}(\lambda) \hat{B}_a^p(\lambda)}{\hat{W}^2(\lambda)}, \quad (2.23)$$

$$\frac{\partial}{\partial \eta} \hat{R}_a^p(\lambda) = w_a \frac{\hat{W}(\lambda) \frac{\partial}{\partial \eta} \hat{B}_a^p(\lambda) - \frac{\partial}{\partial \eta} \hat{W}(\lambda) \hat{B}_a^p(\lambda)}{\hat{W}^2(\lambda)},$$

onde  $\hat{W}$  é a função peso, dada por:

$$\hat{W}(\lambda) = \sum_{a=0}^{N_{cp}} \hat{B}_a^p(\lambda) w_a, \quad (2.24)$$

e as derivadas parciais da função peso são avaliadas por:

$$\begin{aligned}\frac{\partial}{\partial \xi} \hat{W}(\lambda) &= \sum_{a=0}^{N_{cp}} \frac{\partial}{\partial \xi} \hat{B}_a^p(\lambda) w_a, \\ \frac{\partial}{\partial \eta} \hat{W}(\lambda) &= \sum_{a=0}^{N_{cp}} \frac{\partial}{\partial \eta} \hat{B}_a^p(\lambda) w_a.\end{aligned}\tag{2.25}$$

Novamente, destaca-se que estas derivadas são utilizadas na formulação dos elementos isogeométricos tratados neste trabalho.

### 2.3 B-Splines

O termo spline se refere a uma ferramenta de desenho utilizada antigamente nos desenhos de peças mecânicas e nos projetos de embarcações<sup>41</sup>. A ferramenta possibilitava, de maneira prática, o traçado de curvas suaves sobre a prancha de desenho. As B-Splines são curvas capazes de descrever vários segmentos distintos ao longo da mesma representação paramétrica. Esta característica é obtida limitando a atuação das funções base a regiões do espaço paramétrico, conhecidas na literatura como *knot spans*, que são definidas por um vetor de valores paramétricos, o vetor de *knots* (a tradução de *knot* é nó, termo relacionado aos laços utilizados por marinheiros, porém para não confundir com os nós de elementos finitos, será considerado o termo em inglês). As B-Splines podem descrever qualquer representação de Bézier não racional. Existem várias aplicações que utilizam B-Splines, como em programas de modelagem tridimensional como Rhinoceros<sup>42</sup>, programas de animação como Maya<sup>43</sup> e sistemas CAD como AutoCAD<sup>44</sup>.

Uma curva B-Spline é construída por uma combinação linear dos pontos de controle  $\mathbf{p}_i$ . As funções de base  $N_{i,p}(\xi)$  definem os coeficientes de combinação linear para cada  $\xi$ , como mostra a expressão:

$$\mathbf{c}(\xi) = \sum_{i=1}^n N_{i,p}(\xi) \mathbf{p}_i,\tag{2.26}$$

onde  $n$  é o número de funções de base B-Splines e  $p$  é o grau da curva.

As bases B-Splines requerem um vetor de *knots*, que consiste num conjunto de valores paramétricos não-negativos e não-decrescentes delimitados ao longo do intervalo paramétrico  $[\xi_1, \xi_{n+p+1}]$  no qual a curva foi definida. Um vetor de *knots* é dito uniforme se os valores paramétricos contidos neste variam segundo um mesmo fator. Por exemplo, os vetores  $\Xi = [0, 0, 0, 1/3, 2/3, 1, 1, 1]$  e  $\Xi = [0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4]$  são uniformes, já os vetores  $\Xi = [0, 0, 0, 0.25, 0.75, 1, 1, 1]$  e  $\Xi = [0, 0, 0, 2, 2, 3, 3, 6, 6, 6, 6]$  não são.

Considerando o vetor de *knots*  $\Xi = [\xi_1, \xi_2, \dots, \xi_{n+p+1}]$ , as funções de base B-Spline são definidas pela fórmula recursiva de Cox-de Boor<sup>38</sup>:

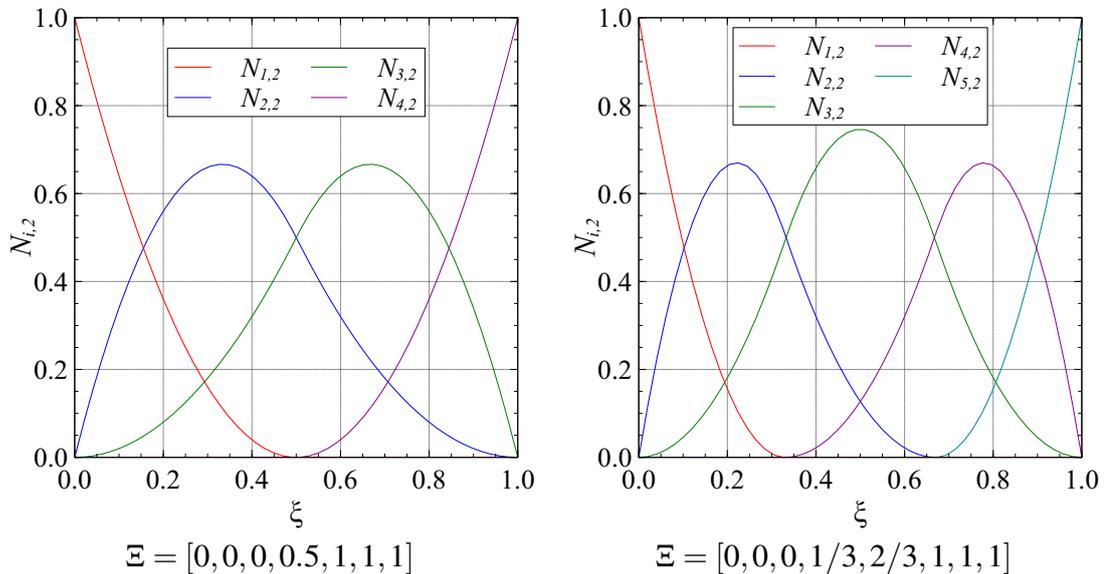
$$N_{i,0}(\xi) = \begin{cases} 1, & \xi_i \leq \xi < \xi_{i+1} \\ 0, & \text{caso contrário} \end{cases}$$

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \quad (2.27)$$

Assim, cada base  $N_{i,p}$  é não nula apenas no intervalo paramétrico  $[\xi_i, \xi_{i+p+1}]$ .

Considerando um exemplo de uma base quadrática com vetor de *knots*  $\Xi = [0,0,0,0.5, 1,1,1]$ , onde as bases estão apresentadas na Figura 15, observa-se que a função  $N_{1,2}$  contribui ao longo do intervalo  $[\xi_1, \xi_4] = [0,0.5]$ . A Figura 15 também mostra as bases B-Spline quadráticas considerando um vetor de *knots* mais discretizado  $\Xi = [0,0,0,0.5, 1, 1, 1]$ . Um exemplo similar mostrando as bases de uma B-Spline cúbica está apresentado na Figura 16. O intervalo de contribuição de cada base mostrada nas duas figuras estão apresentados na Tabela 1.

Figura 15 – Funções base B-Spline quadráticas.



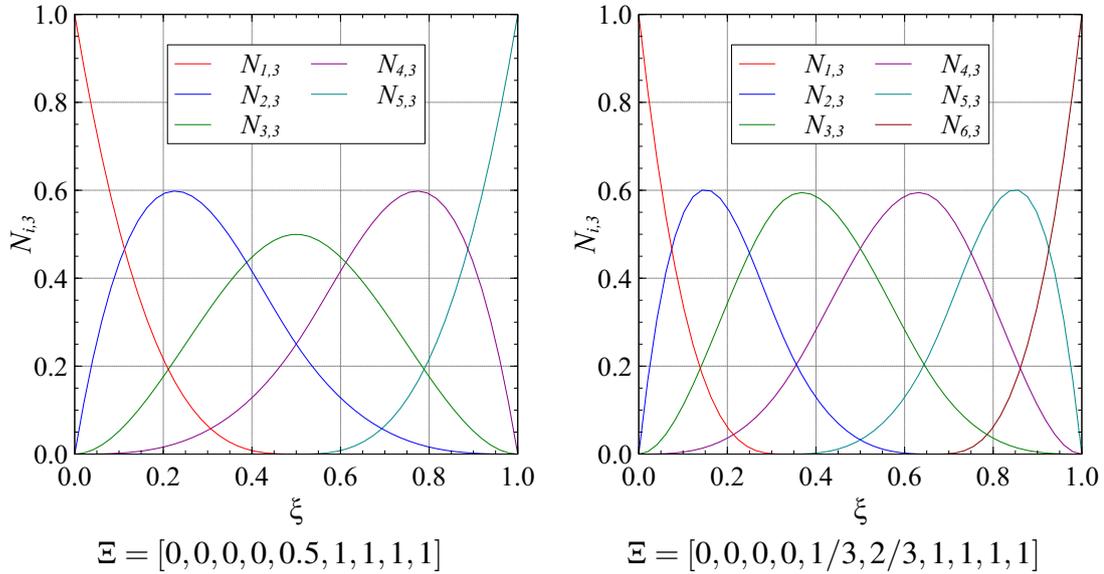
Fonte: Elaborada pelo autor.

Pode-se observar nos exemplos que o número de bases aumenta a medida que são utilizados vetores de *knots* maiores. O número de bases  $n$  pode ser calculado em função do tamanho do vetor de *knots* ( $ks$ ) e do grau das bases ( $p$ ):

$$n = ks - p - 1 \quad (2.28)$$

Outra observação importante é que o número de bases não nulas em cada *knot span* é sempre igual a  $p+1$ . Esta propriedade permite que as B-Splines sejam avaliadas de forma bastante eficiente.

Figura 16 – Funções de base B-Spline cúbica.



Fonte: Elaborada pelo autor.

Tabela 1 – Intervalo de contribuição das bases B-Spline para diferentes vetores  $\Xi$ .

$N_{i,2}$	B-Spline quadrática		$N_{i,3}$	B-Spline cúbica	
	$[0, 0, 0, 0.5, 1, 1, 1]$	$[0, 0, 0, 1/3, 2/3, 1, 1, 1]$		$[0, 0, 0, 0.5, 1, 1, 1, 1]$	$[0, 0, 0, 0, 1/3, 2/3, 1, 1, 1, 1]$
$N_{1,2}$	$[0.0, 0.5]$	$[0.00, 1/3]$	$N_{1,3}$	$[0.0, 0.5]$	$[0.00, 1/3]$
$N_{2,2}$	$[0.0, 1.0]$	$[0.00, 2/3]$	$N_{2,3}$	$[0.0, 1.0]$	$[0.00, 2/3]$
$N_{3,2}$	$[0.0, 1.0]$	$[0.00, 1.00]$	$N_{3,3}$	$[0.0, 1.0]$	$[0.00, 1.00]$
$N_{4,2}$	$[0.5, 1.0]$	$[1/3, 1.00]$	$N_{4,3}$	$[0.0, 1.0]$	$[0.00, 1.00]$
$N_{5,2}$		$[2/3, 1.00]$	$N_{5,3}$	$[0.5, 1.0]$	$[1/3, 1.00]$
			$N_{6,3}$		$[2/3, 1.00]$

Fonte: Elaborada pelo autor.

As derivadas de primeira ordem das funções base podem ser calculadas por:

$$\frac{d}{d\xi} N_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi). \quad (2.29)$$

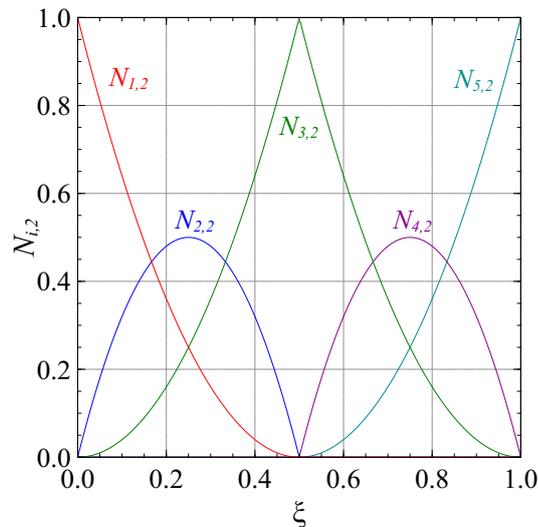
A expressão para cálculo de derivadas de ordem superior pode ser encontrada na literatura<sup>38</sup>.

Os valores paramétricos no interior dos vetores de *knots* podem aparecer repetidamente, sendo o número de repetições que um dado valor paramétrico  $\xi_i$  possui conhecido como a multiplicidade do *knot*. Uma das propriedades das B-Splines é possuir continuidade  $C^{p-1}$ . Em caso de algum *knot* ter multiplicidade  $m$ , a continuidade da curva naquela coordenada paramétrica será  $C^{p-m}$ .

Uma importante observação é que caso um *knot* interno  $\xi_i$  possua multiplicidade igual ao grau da B-Spline ( $m = p$ ), então a curva interpolará um ponto de controle em  $\xi_i$ . Se os *knots* extremos tiverem esta multiplicidade  $m = p + 1$ , então os pontos de controle extremos serão interpolados. Por este motivo que a maioria das representações B-Splines utilizadas em Análise Isogeométrica possuem multiplicidade  $p + 1$  nos *knots* extremos, garantindo que os pontos de controle inicial e final sejam interpolados, como ocorre com as curvas de Bézier. Este tipo de vetor de *knots* é conhecido na literatura como *open knot vector*, ou vetor de *knots* aberto.

A Figura 17 mostra as funções de base quadrática considerando um *knot* com multiplicidade 2 no valor paramétrico 0.5. Podemos observar o caráter interpolador da função de base na posição 0.5 pelo fato de apenas a base  $N_{3,2}$  contribuir para o cálculo da curva definida pela Equação (2.26), resultando em  $\mathbf{c}(0.5) = \mathbf{p}_3$ . A Figura 18 mostra o efeito da consideração de um vetor de *knots* com multiplicidade  $m = p$  em *knots* internos, comparando duas curvas B-Splines quadráticas que possuem os mesmos pontos de controle, mas vetores de *knots* diferentes.

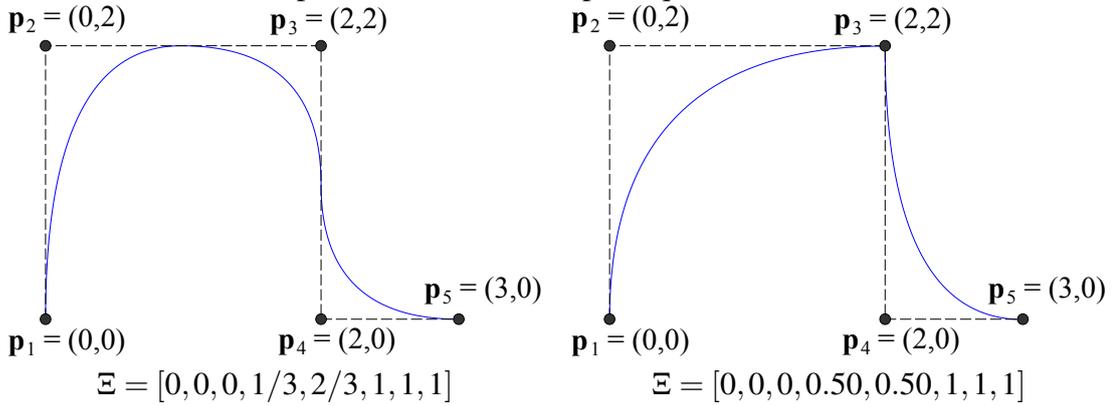
Figura 17 – Funções de base B-Spline quadráticas com  $\Xi = [0, 0, 0, 0.5, 0.5, 1, 1, 1]$ .



Fonte: Elaborada pelo autor.

É importante notar que as bases são definidas por partes (*piecewise function*), pois uma dada base  $N_{i,p}(\xi)$  é nula fora do intervalo  $[\xi_i, \xi_{i+p+1}]$ , possibilitando que a influência dos pontos de controle se restrinjam a partes da curva. Matematicamente, diz-se que as bases B-Spline tem suporte compacto. Esta é uma grande vantagem que a representação B-Splines oferece em relação a representação de Bézier, em que todos os pontos de controle influenciam na curva.

Figura 18 – Efeito da multiplicidade em uma B-Spline quadrática.



Fonte: Elaborada pelo autor.

As curvas B-Splines podem representar curvas de Bézier caso seja considerado um único *knot span* e as multiplicidades dos *knots* sejam  $p + 1$ :

$$\Xi = [\underbrace{0, \dots, 0}_{p+1}, \underbrace{1, \dots, 1}_{p+1}]$$

O mesmo conceito é utilizado no caso de superfícies.

As funções de base B-Splines  $N_{i,p}(\xi)$  possuem as várias propriedades importantes<sup>38</sup>:

- Não-negatividade:  $N_{i,p}(\xi) \geq 0$ ;
- Partição da unidade:  $\sum_{i=1}^n N_{i,p}(\xi) = 1$
- Suporte compacto:  $N_{i,p}(\xi) = 0$  se  $\xi$  estiver fora do intervalo  $[\xi_i, \xi_{i+p+1}]$ ;
- Dado um *knot span*  $[\xi_j, \xi_{j+1}]$ , exatamente  $p + 1$  funções de base são não nulas;
- Todas as derivadas de  $N_{i,p}(\xi)$  existem no interior dos *knot spans*. Nos *knots* as bases são diferenciáveis  $p - m$  vezes, onde  $m$  é a multiplicidade do *knot*.

### 2.3.1 Inserção de knot e elevação de grau

A *inserção de knot* e *elevação do grau* são operações realizadas em B-Splines que alteram a descrição da curva sem modificar a forma desta. A *Inserção de knot* adiciona um novo valor  $\xi_i$  no vetor de *knots*  $\Xi$ , uma nova base  $N_{i,p}$  e um novo ponto de controle. Para manter a curva idêntica, alguns dos pontos de controle são alterados, como ilustrado na Figura 19. Considerando um vetor de *knots*  $\Xi = [\xi_1, \xi_2, \dots, \xi_{n+p+1}]$  e um valor paramétrico a ser inserido  $\bar{\xi} \in [\xi_k, \xi_{k+1}]$ , os novos pontos de controle  $\bar{\mathbf{p}}$  podem ser obtidos em função dos pontos de controle antigos  $\mathbf{p}$

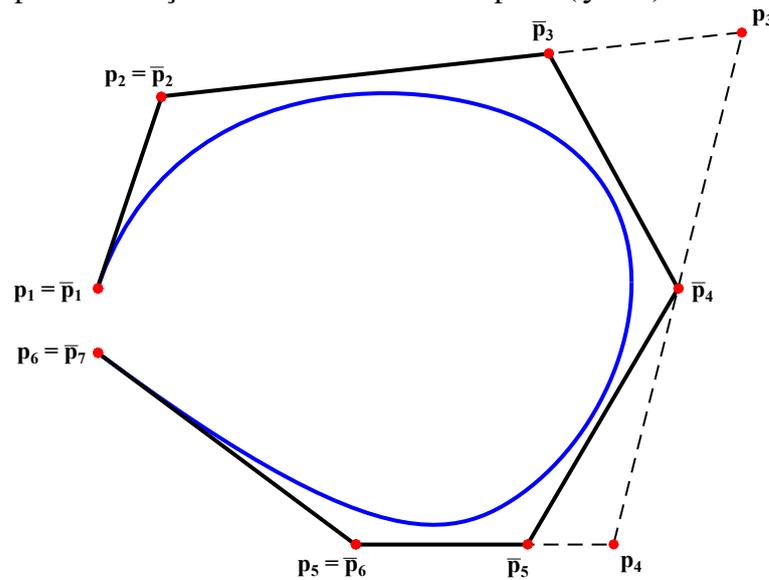
pela expressão:

$$\bar{\mathbf{p}}_i = \begin{cases} \mathbf{p}_1, & i = 1, \\ \alpha_i \mathbf{p}_i + (1 - \alpha_i) \mathbf{p}_{i-1}, & 1 < i < h, \\ \mathbf{p}_{h-1} & i = h. \end{cases} \quad (2.30)$$

onde  $h$  é o tamanho do novo vetor de pontos de controle  $\bar{\mathbf{p}}$ . O termo  $\alpha$  é calculado por:

$$\alpha_i = \begin{cases} 1, & 1 \leq i \leq k - p, \\ \frac{\bar{\xi} - \xi_i}{\xi_{i+p} - \xi_i}, & k - p + 1 \leq i \leq k, \\ 0, & i \geq k + 1. \end{cases} \quad (2.31)$$

Figura 19 – Exemplo de inserção de *knot* em curva B-Spline ( $\bar{\xi} = 2$ ).



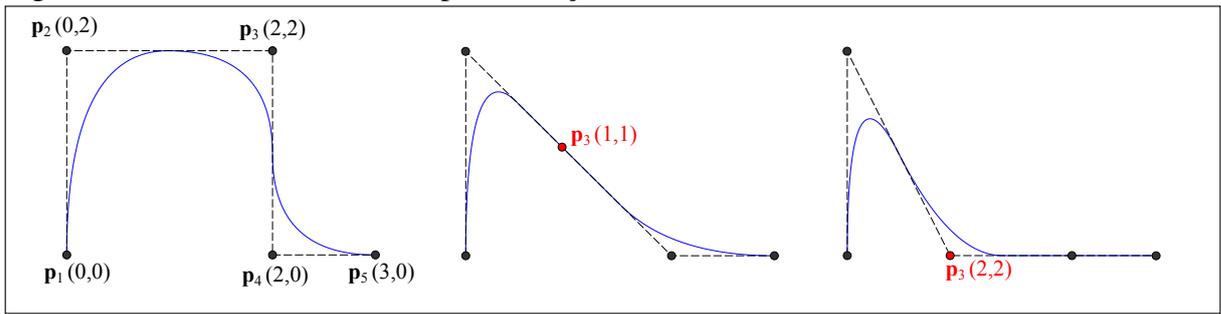
Vetor de *knots* antes = [0,0,0,1,3,4,4,4]

Vetor de *knots* depois = [0,0,0,1,2,3,4,4,4]

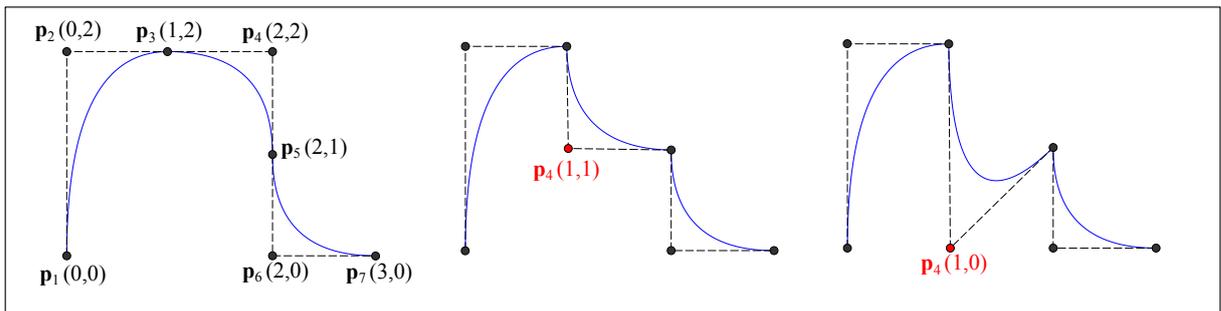
Fonte: Elaborada pelo autor.

Do ponto de vista de modelagem geométrica, a *inserção de knot* permite um controle local maior da curva. Uma vez que sejam inseridos *knots* com multiplicidade  $m = p$ , a quebra da continuidade possibilita isolar em intervalos paramétricos as modificações provenientes das alterações de posição de alguns pontos de controle da curva, como está ilustrado na Figura 20. Do ponto de vista da Análise Isogeométrica, o *inserção de knot* possibilita discretizar o modelo inserindo um número maior de graus de liberdade.

Figura 20 – Controle local obtido pelo inserção de *knots*.



(a)  $\Xi = [0, 0, 0, 1/3, 2/3, 1, 1, 1]$ .

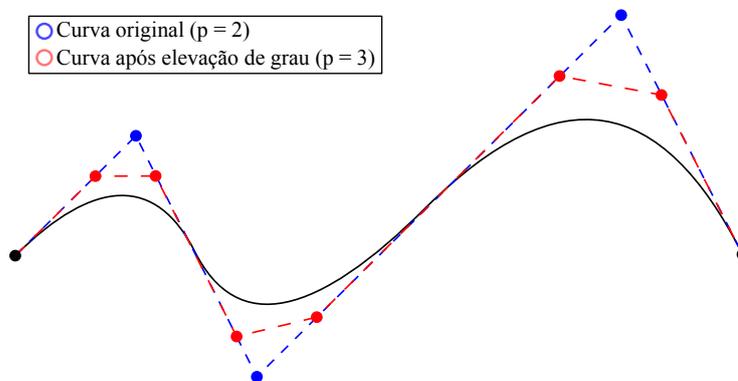


(b)  $\Xi = [0, 0, 0, 1/3, 1/3, 2/3, 2/3, 1, 1, 1]$ .

Fonte: Elaborada pelo autor.

A *elevação de grau* atua nas B-Splines elevando o grau da curva em cada intervalo paramétrico, mantendo a continuidade original em cada *knot*. A multiplicidade de cada *knot* é aumentada em 1, preservando a continuidade original existente nos *knots*. A Figura 21 mostra o processo de elevação de grau sendo aplicado a uma curva B-Spline quadrática com  $\Xi = [0, 0, 0, 1/3, 2/3, 1, 1, 1]$  com 5 pontos de controle, e passa a ser representada por uma curva cúbica com  $\Xi = [0, 0, 0, 0, 1/3, 1/3, 2/3, 2/3, 1, 1, 1, 1]$  e 8 pontos de controle.

Figura 21 – Elevação de grau aplicada a uma curva B-Spline quadrática.



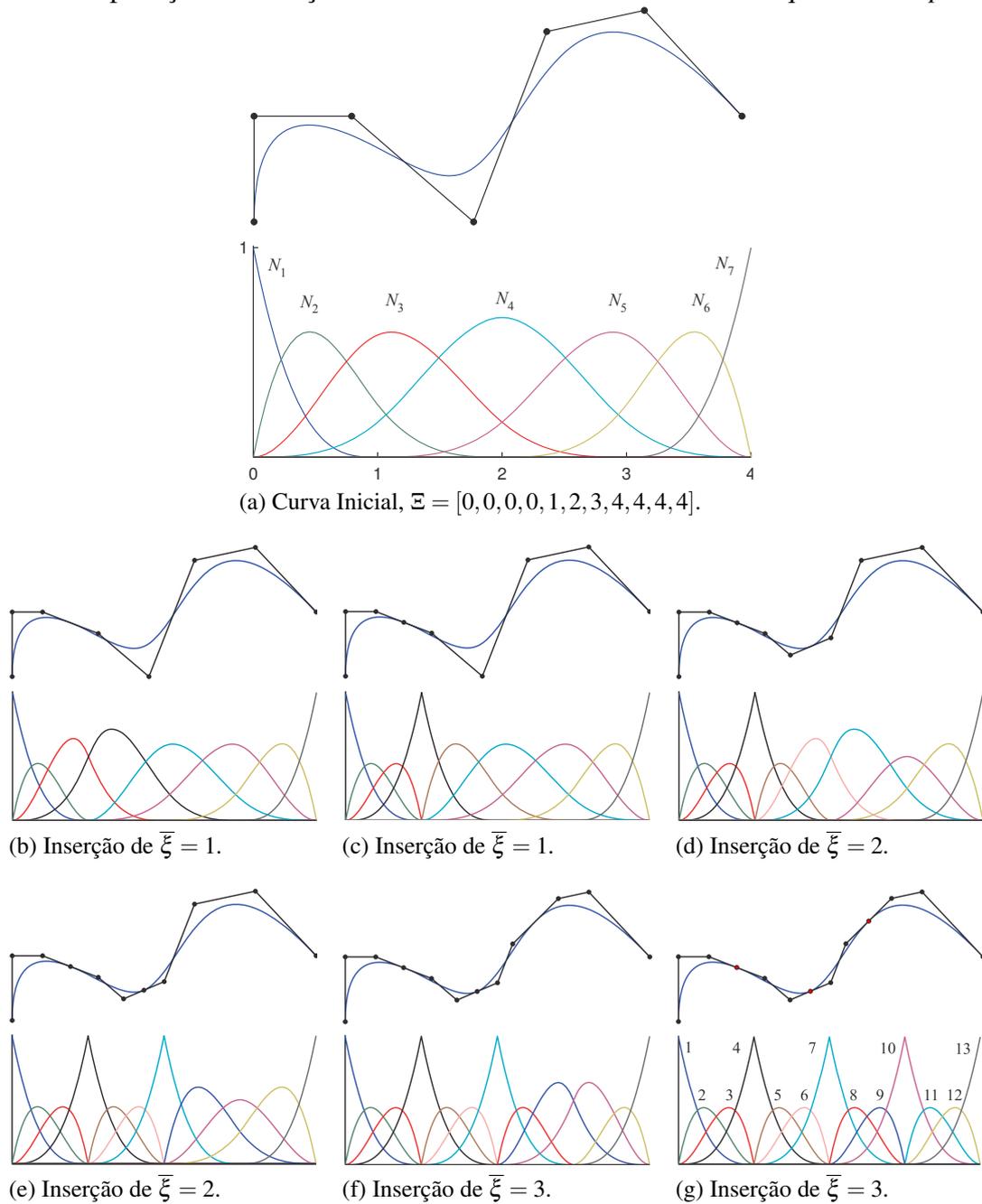
Fonte: Elaborada pelo autor.

### 2.3.2 Extração de Bézier

A extração de Bézier é um procedimento utilizado para obter os pontos das curvas de Bézier que representam uma curva B-Spline. Neste procedimento, cada *knot span* da B-Spline é representado por uma curva de Bézier. A forma mais simples de realizar a extração de Bézier é aplicando operações de inserção de *knots* até que todos os *knots* internos da B-Spline tenham multiplicidade  $m = p$ . Também é possível aplicar multiplicidade  $m = p + 1$ , resultando em uma B-Spline com repetição de pontos de controle internos<sup>38</sup>. A Figura 22 ilustra a aplicação da extração de Bézier em uma curva B-Spline cúbica com 4 *knot spans*. Pode-se observar que as bases não-nulas em cada *knot span* se tornam iguais aos polinômios de Bernstein apresentados na Figura 4c. Os pontos de controle internos interpolados (destacados em vermelho em Figura 22g) são pontos de controle compartilhados entre as curvas de Bézier.

A extração de Bézier também pode ser usada para relacionar as bases de Bézier, que são iguais para todas as curvas, com as bases B-Spline, que variam conforme a definição do vetor de *knots*. A relação consiste em uma combinação linear e pode ser descrita por uma matriz, conhecida na literatura como Matriz de Extração. O uso desta matriz facilita a implementação da AIG baseada em NURBS em código de elementos finitos tradicionais, pois nestes as funções de base são iguais para todos os elementos<sup>45</sup>. No presente trabalho, a AIG proposta utiliza elementos baseados em superfícies e triângulos de Bézier, não sendo necessário o uso da matriz de extração. Por outro lado, a extração de Bézier é utilizada nas técnicas de geração de malhas apresentadas posteriormente no Capítulo 3.

Figura 22 – Aplicação da extração de Bézier em uma curva cúbica com quatro *knot spans*.



Fonte: Modificado de Borden e colaboradores<sup>45</sup>.

## 2.4 NURBS

As B-Splines não são capazes de representar exatamente cônicas, como circunferências ou elipses, uma vez que estas curvas não são representadas por polinômios. Esta limitação pode ser eliminada utilizando bases racionais, o mesmo procedimento realizado para curvas de Bézier na Seção 2.1.1. As NURBS (*Non Uniform Rational B-Splines*) são B-Splines racionais com vetores de *knots* não uniformes. São amplamente utilizadas na atualidade por

modeladores geométricos em geral. A grande aceitação das NURBS na indústria de modelagem computacional se deve as vantagens descritas abaixo<sup>46</sup>:

- Oferecem uma forma matemática capaz de representar tanto modelos matemáticos padrão (quádricas, cônicas, superfícies de revolução, ...) quanto modelos de forma livre, utilizando a mesma base de dados para armazenamento de ambos;
- Uma grande variedade de formas podem ser modeladas alterando os pontos de controle e pesos do modelo;
- Podem ser avaliadas de forma rápida e numericamente estável;
- Transformações afins, como translação, rotação, espelhamento e cisalhamento podem ser aplicadas diretamente nos pontos de controle da NURBS;
- Podem representar outras representações paramétricas importantes, como B-Splines, Bézier racionais e não-rationais.

As bases racionais requerem que cada ponto de controle tenha um peso, estes alteram a influência de cada ponto de controle sobre NURBS. Uma curva NURBS é expressa por:

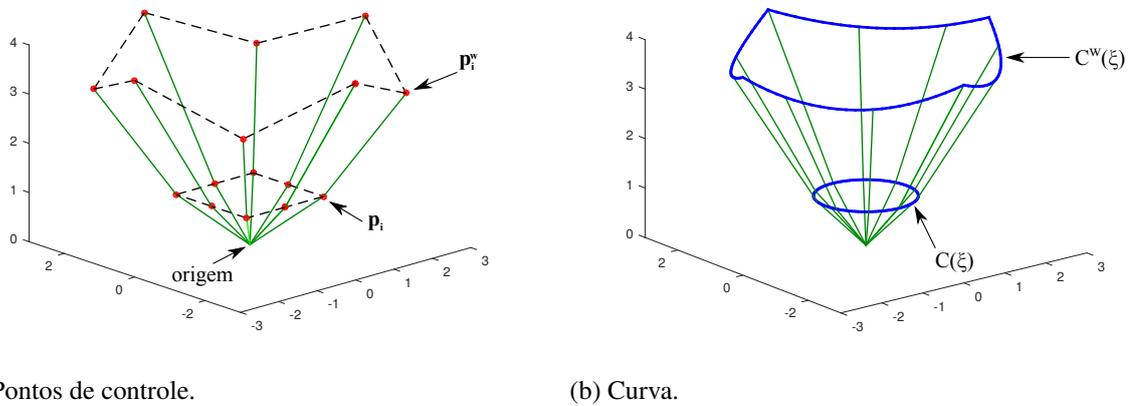
$$\mathbf{c}(\xi) = \frac{\sum_{i=1}^n w_i N_{i,p}(\xi) \mathbf{p}_i}{\sum_{i=1}^n w_i N_{i,p}(\xi)} \quad (2.32)$$

onde  $w_i$  é o peso associado a cada ponto de controle  $\mathbf{p}_i$ . Uma NURBS descrita no espaço  $\mathbb{R}^d$  é obtida da projeção de uma B-Spline descrita em  $\mathbb{R}^{d+1}$ . Os pontos de projeção  $\mathbf{p}_i^w$  podem ser calculados por:

$$(\mathbf{p}_i^w)_j \begin{cases} (\mathbf{p}_i)_j w_i & \text{se } j < d + 1 \\ w_i & \text{se } j = d + 1 \end{cases} \quad (2.33)$$

Uma circunferência modelada por uma curva NURBS  $\mathbf{c}(\xi)$  em  $\mathbb{R}^2 (x, y)$  é obtida da projeção de uma B-Spline  $\mathbf{c}(\xi)^w$  em  $\mathbb{R}^3 (x, w, y, w)$  sobre o plano  $w = 1$ , como mostra a Figura 23.

Figura 23 – Curva NURBS  $\mathbf{c}(\xi)$  em  $\mathbb{R}^2$  sendo obtida da projeção de uma curva B-Spline  $\mathbf{c}^w(\xi)$  em  $\mathbb{R}^3$ .



Fonte: Elaborada pelo autor.

É importante lembrar que os algoritmos de *inserção de knot* e *elevação de grau* podem ser aplicados a NURBS da mesma forma que são aplicados a B-Splines se forem aplicados considerando os pontos de controle em  $\mathbb{R}^{d+1}$ . A *inserção de knots* pode ser aplicada transformando os pontos de controle em  $\mathbb{R}^d$  para o espaço  $\mathbb{R}^{d+1}$  ( $\mathbf{p} \rightarrow \mathbf{p}^w$ ) através da Equação (2.33), em seguida aplicando a Equação (2.30) para encontrar novos pontos de controle  $\bar{\mathbf{p}}^w$ , e finalmente transformado os pontos de volta para o espaço  $\mathbb{R}^d$ :

$$(\bar{\mathbf{p}}_i)_j = \frac{(\bar{\mathbf{p}}_i^w)_j}{\bar{w}_i} \quad j = 0, \dots, d \quad (2.34)$$

onde  $\bar{w}_i = (\bar{\mathbf{p}}_i^w)_{d+1}$

O mesmo procedimento também é utilizado na aplicação dos algoritmos de *subdivisão* e *elevação de grau* em curvas, superfícies e triângulos de Bézier racionais.

As funções de base NURBS  $\tilde{R}_{i,p}$  podem ser escritas em função das bases B-Spline  $N_{i,p}$  e dos pesos  $w_i$  como:

$$\tilde{R}_{i,p}(\xi) = \frac{N_{i,p}(\xi) w_i}{\tilde{W}(\xi)} = \frac{N_{i,p}(\xi) w_i}{\sum_{\hat{i}=1}^n N_{\hat{i},p}(\xi) w_{\hat{i}}} \quad (2.35)$$

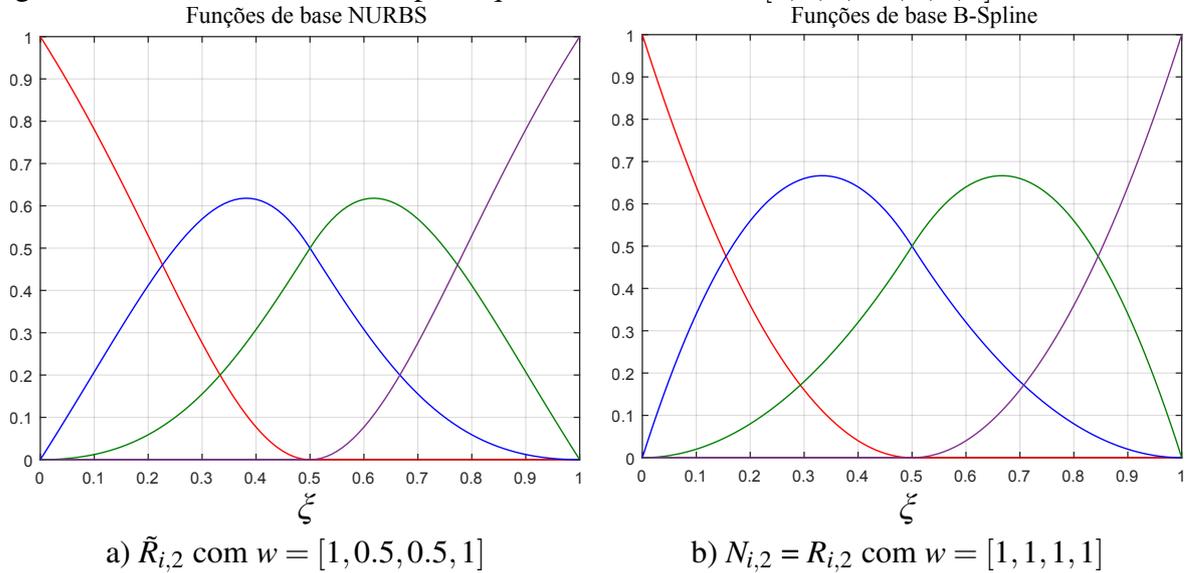
onde  $\tilde{W}(\xi)$  é a função peso, dada por:

$$\tilde{W}(\xi) = \sum_{\hat{i}=1}^n N_{\hat{i},p}(\xi) w_{\hat{i}} \quad (2.36)$$

A Figura 24 mostra as funções de base racionais  $R_{i,2}$  e não racionais  $N_{i,2}$ , considerando um vetor de *knots*  $\Xi = [0, 0, 0, 0.5, 1, 1, 1]$  e pesos  $w = [1, 0.5, 0.5, 1]$ , pode-se observar uma pequena

diferença nas curvas de cada base. A Figura 25 mostra as curvas de uma semi-circunferência modelada com NURBS e o arco de 180° resultante de B-Spline não racional ( $w = 1$ ).

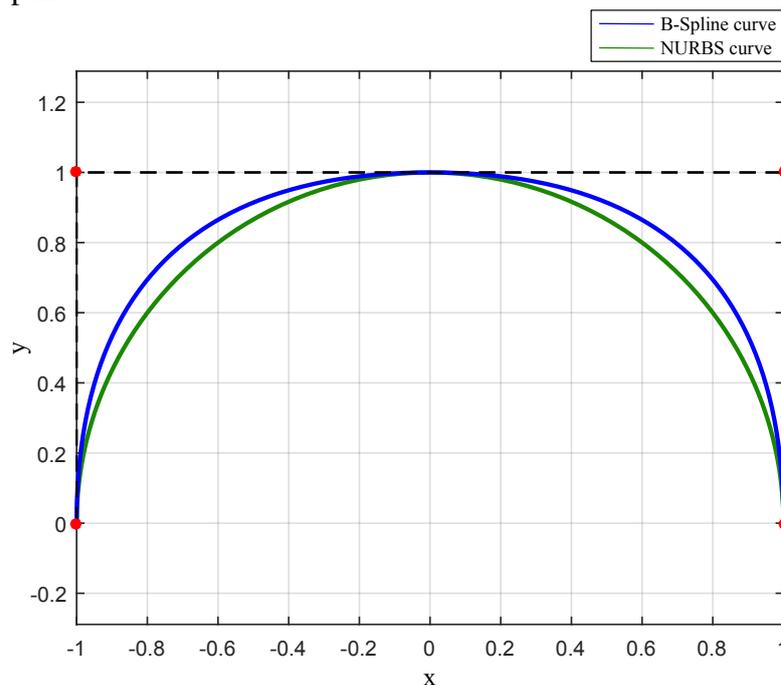
Figura 24 – Bases NURBS e B-Spline quadráticas com  $\Xi = [0, 0, 0, 0.5, 1, 1, 1]$ .



Fonte: Elaborada pelo autor.

É importante ressaltar que as funções de base racionais  $\tilde{R}_{i,p}(\xi)$  herdam as mesmas propriedades das funções de base B-Spline  $N_{i,p}(\xi)$ , descritas na Seção 2.3.

Figura 25 – Semi circunferência construída com NURBS e um arco de 180° construído com B-Spline.



Fonte: Elaborada pelo autor.

### 2.4.1 Superfícies NURBS

Uma superfície NURBS definida por produto tensorial é avaliada pelo produto de duas funções de base univariantes NURBS. Dada uma superfície  $s$  definida por uma matriz de pontos de controle  $\mathbf{P}$  ( $n \times m$ ), uma NURBS de grau  $p$  na direção  $\xi$  com vetor de *knots*  $\Xi = [\xi_1, \xi_1, \dots, \xi_{n+p+1}]$  e outra NURBS de grau  $q$  na direção  $\eta$  com vetor de *knots*  $\mathcal{H} = [\eta_1, \eta_1, \dots, \eta_{m+q+1}]$ , a superfície pode ser calculada por:

$$\mathbf{s}(\xi, \eta) = \sum_{i=1}^n \sum_{j=1}^m \tilde{R}(\xi, \eta)_{ij} \mathbf{P}_{ij} \quad (2.37)$$

onde  $\tilde{R}(\xi, \eta)$  é a função de base racional bivariada dada por:

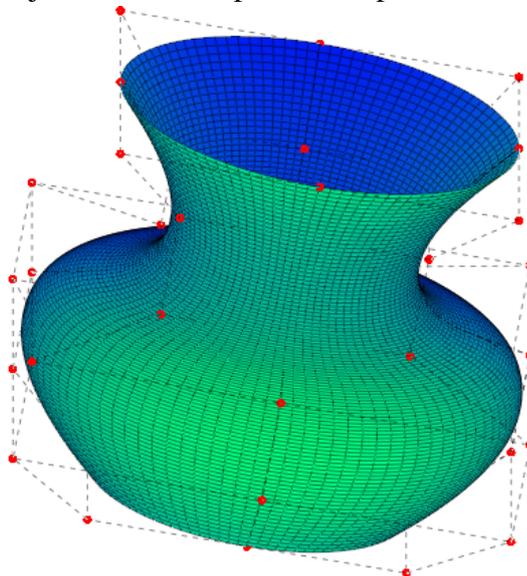
$$\tilde{R}(\xi, \eta)_{ij} = \frac{w_{ij} N_{i,p}(\xi) N_{j,q}(\eta)}{\tilde{W}(\xi, \eta)}, \quad (2.38)$$

onde  $\tilde{W}(\xi, \eta)$  é a função peso bivariada:

$$\tilde{W}(\xi, \eta) = \sum_{i=1}^n \sum_{j=1}^m w_{ij} N_{i,p}(\xi) N_{j,q}(\eta) \quad (2.39)$$

A Figura 26 mostra um jarro modelado por uma superfície NURBS.

Figura 26 – Exemplo de um jarro modelado por uma superfície NURBS.



Fonte: Elaborada pelo autor.

## 2.5 Superfície de Coons

Na modelagem de superfícies, em muitos casos é mais prático definir a forma de uma superfície a partir de curvas de contorno. Existem várias superfícies que podem ser definidas desta forma, algumas alternativas utilizadas neste trabalho são apresentadas a seguir.

Uma *ruled surface*<sup>14</sup> é obtida pela combinação linear de duas curvas de contorno:

$$\mathbf{s}^r(\xi, \eta) = (1 - \eta) \mathbf{c}_1(\xi) + \eta \mathbf{c}_2(\xi) \quad (2.40)$$

onde  $\mathbf{c}_1$  e  $\mathbf{c}_2$  são as curvas de contorno. O grau de  $\mathbf{s}^r$  em  $\eta$  é linear, e em  $\xi$  é o maior grau das curvas de contorno. Se  $\mathbf{c}_1$  e  $\mathbf{c}_2$  forem curvas NURBS de mesmo grau  $p$  e vetor de *knots*  $\Xi$ , então  $\mathbf{s}^r$  pode ser representada por uma superfície NURBS de grau  $p \times 1$ , com vetores de *knots*  $\Xi$  e  $\mathcal{H} = [0 \ 0 \ 1 \ 1]$ , e os pontos de controle são os mesmos das curvas  $\mathbf{c}_1$  e  $\mathbf{c}_2$ .

Uma superfície de Coons<sup>15, 14</sup>, ou *patch* de Coons, é uma superfície que interpola quatro curvas de contorno, duas em  $\xi$  ( $\mathbf{c}_1$  e  $\mathbf{c}_2$ ) e duas em  $\eta$  ( $\mathbf{g}_1$  e  $\mathbf{g}_2$ ), e é dada por:

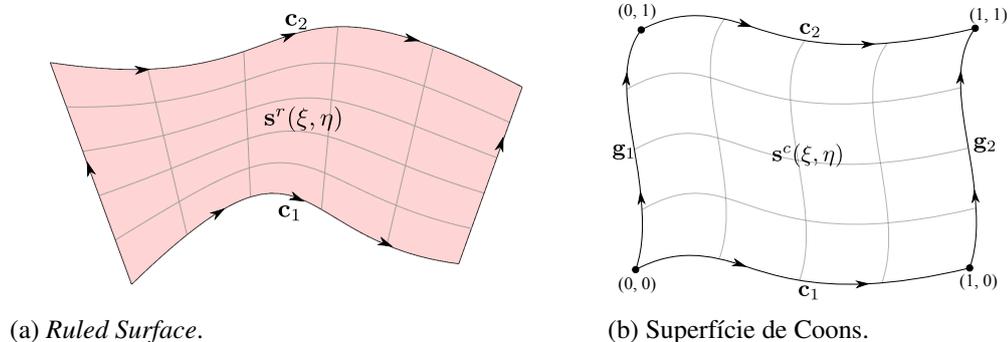
$$\mathbf{s}^c(\xi, \eta) = \mathbf{s}_C^r(\xi, \eta) + \mathbf{s}_G^r(\xi, \eta) - \mathbf{s}_M(\xi, \eta) \quad (2.41)$$

onde  $\mathbf{s}_C^r$  e  $\mathbf{s}_G^r$  são *ruled surfaces* definidas para os pares de curvas  $(\mathbf{c}_1, \mathbf{c}_2)$  e  $(\mathbf{g}_1, \mathbf{g}_2)$ , e  $\mathbf{s}_M$  é uma superfície bilinear definida por:

$$\mathbf{s}_M(\xi, \eta) = (1 - \xi) \eta \mathbf{m}_{01} + \xi \eta \mathbf{m}_{11} + \xi (1 - \eta) \mathbf{m}_{10} + (1 - \xi) (1 - \eta) \mathbf{m}_{00} \quad (2.42)$$

$\mathbf{m}_{00}$ ,  $\mathbf{m}_{01}$ ,  $\mathbf{m}_{10}$  e  $\mathbf{m}_{11}$  são, respectivamente, os pontos avaliados nas coordenadas paramétricas  $(0, 0)$ ,  $(0, 1)$ ,  $(1, 0)$  e  $(1, 1)$ , pontos extremos das curvas de contorno.  $\xi$  e  $\eta$  são definidos no intervalo paramétrico  $[0, 1]$ , tanto para as superfícies quanto para as curvas. Vale notar que  $\mathbf{s}_M$  pode ser representada por uma NURBS de grau  $1 \times 1$ , vetores de *knots*  $\Xi = \mathcal{H} = [0 \ 0 \ 1 \ 1]$ , e os pontos de controle são os pontos  $\mathbf{m}_{00}$ ,  $\mathbf{m}_{01}$ ,  $\mathbf{m}_{10}$  e  $\mathbf{m}_{11}$ . A Figura 27 mostra uma *ruled surface* e uma superfície de Coons.

Figura 27 – Exemplo de superfícies construídas a partir das curvas do contorno.



Fonte: Elaborada pelo autor.

Se as curvas de contorno forem curvas NURBS com mesmo grau e vetor *knots*, em cada par, e seja  $p$  o grau das curvas e  $\Xi$  o vetor de *knots* do par de curvas  $(\mathbf{c}_1, \mathbf{c}_2)$ ,  $q$  o grau

das curvas e  $\mathcal{H}$  o vetor de *knots* do par de curvas  $(\mathbf{g}_1, \mathbf{g}_2)$ , então  $\mathbf{s}^c$  pode ser representada por uma superfície NURBS de grau  $p \times q$ , com vetores de *knots*  $\Xi$  e  $\mathcal{H}$ , e pontos de controle são definidos por:

$$\mathbf{P}^c = \mathbf{P}_C^r + \mathbf{P}_G^r - \mathbf{P}^M \quad (2.43)$$

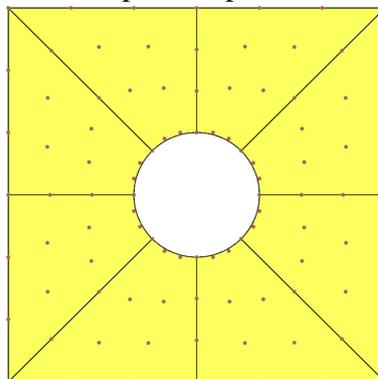
onde  $\mathbf{P}_C^r$  são os pontos de controle da superfície  $\mathbf{s}_C^r$  após realizar elevação de grau em  $\eta$  para o grau  $q$ ,  $\mathbf{P}_G^r$  são os pontos de controle da superfície  $\mathbf{s}_G^r$  após realizar elevação de grau em  $\xi$  para o grau  $p$ , e  $\mathbf{P}^M$  são os pontos de controle da superfície  $\mathbf{s}_M$  após realizar elevação de grau para grau  $p \times q$ . Este resultado é obtido diretamente da Equação (2.41), uma vez que  $\mathbf{s}_C^r(\xi, \eta)$ ,  $\mathbf{s}_G^r(\xi, \eta)$  e  $\mathbf{s}_M(\xi, \eta)$  sejam representadas por superfícies NURBS de bases idênticas, o que é satisfeito caso tenham mesmo grau e vetor de *knots*.

É importante notar que no caso de curvas racionais, a Equação (2.43) deve ser aplicada no espaço de projeção  $\mathbb{R}^{d+1}$ , analogamente ao que é feito nos casos da *inserção de knot* e *elevação de grau*.

## 2.6 Superfícies aparadas

A grande maioria das representações de superfícies utilizadas na modelagem geométrica não são capazes de representar furos arbitrários, pois usualmente são superfícies contínuas em seu domínio. Uma forma de considerar furos é utilizar modelos com múltiplas superfícies. Por exemplo, uma chapa quadrada com furo circular pode ser representada por 8 superfícies de Bézier, como ilustrado na Figura 28. A obtenção do particionamento de regiões em sub-regiões contínuas é um problema complexo no caso geral, geralmente feito de forma manual pelo usuário.

Figura 28 – Placa com furo representado por 8 superfícies de Bézier racionais.

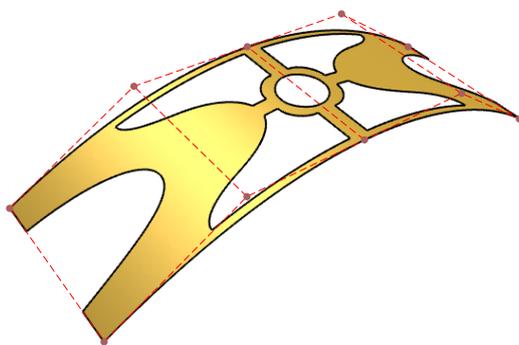


Fonte: Elaborada pelo autor.

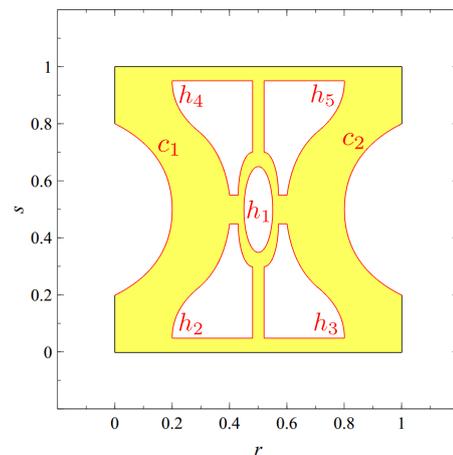
Outra abordagem para o problema é utilizar curvas de aparo (*trimming curves*). Estas

curvas são definidas no espaço paramétrico da superfície, e são utilizadas para considerar furos e recortes. A Figura 29 mostra uma superfície com vários furos e recortes gerada utilizando esta abordagem. Uma das vantagens dessa abordagem é que a descrição da superfície original, como o número dos pontos de controle e suas coordenadas cartesianas, permanece inalterada, sendo trivial a adição ou remoção de furos na superfície. Além disto, o uso de curvas de aparo, bem como os algoritmos utilizados para visualização e manipulação das superfícies aparadas, são independentes do tipo de superfície paramétrica utilizada<sup>47</sup>.

Figura 29 – Superfície NURBS com aparos.



(a) Geometria e pontos de controle.



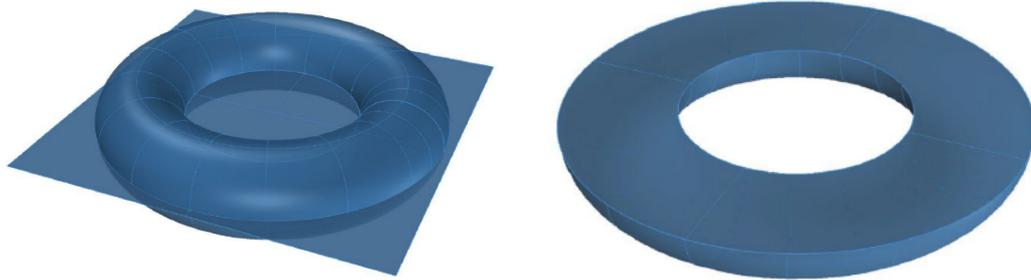
(b) Curvas de aparo no espaço paramétrico.

Fonte: Elaborada pelo autor.

O uso de superfícies aparadas é a abordagem predominante em sistemas CAD. Na prática, as curvas de aparo são utilizadas para considerar interseção entre objetos, quando por exemplo, operações de subtração são realizadas. A Figura 30 ilustra uma operação de subtração para modelagem de um toro cortado ao meio. Neste caso específico, existem duas superfícies aparadas, a do toro, onde metade do modelo é eliminado, e a do plano de corte, com recortes em formato circular para representar a face plana resultante do modelo.

Neste contexto, as curvas de aparo são aproximações das curvas de interseção e são geradas de forma independente para cada superfície. Logo, o modelo resultante pode apresentar inconformidades, como a presença de pequenos vazios próximos a região de interseção. Mesmo que tais inconformidades não sejam perceptíveis visualmente, ainda assim limitam o uso destes modelos aparados em simulações numéricas, sendo necessário o uso de rotinas de correções e reparos geométricos durante a etapa de geração de malhas<sup>48, 47</sup>.

Figura 30 – Operação de subtração entre um toro e um subespaço definido por um plano.



(a) Modelos do torus e do subespaço plano.

(b) Modelo aparado resultante.

Fonte: Modificado de Marussig e Hughes<sup>47</sup>.

Além disto, as curvas de aparo podem apresentar alta complexidade no espaço Cartesiano. Suponha que uma curva de aparo cúbica  $\mathbf{c}(t) = (t^3, t^3)$  seja utilizada em uma superfície de Bézier cúbica  $\mathbf{s}(\xi, \eta)$ . A curva pode ser avaliada no espaço Cartesiano por:

$$\mathbf{c}(t) = \mathbf{s}(\mathbf{c}(t)) = \mathbf{s}(t^3, t^3) = \sum_{i=1}^{p+1} \sum_{j=1}^{q+1} B_{i,p}(t^3) B_{j,q}(t^3) \mathbf{P}_{ij} \quad (2.44)$$

no somatório, há termos com multiplicação de bases cúbicas em  $\xi$  e  $\eta$ , logo

$$B_{i,p}(\xi) B_{j,q}(\eta) = \xi^3 \eta^3 = (t^3)^3 (t^3)^3 = t^9 t^9 = t^{18} \quad (2.45)$$

Portanto, a curva resultante apresenta grau 18. Vale notar que a alta complexidade da curva de aparo acaba limitando sua descrição exata no contexto de análises isogeométricas, sendo adotado aproximações de menor grau.

Neste trabalho são utilizados modelos de superfícies aparadas para simulação numérica de cascas. As curvas de aparo são aproximadas por curvas de Bézier racionais no espaço Cartesiano, permitindo a parametrização do modelo em elementos finitos de triângulos e superfícies de Bézier. O procedimento para obtenção das curvas aproximadas é descrito na próxima seção. É importante notar que os modelos tratados neste trabalho não são sólidos, portanto, neste caso, não existe o problema de compatibilidade entre curvas de aparo descrito anteriormente.

## 2.7 Ajuste de curvas e superfícies

O ajuste de curvas e superfícies é composto por um conjunto de procedimentos que calculam uma curva ou superfície de modo a aproximar ou interpolar um conjunto de pontos no espaço Cartesiano. A regressão linear é um procedimento de ajuste aplicado a modelos lineares, onde a relação entre os parâmetros do modelo e as respostas pode ser descrita de forma matricial.

No caso das representações de Bézier e B-Splines, deseja-se determinar as coordenadas dos pontos de controle de modo a aproximar uma curva ou superfície a um conjunto de pontos no espaço. Uma forma simples de obter um modelo linear neste caso é associando, a cada ponto amostral cartesiano, uma localização paramétrica específica na superfície. Assim, a relação entre os pontos de controle e as respostas nas localidades consideradas é linear.

O procedimento é descrito a seguir. Uma curva de Bézier de grau  $p$  pode ser escrita matricialmente por:

$$\mathbf{c}(\xi) = [B_{1,p}(\xi) \dots B_{p+1,p}(\xi)] \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_{p+1} \end{bmatrix} = \tilde{\mathbf{b}} \mathbf{p} \quad (2.46)$$

logo, para um conjunto  $n_{sp}$  de pontos amostrais  $\bar{\mathbf{p}}$ , associados a um conjunto de posições paramétricas  $(\xi_1, \dots, \xi_{n_{sp}})$ , chega-se ao seguinte sistema de equações:

$$\begin{bmatrix} \tilde{\mathbf{b}}_{\xi_1} \\ \vdots \\ \tilde{\mathbf{b}}_{\xi_{n_{sp}}} \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_{p+1} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{p}}_1 \\ \vdots \\ \bar{\mathbf{p}}_{n_{sp}} \end{bmatrix} \implies \tilde{\mathbf{B}} \mathbf{p} = \bar{\mathbf{p}} \quad (2.47)$$

onde  $\tilde{\mathbf{B}}$  é a matriz das funções de base avaliadas nas posições paramétricas das amostras. Se  $n_{sp}$  for igual ao número de pontos de controle da curva considerada, então o sistema pode interpolar os valores amostrais caso  $\det(\tilde{\mathbf{B}}) \neq 0$ . Se  $n_{sp}$  for maior que o número de pontos de controle, então os pontos amostrais são aproximados, e os pontos de controle são determinados resolvendo o sistema do problema de mínimos quadrados<sup>14</sup>:

$$(\tilde{\mathbf{B}}^T \tilde{\mathbf{B}}) \mathbf{p} = (\tilde{\mathbf{B}}^T \bar{\mathbf{p}}) \quad (2.48)$$

O procedimento de ajuste é análogo para superfícies e triângulos de Bézier, curvas e superfícies B-Spline. No caso de superfícies, as bases devem ser organizadas de forma vetorial, semelhante ao descrito para triângulos de Bézier, e a coordenada paramétrica  $\eta$  também é considerada. Nas B-Splines, os vetores de *knots* também devem ser fornecidos.

Uma variação importante do problema de ajuste descrito é quando existem pontos de controle conhecidos, e o ajuste é realizado apenas em um sub-grupo de pontos de controle. Neste caso, a equação de ajuste é escrita por:

$$\tilde{\mathbf{B}} \mathbf{p} = \bar{\mathbf{p}} - \tilde{\mathbf{B}}_{ext} \bar{\mathbf{p}}_{ext} \quad (2.49)$$

onde  $\bar{\mathbf{p}}_{ext}$  são os pontos de controle conhecidos,  $\tilde{\mathbf{B}}_{ext}$  é a matriz das funções de base relacionadas a  $\bar{\mathbf{p}}_{ext}$ , avaliadas nas posições paramétricas das amostras. Aqui  $\mathbf{p}$  e  $\tilde{\mathbf{B}}$  não consideram os pontos conhecidos. Um caso usual deste versão de ajuste é quando se tem os pontos de controle extremos de uma representação, e se deseja encontrar os pontos internos que aproximem uma amostra, seja em curvas ou superfícies.

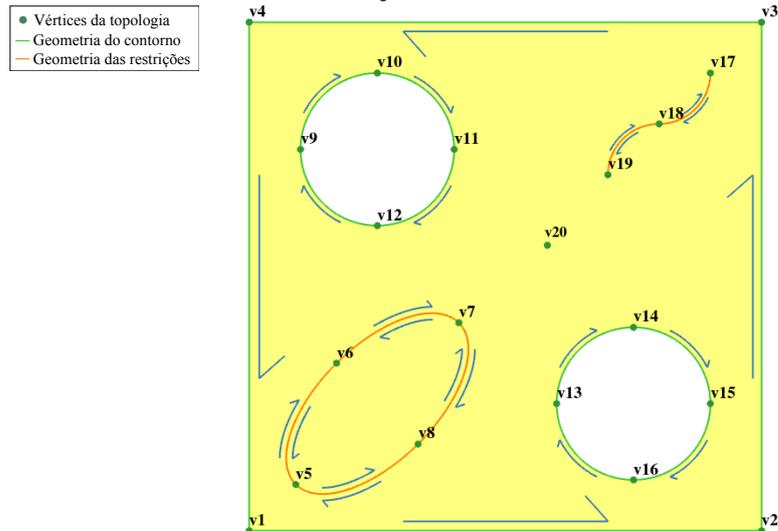
É importante notar que o procedimento de ajuste também pode ser realizado em superfícies racionais. Em alguns casos, os pontos amostrais são obtidos de outras representações racionais, onde a coordenada  $w$  é conhecida. Neste caso, ajuste pode ser feito no espaço  $\mathbb{R}^{d+1}$ , e ao final do procedimento, projetar os pontos de controle obtidos para o espaço  $\mathbb{R}^d$  usando a Equação (2.34). Este procedimento pode ser utilizado para implementar todos os algoritmos de modificação de representação de curvas e superfícies mencionados anteriormente, como *subdivisão*, *elevação* e *redução de grau*, *inserção* e *eliminação de knot*.

Por fim, é importante destacar que existem outros procedimentos de ajuste na literatura, como a *projeção de Bézier*, onde é feito um ajuste não-linear entre diversas representações paramétricas, incluindo NURBS e T-Splines<sup>49</sup>. Contudo, estes procedimentos não serão considerados neste trabalho.

## 2.8 Representação de Fronteira

A representação de fronteira é um paradigma de modelagem que consiste em representar objetos pelo seu contorno. Os modelos B-Rep definem informações geométricas, relacionadas à forma de curvas e superfícies, e topológicas, relacionadas à conectividade entre vértices, arestas e faces. Neste trabalho são utilizados modelos B-Rep bidimensionais definidos por um conjunto de curvas NURBS, organizadas em ciclos. Os ciclos são agrupados em regiões, e podem representar tanto o contorno externo de uma região como furos ou restrições no interior desta.

Figura 31 – Placa com furos circulares e restrições.

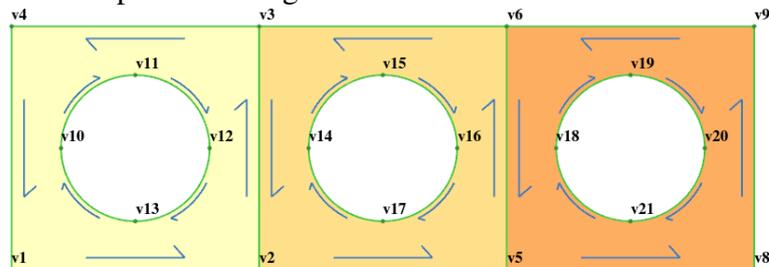


Fonte: Elaborada pelo autor.

A Figura 31 mostra um modelo B-Rep com vários furos circulares e restrições. Pode-se notar que uma aresta pode fazer parte de até 2 ciclos, e que a orientação dos ciclos do contorno da região é anti-horária, enquanto que a orientação dos furos é horário. As restrições são vértices e arestas que não fazem parte do contorno do modelo, mas são impostas durante o processo de geração de malha, garantindo que tais vértices e arestas existam em malhas do modelo.

Modelos B-Rep com várias regiões podem ser utilizados para atribuir propriedades distintas em cada região do modelo. Na Figura 32, uma chapa com três furos é modelada com três regiões, permitindo que parâmetros relacionados à simulação numérica, como as propriedades mecânicas dos materiais, sejam atribuídos em partes do modelo de forma distinta. Além disto, também pode-se utilizar algoritmos de geração de malha distintos em cada região específica.

Figura 32 – Modelo B-Rep com três regiões.

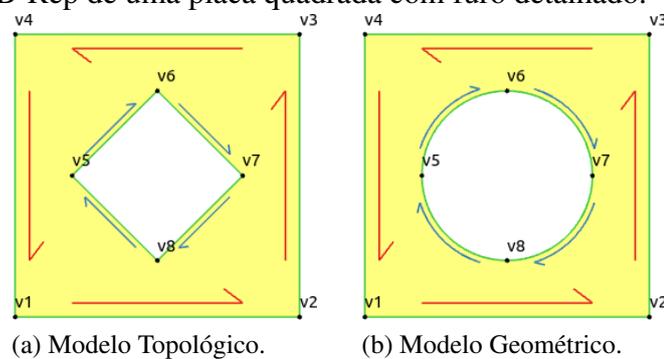


Fonte: Elaborada pelo autor.

O modelo topológico de uma B-Rep é uma idealização poligonal do objeto, na qual fica evidente as relações de adjacência existentes nas entidades topológicas do modelo, que são

vértices, arestas, faces e regiões. O modelo geométrico de uma B-Rep consiste na descrição das formas de cada entidade topológica, sendo curvas no caso de arestas, superfícies no caso das faces. Neste trabalho, os vértices de uma aresta do modelo topológico coincidem com os pontos de controle extremos da curva NURBS que descreve a forma da aresta. Desta forma, são utilizadas sempre NURBS com vetor de *knots* aberto, garantindo pontos de controle interpolados nos extremos. A Figura 33 apresenta um modelo B-Rep de uma placa com furo circular, mostrando os modelos topológico e geométrico, além da descrição detalhada da estrutura de dados do modelo.

Figura 33 – Modelo B-Rep de uma placa quadrada com furo detalhado.



Modelo B-Rep	Lista de Vértices	Lista de Arestas	Geometria das Arestas
Regiões: 1	8 <id x y>	8 <id v1 v2>	8 <id p $\Xi$ p <sub>i</sub> >
1	1 0 0	1 (1,2)	1 1 [0 0 1 1]
Contorno:	2 10 0	2 (2,3)	2 1 [0 0 1 1]
1 2 4 5	3 10 10	3 (3,4)	3 1 [0 0 1 1]
Furos: 1	4 0 10	4 (4,1)	4 1 [0 0 1 1]
1	5 2 5	5 (5,6)	5 2 [0 0 0 1 1 1] 2 8 $\sqrt{2}/2$
5 8 7 6	6 5 8	6 (6,7)	6 2 [0 0 0 1 1 1] 8 8 $\sqrt{2}/2$
	7 8 5	7 (7,8)	7 2 [0 0 0 1 1 1] 8 2 $\sqrt{2}/2$
	8 5 2	8 (8,5)	8 2 [0 0 0 1 1 1] 2 2 $\sqrt{2}/2$

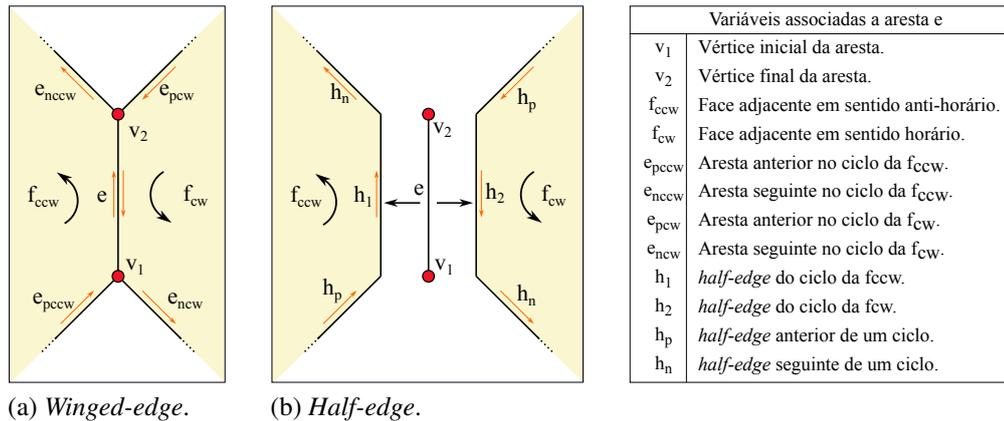
\*p<sub>i</sub> são os pontos de controle internos da curva.

(c) Dados do modelo.

Fonte: Elaborada pelo autor.

A estrutura de dados utilizadas nas B-Reps são otimizadas para processar consultas e modificações nos modelos de forma localizada e eficiente. Normalmente são utilizadas estruturas de dados baseadas em aresta. Estas estruturas são caracterizadas por definirem tipos de dados de arestas com acesso direto, via referências ou ponteiros, a vértices, faces e ciclos adjacentes. As estruturas mais utilizadas são as *winged-edge* e *half-edge*<sup>11, 50</sup>, ilustradas na Figura 34. A estrutura de dados implementada neste trabalho é baseada em *winged-edge*.

Figura 34 – Estrutura da dados baseadas em aresta.

(a) *Winged-edge*.(b) *Half-edge*.

Fonte: Elaborada pelo autor.

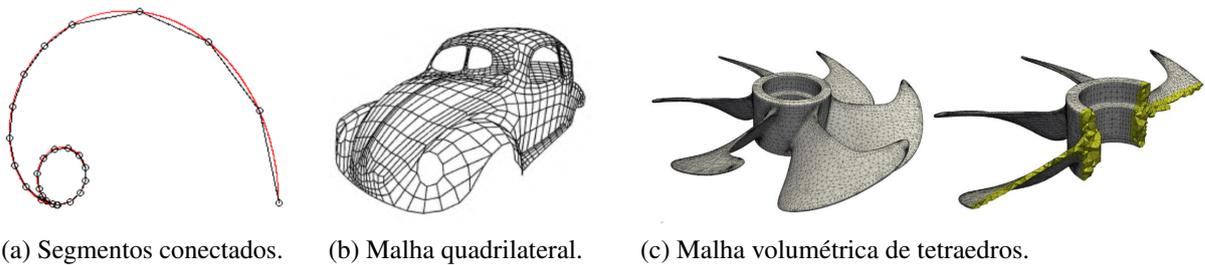
A B-Rep descrita anteriormente é utilizada para modelagem de geometrias planas. Neste caso, cada região possui uma única face plana, definida pelos ciclos do contorno externo e dos furos desta. Na modelagem de cascas tratadas neste trabalho, são utilizadas B-Reps similares as do caso plano, mas com algumas particularidades. A face da região possui geometria dada por uma superfície NURBS, que pode considerar inclusive aparos. As curvas de aparo são definidas nas arestas dos furos, que resultam em recortes fechados, e nas arestas das restrições, que definem recortes abertos, como naqueles apresentados na Figura 29.

Assim, cada região pode ser mapeada para o espaço paramétrico da NURBS que representa sua face, onde os vértices e arestas são definidos em coordenadas paramétricas  $\xi$  e  $\eta$ . Este mapeamento é importante na etapa de geração de malhas do modelo, pois permite que rotinas similares de geração de malhas sejam empregadas para ambos os casos de superfícies planas ou no espaço 3D.

### 3 GERAÇÃO DE MALHAS

Malhas são representações geométricas que definem um objeto a partir de um conjunto de formas geométricas simples e conectadas entre si, como segmentos no caso de curvas, triângulos e quadriláteros no caso de superfícies, tetraedros e hexaedros no caso de sólidos, entre outros. A Figura 35 mostra exemplos de malhas de várias dimensões. A descrição de uma malha consiste nos vértices, arestas e faces de seu conjunto de elementos. As malhas possuem importantes aplicações, como na definição de modelos numéricos em diversos problemas de engenharia e na renderização de cenários em computação gráfica.

Figura 35 – Exemplo de malhas em várias dimensões.



Fonte: Modificada de McLaurin e colaboradores<sup>51</sup> (a), Angel<sup>52</sup> (b) e Engvall e Evans<sup>6</sup> (c).

Em geral, as representações utilizadas na modelagem geométrica dos objetos não podem ser usadas diretamente nas aplicações. No contexto de sistemas CAE, o modelo B-Rep não pode ser empregado diretamente no analisador numérico baseado em MEF. Portanto, é necessário fazer a parametrização do modelo geométrico em uma malha de elementos finitos, possibilitando o processamento da simulação. Esta parametrização é realizada por técnicas de geração de malhas.

Os algoritmos de geração de malha vêm sendo desenvolvidos ao longo dos anos, e atingiram alto grau de maturidade em termos de robustez e eficiência computacional. Há uma grande variedade de algoritmos de geração de malhas implementados em programas livres<sup>53, 54, 55, 56</sup> e comerciais<sup>57, 58</sup>, além daqueles que são utilizados como componentes de outras aplicações<sup>59, 60</sup>.

Os dados de entrada utilizados por geradores de malhas são usualmente uma discretização inicial do modelo geométrico, como um conjunto segmentos retos ou curvos. A maior parte dos algoritmos constroem malhas que contém a discretização de entrada. Esta abordagem facilita a manipulação de modelos com múltiplas regiões, como o apresentado na Figura 32. Além de organizar os atributos das malhas em suas respectivas regiões correspondentes, também

garantem a compatibilidade entre malhas de diferentes regiões, que podem ser geradas considerando algoritmos distintos e tipos de elementos diferentes (como triângulos e quadriláteros). A discretização *a priori* das curvas do modelo pode ser realizada de forma manual, pelo usuário da aplicação, ou automática, utilizando algoritmos de subdivisão de curvas (geração de malhas unidimensionais).

Por outro lado, é importante notar que existem algoritmos de geração de malhas ditos automáticos, em que o próprio algoritmo discretiza as curvas do modelo geométrico durante o processamento da malha. Tais algoritmos não consideram uma discretização do contorno de entrada fixa, e geram malhas para todas as regiões do modelo ao mesmo tempo<sup>61, 62</sup>. Destaca-se que o presente trabalho foca no desenvolvimento de algoritmos que garantem a discretização de entrada fornecida.

As técnicas de geração de malha podem ser classificadas em relação à topologia da malha gerada. Algoritmos estruturados são aqueles onde há um padrão claro na topologia da malha, de modo que a quantidade de elementos da malha, bem como as informações de adjacências destes, são previamente conhecidas. Estes métodos são extremamente eficientes, e normalmente geram malhas com boa qualidade sem considerar etapas de melhoramento ou suavização. Entretanto, uma série de restrições são impostas aos dados de entrada do problema, limitando sua aplicação em caso de geometrias complexas e requerendo maior interação para utilização do algoritmo. É importante notar que existem trabalhos que buscam melhorar a robustez de tais técnicas através da decomposição do domínio, permitindo a geração de malhas estruturadas em cada partição separadamente<sup>63, 64</sup>.

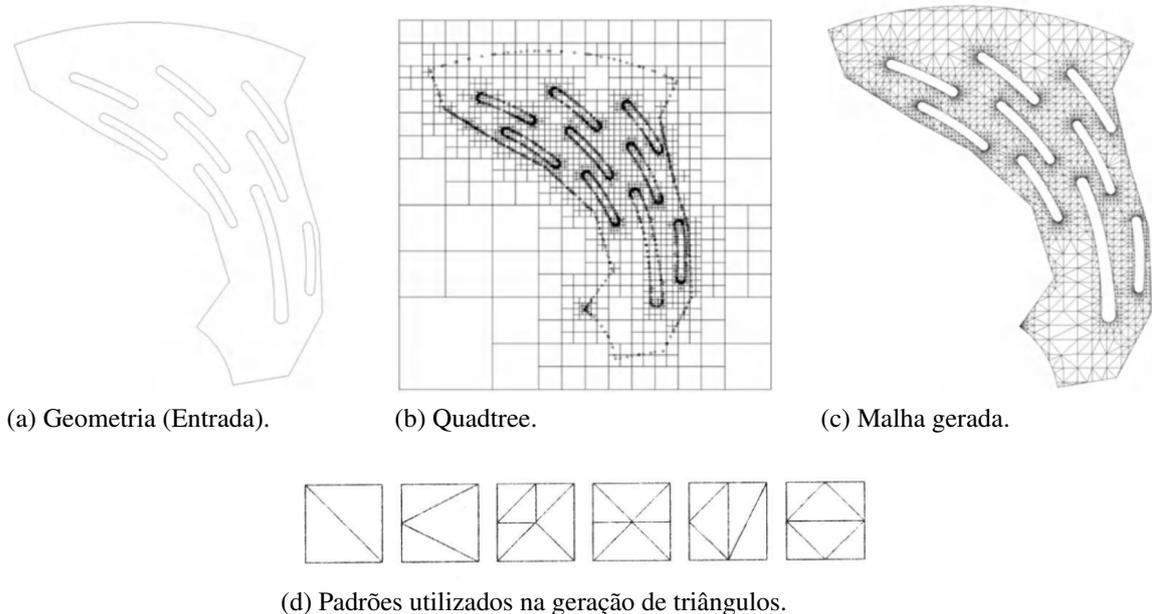
Os algoritmos não estruturados são aqueles onde não há padrão claro na topologia da malha gerada. O processo de geração é guiado pela inserção iterativa de elementos ou pontos no domínio. Estes algoritmos são adequados para geometrias arbitrárias, de qualquer complexidade. A resolução das malhas geradas são controladas através de funções de densidade de elementos, que são definidas no espaço do modelo de diversas formas, como por expressões matemáticas ou utilizando malhas auxiliares (*background meshes*)<sup>65</sup>. Destacam-se três categorias importantes de métodos não estruturados: triangulação de Delaunay, decomposição espacial e avanço de fronteira.

As técnicas de geração de Delaunay se baseiam no algoritmo de triangulação de Delaunay (TD), estudado no campo da geometria computacional<sup>66</sup>. O problema da triangulação consiste em determinar um conjunto de triângulos a partir de um conjunto de pontos no espaço.

Uma TD se destaca por possuir propriedades geométricas que evitam triângulos distorcidos. As técnicas de geração de Delaunay utilizam conceitos e algoritmos relacionados a TD no problema geração de malhas. Uma das vantagens desta abordagem é que alguns destes algoritmos apresentam garantias matemáticas em relação à qualidade das malhas geradas<sup>67, 68, 69, 70</sup>, o que não é observado nos outros métodos de geração. Contudo, é importante notar que tais garantias não se aplicam no caso algoritmos que respeitam uma discretização do contorno de entrada, onde refinamentos adicionais no contorno do modelo não podem ser realizados. Tais refinamentos são necessários para garantir a qualidade da malha no caso geral.

As técnicas de geração por decomposição espacial<sup>71, 72, 73, 74, 75, 76</sup> se baseiam no refinamento recursivo do espaço utilizando árvores de decomposição espacial, como *quadtrees* e *octrees*. Uma árvore é construída na região delimitada pelo modelo geométrico, e refinada de acordo com a resolução desejada da malha. As células que estão no interior do modelo geram elementos de modo eficiente, aplicando padrões pré-definidos. A geração dos demais elementos é mais complexa, e requer cálculos geométricos similares aos realizados nas técnicas de avanço de fronteira e Delaunay. A Figura 36 ilustra uma malha gerada por decomposição espacial.

Figura 36 – Exemplo de aplicação da técnica de geração de malhas via decomposição espacial.

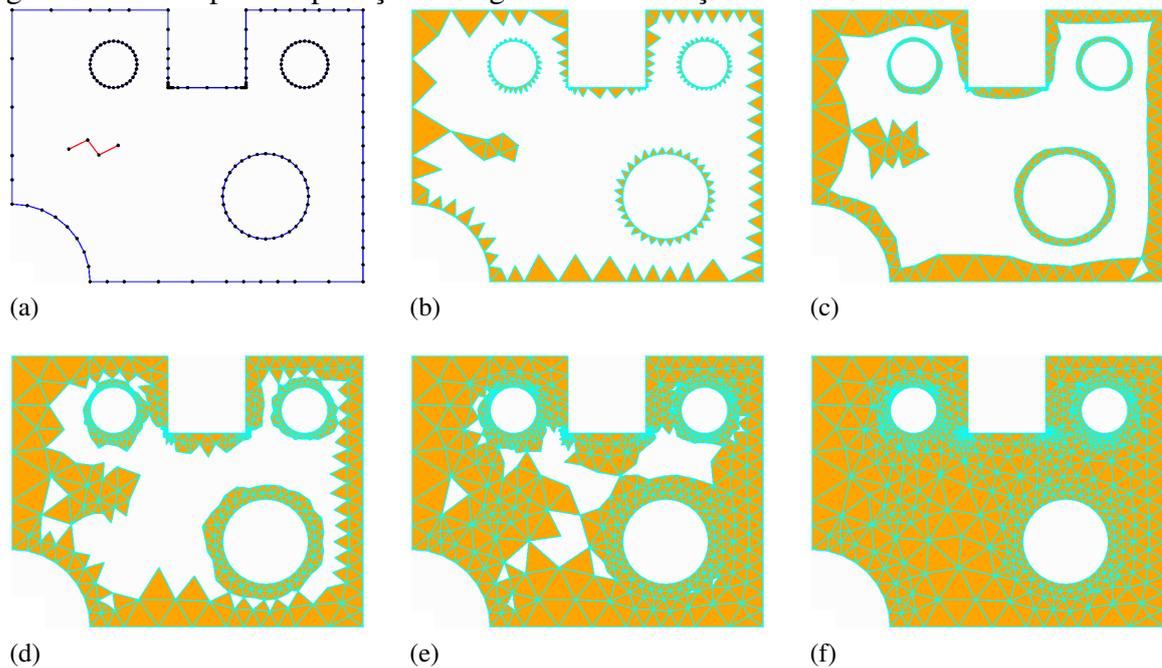


Fonte: Modificado de Frey & George<sup>65</sup> (a-c) e Cavalcante-Neto e colaboradores<sup>75</sup> (d).

As técnicas de avanço de fronteira<sup>77, 78, 79, 80, 81</sup> são caracterizadas pela geração consecutiva de triângulos a partir de uma aresta base, sendo esta pertencente a uma lista de arestas de fronteira. Inicialmente, a fronteira é definida pelas arestas pertencentes ao contorno do

problema, contendo as arestas externas, e pelas arestas pertencentes a furos e restrições internas. O processo de avanço consiste na geração de triângulos para cada aresta da fronteira. Neste processo, o triângulo gerado pode incidir em um vértice novo ou um vértice existente da fronteira, que deve ser atualizada adequadamente ao final de cada inserção de novo elemento. O processo iterativo cessa quando não há mais arestas na fronteira. A Figura 37 apresenta um exemplo de aplicação desta técnica.

Figura 37 – Exemplo de aplicação do algoritmo de avanço de fronteira.



Fonte: Elaborada pelo autor.

A maioria dos algoritmos de geração de malha foi desenvolvida considerando geometrias poligonais. Entretanto, existem trabalhos que abordam a geração de elementos de alta ordem, na geração e adaptação de malhas de alta ordem<sup>82, 83, 84</sup>, no estudo de métricas de qualidade e otimização de malhas<sup>85, 86, 87, 88, 89</sup>, e em problemas envolvendo malhas móveis<sup>90, 91</sup>. Triângulos de Bézier quadráticos têm sido usados no contexto de malhas móveis<sup>92</sup> e em aplicações elastostáticas e elastodinâmicas<sup>93</sup>. A geração de malhas válidas de triângulos de Bézier não racionais para domínios planos é apresentada em Mandad e Campen<sup>94</sup>. É importante notar que estes algoritmos são necessários no contexto de simulações numéricas com elementos de alta ordem, como nas formulações  $p$  de elementos finitos<sup>95</sup>.

Na análise isogeométrica, inicialmente buscou-se otimizar as etapas realizadas por sistemas CAE de modo a eliminar a etapa de geração de malhas do processo<sup>2, 3</sup>. Isto seria possível

se os modelos utilizados na modelagem e simulação do problema fossem iguais. Entretanto, a modelagem feita usando B-Reps exige a parametrização do domínio de análise, o que é difícil de se obter de forma automática no caso de geometrias complexas.

Outra questão relevante surge quando se consideram os modelos aparados. Na simulação de cascas via elementos de superfície, o domínio de análise é obtido diretamente da representação de superfície utilizada na modelagem, como superfícies NURBS<sup>96, 97</sup>. Entretanto, quando há aparos nas superfícies, é necessário considerar esquemas especiais de integração dos elementos na vizinhança dos aparos<sup>98, 4</sup>. Além disto, é necessário considerar pontos de controle ditos desativados, que não influenciam na geometria por terem suas regiões de contribuição recortadas, e portanto não definem graus de liberdade no problema numérico<sup>99, 98</sup>.

É importante mencionar que os problemas citados são recorrentes na área de análise numérica no contexto do MEF, e são resolvidos com o uso de técnicas de geração de malhas, que constroem modelos adequados para simulação de forma robusta e automática.

Existem trabalhos recentes abordando a construção de modelos NURBS e T-Splines para geometrias descritas usando B-Rep, tanto em casos 2D quanto 3D<sup>100, 101, 102, 103, 104, 105, 106</sup>. Contudo, essas abordagens não atendem a requisitos importantes, como discutido por Engvall e Evans<sup>5, 6</sup>:

- Ser aplicável a geometrias complexas, e.g., objetos com *genus* arbitrário ou objetos com grande variação de tamanho.
- Ser automática, ou seja, não requerer intervenção do usuário.
- Garantir a representação geométrica exata.
- Fornecer uma parametrização adequada para análise numérica.

Desta forma, o uso de triângulos e tetraedros racionais Bézier para geração de malhas isogeométricas não estruturadas tem sido explorado como uma alternativa para integrar modelos B-Rep na AIG, visando atender aos requisitos acima. A geração de triângulos de Bézier de alta ordem é descrita por Jaxon e Qian<sup>9</sup>. Uma malha triangular é usada para construir uma *rational Triangular Bézier Spline* (rTBS) com alta continuidade e geometria exata. A extensão deste trabalho para o caso 3D é apresentada por Xia e Qian<sup>48</sup>, onde são consideradas malhas com tetraedros de Bézier, e rTBS são geradas no caso de superfícies aparadas.

A geração automática de malhas triangulares de Bézier para problemas planos é estudada por Engvall e Evans<sup>5</sup>. O artigo foca na reutilização de rotinas existentes de geração de malhas poligonais para criação de malhas  $C^0$  de alta ordem e geometria exata. A extensão dessa

abordagem para a geração de malha de tetraedros, hexaedros, cunhas e pirâmides de Bézier é apresentada em outro trabalho<sup>6</sup>, onde a projeção Bézier é usada para reconstrução de superfícies NURBS e T-Spline usando triângulos ou superfícies de Bézier. Os autores também discutem o uso de elementos superparamétricos capazes de manter a geometria exata, uma característica relevante, pois não há garantia de que triângulos de Bézier possam representar exatamente superfícies NURBS e T-Spline de mesmo grau. Os mesmos autores também estudam métricas de qualidade para triângulos e tetraedros de Bézier racionais<sup>107, 108</sup>.

O algoritmo desenvolvido neste trabalho realiza a parametrização do modelo de análise para modelos B-Rep planos e modelos de superfícies aparadas, construindo malhas de elementos de superfícies e triângulos de Bézier racionais. Os algoritmos desenvolvidos se baseiam em vários procedimentos abordados anteriormente na área de geração de malhas, que são adaptados e modificados afim de construir malhas que, além de preservarem a geometria do problema, também produzem em elementos de alta ordem com boa qualidade. As malhas geradas são geometricamente exatas no caso plano, e aproximam com precisão a geometria de superfícies aparadas.

### **3.1 Geração de malhas unidimensional**

A parametrização da entrada é uma etapa de pré-processamento para os algoritmos conformes, onde a discretização de entrada é fornecida. Contudo, a qualidade da discretização do contorno afeta diretamente o desempenho do algoritmo de geração de malhas aplicado posteriormente. Portanto, esta etapa deve ser realizada de modo a garantir uma parametrização do contorno adequada no caso de geometrias complexas, com regiões com alta curvatura e regiões estreitas.

Diversas abordagens foram proposta na literatura para realização desta etapa no contexto de geração de malhas de elementos finitos<sup>109, 110, 65</sup>. Neste trabalho, optou-se por desenvolver novos algoritmos heurísticos simples e eficientes para curvas racionais utilizadas na análise isogeométrica. Estes algoritmos, apresentados a seguir, são capazes de gerar parametrizações adequadas do contorno mesmo no caso de geometrias complexas<sup>111</sup>. As rotinas são disponibilizadas no programa PMGen.

### 3.1.1 Discretização uniforme

A discretização uniforme é a forma mais simples de realizar a discretização do contorno. Dado um número de divisões ( $n_d$ ), o comprimento de discretização de uma curva paramétrica  $\mathbf{c}(t)$  definida no intervalo  $[t_a, t_b]$  é dado por:

$$L_u = \text{Arco}(t_a, t_b) / n_d, \quad (3.1)$$

onde o arco da curva é avaliado por:

$$\text{Arco}(t_a, t_b) = \int_{t_a}^{t_b} |\mathbf{c}'(t)| dt. \quad (3.2)$$

A Equação (3.2) pode ser avaliada numericamente por:

$$\text{Arco}(t_a, t_b) \cong \sum_{i=1}^{N_{int}} |\mathbf{c}'(\xi_i)| w_i J, \quad (3.3)$$

onde  $N_{int}$  é o número de pontos de integração,  $\xi_i$  e  $w_i$  são respectivamente os pontos de integração e o peso associado a cada ponto, e  $J$  é o jacobiano da transformação entre os sistemas onde são definidos os pontos de integração e o intervalo  $[t_a, t_b]$  da curva paramétrica  $\mathbf{c}(t)$ . Neste trabalho a quadratura de Gauss é usada para realizar a integração dos arcos.

O conjunto de valores paramétricos  $[t_1, \dots, t_{n_d}]$  que dividem o intervalo  $[t_a, t_b]$  em segmentos iguais são avaliados em sequência, usando o algoritmo apresentado no Quadro 1, que é baseado na busca binária, com tolerância  $\delta$ . Neste trabalho foi adotado  $\delta = 10^{-3}$ .

Quadro 1 – Algoritmo para determinação da coordenada paramétrica que define um segmento com o comprimento fornecido.

<b>Entrada:</b> Intervalo $[t_1, t_2]$ , comprimento desejado $l$ e a tolerância adotada $\delta$ .
<b>Saída</b> : $t_m \implies \text{Arco}(t_1, t_m) \cong l$ .
1 $t_i \leftarrow t_1$ ;
2 <b>repita</b>
3 $t_m \leftarrow (t_1 + t_2)/2$ ;
4 $l_m \leftarrow \text{Arco}(t_i, t_m)$ ;
5 <b>se</b> $l < l_m$ <b>então</b>
6 $t_2 \leftarrow t_m$
7 <b>senão</b>
8 $t_1 \leftarrow t_m$
9 <b>até</b> $ l - l_m  \leq l * \delta$ ;
10 <b>retorna</b> $t_m$

Fonte: Elaborada pelo autor.

Alternativamente, também é possível assumir um comprimento desejado  $L_{ut}$  de modo que todo o modelo apresente comprimento de aresta similar. Deste modo, o número de

divisões adotado pela aresta é calculado por:

$$n_d = \text{round}(\text{Arco}(t_1, t_2) / L_{ut}). \quad (3.4)$$

Critérios de discretização baseados na curvatura são apresentados em vários trabalhos <sup>112, 110, 51</sup>. Uma forma de controlar a curvatura das arestas é limitando a razão entre o arco e a corda do segmento:

$$\frac{\text{Arco}(t_a, t_b)}{\text{Corda}(t_a, t_b)} \leq AC_m, \quad (3.5)$$

onde  $AC_m$  é o limite adotado para razão arco-corda. A Figura 38 ilustra vários arcos circulares com diferentes valores de  $AC_m$ . Para arcos circulares, o valor da razão  $AC_m$  pode ser calculado em função do ângulo de curvatura máxima permitido ( $\theta_{max}$ ):

$$AC_m = \frac{\theta_{max}}{2 \sin(\theta_{max}/2)}. \quad (3.6)$$

Logo,

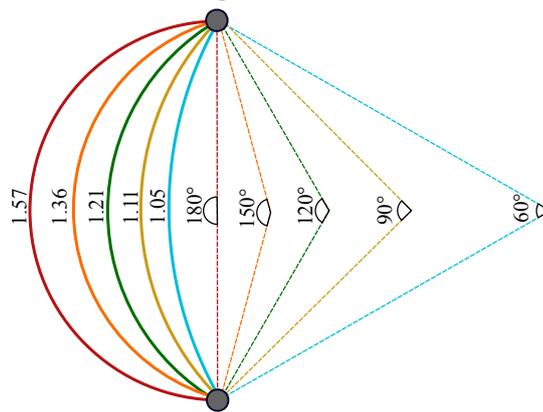
$$\frac{\text{Arco}(t_a, t_b)}{\text{Corda}(t_a, t_b)} \leq \frac{\theta_{max}}{2 \sin(\theta_{max}/2)}. \quad (3.7)$$

Além disto, a curvatura é aproximada pelos vetores tangentes nos pontos extremos do trecho <sup>80</sup>:

$$\frac{\mathbf{t}_g^a \cdot \mathbf{t}_g^b}{|\mathbf{t}_g^a| |\mathbf{t}_g^b|} \geq \cos \theta_{max}, \quad (3.8)$$

onde  $\mathbf{t}_g^a$  é o vetor tangente calculado em  $t_a$  e  $\mathbf{t}_g^b$  é o vetor tangente calculado em  $t_b$ .

Figura 38 – Razão arco/corda de vários segmentos circulares.



Fonte: Elaborada pelo autor.

As abordagens apresentadas são suficientes em vários casos, entretanto, podem requerer considerável intervenção do usuário em casos complexos. O algoritmo apresentado neste trabalho possibilita realizar a discretização do contorno de modelos complexos definindo apenas três parâmetros, conforme discutido na sequência.

### 3.1.2 Discretização recursiva

O algoritmo de discretização proposto consiste em realizar subdivisões recursivas no modelo. Uma discretização inicial é adotada considerando os *knots* das curvas NURBS. É importante notar que todos os pontos de continuidade  $C^0$  ao longo de cada curva estão necessariamente localizados nos *knots* com multiplicidade  $p - 1$ , onde  $p$  é o grau da curva. Em seguida, cada intervalo  $[t_1, t_2]$  é refinado de acordo com os seguintes critérios:

- **Comprimento máximo:** O comprimento do arco do segmento é maior que  $L_{max}$ .
- **Curvatura:** O ângulo de curvatura aproximado é maior que  $\theta_{max}$  (Equação (3.8)) ou a razão entre o comprimento do arco e da corda do segmento é maior que  $AC_m$  (Equação (3.7)).
- **Proximidade:** Se é detectada uma região próxima, pela aplicação do procedimento apresentado na Seção 3.1.3.
- **Variação do vetor tangente:** Se o módulo do vetor tangente variar excessivamente, como discutido na Seção 3.1.4.
- **Disparidade:** Se a metade da maior componente cartesiana do vetor formado pela corda do segmento for maior que o tamanho da aresta da *quadtree* apresentada na Seção 3.1.5, considerando as células que contenham os pontos extremos e o ponto médio da corda do segmento.

Se algum dos critérios for satisfeito, o segmento  $[t_1, t_2]$  é dividido no ponto médio  $t_m$  (aplicando o Algoritmo apresentado na Figura 1), e os dois intervalos  $([t_1, t_m]$  e  $[t_m, t_2])$  são verificados novamente. A fim de evitar uma discretização excessiva, é adotado um critério de parada caso  $Arco(t_1, t_2) * 0.75 < L_{min}$ . Deste modo, três parâmetros de entrada são usados ( $L_{max}$ ,  $\theta_{max}$ ,  $L_{min}$ ).

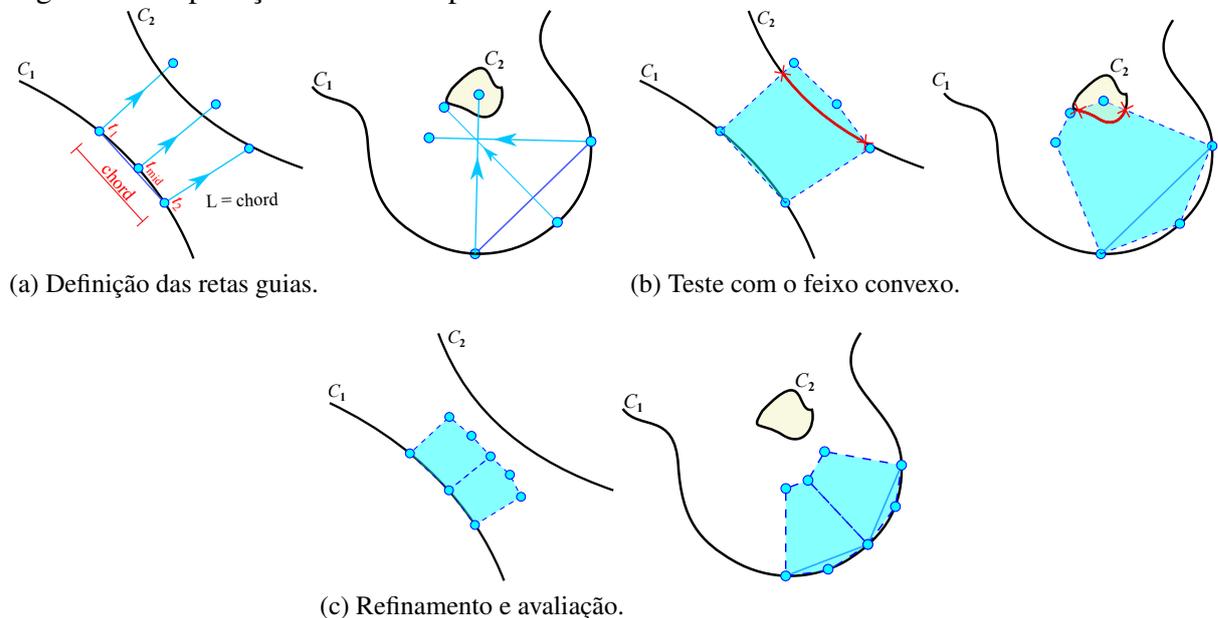
Após a aplicação da discretização recursiva, uma etapa de suavização é realizada para reduzir as diferenças entre o comprimento de arestas vizinhas. Esta etapa é importante para evitar uma grande variação no tamanho de arestas adjacentes, devido ao padrão de subdivisão realizado pelo algoritmo recursivo.

Vale ressaltar que o algoritmo proposto pode ser aplicado em um subconjunto de arestas do modelo, de forma que discretizações já existentes sejam mantidas. Na sequência, cada etapa do algoritmo recursivo é detalhada.

### 3.1.3 Detecção de regiões próximas

No segmento da curva  $[t_1, t_2]$ , três retas perpendiculares ao segmento são definidas nos pontos extremos,  $t_1$  e  $t_2$ , e no ponto  $t_m$ , que divide o segmento ao meio. As retas têm comprimento igual a  $Corda(t_1, t_2)$ . O sentido das retas é definido pelos ciclos em que o segmento é adjacente, sendo possível a aplicação do procedimento nos dois sentidos (e.g. arestas internas representando restrições). A área correspondente ao fecho convexo dos pontos extremos das retas é usado para testar interseções com o resto do modelo, exceto no trecho  $[t_1, t_2]$  da curva. A Figura 39 ilustra a aplicação do procedimento em dois casos.

Figura 39 – Aplicação do teste de proximidade.

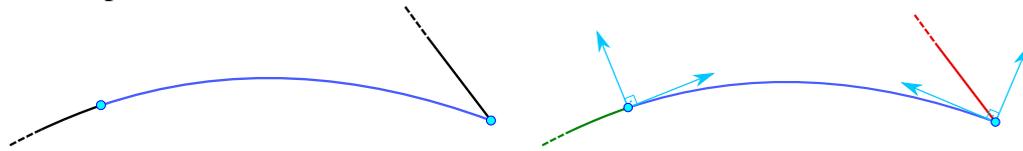


Fonte: Elaborada pelo autor.

O procedimento descrito apresenta repetição infinita quando o modelo possui cantos com pequenos ângulos. Para evitar o problema, são definidas retas ortogonais aos pontos extremos  $t_1$  e  $t_2$ , formando um quadrante em cada ponto extremo. É verificado se segmentos próximos aos pontos extremos estão situados nestes quadrantes, e, caso estejam, o processamento da proximidade é interrompido. A Figura 40 ilustra um exemplo com tal problema. É importante notar que os outros critérios de refinamento, como o de comprimento máximo e curvatura, ainda são avaliados e podem refinar o segmento.

É importante ressaltar que uma forma alternativa de realizar a verificação de proximidade é usando o eixo medial do modelo<sup>110, 113</sup>. Neste trabalho, optou-se por utilizar a estratégia proposta por ser mais simples em comparação as alternativas da literatura.

Figura 40 – Detecção de cantos com ângulos pequenos. O segmento à direita interrompe a etapa de proximidade.



Fonte: Elaborada pelo autor.

### 3.1.4 Variação do vetor tangente

A parametrização existente nas curvas NURBS utilizadas no modelo afeta diretamente a qualidade dos elementos de alta ordem da fronteira. A qualidade da parametrização é avaliada pela razão entre os módulos mínimo ( $\mathbf{t}_g^{min}$ ) e máximo ( $\mathbf{t}_g^{max}$ ) do vetor tangente da curva ao longo do segmento:

$$t_v = \frac{|\mathbf{t}_g^{min}|}{|\mathbf{t}_g^{max}|} \quad (3.9)$$

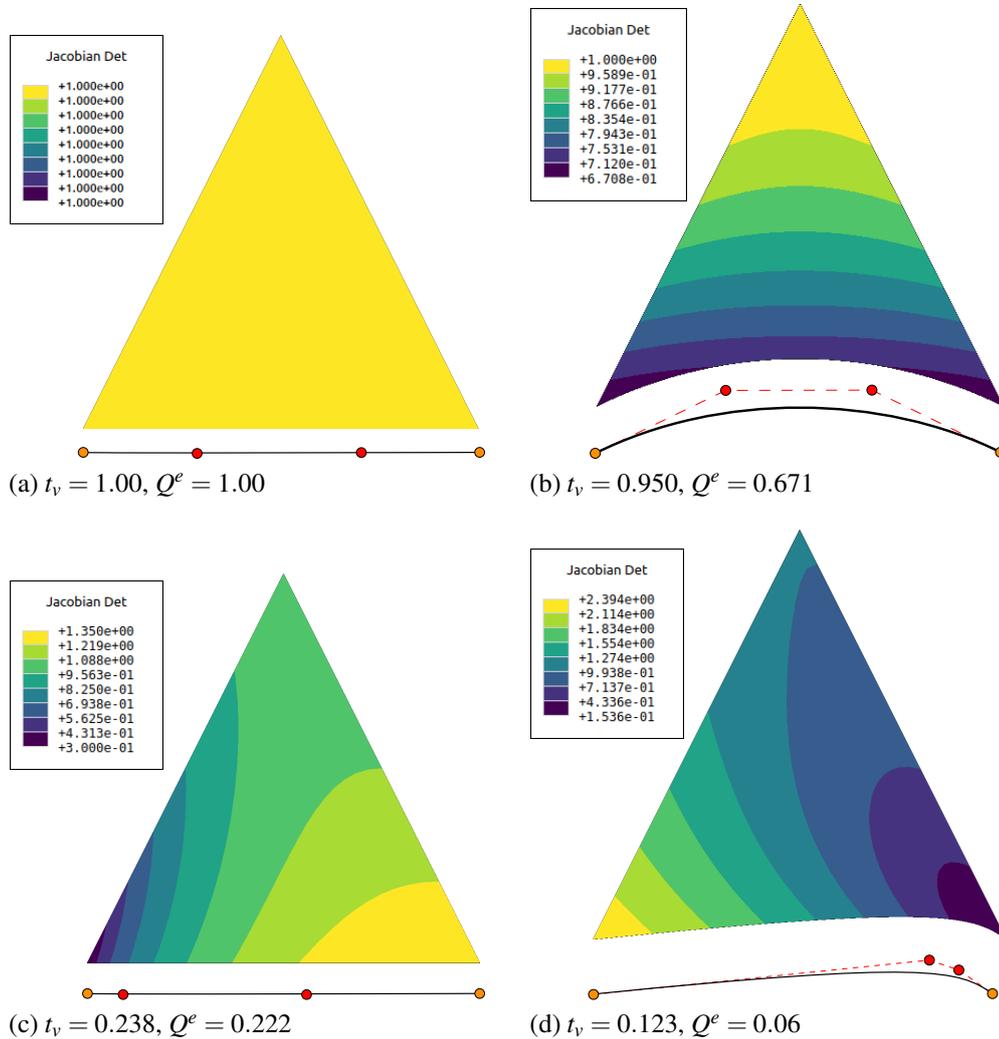
O segmento é refinado caso  $t_v < 0.5$ . Vale notar que tal métrica é semelhante à métrica de qualidade *scaled Jacobian* usada pra elementos de alta ordem<sup>82, 114, 115, 116, 5</sup>, avaliada por:

$$Q^e = \frac{\det(J^e)_{min}}{\det(J^e)_{max}} \quad (3.10)$$

onde  $\det(J^e)_{min}$  e  $\det(J^e)_{max}$  são, respectivamente, o menor e maior valor do determinante da matriz Jacobiana do elemento. Uma grande variação de  $|J^e|$  prejudica o desempenho do elemento, mesmo que  $|J| > 0$ . A Figura 41 ilustra o efeito que a parametrização de arestas curvas têm sob a variação do determinante da matriz Jacobiana de elementos de alta ordem.

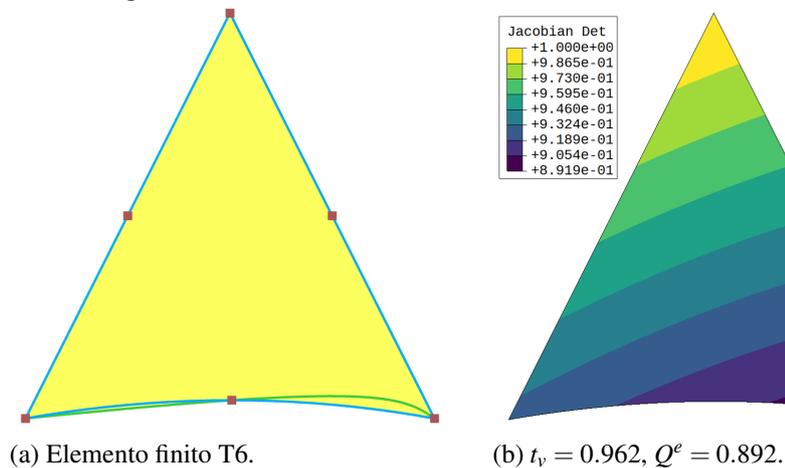
É importante notar que este problema não ocorre no caso de malhas de elementos finitos. As arestas curvas dos elementos Lagrangianos de alta ordem aproximam a geometria do modelo sem herdar sua parametrização. Normalmente, os nós internos destas arestas são avaliados de modo que sejam distribuídos uniformemente ao longo da curva no espaço Cartesiano. Na Figura 42, um elemento finito T6 é construído sobre uma aresta com parametrização ruim. O nó central da aresta base é posicionado na metade do comprimento da curva. Embora a aresta base apresente parametrização ruim, a aresta Lagrangiana, que é uma aproximação da curva do modelo geométrico, apresenta excelente parametrização. Além disto, o elemento finito construído também possui excelente qualidade.

Figura 41 – Efeito da parametrização de arestas curvas sobre a variação do determinante da matriz Jacobiana em elementos de alta ordem. Os casos (a) e (b) apresentam curvas com boa parametrização, enquanto que os casos (c) e (d) apresentam curvas com parametrização ruim.



Fonte: Elaborada pelo autor.

Figura 42 – Elemento finito T6 de boa qualidade construído sobre uma aresta com parametrização ruim (Figura 41d).



Fonte: Elaborada pelo autor.

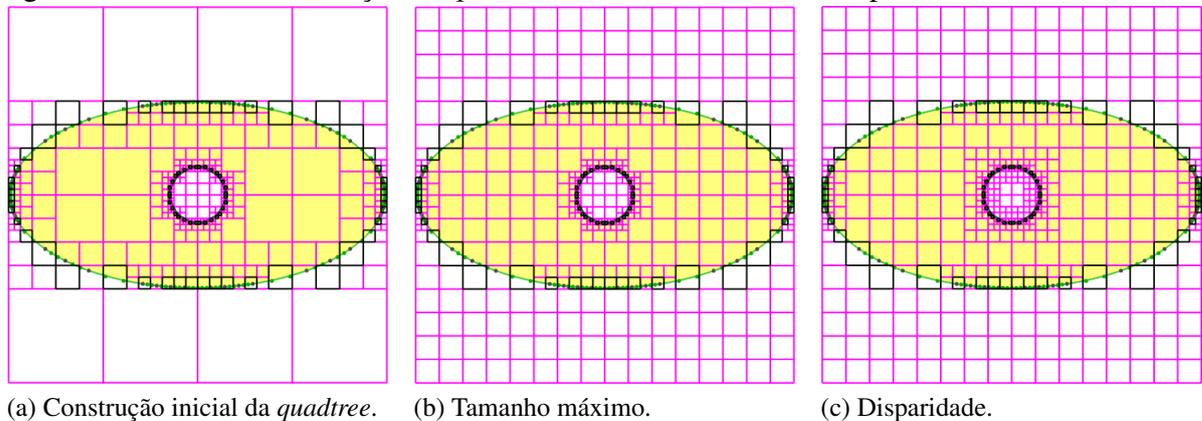
### 3.1.5 Disparidade

O critério de disparidade tem como objetivo reduzir as diferenças entre o comprimento de segmentos próximos. Para isto, uma *quadtree* é construída em função do tamanho das arestas, seguindo a mesma regra apresentada em Miranda et al.<sup>78</sup>:

- **Construção:** O tamanho inicial da *quadtree* é dada pela caixa delimitadora do modelo. Assim, para cada segmento do modelo, é avaliado o comprimento de sua corda  $s_l$  e o ponto médio de sua corda  $s_m$ . A *quadtree* é refinada caso  $q_{side} > s_l$ , onde  $q_{side}$  é o lado da folha que contém o ponto  $s_m$ .
- **Tamanho máximo:** O maior tamanho ( $q_{max}$ ) das células que foram refinadas pelas arestas do modelo, na etapa anterior, é usado para refinar as demais células da árvore, até que  $q_{max}$  seja o tamanho máximo em todas as células folhas.
- **Disparidade:** Refinamentos são aplicados<sup>75, 78, 79</sup> até que a diferença entre o tamanho de células adjacentes seja 1.

A Figura 43 ilustra a aplicação dessas etapas na construção da *quadtree*.

Figura 43 – Processo de criação da *quadtree* usada no critério de disparidade.



Fonte: Elaborada pelo autor.

Usando a *quadtree*, o segmento  $[t_1, t_2]$  é refinado caso a metade da maior componente do vetor formado pela corda do segmento seja maior que o menor lado das células da *quadtree* que contenham os pontos extremos e médio da corda do segmento.

É importante notar que a *quadtree* é construída após uma primeira aplicação dos critérios de comprimento máximo, curvatura e proximidade. Em seguida, o modelo continua a ser refinado considerando todos os critérios. Vale ressaltar que as arestas do modelo previamente discretizadas, mas que não são discretizadas na execução do algoritmo, também contribuem para

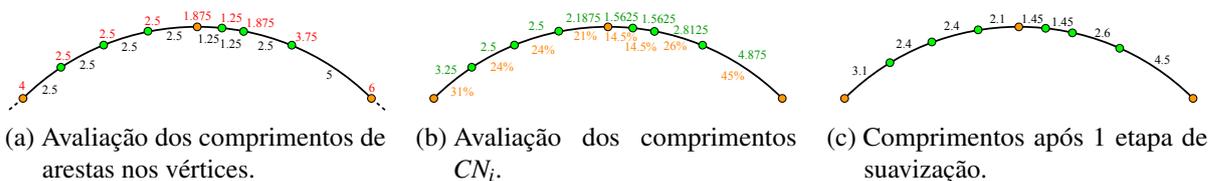
na construção da *quadtree*.

### 3.1.6 Suavização

A discretização realizada nas etapas anteriores apresenta o inconveniente de não garantir uma boa transição entre arestas adjacentes. Internamente, a discretização obtida em cada trecho pode apresentar arestas com diferenças de até 50% no tamanho. Diferenças maiores podem ser observadas nas arestas adjacentes aos vértices do modelo topológico. Uma transição ruim entre arestas pode prejudicar a etapa de geração de malhas, diminuindo a qualidade dos elementos gerados. Por este motivo é aplicado o procedimento de suavização descrito a seguir.

O método proposto começa com a avaliação do comprimento de cada segmento. Nos vértices são avaliados os valores médios de comprimento de aresta, considerando as arestas adjacentes, procedimento semelhante à avaliação das normais dos vértices no contexto de computação gráfica. Em cada aresta, é avaliado um novo comprimento de aresta  $CN_i$ , dado pela média dos comprimentos calculados anteriormente em seus vértices adjacentes. Note que os valores  $CN_i$  não somam, necessariamente, os comprimentos das curvas do contorno inicial. Assim, em cada curva do modelo geométrico, é avaliado um novo comprimento  $CN$  dado pelo somatório de cada  $CN_i$  de seus segmentos de subdivisão. A razão  $CN_i/CN$  de cada segmento é utilizado para avaliar o novo tamanho deste, em relação ao tamanho real da curva. Deste modo, os vértices internos de cada segmento são reposicionados, utilizando o algoritmo descrito no Quadro 1. A heurística descrita busca reduzir as diferenças no tamanho de segmentos adjacentes.

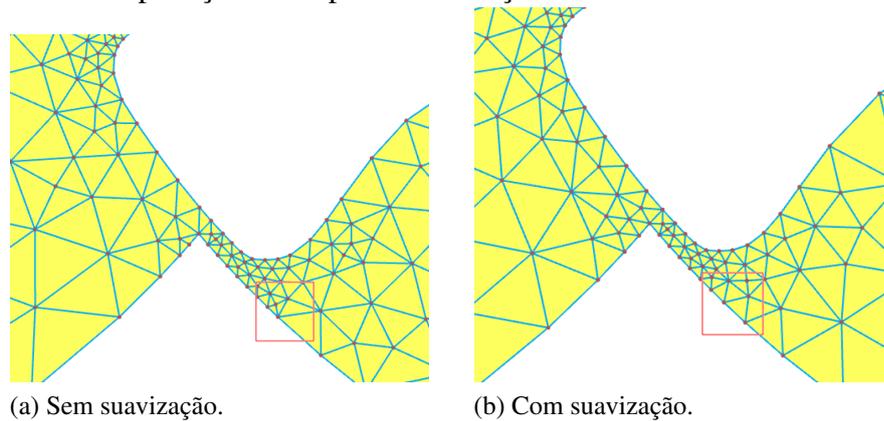
Figura 44 – Processo de suavização da discretização. Os vértices de cor laranja não são movidos.



Fonte: Elaborada pelo autor.

O processo descrito é aplicado duas vezes. Vale ressaltar que apenas os vértices gerados durante o processo de discretização são movidos. A Figura 44 ilustra a aplicação do algoritmo de suavização. O efeito da etapa de suavização na malha gerada é ilustrado na Figura 45. Pode-se observar que há uma melhor transição dos comprimentos dos segmentos do contorno e do tamanho de seus elementos adjacentes, quando utilizada a suavização proposta.

Figura 45 – Efeito da aplicação da etapa de suavização.



Fonte: Elaborada pelo autor.

### 3.2 Geração de malhas estruturadas

As técnicas de geração de malhas estruturadas são bastante utilizadas em aplicações de Engenharia. Apesar de terem aplicação mais restrita que as técnicas de geração de malhas não estruturadas, seu emprego torna-se atrativo em casos onde não há grande complexidade na geometria ou na forma de refinamento desejada do modelo, podendo gerar malhas de alta ordem com boa qualidade e baixo custo computacional.

Métodos de geração de malhas estruturadas vem sendo desenvolvidos no contexto da análise isogeométrica, como técnicas de parametrização de domínio usando funções harmônicas<sup>117</sup>, grade elíptica<sup>118</sup> e suavização elástica geometricamente não linear, considerando grandes deformações<sup>106</sup>.

Neste trabalho, optou-se pelo uso da técnica de Mapeamento Transfinito<sup>119, 120</sup> para geração de elementos de Bézier, no contexto de análise estrutural. Embora o uso de tal técnica para gerações de superfícies NURBS seja conhecido e utilizado em sistemas CAD, não foi encontrado na literatura o uso desta para geração de malhas de elementos de Bézier racionais com geometria exata. É válido ressaltar que, apesar do mapeamento transfinito ser menos robusto em comparação a outras técnicas propostas recentemente, em muitos casos seu uso permite obter malhas de boa qualidade de forma simples e eficiente. A técnica é detalhada a seguir.

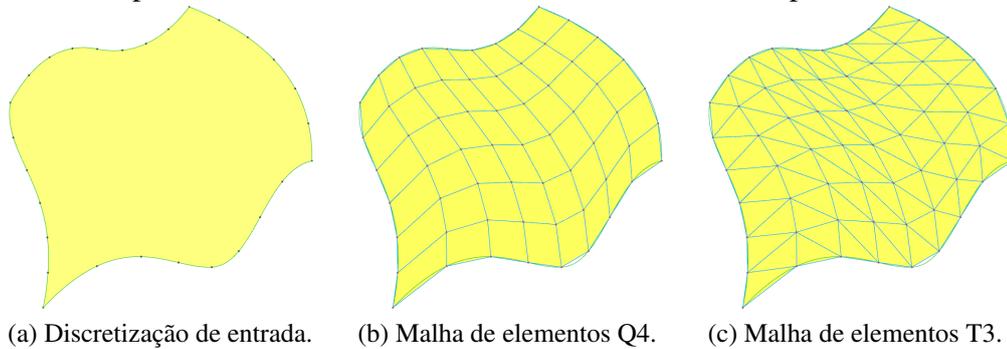
#### 3.2.1 Mapeamento Transfinito Bilinear

O mapeamento transfinito bilinear<sup>119, 121</sup> é uma técnica de geração de malhas de superfícies planas e no espaço tridimensional. As malhas são obtidas utilizando uma divisão

regular no espaço paramétrico  $(\xi, \eta)$  de uma superfície de Coons. Os dados de entrada consistem em quatro curvas paramétricas e a subdivisão (ou discretização) desejada em cada par de curvas. Define-se então uma grade no espaço paramétrico, que é utilizada para gerar a malha.

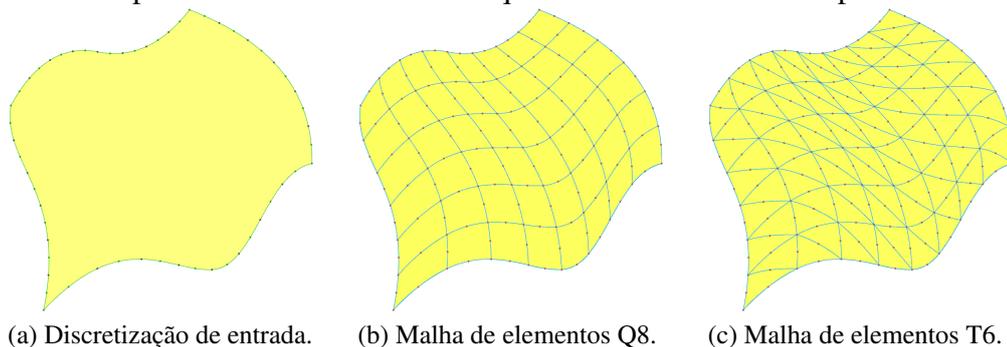
No caso de elementos finitos quadrilaterais, como os elementos Q4 e Q8, os nós dos elementos são definidos na grade paramétrica e mapeados para o espaço cartesiano usando a Equação (2.41). No caso de elementos triangulares, a grade paramétrica é dividida de acordo com padrões definidos *a priori*, obtendo-se uma nova grade com padrão triangular. Assim, elementos triangulares, como os elementos finitos T3 e T6, são obtidos de forma análoga ao realizado para os elementos quadrilaterais. A Figura 46 e Figura 47 ilustram a técnica sendo utilizada na geração de malhas de elementos finitos lineares e quadráticos.

Figura 46 – Exemplo de malhas de elementos lineares obtidas via mapeamento transfinito.



Fonte: Elaborada pelo autor.

Figura 47 – Exemplo de malhas de elementos quadráticos obtidas via mapeamento transfinito.



Fonte: Elaborada pelo autor.

É importante notar que esta técnica também pode ser utilizada para geração de malhas de superfície no espaço tridimensional. Esta abordagem é utilizada quando a geometria do problema é modelada usando superfícies de Coons. Neste trabalho, a técnica será aplicada de forma direta em modelos planos, e em alguns modelos específicos de superfícies 3D que possam

ser representados por superfícies de Coons. No caso geral de superfícies aparadas, a técnica é adaptada para aumentar o nível de aproximação geométrica dos elementos gerados, uma vez que as superfícies de Coons não são capazes de representar exatamente superfícies NURBS. Tais adaptações são discutidas na Seção 3.5.1.

### 3.2.2 Mapeamento Transfinito na AIG

Mapeamentos transfinitos podem ser realizados de forma a gerar malhas que representem exatamente a geometria do modelo. No caso de malhas planas, esta propriedade é satisfeita quando as curvas de contorno são representadas de forma exata.

Em elementos finitos, apesar da superfície de Coons interpolar exatamente as curvas de contorno utilizadas em sua definição, as funções de aproximação polinomiais da geometria dos elementos não podem representar exatamente todas as formas geométricas utilizadas na modelagem, como por exemplo curvas cônicas. Por outro lado, malhas de elementos triangulares e quadrilaterais de Bézier racional podem representar de forma exata as cônicas e outras curvas racionais.

Considerando uma região do modelo B-Rep definida por quatro curvas de contorno NURBS, os requisitos para aplicação do mapeamento, no caso quadrilateral, são que cada par de curvas  $\{\mathbf{c}_1, \mathbf{c}_2\}$  e  $\{\mathbf{g}_1, \mathbf{g}_2\}$  devem possuir mesmo grau e o mesmo vetor de *knots*. No caso triangular, todas as curvas devem ter mesmo grau. Vale ressaltar que tais restrições não representam limitação séria para utilização da técnica. Por exemplo, se as curvas  $\mathbf{c}_1$  e  $\mathbf{c}_2$  apresentam grau e *knots* diferentes, pode-se utilizar o algoritmo de *inserção de knot* para compatibilizar os *knots* e o algoritmo da *elevação de grau* para compatibilizar os graus das curvas, sem modificar a geometria do modelo.

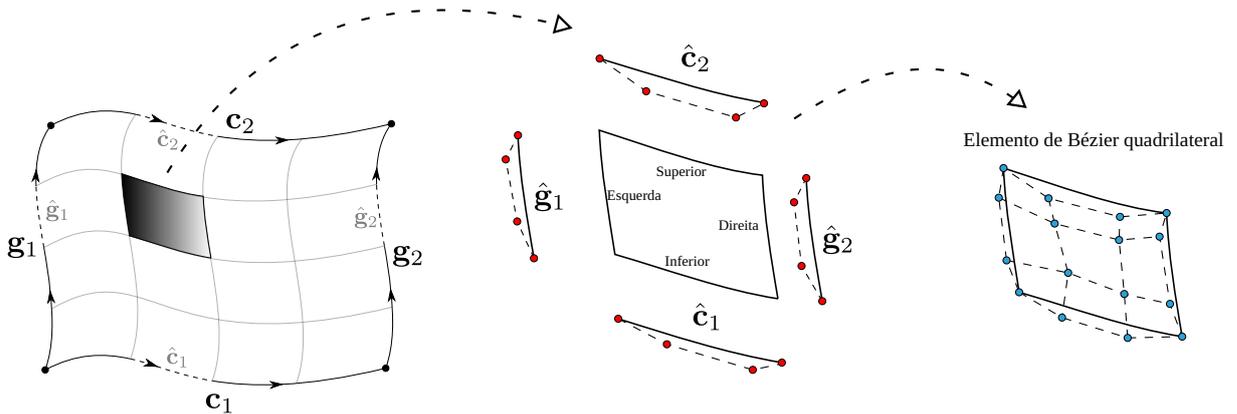
Com o grau e discretização das curvas definidos, a extração de Bézier é utilizada para obter curvas de Bézier em cada *knot span* da NURBS. Os pontos de controle dos elementos são avaliados considerando os pares de curvas referentes a cada *knot span* das curvas  $\mathbf{c}_1$ ,  $\mathbf{c}_2$ ,  $\mathbf{g}_1$  e  $\mathbf{g}_2$ , utilizando a Equação (2.43). Em cada elemento isogeométrico, as curvas de contorno são representadas pelas curvas de Bézier extraídas,  $\hat{\mathbf{c}}_1$ ,  $\hat{\mathbf{c}}_2$ ,  $\hat{\mathbf{g}}_1$  e  $\hat{\mathbf{g}}_2$ . Assim, os pontos de controle do elemento de Bézier quadrilateral, de grau  $p \times q$ , são computados por:

$$\begin{aligned} \mathbf{P}_{ij} &= (1 - \bar{\eta}) \mathbf{p}_i^{\hat{\mathbf{c}}_1} + \bar{\eta} \mathbf{p}_i^{\hat{\mathbf{c}}_2} + (1 - \bar{\xi}) \mathbf{p}_j^{\hat{\mathbf{g}}_1} + \bar{\xi} \mathbf{p}_j^{\hat{\mathbf{g}}_2} - \mathbf{p}_M, \\ \mathbf{p}_M &= (1 - \bar{\xi}) \bar{\eta} \mathbf{p}_1^{\hat{\mathbf{c}}_2} + \bar{\xi} \bar{\eta} \mathbf{p}_{p+1}^{\hat{\mathbf{c}}_2} + \bar{\xi} (1 - \bar{\eta}) \mathbf{p}_{p+1}^{\hat{\mathbf{c}}_1} + (1 - \bar{\xi}) (1 - \bar{\eta}) \mathbf{p}_1^{\hat{\mathbf{c}}_1}, \end{aligned} \quad (3.11)$$

onde  $\bar{\xi} = (i - 1)/p$  e  $\bar{\eta} = (j - 1)/q$ .  $\mathbf{p}_i^{\hat{\mathbf{c}}_1}$ ,  $\mathbf{p}_i^{\hat{\mathbf{c}}_2}$ ,  $\mathbf{p}_j^{\hat{\mathbf{g}}_1}$  e  $\mathbf{p}_j^{\hat{\mathbf{g}}_2}$  são os pontos de controle das curvas de

Bézier  $\hat{c}_1$ ,  $\hat{c}_2$ ,  $\hat{g}_1$  e  $\hat{g}_2$ . É importante observar que o procedimento deve ser aplicado no espaço de projeção  $\mathbb{R}^{d+1}$  no caso de curvas racionais, semelhante aos demais algoritmos (e.g. *elevação de grau*). A Figura 48 ilustra o procedimento de avaliação dos elementos quadrilaterais.

Figura 48 – Mapeamento transfinito aplicado para geração de elementos quadrilaterais de Bézier racional.



Fonte: Elaborada pelo autor.

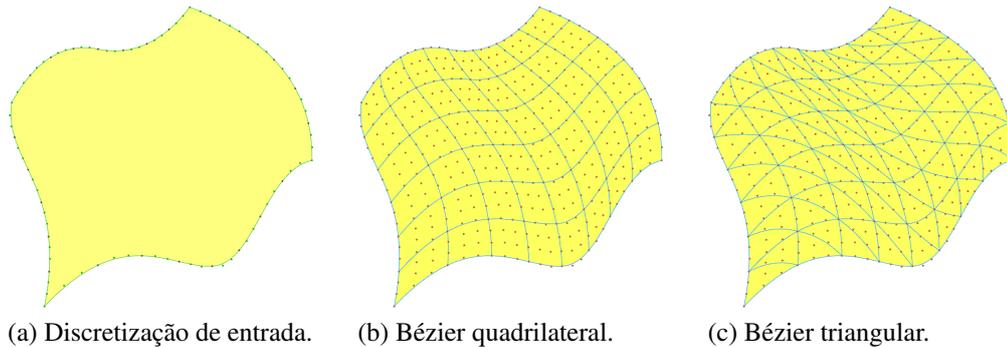
Portanto, o cálculo dos pontos de controle de elementos de Bézier quadrilaterais consiste na aplicação de formulas algébricas semelhantes às utilizadas no caso de elementos finitos. Por outro lado, o mesmo não ocorre no caso dos elementos de Bézier triangulares.

No caso de malhas triangulares, não é possível obter triângulos de Bézier de mesmo grau das curvas de contorno de forma direta. De fato, triângulos de Bézier de grau  $(p+q)$  podem ser obtidos a partir de superfícies de Bézier de grau  $(p \times q)^{40, 122}$ . Entretanto, na metodologia de modelagem desenvolvida neste trabalho, os graus das curvas de contorno são iguais aos graus dos elementos gerados.

Logo, adota-se um procedimento de ajuste para determinação dos triângulos. Este procedimento considera que alguns pontos de controle são conhecidos e os demais são calculados utilizando a Equação (2.49). Os pontos de controle são determinados da seguinte forma. Primeiro, os pontos de controle de canto são computados diretamente pela Equação (2.41), pois são interpoladores. Em seguida, os pontos de controle de arestas ortogonais no espaço paramétrico, ou seja, arestas com  $\xi$  ou  $\eta$  constantes, são obtidas de forma semelhante ao realizado nos elementos quadrilaterais, como descrito por Barroso e colaboradores<sup>123</sup>. No caso das arestas que não sejam ortogonais, é feito um ajuste dos pontos de controle das curvas de Bézier usando uma amostra avaliada de forma uniformemente distribuída no espaço paramétrico da superfície de

Coons. Finalmente, o mesmo é realizado no caso dos pontos de controle do interior do elemento, que não são adjacentes a nenhuma aresta. A Figura 49 mostra uma aplicação das técnicas de mapeamento transfinito isogeométrico descritas.

Figura 49 – Exemplo de malhas de elementos cúbicos isogeométricos obtidos via mapeamento transfinito.



Fonte: Elaborada pelo autor.

Vale notar que os procedimentos descritos geram malhas triangulares e quadrilaterais de geometria exata no caso de geometrias planas. No caso de superfícies no espaço, onde o modelo geométrico é dado pela superfícies de Coons, as malhas quadrilaterais são sempre exatas. Já as malhas triangulares aproximam a geometria da superfície de Coons no caso geral, mas podem representar a geometria exata da superfície caso o grau considerado na malha seja suficiente. Por exemplo, uma superfície cilíndrica, que requer no mínimo grau  $(2 \times 1)$  para ser modelada, pode ser representada por triângulos de Bézier racionais cúbicos ou de ordem superior, mas não por triângulos quadráticos.

### 3.3 Geração de malhas não estruturadas para modelos planos

O algoritmo proposto busca gerar malhas triangulares de Bézier de grau arbitrário, geometria exata e boa qualidade. As malhas são conformes às curvas de contorno e do interior do modelo geométrico. Os dados de entrada do algoritmo são uma região plana do modelo B-Rep e seus dados de parametrização, que consistem no grau da malha gerada e nas informações de subdivisão de cada curva da região. A subdivisão (ou discretização) de cada curva é fornecida em um vetor de valores paramétricos, definidos nos limites de cada curva. Duas restrições são consideradas para garantir a representação exata do modelo. Primeiro, o grau da malha deve ser maior ou igual ao maior grau das curvas da região. A segunda restrição é que a discretização de entrada das curvas deve conter todos os *knots* das curvas NURBS do modelo geométrico,

permitindo que a *inserção de knot* e *extração de Bézier* sejam utilizadas para representar as curvas NURBS por curvas de Bézier.

Uma vez disponíveis os dados de parametrização, os segmentos Bézier são obtidos da seguinte forma:

- 1) Os graus das curvas NURBS são elevados ao grau desejado da malha; as curvas são subdivididas usando inserção de *knots*.
- 2) Segmentos de Bézier são obtidos usando extração Bézier. A orientação de cada segmento Bézier é herdada de seu ciclo correspondente na B-Rep.

O algoritmo de geração de malha consiste em quatro etapas:

- 1) **Geração de malha poligonal:** Esta etapa constrói a malha inicial de triângulos lineares que define a topologia da malha. Aqui, todos os segmentos curvos da fronteira de entrada são representados por suas respectivas cordas.
- 2) **Elevação de grau e restauração da fronteira.** Esta etapa eleva o grau da malha linear para o grau desejado, e restaura a geometria exata das curvas de entrada.
- 3) **Tratamento de singularidades.** Esta etapa modifica localmente a topologia da malha para evitar elementos com singularidades.
- 4) **Suavização de alta ordem.** Esta etapa melhora a qualidade dos elementos próximos às arestas curvas através do procedimento de suavização de pesos e coordenadas dos pontos de controle discutido na Seção 3.3.4.

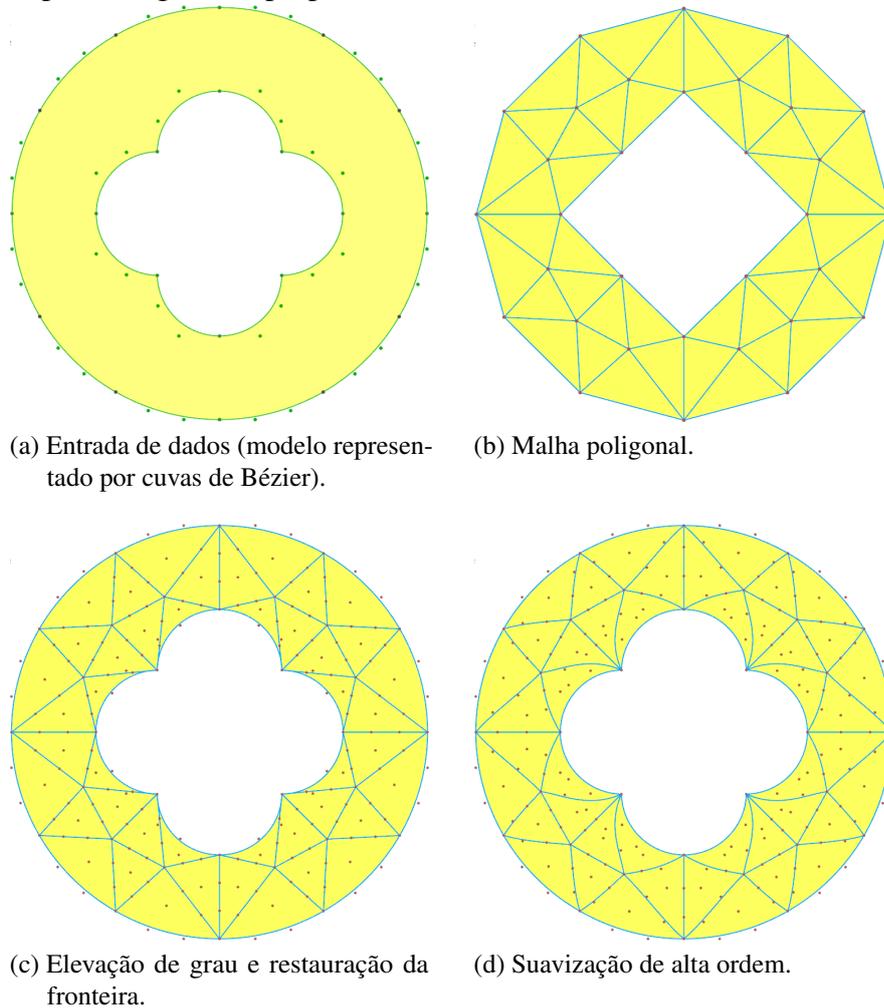
Muitos algoritmos de malha estruturada geram elementos de alta ordem diretamente<sup>119, 118, 117</sup>. No entanto, no contexto de malhas não estruturadas, as malhas de alta ordem são comumente construídas a partir de malhas poligonais<sup>82, 124, 88, 5</sup>.

A geração de malhas isogeométricas de triângulos de Bézier racionais, similar ao algoritmo proposto, foi desenvolvida em outros algoritmos encontrados na literatura<sup>9, 5, 7</sup>. No entanto, destacam-se importantes contribuições na abordagem desenvolvida nesta tese. Consideramos os segmentos curvos na avaliação da função densidade utilizada na etapa de geração da malha poligonal (Passo 1), que usualmente é processada sem receber nenhuma informação do modelo geométrico, além das cordas dos segmentos. Além disso, o algoritmo inclui um procedimento (Passo 3) para evitar a geração de elementos de alta ordem com singularidades, um problema que não é discutido em outros trabalhos relacionados a modelos planos.

Além disso, a etapa de suavização de alta ordem (Passo 4) do algoritmo proposto melhora a qualidade da malha, atuando não apenas nas coordenadas dos pontos de controle,

mas também nos pesos destes, o que de fato foi feito no contexto de malhas 3D<sup>6</sup>, mas não em malhas 2D, onde apenas os pesos dos pontos de controle foram suavizados<sup>5</sup>. Outra vantagem da abordagem proposta é que essa etapa é executada de forma eficiente, considerando subconjuntos de elementos da malha localizados na vizinhança dos segmentos curvos. Um algoritmo para a avaliação destes conjuntos disjuntos de elementos é apresentado em detalhes. Não se verifica na literatura outra abordagem que propõe a suavização local das coordenadas e pesos dos pontos de controle baseadas na suavização por deformação da malha.

Figura 50 – Etapas do algoritmo proposto.



Fonte: Elaborada pelo autor.

Finalmente, ressalta-se que a implementação da técnica proposta é otimizada, escrita em linguagem C++, sendo superior à implementação apresentada em Engvall e Evans<sup>5</sup>, escrita em MATLAB, em ordens de magnitude em relação ao tempo de processamento. O desempenho computacional é uma característica particularmente importante em aplicações que processam

malhas repetidas vezes, como na otimização de forma estrutural<sup>8</sup> e no contexto de análises adaptativas<sup>80</sup>. A Figura 50 ilustra as etapas do algoritmo proposto para geração de uma malha cúbica. Cada etapa do algoritmo é detalhada a seguir.

### 3.3.1 Geração de malha poligonal

Neste trabalho, a etapa de geração da malha poligonal é realizada por um algoritmo de avanço de fronteira baseado na técnica desenvolvida por Miranda et al.<sup>78</sup>, e modificada neste trabalho. As modificações propostas foram apresentadas inicialmente em Barroso et al.<sup>111</sup> e são detalhadas a seguir. Nesta técnica, a função de densidade dos elementos é definida por uma *quadtree* auxiliar construída em função dos tamanhos das arestas do contorno de entrada, como apresentado na Seção 3.1.5.

Contudo, este critério de refinamento é inadequado no caso de malhas de alta ordem, quando há grande diferença entre os comprimentos da corda e do arco dos segmentos, como ilustrado na Figura 51. Portanto, a seguinte modificação é realizada para avaliação de uma função densidade mais adequada para geração de malhas de alta ordem. A *quadtree* é refinada se

$$q_{side} > l_{ce}, \quad l_{ce} = s_l - \beta d_l \quad (3.12)$$

onde  $s_l = \|\mathbf{p}_2 - \mathbf{p}_1\|$ , sendo  $\mathbf{p}_1$  e  $\mathbf{p}_2$  os pontos extremos do segmento,  $\beta$  é um coeficiente de ajuste, e  $d_l$  é a flecha central do segmento, calculada por:

$$d_l = \hat{\mathbf{n}} \cdot (\mathbf{c}(t_m) - \mathbf{s}_m), \quad (3.13)$$

onde  $\hat{\mathbf{n}}$  é um vetor unitário perpendicular à corda do segmento, dado por:

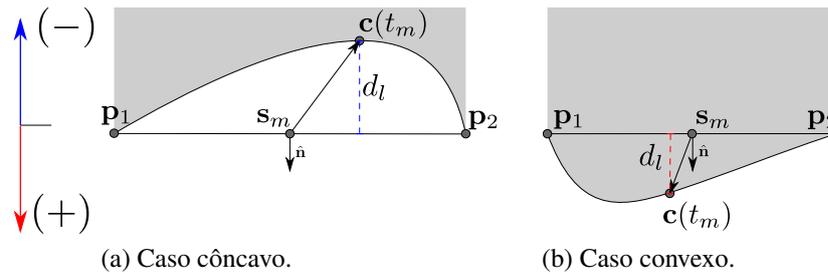
$$\hat{\mathbf{n}} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \frac{\mathbf{p}_2 - \mathbf{p}_1}{\|\mathbf{p}_2 - \mathbf{p}_1\|}; \quad (3.14)$$

e  $t_m$  é o centro do intervalo paramétrico onde o segmento é definido na curva NURBS  $\mathbf{c}(t)$ . O valor  $l_{ce}$  é limitado no intervalo  $s_l[0.5, 1.5]$  para evitar variações excessivas, que podem resultar inclusive em valores negativos.

Equação (3.12) objetiva a definição de de uma função densidade que leve em conta a deflexão das arestas curvas. É importante notar que  $d_l < 0$  para segmentos côncavos,  $d_l > 0$  para segmentos convexos e  $d_l = 0$  para segmentos retos, que é o valor padrão usado no algoritmo original (veja exemplos na Figura 51). A orientação do segmento é dada pelos ciclos adjacentes a

ele. No caso de arestas internas de restrição, onde ambas as orientações são consideradas durante a geração da malha, adota-se  $d_l = 0$  para evitar um baixo valor de tamanho de elemento para o segmento côncavo. O efeito Equação (3.12) na qualidade dos elementos é discutido em mais detalhes na Seção 3.4.

Figura 51 –  $d_l$  medida em segmentos côncavos e convexos.

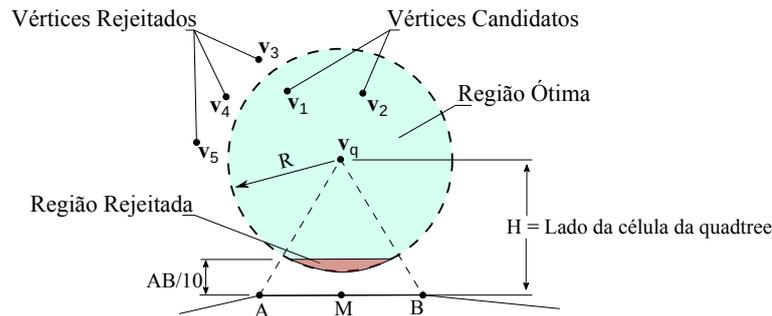


Fonte: Elaborada pelo autor.

Vale ressaltar que critérios adicionais podem ser adotados na construção da função densidade, por exemplo, quando um mapa de tamanhos definidos pelo usuário é especificado ou quando um critério de curvatura é considerado, como no caso da geração de malha de superfícies paramétricas. Contudo, os resultados obtidos utilizando apenas o critério proposto foram satisfatórios, como mostrado em Barroso et al.<sup>10</sup> ou no Capítulo 5, não sendo necessários a consideração de critérios adicionais.

Em seguida, é realizada a inserção dos triângulos no domínio. Nesta etapa, cada segmento curvo é considerado como sua respectiva corda. O processo de geração é dividido em duas fases, o processamento geométrico e o processamento topológico. Na fase geométrica, a fronteira inicial é armazenada na lista de arestas ativas  $\mathcal{L}_{act}$ , que é processada em seguida para geração de triângulos. No processamento de uma aresta, o ponto médio da aresta é localizado na *quadtree*, e o tamanho da célula é utilizado para posicionar o vértice candidato ideal para definição de um triângulo. O candidato é posicionado perpendicularmente à aresta base, partindo do ponto médio da aresta com distância igual ao tamanho da célula da *quadtree*. Do ponto ideal, é definida uma região ótima para busca de vértices existentes, com raio igual a 85% do tamanho da célula. Pontos que estejam muito próximos da aresta base, com distância perpendicular à aresta base menor que 1/10 do comprimento da base, são rejeitados. Este procedimento é ilustrado na Figura 52.

Figura 52 – Seleção de vértice ideal para criação de novo triângulo



Fonte: Elaborada pelo autor.

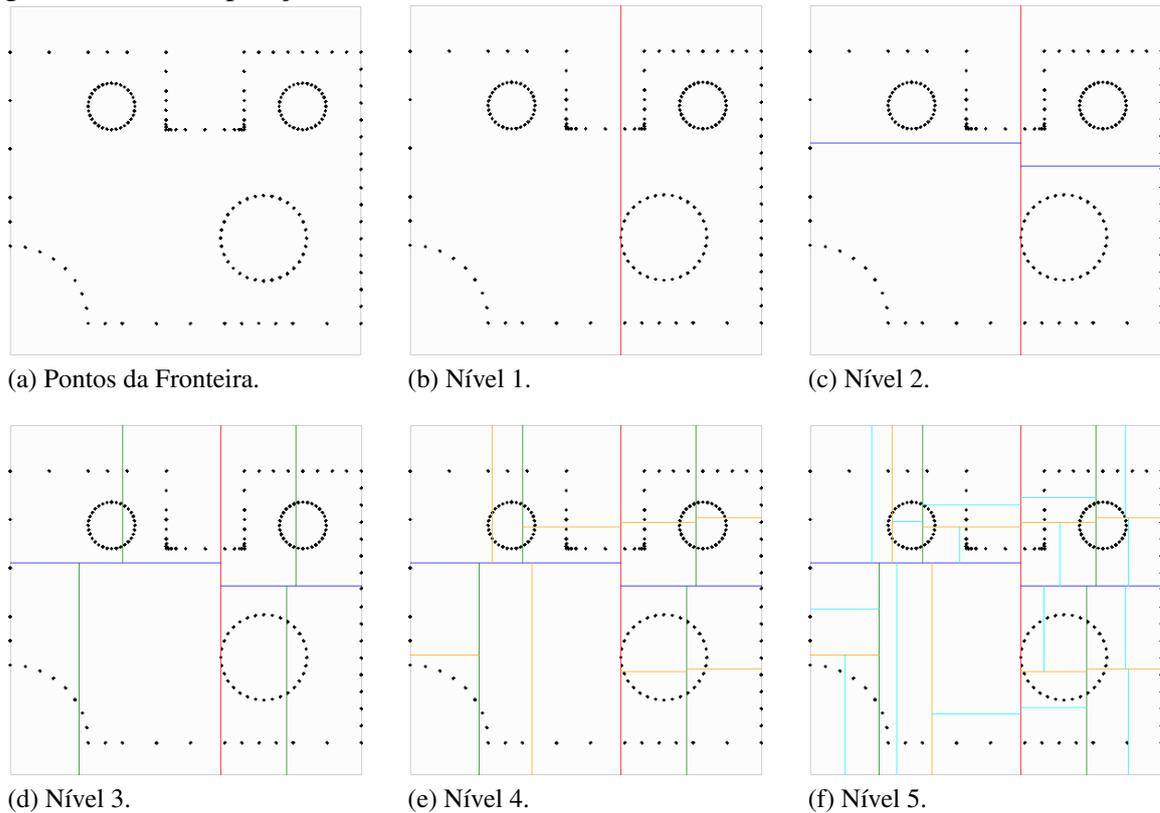
Os vértices existentes que atenderem aos requisitos são ordenados segundo o critério de Delaunay (maior ângulo). Se nenhum vértice for selecionado, o vértice candidato ideal é adicionado. É verificado se um triângulo válido pode ser gerado, com a aresta base e os vértices selecionados. Um triângulo válido é aquele que não intersecta a fronteira atual.

Devido às restrições impostas, é possível que algumas arestas da fronteira não gerem triângulos válidos. Neste caso, a aresta é adicionada à lista de arestas rejeitadas  $\mathcal{L}_{rec}$ . Ao final do processamento da  $\mathcal{L}_{act}$ , as arestas rejeitadas são reinseridas em  $\mathcal{L}_{act}$  e o processamento geométrico é realizado novamente. A motivação desta etapa é de que foi observado que algumas arestas rejeitadas podem vir a atender às restrições e conseguir gerar triângulos em uma segunda tentativa.

Na sequência, o processamento topológico é realizado. As arestas rejeitadas são processadas sem a consideração das restrições descritas anteriormente e sem a inserção de novos pontos, caracterizando um problema de Triangulação por Avanço de Fronteira. Um domínio válido é obtido ao final do processo.

É importante destacar o elevado custo computacional dos procedimentos descritos caso os testes geométricos sejam realizados de forma não otimizada, considerando todos os vértices e arestas da fronteira. Assim, neste trabalho utilizou-se uma  $KDTree$ <sup>125</sup> para otimizar as buscas dos vértices e arestas, evitando cálculos desnecessários. A  $KDTree$  é construída utilizando os vértices da fronteira, com profundidade máxima igual ao número de vértices. A dimensão e o valor numérico escolhidos em cada particionamento é determinado recursivamente, sendo a direção aquela com maior variância, e a posição a média das coordenadas dos pontos no interior da célula na direção escolhida. A Figura 53 ilustra a  $KDTree$  obtida para várias profundidades.

Figura 53 – Decomposição do domínio usando *KDTree*.



Fonte: Elaborada pelo autor.

Finalmente, são aplicadas 5 iterações de técnicas de melhoramento de malhas, a Suavização Laplaciana e o *Back-tracking*. As iterações são executadas em conjunto, primeiro realiza-se a iteração da Suavização Laplaciana e em seguida a iteração do *Back-tracking*.

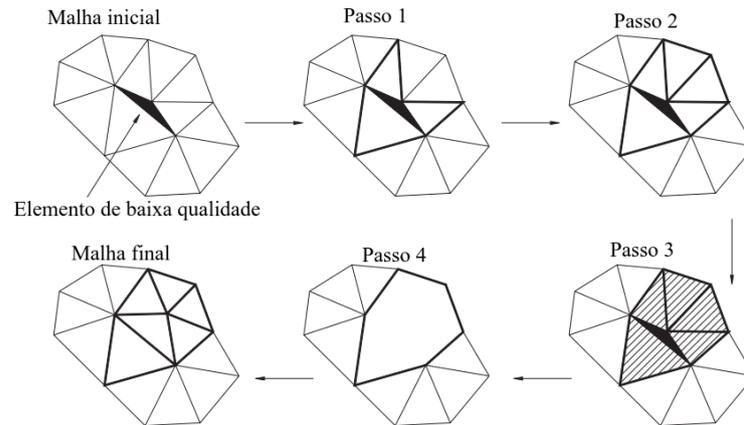
A Suavização Laplaciana<sup>78</sup> é uma técnica de suavização de malhas baseada na reposição dos vértices. Em cada iteração são avaliadas novas posições para os vértices das malhas, exceto os vértices pertencentes ao contorno, de acordo com:

$$\mathbf{p}_v = \mathbf{p}_v + \phi \frac{\sum_{i=1}^m \tilde{w}_{i,v}(\mathbf{p}_i - \mathbf{p}_v)}{\sum_{i=1}^m \tilde{w}_i^O} \quad (3.15)$$

onde  $\mathbf{p}_v$  é o vértice modificado,  $\phi$  é um coeficiente de ajuste,  $m$  é o número de vértices adjacente ao vértice  $v$ ,  $\mathbf{p}_i$  é o  $i$ ésimo vértice adjacente ao vértice  $v$ ,  $\tilde{w}_{i,v}$  é o peso adotado na contribuição da aresta  $(\mathbf{p}_v, \mathbf{p}_i)$ . Neste trabalho foi adotado os valores  $\tilde{w}_{i,v} = 1.0$  e  $\phi = 0.5$ , semelhante ao da referência original<sup>78</sup>.

O *Back-tracking*<sup>78, 126</sup> é uma técnica de melhoramento que consiste na remoção de elementos de baixa qualidade e seus vizinhos, seguida pela geração de uma nova malha na região removida. A Figura 54 ilustra o procedimento.

Figura 54 – Processamento de uma iteração do *Back-tracking*



Fonte: Modificado de Miranda et al. (1999).

O procedimento é aplicado a todos os elementos da malha classificados como de baixa qualidade. Neste trabalho, a qualidade dos triângulos é mensurada pela métrica:

$$T_s = \frac{\sqrt{3} L_m^2}{4 A_T}, \quad (3.16)$$

onde  $L_m$  é o comprimento médio da aresta:

$$L_m = \sqrt{\frac{1}{3} \sum_{i=1}^3 L_i^2}, \quad (3.17)$$

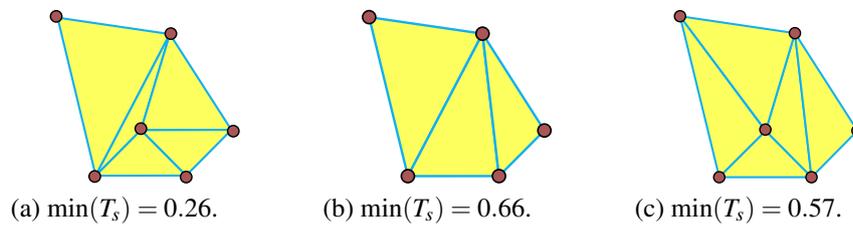
em que  $L_i$  é o comprimento da  $i$ -ésima aresta do triângulo e  $A_T$  é a área do triângulo. A métrica  $T_s$  varia no intervalo  $[0,1]$ , atingindo valor máximo para triângulos equiláteros.

Assim, um elemento é classificado como ruim se  $T_s < q_{min}$ , sendo  $q_{min}$  a qualidade mínima admitida. A cada iteração do *Back-tracking*, considera-se um limite de qualidade  $q_{min}$  variando linearmente entre 0.67 e 0.85.

No remalhamento da região removida, são consideradas 2 regiões candidatas para substituição desta. Uma é a malha obtida via triangulação por avanço de fronteira, que é realizada aplicando a etapa de processamento topológico do método, onde não há a geração de novos vértices. A outra é a malha obtida pelo próprio algoritmo de geração de malha, utilizando a mesma *quadtree* avaliada inicialmente. Neste caso, novos vértices podem ser gerados. É escolhida a malha com maior valor de qualidade para os piores elementos. A Figura 55 ilustra as malhas geradas para um caso de aplicação do *Back-tracking*.

Finalmente, é importante notar que qualquer algoritmo de geração de malhas triangulares que seja conforme à discretização de entrada pode ser utilizado.

Figura 55 – Regiões candidatas avaliadas na etapa de *Back-tracking*: (a) Região inicial; (b) Triangulação; (c) Geração de malha.

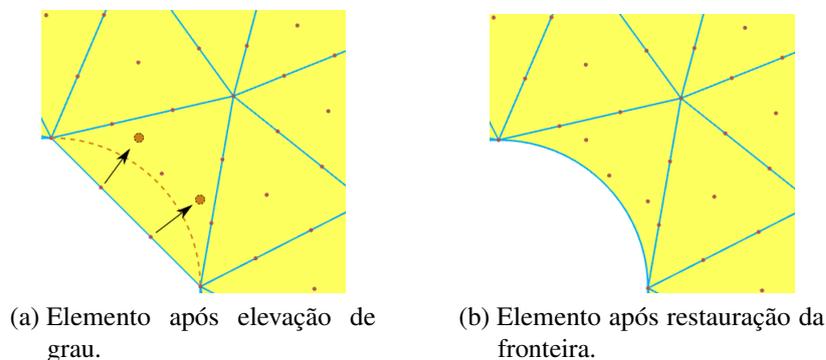


Fonte: Elaborada pelo autor.

### 3.3.2 Elevação de grau e restauração da fronteira

Após obter uma boa malha poligonal, o algoritmo de Elevação de Grau é aplicado a cada triângulo da malha para construir uma malha inicial de alta ordem. A estrutura de dados usada para armazenamento da malha deve ser considerada na implementação desta etapa. Inicialmente, são determinados os pontos de controle interno das arestas, que são curvas Bézier de grau  $p$ . Em seguida, são calculados os pontos de controle internos de cada triângulo de Bézier de grau  $p$ , e também é avaliada a incidência dos pontos de controle pertencentes às arestas do elemento. Em ambos os casos, por se tratar de geometrias poligonais, a *elevação de grau* pode ser realizada por interpolação linear a partir dos vértices dos cantos do triângulo, onde os pontos de controle internos são distribuídos uniformemente no espaço paramétrico. Finalmente, os segmentos curvos associados a fronteira de entrada, dada pelo modelo geométrico, são atribuídos aos elementos correspondentes adjacentes às arestas de entrada, conforme ilustrado na Figura 56.

Figura 56 – Restauração da fronteira realizada em um triângulo de Bézier cúbico.

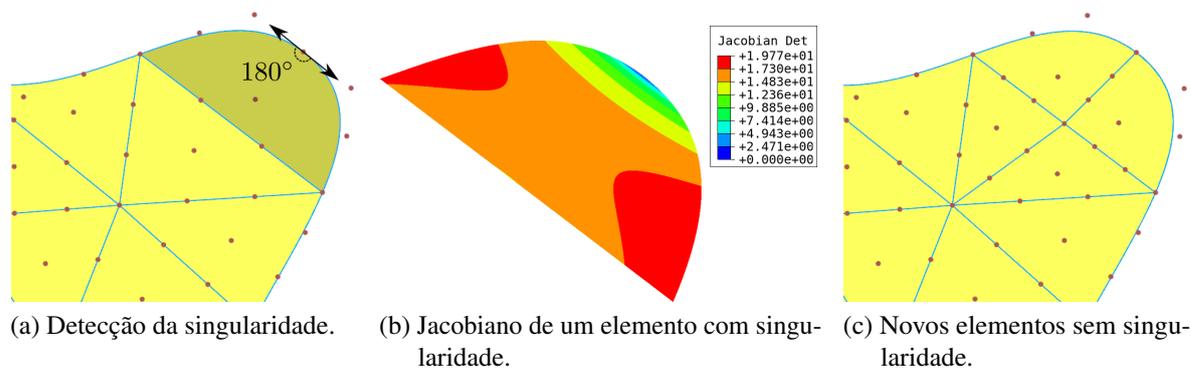


Fonte: Elaborada pelo autor.

### 3.3.3 Tratamento de singularidades

Após a geração da malha de alta ordem inicial, é avaliada a presença de singularidades que eventualmente se manifestem nos pontos de controle (ou vértices) de canto dos triângulos. Em elementos adjacentes a duas arestas curvas, o Jacobiano do elemento tende a zero nos vértices de canto que conectem dois segmentos curvos, quando o ângulo entre esses segmentos se aproxima de  $180^\circ$ , conforme mostrado na Figura 57 (b). É importante ressaltar que a convergência ótima da AIG só pode ser alcançada se o determinante da matriz Jacobiana se tornar constante com o refinamento do modelo<sup>127, 5</sup>, o que não pode ser garantido neste caso.

Figura 57 – Procedimento de divisão em quatro elementos aplicado a elementos adjacentes a duas arestas curvas da entrada.

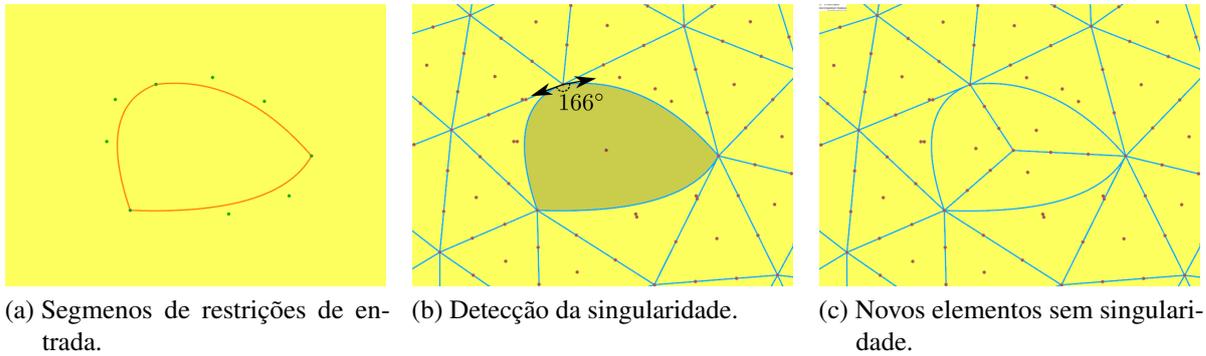


Fonte: Elaborada pelo autor.

Neste trabalho, um ângulo limite  $\theta_c = 155^\circ$  é adotado para detectar a anomalia. Este valor foi testado com sucesso em vários exemplos. Após a detecção deste problema, as seguintes modificações locais na malha são realizadas para remover a singularidade. Elementos adjacentes a duas arestas de entrada são agrupados com o elemento vizinho de sua aresta interna, e divididos em quatro elementos. O novo vértice de canto, compartilhado por todos os novos triângulos, está localizado no meio da borda colapsada. Figura 57 ilustra este procedimento. Outro caso atípico ocorre quando um elemento é adjacente a três arestas de entrada, que pode ocorrer quando são considerados ciclos de restrições fechadas. Neste caso, o elemento é dividido em três, como mostrado na Figura 58.

É importante notar que o problema da singularidade não pode ser solucionado sem alterar a topologia da malha localmente, mesmo que um processo de realocação nodal via otimização seja realizado. Este problema também é discutido no contexto de geração de malha isogeométricas de tetraedros, hexaedros, cunhas e pirâmides de Bézier por Engall e Evans<sup>6</sup>.

Figura 58 – Procedimento de divisão em três elementos aplicado a elementos adjacentes a três arestas curvas da entrada.



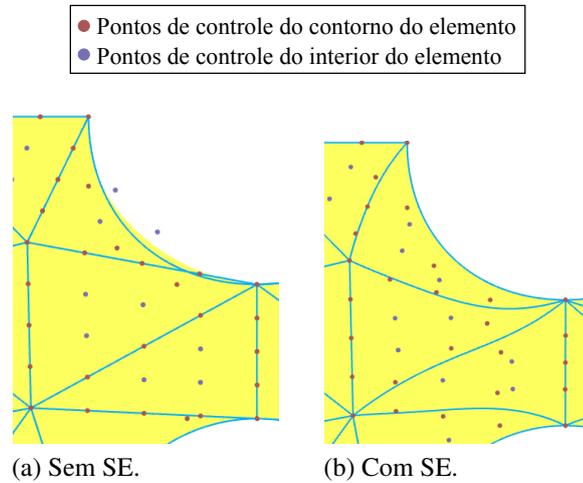
Fonte: Elaborada pelo autor.

### 3.3.4 Suavização de malhas de alta ordem

Garantir a validade das malhas é um problema de maior complexidade na área da geração de malhas de alta ordem. Elementos inválidos ou de baixa qualidade podem ser gerados durante uma etapa de restauração da fronteira, onde arestas retas são substituídas por segmentos curvos. A garantia da validade tende a ser mais complexa à medida que a curvatura dos segmentos e o grau dos elementos aumentam. Assim, esquemas de deformação de malhas são adotados em muitos trabalhos para mitigar este problema.

Na suavização por elasticidade (SE), uma análise elástica é realizada considerando deslocamentos prescritos nos nós (ou pontos de controle) do contorno, de modo a movê-los de uma configuração reta para uma configuração curva dada pelo modelo geométrico. Análises lineares são geralmente empregadas<sup>114, 115</sup>, mas o uso de análises lineares incrementais<sup>128</sup>, análises elásticas não lineares<sup>129, 106</sup> e análises termoelásticas<sup>116</sup>, também é considerado na literatura. As duas últimas metodologias, em comparação com a abordagem de análise linear, são mais eficazes e produzem malhas com melhor qualidade, mas possuem maior custo computacional. A Figura 59 ilustra um exemplo de aplicação da suavização por elasticidade linear. Pode-se observar que a malha sem aplicação da SE é inválida, pois ocorre o cruzamento de arestas do elemento e, além disto, o posicionamento incorreto dos pontos de controle do interior do elemento resulta na expansão do modelo para além da fronteira estabelecida.

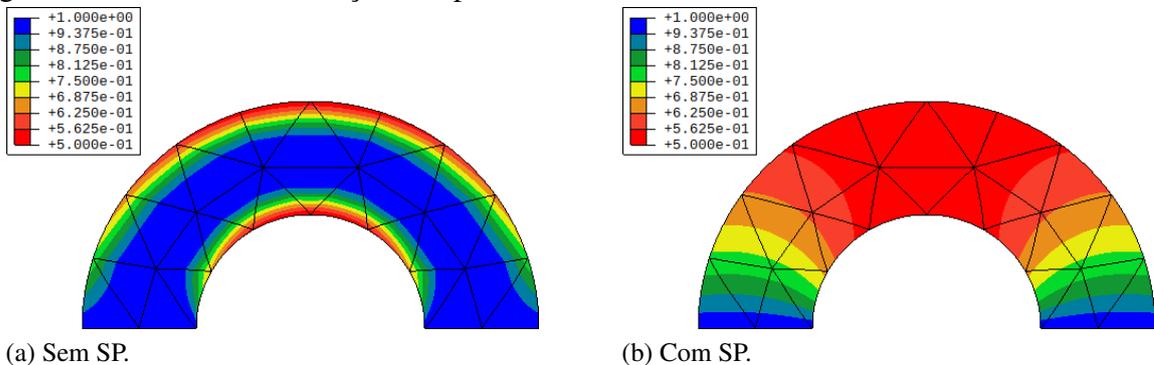
Figura 59 – Substituição do contorno de uma malha quártica: (a) com suavização por elasticidade e (b) sem suavização por elasticidade.



Fonte: Elaborada pelo autor.

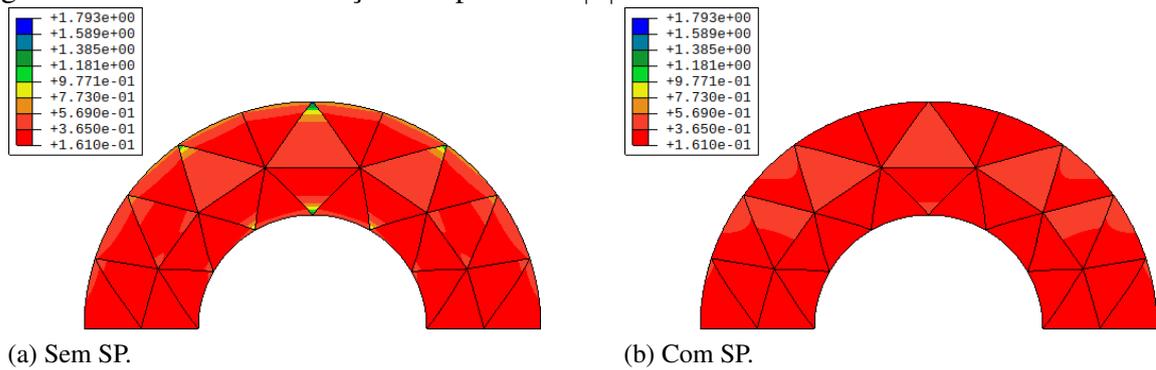
A suavização dos pesos (SP) dos pontos de controle é uma abordagem análoga que visa evitar a variação brusca dos pesos no domínio<sup>5</sup>. Neste caso, uma análise de transferência de calor em regime permanente é realizada usando os pesos dos pontos de controle do contorno como temperatura prescrita, suavizando assim os pesos internos da malha. Embora uma variação excessiva dos pesos não resulte em elementos inválidos, ela afeta diretamente o mapeamento paramétrico e a qualidade do elemento, e pode inibir as taxas de convergência ótimas em aplicações de elementos finitos<sup>108</sup>. A Figura 60 e a Figura 61 mostram o efeito da suavização de peso em  $W$  e  $|\mathbf{J}|$ .

Figura 60 – Efeito da suavização dos pesos em  $W$ .



Fonte: Elaborada pelo autor.

Figura 61 – Efeito da suavização dos pesos em  $|\mathbf{J}|$ .



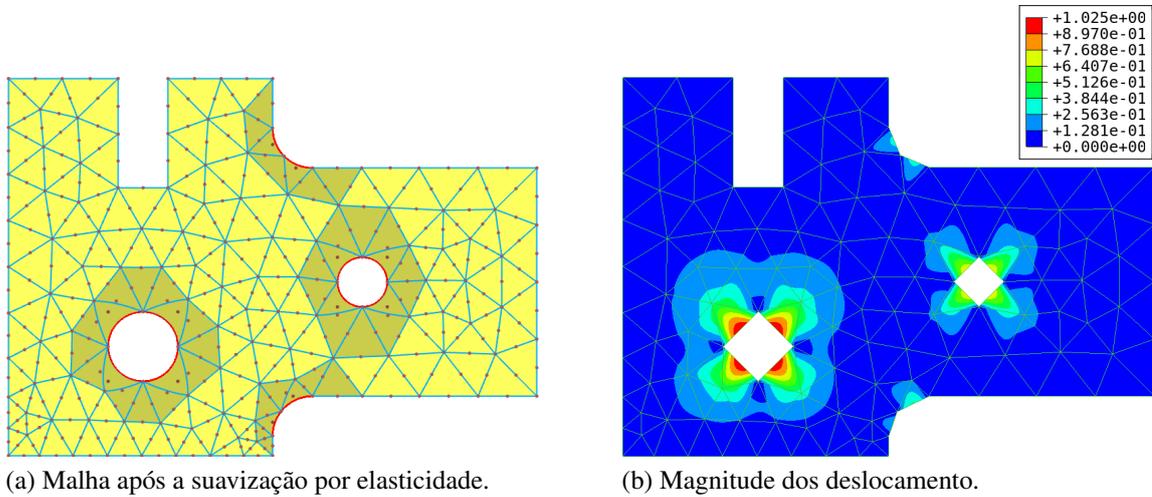
Fonte: Elaborada pelo autor.

A etapa de suavização de alta ordem (SAO) proposta é realizada em dois passos. Primeiro, os pesos dos pontos de controle são suavizados usando a abordagem de transferência de calor descrita. Em seguida, é realizada uma análise elástica linear sobre a malha com os pesos modificados, suavizando as coordenadas dos pontos de controle. A análise linear é escolhida, em detrimento de alternativas mais sofisticadas, para preservar a eficiência computacional do algoritmo proposto.

Mesmo que a análise linear seja adotada, o custo computacional da etapa SAO é alto em comparação a etapa de geração da malha poligonal. Se forem empregados métodos diretos de solução de sistemas lineares, a análise elástica tem complexidade  $O(N^3)$ , enquanto a geração da malha poligonal pode ser implementada com complexidade  $O(N \log N)$ . O alto custo da SAO pode limitar o uso do algoritmo proposto em algumas aplicações, como na otimização de forma, especialmente quando algoritmos heurísticos (e.g. algoritmos genéricos ou nuvem de partículas) são usados.

Por outro lado, um número considerável de elementos da malha apresentam deslocamentos desprezíveis. A Figura 62 mostra um exemplo da aplicação da suavização por elasticidade, e o campo de deslocamentos obtido. Pode-se observar que apenas os elementos próximos às bordas curvas do contorno apresentam deslocamentos significativos, enquanto os demais apresentam valores pequenos ou até desprezíveis. Este comportamento é análogo ao observado na suavização dos pesos. Portanto, neste trabalho ambas, as análises são realizadas localmente, reduzindo drasticamente o custo computacional da etapa SAO, como discutido a seguir.

Figura 62 – Exemplo de aplicação da suavização por elasticidade e os dos deslocamentos obtidos a partir da análise linear. Apenas a vizinhança de arestas curvas apresentam deslocamentos relevantes.



Fonte: Elaborada pelo autor.

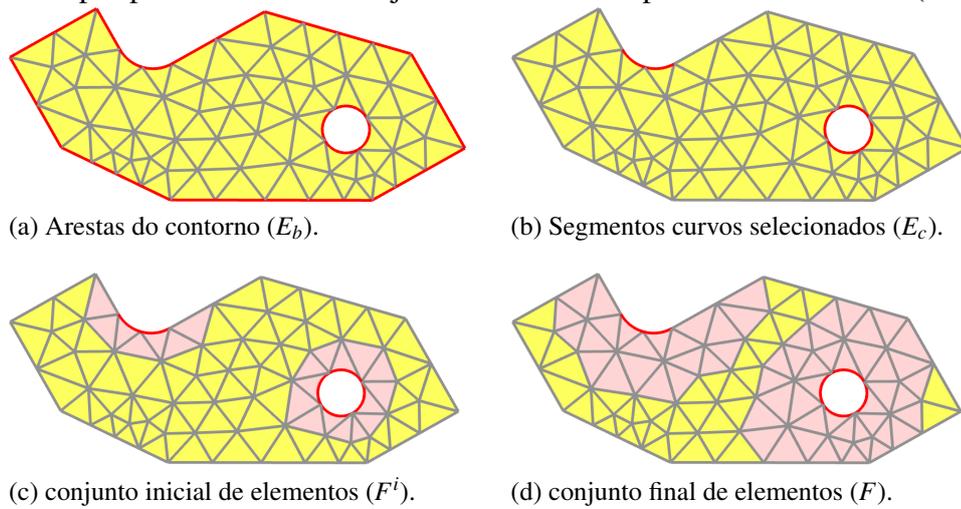
### 3.3.4.1 Análise Linear Localizada

Busca-se encontrar um conjunto de elementos  $F$  da malha para processar análises lineares localmente. Um subconjunto  $E_c$  das arestas de entrada  $E_b$  é avaliado, e os elementos adjacentes aos seus vértices formam um conjunto inicial de elementos  $F^i$  (Na Figura 62 (a),  $F^i$  corresponde aos elementos marcados). Em seguida,  $F^i$  é expandido adicionando todos os elementos adjacentes a seus vértices. Este processo é repetido  $(\alpha - 1)$  vezes ( $\alpha = 1 \implies F = F^i$ ), onde  $\alpha$  é o grau de adjacência, um parâmetro de entrada.

No caso da análise elástica linear, as arestas curvas são selecionadas para formar  $F^i$ . Uma aresta é classificada como curva se o comprimento do seu polígono de controle não for igual à sua corda. Uma tolerância de 1% é adotada para verificar essa condição. Na análise térmica, apenas arestas racionais são selecionadas, ou seja, arestas com pesos não unitários ( $w_i \neq 1$ ).

A Figura 63 demonstra a aplicação das etapas discutidos para encontrar os elementos considerados na SAO. O conjunto de elementos  $F$  para cada análise geralmente possui grupos disjuntos de elementos, que devem ser analisados separadamente. A avaliação de cada grupo é realizada com uma estrutura de dados de sub-malha e sub-aresta. Essas estruturas estão diretamente relacionadas à estrutura topológica de faces e arestas da malha.

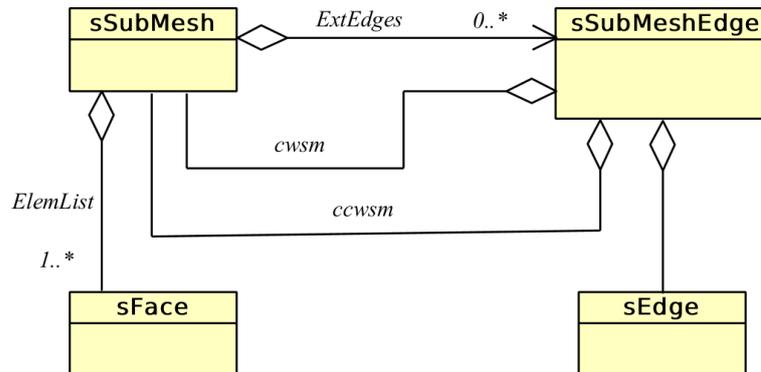
Figura 63 – Etapas para encontrar o conjunto de elementos processados na SAO ( $\alpha = 2$ ).



3.3.4.2 Discussão sobre a implementação

A classe *sSubMesh* armazena as informações necessárias para o processamento da análise linear: uma lista de elementos (*ElemList*), que contém os elementos usados na simulação local, e uma lista de arestas da sub-malha (*ExtEdge*), onde são definidas as condições de contorno. A classe *sSubMeshEdge* tem referências a elementos adjacentes nas orientações no sentido horário ( $cw_{sm}$ ) e anti-horário ( $ccw_{sm}$ ), semelhante à estrutura *winged-edge* (ver Figura 34), e também uma referência à aresta da malha correspondente. A Figura 64 mostra o diagrama de classes UML da estrutura de dados da sub-malha.

Figura 64 – Diagrama de classes da estrutura de dados de sub-malha.



Fonte: Elaborada pelo autor.

A avaliação das sub-malhas existentes se baseia na estratégia usada para a estrutura de dados de conjuntos disjuntos<sup>125</sup>. O operador *MakeSet* inicializa um objeto de sub-malha

para cada elemento em  $F$ , inserindo cada objeto  $sSubMeshEdge$  correspondente das arestas do elemento em  $ExtEdge$ , conforme apresentado no algoritmo ilustrado no Quadro 2.

Quadro 2 – Pseudocódigo do operador *MakeSet*.

<p><b>Entrada:</b> Elemento da malha <math>elm</math>; Arestas da sub-malha adjacentes ao elemento <math>E_{adj}</math>;</p> <p><b>Saída</b> : Sub-malha <math>sm</math> inicializada;</p> <pre> 1 <math>sm.ElemList.adicionar(elm)</math>; 2 <b>para cada</b> <math>e_i</math> em <math>E_{adj}</math> <b>faça</b> 3     <math>sm.ExtEdge.adicionar(e_i)</math>; 4     <b>se</b> <math>AdjacenteAntihorário(e_i.edg,elm)</math> <b>então</b> 5         <math>e_i.ccw_{sm} \leftarrow sm</math>; 6     <b>senão se</b> <math>AdjacenteHorário(e_i.edg,elm)</math> <b>então</b> 7         <math>e_i.cw_{sm} \leftarrow sm</math>; 8 <b>fim</b></pre>
---

Fonte: Elaborada pelo autor.

Após a construção inicial, as arestas internas que dividem duas sub-malhas são identificadas, verificando se ambas as referências  $ccw_{sm}$  e  $cw_{sm}$  existem. O operador *Union* é aplicado a cada  $sSubMshEdge$  interna, resultando em um conjunto de sub-malhas disjuntas. O Quadro 3 mostra o algoritmo do operador de União, onde as listas *ElemList* e *ExtEdge* da sub-malha de saída são atualizadas adequadamente. É importante notar que uma  $sSubMeshEdge$  pode armazenar referências para o mesmo objeto em  $ccw_{sm}$  e  $cw_{sm}$  (esta situação ocorre na Figura 65 (g)). Nesse caso, a operação *Union* é interrompida.

Quadro 3 – Pseudocódigo do operador *Union*.

<p><b>Entrada:</b> Sub-malhas <math>sm_1, sm_2</math>; Aresta de sub-malha adjacente <math>e_a</math>;</p> <p><b>Saída</b> : União <math>sm_1 \cup sm_2</math> armazenada em <math>sm_1</math>;</p> <pre> 1 <math>sm_1.ExtEdge.remove(e_a)</math>; 2 <math>sm_2.ExtEdge.remove(e_a)</math>; 3 <b>se</b> <math>e_i.ccw_{sm} = e_i.cw_{sm}</math> <b>então retorne</b>; 4 <math>sm_1.ElmList.adicionar(sm_2.ElmList)</math>; 5 <b>para cada</b> <math>e_i</math> in <math>sm_2.ExtEdg</math> <b>faça</b> 6     <b>if</b> <math>e_i.ccw_{sm} = sm_2</math> <b>then</b> 7         <math>e_i.ccw_{sm} \leftarrow sm_1</math> 8     <b>else if</b> <math>e_i.cw_{sm} = sm_2</math> <b>then</b> 9         <math>e_i.cw_{sm} \leftarrow sm_1</math> 10 <b>fim</b> 11 <math>sm_1.ExtEdge.adicionar(sm_2.ExtEdge)</math>; 12 <math>sm_2.ElmList.esvaziar()</math>;</pre>
--

Fonte: Elaborada pelo autor.

O algoritmo completo para a avaliação de sub-malhas é apresentado no Quadro 4. A rotina *SelecionarArestas* avalia um conjunto de arestas de acordo com os critérios discutidos anteriormente. Vale ressaltar que qualquer conjunto de elementos pode ser passado como entrada. Um exemplo de aplicação do algoritmo é mostrado na Figura 65.

Quadro 4 – Pseudocódigo do algoritmo para avaliar sub-malhas.

```

Entrada: Lista de arestas do contorno  $E_b$ ;
Saída : Lista com todas as sub-malhas  $M_{sub}$ ;
         Lista com todas as arestas das sub-malhas  $E_{sub}$ ;

// Forma o conjunto inicial de elementos  $F^i$ .
1  $E_c \leftarrow SelecionarArestas(E_b)$ ;
2  $F \leftarrow E_c.vertices.ElementosAdjacentes$ ;

// Expande  $F^i$  para  $F$ .
3 para  $i = 1$  até  $\alpha$  faça  $F.adicionar(F.ElementosAdjacentes)$ ;

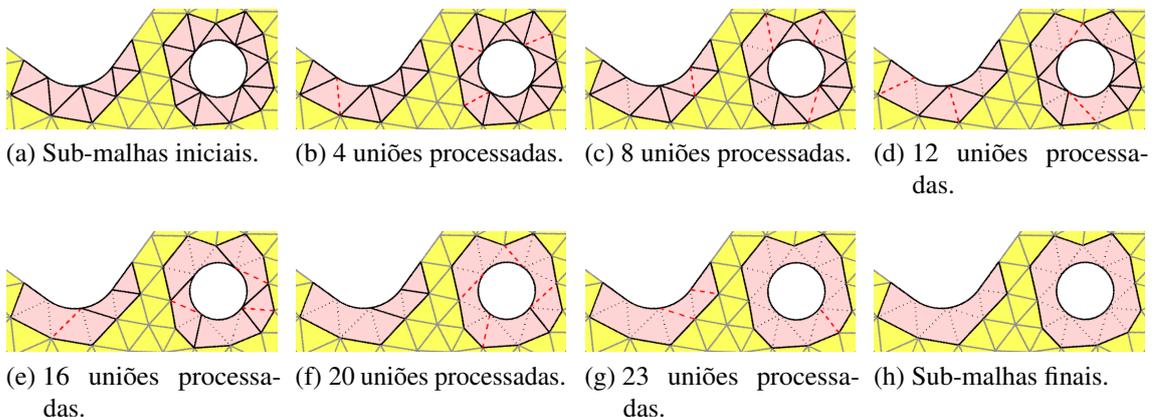
// Inicialização das listas  $M_{sub}$  e  $E_{sub}$ .
4 para cada  $elm$  in  $F$  faça
5   |  $E_{adj} \leftarrow elm.SubArestasAdjacentes$ ;
6   |  $E_{sub}.adicionar(E_{adj})$ ;
7   |  $M_{sub}.adicionar(MakeSet(elm, E_{adj}))$ ;
8 fim

// Avalia as sub-malhas disjuntas.
9 para cada  $e_i$  in  $E_{sub}$  faça
10  | se  $e_i.ccw_{sm}$  e  $e_i.cw_{sm}$  existem então
11  |   |  $Union(e_i.ccw_{sm}, e_i.cw_{sm})$ ;
12  |   | se  $e_i.ccw_{sm}.ElmList = \emptyset$  então
13  |   |   |  $M_{sub}.remove(e_i.ccw_{sm})$ 
14  |   |   | senão se  $e_i.cw_{sm}.ElmList = \emptyset$  então
15  |   |   |   |  $M_{sub}.remove(e_i.cw_{sm})$ 
16 fim

```

Fonte: Elaborada pelo autor.

Figura 65 – Aplicação do algoritmo mostrado no Quadro 4 a um conjunto de 24 elementos, resultando em duas sub-malhas com 8 e 16 elementos.



Fonte: Elaborada pelo autor.

Uma vez que as sub-malhas são determinadas, a análise linear é realizada em cada uma delas. Esta etapa pode ser realizada em paralelo diretamente, já que não há repetição de elementos ou pontos de controle entre as sub-malhas. As condições de contorno adotadas em cada sub-malha são valores prescritos aplicados aos pontos de controle das arestas da lista *ExtEdge*. Vale notar que as condições de contorno também são aplicadas aos vértices e arestas de restrições. Um software acadêmico de análise de elementos finitos desenvolvido no Laboratório de Mecânica Computacional e Visualização é empregado para processamento das análises elásticas e térmicas, e suas rotinas estão disponíveis no PMGen.

### 3.4 Métricas de qualidade

As métricas de qualidade são medidas utilizadas para avaliar a qualidade dos elementos de uma malha. Embora seja importante considerar os resultados numéricos derivados da aplicação de uma malha, é aceito na literatura que elementos com distorções excessivas, como triângulos com ângulos elevados, diminuem a precisão das simulações numéricas. Portanto, métricas baseadas na geometria dos elementos são comumente usadas no contexto de geração e melhoramento de malhas.

No caso de elementos de alta ordem, as métricas de qualidade são mais complexas devido ao mapeamento não linear existente entre o espaço de referência e o espaço físico do elemento, onde podem existir distorções localizadas no espaço do elemento. A matriz Jacobiana é comumente usada na definição de métricas para elementos de alta ordem, devido às informações geométrica que contém.

A razão entre o Jacobiano mínimo e máximo no elemento, conhecido na literatura como *scaled Jacobian*, é uma métrica usada em muitos trabalhos<sup>82, 114, 115, 116, 5</sup>:

$$Q_j^e = \frac{J_{min}^e}{J_{max}^e}, \quad (3.18)$$

onde  $J_{min}^e$  e  $J_{max}^e$  são o menor e o maior valor do Jacobiano dentro do elemento, e  $Q_j^e = 0$  se  $J_{min}^e \leq 0$ . Grandes variações no Jacobiano diminuem o desempenho de um elemento, mesmo se  $J > 0$ . No entanto, um elemento extremamente distorcido pode mostrar pequena ou nenhuma variação em  $|\mathbf{J}|$ <sup>108</sup>.

Por outro lado, existem métricas baseadas no Jacobiano para elementos lineares, que podem ser usadas para definir métricas de alta ordem e, neste caso, avaliam distorções de forma eficaz<sup>130, 85, 88</sup>. Considere a métrica de elementos triangulares dada pela Equação (3.16). Knupp

mostrou que esta métrica linear pode ser avaliada usando a matriz Jacobiana <sup>131</sup>, ou seja,

$$T_j = \frac{\sqrt{3}\gamma}{\lambda_{11} + \lambda_{22} - \lambda_{12}}, \quad (3.19)$$

onde  $\lambda_{11}$ ,  $\lambda_{22}$  e  $\lambda_{12}$  são os componentes do tensor métrico  $A = \mathbf{J}^T \mathbf{J}$  e  $\gamma = \sqrt{\lambda_{11} \lambda_{22} - \lambda_{12}^2}$ .

A Equação (3.19) pode ser usada em elementos triangulares de alta ordem para avaliar sua qualidade em cada coordenada paramétrica  $(\xi, \eta)$ . Assim, uma possível métrica de qualidade para o elemento consiste no menor valor de  $T_j$  no seu interior, ou seja,

$$J_{ts}^e = \min_{(\xi, \eta)} (T_j). \quad (3.20)$$

Duas métricas para avaliação global de malhas são definidas a partir de  $J_{ts}^e$ , o menor  $J_{ts}^e$  da malha:

$$J_{ts} = \min_e (J_{ts}^e), \quad (3.21)$$

e o valor médio de  $J_{ts}^e$  da malha:

$$J_{ts}^m = \frac{\sum_{e=1}^{\hat{n}} J_{ts}^e}{\hat{n}}, \quad (3.22)$$

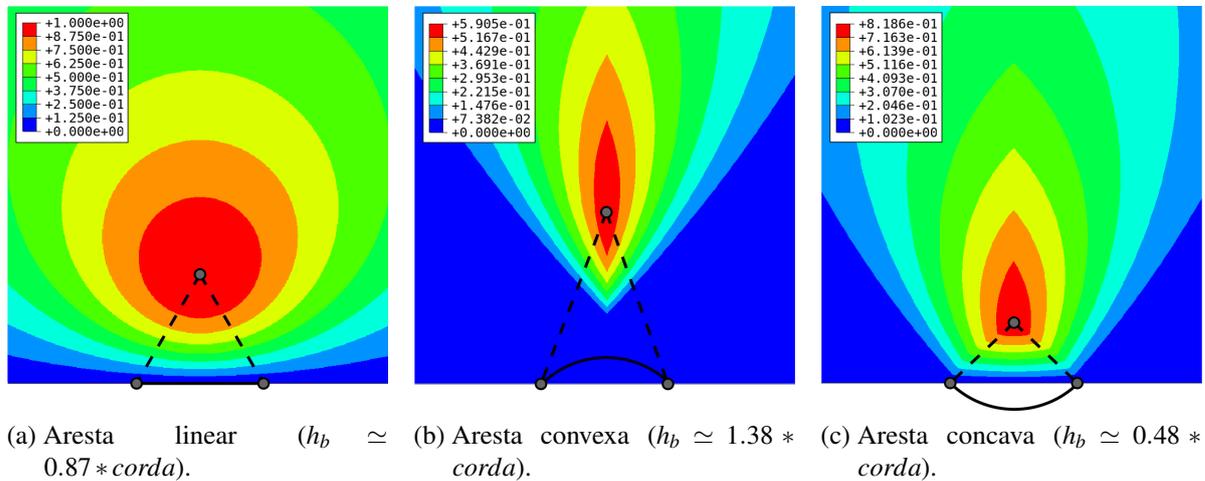
onde  $\hat{n}$  é o número de elementos da malha.

É importante notar que  $J_{ts}^e$  é igual a  $T_s$  para elementos lineares. Assim,  $J_{ts}$  é adequado tanto para elementos de alta ordem quanto para elementos lineares, não sendo necessário combinar duas métricas distintas para obter uma métrica combinada, como feito em <sup>83</sup>. Além disso, para uma dada malha,  $J_{ts}$  não muda quando realizado refinamento uniforme (no espaço paramétrico) e representa um limite inferior para qualquer malha obtida desta forma. Em contraste, o comportamento esperado do *scaled Jacobian* é convergir para 1 quando realizado refinamento uniforme (somente para refinamento- $h$ ). Por fim, destacam-se abordagens na literatura que definem métricas de alta ordem integrando métricas lineares, ao invés de avaliá-las em um conjunto de pontos no domínio do elemento <sup>85</sup>. Essas abordagens foram aplicadas com sucesso no contexto de otimização de malhas <sup>88</sup> e malhas móveis <sup>91</sup>.

Nos algoritmos de avanço de fronteira, o processo de geração de malha é guiado por funções densidade que visam criar triângulos de alta qualidade, principalmente na fronteira da geometria. Entretanto, no caso da geração de malhas de alta ordem, a altura do triângulo estabelecida pela função densidade não leva em consideração o efeito da aresta base curva, pois em geral a etapa de geração de malha poligonal é realizada separadamente, sem nenhuma consideração da aresta curva.

Por outro lado, neste trabalho, a função densidade usada na geração da malha poligonal é modificada (Equação (3.12)) para levar em consideração as arestas curvas, conforme discutido na Seção 3.3.1. Para mostrar a importância dessa modificação, considere triângulos formados por duas linhas e uma base reta, duas linhas e uma base circular côncava e duas linhas e uma base circular convexa, onde o ângulo das bases é de  $90^\circ$ . A Figura 66 mostra as métricas  $J_{ts}^e$  para os três casos, variando a localização do vértice oposto à base, e também mostra os triângulos que apresentam as maiores métricas. Pode-se notar que existe uma grande diferença nas alturas ideais  $h_b$  que maximizam a métrica  $J_{ts}$  nos três casos.

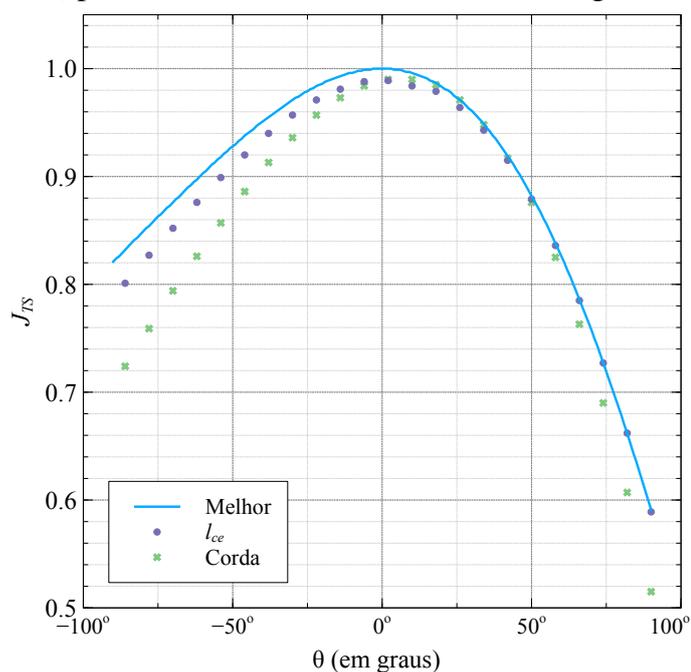
Figura 66 –  $J_{ts}$  dos triângulos com base reta e duas bases circulares.



Fonte: Elaborada pelo autor.

A Figura 67 mostra a variação da métrica  $J_{ts}^e$  para triângulos isósceles de base circular com arco  $\theta$ , considerando três regras diferentes para avaliação da altura do triângulo. A primeira regra é a melhor altura para cada  $\theta$ , a segunda regra é o comprimento  $l_{ce}$  definido na Equação (3.12), e a terceira regra é a corda da curva, que é comumente adotado por algoritmos de geração de malha poligonal, como o apresentado por Barroso e colaboradores<sup>111</sup>. Os resultados para a altura igual a  $l_{ce}$  são melhores em comparação ao terceiro caso. Assim, malhas de maior qualidade são esperadas quando, na construção da *quadtree*, são utilizados os comprimentos  $l_{ce}$ , ao invés dos comprimentos usados no algoritmo padrão. Vale notar que essa modificação afeta apenas um dos critérios utilizados na definição da função densidade, portanto é difícil avaliar o impacto da modificação proposta em casos práticos.

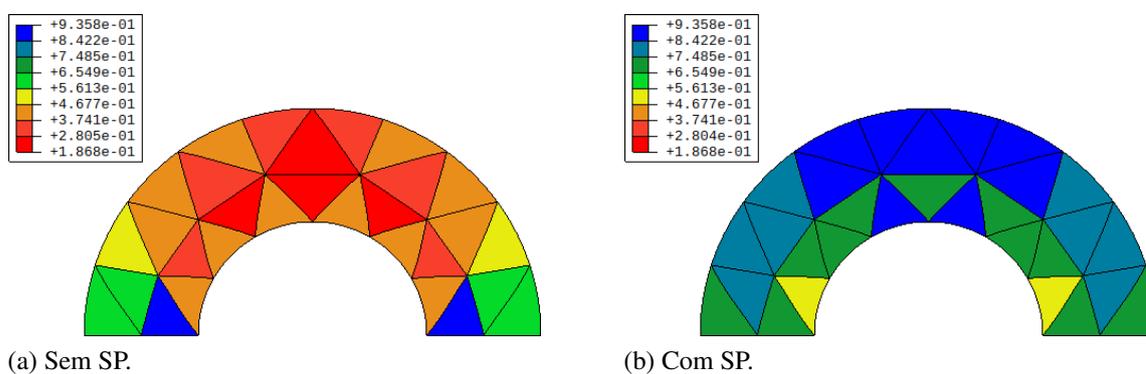
Figura 67 – Valores de  $J_{ts}$  por curvatura  $\theta$  da base circular de triângulos.



Fonte: Elaborada pelo autor.

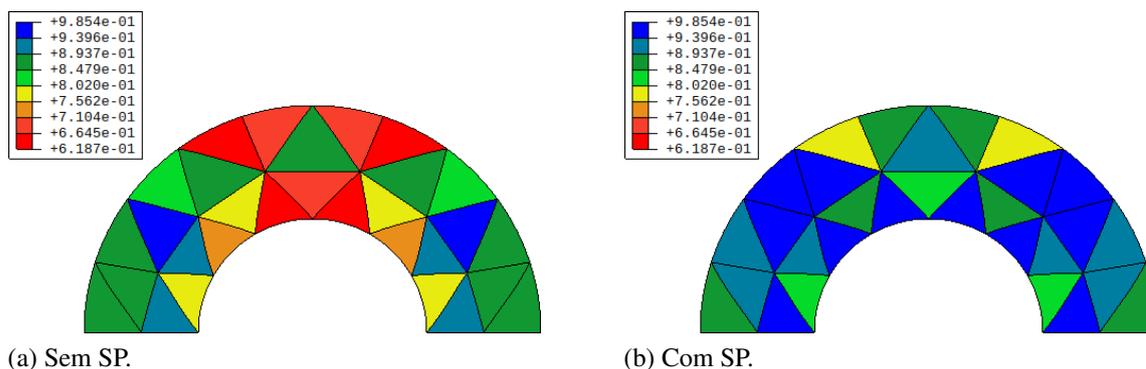
Finalmente, é importante notar que a qualidade dos elementos racionais é afetada não apenas pelas coordenadas dos pontos de controle, mas também pelos pesos dos pontos de controle. A etapa de suavização de pesos discutida na Seção 3.3.4 evita a perda de qualidade dos elemento ao considerar pesos unitários em todos os pontos de controle internos (ver Figura 60). A eficácia da suavização de peso nas métricas  $Q_j^e$  e  $J_{ts}^e$  é mostrada na Figura 68 e na Figura 69.

Figura 68 – Efeito da suavização dos pesos em  $Q_j^e$ .



Fonte: Elaborada pelo autor.

Figura 69 – Efeito da suavização dos pesos em  $J_{IS}^e$ .

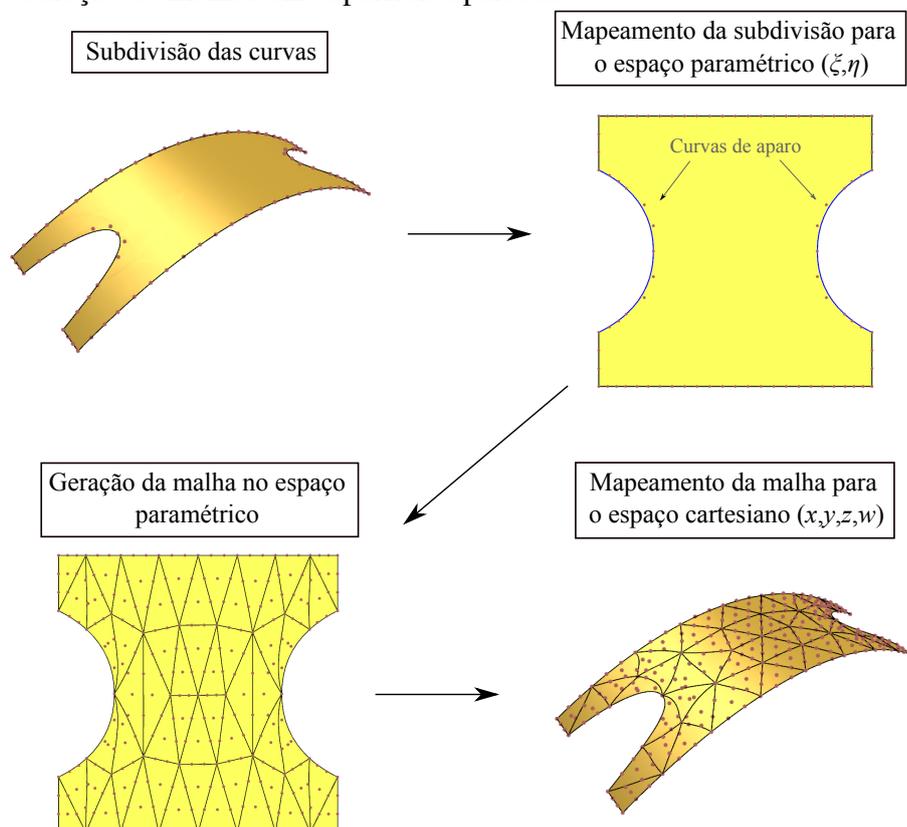


Fonte: Elaborada pelo autor.

### 3.5 Geração de malhas em superfícies aparadas

Neste seção, é discutida a geração de malhas de superfícies aparadas. Os algoritmos apresentados anteriormente para domínios planos são adaptados para o caso de geração de malhas no espaço paramétrico de superfícies NURBS. O processo proposto consiste em quatro etapas, como é ilustrado na Figura 70.

Figura 70 – Geração de malhas em superfície aparadas



Fonte: Elaborada pelo autor.

O método proposto parte de uma superfície NURBS aparada representada por um modelo B-Rep, onde cada região possui uma superfície NURBS associada e um conjunto de curvas de contorno definidas explicitamente. O modelo geométrico contém as curvas do contorno externo e as curvas de aparo. As curvas do contorno externo herdam a geometria das curvas de contorno da superfície NURBS, portanto são definidas por curvas NURBS convencionais em 3D. No caso das curvas de aparo, estas são definidas por curvas NURBS no espaço paramétrico da superfície. A topologia das curvas de aparo é dada por arestas no espaço cartesiano onde os vértices iniciais e finais são obtidos avaliando os pontos extremos das curvas de aparo. Os ciclos são definidos da mesma forma que no caso plano.

Apesar das particularidades das curvas, as subdivisões (ou discretizações) destas são definidas da mesma forma que apresentado anteriormente nos modelos planos, utilizando vetores de valores paramétricos. Os segmentos de Bézier são obtidos aplicando os algoritmos de inserção de *knots* e elevação de grau. Em seguida, os segmentos do contorno externo são mapeados para o espaço paramétrico da superfície. Neste etapa, cada segmento é representado por uma reta ortogonal correspondente no espaço paramétrico, de mesmo grau atribuído na discretização. As curvas de aparo não são mapeadas, pois já estão escritas em  $(\xi, \eta)$ .

Em seguida, um algoritmo de geração de malhas é empregado para obtenção da malha de elementos no espaço paramétrico, onde o grau dos elementos da malha é compatível aos graus atribuídos ao contorno. É importante notar que neste caso, os algoritmos de geração de malhas de superfície buscam gerar elementos de boa qualidade no espaço Cartesiano, e não no espaço paramétrico. Assim, elementos muito distorcidos no espaço paramétrico, podem corresponder a elementos de boa qualidade no espaço Cartesiano, como observado na Figura 70. Isto também evidencia que não é possível aplicar os algoritmos planos nesta etapa de forma direta, sem adaptações.

Por fim, os elementos descritos no espaço paramétrico devem ser mapeados para o espaço Cartesiano. Esta etapa é realizada através do procedimento de ajuste descrito a seguir. Os pontos de controle de canto dos elementos são obtidos diretamente pela equação da superfície base, pois tais pontos são interpoladores nos elementos. Os pontos de controle das arestas são ajustados através da Equação (2.49), considerando amostras distribuídas de forma equidistante no espaço paramétrico da curva. As amostras são avaliadas pela curva de Bézier correspondente no espaço paramétrico, com coordenadas  $(\xi, \eta)$ , e depois avaliadas no espaço Cartesiano pela equação da NURBS, onde passam a ter coordenadas  $(x, y, z, w)$ . Finalmente, o

mesmo procedimento é realizado para os pontos de controle internos dos elementos. As arestas de curvas do contorno externo não precisam ser avaliadas, pois já são definidas inicialmente no espaço Cartesiano.

Para elementos triangulares com geometria poligonal no espaço paramétrico e que tenham grau  $(p + q)$ , onde o  $p \times q$  é o grau da superfície NURBS, o procedimento de ajuste descrito acima resulta em elementos com geometria exata. No caso dos elementos quadrilaterais com geometria poligonal no espaço paramétrico, as arestas destes elementos precisam ter grau  $(p + q)$  no caso geral, portanto, estes também precisam ter grau  $(p + q)$  nas duas direções para garantir a geometria exata. Por outro lado, em casos onde as arestas dos elementos são paralelas aos eixos  $\xi$  e  $\zeta$  da superfície NURBS, os elementos quadrilaterais de grau  $p \times q$  representam a geometria exata.

Nos casos onde existem elementos que não tenham geometria poligonal no espaço paramétrico, pode ser necessário considerar graus de elementos excessivamente elevados para garantir a representação exata no espaço Cartesiano, como descrito para o caso de curvas de arco na Seção 2.6. Assim, embora seja inadequado a utilização de graus muito elevados na prática, a geometria exata pode ser obtida pelo procedimento proposto desta forma.

Vale destacar que o procedimento de geração de malhas descrito pode ser realizado inclusive em modelos com múltiplas regiões, onde as curvas do modelo são discretizadas *a priori*, e as malhas são geradas para cada região individualmente. A seguir, são discutidas as adaptações realizadas nas técnicas de geração de malhas planas para serem utilizadas no caso das superfícies paramétricas.

### **3.5.1 Adaptações para geração de malhas estruturadas**

As técnicas de mapeamento transfinito discutidas na Seção 3.2 podem gerar malhas de superfícies quando curvas de contorno no espaço 3D são utilizadas. No caso de superfícies aparadas, em casos mais simples, o modelo pode ser particionado manualmente em regiões quadrilaterais ou triangulares, permitindo que a técnica seja aplicada em cada partição.

Entretanto, as superfícies de Coons não podem representar superfícies NURBS de forma exata, o que resulta em erros adicionais na aproximação da geometria. Portanto, neste trabalho a técnica de mapeamento transfinito é aplicada no espaço paramétrico da superfície, e em seguida, as malhas são mapeadas para o espaço Cartesiano, como discutido na seção anterior.

### 3.5.2 Geometria diferencial

Na geração de malhas não estruturadas de superfícies paramétricas<sup>132, 133, 134</sup>, é necessário relacionar comprimentos e ângulos existentes no espaço paramétrico a comprimentos e ângulos correspondentes no espaço Cartesiano. Também é importante mensurar a variação do vetor normal ao longo da superfície, o que pode ser estimado a partir da curvatura da superfície. Estes são problemas tratados pela Geometria Diferencial, no estudo de métricas de superfícies. Uma breve apresentação das formulações matemáticas utilizadas é realizar a seguir.

Uma superfície paramétrica  $s(\xi, \eta)$  possui propriedades que variam, no caso geral, ao longo do espaço paramétrico onde é definida. A primeira forma fundamental ( $\mathbb{I}$ ) de  $s$ , em um ponto  $(\xi, \eta)$  qualquer da superfície, é avaliada por:

$$\mathbb{I} = \begin{bmatrix} E & F \\ F & G \end{bmatrix}, \quad (3.23)$$

onde

$$E = \mathbf{t}_\xi \cdot \mathbf{t}_\xi, \quad F = \mathbf{t}_\xi \cdot \mathbf{t}_\eta, \quad G = \mathbf{t}_\eta \cdot \mathbf{t}_\eta, \quad (3.24)$$

sendo  $\mathbf{t}_\xi$  e  $\mathbf{t}_\eta$  os vetores gradientes da superfície em cada direção paramétrica.

Dado um vetor  $\mathbf{p}$  no espaço paramétrico, a rotação em  $90^\circ$  deste vetor no plano tangente da localidade correspondente no espaço cartesiano, é calculada por:

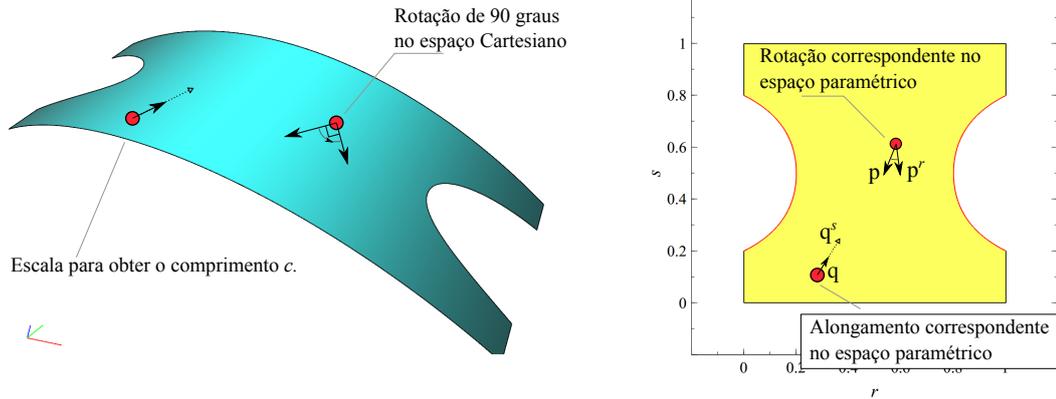
$$\mathbf{p}^r = \begin{bmatrix} -F & -G \\ E & F \end{bmatrix} \mathbf{p}. \quad (3.25)$$

Dado um vetor  $\mathbf{q} = (\xi_q, \eta_q)^T$ , o valor escalar  $s$  tal que o módulo do vetor  $\mathbf{q}^s = s\mathbf{q}$  tenha comprimento  $c$  no espaço cartesiano é dado por:

$$s = \frac{c}{\sqrt{E\xi_q^2 + 2F\xi_q\eta_q + G\eta_q^2}}. \quad (3.26)$$

A Figura 71 ilustra os dois procedimentos.

Figura 71 – Rotação e escala de vetores no espaço Cartesiano da superfície.



Fonte: Elaborada pelo autor.

A segunda forma fundamental (III) de  $S$ , em um ponto  $(\xi, \eta)$  da superfície, é avaliado por:

$$\text{III} = \begin{bmatrix} L & M \\ M & N \end{bmatrix}, \quad (3.27)$$

onde

$$L = \frac{\partial \mathbf{t}_\xi}{\partial \xi} \cdot \tilde{\mathbf{n}}, \quad M = \frac{\partial \mathbf{t}_\xi}{\partial \eta} \cdot \tilde{\mathbf{n}}, \quad N = \frac{\partial \mathbf{t}_\eta}{\partial \eta} \cdot \tilde{\mathbf{n}}, \quad (3.28)$$

sendo  $\tilde{\mathbf{n}} = \mathbf{t}_\xi \times \mathbf{t}_\eta$  o vetor normal ao plano tangente da localidade. A segunda forma fundamental pode ser utilizada para avaliar a curvatura Gaussiana ( $\kappa_g$ ) e curvatura média ( $\kappa_m$ ):

$$\kappa_g = \frac{LN - M^2}{EG - F^2}, \quad \kappa_m = \frac{LG - 2MF + NE}{2(EG - F^2)}. \quad (3.29)$$

Finalmente, as curvaturas principais  $\kappa_{p1}$  e  $\kappa_{p2}$  da superfície são obtidas por:

$$\kappa_{p1} = \kappa_m + \sqrt{\kappa_m^2 - \kappa_g}, \quad \kappa_{p2} = \kappa_m - \sqrt{\kappa_m^2 - \kappa_g}. \quad (3.30)$$

### 3.5.3 Adaptações na função densidade

Na geração de malhas de superfícies, a função densidade pode ser definida no espaço paramétrico da superfície ou diretamente no espaço Cartesiano. Na primeira estratégia, malhas auxiliares planas, como por exemplo *quadtrees*, podem ser adotadas<sup>132, 133</sup>. Na segunda estratégia, malhas auxiliares volumétricas podem ser consideradas<sup>134, 80</sup>.

Neste trabalho, a função densidade adota para geração de malhas de superfícies aparadas é dada por uma *quadtree* definida no espaço paramétrico da superfície. Na geração

dos modelos planos, a densidade ou tamanho dos elementos de uma posição  $(x,y)$  do domínio é dado pelo lado da folha da *quadtree* que contém a posição. O tamanho do lado da folha depende de sua profundidade na árvore e do tamanho da célula raiz, que é dado pela caixa delimitadora quadrada da geometria do problema ( $L_{pl}$ ). Assim, a densidade  $d_{pl}$  em  $(x,y)$  é avaliada por:

$$d_{pl} = \frac{L_{pl}}{2^k} \quad (3.31)$$

onde  $L_{pl}$  é o tamanho da caixa delimitadora quadrada do modelo e  $k$  é a profundidade da folha que contém  $(x,y)$ . Tal metodologia não pode ser aplicada diretamente no caso das malhas de superfícies, pois neste caso os tamanhos das células da *quadtree* são medidos no espaço paramétrico, enquanto os tamanhos dos elementos da malha são dados no espaço Cartesiano.

Logo, associa-se a *quadtree* tamanhos relativos ao espaço Cartesiano. O tamanho correspondente à célula raiz é dado por:

$$L_{srf} = \sqrt{A_{cp}}, \quad (3.32)$$

onde  $A_{cp}$  é a área do malha de controle da superfície NURBS. Assim, a densidade em um ponto  $(\xi,\eta)$  na *quadtree* é dado por:

$$d_{srf} = \frac{L_{srf}}{2^k}. \quad (3.33)$$

Na construção da *quadtree*, os segmentos da discretização de entrada do problema são localizados usando sua representação no espaço paramétrico, mas o valor do tamanho utilizado no refinamento da *quadtree* é dado em função do comprimento da corda ( $s_l$ ) do segmento no espaço Cartesiano. O comprimento  $l_{ce}$ , que no caso plano é dado pela Equação (3.12), é modificado da seguinte forma: a flecha  $d_l$  é avaliada no espaço paramétrico, mas considerando a direção ortogonal à base no espaço Cartesiano. Assim, a direção  $\hat{n}$  é obtida usando a Equação (3.25), com as métricas de superfície avaliadas em  $\mathbf{s}_m$ . Uma vez avaliado  $d_l$  no espaço paramétrico, o comprimento correspondente no espaço cartesiano é obtido por  $|\mathbf{s}(\mathbf{c}(t_m)) - \mathbf{s}(\mathbf{s}_m)|$ , acrescentando o mesmo sinal obtido em  $d_l$ . Vale notar que  $d_l$  será não nulo apenas em segmentos que sejam curvos no espaço paramétrico, ou seja, apenas nas curvas de aparo.

Após a fase inicial de construção da *quadtree* em função dos segmentos do contorno, a *quadtree* é refinada pelos critérios de tamanho máximo e refinamento 1:2, que são realizados da mesma forma que nos algoritmos para domínio plano.

Em seguida, é realizado o refinamento da *quadtree* em função da curvatura da superfície. Este critério busca controlar a variação do vetor normal nos elementos gerados.

Considerando um ângulo limite  $\theta_{max}^S$ , cada célula da *quadtree* é refinada se a variação dos vetores normais na célula for maior que o limite estabelecido. Matematicamente, a condição é verificada como  $\cos(\theta_{max}^{cell}) < \cos(\theta_{max}^S)$ , sendo  $\theta_{max}^{cell}$  o ângulo máximo entre os vetores normais na célula. Esta verificação é realizada de forma aproximada. As curvaturas principais  $\kappa_{p1}$  e  $\kappa_{p2}$  são avaliadas no centro da célula, usando a Equação (3.30). Em seguida, estima-se o ângulo máximo entre os vetores normais na célula por:

$$\theta_{max}^{cell} = \max(|\kappa_{p1}|, |\kappa_{p2}|) d_{srf} \quad (3.34)$$

onde  $d_{srf}$  é o tamanho da célula, dado pela Equação (3.33). O critério é avaliado recursivamente até a condição respectiva seja satisfeita.

Após a aplicação do critério de curvatura, o critério de refinamento 1:2 é realizado novamente.

#### 3.5.4 Adaptações na geração de malhas unidimensionais

Os critérios de discretização uniforme abordados na Seção 3.1.1 são utilizados da mesma forma, considerando os segmentos no  $\mathbb{R}^3$ . Na discretização recursiva descrita na Seção 3.1.2, os critérios de comprimento máximo, curvatura, comprimento mínimo, variação do vetor tangente e também a etapa de suavização das arestas, são realizados de forma idêntica ao caso plano, mas considerando as curvas no  $\mathbb{R}^3$ .

O critério de proximidade é aplicado no espaço paramétrico. As três retas perpendiculares ao segmento, que são usadas para definir a área de proximidade para cálculo dos testes de interseção, agora são perpendiculares no espaço Cartesiano, onde o ângulo correspondente no espaço paramétrico é avaliado pela Equação (3.25). O comprimento das retas é avaliado pela Equação (3.26), de modo que o tamanho de cada uma no espaço Cartesiano seja igual ao tamanho da corda do segmento no 3D. As propriedades da superfície consideradas nas equações são avaliadas no ponto de partida de cada reta no segmento.

No critério de disparidade, a *quadtree* é construída conforme foi descrito na Seção 3.5.3. O critério é realizado no espaço paramétrico, onde são definidos os pontos de avaliação da função densidade da *quadtree*. Vale notar que neste caso  $s_l$  é a corda do segmento no  $\mathbb{R}^3$ .

### 3.5.5 Adaptações na geração de malhas não-estruturadas

A geração da malha não estruturada é realizada com as alterações descritas a seguir. Na etapa de geração da malha poligonal, a *quadtree* é construída conforme descrito na Seção 3.5.3. A fronteira é definida no espaço paramétrico, mas é ordenada considerando os comprimentos das arestas no espaço cartesiano, dados pelo  $l_{ce}$  modificado. Nas iterações de geração de triângulos pelo método de avanço de fronteira, que ocorrem nas etapas de processamento geométrico e processamento topológico, na fase de ordenação dos vértices pelo critério de ângulo máximo, o cálculo do ângulo é realizado considerando os vértices, da aresta base e os candidatos, mapeados no espaço Cartesiano. Os testes de validade dos triângulos candidatos continuam a ser realizados no espaço paramétrico.

Além disso, a localização do candidato ótimo é definida partindo do ponto médio da aresta base, por um segmento que é ortogonal à base no espaço Cartesiano, sendo a direção correspondente no espaço paramétrico avaliada pela Equação (3.25). A altura do triângulo é calculada usando a Equação (3.26), considerando o tamanho obtido na *quadtree* avaliado no centro da aresta base. As propriedades da superfície utilizadas nestes cálculos são definidas no centro da aresta base. Nos critérios de exclusão dos candidatos, a distância entre o candidato ótimos e os vértices existentes, e também a área mínima dos elementos considerados, são avaliadas considerando os vértices mapeados no  $\mathbb{R}^3$ .

A etapa de suavização Laplaciana é realizada de forma similar à apresentada por Miranda e Martha<sup>133</sup>, onde o coeficiente  $\phi = 0.7$  é utilizado. Além disso, os pesos entre as arestas são dados pela razão de seu comprimento no espaço cartesiano pelo comprimento no espaço paramétrico:

$$\tilde{w}_{i,v} = \frac{|S(\mathbf{p}_v) - S(\mathbf{p}_i)|}{|\mathbf{p}_v - \mathbf{p}_i|}. \quad (3.35)$$

Assim, apesar do deslocamento dos vértices ser avaliado no espaço paramétrico, os comprimentos no espaço cartesiano são contabilizados indiretamente pelo esquema de ponderação adotado.

Na etapa do *Back-tracking*, a métrica de qualidade dos triângulos é aplicada considerando seus vértices mapeados para o  $\mathbb{R}^3$ . O remalhamento processado também considera as alterações descritas anteriormente, no processamento geométrico e topológico.

O processo de suavização de alta ordem é idêntico ao realizado no caso plano. Sua aplicação desta forma mostrou efetividade em vários exemplos abordados neste trabalho, evitando elementos inválidos e melhorando a qualidade das malhas. Entretanto, as arestas e faces

dos elementos afetados deixam de ter geometrias poligonais no espaço paramétrico, resultando em geometrias de alta complexidade no espaço Cartesiano, efeito semelhante ao observado nas curvas de aparo.

Embora o caráter isogeométrico seja dificilmente satisfeito nestes elementos, os demais elementos da malha que não sejam afetados pelo SAO permanecem com geometria poligonal no espaço paramétrico. Além disto, usualmente não se busca garantir a exatidão geométrica nos elementos adjacentes às curvas de aparo<sup>48</sup>, mesmo quando não é realizada suavização nestes. Portanto, a SAO proposta é efetiva para evitar elementos inválidos ou de baixa qualidade quando curvas de aparo de geometria complexas são utilizadas, gerando elementos de alta complexidade geométrica apenas nas proximidades de tais curvas.

Por fim, é importante destacar que as métricas de alta ordem definidas na Equação (3.20), Equação (3.21) e Equação (3.22) podem ser empregadas em elementos de superfície. Neste caso, deve ser considerado na Equação (3.19) a matriz Jacobiana de superfícies e triângulos de Bézier no caso 3D, que tem dimensão  $3 \times 2$ .

### 3.6 Otimização de malhas

A otimização de malhas busca melhorar a qualidade dos elementos, considerando uma dada métrica de qualidade. Existem métodos que otimizam malhas realizando modificações topológicas, como no procedimento de troca de arestas<sup>65</sup> (*edge swap*). Também existem métodos baseados no reposicionamento de vértices das malhas, onde neste caso, a topologia da malha não é modificada. Vários métodos desta categoria foram desenvolvidos no contexto de elementos finitos de alta ordem<sup>86, 84, 88</sup> e também em elementos isogeométricos<sup>108</sup>.

Neste trabalho foi desenvolvido uma metodologia de otimização de malhas isogeométricas compostas por elementos de Bézier racionais baseada no reposicionamento dos pontos de controle, onde são considerados além das coordenadas cartesianas, também os pesos dos pontos de controle. Vale notar a importância da função peso na parametrização e qualidade dos elementos racionais.

Seja  $\mathcal{M}$  uma malha de elementos finitos de alta ordem planos, com  $n_{oe}$  elementos, e seja  $\mathbf{d}_{uv}$  um vetor de deslocamentos  $u, v$  e  $\mathbf{d}_w$  um vetor de pesos, ambos de um subconjunto  $\mathbf{p}_{act}$  de pontos de controle de  $\mathcal{M}$ , de tamanho  $n_{act}$ . A malha modificada  $\mathcal{M}_x$  tem os mesmos elementos de  $\mathcal{M}$ , mas com os deslocamentos  $\mathbf{d}_{uv}$  somados nos pontos  $\mathbf{p}_{act}$  e os pesos  $\mathbf{d}_w$  atribuídos aos pontos  $\mathbf{p}_{act}$ . Assim, considerando uma métrica de qualidade das malhas  $Q_{msh}$ , o problema de

otimização de malhas é definido como:

$$\left\{ \begin{array}{l} \text{Determinar } \mathbf{x} = [(u_1, v_1, w_1), (u_2, v_2, w_2), \dots, (u_{n_{act}}, v_{n_{act}}, w_{n_{act}})] \\ \text{que maximiza } Q_{msh}(\mathcal{M}_x) \end{array} \right. \quad (3.36)$$

O subconjunto  $\mathbf{p}_{act}$  é definido usualmente como os pontos de controle internos da malha, que não fazem parte do contorno do problema, ou não são restrições definidas *a priori*. O problema de otimização descrito é de difícil solução, especialmente quando se consideram malhas com muitos elementos. O número de variáveis é extremamente elevado, e além disso, existem várias combinações de valores em  $\mathbf{x}$  que levam a elementos inválidos. Uma forma de mitigar tais problemas é realizar a otimização de forma localizada<sup>84</sup>. No presente trabalho, as otimizações são realizadas localmente utilizando o mesmo algoritmo de sub-malhas adotado na SAO proposta, onde o conjunto inicial de elementos  $F^i$  é dado por elementos de baixa qualidade. Logo, cada sub-malha disjunta do problema resulta em uma malha de otimização  $\mathcal{M}$  respectiva.

Além das otimizações locais, também são consideradas restrições laterais, limitando os deslocamentos  $u$  e  $v$ , respectivamente, a  $[-u_{max}, u_{max}]$  e  $[-v_{max}, v_{max}]$ , e o peso dos pontos de controle a  $[w_{min}, w_{max}]$ . Os deslocamentos máximos  $u_{max}$  e  $v_{max}$  são frações do tamanhos máximos de  $\mathcal{M}$  ao longo de  $x$  e  $y$ , controladas pelo parâmetro de redução de tamanho de busca. Os limites dos pesos são os pesos mínimo e máximo de  $\mathcal{M}$ . O uso das restrições laterais permite ajustar a dificuldade do problema, reduzindo o percentual do espaço de busca com malhas inválidas.

Neste trabalho os problemas de otimização são solucionados utilizando algoritmos heurísticos de ordem zero, como algoritmos genéticos, nuvem de partículas e evolução diferencial<sup>135</sup>. Algumas modificações adicionais ao problema de otimização melhoraram a efetividade desses algoritmos. Primeiro, são consideradas restrições de validade dos elementos de  $\mathcal{M}_x$ . Tal restrição é útil no contexto de algoritmos que utilizem tratamento de restrições via métodos de função objetivo penalizada. Nesta abordagem, a função objetivo das soluções  $\mathbf{x}$  que violam restrições é modificada, tornando-as piores em relação as demais. A abordagem é útil quando são consideradas métricas  $Q_{msh}$  definidas como a qualidade mínima dos elementos da malha, como a dada pela a métrica  $J_{ts}$ , pois permite comparar duas malhas inválidas baseada no número de elementos inválidos de cada uma. Neste caso, observa-se que  $Q_{msh} = 0$  para malhas que tenham pelo menos um elemento inválido.

Outra detalhe importante foi considerar, no conjunto de soluções iniciais do problema, a própria malha  $\mathcal{M}$ . Tal consideração melhorou a convergência dos métodos de otimização

testados. Assim, o problema de otimização de malha passa a ser:

$$\left\{ \begin{array}{l} \text{Determinar } \mathbf{x} = [(u_1, v_1, w_1), (u_2, v_2, w_2), \dots, (u_{n_{act}}, v_{n_{act}}, w_{n_{act}})] \\ \text{que maximiza } Q_{msh}(\mathcal{M}_{\mathbf{x}}) \\ \text{Sujeito a} \\ \quad -u_{max} \leq u_i \leq u_{max}, \quad i = 1, 2, \dots, n_{act} \\ \quad -v_{max} \leq v_i \leq v_{max}, \quad i = 1, 2, \dots, n_{act} \\ \quad w_{min} \leq w_i \leq w_{max}, \quad i = 1, 2, \dots, n_{act} \\ \quad \text{o elemento } e_i \text{ é válido, } i = 1, 2, \dots, n_{oe} \end{array} \right. \quad (3.37)$$

Vale notar que as restrições laterais são satisfeitas explicitamente pelos métodos de otimização utilizados<sup>136, 137</sup>.

No procedimento proposto neste trabalho a otimização é realizada em duas etapas. Primeiro se otimiza a métrica de qualidade mínima  $J_{ts}$ , buscando melhorar a qualidade dos piores elemento da malha. Essa etapa é importante para garantir a validade da malha, que é satisfeita se  $J_{ts} > 0$ . Por outro lado, no caso de malhas pouco discretizadas, observou-se que o processo de otimização da qualidade mínima pode piorar significativamente a qualidade de elementos da malha que inicialmente apresentavam boa qualidade.

Portanto, após a maximização de  $J_{ts}$ , é realizada uma nova etapa de otimização visando maximizar a qualidade média da malha ( $Q_{msh} = J_{ts}^m$ ), partindo da malha otimizada na etapa anterior. Nesta nova otimização, considera-se uma nova restrição limitando  $J_{ts}$  ao valor obtido na etapa anterior, evitando assim que a qualidade do pior elemento da malha seja reduzida. O processo de otimização em duas etapas é realizado em cada sub-malha obtida ao considerar um valor de métrica mínima  $Q_{min}$ , para selecionar os elementos da malha.

### 3.6.1 Otimização de malhas de superfície

O processo de otimização de malhas de superfície é semelhante. As malhas são otimizadas no espaço paramétrico da superfície, onde as variáveis de projeto são deslocamentos nas coordenadas  $(\xi, \eta)$  dos elementos. A avaliação da malha é feita considerando seus elementos no espaço Cartesiano. Portanto, a avaliação de cada projeto  $\mathbf{x}$  requer utilizar o mapeamento dos elementos para o  $\mathbb{R}^3$ , descrito na Seção 3.5. Além disso, a coordenada  $w$  não é considerada, pois usualmente as curvas de aparo não são racionais. Assim, o problema de otimização de malhas de

superfície é definido por:

$$\left\{ \begin{array}{l}
 \text{Determinar } \mathbf{x} = [(\Delta\xi_1, \Delta\eta_1), (\Delta\xi_2, \Delta\eta_2), \dots, (\Delta\xi_{n_{act}}, \Delta\eta_{n_{act}})] \\
 \text{que maximiza } Q_{msh}(\mathcal{M}_{\mathbf{x}}), \text{ no } \mathbb{R}^3 \\
 \text{Sujeito a} \\
 -\xi_{max} \leq \xi_i \leq \xi_{max}, i = 1, 2, \dots, n_{act} \\
 -\eta_{max} \leq \eta_i \leq \eta_{max}, i = 1, 2, \dots, n_{act} \\
 w_{min} \leq w_i \leq w_{max}, i = 1, 2, \dots, n_{act} \\
 \text{o elemento } e_i \text{ é válido no } \mathbb{R}^3, i = 1, 2, \dots, n_{oe}
 \end{array} \right. \quad (3.38)$$

## 4 ANÁLISE ISOGOMÉTRICA

A Análise Isogeométrica (AIG) é um método de análise estrutural que utiliza na solução numérica aproximada as mesmas funções empregadas pelos sistemas CAD para modelagem da geometria. O método possui muitas semelhanças com o Método dos Elementos Finitos (MEF), que é o método mais utilizado atualmente para análise de sólidos e estruturas. A grande diferença de ambos reside na capacidade que a Análise Isogeométrica tem de tratar a geometria do modelo de forma exata, independente do nível de discretização adotado.

No MEF, a estrutura é dividida em um conjunto de elementos conectados através de nós. Os deslocamentos no interior de cada elemento são interpolados a partir dos deslocamentos de seus nós utilizando polinômios. As equações de equilíbrio da estrutura são obtidas utilizando princípios variacionais (trabalho virtual ou energia potencial) ou Métodos de Resíduos Ponderados (e.g. Galerkin). Na formulação isoparamétrica do MEF a geometria dos elementos é descrita pelas mesmas funções utilizadas para aproximar os deslocamentos. Esta formulação é largamente utilizada, pois garante que as condições necessárias para a convergência do MEF para a solução exata do problema sejam automaticamente satisfeitas<sup>1</sup>.

A Análise Isogeométrica utiliza a mesma ideia da formulação isoparamétrica do MEF, mas neste caso a sequência é invertida: os deslocamentos no interior do modelo são aproximados utilizando as mesmas funções utilizadas para definir a geometria do modelo (e.g. B-splines e NURBS). Desta forma, consegue-se representar de forma exata a geometria do modelo e manter a garantia de convergência para a solução do problema.

A AIG foi proposta inicialmente em Hughes e colaboradores<sup>2, 3</sup>, onde foi aplicada para problemas estruturais modelados como sólidos NURBS. Procedimentos de discretização semelhantes aos refinamentos  $p$  e  $h$  existentes no MEF<sup>1</sup> são aplicados utilizando algoritmos tradicionais da área da Modelagem Geométrica, como elevação de grau e inserção de *knots*. Outra vantagem da abordagem é o uso de modelos com continuidade elevada. Além de facilitar o uso de formulações que requerem continuidade  $C^1$ , como em elementos de placa e casca<sup>96, 138, 139, 140</sup>, também permite realizar uma nova estratégia de discretização inexistente nas formulações clássicas de elementos finitos, o refinamento  $k^{141}$ , onde aumenta-se simultaneamente o número, o grau e a continuidade das funções de aproximação.

Uma desvantagem das NURBS é que o arranjo em produto tensorial de seus pontos de controle impede que sejam realizados refinamentos localizados. Por outro lado, ressalta-se que outras representações geométricas capazes de realizar tais refinamentos vem sendo usadas

na AIG, como as B-Splines hierárquicas<sup>142, 143, 144</sup>, T-Splines<sup>18, 145, 146</sup>, LR B-Splines<sup>23, 24, 25</sup>, PHT-Splines<sup>26, 27</sup> e THB-Splines<sup>28, 29</sup>. Apesar das vantagens destacadas, o uso da AIG clássica, baseada em NURBS ou nas alternativas mencionadas, é limitado pelo problema da parametrização do domínio de análise discutido no Capítulo 3.

Por outro lado, a obtenção de modelos isogeométricos usando malhas compostas por triângulos de Bézier racionais com continuidade  $C^0$  apresentam maior robustez, como discutido anteriormente. Elementos triangulares racionais de Bézier foram utilizados em problemas de elasticidade<sup>9</sup>, transferência de calor<sup>5</sup> e em problemas de otimização estrutural de forma<sup>8</sup>. Também são utilizados em problemas de análise de placas e casca finas baseadas na formulação de Kirchhoff-Love, onde a continuidade  $C^1$  e  $G^1$  entre os elementos é imposta usando restrições e multiplicadores de Lagrange<sup>7, 147</sup>. Além disso, as *rational Triangular Bézier Spline*<sup>148</sup>, que são construídas a partir de malhas de triângulos de Bézier racionais e tem continuidade elevada, foram utilizadas<sup>148, 149</sup> na análise de cascas com formulação de Kirchhoff-Love que requer continuidade  $C^1$ .

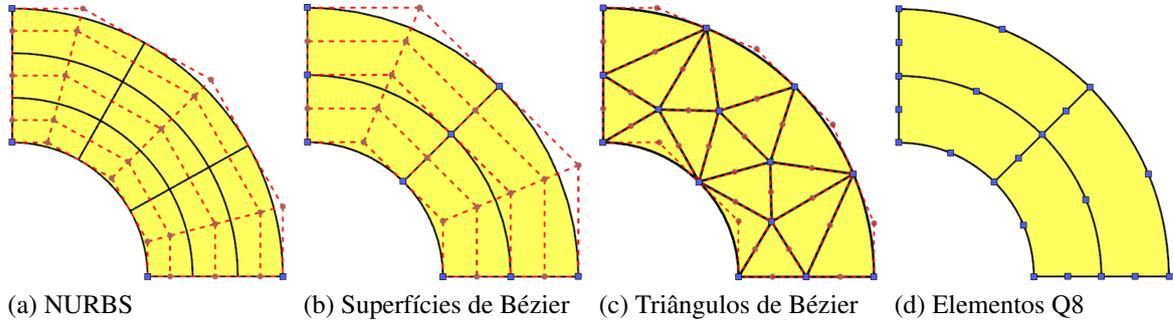
Neste trabalho é proposto a utilização de modelos AIG de continuidade  $C^0$ , compostos por elementos de Bézier racionais. Nesta abordagem, a capacidade de representação exata da geometria se mantém, e refinamentos  $h$  e  $p$  podem ser empregados da mesma forma que na AIG clássica. Refinamentos localizados podem ser aplicados durante a etapa de geração da malha, usando funções de densidade de elementos apropriadas, da mesma forma como é feito no MEF.

A parametrização do domínio de análise pode ser obtida de forma robusta, através de técnicas de geração de malhas isogeométricas, como as propostas no Capítulo 3 ou em outros trabalhos da literatura<sup>9, 5</sup>. A análise de cascas modeladas com superfícies aparadas é realizada usando os mesmos elementos de superfícies sem aparos, pois neste caso um domínio de análise adequado é obtido na etapa de geração de malhas. A Figura 72 mostra modelos de análise quadráticos usando NURBS, malhas isogeométricas de superfície (quadriláteros) e de triângulo de Bézier racionais e uma malha de elementos finitos Q8.

Uma desvantagem da AIG em relação ao MEF é perda de sentido físico dos graus de liberdade do problema estrutural. No MEF, todos os graus de liberdade são deslocamentos da estrutura, enquanto que na AIG, é comum que a maior parte dos graus de liberdade não sejam interpolados, o que dificulta a aplicação de condições de contorno, além de tornar o uso prático do método mais desafiador. Por outro lado, pode-se observar que os modelos IGA baseados em elementos de Bézier apresentam mais graus de liberdade interpolados, sendo assim mais

próximos aos elementos finitos tradicionais.

Figura 72 – Malhas da AIG e MEF para um problema bidimensional. Os pontos de controle interpolados são destacados em azul.



Fonte: Elaborada pelo autor.

#### 4.1 Estado Plano de Tensões

Utilizando os conceitos de modelagem geométrica discutidos no Capítulo 2, a geometria de cada elemento é escrita como:

$$x = \sum_{a=1}^{np} R_a x_a, \quad y = \sum_{a=1}^{np} R_a y_a \quad (4.1)$$

onde  $np$  é o número de pontos de controle,  $R_a$  são as funções base utilizadas e  $x_a, y_a$  são as coordenadas dos pontos de controle. Neste caso, as funções base utilizadas são as bases racionais das superfícies (quadriláteros) e triângulos de Bézier racionais, definidas respectivamente, na Equação (2.6) e na Equação (2.21).

Na Análise Isogeométrica, os deslocamentos ( $u, v$ ) no interior do elemento são aproximados com as mesmas funções base utilizadas para descrever a sua geometria. Portanto:

$$u = \sum_{a=1}^{np} R_a u_a, \quad v = \sum_{a=1}^{np} R_a v_a \quad (4.2)$$

onde  $u_a, v_a$  são os deslocamentos dos pontos de controle. Estas equações podem ser escritas de forma matricial como

$$\hat{\mathbf{u}} = \begin{bmatrix} u \\ v \end{bmatrix} = \sum_{a=1}^{np} \begin{bmatrix} R_a & 0 \\ 0 & R_a \end{bmatrix} \begin{bmatrix} u_a \\ v_a \end{bmatrix} = \sum_{a=1}^{np} \mathbf{N}_a \mathbf{u}_a = \mathbf{N} \mathbf{u} \quad (4.3)$$

onde  $\mathbf{u}$  é o vetor dos graus de liberdade do problema (i.e. deslocamentos dos pontos de controle) e a matriz  $\mathbf{N}$  é dada por

$$\mathbf{N} = \begin{bmatrix} \mathbf{N}_1 & \mathbf{N}_2 & \dots & \mathbf{N}_{np} \end{bmatrix} \quad (4.4)$$

Desta forma, a matriz de aproximação dos deslocamentos  $\mathbf{N}$  é formada por uma série de sub-matrizes correspondentes a cada ponto de controle.

É importante notar que diferente do MEF, os pontos de controle não necessariamente pertencem ao elemento e seus deslocamentos não possuem um sentido físico claro, como ocorrem com os deslocamentos nodais, que correspondem aos deslocamentos de pontos do domínio do problema. Desta forma, a AIG tem características similares às do Método de Rayleigh-Ritz.

Um aspecto importante da AIG é que embora a geometria possa ser descrita exatamente utilizando poucos pontos de controle, a obtenção de resultados precisos na análise requer que um número adequado de graus de liberdade seja considerado. Assim, o número de pontos de controle mínimo a ser utilizado é ditado pela geometria do problema, mas o número efetivamente utilizado depende da precisão requerida da análise.

#### 4.1.1 Deformações

Considerando que os deslocamentos são pequenos, as deformações podem ser calculadas como

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} u_{,x} \\ v_{,y} \\ u_{,y} + v_{,x} \end{bmatrix} = \sum_{a=1}^{np} \begin{bmatrix} R_{a,x} & 0 & 0 \\ 0 & R_{a,y} & 0 \\ R_{a,y} & R_{a,x} & 0 \end{bmatrix} \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \mathbf{B} \mathbf{u} \quad (4.5)$$

É importante notar que a matriz deformação-deslocamento ( $\mathbf{B}$ ) tem o mesmo padrão da matriz  $\mathbf{N}$  mostrada na Equação (4.4).

As funções base  $R_a$  são definidas em função das coordenadas paramétricas  $(\xi, \eta)$ , mas para calcular a matriz deformação-deslocamento é necessário calcular as derivadas destas funções em relação às coordenadas cartesianas  $(x, y)$ . Assim, como na formulação isoparamétrica do MEF<sup>1</sup>, na AIG estas derivadas podem ser calculadas utilizando a matriz Jacobiana ( $\mathbf{J}$ ):

$$\mathbf{J} = \begin{bmatrix} \sum R_{a,\xi} x_a & \sum R_{a,\xi} y_a \\ \sum R_{a,\eta} x_a & \sum R_{a,\eta} y_a \end{bmatrix} \Rightarrow \begin{bmatrix} R_{a,x} \\ R_{a,y} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} R_{a,\xi} \\ R_{a,\eta} \end{bmatrix} \quad (4.6)$$

#### 4.1.2 Equações de Equilíbrio

As equações de equilíbrio do sólido na AIG podem ser obtidas a partir do Princípio dos Trabalhos Virtuais (PTV). De acordo com o PTV, o equilíbrio ocorre quando o trabalho virtual interno ( $\delta U$ ) é igual ao trabalho virtual externo ( $\delta W_{ext}$ ), qualquer que seja o campo de

deslocamentos virtuais  $\delta \hat{\mathbf{u}}$ , desde que este seja pequeno e obedeça às condições de contorno essenciais do problema <sup>1</sup>. Matematicamente:

$$\delta U = \delta W_{ext} \Rightarrow \int_V \delta \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} dV = \int_V \delta \hat{\mathbf{u}}^T \mathbf{b} dV + \int_S \delta \hat{\mathbf{u}}^T \mathbf{q} dS + \sum \delta \hat{\mathbf{u}}_j^T \mathbf{F}_j \quad (4.7)$$

onde  $\delta \boldsymbol{\varepsilon}$  é o vetor de deformações virtuais,  $\boldsymbol{\sigma}$  é o vetor das tensões,  $\mathbf{b}$  são as forças de corpo,  $\mathbf{q}$  são as forças de superfície e  $\mathbf{F}_j$  são as cargas concentradas atuantes no corpo.

Utilizando a Equação (4.5) e considerando apenas pequenos deslocamentos, as deformações virtuais são dadas por:

$$\delta \boldsymbol{\varepsilon} = \mathbf{B} \delta \mathbf{u} \quad (4.8)$$

Assim, o trabalho virtual interno pode ser escrito como:

$$\delta U = \int_V \delta \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} dV = \delta \mathbf{u}^T \int_V \mathbf{B}^T \boldsymbol{\sigma} dV = \delta \mathbf{u}^T \mathbf{g} \quad (4.9)$$

onde o vetor das forças internas da estrutura ( $\mathbf{g}$ ) é dado por

$$\mathbf{g} = \int_V \mathbf{B}^T \boldsymbol{\sigma} dV \quad (4.10)$$

Utilizando a Equação (4.4), o trabalho virtual externo pode ser escrito como:

$$\delta W_{ext} = \delta \mathbf{u}^T \int_V \mathbf{N}^T \mathbf{b} dV + \delta \mathbf{u}^T \int_S \mathbf{N}^T \mathbf{q} dS + \delta \mathbf{u}^T \sum \mathbf{N}_j^T \mathbf{F}_j = \delta \mathbf{u}^T \mathbf{f} \quad (4.11)$$

Portanto, o vetor das cargas externas ( $\mathbf{f}$ ) é dado por

$$\mathbf{f} = \int_V \mathbf{N}^T \mathbf{b} dV + \int_S \mathbf{N}^T \mathbf{q} dS + \sum \mathbf{N}_j^T \mathbf{F}_j \quad (4.12)$$

Nesta expressão, a matriz  $\mathbf{N}_j$  corresponde à matriz  $\mathbf{N}$  da Equação (4.4) avaliada no ponto de aplicação da força  $j$ .

As equações de equilíbrio do modelo discreto são obtidas substituindo as Equações (4.9) e (4.11) na expressão do trabalho virtual:

$$\delta U = \delta W_{ext} \Rightarrow \delta \mathbf{u}^T \mathbf{g} = \delta \mathbf{u}^T \mathbf{f} \quad (4.13)$$

Como os deslocamentos virtuais  $\delta \mathbf{u}$  são arbitrários, as equações de equilíbrio da AIG são dadas por

$$\mathbf{g} = \mathbf{f} \quad (4.14)$$

Neste trabalho, considera-se que o material tem um comportamento elástico linear:

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{12} \end{bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{bmatrix} \Rightarrow \boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\epsilon}, \quad (4.15)$$

onde  $E$  é o módulo de elasticidade do material,  $\nu$  é o coeficiente de Poisson e  $\mathbf{C}$  é a matriz constitutiva elástica<sup>1</sup>. Substituindo a relação tensão-deformação na expressão do vetor de forças internas e utilizando a Equação (4.5), pode-se escrever:

$$\mathbf{g} = \int_V \mathbf{B}^T \boldsymbol{\sigma} dV = \int_V \mathbf{B}^T \mathbf{C} \boldsymbol{\epsilon} dV = \int_V \mathbf{B}^T \mathbf{C} \mathbf{B} dV \mathbf{u} = \mathbf{K} \mathbf{u} \quad (4.16)$$

onde a matriz de rigidez da estrutura é dada por:

$$\mathbf{K} = \int_V \mathbf{B}^T \mathbf{C} \mathbf{B} dV \quad (4.17)$$

Finalmente, substituindo a relação  $\mathbf{g} = \mathbf{K} \mathbf{u}$  na Equação (4.14), podemos escrever as equações de equilíbrio como:

$$\mathbf{K} \mathbf{u} = \mathbf{f} \quad (4.18)$$

A solução da Equação (4.18) permite a determinação dos graus de liberdade do problema ( $\mathbf{u}$ ), que correspondem aos deslocamentos dos pontos de controle do modelo. Após o cálculo do vetor de deslocamentos  $\mathbf{u}$ , as deformações são calculadas utilizando a Equação (4.5) e as tensões são calculadas utilizando a Equação (4.15), completando a solução do problema de análise de tensões.

É importante notar que os elementos de Bézier tem continuidade  $C^0$ , portanto as funções de aproximação dos deslocamentos são definidas dentro de cada elemento de forma independente, da mesma forma que no MEF. Assim, a integração de  $\mathbf{K}$  e  $\mathbf{f}$  de cada elemento é realizada numericamente usando quadraturas de integração com número de pontos adequados, como a quadratura de Gauss<sup>1</sup> no caso das superfícies de Bézier e quadraturas para domínio triangular, como a proposta por Dunavant<sup>150</sup>, no caso dos triângulos de Bézier.

Desta forma,  $\mathbf{K}$  e  $\mathbf{f}$  globais são obtidas a partir das matrizes e vetores dos elementos através do mesmo processo de montagem usado no MEF. Por outro lado, no caso de NURBS e T-Splines, que tem continuidade elevada, a região de atuação das funções de base são irregulares, de modo que a integração das matrizes e vetores é mais complexa em comparação ao MEF<sup>137</sup>.

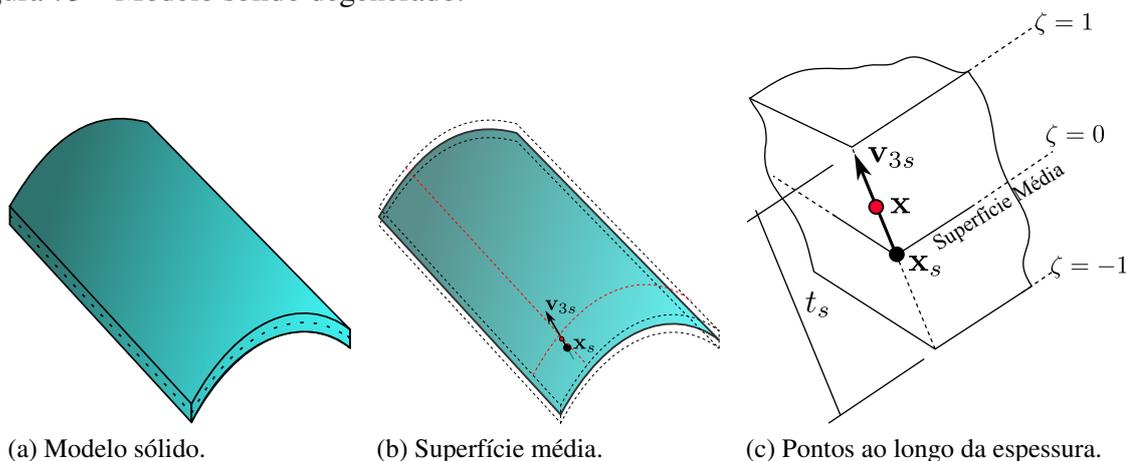
O problema de transferência de calor é definido de modo análogo ao problema estado plano de tensões. Sua descrição é detalhada em outras fontes<sup>1, 151</sup> e será omitida aqui.

## 4.2 Análise de Cascas

Neste trabalho, análises estruturais de cascas são realizadas utilizando uma formulação isoparamétrica de elementos sólidos degenerados baseados nas hipóteses de Reissner-Mindlin<sup>1</sup>. Esta formulação é amplamente utilizada por vários pesquisadores devido à simplicidade na sua formulação e implementação, sua capacidade de tratar tanto cascas de parede finas e quanto cascas de parede espessa, e sua capacidade de modelagem de geometrias diversas<sup>152</sup>. A formulação clássica utilizada para elementos finitos foi adaptada neste trabalho para elementos de Bézier racionais de grau arbitrário.

No modelo de sólido degenerado, a geometria da casca é representada por sua superfície média e pelo vetor normal à esta superfície ( $\mathbf{v}_s$ ). Pontos fora da superfície média são definidos pela coordenada  $\zeta$ . A Figura 73 apresenta um exemplo de modelo sólido degenerado, detalhando a definição dos pontos ao longo da espessura.

Figura 73 – Modelo sólido degenerado.



Fonte: Elaborada pelo autor.

Desta forma, um ponto  $\mathbf{x}$  do sólido é definido por um ponto  $\mathbf{x}_s$  da superfície média, pelo vetor normal  $\mathbf{v}_{3s}$  em  $\mathbf{x}_s$  e pela espessura  $t_s$  em  $\mathbf{x}_s$ , por:

$$\mathbf{x}(\xi, \eta, \zeta) = \mathbf{x}_s(\xi, \eta) + \mathbf{v}_{3s}(\xi, \eta) \zeta \frac{t_s}{2}. \quad (4.19)$$

A geometria da superfície média é modelada pelos elementos finitos de superfície:

$$\mathbf{x}_s = \sum_{a=i}^{np} R_a \mathbf{x}_a \quad (4.20)$$

onde  $\mathbf{x}_a$  são os pontos de controle do elemento e  $R_a$  são as funções de base da superfície de Bézier. Um campo aproximado dos vetores normais da superfície média é definido utilizando as

mesmas funções de forma dos elementos finitos na definição da geometria:

$$\mathbf{v}_{3s} = \sum_{a=i}^{np} R_a \mathbf{v}_{3a}, \quad (4.21)$$

onde  $\mathbf{v}_{3a} = (l_{3a}, m_{3a}, n_{3a})^T$  é um vetor diretor normalizado associado ao ponto de controle  $a$ . Um campo escalar de espessuras é definido de forma análoga:

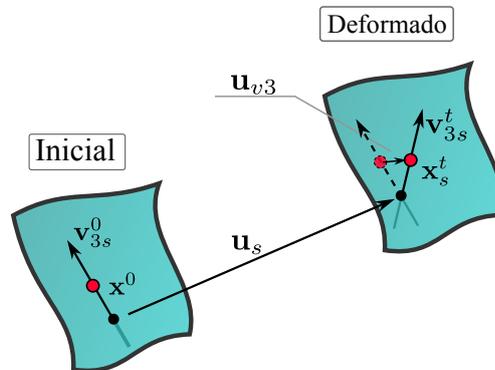
$$t_s = \sum_{a=i}^{np} R_a t_a. \quad (4.22)$$

Substituindo as Equações (4.20), (4.21) e (4.22) na Equação (4.19), a geometria do elemento de casca de Bézier é escrita como:

$$\begin{aligned} x &= \sum_{a=1}^{np} R_a \left( x_a + \zeta \frac{t_a}{2} l_{3a} \right), \\ y &= \sum_{a=1}^{np} R_a \left( y_a + \zeta \frac{t_a}{2} m_{3a} \right), \\ z &= \sum_{a=1}^{np} R_a \left( z_a + \zeta \frac{t_a}{2} n_{3a} \right). \end{aligned} \quad (4.23)$$

Assim, além das coordenadas cartesianas, esta formulação também requer a definição do vetor  $\mathbf{v}_{3a}$  e da espessura  $t_a$  para cada ponto de controle.

Figura 74 – Deslocamento de um ponto arbitrário no modelo de sólido degenerado.



Fonte: Elaborada pelo autor.

O deslocamento  $\mathbf{u}$  de um ponto  $\mathbf{x}^0$  para uma nova configuração  $\mathbf{x}^t$  (ver Figura 74) é dado por:

$$\mathbf{u} = \mathbf{x}^t - \mathbf{x}^0 = \mathbf{u}_s + \mathbf{u}_{v3} = \sum_{a=1}^{np} R_a \mathbf{u}_i + \zeta \frac{t_a}{2} (\mathbf{v}_{3s}^t - \mathbf{v}_{3s}^0). \quad (4.24)$$

Escrevendo  $\mathbf{v}_{3s}^t$  e  $\mathbf{v}_{3s}^0$  em função das normais associadas as funções de base (Equação (4.21)), temos:

$$\mathbf{v}_{3s}^t - \mathbf{v}_{3s}^0 = \sum_{a=i}^{np} R_a \mathbf{v}_{3a}^t - \sum_{a=i}^{np} R_a \mathbf{v}_{3a}^0 = \sum_{a=i}^{np} R_a (\mathbf{v}_{3a}^t - \mathbf{v}_{3a}^0), \quad (4.25)$$

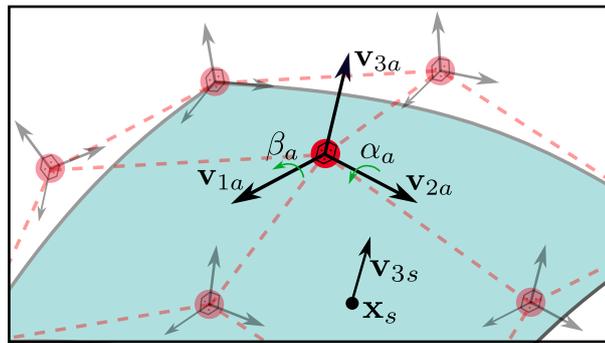
onde  $\mathbf{v}_{3a}^0$  e  $\mathbf{v}_{3a}^t$  são os vetores normais associados aos pontos de controle, respectivamente, na configuração inicial e configuração deformada da estrutura.

Em análises com pequenos deslocamentos e rotações, a rotação do vetor normal  $\mathbf{v}_{3a}$  pode ser expressa como a soma de duas rotações independentes ( $\alpha_a$  e  $\beta_a$ ), em torno de eixos locais  $\mathbf{v}_{1a}$  e  $\mathbf{v}_{2a}$ <sup>152, 1</sup>:

$$\mathbf{v}_{3a}^t - \mathbf{v}_{3a}^0 = -\alpha_a \mathbf{v}_{2a} + \beta_a \mathbf{v}_{1a}, \quad (4.26)$$

onde  $\mathbf{v}_{1a} = (l_{1a}, m_{1a}, n_{1a})^T$  e  $\mathbf{v}_{2a} = (l_{2a}, m_{2a}, n_{2a})^T$  são vetores diretores associados ao ponto de controle  $a$ , e  $\alpha_a$  e  $\beta_a$  são rotações associadas ao ponto de controle  $a$ . Os vetores diretores  $\mathbf{v}_{1a}$ ,  $\mathbf{v}_{2a}$  e  $\mathbf{v}_{3a}$  são perpendiculares entre si, conforme ilustrado na Figura 75.

Figura 75 – Vetores diretores nos pontos de controle da casca.



Fonte: Elaborada pelo autor.

Finalmente, os deslocamentos no interior do modelo são dados por:

$$\begin{aligned} u &= \sum_{a=1}^{np} R_a (u_a + \zeta \frac{t_a}{2} (\beta_a l_{1a} - \alpha_a l_{2a})), \\ v &= \sum_{a=1}^{np} R_a (v_a + \zeta \frac{t_a}{2} (\beta_a m_{1a} - \alpha_a m_{2a})), \\ w &= \sum_{a=1}^{np} R_a (w_a + \zeta \frac{t_a}{2} (\beta_a n_{1a} - \alpha_a n_{2a})), \end{aligned} \quad (4.27)$$

Estas equações podem ser escritas de forma matricial como:

$$\hat{\mathbf{u}} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \sum_{a=1}^{np} \begin{bmatrix} R_a & 0 & 0 & -l_{2a} R_a \zeta t_a/2 & l_{1a} R_a \zeta t_a/2 \\ 0 & R_a & 0 & -m_{2a} R_a \zeta t_a/2 & m_{1a} R_a \zeta t_a/2 \\ 0 & 0 & R_a & -n_{2a} R_a \zeta t_a/2 & n_{1a} R_a \zeta t_a/2 \end{bmatrix} \begin{bmatrix} u_a \\ v_a \\ w_a \\ \alpha_a \\ \beta_a \end{bmatrix}, \quad (4.28)$$

e

$$\hat{\mathbf{u}} = \sum_{a=1}^{np} \mathbf{N}_a \mathbf{u}_a = \mathbf{N} \mathbf{u} \quad (4.29)$$

onde  $\mathbf{u}$  é o vetor dos graus de liberdade do problema (i.e. deslocamentos e rotações dos pontos de controle) e a matriz  $\mathbf{N}$  é dada por

$$\mathbf{N} = \begin{bmatrix} \mathbf{N}_1 & \mathbf{N}_2 & \dots & \mathbf{N}_{np} \end{bmatrix} \quad (4.30)$$

Desta forma, a matriz de aproximação dos deslocamentos  $\mathbf{N}$  é formada por uma série de sub-matrizes correspondentes a cada ponto de controle, similar ao caso dos elementos de estado plano.

#### 4.2.1 Cálculo dos vetores diretores

Em elementos finitos, o cálculo dos vetores normais ( $\mathbf{v}_{3s}$ ) via interpolação permite que sejam consideradas as normais exatas do modelo geométrico em todos os nós da malha. Os vetores nodais exatos são usados para interpolar o vetor normal dentro do elemento, reduzindo os erros de aproximação da geometria do problema. O campo de vetores normais definido desta forma é mais preciso do que utilizar as normais obtidas naturalmente pela geometria dos elementos, que não são exatas no caso geral, especialmente em elementos facetados. Além disso, o uso das mesmas funções de interpolação dos deslocamentos para aproximação dos vetores normais, das espessuras e das rotações do modelo, simplifica a formulação do elemento. Contudo, é importante destacar que embora os vetores  $\mathbf{v}_{3s}$ ,  $\mathbf{v}_{2s}$  e  $\mathbf{v}_{1s}$  sejam definidos de forma consistentes nos nós, i.e. de modo que as normais exatas e eixos de rotação ortogonais e unitários são considerados, o mesmo não se verifica nos demais pontos do elemento<sup>153</sup>.

No contexto da AIG, a definição dos vetores  $\mathbf{v}_{3a}$  é mais complexa, pois nem todos os pontos de controle são interpoladores. Neste caso, os vetores  $\mathbf{v}_{3a}$  não tem sentido físico, pois não representam vetores normais da superfície média da casca. Assim, os vetores normais nos pontos de controle são ajustados de forma que os vetores normais no modelo ( $\mathbf{v}_{3s}$ ) sejam exatos em alguns pontos da casca. O procedimento de ajuste das normais é análogo ao ajuste realizado nas coordenadas dos pontos de controle, descrito na Seção (2.7).

A forma mais comum de realizar o procedimento é associar a cada ponto de controle um ponto  $(\xi_p, \eta_p)$  dada pela projeção deste na superfície. Outra alternativa é associar, a cada ponto de controle, a posição paramétrica que maximiza a função de base correspondente a este.

Dornisch et al.<sup>153</sup> propuseram avaliar as normais nos pontos de controle de forma que os vetores  $\mathbf{v}_{3s}$ ,  $\mathbf{v}_{2s}$  e  $\mathbf{v}_{1s}$  sejam consistentes nos pontos de integração do elemento. De fato, o requisito mínimo para escolha das amostras no espaço paramétrico é que todas as funções de base da superfície devem ser não-nulas em pelo menos um ponto<sup>154</sup>. Em NURBS e T-Splines, tais procedimentos são complexos, pois as regiões, ou *knot span*, de influência das bases seguem padrões irregulares<sup>154</sup>. Em contraste, no caso de elementos finitos, a função de forma de um nó é não nula apenas nos elementos adjacentes a este. O mesmo se verifica no caso dos elementos de Bézier desenvolvidos neste trabalho.

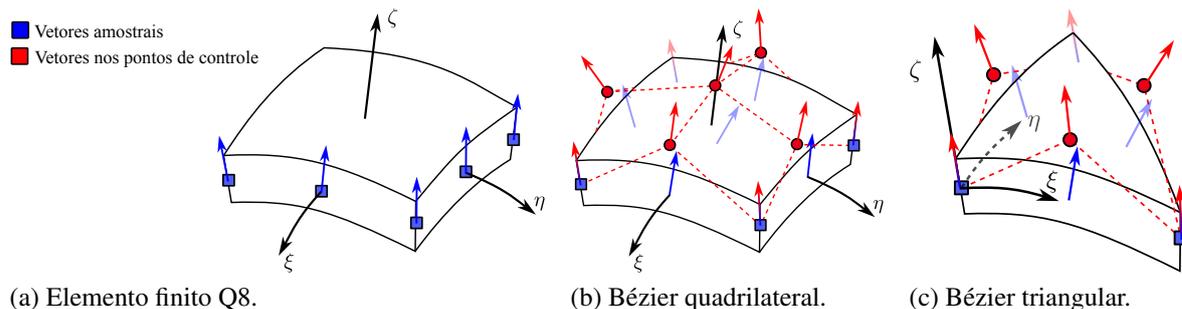
É importante destacar também que no contexto da AIG, como a geometria dos elementos é exata, pode-se considerar as normais dadas diretamente pela geometria do elemento. Esta abordagem é utilizada em Kang e Youn<sup>4</sup>, neste caso o vetor normal é dado por:

$$\mathbf{v}_{3s} = \frac{\mathbf{x}_{s,\xi} \times \mathbf{x}_{s,\eta}}{|\mathbf{x}_{s,\xi} \times \mathbf{x}_{s,\eta}|}. \quad (4.31)$$

A desvantagem desta abordagem é que a formulação do elemento se torna mais complexa.

Neste trabalho, optou-se por utilizar a formulação clássica de aproximação do campo de normais no modelo. Logo, a mesma formulação desenvolvida para elementos finitos é utilizada na AIG. A diferença se dá na forma de avaliação dos vetores diretores associados aos graus de liberdade do modelo. Vetores normais correspondentes a pontos distribuídos de forma equidistante no espaço paramétricos dos elementos são fornecidos, como ilustrado na Figura (76). Tais localizações paramétricas coincidem com os nós de elementos finitos convencionais. Em seguida, é realizado um procedimento de ajuste para cálculo de cada componente dos vetores  $\mathbf{v}_{3a}$  nos pontos de controle dos elementos de Bézier. Vale destacar que este esquema de escolha das amostras associa, a cada ponto de controle, uma amostra onde sua função de aproximação ( $R_a$ ) é máxima, no caso de elementos de Bézier não racionais.

Figura 76 – Esquema de cálculo dos vetores normais nos pontos de controle.



Fonte: Elaborada pelo autor.

Os demais diretores são computados por:

$$\text{Se } \mathbf{e}_2 \times \mathbf{v}_{3a} \neq 0, \begin{cases} \mathbf{v}_{1a} = \mathbf{e}_2 \times \mathbf{v}_{3a} \\ \mathbf{v}_{2a} = \mathbf{v}_{3a} \times \mathbf{v}_{1a} \end{cases} \quad (4.32)$$

$$\text{Se } \mathbf{e}_2 \times \mathbf{v}_{3a} = 0, \begin{cases} \mathbf{v}_{2a} = \mathbf{v}_{3a} \times \mathbf{e}_1 \\ \mathbf{v}_{1a} = \mathbf{v}_{2a} \times \mathbf{v}_{3a} \end{cases}$$

onde  $\mathbf{e}_1$ ,  $\mathbf{e}_2$  e  $\mathbf{e}_3$  são vetores unitários nos eixos  $x$ ,  $y$  e  $z$ .

#### 4.2.2 Deformações

Considerando que os deslocamentos e rotações são pequenas, as deformações podem ser calculadas como

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{xz} \\ \gamma_{yz} \end{bmatrix} = \begin{bmatrix} u_{,x} \\ v_{,y} \\ w_{,z} \\ u_{,y} + v_{,x} \\ u_{,z} + w_{,x} \\ v_{,z} + w_{,y} \end{bmatrix} = \sum_{a=1}^{np} \mathbf{B}_a \begin{bmatrix} u_a \\ v_a \\ w_a \\ \alpha_a \\ \beta_a \end{bmatrix} = \mathbf{B} \mathbf{u}, \quad (4.33)$$

A matriz deformação-deslocamento do elemento ( $\mathbf{B}_a$ ) é dada por:

$$\mathbf{B}_a = \begin{bmatrix} R_{a,x} & 0 & 0 & -l_{2a} T_{a,x} t_a/2 & l_{1a} T_{a,x} t_a/2 \\ 0 & R_{a,y} & 0 & -m_{2a} T_{a,y} t_a/2 & m_{1a} T_{a,y} t_a/2 \\ 0 & 0 & R_{a,z} & -n_{2a} T_{a,z} t_a/2 & n_{1a} T_{a,z} t_a/2 \\ R_{a,y} & R_{a,x} & 0 & (-l_{2a} T_{a,y} - m_{2a} T_{a,x}) t_a/2 & (l_{1a} T_{a,y} + m_{1a} T_{a,x}) t_a/2 \\ R_{a,z} & 0 & R_{a,x} & (-l_{2a} T_{a,z} - n_{2a} T_{a,x}) t_a/2 & (l_{1a} T_{a,z} + n_{1a} T_{a,x}) t_a/2 \\ 0 & R_{a,z} & R_{a,y} & (-m_{2a} T_{a,z} - n_{2a} T_{a,y}) t_a/2 & (m_{1a} T_{a,z} + n_{1a} T_{a,y}) t_a/2 \end{bmatrix} \quad (4.34)$$

onde

$$\begin{aligned} T_a &= R_a \zeta, \\ T_{a,x} &= R_{a,x} \zeta + R_a \zeta_{,x}, \\ T_{a,y} &= R_{a,y} \zeta + R_a \zeta_{,y}, \\ T_{a,z} &= R_{a,z} \zeta + R_a \zeta_{,z}. \end{aligned} \quad (4.35)$$

Novamente, destaca-se que a matriz deformação-deslocamento (**B**) tem o mesmo padrão da matriz **N** mostrada na Equação (4.30).

Semelhante ao discutido na formulação de estado plano, para calcular a matriz deformação-deslocamento é necessário calcular as derivadas de  $R_a$  em relação às coordenadas cartesianas  $(x, y, z)$ . Estas derivadas são avaliadas utilizando a matriz Jacobiana (**J**),

$$\mathbf{J} = \begin{bmatrix} x_{,\xi} & y_{,\xi} & z_{,\xi} \\ x_{,\eta} & y_{,\eta} & z_{,\eta} \\ x_{,\zeta} & y_{,\zeta} & z_{,\zeta} \end{bmatrix} = \begin{bmatrix} \sum R_{a,\xi} (x_a + \zeta \frac{t_a}{2} l_{3a}) & \sum R_{a,\xi} (y_a + \zeta \frac{t_a}{2} m_{3a}) & \sum R_{a,\xi} (z_a + \zeta \frac{t_a}{2} n_{3a}) \\ \sum R_{a,\eta} (x_a + \zeta \frac{t_a}{2} l_{3a}) & \sum R_{a,\eta} (y_a + \zeta \frac{t_a}{2} m_{3a}) & \sum R_{a,\eta} (z_a + \zeta \frac{t_a}{2} n_{3a}) \\ \sum R_a \frac{t_a}{2} l_{3a} & \sum R_a \frac{t_a}{2} m_{3a} & \sum R_a \frac{t_a}{2} n_{3a} \end{bmatrix}. \quad (4.36)$$

Assim,

$$\begin{bmatrix} R_{a,x} \\ R_{a,y} \\ R_{a,z} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} R_{a,\xi} \\ R_{a,\eta} \\ R_{a,\zeta} \end{bmatrix}. \quad (4.37)$$

Além das funções  $R_a$ , que aproximam o deslocamento da superfície média dos elementos, também são consideradas as derivadas de  $\zeta$ ,

$$\begin{bmatrix} \zeta_{,x} \\ \zeta_{,y} \\ \zeta_{,z} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \zeta_{,\xi} \\ \zeta_{,\eta} \\ \zeta_{,\zeta} \end{bmatrix} \quad \text{e} \quad \begin{bmatrix} \zeta_{,\xi} \\ \zeta_{,\eta} \\ \zeta_{,\zeta} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \quad (4.38)$$

### 4.2.3 Matriz constitutiva

A formulação do presente trabalho considera o comportamento do material isotrópico elástico linear. Além disso, a hipótese do modelo de sólido degenerado considera que a deformação e tensão na direção da espessura ( $\zeta$ ) são nulas. Assim, a relação constitutiva no sistema local da casca é dada por:

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \tau_{12} \\ \tau_{13} \\ \tau_{23} \end{bmatrix} = \begin{bmatrix} \bar{E} & \nu \bar{E} & 0 & 0 & 0 & 0 \\ \nu \bar{E} & \nu \bar{E} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & G & 0 & 0 \\ 0 & 0 & 0 & 0 & \bar{G} & 0 \\ 0 & 0 & 0 & 0 & 0 & \bar{G} \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{bmatrix} \Rightarrow \boldsymbol{\sigma} = \bar{\mathbf{C}} \boldsymbol{\varepsilon}, \quad (4.39)$$

onde  $E$  é o módulo de elasticidade do material,  $\nu$  é o coeficiente de Poisson,  $G = 0.5E/(1 + \nu)$ ,  $\bar{E} = E/(1 - \nu)^2$  e  $\bar{G} = 5G/6$ , em concordância com a literatura<sup>1</sup>.

O sistema local do elemento, onde as propriedades materiais da casca são definidas, é dado pelos vetores diretores  $\mathbf{v}_{1s}$ ,  $\mathbf{v}_{2s}$  e  $\mathbf{v}_{3s}$ . Os vetores  $\mathbf{v}_{1s} = (l_{1s}, m_{1s}, n_{1s})^T$  e  $\mathbf{v}_{2s} = (l_{2s}, m_{2s}, n_{2s})^T$  são interpolados de maneira análoga ao realizado para no cálculo do diretor  $\mathbf{v}_{3s}$ :

$$\mathbf{v}_{1s} = \sum_{a=i}^{np} R_a \mathbf{v}_{1a}, \quad \mathbf{v}_{2s} = \sum_{a=i}^{np} R_a \mathbf{v}_{2a}, \quad (4.40)$$

As propriedades dos materiais são transformadas para o sistema global do problema, onde a matriz de deslocamento-deformações foi definida e o equilíbrio da estrutura é estabelecido. A matriz constitutiva no sistema global é dada por<sup>1</sup>:

$$\mathbf{C} = \mathbf{T}^T \bar{\mathbf{C}} \mathbf{T} \quad (4.41)$$

onde

$$\mathbf{T} = \begin{bmatrix} l_1^2 & m_1^2 & n_1^2 & l_1 m_1 & l_1 n_1 & m_1 n_1 \\ l_2^2 & m_2^2 & n_2^2 & l_2 m_2 & l_2 n_2 & m_2 n_2 \\ l_3^2 & m_3^2 & n_3^2 & l_3 m_3 & l_3 n_3 & m_3 n_3 \\ 2l_1 l_2 & 2m_1 m_2 & 2n_1 n_2 & l_1 m_2 + l_2 m_1 & l_1 n_2 + l_2 n_1 & n_1 n_2 + m_2 n_1 \\ 2l_1 l_3 & 2m_1 m_3 & 2n_1 n_3 & l_1 m_3 + l_3 m_1 & l_1 n_3 + l_3 n_1 & n_1 n_3 + m_3 n_1 \\ 2l_2 l_3 & 2m_2 m_3 & 2n_2 n_3 & l_2 m_3 + l_3 m_2 & l_2 n_3 + l_3 n_2 & n_2 n_3 + m_3 n_2 \end{bmatrix} \quad (4.42)$$

#### 4.2.4 Integração do elemento

O equilíbrio da casca é obtido de forma análoga ao apresentado anteriormente na formulação de estado plano de tensões. Assim, o problema estático é solucionado resolvendo o sistema

$$\mathbf{K}\mathbf{u} = \mathbf{f} \quad (4.43)$$

A matriz de rigidez do problema ( $\mathbf{K}$ ) é montada a partir das matrizes de rigidez dos elementos de casca ( $\mathbf{K}_E$ ), que são integradas utilizando uma quadratura trivariada:

$$\mathbf{K}_E = \int_V \mathbf{B}^T \mathbf{C} \mathbf{B} dV = \sum_{i=1}^{n_{pg}} (\mathbf{B}^T \mathbf{C} \mathbf{B} |\mathbf{J}|)_i W_i, \quad (4.44)$$

onde  $n_{pg}$  é número de pontos de integração e  $W_i$  são os pesos de cada ponto. É importante notar que o número de pontos de integração considerados nas direções  $\xi$  e  $\eta$  devem ser adequados ao grau do elemento adotado para representação da superfície média.

O vetor de forças do problema ( $\mathbf{f}$ ) é montado a partir dos vetores de forças dos elementos de casca ( $\mathbf{f}_e$ ), onde há contribuição referente às forças concentradas ( $\mathbf{f}_{ec}$ ), às forças de superfície ( $\mathbf{f}_{es}$ ) e às forças de corpo ( $\mathbf{f}_{eb}$ ):

$$\mathbf{f}_e = \int_V \mathbf{N}^T \mathbf{b} dV + \int_S \mathbf{N}^T \mathbf{q} dS + \sum \mathbf{N}_j^T \mathbf{F}_j = \mathbf{f}_{eb} + \mathbf{f}_{es} + \mathbf{f}_{ec}. \quad (4.45)$$

As forças de superfícies do elemento são integradas usando quadraturas bivariadas na superfície média do elemento:

$$\mathbf{f}_{es} = \int_S \mathbf{N}^T \mathbf{q} dS = \sum_{i=1}^{n_{pg}} (\mathbf{N}^T \mathbf{q} | \mathbf{J} |)_i W_i. \quad (4.46)$$

As forças de corpo do elemento são integradas usando quadraturas trivariadas:

$$\mathbf{f}_{eb} = \int_V \mathbf{N}^T \mathbf{b} dV = \sum_{i=1}^{n_{pg}} (\mathbf{N}^T \mathbf{b} | \mathbf{J} |)_i W_i. \quad (4.47)$$

Neste trabalho foi utilizada a quadratura de Gauss<sup>1</sup> para integração dos elementos quadrilaterais nas três direções. Quadraturas específicas para domínio triangular são utilizadas no caso dos elementos de triângulo de Bézier racional<sup>150</sup>. Na direção da espessura, em ambos os elementos, foi utilizada a quadratura de Gauss com dois pontos de integração.

#### 4.2.5 Vibrações livres

A formulação dos elemento de casca apresentada foi expandida para análise de vibração livre. Neste problema, busca-se avaliar as frequências e os modos de vibração da estrutura, que são determinadas resolvendo o problema de auto-valor:

$$[\mathbf{K} - \omega^2 \mathbf{M}] \phi = \mathbf{0} \quad (4.48)$$

onde  $\phi$  são os modos de vibração,  $\omega$  são as frequências de vibração e  $\mathbf{M}$  é matriz de massa da estrutura. Métodos de solução específicos para problemas de autovalores generalizados são utilizados para encontrar os  $n$  primeiros pares das frequências e modos de vibração, como o método de iteração em sub-espacos e o método de Lanczos<sup>155</sup>. Neste trabalho o problema de auto-valor é solucionado utilizando o método de iteração em sub-espacos. Uma discussão mais detalhada sobre o problema de vibração livre e os métodos de solução empregados é realizada na bibliografia de elementos finitos<sup>1, 155</sup>.

A matriz de massa  $\mathbf{M}$  é obtida por um processo de montagem similar ao realizado na matriz de rigidez  $\mathbf{K}$ , a partir da matriz de massa dos elementos, definida por:

$$\mathbf{M}_E = \int_V \mathbf{N}^T \rho \mathbf{I} \mathbf{N} dV = \sum_{i=1}^{n_{pg}} (\mathbf{N}^T \rho \mathbf{I} \mathbf{N} | \mathbf{J} |)_i W_i \quad (4.49)$$

onde  $\mathbf{M}_E$  é matriz de massa do elemento,  $\rho$  é a densidade do material,  $\mathbf{N}$  é a matriz de aproximação dos deslocamento dada pela Equação (4.30) e  $\mathbf{I}$  é a matriz identidade. Vale notar que  $\mathbf{M}$  tem tamanho  $5np \times 5np$ ,  $\mathbf{N}$  tem tamanho  $3np \times 5np$ , e  $\mathbf{I}$  tem tamanho  $3 \times 3$ . Novamente, destaca-se que a integração dos elementos deve levar em consideração o grau adotado para representação da superfície média.

### 4.3 Implementação computacional

As formulações isogeométrica apresentadas foram implementadas no software acadêmico de Elementos Finitos FAST<sup>137</sup>, desenvolvido em linguagem de programação C++ e utilizando o paradigma de Programação Orientada a Objetos (POO).

É importante notar que as análises em domínio 2D (estrutural e térmico) implementadas são utilizadas diretamente na etapa de Suavização de Alta Ordem do método de geração de malhas, como descrito na Seção 3.3.4.

## 5 EXEMPLOS NUMÉRICOS

Este capítulo apresenta vários exemplos numéricos de aplicação dos algoritmos de geração de malha e simulação numérica apresentados neste trabalho, sendo estes avaliados em termos de eficácia, precisão e eficiência.

Inicialmente, são abordadas as técnicas de geração de malhas desenvolvidas, com foco nos algoritmos não estruturados, avaliando o desempenho destes com base nas propriedades geométricas dos elementos gerados. As técnicas de geração de malha desenvolvidas neste trabalho são abreviadas da seguinte forma:

- AFP - Avanço de fronteira em domínio plano;
- AFS - Avanço de fronteira em domínio paramétrico de superfície 3D;
- MTB - Mapeamento transfinito bilinear.

Em seguida, são abordados exemplos de aplicações numéricas, onde as malhas obtidas pelos algoritmos desenvolvidos são aplicadas em problemas de engenharia estrutural e transferência de calor, tanto para problemas planos quanto para problemas de superfícies 3D.

A descrição da geometria de todos os exemplos tratados neste trabalho, como as coordenadas dos pontos de controles, vetores de *knots*, e os dados das curvas de aparos, são disponibilizadas no Apêndice A.

A máquina utilizada para processar todos os exemplos deste trabalho tem processador Intel Core 2 Quad Q6600 com 2.40GHz e 8GB de memória RAM.

### 5.1 Geração de Malhas

As técnicas de geração de malha não estruturadas são aplicadas em vários exemplos. O grau de adjacência  $\alpha = 2$  e o coeficiente de função densidade  $\beta = 1.6$  são adotados em todos os exemplos.

No Item 5.1.1, o desempenho do algoritmo AFP é aferido em modelos com complexidade variada, comparando a qualidade das malhas isogeométricas geradas pelo algoritmo com as obtidas usando uma alternativa da literatura. No Item 5.1.2 é demonstrada a aplicação da técnica de discretização automática de contorno descrita na Seção 3.1.

No Item 5.1.3, a eficiência do programa PMGen, que implementa o algoritmo AFP, é comparada a de um gerador eficiente de malhas de elementos finitos de alta ordem, onde é avaliado o tempo de processamento e qualidade das malhas geradas pelos dois programas. O

efeito da etapa SAO, que modifica as coordenadas e pesos dos pontos de controle dos elementos, sobre a qualidade das malhas é avaliado para diferentes graus de elemento e nível de discretização no Item 5.1.4.

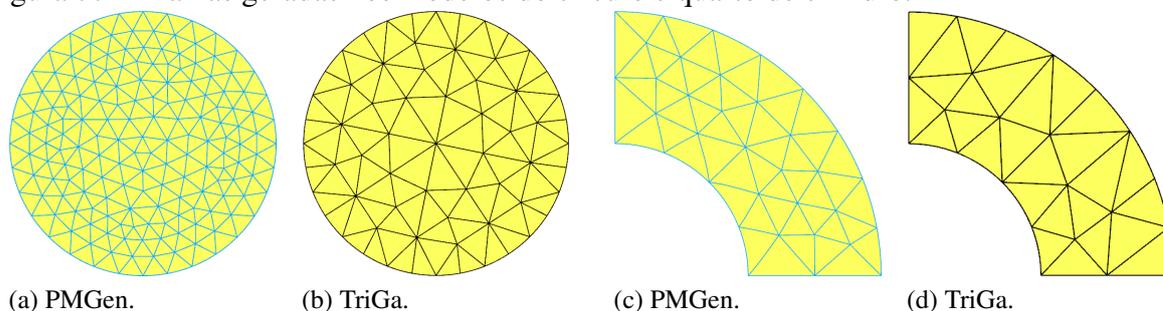
Por fim, a generalização dos algoritmos planos para problemas de cascas é apresentado nos Itens 5.1.5 e 5.1.6, onde é abordado a geração de malhas para problemas de superfícies, incluindo o caso de superfícies aparadas.

### 5.1.1 Desempenho do algoritmo AFP

Nesta seção, o algoritmo AFP implementado no PMGen é avaliado em vários exemplos de aplicação, comparando seu desempenho com o do programa TriGA (Triangular IGA)<sup>62</sup>. O TriGA implementa o método de geração de malhas isogeométricas *Dynamic Quadtree*, apresentado em Engvall e Evans<sup>5</sup>. Este método consiste em um algoritmo automático de geração de malhas, onde o contorno do problema é discretizado automaticamente. A fim de permitir a comparação entre os algoritmos, a parametrização de entrada utilizada no AFP é a mesma obtida pelo TriGA. Em todos os casos são considerados elementos cúbicos, pois é o padrão utilizado no TriGA e em sistemas de modelagem geométrica.

Os dois primeiros exemplos considerados são um círculo e um quarto de cilindro espesso. As malhas obtidas pelos dois programas são apresentadas na Figura 77. A Tabela 2 mostra o número de elementos ( $\hat{n}$ ) e as métricas de qualidade mínima ( $J_{ts}$ ) e média ( $J_{ts}^m$ ) obtidas por cada programa, onde (P) são os resultados referentes ao PMGen e (T) os referentes ao TriGA. A distribuição dos elementos por faixa de qualidade é apresentada na Figura 78.

Figura 77 – Malhas geradas nos modelos do círculo e quarto de cilindro.



Fonte: Elaborada pelo autor.

Os dois programas geram malhas de alta qualidade, com métrica  $J_{ts}$  maior que 0.8 em todos os casos. O bom desempenho dos algoritmos é esperado pela simplicidade dos

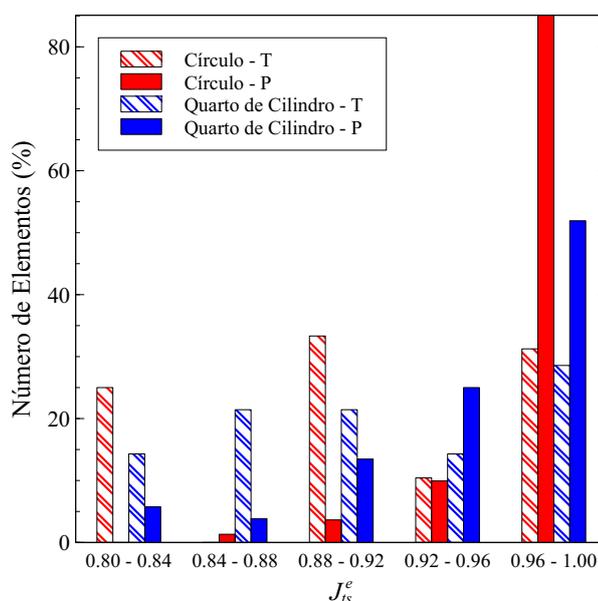
dois problemas, onde não há complexidade na geometria dos modelos e o tamanho das arestas curvas do contorno são similares, resultando numa densidade de elementos sem variações. O TriGA gera malhas com quantidade menor de elementos do que o PMGen, o que sugere que sua implementação utiliza uma função densidade maior em comparação ao com a adotada no PMGen. Também nota-se, para o TriGA, elementos maiores e de maior qualidade no interior do modelo do que no contorno. Isto se deve ao fato deste utilizar uma técnica de geração de malhas baseada em triangulação de Delaunay. Por outro lado, o PMGen apresenta elementos de tamanho similar e qualidade similar em todo o modelo. Em termo de qualidade dos elementos, o PMGen apresenta malhas com qualidade maior em ambas as métricas mínima e média, além de uma melhor distribuição de elementos por qualidade.

Tabela 2 – Qualidade das malhas obtidas nos modelos do círculo e quarto de cilindro.

Exemplo	$\hat{n}$		$J_{TS}$	$J_{TS}$	$J_{TS}^m$	$J_{TS}^m$
	(T)	(P)	(T)	(P)	(T)	(P)
Círculo	96	302	0.7895	0.8594	0.9099	0.9779
Quarto de Cilindro	28	52	0.7243	0.8268	0.8963	0.9494

Fonte: Elaborada pelo autor.

Figura 78 – Distribuição dos elementos por faixas de qualidade para os modelos do círculo e quarto de cilindro.



Fonte: Elaborada pelo autor.

Em seguida, são considerados dois exemplos de maior dificuldade, uma placa com furo e um bocal, ambos apresentados por Envall e Evans<sup>5</sup>. Neste caso, há maior variação nos

tamanhos das arestas do contorno nos modelos, tornando necessário que haja uma transição adequada da função densidade dos elementos. Além da comparação do PMGen com o TriGA, também será considerado a malha obtida pela otimização da métrica mínima ( $J_{ts}$ ), realizada na malha gerada pelo PMGen sem aplicação da SAO, onde as coordenadas cartesianas e pesos dos pontos de controle são as variáveis da otimização. Deste modo, compara-se o desempenho das malhas geradas pelo AFP com o de malhas otimizadas para maximização da qualidade. Neste trabalho, consideram-se elementos de boa qualidade aqueles com  $J_{ts}^e \geq 0.75$ , e:

$$J_{ts}^{sup} = \frac{\sum_{e=1}^{\hat{n}} q_s(e)}{\hat{n}} \cdot 100, \quad q_s(e) = \begin{cases} 1 & \text{se } J_{ts}^e \geq 0.75 \\ 0 & \text{caso contrário} \end{cases} \quad (5.1)$$

é o percentual de elementos classificados como de boa qualidade.

O processo de otimização é aplicado localmente, conforme descrito na Seção 3.6, onde é adotada o valor limite de 0.75 para escolha dos elementos para formação das sub-malhas, com grau de adjacência igual a 2 e parâmetro de redução do tamanho de busca igual a 0.05. O algoritmo de otimização de Evolução Diferencial apresentado em Ribeiro et al.<sup>136</sup> é utilizado, com os parâmetros apresentados na Tabela 3. Embora este algoritmo não avalie o critério de otimalidade para solução encontrada, o emprego deste se mostrou adequado para otimizar consistentemente os exemplos analisados neste trabalho.

Tabela 3 – Parâmetros usados para otimização das malhas.

Parâmetro	Valor
População	50
Número de Gerações	2000
Taxa de Cruzamento	0.8
Taxa de Mutação	0.5
Peso de F	0.85
Método de Diferenciação	<i>LocaltoBest</i>

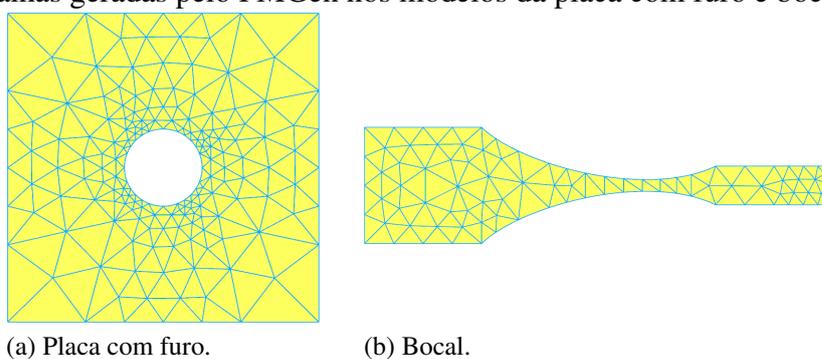
Fonte: Elaborada pelo autor.

As malhas obtidas pelo PMGen são apresentadas na Figura 79. A qualidade das malhas é apresentada na Tabela 4, sendo (O) os resultados relacionados a malha obtida via otimização. Nesta tabela também é apresentado o percentual de elementos classificados como de boa qualidade. A distribuição dos elementos por faixa de qualidade é apresentada na Figura 80.

Os resultados mostram que todos os algoritmos geram malhas de boa qualidade, com alto percentual de elementos classificados como de boa qualidade. O PMGen, novamente, obteve métricas melhores do que o TriGA nos dois exemplos. As malhas geradas via otimização

apresentaram os melhores resultados de métricas. Apesar disto, no caso da placa com furo, há uma leve redução no percentual de elementos da última faixa de qualidade, em relação ao resultado do PMGen. Isto mostra que para elevar a qualidade dos piores elementos da malha, o processo de otimização pode acabar reduzindo a qualidade dos melhores elementos.

Figura 79 – Malhas geradas pelo PMGen nos modelos da placa com furo e bocal.



(a) Placa com furo.

(b) Bocal.

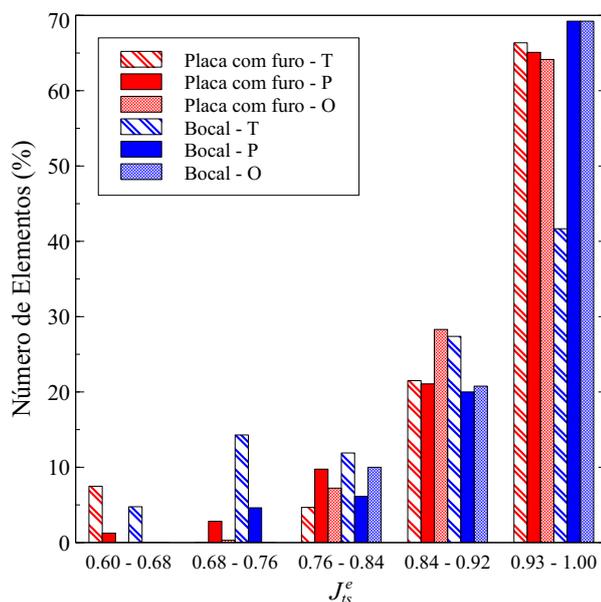
Fonte: Elaborada pelo autor.

Tabela 4 – Qualidade das malhas obtidas nos modelos da placa com furo e bocal.

Exemplo	$\hat{n}$		$J_{ts}$			$J_{ts}^m$			$J_{ts}^{sup}$		
	(T)	(P) e (O)	(T)	(P)	(O)	(T)	(P)	(O)	(T)	(P)	(O)
Placa com furo	214	318	0.6004	0.6652	0.7584	0.9100	0.9243	0.9311	93	96	100
Bocal	84	130	0.5326	0.7288	0.7735	0.8708	0.9294	0.9295	83	96	100

Fonte: Elaborada pelo autor.

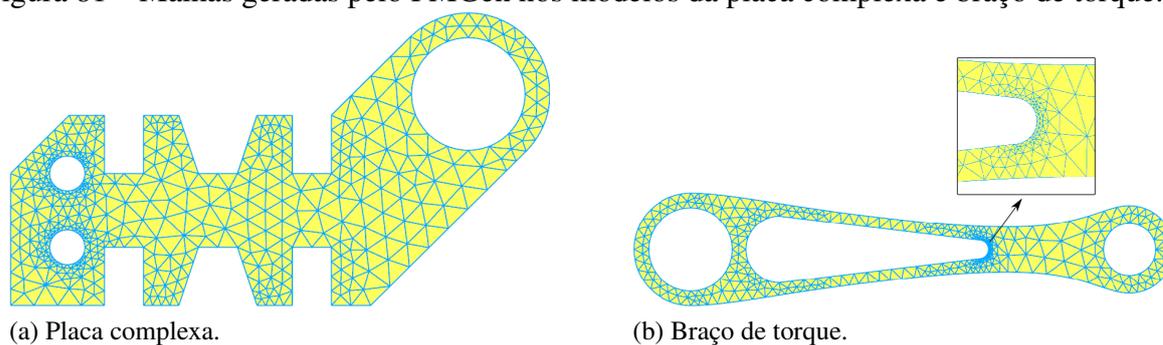
Figura 80 – Distribuição dos elementos por faixas de qualidade nos modelos da placa com furo e bocal.



Fonte: Elaborada pelo autor.

Por fim, mais dois problemas são considerados, uma placa complexa<sup>110, 79</sup> apresentada originalmente em Persson<sup>110</sup>, e um braço de torque inspirado no exemplo estudado por López e colaboradores<sup>8</sup>. As geometrias apresentam maior complexidade em relação aos exemplos anteriores, com vários furos e regiões estreitas. A Figura 81 mostra as malhas obtidas pelo PMGen. A Tabela 4 mostra os resultados de qualidade dos problemas, e a distribuição dos elementos por faixa de qualidade é apresentada na Figura 82.

Figura 81 – Malhas geradas pelo PMGen nos modelos da placa complexa e braço de torque.



Fonte: Elaborada pelo autor.

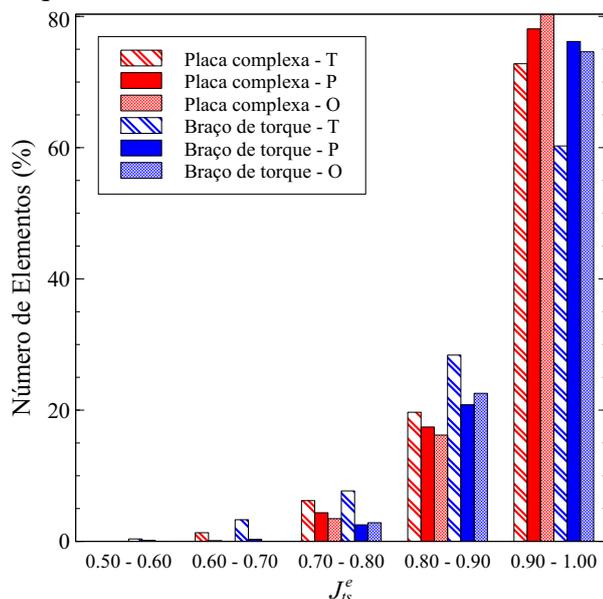
Tabela 5 – Qualidade das malhas obtidas nos modelos da placa complexa e braço de torque.

Exemplo	$\hat{n}$			$J_{ts}$			$J_{ts}^m$		$J_{ts}^{sup}$		
	(T)	(P)	(O)	(T)	(P)	(O)	(T)	(P)	(O)	(T)	(P)
Placa complexa	757	895	0.6461	0.6419	0.7502	0.9258	0.9365	0.9389	97	98	100
Braço de torque	546	634	0.5588	0.5517	0.7089	0.9038	0.9339	0.9314	92	98	99

Fonte: Elaborada pelo autor.

Neste último conjunto de problemas, o PMGen obteve métricas mínimas levemente inferiores às do TriGA, porém métricas médias e distribuição dos elementos melhor. As malhas obtidas via otimização apresentaram as melhores métricas mínimas, entretanto, no caso do braço de torque, não foi possível obter  $J_{ts}^{sup} = 100$ . Destaca-se que o PMGen apresentou a melhor métrica média para a placa complexa. Nos demais resultados, as malhas otimizadas obtiveram as melhores métricas médias e percentual de elementos classificados como de boa qualidade. Além disso, as malhas otimizadas obtiveram melhor distribuição dos elementos para os dois exemplos, com maior percentual de elementos no último intervalo de qualidade.

Figura 82 – Distribuição dos elementos por faixas de qualidade nos modelos da placa complexa e braço de torque.

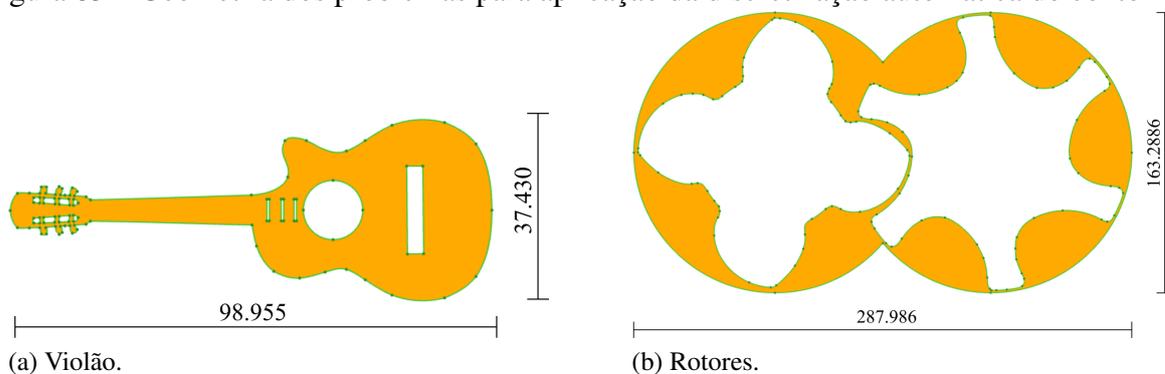


Fonte: Elaborada pelo autor.

### 5.1.2 Aplicação da discretização recursiva em problemas planos

Nesta seção o algoritmo de discretização automática de contorno desenvolvido neste trabalho é aplicado em dois exemplos com geometria complexa, um violão e o fluido existente entre dois rotores, onde a geometria é baseada no problema apresentado por Hinz e colaboradores<sup>118</sup>. O desempenho algoritmo é analisado em função das malhas geradas posteriormente pelo algoritmo AFP, utilizando a parametrização de saída como entrada para geração das malhas. A Figura 83 ilustra os dois problemas.

Figura 83 – Geometria dos problemas para aplicação da discretização automática do contorno.



Fonte: Elaborada pelo autor.

Quatro discretizações são obtidas usando diferentes parâmetros de entrada  $\Lambda = (L_{max}, \theta_{max}, L_{min})$ , de modo a gerar gradativamente malhas mais refinadas para cada problema. A

Tabela 6 – Qualidade das malhas para diferentes níveis de discretização.

$L_{max}$	$\theta_{max}$	$L_{min}$	$\hat{n}$	$J_{ts}$	$J_{ts}^m$	$J_{ts}^{sup}$	$L_{max}$	$\theta_{max}$	$L_{min}$	$\hat{n}$	$J_{ts}$	$J_{ts}^m$	$J_{ts}^{sup}$
12	120°	8	319	0.1595	0.7683	64	30	120°	20	392	0.0000	0.7722	68
12	90°	4	373	0.1598	0.8171	74	25	90°	10	610	0.0004	0.8469	81
8	90°	2	461	0.1598	0.8570	84	20	60°	5	917	0.1983	0.8859	86
4	60°	0	1284	0.4869	0.9395	99	10	30°	0	2572	0.5794	0.9379	98

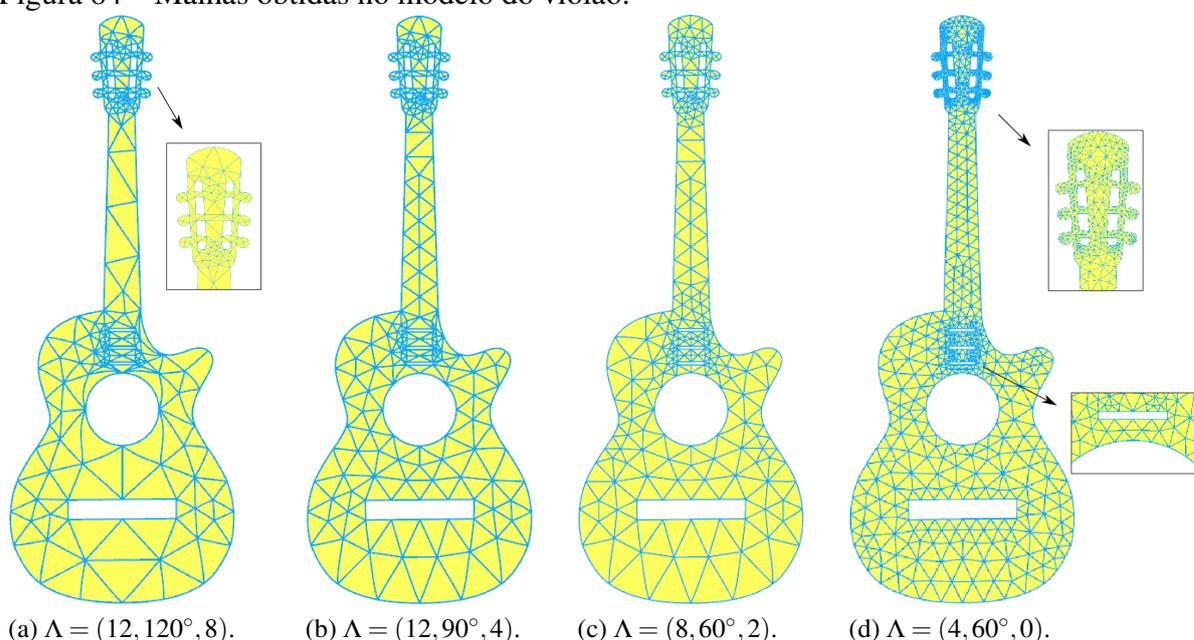
(a) Violão.

(b) Rotores.

Tabela 6 apresenta os parâmetros de entrada do algoritmo e informações relacionadas as malhas geradas, como o número de elementos e as métricas de qualidade obtidas.

A Figura 84 ilustra as malhas geradas no caso do Violão. As malhas iniciais são válidas, mas apresentam elementos de baixa qualidade em regiões estreitas, como na parte superior do modelo e acima do furo circular. As métricas de qualidade média e percentual de elementos classificados como de boa qualidade aumentam com o refinamento. Nota-se, nas malhas mais refinadas, uma maior densidade de elementos nas regiões estreitas, indicando a aplicação do critério de proximidade. A escolha do valor  $L_{min} = 0$  permitiu um refinamento adequado de todas as regiões do modelo, melhorando o valor obtido de  $J_{ts}$  na última discretização considerada.

Figura 84 – Malhas obtidas no modelo do violão.

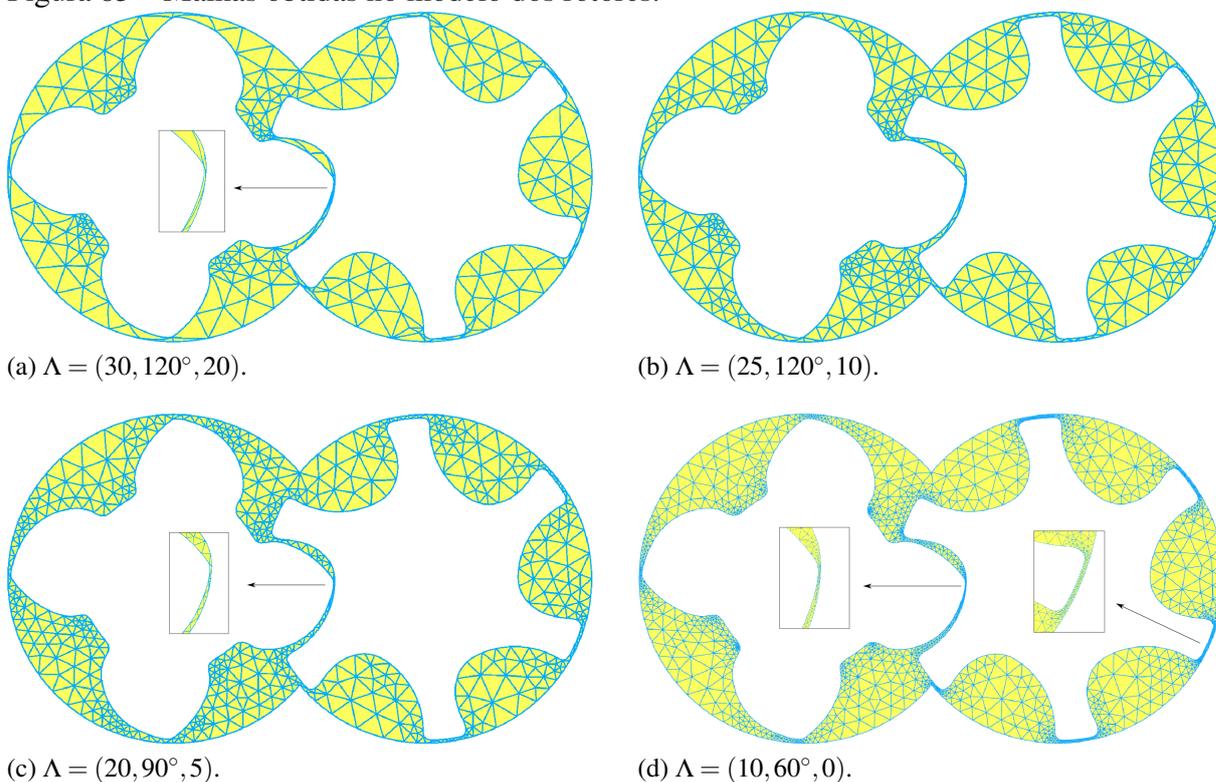


Fonte: Elaborada pelo autor.

A Figura 85 ilustra as malhas obtidas para o caso dos rotores. Neste caso, malhas com elementos inválidos são gerados nos primeiros níveis de discretização. De modo similar ao

caso do violão, a qualidade das malhas melhoram com o aumento do nível de discretização. No último caso, há um refinamento adequado nas regiões mais estreitas do modelo e boa transição nos tamanhos dos elementos. Por outro lado, há um aumento expressivo no número de elementos da malha mais refinada, também observado no caso do violão.

Figura 85 – Malhas obtidas no modelo dos rotores.



Fonte: Elaborada pelo autor.

É importante notar que a escolha de qual malha deva ser utilizada em uma aplicação numérica depende, principalmente, do nível de precisão e custo computacional desejado. É esperado que malhas com maior qualidade apresentem respostas numéricas com maior precisão. Geralmente busca-se utilizar malhas onde a maioria dos elementos seja de boa qualidade e não haja nenhum elemento inválido. O algoritmo proposto mostrou-se capaz de discretizar modelos complexos, permitindo obter malhas de diversos tamanhos e qualidade.

### 5.1.3 Eficiência do algoritmo AFP em geometrias complexas

A eficiência do algoritmo proposto é comparada com a do programa Gmsh<sup>54</sup>, nos exemplos do violão e dos rotores. No Gmsh, é usada a mesma subdivisão de contorno adotada no PMGen, que utiliza os parâmetros apresentados na Tabela 7, elementos finitos cúbicos T10, e

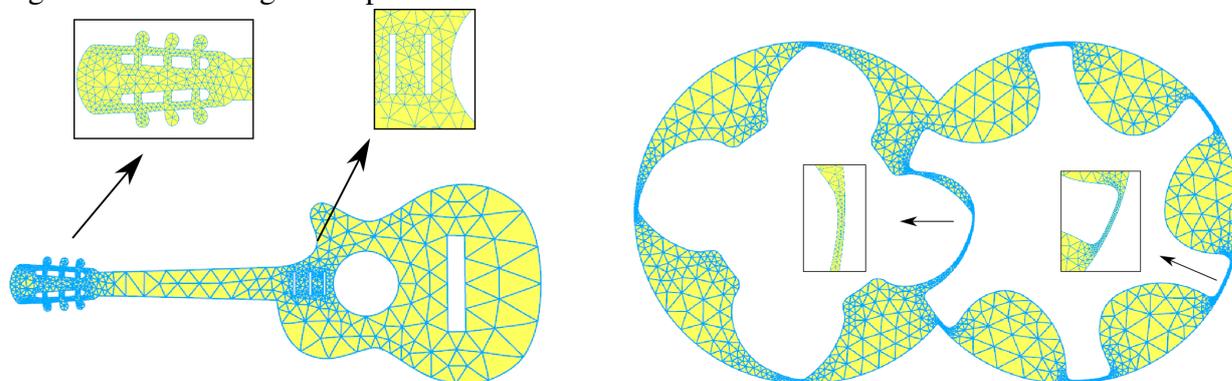
um algoritmo de suavização de elementos de alta ordem baseado na otimização de métricas de qualidade, disponível no Gmsh<sup>87, 86, 84</sup>. As malhas obtidas pelo PMGen com a parametrização considerada são ilustradas na Figura 86. Os parâmetros utilizados no Gmsh são apresentados na Tabela 8.

Tabela 7 – Parâmetros de discretização do contorno usados na comparação com o Gmsh.

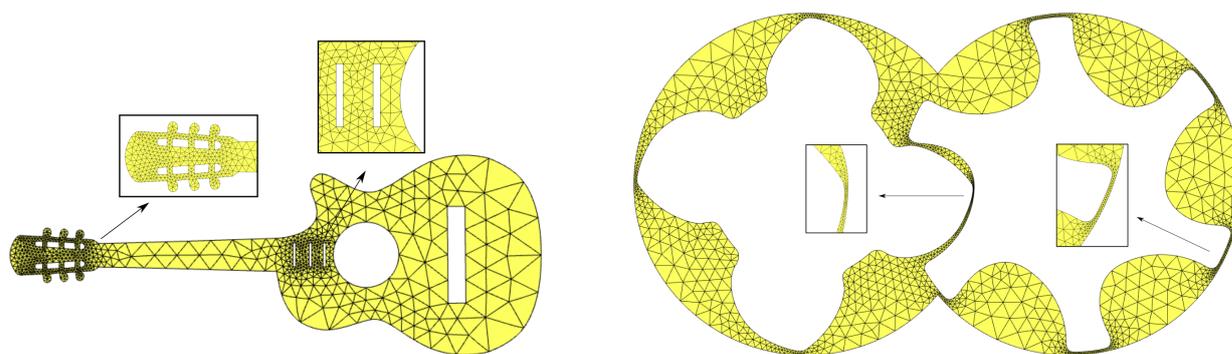
Exemplo	$L_{max}$	$L_{min}$	$\theta_{max}$
Violão	8	0	75°
Rotores	30	0	60°

Fonte: Elaborada pelo autor.

Figura 86 – Malhas geradas pelo PMGen e Gmsh.



(a) PMGen.



(b) Gmsh.

Fonte: Elaborada pelo autor.

É importante notar que os dois algoritmos não são diretamente comparáveis, pois as malhas geradas pelo Gmsh não representam geometrias racionais de forma exata. Entretanto, a capacidade de produzir malhas de alta ordem válidas em geometrias complexas ainda é uma característica relevante para os dois programas. Além disso, o Gmsh é um programa maduro de geração de malhas de elementos finitos com algoritmos eficientes e é usado em diversos

Tabela 8 – Parâmetros do Gmsh (versão 4.4.1) usados para gerar as malhas do violão e rotores.

Parâmetro	Valor
Linear meshing algoritm	Frontal-Delaunay
Regularization algorithm	Optimization
Target jacobian range	0.8 - 1.2
Number of layers	6
Distance factor	12
Boundary nodes	Fixed
Weight on node displacement	1
Maximum number of iterations	100
Max. number of barrier updates	25
Strategy	Disjoint strong

Fonte: Elaborada pelo autor.

programas CAE. Portanto, é razoável a comparação entre os dois programas com relação à eficiência computacional e qualidade das malhas, considerando que a mesma subdivisão do contorno e grau dos elementos sejam adotados para ambos.

A Tabela 9 mostra a qualidade das malhas obtidas (P para PMGen e G para GMsh), e a Tabela 10 mostra os tempos de execução obtidos em cada caso, onde  $t_h$  é o tempo de execução da SAO, incluindo a etapa de elevação de grau, e  $t_l$  é o tempo de execução da etapa de geração de malha linear. O algoritmo proposto gerou malhas de alta ordem com boa qualidade para os dois casos, com métricas levemente inferiores às obtidas pelo Gmsh. Apesar disso, o algoritmo proposto foi mais rápido e produziu malhas com número menor de elementos. A etapa de SAO representa uma parte considerável do tempo de execução total do algoritmo proposto, mas é comparável com o tempo de execução da etapa de geração da malha linear. Vale ressaltar que apesar do procedimento de otimização usado no Gmsh resultar em elementos com qualidade superior aos obtidos pela SAO proposta, o custo computacional dele foi maior.

Tabela 9 – Tamanho e qualidade das malhas obtidas pelo PMGen e Gmsh.

Exemplo	$\hat{n}$ (P)	$\hat{n}$ (G)	$J_{ts}$ (P)	$J_{ts}$ (G)	$J_{ts}^m$ (P)	$J_{ts}^m$ (G)	$J_{ts}^{sup}$ (P)	$J_{ts}^{sup}$ (G)
Violão	930	1240	0.464	0.600	0.922	0.948	97	98
Rotores	2117	2427	0.565	0.653	0.939	0.950	98	99

Fonte: Elaborada pelo autor.

A Tabela 11 mostra o número de sub-malhas ( $n_{sm}$ ) e o número de elementos nas sub-malhas ( $\hat{n}_{sm}$ ) obtidas em cada caso. Pode-se notar que a análise térmica é aplicada em um único grupo de elementos nos dois casos, enquanto a análise elástica é aplicada várias vezes em grupos de elementos disjuntos. O grupo de elementos disjuntos escolhido na análise térmica

Tabela 10 – Tempos de processamento obtidos pelo PMGen e Gmsh.

Exemplo	$t_l^*$		$t_h^*$		$t_h + t_l$		$t_h/t_l$	
	(P)	(G)	(P)	(G)	(P)	(G)	(P)	(G)
Violão	63	46	67	471	130	517	0.92	10.24
Rotores	164	95	190	302	354	397	0.95	3.18

\*Tempo de processamento medido em milissegundos.

Fonte: Elaborada pelo autor.

está localizado nas curvas racionais dos modelos. Estes modelos apresentam apenas um ciclo com curvas racionais, o furo circular no modelo do violão e o contorno externo no modelo dos rotores. Os ciclos compostos apenas por curvas racionais sempre resultam em um único grupo de elementos, como observado no caso dos rotores, onde quase metade dos elementos da malha são considerados na etapa da análise térmica. Este aspecto afeta o custo computacional do algoritmo de geração de malhas e pode ser considerado em alguns modelos para exclusão da etapa de análise térmica, preservando a eficiência computacional do algoritmo.

Tabela 11 – Dados referentes as sub-malhas do violão e rotores.

Modelo	$n_{sm}$		$\hat{n}_{sm}$	
	Elástica	Térmica	Elástica	Térmica
Violão	8	1	470	91
Rotores	17	1	469	1102

Fonte: Elaborada pelo autor.

Por fim, a Tabela 12 mostra os resultados de métricas de qualidade e tempo de processamento para alguns dos exemplos tratados neste trabalho, considerando a aplicação da SAO de forma localizada (*Loc*) ou global (*Glob*), onde todos os elementos da malha são considerados na suavização dos pesos e coordenadas dos pontos de controle. Pode-se observar variações desprezíveis nas métricas de qualidade das malhas, enquanto há ganhos expressivos de eficiência computacional.

Tabela 12 – Métricas de qualidade e custo computacional da SAO em análises globais e locais.

Modelo	$J_{ts}$		$J_{ts}^m$		$t_h/t_l$	
	<i>Loc</i>	<i>Glob</i>	<i>Loc</i>	<i>Glob</i>	<i>Loc</i>	<i>Glob</i>
Placa complexa	0.64	0.64	0.94	0.94	0.89	3.72
Braço de Torque	0.55	0.55	0.93	0.93	1.70	2.98
Violão	0.46	0.46	0.92	0.92	1.08	4.75
Rotores	0.56	0.57	0.94	0.94	1.16	3.77

Fonte: Elaborada pelo autor.

### 5.1.4 Efeito da SAO na qualidade das malhas

Nesta seção é estudado o efeito da SAO proposta sobre a qualidade das malhas. A geometria consiste em uma parte mecânica apresentada por Zienkiewicz e colaboradores<sup>156</sup>. O contorno do problema é subdividido usando o algoritmo proposto neste trabalho, com  $L_{min} = 0$ ,  $\theta_{max} = 90^\circ$  e  $L_{max} = 6, 4$  e  $2$ . Malhas quadráticas, cúbicas e quárticas são geradas pelo algoritmo AFP em cada caso, com e sem a aplicação da etapa da SAO. Neste exemplo, as métricas  $J_{ts}$  e  $J_{ts}^m$  são avaliadas considerando apenas elementos afetados pela SAO, dado que não há efeito nos demais.

A Tabela 13 mostra as qualidades das malhas obtidas. Ambas as métricas mínimas e médias aumentam em todos os casos e tendem a ser muito melhores para os graus mais elevados. De modo geral, a qualidade dos elementos é boa mesmo sem a aplicação da SAO, porém elementos de baixa qualidade ou inválidos são gerados em alguns casos. Além disto, a distribuição dos elementos pela qualidade melhora em todos os casos quando aplicada a SAO, como mostrado na Figura 88.

Tabela 13 – Influência da SAO na qualidade das malhas e tempos de processamento.

$L_{max}$	Degree	$J_{ts}$	$J_{ts}$ (SAO)	$J_{ts}^m$	$J_{ts}^m$ (SAO)	$\hat{n}$	$*n_{sm}$	$*\hat{n}_{sm}$	$t_h/t_l$
6	2	0.0319	0.4470	0.9135	0.9226	200	1 (1)	115 (115)	1.10
	3	0.0000	0.5648	0.9132	0.9255				2.27
	4	0.0000	0.6113	0.9056	0.9202				5.20
4	2	0.2471	0.4634	0.9094	0.9095	399	1 (1)	134 (134)	0.72
	3	0.2471	0.6170	0.9096	0.9165				1.61
	4	0.1448	0.6823	0.9023	0.9155				3.65
2	2	0.6538	0.7265	0.9428	0.9419	1248	4 (4)	215 (215)	0.41
	3	0.6538	0.7275	0.9428	0.9416				0.84
	4	0.6519	0.7275	0.9411	0.9389				1.41

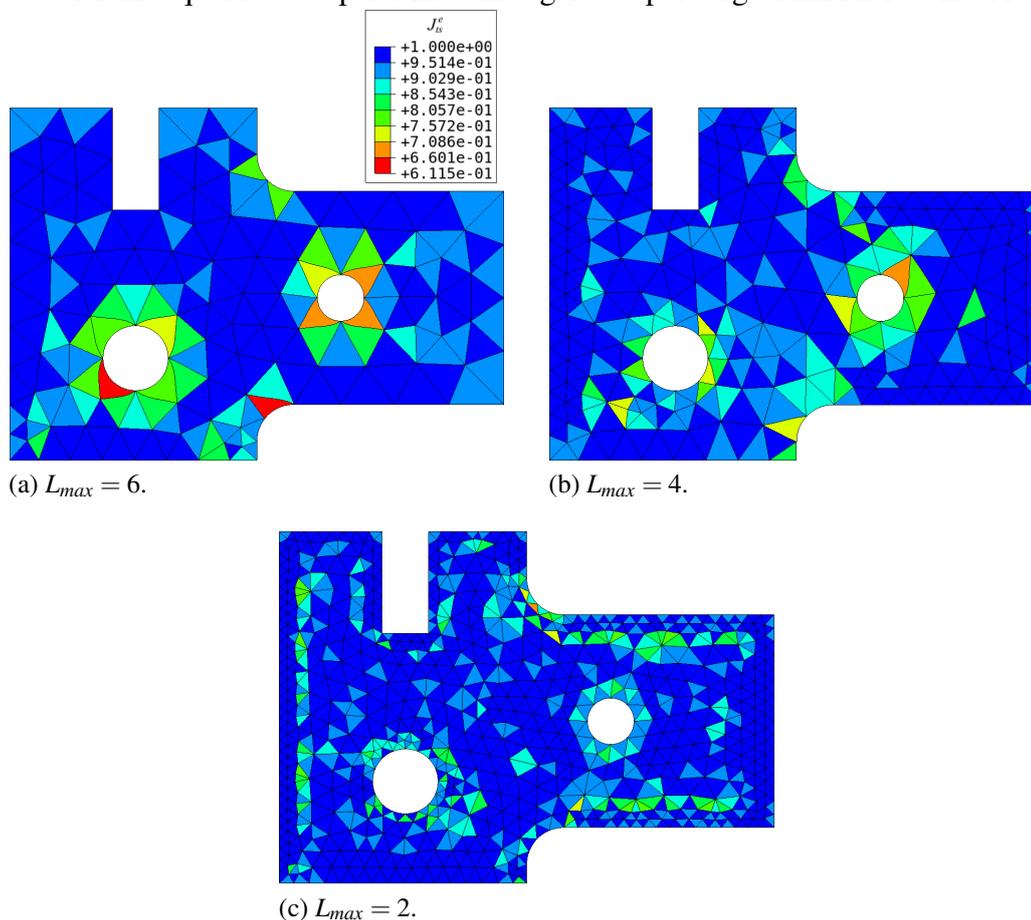
\*Valores reportados nas análises elástica e (térmica).

Fonte: Elaborada pelo autor.

É importante ressaltar que o efeito da suavização de coordenadas é reduzido com a diminuição da curvatura dos elementos, onde conseqüentemente, a magnitude dos deslocamentos prescritos torna-se pequena. Esse comportamento é observado nos resultados apresentados a medida que o parâmetro de discretização  $L_{max}$  diminui, por exemplo, quando  $L_{max} = 2$ , onde boas malhas são geradas mesmo sem a etapa SAO. Nota-se que o grupo de elementos afetados pela SAO localizada adapta-se conforme a discretização, resultando em um conjunto vazio para

a etapa de análise elástica. Logo, apenas a suavização dos pesos é aplicada em níveis muito refinados de discretização do modelo. A Figura 87 apresenta as malhas quárticas (com SAO) obtidas em cada discretização e sua distribuição de qualidade.

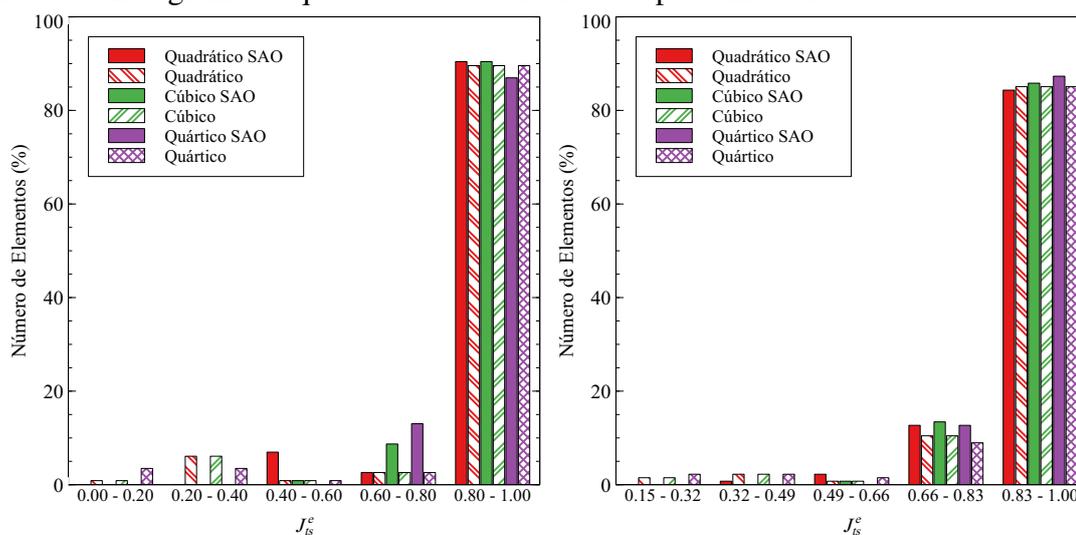
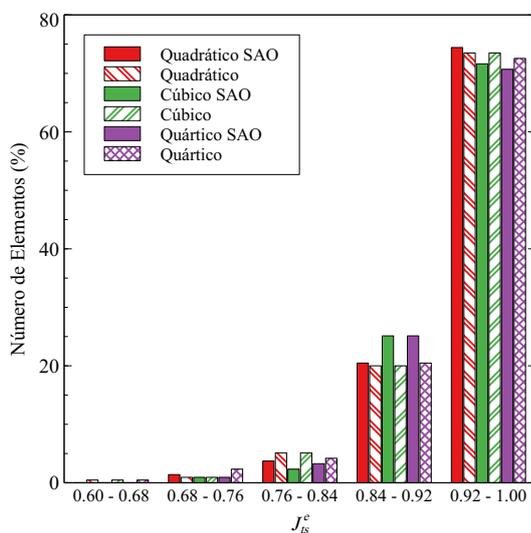
Figura 87 – Malhas quárticas da parte mecânica geradas pelo algorithm AFP com SAO.



Fonte: Elaborada pelo autor.

A razão entre os tempos de processamento  $t_h/t_l$  para cada caso também são reportados aqui. A etapa de SAO requer maior tempo de processamento em comparação a etapa de geração de malha linear a medida que graus mais elevados são usados para o mesmo nível de discretização do modelo. Vale ressaltar que, neste caso, tanto o tamanho do sistema linear global aumenta como também aumenta o custo computacional da integração numérica das matrizes dos elementos. Por outro lado, para um mesmo grau, a razão  $t_h/t_l$  diminui com o aumento do nível de discretização do modelo.

Figura 88 – Histograma de qualidade dos elementos da parte mecânica.

(a)  $L_{max} = 6$ .(b)  $L_{max} = 4$ .(c)  $L_{max} = 2$ .

Fonte: Elaborada pelo autor.

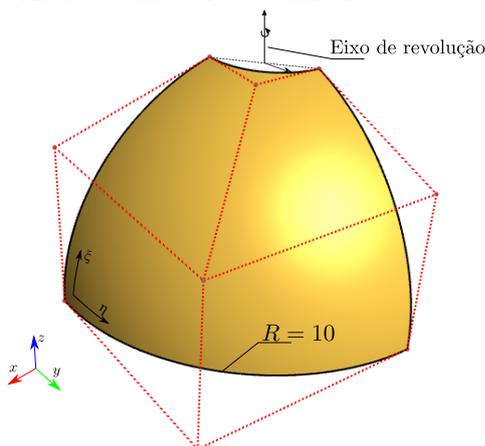
### 5.1.5 Geração de malhas de superfícies

Nesse seção, o algoritmo AFS é aplicado para geração de malhas de superfícies. O desempenho do algoritmo é avaliado pela qualidade das malhas geradas, considerando a métrica  $J_{TS}^e$  formulada para o caso de superfícies no espaço 3D, onde a matriz Jacobiana tem dimensão  $3 \times 2$ . As malhas geradas são comparadas com as obtidas pelo processo de otimização descrito na Seção 3.6.1, que recebe como entrada a malha gerada pelo AFS, considerando inclusive a etapa de SAO. Desta forma, busca-se obter a melhor malha possível para a topologia construída pela AFS, em termos das métricas  $J_{TS}$  e  $J_{TS}^m$ , e comparar sua qualidade a malha obtida sem otimização.

A otimização é realizada com o mesmo algoritmo e parâmetros apresentados na Seção 5.1.1. Apesar do algoritmo adotado não avaliar critério de otimalidade, observou-se que o uso deste com os parâmetros considerados possibilitou otimizar consistentemente os exemplos tratados neste trabalho.

Inicialmente, são consideradas duas superfícies NURBS sem aparos, onde não há utilização da SAO. O primeiro exemplo consiste em um oitavo de uma esfera com corte de  $18^\circ$  no polo. O modelo é representado por uma superfície NURBS biquadrática com 9 pontos de controle, que pode ser obtido pela revolução de um arco circular de  $72^\circ$  em torno do eixo polar<sup>38</sup>, conforme ilustrado na Figura 89. A superfície é parametrizada em valores unitários, no intervalo de  $[0, 1]$  nas direções  $\xi$  e  $\eta$ . Essa forma padrão de parametrização é comumente adotada nos modeladores e será considerada em todos os exemplos deste trabalho.

Figura 89 – Descrição da superfície do modelo de um oitavo de esfera com corte.



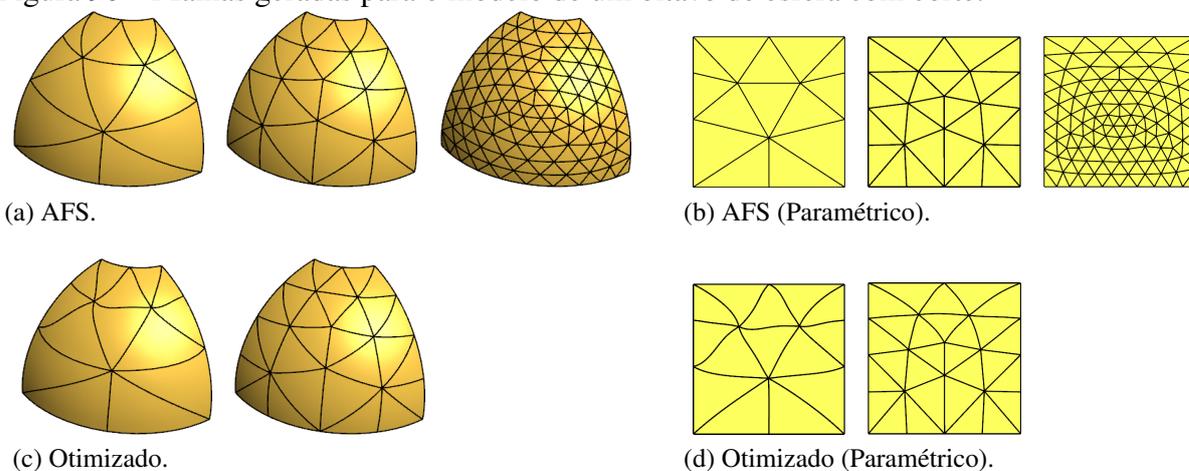
Fonte: Elaborada pelo autor.

Três malhas são obtidas considerando subdivisões do contorno e parâmetro  $\theta_{max}^S$  diferentes, de maneira a gerar malhas gradativamente mais refinadas. O algoritmo de discretização automática de contorno, generalizado para o caso de superfícies na Seção 3.5.4, é utilizado. Nos exemplos deste trabalho, foi adotado  $\theta_{max}^S = \theta_{max}$ , de modo que o mesmo ângulo limite é considerado tanto para discretização do contorno quanto para geração da malha de superfície.

Por se tratar de uma superfície biquadrática, são considerados elementos triangulares de quarto grau para representar exatamente a superfície NURBS. Vale ressaltar que no caso da malha otimizada, onde as arestas dos elementos deixam de ser retas no espaço paramétrico, não há garantia de representação exata da geometria, embora o processo de ajuste da superfície do elemento já garanta boa aproximação. O mesmo ocorre nos elementos próximos às curvas de

aparo, bem como aqueles afetados pela SAO.

Figura 90 – Malhas geradas para o modelo de um oitavo de esfera com corte.



Fonte: Elaborada pelo autor.

Tabela 14 – Qualidade das malhas para o modelo de um oitavo de esfera com corte.

$L_{max}$	$\theta_{max}$	$L_{min}$	$\hat{n}$	$J_{ts}$		$J_{ts}^m$		$J_{ts}^{sup}$	
				(S)	(O)	(S)	(O)	(S)	(O)
8	60°	2	15	0.4989	0.7717	0.7857	0.8602	71	100
5	45°	1	28	0.7254	0.8178	0.8911	0.9224	93	100
2	30°	0	188	0.7941	-	0.9523	-	100	-

Fonte: Elaborada pelo autor.

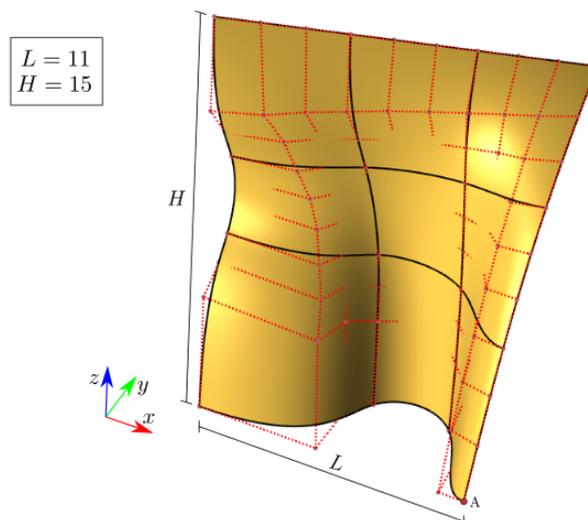
As malhas obtidas são apresentadas na Figura 90, onde também são ilustradas a representação destas no espaço paramétrico. As métricas de qualidade obtida são apresentadas na Tabela 14, onde (S) representa os resultados do AFS e (O) o das malhas otimizadas. O algoritmo AFS consegue gerar malhas de boa qualidade mesmo nos primeiros níveis de refinamento. Apesar disto, as malhas otimizadas obtiveram métricas superiores ao AFS em todos os casos. No terceiro caso não foi realizada a otimização, pois a malha gerada pelo AFS já apresentava  $J_{ts}^{sup}$  máximo. O bom desempenho do AFS é esperado pela facilidade do problema, que apresenta curvatura constante e ausência de furos.

Em seguida é considerada uma superfície de forma livre proposta por Dornisch e colaboradores<sup>153</sup>, apresentada na Figura 91. A superfície é bicúbica, não-racional e possui 36 ponto de controle.

Neste exemplo, a superfície apresenta variação considerável da curvatura. As maiores curvaturas principais absolutas são observadas nas proximidades do ponto A, na quina inferior

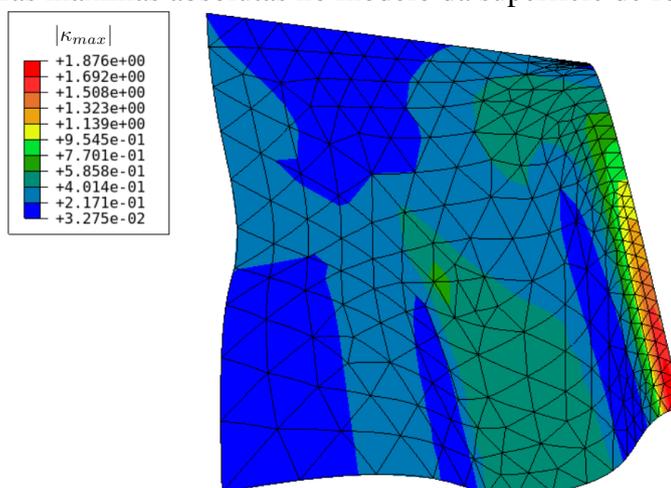
direita do modelo. A Figura 92 mostra a maior curvatura principal em valores absolutos  $|\kappa_{max}|$  ao longo da superfície.

Figura 91 – Descrição da superfície de fomra livre.



Fonte: Elaborada pelo autor.

Figura 92 – Curvaturas máximas absolutas no modelo da superfície de forma livre.



Fonte: Elaborada pelo autor.

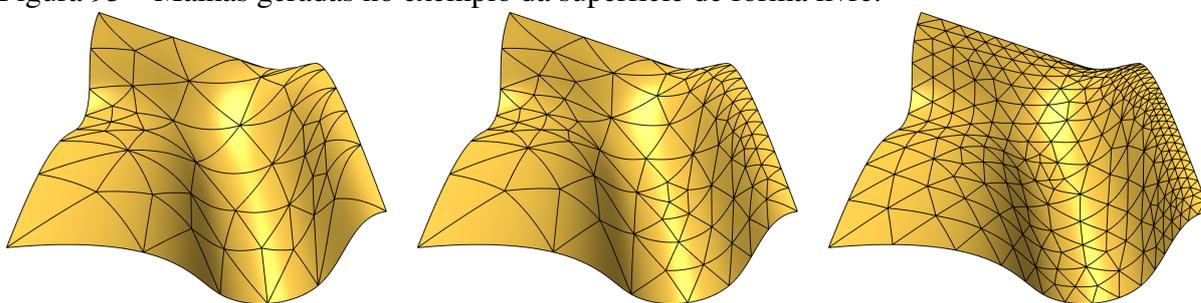
Além das malhas geradas pelo algoritmo AFS e as obtidas via otimização, também são apresentadas malhas estruturadas com elementos triangulares obtidas pelo algoritmo MTB, considerando subdivisão uniforme nas curvas de contorno da superfície. Foram escolhidos parâmetros de discretização de modo a gerar malhas de tamanho similar para ambas as técnicas. O grau sexto é adotado nas malhas para possibilitar a representação exata da superfície bicúbica.

As três malhas geradas para cada técnica são apresentadas na Figura 93. Pode-se observar, para o AFS, maior refinamento nas proximidades do ponto A devido ao critério de

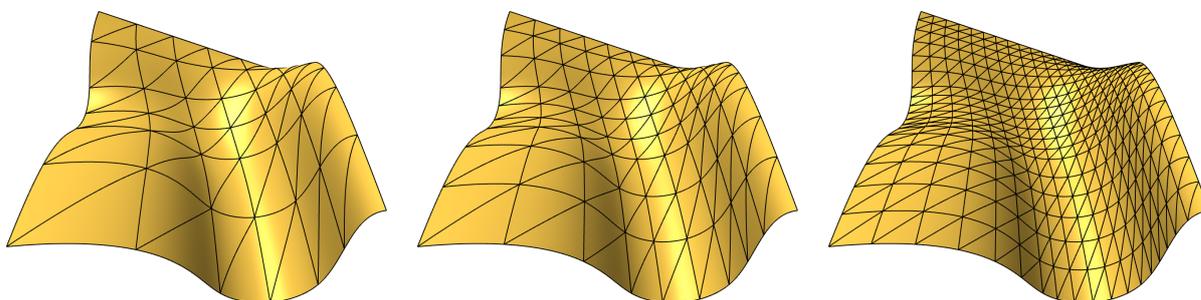
curvatura adotado nas etapas de geração da malha e discretização do contorno.

Os resultados de qualidade são apresentados na Tabela 15. O algoritmo AFS gerou malhas de boa qualidade em todos os casos considerados, obtendo métricas médias quase iguais às obtidas pela otimização, além de alto percentual de elementos de boa qualidade. Além disto, o AFS obteve métricas superiores em comparação as obtidas pelo MTB em todos os casos, evidenciando a importância da consideração da curvatura durante a geração da malha para o exemplo analisado. Por fim, ressalta-se que a estratégia de discretização baseado na curvatura proposta na AFS mostrou-se satisfatória nos dois exemplos analisados.

Figura 93 – Malhas geradas no exemplo da superfície de forma livre.



(a) AFS.



(b) MTB.

Fonte: Elaborada pelo autor.

Tabela 15 – Qualidade das malhas para o modelo da superfície de forma livre.

$L_{max}$	$\theta_{max}$	$L_{min}$	$\hat{n}$	$J_{ts}$		$J_{ts}^m$		$J_{ts}^{sup}$	
				(S)	(O)	(S)	(O)	(S)	(O)
8	90	2	90	0.6746	0.7716	0.8711	0.8889	92	100
5	60	1	169	0.6985	0.7755	0.9131	0.9181	98	100
2	30	0	521	0.7120	0.7629	0.9399	0.9408	100	100

(a) AFS.

Elementos por lado	$\hat{n}$	$J_{ts}$	$J_{ts}^e$	$J_{ts}^{sup}$
6	72	0.5590	0.7579	53
9	162	0.5762	0.7921	67
18	648	0.5934	0.8220	77

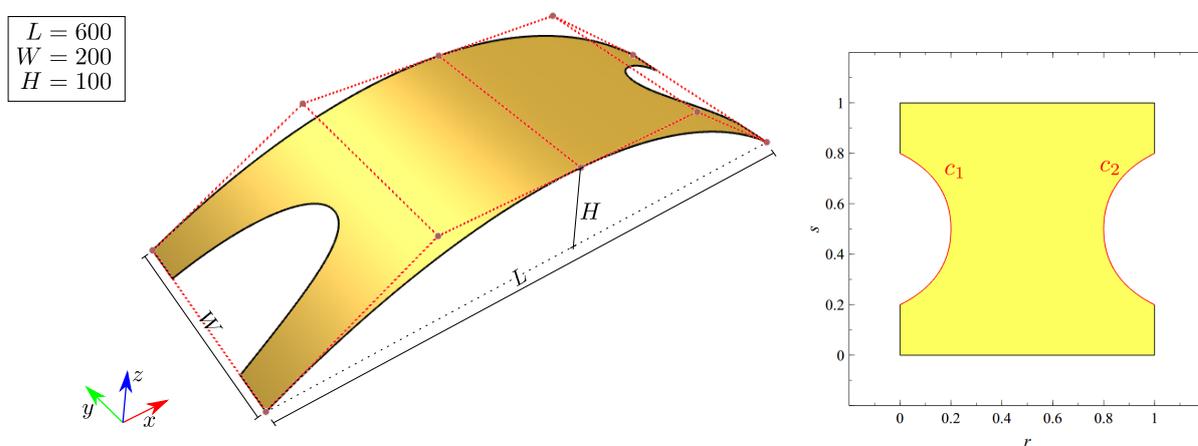
(b) MTB.

Fonte: Elaborada pelo autor.

### 5.1.6 Geração de malhas em superfícies aparadas

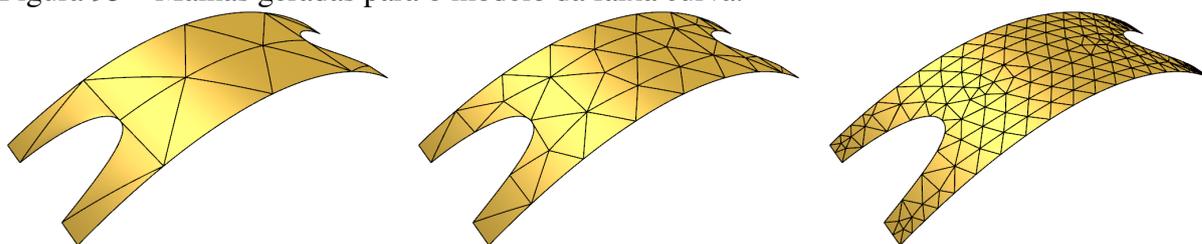
A seguir o algoritmo AFS é aplicado em três geometrias com aparos. O primeiro é uma faixa curva, com a geometria baseada no exemplo apresentado por Kang e Youn<sup>157</sup>. A geometria tratada na referência não é utilizada, pois sua descrição completa não é fornecida no trabalho. Vale ressaltar a dificuldade na reprodução de exemplos de superfícies aparadas disponíveis na literatura. A geometria da faixa curva é dada por uma superfície não-racional de grau  $2 \times 1$ , com duas curvas de aparo quadráticas. A Figura 94 ilustra o modelo 3D, no espaço cartesiano, e também o modelo bidimensional no espaço paramétrico.

Figura 94 – Descrição do modelo da faixa curva.

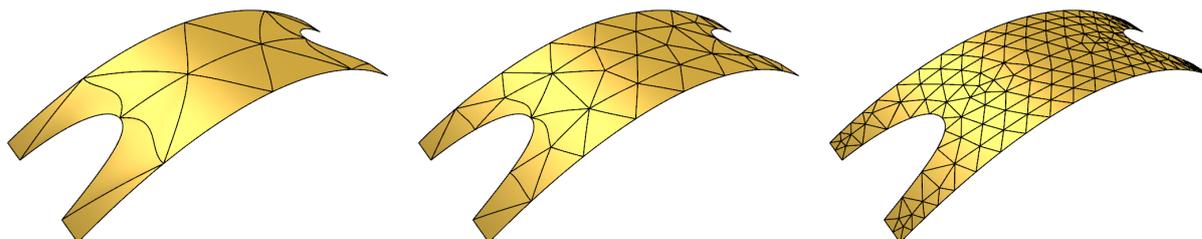


Fonte: Elaborada pelo autor.

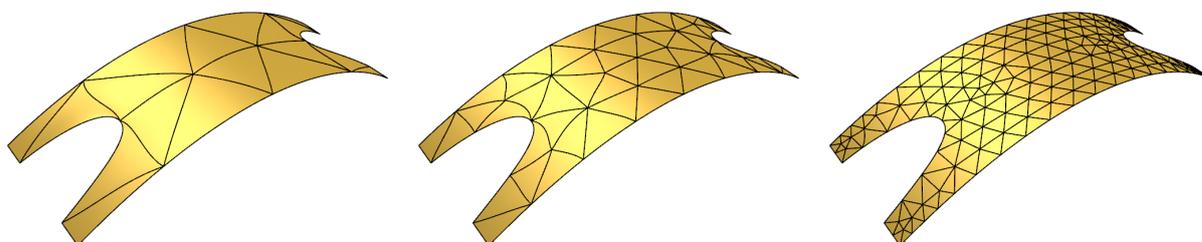
Figura 95 – Malhas geradas para o modelo da faixa curva.



(a) AFS sem SAO.



(b) AFS.



(c) Otimizado.

Fonte: Elaborada pelo autor.

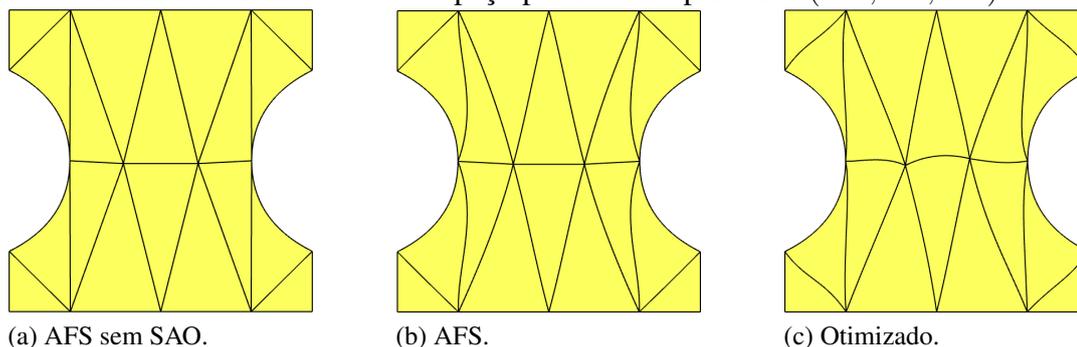
Tabela 16 – Qualidade das malhas para o modelo da faixa curva.

$L_{max}$	$\theta_{max}$	$L_{min}$	$\hat{n}$	$J_{ts}$			$J_{ts}^m$			$J_{ts}^{sup}$		
				(T)	(S)	(O)	(T)	(O)	(T)	(S)	(O)	
300	90°	150	18	0.0808	0.4150	0.4419	0.6100	0.6269	0.7183	56	33	56
150	60°	75	56	0.0016	0.4479	0.7412	0.8445	0.8416	0.8883	86	79	86
75	45°	25	264	0.6888	0.6888	0.7838	0.9420	0.9420	0.9469	98	98	100

Três malhas com tamanhos variados são obtidas utilizando diferentes parâmetros para discretização do contorno. No caso de superfícies aparadas, o algoritmo AFS aplica a etapa SAO nas arestas curvas do espaço paramétrico. Assim, para ilustrar a importância da SAO, são mostradas as malhas geradas pelo AFS com SAO (S) e sem SAO (T), além das malhas otimizadas. Neste exemplo são consideradas malhas cúbicas. As malhas obtidas são apresentadas na Figura 95 e a Tabela 16 mostra os resultados de qualidade obtidos.

Observa-se que o uso da SAO melhorou consistentemente a métrica mínima ( $J_{ts}$ ) nos dois primeiros casos, onde há geração de elementos inválidos no algoritmo (T). Vale notar que os elementos inválidos ocorrem também no espaço paramétrico, como mostra a Figura 96.

Figura 96 – Malhas da faixa curva no espaço paramétrico para  $\Lambda = (300, 90^\circ, 150)$ .

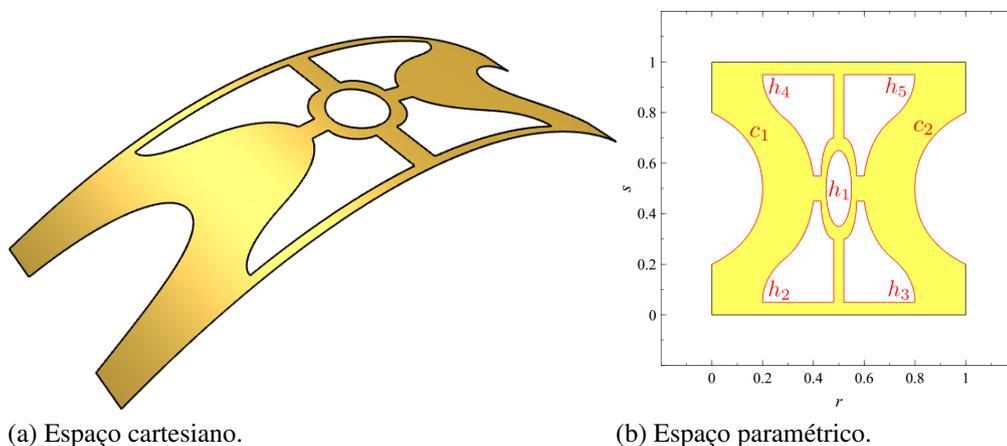


Fonte: Elaborada pelo autor.

No terceiro caso, não houve aplicação da SAO devido ao alto nível de refinamento considerado, resultando em malhas idênticas para (T) e (S). Vale notar que a etapa de SAO é aplicada de forma eficiente, não sendo utilizada em regiões muito discretizadas onde não há ocorrência de elementos inválidos. Com relação as métricas médias, as duas alternativas (T e S) apresentam respostas similares, exceto na primeira discretização considerada, onde o algoritmo com suavização apresenta redução na resposta  $J_{IS}^{SUP}$ . O algoritmo com a otimização apresentou melhores métricas em todos os casos, mas a diferença em relação ao (S) diminuiu com o aumento do refinamento, apresentando métricas médias quase iguais no último caso.

Apesar da geometria apresentar aparos, a mesma não possui características complexas. Portanto, é esperado bom desempenho dos algoritmos neste exemplo. A seguir, a mesma geometria é modificada, adicionando vários furos para aumentar a dificuldade do problema de geração de malha. A Figura 97 ilustra a geometria da faixa curva com vários furos no espaço cartersiano e paramétrico.

Figura 97 – Descrição do modelo da faixa curva modificada.



Fonte: Elaborada pelo autor.

Figura 98 – Malhas geradas no exemplo da faixa modificada.



Fonte: Elaborada pelo autor.

Três malhas são geradas utilizando parâmetros de refinamento diferentes dos usados anteriormente. As malhas para cada caso são ilustrados na Figura 98, e os resultados obtidos são apresentados na Tabela 17. Na primeira discretização considerada, o nível de refinamento da malha na região próxima ao furo central foi insuficiente para evitar elementos de baixa qualidade. Elementos inválidos são gerados pelo algoritmo sem suavização, enquanto malhas válidas são obtidas quando se usa a SAO. Neste caso, a malha otimizada obteve melhor valor de métrica mínima ( $J_{ts}$ ), mas obteve os piores valores de métrica média ( $J_{ts}^m$ ) e percentual de elementos de boa qualidade ( $J_{ts}^{sup}$ ). Portanto, neste caso, a otimização da métrica mínima prejudicou consideravelmente a qualidade dos demais elementos da malha. A malha com suavização (S) também obteve respostas  $J_{ts}^m$  e  $J_{ts}^{sup}$  inferiores quando comparado a malha não suavizada (T).

Nas outras discretizações, nenhum algoritmo gera elementos inválidos e as respostas das malhas suavizadas apresentaram melhores métricas em comparação às malhas sem suavização. Além disto, as respostas das malhas otimizadas também são melhores ou similares em comparação as demais respostas. Conclui-se que embora os algoritmos com suavização e

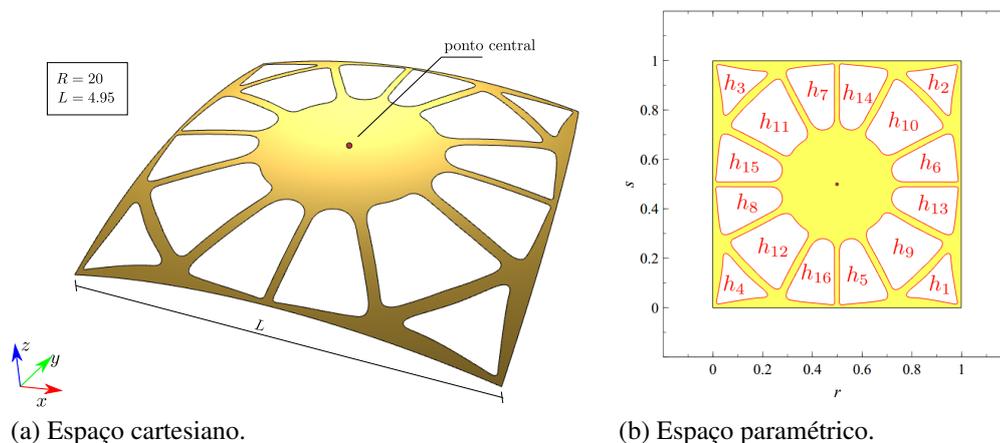
otimização tenham maior robustez na geração de malhas válidas, dependendo da complexidade da geometria e nível de discretização adotado, estes podem piorar o restante da malha para garantir a validade da mesma.

O último problema tratado nesta seção consiste em uma casca esférica com vários furos, baseada no exemplo apresentado por Kang e Youn<sup>158</sup>. A geometria da casca é dada por uma superfície racional biquadrática, com 16 curvas de aparo quadráticas, e apresenta simetria nos eixos  $x$  e  $y$ . A Figura 99 ilustra o modelo no espaço cartesiano (3D) e no espaço paramétrico.

Tabela 17 – Qualidade das malhas para o modelo da faixa modificada.

$L_{max}$	$\theta_{max}$	$L_{min}$	$\hat{n}$	$J_{ts}$			$J_{ts}^m$			$J_{ts}^{sup}$		
				(T)	(S)	(O)	(T)	(S)	(O)	(T)	(S)	(O)
150	90°	50	172	0.0002	0.2096	0.3194	0.7603	0.7430	0.6981	65	62	53
100	60°	25	336	0.4580	0.5525	0.6868	0.8655	0.8655	0.8817	66	83	87
50	30°	0	1480	0.6582	0.6471	0.6948	0.9408	0.9407	0.9398	67	99	99

Figura 99 – Descrição do modelo da casca esférica com furos.



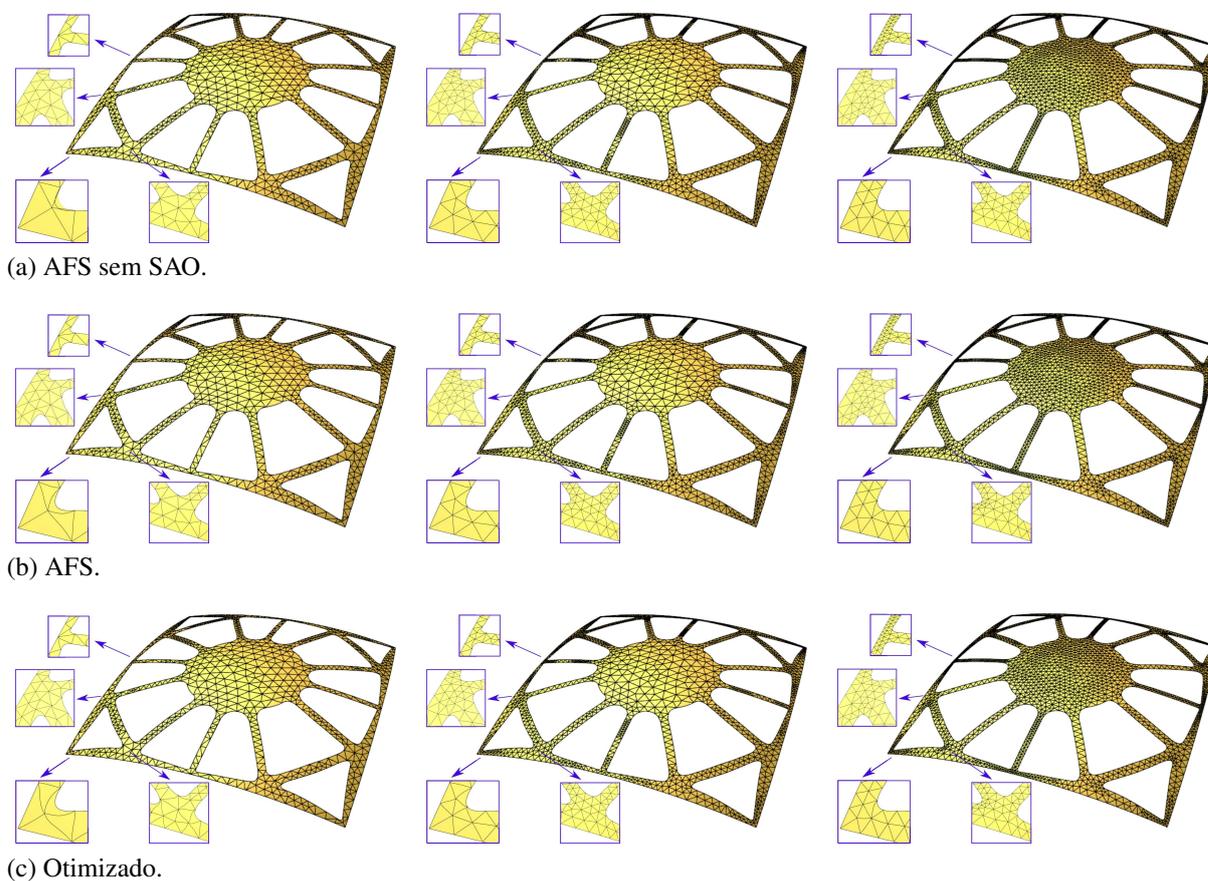
Fonte: Elaborada pelo autor.

O modelo geométrico possui várias regiões estreitas e arestas com curvatura considerável. Além disto, o modelo possui uma restrição de um ponto localizado no centro do modelo. É importante ressaltar a capacidade da implementação realizada neste trabalho para considerar restrições, que naturalmente são satisfeitas durante a etapa de geração da malha linear. Tal restrição é utilizada mais adiante para modelagem de um problema estrutural, onde há aplicação de carga pontual no centro do modelo.

Novamente, são geradas malhas para três conjuntos de parâmetros de discretização de forma a aumentar a resolução da malha gradativamente. Além disso, é considerado o grau

quártico para os elementos. A Figura 100 mostra as malhas obtidas por cada algoritmo e Tabela 18 apresenta os resultados de qualidade obtidos.

Figura 100 – Malhas geradas no exemplo casca esférica com furos.



Fonte: Elaborada pelo autor.

No geral, todos os algoritmos conseguiram gerar malhas com boa qualidade, obtendo bom valor de métricas médias  $J_{ts}^m$  e alto percentual de elementos classificados como de boa qualidade. Na primeira discretização considerada, observam-se refinamentos insuficientes em algumas partes do modelo, como nas quinas do painel e regiões entre os furos. Apesar disto, os algoritmos com suavização (S) e otimização (O) geraram malhas válidas, embora com alguns elementos de baixa qualidade. De maneira similar ao observado no exemplo anterior, estes algoritmos obtiveram métricas  $J_{ts}^m$  e  $J_{ts}^{sup}$  inferiores ao algoritmo sem suavização para primeira discretização.

Nas demais discretizações, as malhas com suavização e otimização apresentam bons resultados para todas as respostas analisadas. A etapa de suavização foi essencial para garantir a validade das malhas geradas, uma vez que em todos os casos processados o algoritmo sem

suavização (T) gerou elementos inválidos.

Tabela 18 – Qualidade das malhas para o modelo de casca esférica.

$L_{max}$	$\theta_{max}$	$L_{min}$	$\hat{n}$	$J_{ts}$			$J_{ts}^m$			$J_{ts}^{sup}$		
				(T)	(S)	(O)	(T)	(S)	(O)	(T)	(S)	(O)
4	90°	0.15	1248	0.0001	0.1439	0.2742	0.8748	0.8700	0.8292	87	83	75
3	75°	0.075	2108	0.0001	0.4226	0.6618	0.9270	0.9239	0.9339	97	96	99
2	60°	0	3320	0.0013	0.6098	0.7516	0.9517	0.9509	0.9555	97	98	100

No geral, observa-se que o algoritmo AFS obteve bom desempenho em todos os exemplos analisados neste trabalho, gerado malhas de qualidade comparáveis as obtidas quando se considera a otimização das métricas de qualidade. Também pode-se observar que, com o aumento do refinamento, a diferença nas respostas de qualidade entre tais abordagens diminuiu.

Também destaca-se o uso do algoritmo de subdivisão de contorno juntamente com AFS, que demonstrou capacidade de gerar malhar com diferentes resoluções e qualidades, uma característica importante tendo em vista o uso destes em diversas aplicações de engenharia.

Os resultados mostraram que a etapa de SAO é importante para evitar elementos inválidos também no caso de superfícies aparadas, além de ter a capacidade de aumentar a qualidade dos elementos nas proximidades das curvas de aparo.

## 5.2 Análises Numéricas

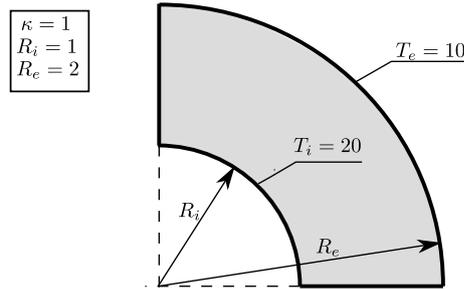
O desempenho numérico das malhas geradas pelos algoritmos apresentados neste trabalho é estudado em diversos exemplos de aplicação, onde são empregados tanto elementos triangulares, baseados em triângulos de Bézier, quanto quadriláteros, baseados em superfície de Bézier.

Problemas planos de transferência de calor e elasticidade linear são tratados nos Itens 5.2.1-5.2.3. São considerados problemas com solução analítica conhecida ou manufaturada<sup>151</sup>, de modo a verificar se as taxas de convergência teóricas são atingidas.

Problemas de análise de casca são descritos nos Itens 5.2.4-5.2.7, onde são realizados vários estudos de convergência em problemas *benchmarks*, sendo três problemas do conhecido *Shell Obstacle Course*<sup>159</sup>.

Por fim, problemas de análise de cascas envolvendo a geração em superfícies aparadas são discutidos nos Itens 5.2.8-5.2.10.

Figura 101 – Descrição do cilindro de parede espessa.



Fonte: Elaborada pelo autor.

### 5.2.1 Transferência de calor em cilindro de parede espessa

O primeiro problema consiste na transferência de calor no cilindro espesso mostrado Figura 101, utilizando o método das soluções manufaturadas. O campo de temperatura ( $T$ ) é dado por:

$$T = \frac{T_e \ln(r/R_i) - T_i \ln(r/R_e)}{\ln(R_e/R_i)}, \quad (5.2)$$

onde  $T_i$  e  $T_e$  são respectivamente as temperaturas interna e externa no cilindro,  $R_i$  e  $R_e$  são os raios interno e externo.

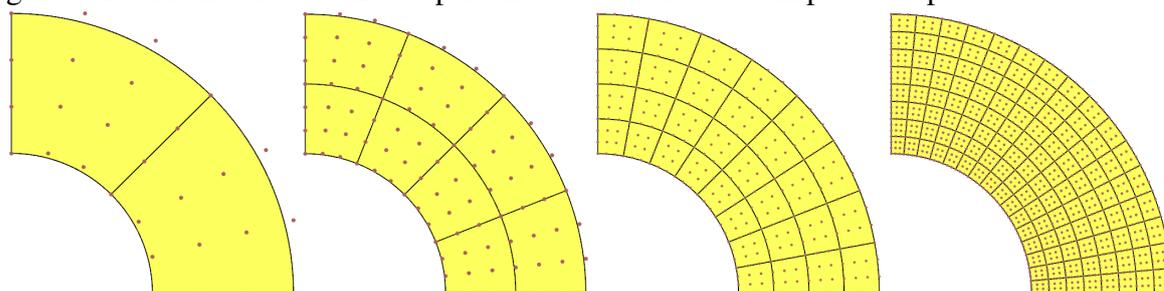
O problema é solucionado utilizando elementos triangulares e quadrilaterais. A malha inicial é obtida pelo algoritmo AFP, no caso triangular, e MTB, no caso quadrilateral. As demais malhas são obtidas através de refinamentos uniforme. Elementos quadráticos (BSD2 e BTD2), cúbicos (BSD3 e BTD3), quárticos (BSD4 e BTD4) e quánticos (BSD5 e BTD5) são considerados. A Figura 102 ilustra as malhas obtidas para o caso cúbico.

A convergência da solução numérica é avaliada pela a norma  $L^2$  dos erros no campo de temperatura, dada por<sup>156</sup>:

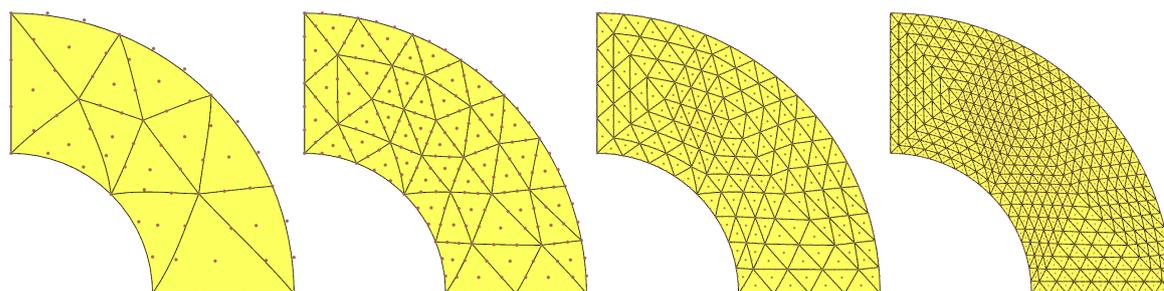
$$\|\mathbf{e}_T\|_{L_2} = \sqrt{\int_{\Omega} (T - \hat{T})^2 d\Omega} \quad (5.3)$$

onde  $T$  é a temperatura obtida no modelo numérico e  $\hat{T}$  é a temperatura exata dada pela Equação (5.2). Esta norma é computada numericamente utilizando uma quadratura de  $15 \times 15$  pontos de Gauss no caso de elementos quadrilaterais, e uma quadratura triangular de 48 pontos, capaz de integrar exatamente polinômios de até grau 15, no caso de elementos triangulares. Vale notar que neste exemplo não há geração interna de calor nem fluxos de calor prescritos no contorno, logo as condições de contorno do problema consistem apenas na aplicação das temperaturas prescritas nos arcos interno e externo do cilindro.

Figura 102 – Malhas 1-4 utilizadas para modelar o cilindro com parede espessa.



(a) Malhas quadrilaterais.

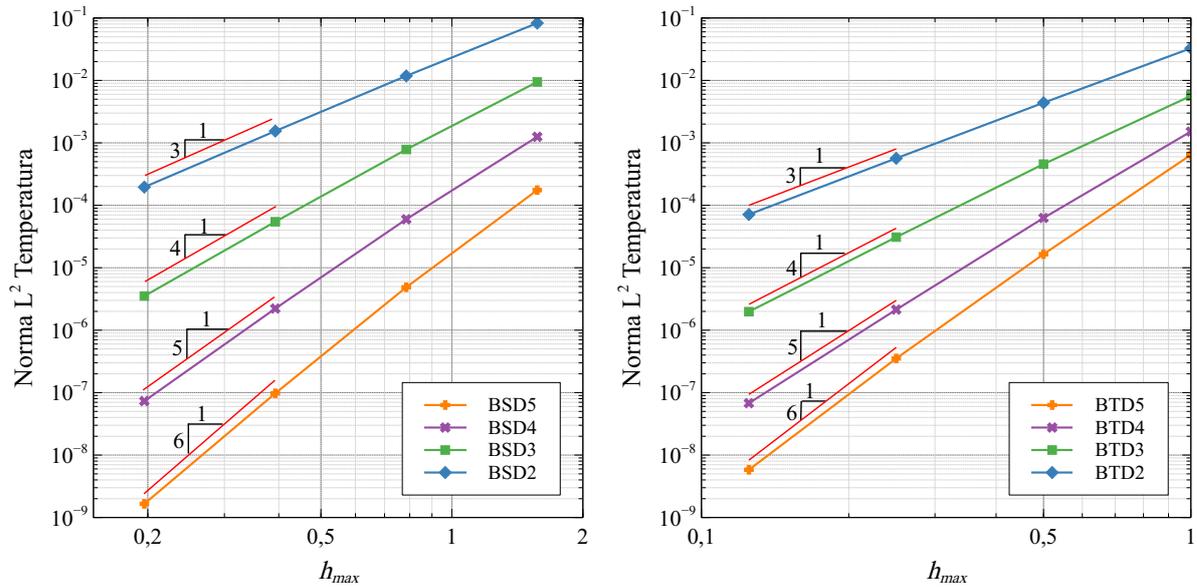


(b) Malhas triangulares.

Fonte: Elaborada pelo autor.

Os resultados são apresentados na Figura 103. Em teoria, as taxas de convergência da norma de erro para o campo de temperatura, em relação ao parâmetro de discretização da malha, tendem a  $p + 1$  com a discretização do modelo<sup>151</sup>, onde  $p$  é o grau do elemento utilizado. As taxas teóricas são obtidas a medida que se reduzem as distorções nos elementos, de modo que o Jacobiano destes tenda a ser constante. O parâmetro de discretização considerado neste trabalho é o tamanho máximo das arestas  $h_{max}$ . Os resultados mostram que taxas de convergência próximas as teóricas são obtidas em todos os casos.

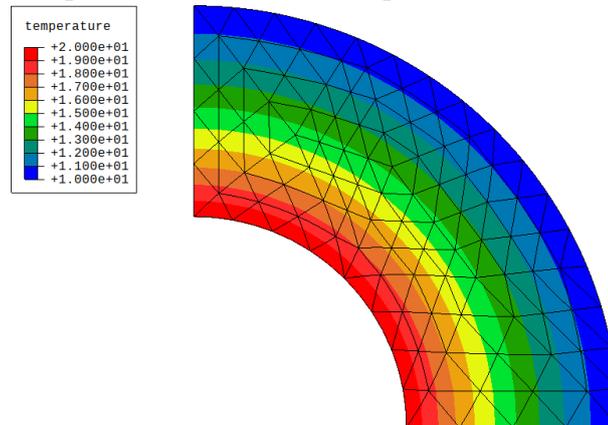
Por fim, é apresentado na Figura 104 o campo de temperatura obtido. Pode-se notar que a solução em questão é relativamente simples, justificando a rápida convergência para solução analítica já nas primeiras malhas utilizadas.

Figura 103 – Norma  $L^2$  do campo de temperatura obtida no cilindro espesso.

(a) Elementos quadrilaterais.

(b) Elementos triangulares.

Fonte: Elaborada pelo autor.

Figura 104 – Campo de temperatura  $T$  no cilindro espesso.

Fonte: Elaborada pelo autor.

### 5.2.2 Chapa infinita tracionada com furo circular

O segundo exemplo trata de um problema de uma chapa infinita com furo circular submetida à uma carga uniforme, conforme ilustrado na Figura 105. O campo de tensões na vizinhança do furo é determinado analiticamente usando coordenadas polares, sendo dado por<sup>2</sup>:

$$\sigma_{rr} = \frac{T_x}{2} \left( 1 - \frac{R^2}{r^2} \right) + \frac{T_x}{2} \left( 1 - 4 \frac{R^2}{r^2} + 3 \frac{R^4}{r^4} \right) \cos 2\theta, \quad (5.4)$$

$$\sigma_{\theta\theta} = \frac{T_x}{2} \left( 1 + \frac{R^2}{r^2} \right) - \frac{T_x}{2} \left( 1 + 3 \frac{R^4}{r^4} \right) \cos 2\theta, \quad (5.5)$$

$$\sigma_{r\theta} = -\frac{T_x}{2} \left( 1 + 2 \frac{R^2}{r^2} - 3 \frac{R^4}{r^4} \right) \sin 2\theta, \quad (5.6)$$

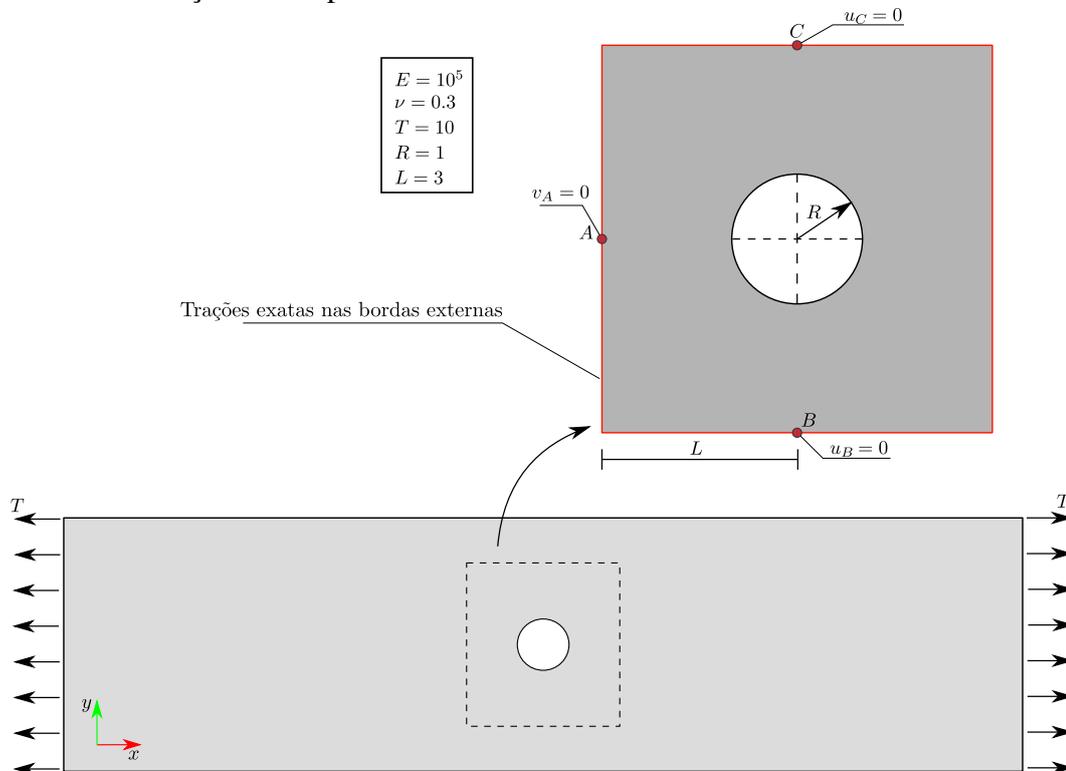
onde  $T_x$  é magnitude da carga e  $R$  o raio do furo circular. As tensões ( $\sigma_{rr}$ ,  $\sigma_{\theta\theta}$  e  $\sigma_{r\theta}$ ) são definidas no sistema de coordenadas polares e podem ser convertidas para o sistema cartesiano por:

$$\begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix} = \mathbf{T}^T \begin{bmatrix} \sigma_{rr} & \sigma_{r\theta} \\ \sigma_{r\theta} & \sigma_{\theta\theta} \end{bmatrix} \mathbf{T}; \quad \mathbf{T} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}, \quad (5.7)$$

onde

$$\tan \theta = \frac{y}{x} \quad \text{e} \quad r^2 = x^2 + y^2. \quad (5.8)$$

Figura 105 – Descrição do chapa infinita com furo circular.



Fonte: Elaborada pelo autor.

Uma região quadrada em torno do furo é considerada para solução numérica do problema. Nesta região, são aplicadas as forças distribuídas (*tractions*) exatas na borda externa

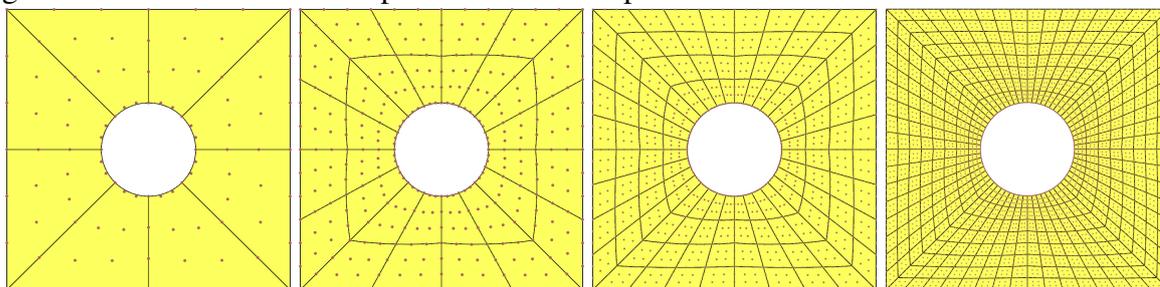
do modelo. Tais forças são computadas em uma posição arbitrária  $(x_i, y_i)$  por:

$$\mathbf{f} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix} \mathbf{n}, \quad (5.9)$$

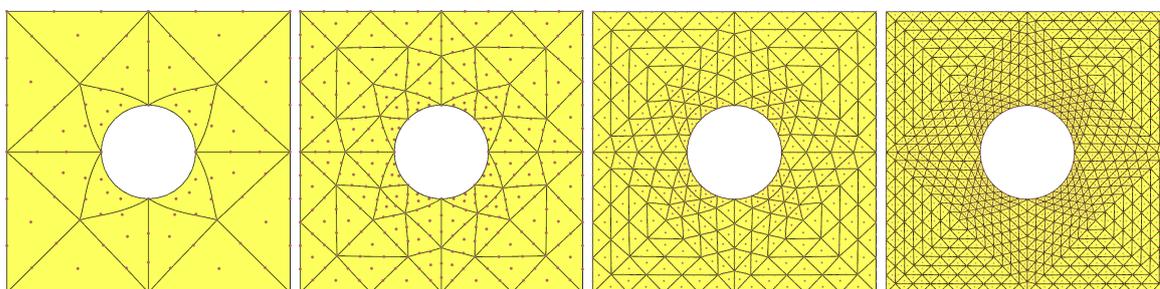
onde  $\mathbf{f}$  é o valor das forças e  $\mathbf{n}$  é o vetor normal à superfície de aplicação da força, na posição considerada. Além das cargas, são aplicadas restrições de deslocamento em três pontos do modelo para evitar deslocamento de corpo rígido.

Também são considerados elementos triangulares e quadrilaterais neste exemplo. As malhas são geradas da mesma forma que no exemplo anterior, e são apresentadas na Figura 106. Vale destacar que não foi possível a aplicação direta do algoritmo MTB, para geração dos elementos quadrilaterais. Como o modelo apresenta um furo, o mesmo teve de ser particionado manualmente em quatro regiões para aplicação da técnica, como ilustrado na Figura 107. Em exemplos de geometrias mais complexas, como no caso dos rotores e violão apresentados no Item 5.1.2, seria impraticável realizar tal operação manualmente, evidenciando as limitações da técnica MTB.

Figura 106 – Malhas utilizadas para modelar a chapa infinita com furo.



(a) Malhas quadrilaterais.



(b) Malhas triangulares.

Fonte: Elaborada pelo autor.

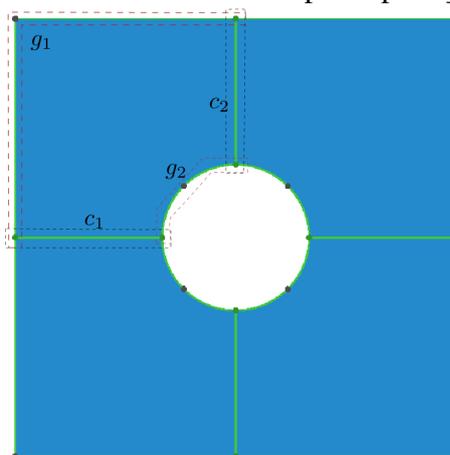
Neste exemplo, a norma  $L^2$  dos erros no campo de tensões é utilizada para avaliar a

convergência da solução. Esta norma é dada por<sup>156</sup>:

$$\|\mathbf{e}_\sigma\|_{L_2} = \sqrt{\int_{\Omega} (\boldsymbol{\sigma} - \hat{\boldsymbol{\sigma}})^T (\boldsymbol{\sigma} - \hat{\boldsymbol{\sigma}}) d\Omega} \quad (5.10)$$

onde  $\boldsymbol{\sigma}$  é o vetor de tensões obtido no modelo numérico e  $\hat{\boldsymbol{\sigma}}$  é o vetor de tensões exatas avaliados pela Equação (5.7). A norma é computada numericamente seguindo o mesmo esquema apresentado no problema térmico.

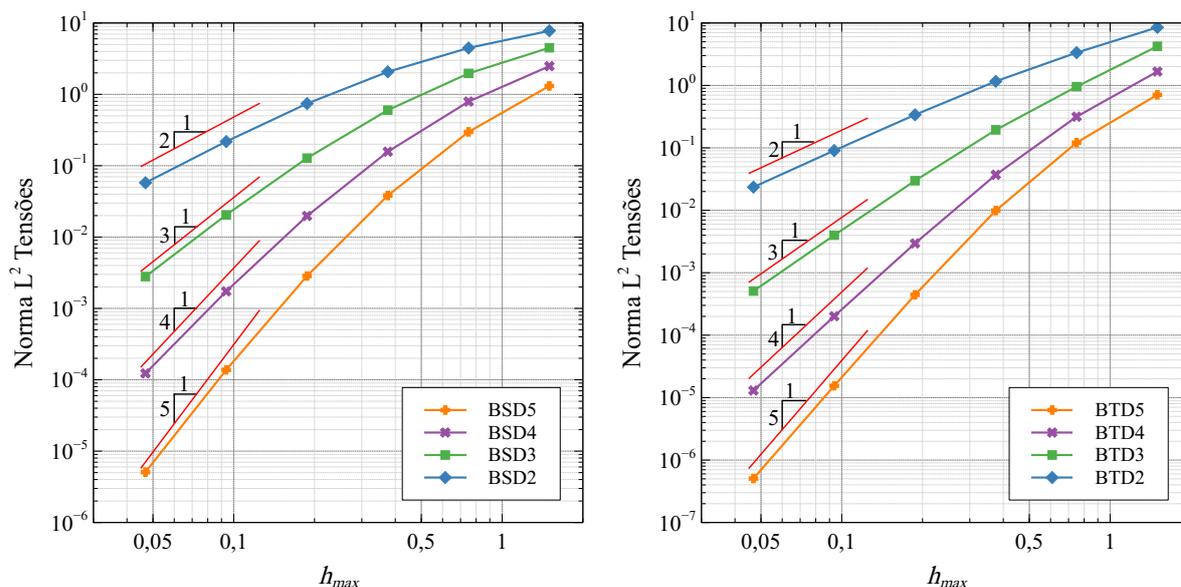
Figura 107 – Particionamento do domínio necessário para aplicação do mapeamento transfinito.



Fonte: Elaborada pelo autor.

Os resultados são apresentados na Figura 108. Neste caso, as taxas de convergência da norma de erro para o campo de tensões em relação a  $h_{max}$  tendem a  $p$  com a discretização da malha<sup>156</sup>. Os resultados numéricos mostram que as taxas de convergência teóricas são obtidas para os modelos mais refinados.

Figura 108 – Norma  $L^2$  do campo de tensões da chapa infinita com furo.



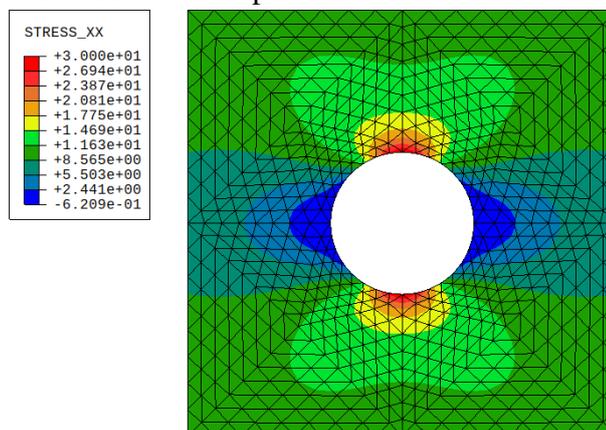
(a) Elementos quadrilaterais.

(b) Elementos triangulares.

Fonte: Elaborada pelo autor.

A resposta do campo de tensões na direção  $x$  é apresentada na Figura 109. Vale ressaltar que este exemplo apresenta maior complexidade em comparação ao exemplo anterior.

Figura 109 – Campo das tensões- $xx$  da chapa infinita com furo.



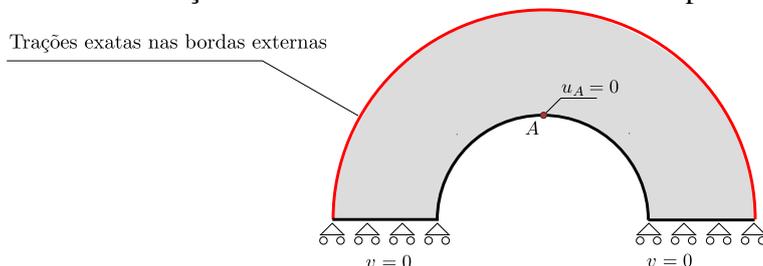
Fonte: Elaborada pelo autor.

### 5.2.3 Efeito da suavização dos pesos

Nesta seção é discutido o efeito da suavização dos pesos do algoritmo AFP em respostas numéricas que utilizem as malhas suavizadas. O problema estudado é o mesmo da chapa infinita com furo circular. Entretanto, neste caso a região adotada é a geometria considerada na Seção 3.4, onde o efeito da suavização dos pesos foi avaliado pelas métricas de qualidade

baseadas na geometria do elemento. A Figura 110 ilustra as condições de contorno consideradas no problema. Neste caso, a simetria do problema foi considerada na solução numérica.

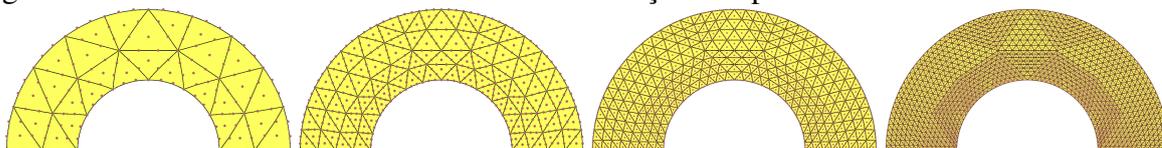
Figura 110 – Descrição do domínio de análise considerado para estudo da suavização dos pesos.



Fonte: Elaborada pelo autor.

São utilizadas malhas cúbicas geradas pelo algoritmo AFP considerando e não considerando a etapa de suavização dos pesos. A etapa de suavização das coordenadas é aplicada em ambos os casos. Malhas mais discretizadas são obtidas via refinamento uniforme, como ilustrado na Figura 111.

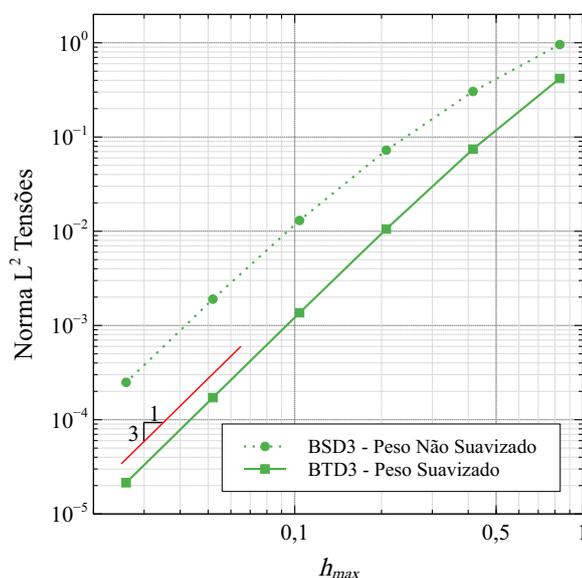
Figura 111 – Malhas utilizadas no estudo da suavização dos pesos.



Fonte: Elaborada pelo autor.

A norma  $L^2$  do campo de tensões das duas famílias de malhas é apresentada na Figura 112. Pode-se notar que a precisão das respostas com peso suavizado é maior, apresentando  $\|\mathbf{e}_\sigma\|_{L_2}$  em torno de 10 vezes menor em comparação com as respostas sem suavização do peso. Além disso, a taxa de convergência teórica (igual a  $p = 3$ ) é obtida com menor nível de refinamento quando os pesos são suavizados, em comparação quando os pesos não são suavizados.

Figura 112 – Norma  $L^2$  do campo de tensões obtidas de malhas com e sem a suavização de pesos.

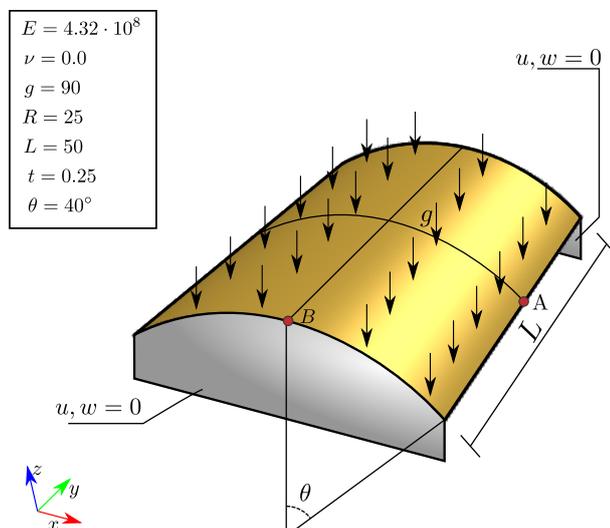


Fonte: Elaborada pelo autor.

#### 5.2.4 Casca de Scordelis-Lo

O primeiro problema de análise de cascas considerado neste trabalho trata de uma casca cilíndrica com restrição nos lados curvos por um diafragma rígido e submetida à carga de peso próprio, como ilustrado na Figura 113. Este problema é conhecido na literatura como Scordelis-Lo *roof*<sup>160</sup> e apresenta soluções utilizando diversas formulações de elementos finitos<sup>159, 152, 2</sup>. A simetria do problema é considerada, assim apenas um quarto da casca é modelada.

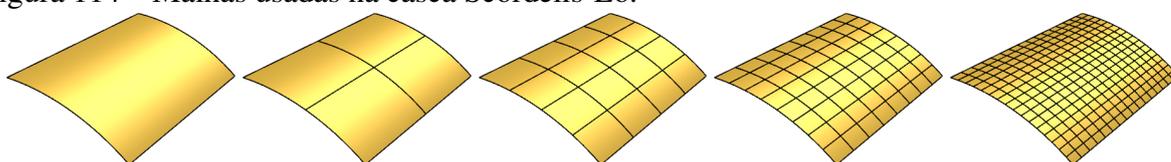
Figura 113 – Descrição da casca de Scordelis-Lo.



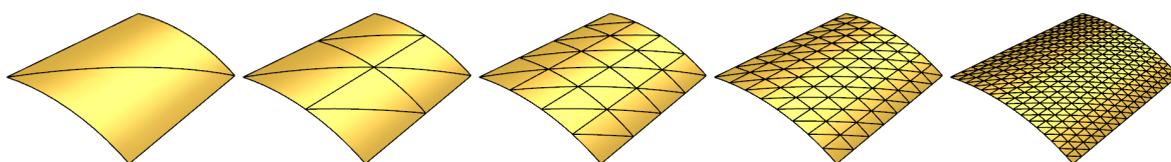
Fonte: Elaborada pelo autor.

O problema é solucionado utilizando elementos isogeométricos de superfície de Bézier racional e triângulos de Bézier racional. Também são utilizados elementos finitos isoparamétricos quadráticos: o elemento quadrilateral de oito nós Q8 e o elemento triangular de seis nós T6. Neste problema, as malhas são geradas utilizando o algoritmo MTB discutida na Seção 3.2. As malhas triangulares e quadrilaterais são apresentadas na Figura 114. Vale notar que a geometria desse exemplo pode ser representada por uma superfície quadrática racional em uma direção e uma reta na outra (e.g. Superfície de Coons ou Superfície de Bézier Racional). Assim, elementos triangulares de Bézier racional de grau cúbico ou superior apresentam geometria exata.

Figura 114 – Malhas usadas na casca Scordelis-Lo.



(a) Malhas quadrilaterais.



(b) Malhas triangulares.

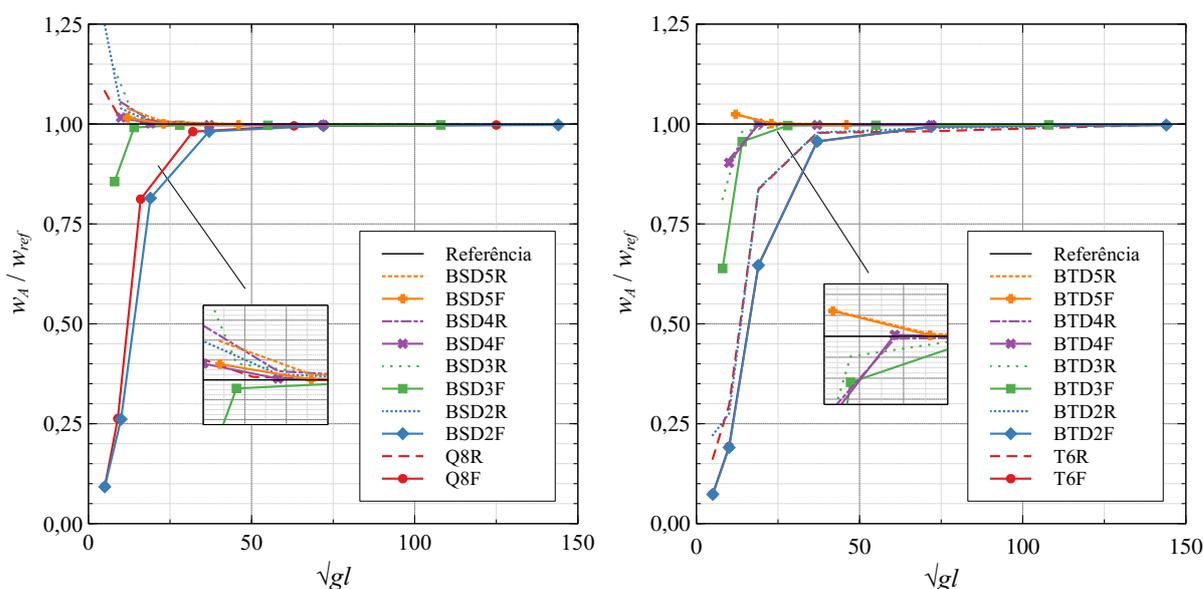
Fonte: Elaborada pelo autor.

Neste problema são considerados elementos com integração completa (com legenda *F*) e elementos com integração reduzida (com legenda *R*). Além de utilizar integração reduzida nos elementos quadrilaterais, também será testado um esquema reduzido para elementos triangulares, onde é subtraído em 1 o grau do polinômio integrado exatamente. Por exemplo, no caso de elementos cúbicos, para a integração completa é utilizado uma quadratura de 12 pontos, capaz de integrar exatamente polinômios de até sexto grau, enquanto a integração reduzida considera 7 pontos, capaz de integrar exatamente polinômios de até quinto grau. Tal esquema de integração é considerado para avaliar uma possível redução do travamento do elemento de casca implementado neste trabalho. Vale ressaltar que na espessura (direção *t*) são considerados 2 pontos de gauss em todos os casos.

Os resultados de deslocamento *w* no ponto A e a raiz quadrada do número de graus de liberdade ( $\sqrt{gI}$ ) obtidos nos diversos modelos são apresentados na Figura 115. Os resultados obtidos são comparados ao deslocamento de referência<sup>159</sup>  $w_{ref} = 0.3024$ . Todos os elementos

convergem rapidamente para solução de referência. Os elementos quadrilaterais com integração reduzida apresentaram maior precisão quando comparados aos elementos de integração completa, e também apresentaram menor rigidez em comparação a solução de referência, convergindo por cima. Por outro lado, as respostas com integração completa apresentam maior rigidez e convergem por baixo, exceto para as malhas quárticas e quánticas, que convergem por cima e bem próximas da solução de referência mesmo nos modelos com pouco refinamento.

Figura 115 – Convergência do deslocamento  $w_A$  para casca de Scordelis-Lo.



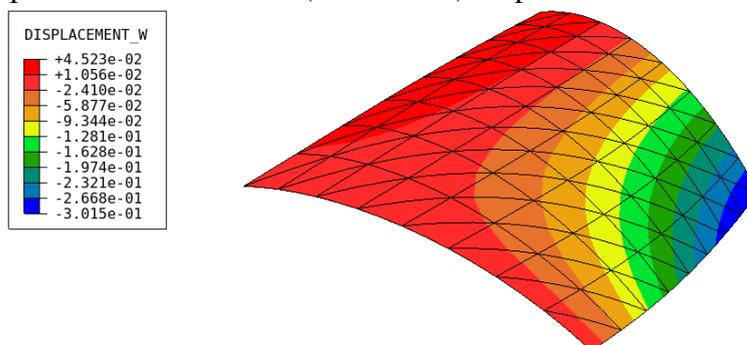
Fonte: Elaborada pelo autor.

No caso das malhas triangulares, não há diferença considerável na precisão dos elementos com integração completa ou reduzida. Todos os modelos convergem por baixo, exceto o modelo quántico. Neste exemplo se verifica travamentos apenas nos elementos quádráticos nos primeiros níveis de refinamento. Por se tratar de um problema com geometria simples, curvatura constante e carregamento distribuído, os campos de deslocamentos e esforços são bem comportados, facilitando sua aproximação pela solução de elementos finitos. O campo de deslocamentos  $w$  do problema é apresentado na Figura 116.

A mesma estrutura foi analisada para o problema de vibração livre por Li e colaboradores<sup>161</sup> utilizando elementos de contorno (SBFEM) e considerando  $\rho = 1$ . Neste caso a simetria não é adotada por haver modos de vibração não simétricos, logo o modelo completo é utilizado. Além disto, o deslocamento de corpo rígido na direção  $y$  é evitado restringindo  $v$  no ponto B. Uma malha cúbica de triângulos de Bézier com  $32 \times 32$  elementos, equivalente à

malha 4 ( $16 \times 16$  elementos) utilizada anteriormente no modelo com simetria, é considerada.

Figura 116 – Campo de deslocamento  $w$  (escala  $20\times$ ) do problema Scordelis-Lo *roof*.



Fonte: Elaborada pelo autor.

O algoritmo de iteração em subespaços implementado neste trabalho foi utilizado com tolerância de  $10^{-7}$ . Além do modelo isogeométrico e da resposta da referência mencionada, o problema também é solucionado utilizando *software* ABAQUS CAE, com um modelo de  $256 \times 256$  elementos finitos quadriláteros de oito nós do tipo S8R. As 10 primeiras frequências de vibração obtidas são apresentadas na Tabela 19, onde também são apresentadas as diferenças percentuais obtidas entre o modelo IGA deste trabalho e os resultados obtidos pelo ABAQUS e o utilizando elementos de contorno (SBFEM).

Tabela 19 – Frequências de vibração da casca Scordelis-Lo *roof*.

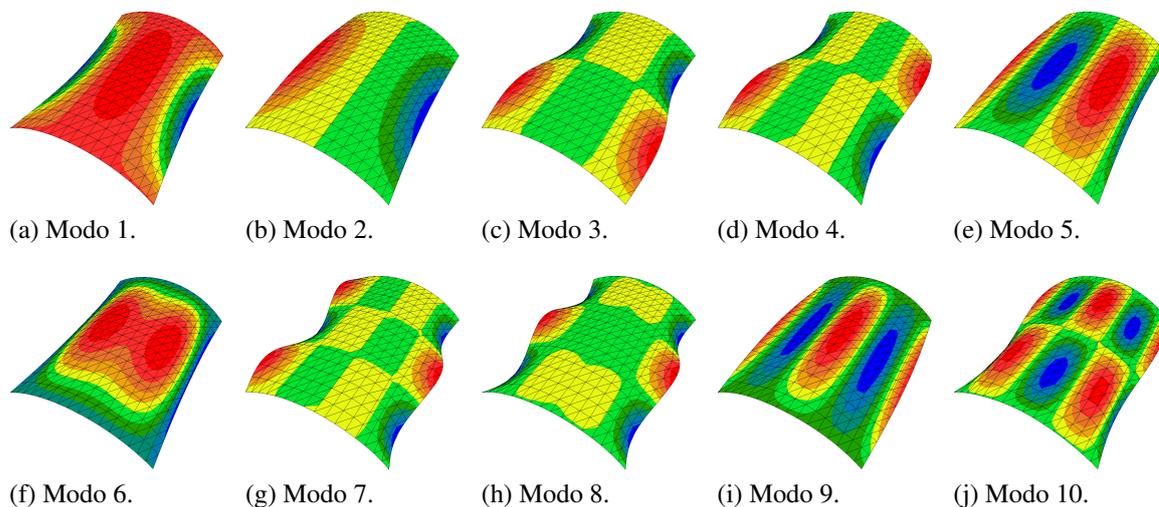
Frequência	BTD3F	S8R	Diferença (%)	SBFEM <sup>161</sup>	Diferença (%)
1	5.9475	5.9417	0.10	5.9453	-0.04
2	6.3703	6.3665	0.06	6.3651	-0.08
3	13.7674	13.7500	0.13	15.1850	9.34
4	14.2141	14.1930	0.15	16.2610	12.59
5	15.1757	15.1720	0.02	16.3950	7.44
6	22.5570	22.4820	0.33	22.8350	1.22
7	22.9661	22.9240	0.18	22.9450	-0.09
8	23.3639	23.3230	0.18	23.3430	-0.09
9	25.1003	25.0480	0.21	25.2750	0.69
10	32.2957	32.1280	0.52	34.7630	7.10

Fonte: Elaborada pelo autor.

Nota-se excelente concordância entre as respostas deste trabalho com as obtidas pelo ABAQUS, com diferença percentual de no máximo 0.52%. Por outro lado, diferenças consideráveis são observadas em relação ao SBFEM nos modos 3, 4, 5 e 10. As diferenças observadas podem estar relacionadas as condições de contorno consideradas, visto que não fica claro, na referência do SBFEM, como o deslocamento de corpo rígido foi evitado. Por outro

lado, as condições de contorno usadas no ABAQUS são idênticas às do FAST. Por fim, os modos de vibração são apresentados na Figura 117. Observa-se que vários modos de vibração não são simétricos.

Figura 117 – Modos de vibração da casca Scordelis-Lo *roof* (diagrama de cores para resposta  $w$ ).



Fonte: Elaborada pelo autor.

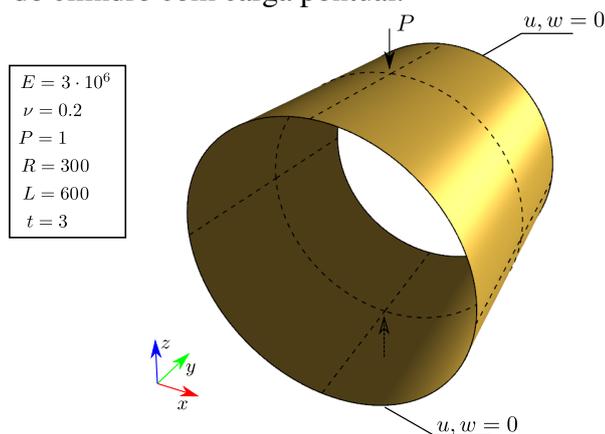
### 5.2.5 Cilindro submetido a carga concentrada

Este exemplo consiste em um cilindro fino apoiado nas extremidades por diafragma rígido e submetido a duas cargas concentradas, conforme ilustrado na Figura 118. O problema também faz parte do *Shell Obstacle Course* com várias soluções disponíveis na literatura usando diversas formulações<sup>159, 152, 2</sup>. A simetria do problema é considerada, assim apenas um oitavo do cilindro é modelado.

Novamente, malhas isogeométricas e de elementos finitos são obtidas pelo algoritmo MTB. A Figura 119 apresenta as malhas quadrilaterais e triangulares obtidas. Por se tratar de uma casca cilíndrica semelhante ao exemplo anterior, elementos triangulares cúbicos de Bézier racional representam a geometria de forma exata.

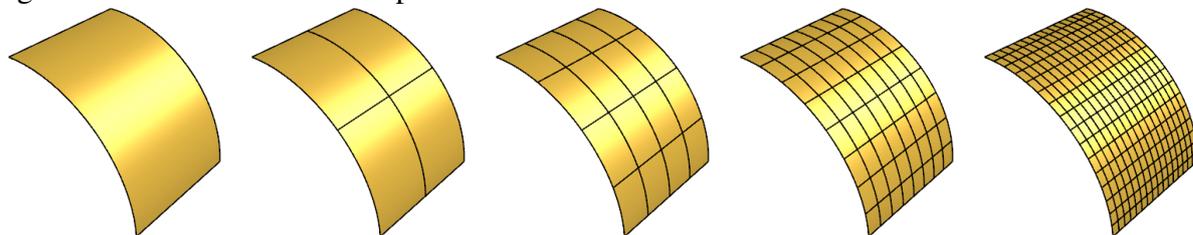
Os resultados de convergência do deslocamento  $w$  no ponto de aplicação da carga são apresentados na Figura 120. Os deslocamentos obtidos são comparados ao deslocamento de referência<sup>162</sup>  $w_{ref} = 1.825 \times 10^{-5}$ . Diferente do exemplo anterior, todos os elementos utilizados apresentaram travamento nos primeiros níveis de refinamento. Nos elementos quadrilaterais, a integração reduzida melhorou consideravelmente a precisão da resposta para todos os graus.

Figura 118 – Descrição do cilindro com carga pontual.

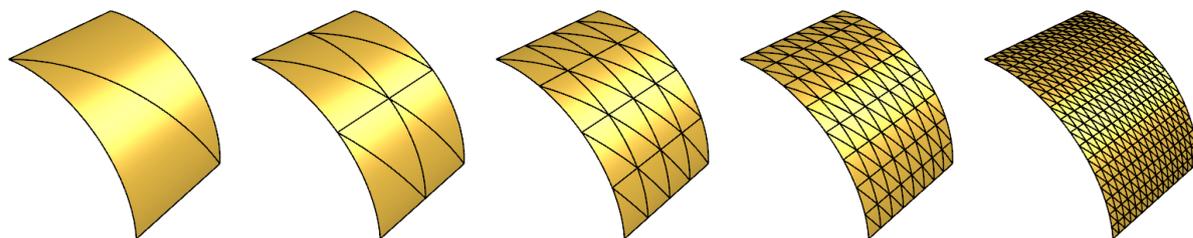


Fonte: Elaborada pelo autor.

Figura 119 – Malhas utilizadas para modelar o cilindro.



(a) Malhas quadrilaterais.



(b) Malhas triangulares.

Fonte: Elaborada pelo autor.

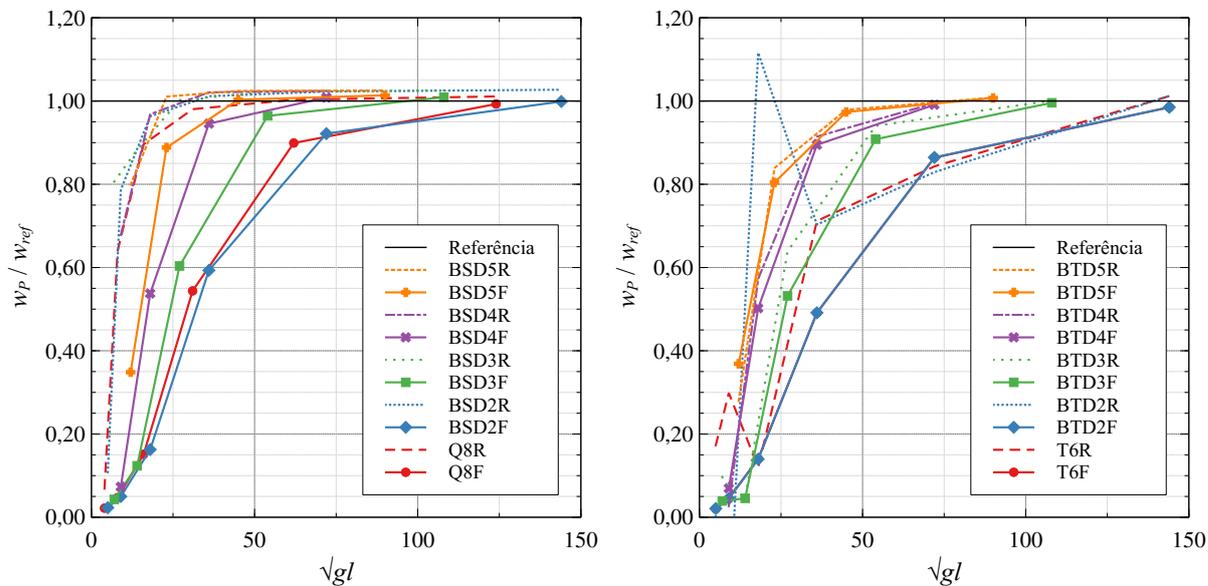
Também nota-se que todos os elementos convergiram por baixo.

No caso dos elementos triangulares, novamente não houve efeito benéfico com o uso da integração reduzida proposta. No grau quadrático, as respostas com integração reduzida apresentam oscilações nos primeiros níveis de refinamento. Nos demais graus, as respostas com integração tem precisão semelhante ao caso de integração total. Portanto, não se mostrou efetivo a utilização do esquema de integração reduzida para aliviar o travamento em elementos triangulares. Logo, nos demais exemplos desse trabalho será utilizado apenas o esquema de integração completa para estes elementos.

É importante notar que devido ao carregamento ser concentrado, este problema apresenta campos de deslocamentos e esforços mais complexos do que os do exemplo anterior.

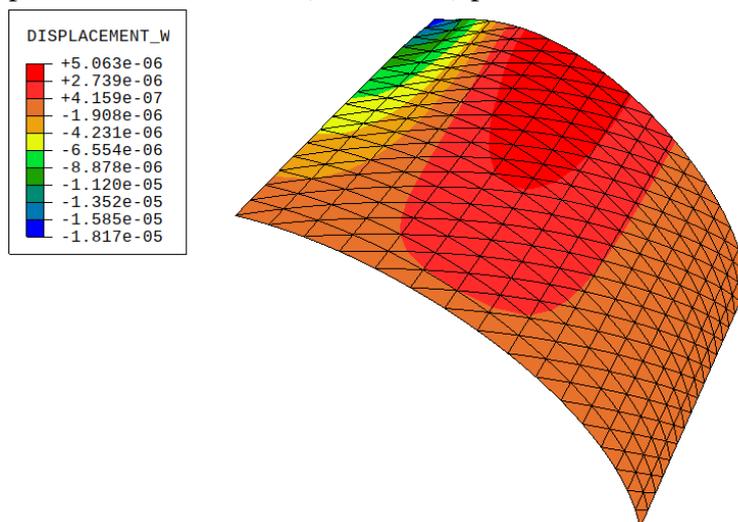
Isto explica também o travamento mais severo observado neste caso. Além disto, foi necessário um número maior de graus de liberdade para obter a convergência da resposta analisada. O campo de deslocamento  $w$  do problema é apresentado na Figura 121.

Figura 120 – Convergência do deslocamento  $w_P$  para o cilindro com carga pontual.



Fonte: Elaborada pelo autor.

Figura 121 – Campo de deslocamento  $w$  (fator =  $1e6$ ) para o cilindro com carga pontual.

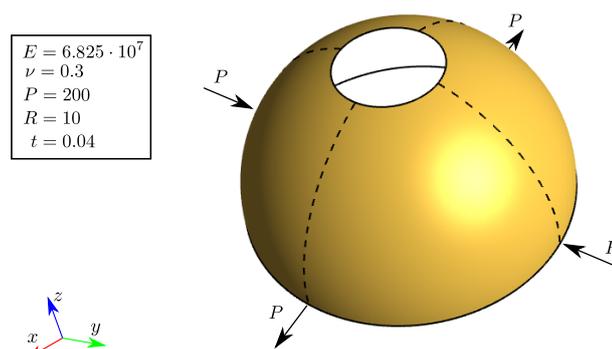


Fonte: Elaborada pelo autor.

### 5.2.6 Hemisfério submetido a carga concentrada

Este problema trata de um hemisfério com corte no polo de  $18^\circ$  e submetido a dois pares de cargas localizadas em pontos antipodais no equador. As cargas têm mesmo módulo e, em cada par, sentidos opostos, como ilustrado na Figura 122. A simetria do problema é considerada, logo apenas um quarto da geometria é modelada, resultando na mesma geometria estudada anteriormente no Item 5.1.5. Este exemplo é uma variação do problema do hemisfério do *Shell Obstacle Course*<sup>159</sup>, onde não há furo. Neste versão modificada do problema, o furo é adicionado para possibilitar a representação da geometria por uma malha estruturada de elementos quadrilaterais, o que não é possível quando considerado a região do polo, onde são utilizados elementos triangulares<sup>163, 152</sup>.

Figura 122 – Descrição do hemisfério com furo.



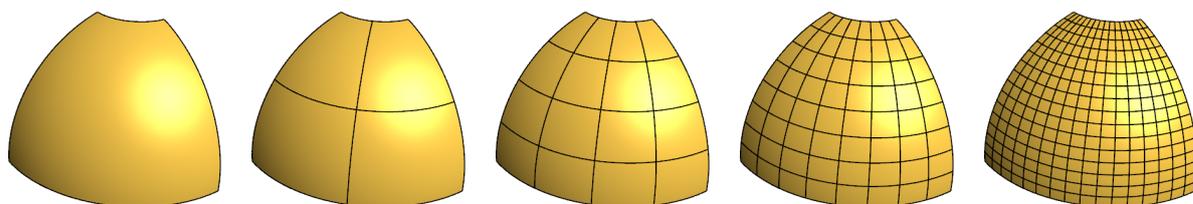
Fonte: Elaborada pelo autor.

Mais uma vez o algoritmo MTB é utilizado para geração das malhas isogeométricas e de elementos finitos. É importante notar que neste caso é utilizada a versão adaptada do algoritmo para a geração de malhas em superfícies paramétricas, discutida no Seção 3.5.1, dado que neste exemplo a geometria do problema não pode ser representada por uma superfície de Coons. A Figura 123 ilustra as malhas utilizadas. Neste exemplo, a superfície NURBS que representa a geometria é biquadrática, logo é necessário utilizar triângulos de Bézier quárticos para garantir a representação exata.

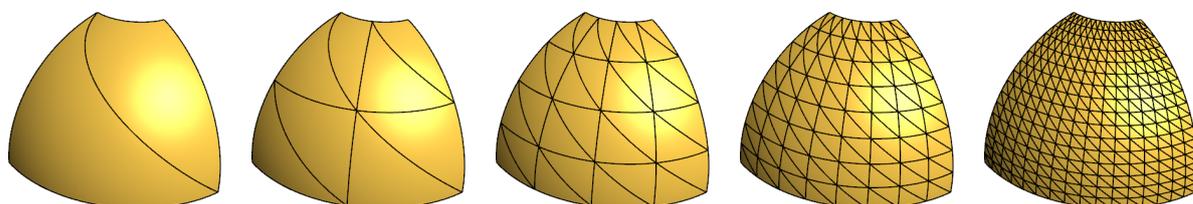
A Figura 124 mostra o resultado da convergência do deslocamento  $u$  no ponto A. O deslocamento de referência<sup>163</sup> é  $u_{ref} = 0.0940$ . Nos elementos quadrilaterais, houve travamento principalmente no elemento Q8, que mesmo utilizado o esquema de integração reduzida não obteve boa precisão para o número de graus de liberdade considerados aqui. Nos elementos isogeométricos, o travamento é reduzido com o aumento do grau do elemento utilizado. A

convergência ocorre por baixo em todos os casos com integração completa e também no elemento Q8 com integração reduzida. O elemento isogeométrico quadrático teve desempenho superior em comparação ao Q8 por possui um ponto de controle a mais e, portanto, ser menos afetado pelo travamento.

Figura 123 – Malhas utilizadas para modelar o hemisfério com furo.



(a) Malhas quadrilaterais.

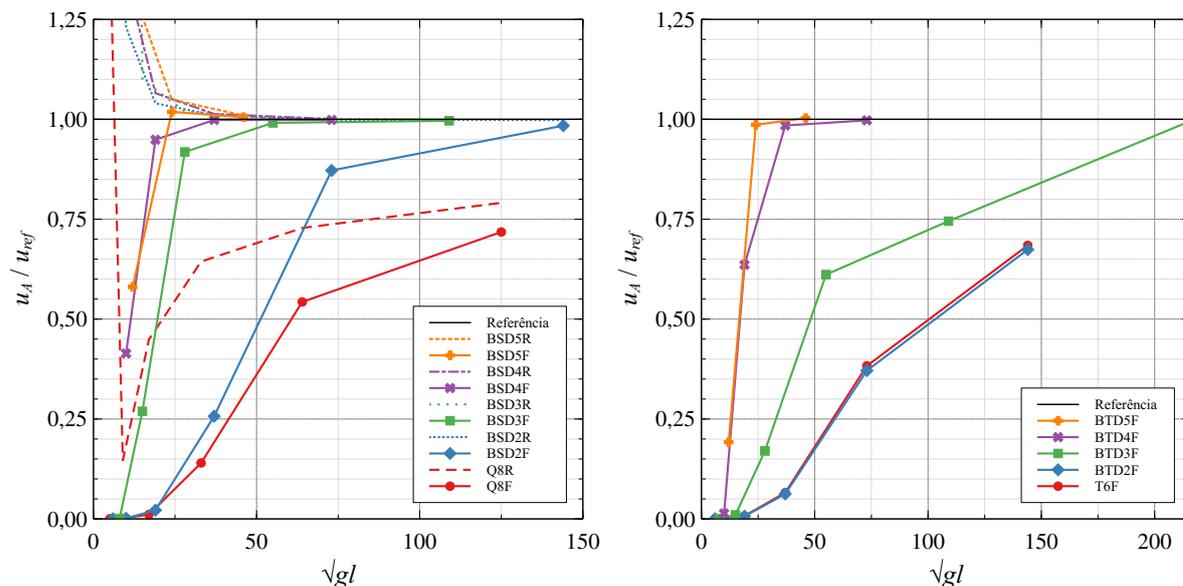


(b) Malhas triangulares.

Fonte: Elaborada pelo autor.

Nos elementos triangulares, há travamento nos elementos quadráticos e cúbicos. Já os elementos quárticos e quánticos convergem rapidamente para resposta de referência. Vale destacar que apesar do travamento, o elemento cúbico obteve boa precisão na última malha considerada.

Figura 124 – Convergência do deslocamento  $u_A$  para o hemisfério com furo.



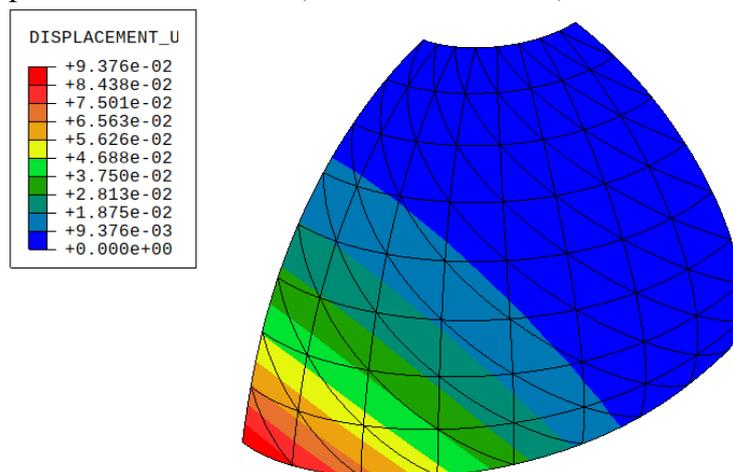
(a) Elementos quadrilaterais.

(b) Elementos triangulares.

Fonte: Elaborada pelo autor.

Os campos de deslocamentos e esforços apresenta grande complexidade neste exemplo devido às cargas concentradas e a geometria com dupla curvatura. Além disto, por se tratar de uma casca fina, há suscetibilidade à ocorrência de travamentos. O uso de elementos com grau mais elevado mostrou-se importante para melhorar a convergência da resposta obtida. O campo de deslocamento  $u$  é apresentado na Figura 125.

Figura 125 – Campo de deslocamento  $u$  (fator de escala = 30) no hemisfério com furo.



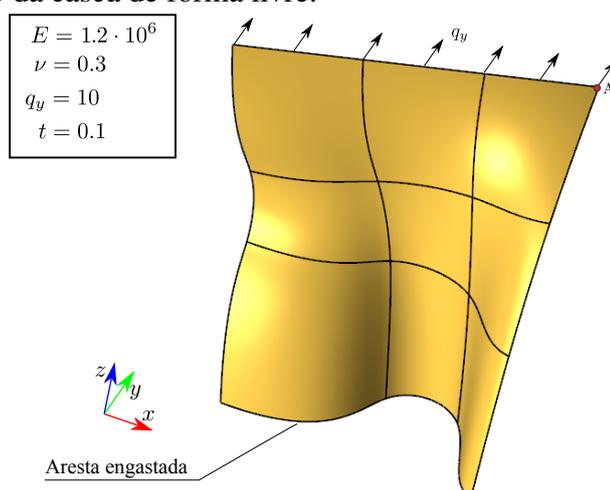
Fonte: Elaborada pelo autor.

### 5.2.7 Casca de forma livre

Este problema trata de uma casca de forma livre com um lado apoiado e outro submetido a um carregamento distribuído<sup>153, 164</sup>. Este exemplo foi proposto por Dornisch e colaboradores<sup>153</sup> para avaliar a importância da consideração de vetores normais exatos na formulação de cascas degeneradas. A descrição do exemplo é apresentado na Figura 126.

Neste problema, para os elementos quadrilaterais, são consideradas malhas estruturadas obtidas pelo algoritmo MTB, considerando subdivisão uniforme aplicada aos *knot spans* da geometria base da superfície. No caso dos elementos triangulares, são adotadas malhas não estruturadas geradas pelo algoritmo AFS, considerando os parâmetros de discretização do contorno apresentados na Tabela 20. As malhas obtidas são ilustradas na Figura 127.

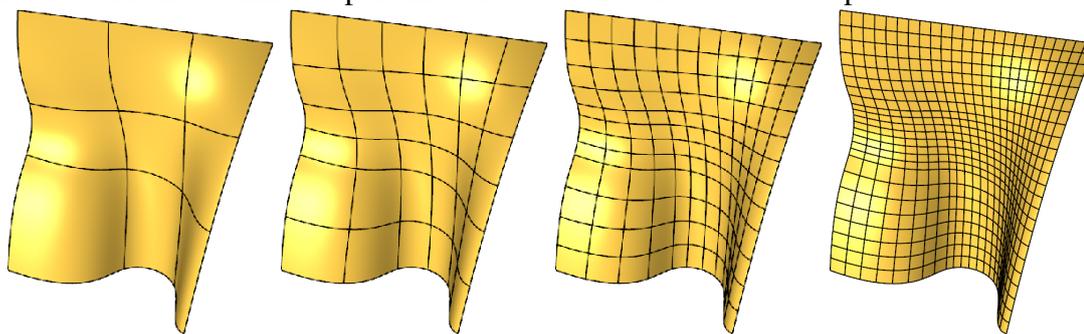
Figura 126 – Descrição da casca de forma livre.



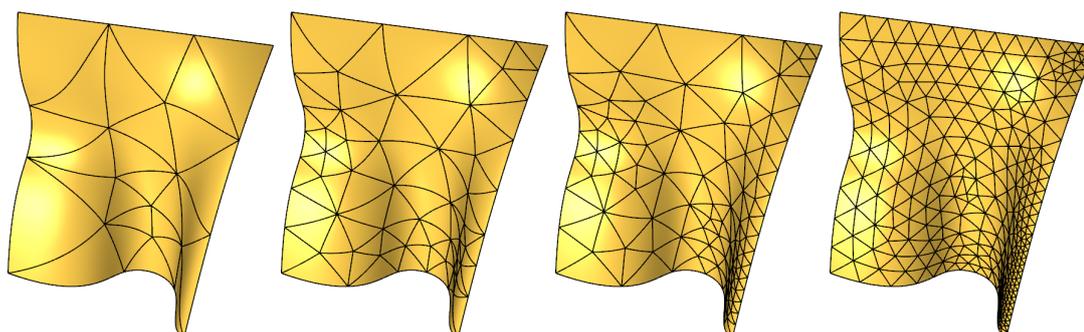
Fonte: Elaborada pelo autor.

A convergência do deslocamento  $v$  no ponto A (localizado na quina superior direita da casca), para ambos os elementos triangulares e quadrilaterais, é apresentada na Figura 128. Os deslocamentos são normalizados pela solução de referência  $v_{ref} = 1.02786$ , mencionada anteriormente<sup>153</sup>. Nos elementos quadrilaterais, todas as alternativas convergiram para resposta referência. A integração reduzida melhorou a precisão do elemento Q8, que apresenta travamento considerável. Já os elementos isogeométricos convergiram rapidamente sem efeito da integração reduzida. Vale ressaltar que como a geometria da casca é cúbica, considera-se no mínimo o grau cúbico nas malhas da AIG. Neste problema o grau cúbico se mostra suficiente para convergência rápida da resposta.

Figura 127 – Malhas utilizadas para modelar a casca com curvatura dupla.



(a) Malhas quadrilaterais.



(b) Malhas triangulares.

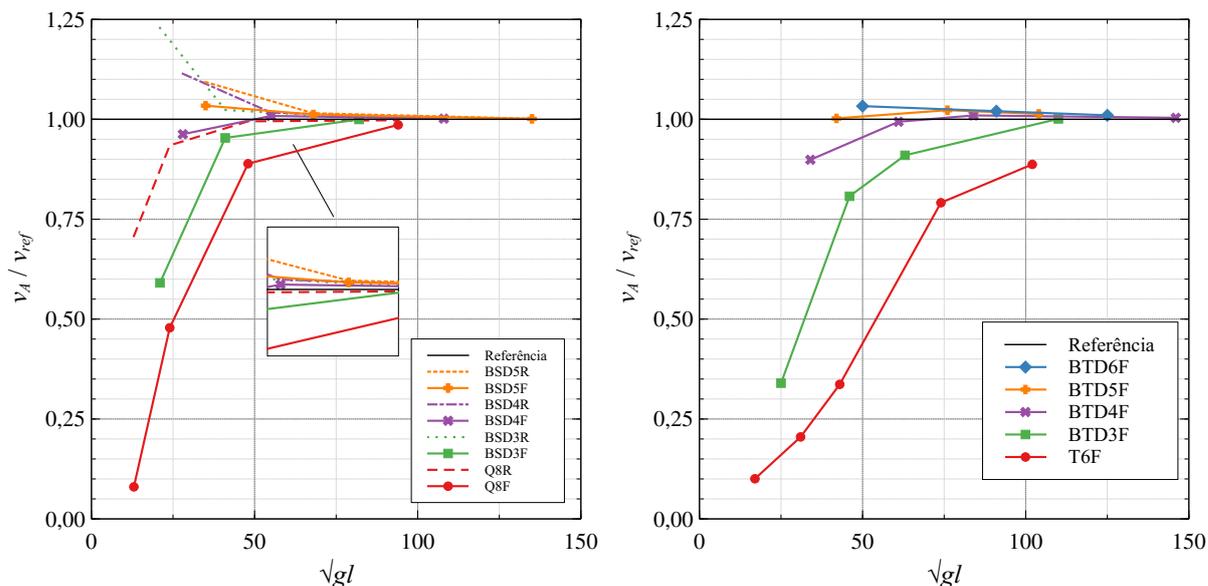
Fonte: Elaborada pelo autor.

Tabela 20 – Parâmetros de discretização das malhas da casca de forma livre.

Malha	$L_{max}$	$\theta_{max}$	$L_{min}$	$\hat{n}$
1	12	120	6	26
2	8	90	2	90
3	5	60	1	169
4	2	30	0	521
5	1.75	20	0	1008

Fonte: Elaborada pelo autor.

Figura 128 – Convergência do deslocamento  $v$  para a casca com curvatura dupla.



(a) Elementos quadriláterais.

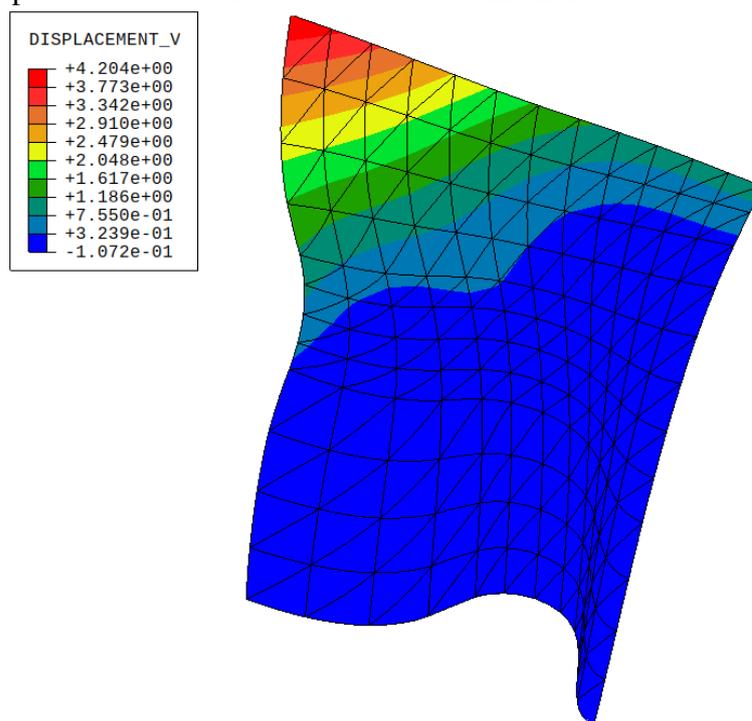
(b) Elementos triangulares.

Fonte: Elaborada pelo autor.

No caso dos elementos triangulares, o elemento finito T6 e o elemento de Bézier cúbico apresentam travamento maior, embora converjam para resposta com o refinamento do modelo. Os elementos de grau mais elevado convergem para resposta com pouco refinamento.

O campo de deslocamentos  $v$  do problema é apresentado na Figura 129. Os resultados obtidos mostram que a formulação desenvolvida é eficaz para solucionar problemas com considerável variação dos vetores normais. A consideração aproximada dos vetores normais, obtida via interpolação de valores definidos nos pontos de controle do modelo, mostra-se suficiente para convergência da resposta numérica do problema.

Figura 129 – Campo de deslocamento  $v$  da casca de forma livre.

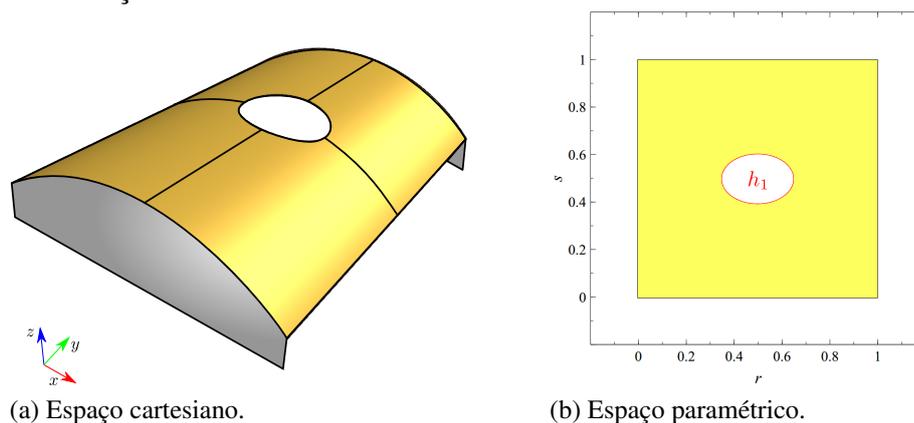


Fonte: Elaborada pelo autor.

### 5.2.8 Casca de Scordelis-Lo com furo

Este problema é um versão modificada da casca de Scordelis-Lo com um furo central. A descrição do furo, no espaço cartesiano e paramétrico, é apresentada na Figura 130.

Figura 130 – Descrição da casca de Scordelis-Lo com furo.

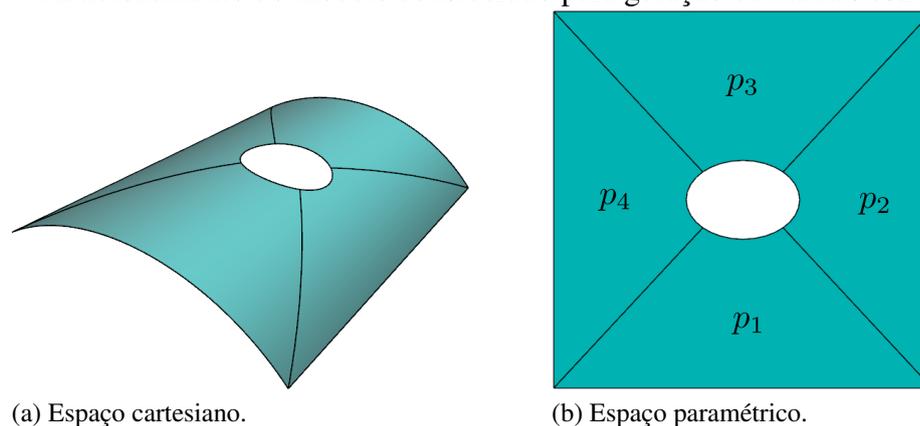


Fonte: Elaborada pelo autor.

Elementos triangulares e quadrilaterais são utilizados, onde as malhas são geradas, respectivamente, pelos algoritmos AFS e MTB. Para obtenção das malhas estruturadas, o modelo foi particionado em quatro sub-regiões, como ilustrado na Figura 131. A técnica MTB pôde ser

aplicada neste caso devido a simplicidade da geometria do problema. Nos problemas anteriores onde o MTB foi utilizado, as malhas geradas poderiam ser obtidas também pelos algoritmos padrão de subdivisão e elevação de grau de superfície de Bézier<sup>14</sup>. Neste problema, por outro lado, não seria possível obter as malhas dessa forma, evidenciando a motivação da técnica MTB apresentada neste trabalho.

Figura 131 – Particionamento do modelo considerado para geração de malhas estruturadas.



Fonte: Elaborada pelo autor.

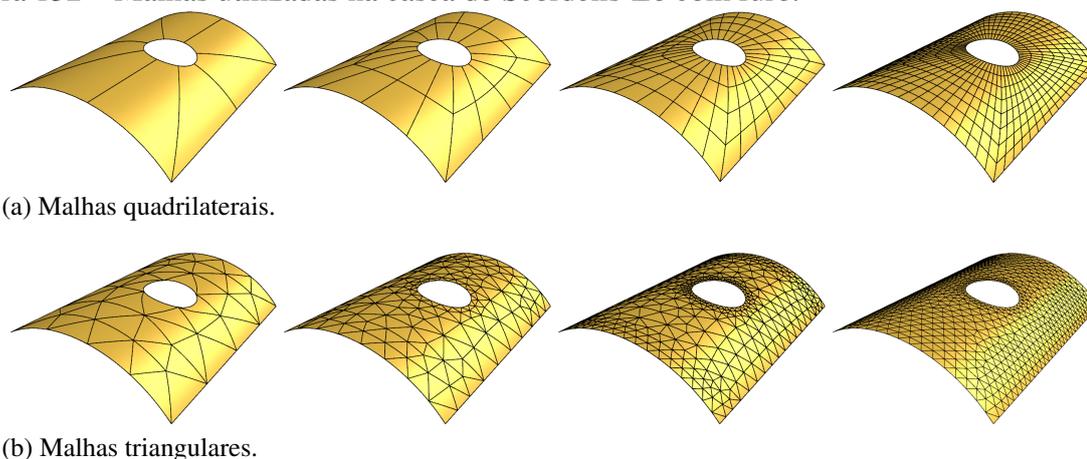
Inicialmente são considerados dois elementos em cada partição. As demais malhas são obtidas refinando uniformemente todas as arestas do modelo. No caso das malhas triangulares, o algoritmo AFS é utilizado com os parâmetros de discretização apresentados na Tabela 21. As malhas geradas são apresentadas na Figura 132. Apenas elementos isogeométricos de grau quadrático, cúbico e quártico são considerados no problema.

Tabela 21 – Parâmetros de discretização das malhas da casca de Scordelis-Lo com furo.

Malha	$L_{max}$	$\theta_{max}$	$L_{min}$	$\hat{n}$
1	15	60	10	56
2	8	30	3	240
3	4	20	1	624
4	3	15	0	1336

Fonte: Elaborada pelo autor.

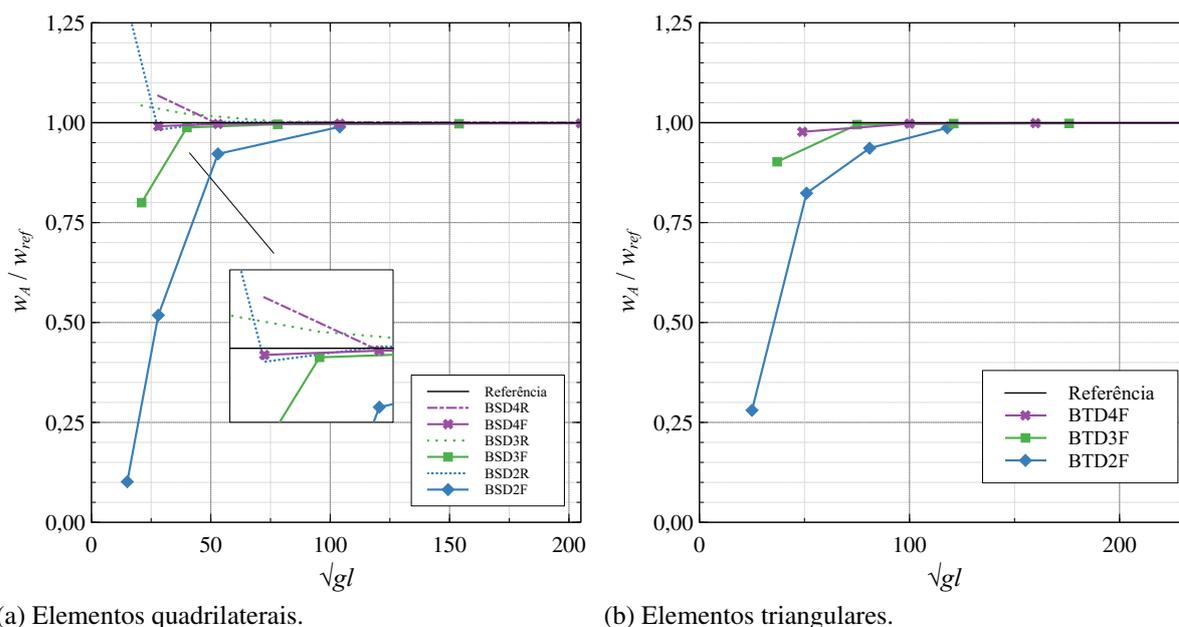
Figura 132 – Malhas utilizadas na casca de Scordelis-Lo com furo.



Fonte: Elaborada pelo autor.

A convergência do deslocamento  $w$  no ponto A é mostrada na Figura 133. O deslocamento de referência  $w_{ref} = -0.3560$  foi obtido pelo programa ABAQUS CAE, utilizando uma malha de 5188 elementos S3. Todos os elementos cúbicos e quárticos convergiram rapidamente para solução de referência. Os elementos quadráticos necessitaram de maior nível de refinamento para convergência da resposta, com exceção ao elemento quadrilateral com integração reduzida, que também convergiu rapidamente.

Figura 133 – Convergência do deslocamento  $w$  para a casca de Scordelis-Lo com furo.

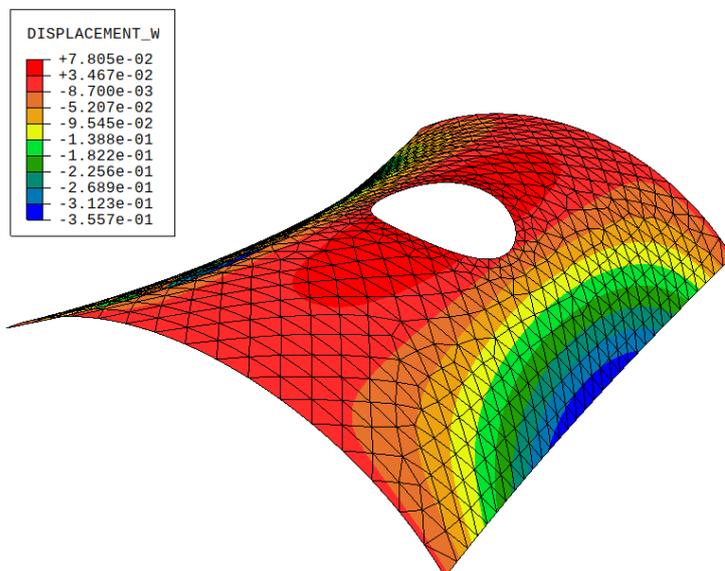


Fonte: Elaborada pelo autor.

A rápida convergência da resposta se deve a facilidade na aproximação do campo de

deslocamento do problema. Apesar do furo, a estrutura ainda apresenta respostas com variação suave. O campo de deslocamento  $w$  do problema é apresentado na Figura 134.

Figura 134 – Campo de deslocamento  $w$  (fator de escala = 8) da casca de forma livre.

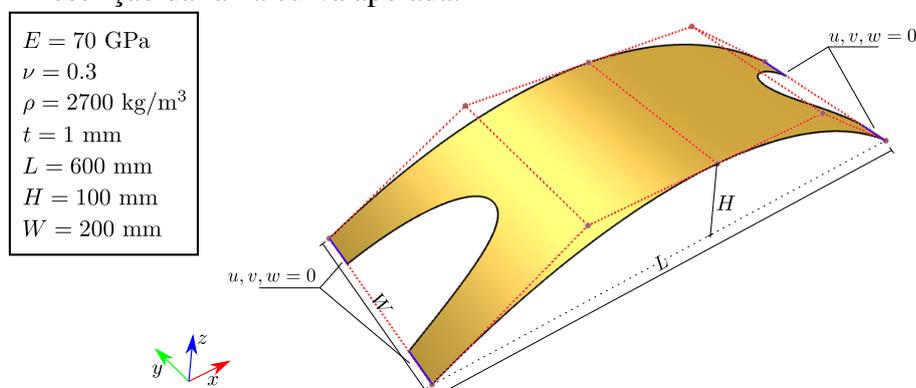


Fonte: Elaborada pelo autor.

### 5.2.9 Faixa curva

Este problema trata da análise de vibração livre de uma faixa curva apoiada nas extremidades. A geometria do problema é a mesma da faixa curva discutida na Seção 5.1.6. A descrição do problema, incluindo as propriedades dos materiais e espessura da casca são apresentadas na Figura 135.

Figura 135 – Descrição da faixa curva apoiada.



Fonte: Elaborada pelo autor.

Foram utilizadas seis malhas triangulares geradas pelo algoritmo AFS, sendo três já

apresentadas anteriormente na Figura 95b. Os parâmetros de discretização considerados para geração das malhas são apresentados na Tabela 22.

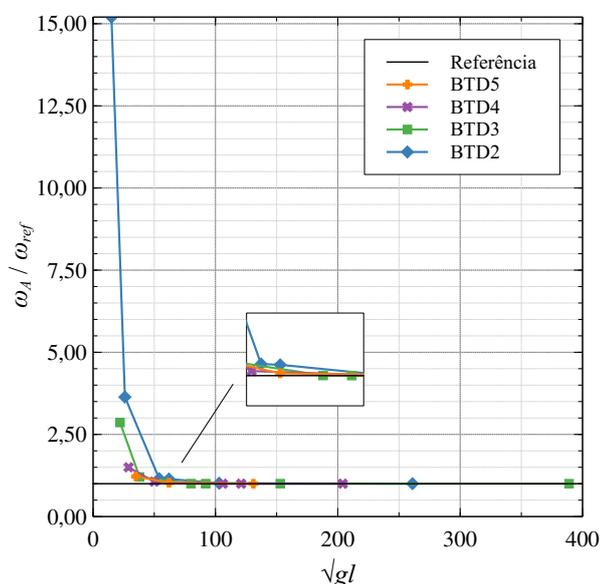
Tabela 22 – Parâmetros de discretização das malhas da faixa curva apoiada.

Malha	$L_{max}$	$\theta_{max}$	$L_{min}$	$\hat{n}$
1	300	90	150	18
2	150	60	75	56
3	75	45	25	264
4	50	30	10	352
5	25	25	0	1010
6	10	15	0	6632

Fonte: Elaborada pelo autor.

A convergência da frequência natural do problema para elementos de diversos graus é apresentada na Figura 136. A frequência natural de referência  $\omega_{ref} = 20.1860$  foi obtida através do ABAQUS CAE, utilizando 28554 elementos S3. O algoritmo de subespaços implementado neste trabalho foi utilizado para obtenção das respostas, considerando uma tolerância de  $10^{-6}$  e número máximo de iterações igual a 30.

Figura 136 – Convergência da frequência natural para faixa curva.



Fonte: Elaborada pelo autor.

Todos os elementos convergiram para a resposta de referência, com as maiores taxas de convergência sendo observadas nos elementos de maior grau. Também são apresentados as 10 primeiras frequências de vibração obtidas pelo FAST e ABAQUS na Tabela 23. Os resultados mostram excelente concordância entre os dois programas, sendo a maior diferença observada de

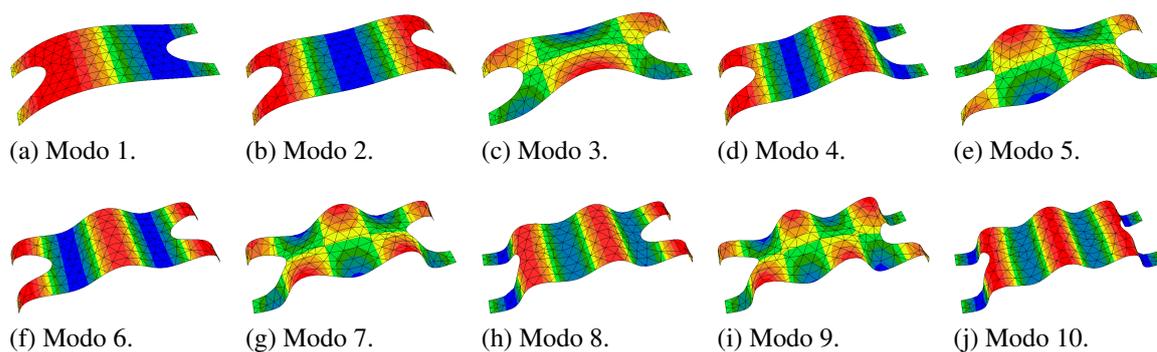
0.55%. Por fim, são apresentados os 10 modos de vibração da estrutura na Figura 137.

Tabela 23 – Frequências de vibração da faixa curva apoiada.

Modo	FAST	ABAQUS	Diferença
1	20.1856	20.1860	0.00%
2	46.5576	46.5480	0.02%
3	73.0362	72.7680	0.37%
4	89.5740	89.5600	0.02%
5	116.5205	115.8800	0.55%
6	137.7613	137.7300	0.02%
7	173.2083	172.5300	0.39%
8	203.2249	203.1700	0.03%
9	233.4962	232.4700	0.44%
10	272.1072	271.8500	0.09%

Fonte: Elaborada pelo autor.

Figura 137 – Modos de vibração da faixa curva apoiada (diagrama de cores para resposta  $w$ ).



Fonte: Elaborada pelo autor.

### 5.2.10 Casca esférica com furos

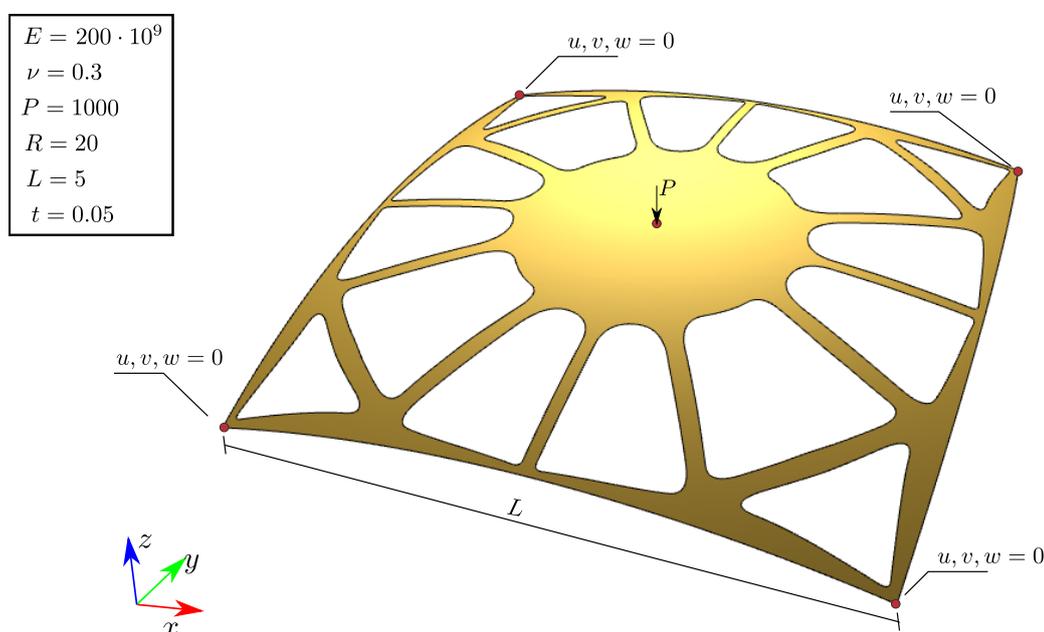
O último problema estudado neste trabalho consiste em uma casca esférica apoiada nas quatro quinas e submetida a uma carga concentrada no centro. O problema tem a mesma geometria da casca esférica com furos discutida na Seção 5.1.6. Os parâmetros do material e descrição do problema de análise são mostrados na Figura 138.

Neste exemplo são utilizadas 7 malhas com diferentes níveis de refinamento, sendo três destas apresentadas na Figura 100.(b). A Tabela 24 mostra os parâmetros de discretização considerados para geração das malhas.

O estudo de convergência do deslocamento  $w$  no ponto de aplicação da carga é apresentado na Figura 139. Novamente, o software ABAQUS CAE foi utilizado para calcular a

resposta de referência  $w_{ref} = -2.5576 \times 10^{-5}$ , obtida com uma malha de 199950 elementos S3.

Figura 138 – Descrição da casca esférica com furos.



Fonte: Elaborada pelo autor.

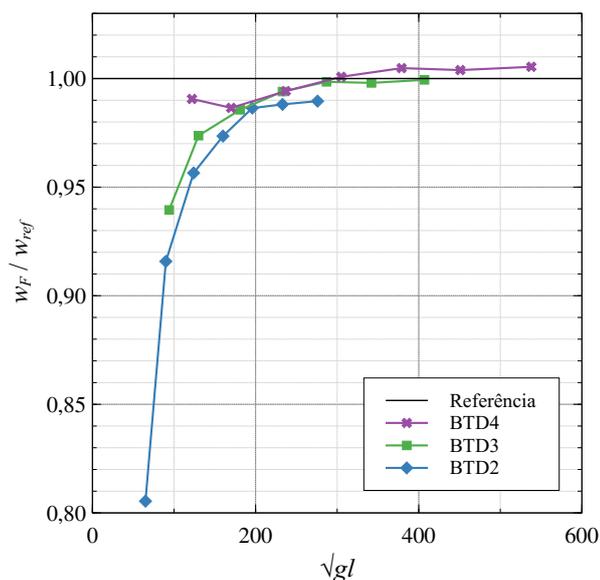
Tabela 24 – Parâmetros de discretização das malhas da casca esférica apoiada.

Malha	$L_{max}$	$\theta_{max}$	$L_{min}$	$\hat{n}$
1	4	90	0.5	320
2	4	90	0.25	634
3	4	90	0.15	1248
4	3	75	0.075	2108
5	2	60	0	3320
6	0.50	30	0	4772
7	0.25	15	0	6838

Fonte: Elaborada pelo autor.

As respostas obtidas neste trabalho pelo FAST convergem para a resposta de referência. Observa-se, entretanto, que os resultados reportados das últimas malhas de cada grau diferem da resposta de referência em torno de 1%. Modelos mais refinados são necessários para reduzir a diferença nas respostas, mas não foram processados pela limitação de recurso computacional disponível no curso deste trabalho. O campo de deslocamento  $w$  do problema é mostrado na Figura 140.

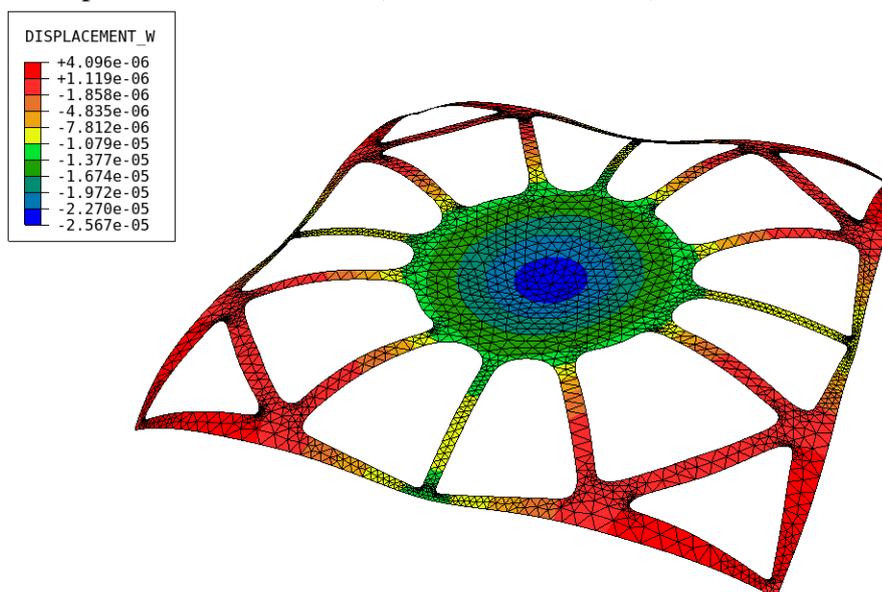
Figura 139 – Convergência do deslocamento  $w_p$  para a casca esférica com furos.



Fonte: Elaborada pelo autor.

Finalmente, ressalta-se que o elemento isogeométrico proposto neste trabalho, juntamente com as técnicas de geração de malhas desenvolvidas, foram utilizadas com sucesso para solução de vários exemplos de análise estrutural de cascas, tanto para problemas estáticos com pequenos deslocamentos quanto para análise de vibração livre, além de problemas bidimensionais de análise térmica e de estado plano de tensão.

Figura 140 – Campo de deslocamento  $w$  (fator de escala =  $2e4$ ) da casca esférica com furos.



Fonte: Elaborada pelo autor.

A capacidade de representação exata da geometria e fácil obtenção de modelos com

grau de aproximação elevados são as principais vantagens desta abordagem em relação ao uso de elementos finitos quadráticos tradicionais. Além disto, o uso de modelos de maior grau mostrou-se eficaz para aliviar o travamento dos elementos nos diversos exemplos apresentados, que em muitos casos, para os elementos MEF tradicionais (Q8 e T6), apresentaram travamentos severos para a formulação apresentada. A abordagem pode ser aprimorada considerando formulações especiais para alívio do travamento.

## 6 CONCLUSÃO

Este trabalho apresentou um algoritmo para a geração automática de malhas isogeométricas não estruturadas compostas por triângulos de Bézier racionais para geometria arbitrárias. O algoritmo obedece a uma discretização (ou parametrização) de entrada do domínio, que é definido por uma B-Rep usando curvas NURBS.

O algoritmo é dividido em quatro etapas: 1) geração de uma malha poligonal usando um algoritmo de avanço de fronteira, que considera os segmentos curvos na avaliação da função densidade; 2) geração de uma malha de alta ordem por elevação de grau e substituição do contorno, restaurando a geometria exata do modelo; 3) realização de modificações na topologia da malha para remover singularidades em elementos de alta ordem; e 4) suavização dos pesos e coordenadas dos pontos de controle para melhorar a qualidade dos elementos nas proximidades das arestas curvas do modelo. Esta última etapa é aplicada localmente, o que melhora o desempenho do algoritmo proposto e evita elementos inválidos e de qualidade ruim.

O algoritmo foi comparado ao TriGA, um programa capaz de gerar malhas isogeométricas não estruturadas compostas por triângulos racionais de Bézier, e resultados superiores foram obtidos em diversos exemplos de com diferentes níveis de complexidade. Além disso, o algoritmo apresentou bom desempenho no caso de geometrias complexas. A técnica de subdivisão automática de curvas proposta possibilitou, em conjunto com o algoritmo de geração de malhas, obter modelos de análise com diferentes resoluções em problemas com geometria complexa.

A etapa de suavização de alta ordem é mais efetiva quanto maior for o grau dos elementos gerados e menor for o nível de refinamento do modelo. Além disso, a estratégia de aplicação localizada da SAO proporcionou ganhos de eficiência notáveis ao método de geração, sem comprometer a efetividade da suavização. O algoritmo proposto também obteve bons resultados em comparação a malhas obtidas via otimização, demonstrando a efetividade da técnica proposta.

Em termos de eficiência, o algoritmo proposto foi comparado ao Gmsh, um gerador de malha de elementos finitos consolidado capaz de produzir e otimizar malhas cúbicas de triângulos não racionais (T10), e resultados semelhantes em termos de qualidade de malha e tempo de execução foram observados. Assim, a implementação do algoritmo proposto apresentou eficiência competitiva, e é capaz de gerar malhas de alta ordem de qualquer grau e geometria exata.

A versão do algoritmo para geração de superfícies aparadas foi comparada com

malhas obtidas via otimização. Novamente, obteve-se malhas válidas em todos os exemplos analisados, e com qualidade comparável a obtida via otimização. A técnica de subdivisão automática de curvas, generalizada para o caso de superfícies, permitiu obter malhas de diferentes resoluções em geometrias complexas. A etapa de SAO adaptada ao caso de superfícies mostrou-se importante para garantir a validade dos elementos de alta ordem gerados. Ressalta-se ainda que a técnica proposta pode ser aplicada também em superfícies T-Splines aparadas.

As técnicas de geração de malhas estruturadas desenvolvidas permitiram a geração de malhas quadrilaterais para os problemas tratados no trabalho. Os elementos de superfície e triângulo de Bézier apresentaram taxas de convergência teóricas, em problemas de elasticidade e transferência de calor. A etapa da SAO mostrou-se importante não só para melhorar a qualidade dos elementos de alta ordem, mas também para melhorar a precisão da resposta numérica obtida pelas malhas suavizadas.

Os elementos isogeométricos para análise de cascas obtiveram resultados em excelente concordância com a literatura. Apesar da formulação apresentada não considerar tratamentos para aliviar travamentos nos elementos, a elevação de grau dos elementos de Bézier mostrou-se uma boa ferramenta para reduzir os travamentos observados. Problemas de cascas complexas, com vários furos, foram solucionados utilizando os elementos propostos, que obtiveram respostas semelhantes as obtidas por elementos finitos convencionais.

Algumas sugestões de trabalhos futuros são apresentadas a seguir:

- a) Generalização do algoritmo de geração de malhas não estruturadas proposto para o caso de malhas isogeométricas volumétricas, onde são gerados tetraedros, hexaedros, cunhas e pirâmides de Bézier racionais.
- b) Desenvolver uma etapa eficiente de otimização de malhas, aplicada depois da SAO, para aumentar a robustez da técnica na geração de elementos válidos. O algoritmo de avaliação de sub-malhas pode ser aplicado para definir sub-regiões do problema de otimização.
- c) Desenvolvimento e implementação de um algoritmo de geração não estruturado de malhas isogeométricas compostas por quadriláteros de Bézier racionais, utilizando um gerador de malhas não estruturadas de quadriláteros.
- d) Formulação de novas métricas de qualidade para elementos de superfície de Bézier racionais de alta ordem.
- e) Elaboração e implementação de técnicas de particionamento automático do domínio para aplicação robusta dos algoritmos de malhas estruturadas isogeométricas.

- f) Extensão da formulação do elemento de casca proposto para análise de carga crítica e análise geometricamente não-linear.
- g) Desenvolvimento de tratamentos para eliminação e redução do travamento de cisalhamento e membrana dos elementos de Bézier.

## REFERÊNCIAS

- 1 COOK, R. D.; MALKUS, D. S.; PLESHA, M. E.; WITT, R. J. **Concepts and applications of finite element analysis**. 4. ed. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2001. 719 p.
- 2 HUGHES, T. J. R.; COTTRELL, J. A.; BAZILEVS, Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 194, n. 39-41, p. 4135–4195, oct 2005.
- 3 COTTRELL, J. A.; HUGHES, T. J.; BAZILEVS, Y. **Isogeometric Analysis: Toward Integration of CAD and FEA**. Chichester, West Sussex, UK: John Wiley & Sons Ltd, 2009. 1–335 p.
- 4 KANG, P.; YOUN, S. K. Isogeometric analysis of topologically complex shell structures. **Finite Elements in Analysis and Design**, Elsevier, v. 99, p. 68–81, jul 2015.
- 5 ENGVALL, L.; EVANS, J. A. Isogeometric triangular Bernstein-Bézier discretizations: Automatic mesh generation and geometrically exact finite element analysis. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 304, p. 378–407, jun 2016.
- 6 ENGVALL, L.; EVANS, J. A. Isogeometric unstructured tetrahedral and mixed-element Bernstein-Bézier discretizations. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 319, p. 83–123, jun 2017.
- 7 LIU, N.; JEFFERS, A. E. A geometrically exact isogeometric Kirchhoff plate: Feature-preserving automatic meshing and  $C^1$  rational triangular Bézier spline discretizations. **International Journal for Numerical Methods in Engineering**, John Wiley & Sons, Ltd, v. 115, n. 3, p. 395–409, jul 2018.
- 8 LÓPEZ, J.; ANITESCU, C.; VALIZADEH, N.; RABCZUK, T.; ALAJLAN, N. Structural shape optimization using Bézier triangles and a CAD-compatible boundary representation. **Engineering with Computers**, Springer London, p. 1–16, may 2019.
- 9 JAXON, N.; QIAN, X. Isogeometric analysis on triangulations. **CAD Computer Aided Design**, Elsevier, v. 46, n. 1, p. 45–57, jan 2014.
- 10 BARROSO, E. S.; EVANS, J. A.; CAVALCANTE-NETO, J. B.; VIDAL, C. A.; PARENTE, E. An efficient automatic mesh generation algorithm for planar isogeometric analysis using high-order rational Bézier triangles. **Engineering with Computers**, Springer, p. 1–22, feb 2022.
- 11 MÄNTYLÄ, M. **Introduction to Solid Modeling**. New York, NY, USA: Computer Science Press, Inc., 1988.
- 12 MEAGHER, D. Geometric modeling using octree encoding. **Computer Graphics and Image Processing**, Academic Press, v. 19, n. 2, p. 129–147, jun 1982.
- 13 Open Cascade. **Open Cascade Technology**. Versão 7.6.0. Disponível em: <dev.opencascade.org>. Acesso em: 24 jul. 2022.
- 14 FARIN, G. **Curves and Surfaces for CAGD A Practical Guide**. 5. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002. v. 3. 521 p.
- 15 COONS, S. A. **Surfaces for computer-aided design of space forms**. Cambridge, MA, USA, 1967.

- 16 RIESENFELD, R. F. **Applications of B-spline Approximation to Geometric Problems of Computer-Aided Design**. Tese (Doutorado) — Syracuse University, Syracuse, NY, USA, 1973.
- 17 VERSPRILLE, K. J. **Computer-Aided Design Applications of the Rational B-Spline Approximation Form**. Tese (Doutorado) — Syracuse University, Syracuse, NY, USA, 1975.
- 18 SEDERBERG, T. W.; ZHENG, J.; BAKENOV, A.; NASRI, A. T-splines and T-NURCCs. **ACM Transactions on Graphics**, ACM, v. 22, n. 3, p. 477, jul 2003.
- 19 SEDERBERG, T. W.; CARDON, D. L.; FINNIGAN, G. T.; NORTH, N. S.; ZHENG, J.; LYCHE, T. T-spline simplification and local refinement. **ACM SIGGRAPH 2004 Papers on - SIGGRAPH '04**, ACM, v. 23, n. 3, p. 276, aug 2004.
- 20 CATMULL, E.; CLARK, J. Recursively generated B-spline surfaces on arbitrary topological meshes. **Computer-Aided Design**, Elsevier, v. 10, n. 6, p. 350–355, nov 1978.
- 21 DOO, D.; SABIN, M. Behaviour of recursive division surfaces near extraordinary points. **Computer-Aided Design**, Elsevier, v. 10, n. 6, p. 356–360, nov 1978.
- 22 LAI, M.-J.; SCHUMAKER, L. L. **Spline Functions on Triangulations**. Cambridge, UK: Cambridge University Press, 2007. (Encyclopedia of Mathematics and its Applications).
- 23 DOKKEN, T.; LYCHE, T.; PETTERSEN, K. F. Polynomial splines over locally refined box-partitions. **Computer Aided Geometric Design**, Elsevier, v. 30, n. 3, p. 331–356, mar 2013.
- 24 JOHANNESSEN, K. A.; KVAMSDAL, T.; DOKKEN, T. Isogeometric analysis using LR B-splines. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 269, p. 471–514, feb 2014.
- 25 OCCELLI, M.; ELGUEDJ, T.; BOUABDALLAH, S.; MORANÇAY, L. LR B-Splines implementation in the Altair Radioss™ solver for explicit dynamics IsoGeometric Analysis. **Advances in Engineering Software**, Elsevier, v. 131, p. 166–185, may 2019.
- 26 DENG, J.; CHEN, F.; LI, X.; HU, C.; TONG, W.; YANG, Z.; FENG, Y. Polynomial splines over hierarchical T-meshes. **Graphical Models**, Academic Press, v. 70, n. 4, p. 76–86, jul 2008.
- 27 WANG, P.; XU, J.; DENG, J.; CHEN, F. Adaptive isogeometric analysis using rational PHT-splines. **Computer-Aided Design**, Elsevier, v. 43, n. 11, p. 1438–1448, nov 2011.
- 28 GIANNELLI, C.; JÜTTLER, B.; SPELEERS, H. THB-splines: The truncated basis for hierarchical splines. **Computer Aided Geometric Design**, Elsevier, v. 29, n. 7, p. 485–498, oct 2012.
- 29 GIANNELLI, C.; JÜTTLER, B.; KLEISS, S. K.; MANTZAFLARIS, A.; SIMEON, B.; ŠPEH, J. THB-splines: An effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 299, p. 337–365, feb 2016.
- 30 CASTELJAU, P. de. **Outillages méthodes calcul**. Paris, 1959.
- 31 CASTELJAU, P. de. **Courbes et surfaces à pôles**. Paris, 1963. 1–58 p.
- 32 HARFORD, A. **GIMP essential reference**. Indianapolis, IN: New Riders, 2000. (New Riders professional library).

- 33 HURTER, B. **The Best of Adobe Photoshop: Techniques and Images from Professional Photographers**. Chicago: Amherst Media Inc., 2006.
- 34 BAH, T. **Inkscape: Guide to a Vector Drawing Program**. First. Upper Saddle River, NJ, USA: Prentice Hall Press, 2007.
- 35 HARREL, W.; STEWARD, W. **CorelDRAW 7 Secrets**. Foster City, CA, USA: IDG Books Worldwide, Inc., 1997.
- 36 LORENTZ, G. G. **Bernstein Polynomials**. second. New York, N.Y.: Chelsea Publishing Company, 1986.
- 37 SEDERBERG, T. W. **Computer Aided Geometric Design**. Provo, UT, USA: Faculty Publications, 2012.
- 38 PIEGL, L.; TILLER, W. **The NURBS Book**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995. (Monographs in Visual Communications).
- 39 MAINAR, E.; PEÑA, J. M. Evaluation algorithms for multivariate polynomials in Bernstein-Bézier form. **Journal of Approximation Theory**, Academic Press, v. 143, n. 1, p. 44–61, nov 2006.
- 40 GOLDMAN, R. N.; FILIP, D. J. Conversion from Bézier rectangles to Bézier triangles. **Computer-Aided Design**, Elsevier, v. 19, n. 1, p. 25–27, jan 1987.
- 41 BARTELS, R. H.; BEATTY, J. C.; BARSKY, B. A. **An Introduction to Splines for Use in Computer Graphics and Geometric Modeling**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1987.
- 42 Robert McNeel & Associates. **Rhinoceros**. Versão 7. Disponível em: <[www.rhino3d.com/](http://www.rhino3d.com/)>. Acesso em: 24 jul. 2022.
- 43 Autodesk. **Maya**. Versão 2023.1. Disponível em: <[www.autodesk.com.br/products/maya/](http://www.autodesk.com.br/products/maya/)>. Acesso em: 24 jul. 2022.
- 44 Autodesk. **AutoCAD**. Versão 2023. Disponível em: <[www.autodesk.com.br/products/autocad/](http://www.autodesk.com.br/products/autocad/)>. Acesso em: 24 jul. 2022.
- 45 BORDEN, M. J.; SCOTT, M. A.; EVANS, J. A.; HUGHES, T. J. R. Isogeometric finite element data structures based on Bézier extraction of NURBS. **International Journal for Numerical Methods in Engineering**, John Wiley & Sons, Ltd, v. 87, n. 1-5, p. 15–47, jul 2011.
- 46 PIEGL, L. On NURBS: A Survey. **IEEE Computer Graphics and Applications**, IEEE, v. 11, n. 1, p. 55–71, jan 1991.
- 47 MARUSSIG, B.; THOMAS, .; HUGHES, J. R. A Review of Trimming in Isogeometric Analysis: Challenges, Data Exchange and Simulation Aspects. **Archives of Computational Methods in Engineering**, Springer, v. 25, p. 1059–1127, 2018.
- 48 XIA, S.; QIAN, X. Isogeometric analysis with Bézier tetrahedra. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 316, p. 782–816, apr 2017.

- 49 THOMAS, D. C.; SCOTT, M. A.; EVANS, J. A.; TEW, K.; EVANS, E. J. Bézier projection: A unified approach for local projection and quadrature-free refinement and coarsening of NURBS and T-splines with particular application to isogeometric design and analysis. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 284, p. 55–105, 2015.
- 50 STROUD, I. **Boundary Representation Modelling Techniques**. London: Springer, 2006. 787 p.
- 51 MCLAURIN, D.; SHONTZ, S. M. Automated Edge Grid Generation Based on Arc-Length Optimization. In: **Proceedings of the 22nd International Meshing Roundtable**. Cham: Springer International Publishing, 2014. p. 385–403.
- 52 ANGEL, E. **Interactive computer graphics : a top-down approach using OpenGL**. Boston, MA, USA: Pearson/Addison-Wesley, 2009.
- 53 FABRI, A.; GIEZEMAN, G.-J.; KETTNER, L.; SCHIRRA, S.; SCHÖNHERR, S. On the design of CGAL a computational geometry algorithms library. **Software: Practice and Experience**, John Wiley & Sons Ltd, v. 30, n. 11, p. 1167–1202, sep 2000.
- 54 GEUZAINÉ, C.; REMACLE, J. F. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. **International Journal for Numerical Methods in Engineering**, John Wiley & Sons, Ltd, v. 79, n. 11, p. 1309–1331, sep 2009.
- 55 Spencer Smith, W.; Adam Lazzarato, D.; CARETTE, J. State of the practice for mesh generation and mesh processing software. **Advances in Engineering Software**, Elsevier, v. 100, p. 53–71, oct 2016.
- 56 HU, Y.; ZHOU, Q.; GAO, X.; JACOBSON, A.; ZORIN, D.; PANOZZO, D. Tetrahedral meshing in the wild. **ACM Transactions on Graphics**, Association for Computing Machinery, v. 37, n. 4, p. 1–14, aug 2018.
- 57 Dassault Systemes. **Catia**. Disponível em: <[www.3ds.com/products-services/catia](http://www.3ds.com/products-services/catia)>. Acesso em: 24 jul. 2022.
- 58 CIMNE. **GiD – The personal pre and post processor**. Disponível em: <[www.gidsimulation.com](http://www.gidsimulation.com)>. Acesso em: 24 jul. 2022.
- 59 SIMULIA. **Abaqus/CAE**. Disponível em: <[www.3ds.com/products-services/simulia/products/abaqus/abaquscae](http://www.3ds.com/products-services/simulia/products/abaqus/abaquscae)>. Acesso em: 24 jul. 2022.
- 60 ANSYS, Inc. **Ansys Mechanical**. Disponível em: <[www.ansys.com](http://www.ansys.com)>. Acesso em: 24 jul. 2022.
- 61 DARREN, E. **MESH2D: Delaunay-based unstructured mesh-generation**. Disponível em: <[github.com/dengwirda/mesh2d](https://github.com/dengwirda/mesh2d)>. Acesso em: 10 jul. 2020.
- 62 ENGVALL, L. **TriGA: Triangular IGA**. Disponível em: <[github.com/CMGLab/TriGA](https://github.com/CMGLab/TriGA)>. Acesso em: 10 jul. 2020.
- 63 KOWALSKI, N.; LEDOUX, F.; FREY, P. A PDE based approach to multidomain partitioning and quadrilateral meshing. In: **Proceedings of the 21st International Meshing Roundtable**. San Jose, California, USA: Kluwer Academic Publishers, 2013. p. 137–154.

- 64 MIRANDA, A. C. O.; MARTHA, L. F. Hierarchical template-based quadrilateral mesh generation. **Engineering with Computers**, Springer London, v. 33, n. 4, p. 701–715, oct 2017.
- 65 FREY, P. J.; GEORGE, P.-L. **Mesh Generation: Application to Finite Elements**. 2nd. ed. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2008. 848 p.
- 66 FIGUEIREDO, L. H.; CARVALHO, P. C. P. **Introdução à Geometria Computacional**. Rio de Janeiro, RJ, Brazil: IMPA Instituto de Matematica Pura Aplicada, 1991.
- 67 CHEW, L. P. Guaranteed-Quality Mesh Generation for Curved Surfaces. In: **Proceedings of the Ninth Annual Symposium on Computational Geometry**. New York, NY, USA: Association for Computing Machinery, 1993. (SCG '93), p. 274–280.
- 68 RUPPERT, J. A delaunay refinement algorithm for quality 2-dimensional mesh generation. **Journal of Algorithms**, Academic Press, v. 18, n. 3, p. 548–585, may 1995.
- 69 SHEWCHUK, J. R. Triangle: Engineering a 2D quality mesh generator and delaunay triangulator. In: **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**. Berlin, DE: Springer Verlag, 1996. v. 1148, p. 203–222.
- 70 ENGWIRDA, D. **Locally-optimal Delaunay-refinement and optimisation-based mesh generation**. Tese (Doutorado) — The University of Sydney, 2014.
- 71 YERRY, M. A.; SHEPHARD, M. S. A Modified Quadtree Approach to Finite Element Mesh Generation. **IEEE Computer Graphics and Applications**, IEEE, v. 3, n. 1, p. 39–46, jan 1983.
- 72 YERRY, M. A.; SHEPHARD, M. S. Automatic three-dimensional mesh generation by the modified-octree technique. **International Journal for Numerical Methods in Engineering**, Wiley-Blackwell, v. 20, n. 11, p. 1965–1990, nov 1984.
- 73 KELA, A.; PERUCCHIO, R.; VOELCKER, H. B. Toward automatic finite element analysis. **ASME Comp. Mech. Eng.**, John Wiley & Sons, Inc., v. 5, 1987.
- 74 PERUCCHIO, R.; SAXENA, M.; KELA, A. Automatic mesh generation from solid models based on recursive spatial decompositions. **International Journal for Numerical Methods in Engineering**, v. 28, n. 11, p. 2469–2501, 1989.
- 75 CAVALCANTE NETO, J. B.; CARVALHO, M. T. M.; MARTHA, L. F. Combinação de Técnicas de Quadtree e Delaunay para Geração Automática de Malhas de Elementos Finitos. In: **VI Brazilian Symposium on Computer Graphics and Image Processing**. Recife, Pernambuco: Sociedade Brasileira de Computação, 1993. p. XII Brazilian Symposium on Computer Graphics and I.
- 76 CAMATA, J. J.; COUTINHO, A. L. Parallel implementation and performance analysis of a linear octree finite element mesh generation scheme. **Concurrency Computation Practice and Experience**, John Wiley & Sons Ltd, v. 25, n. 6, p. 826–842, apr 2013.
- 77 LO, S. A new generation scheme for arbitrary planar domains. **International Journal for Numerical Methods in Engineering**, John Wiley & Sons Ltd, v. 21, p. 1403–1426, 1985.

- 78 MIRANDA, A. C. O.; CAVALCANTE NETO, J. B.; MARTHA, L. F. An algorithm for two-dimensional mesh generation for arbitrary regions with cracks. In: **Proceedings - 12th Brazilian Symposium on Computer Graphics and Image Processing, SIBGRAPI 1999**. Campinas, SP, Brasil: IEEE Comput. Soc, 1999. p. 29–38.
- 79 FREITAS, M. O.; WAWRZYNEK, P. A.; CAVALCANTE-NETO, J. B.; VIDAL, C. A.; MARTHA, L. F.; INGRAFFEA, A. R. A distributed-memory parallel technique for two-dimensional mesh generation for arbitrary domains. **Advances in Engineering Software**, Elsevier, v. 59, p. 38–52, 2013.
- 80 MIRANDA, A. C. O.; LIRA, W. W. M.; CAVALCANTE-NETO, J. B.; SOUSA, R. A.; MARTHA, L. F. A Three-Dimensional Adaptive Mesh Generation Approach Using Geometric Modeling With Multi-Regions and Parametric Surfaces. **Journal of Computing and Information Science in Engineering**, American Society of Mechanical Engineers, v. 13, n. 2, apr 2013.
- 81 FREITAS, M. O.; WAWRZYNEK, P. A.; CAVALCANTE-NETO, J. B.; VIDAL, C. A.; CARTER, B. J.; MARTHA, L. F.; INGRAFFEA, A. R. Parallel generation of meshes with cracks using binary spatial decomposition. **Engineering with Computers**, Springer London, v. 32, n. 4, p. 655–674, oct 2016.
- 82 DEY, S.; O'BARA, R. M.; SHEPHARD, M. S. Towards curvilinear meshing in 3D: The case of quadratic simplices. **CAD Computer Aided Design**, Elsevier, v. 33, n. 3, p. 199–209, mar 2001.
- 83 LU, Q.; SHEPHARD, M. S.; TENDULKAR, S.; BEALL, M. W. Parallel mesh adaptation for high-order finite element methods with curved element geometry. **Engineering with Computers**, Springer London, v. 30, n. 2, p. 271–286, apr 2014.
- 84 GEUZAINÉ, C.; JOHNEN, A.; LAMBRECHTS, J.; REMACLE, J. F.; TOULORGE, T. The generation of valid curvilinear meshes. **Notes on Numerical Fluid Mechanics and Multidisciplinary Design**, Springer Verlag, v. 128, p. 15–39, 2015.
- 85 ROCA, X.; GARGALLO-PEIRO, A.; SARRATE, J. Defining quality measures for high-order planar triangles and curved mesh generation. In: **Proceedings of the 20th International Meshing Roundtable**. Paris, França: Springer Berlin Heidelberg, 2011. p. 365–383.
- 86 TOULORGE, T.; GEUZAINÉ, C.; REMACLE, J. F.; LAMBRECHTS, J. Robust untangling of curvilinear meshes. **Journal of Computational Physics**, Academic Press Inc., v. 254, p. 8–26, dec 2013.
- 87 JOHNEN, A.; REMACLE, J. F.; GEUZAINÉ, C. Geometrical validity of curvilinear finite elements. **Journal of Computational Physics**, Academic Press Inc., v. 233, n. 1, p. 359–372, jan 2013.
- 88 GARGALLO-PEIRÓ, A.; ROCA, X.; PERAIRE, J.; SARRATE, J. Distortion and quality measures for validating and generating high-order tetrahedral meshes. **Engineering with Computers**, Springer London, v. 31, n. 3, p. 423–437, jul 2015.
- 89 JOHNEN, A.; GEUZAINÉ, C.; TOULORGE, T.; REMACLE, J. F. Efficient computation of the minimum of shape quality measures on curvilinear finite elements. **CAD Computer Aided Design**, Elsevier, v. 103, p. 24–33, oct 2018.

- 90 LUO, X. J.; SHEPHARD, M. S.; LEE, L. Q.; GE, L.; NG, C. Moving curved mesh adaptation for higher-order finite element simulations. **Engineering with Computers**, Springer-Verlag, v. 27, n. 1, p. 41–50, jan 2011.
- 91 RUIZ-GIRONÉS, E.; GARGALLO-PEIRÓ, A.; SARRATE, J.; ROCA, X. Automatically imposing incremental boundary displacements for valid mesh morphing and curving. **Computer-Aided Design**, Elsevier, v. 112, p. 47–62, jul 2019.
- 92 CARDOZE, D.; CUNHA, A.; MILLER, G. L.; PHILLIPS, T.; WALKINGTON, N. A bézier-based approach to unstructured moving meshes. In: **Proceedings of the twentieth annual symposium on Computational geometry - SCG '04**. New York, New York, USA: ACM Press, 2004. p. 310.
- 93 KADAPA, C. Novel quadratic Bézier triangular and tetrahedral elements using existing mesh generators: Applications to linear nearly incompressible elastostatics and implicit and explicit elastodynamics. **International Journal for Numerical Methods in Engineering**, John Wiley & Sons, Ltd, v. 117, n. 5, p. 543–573, feb 2019.
- 94 MANDAD, M.; CAMPEN, M. Bézier guarding: precise higher-order meshing of curved 2D domains. **ACM Transactions on Graphics (TOG)**, ACM PUB27 New York, NY, USA, v. 39, n. 4, jul 2020.
- 95 BABUSKA, I.; SZABO, B. A.; KATZ, I. N. The  $p$ -Version of the Finite Element Method. **SIAM Journal on Numerical Analysis**, Society for Industrial and Applied Mathematics, v. 18, n. 3, p. 515–545, jun 1981.
- 96 KIENDL, J.; BLETZINGER, K. U.; LINHARD, J.; WÜCHNER, R. Isogeometric shell analysis with Kirchhoff-Love elements. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 198, n. 49-52, p. 3902–3914, nov 2009.
- 97 BENSON, D. J.; BAZILEVS, Y.; HSU, M. C.; HUGHES, T. J. Isogeometric shell analysis: The Reissner-Mindlin shell. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 199, n. 5-8, p. 276–289, jan 2010.
- 98 BREITENBERGER, M.; APOSTOLATOS, A.; PHILIPP, B.; WÜCHNER, R.; BLETZINGER, K. U. Analysis in computer aided design: Nonlinear isogeometric B-Rep analysis of shell structures. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 284, p. 401–457, feb 2015.
- 99 KIM, H. J.; SEO, Y. D.; YOUN, S. K. Isogeometric analysis for trimmed CAD surfaces. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 198, n. 37-40, p. 2982–2995, aug 2009.
- 100 BROVKA, M.; LÓPEZ, J. I.; ESCOBAR, J. M.; CASCÓN, J. M.; MONTENEGRO, R. A new method for T-spline parameterization of complex 2D geometries. **Engineering with Computers**, Springer London, v. 30, n. 4, p. 457–473, oct 2014.
- 101 WANG, W.; ZHANG, Y.; LIU, L.; HUGHES, T. J. R. R. Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology. **CAD Computer Aided Design**, Elsevier, v. 45, n. 2, p. 351–360, feb 2013.

- 102 LIU, L.; ZHANG, Y.; HUGHES, T. J. R.; SCOTT, M. A.; SEDERBERG, T. W. Volumetric T-spline construction using Boolean operations. **Engineering with Computers**, Springer London, v. 30, n. 4, p. 425–439, oct 2013.
- 103 ESCOBAR, J. M.; MONTENEGRO, R.; RODRÍGUEZ, E.; CASCÓN, J. M. The mecano method for isogeometric solid modeling and applications. **Engineering with Computers**, Springer London, v. 30, n. 3, p. 331–343, jul 2014.
- 104 Al Akhras, H.; ELGUEDJ, T.; GRAVOUIL, A.; ROCHETTE, M. Isogeometric analysis-suitable trivariate NURBS models from standard B-Rep models. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 307, p. 256–274, aug 2016.
- 105 LÓPEZ, J. I.; BROVKA, M.; ESCOBAR, J. M.; MONTENEGRO, R.; SOCORRO, G. V. Spline parameterization method for 2D and 3D geometries based on T-mesh optimization. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 322, p. 460–482, aug 2017.
- 106 SHAMANSKIY, A.; GFRERER, M. H.; HINZ, J.; SIMEON, B. Isogeometric parametrization inspired by large elastic deformation. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 363, p. 112920, may 2020.
- 107 ENGVALL, L. **Geometrically Exact and Analysis Suitable Mesh Generation Using Rational Bernstein–Bezier Elements**. Tese (Doutorado) — University of Colorado, jan 2018.
- 108 ENGVALL, L.; EVANS, J. A. Mesh quality metrics for isogeometric Bernstein–Bézier discretizations. **Computer Methods in Applied Mechanics and Engineering**, v. 371, p. 113305, 2020.
- 109 PAV, S. E.; WALKINGTON, N. J. Delaunay refinement by corner lopping. **Proceedings of the 14th International Meshing Roundtable**, Kluwer Academic Publishers, p. 165–181, 2005.
- 110 PERSSON, P. O. Mesh size functions for implicit geometries and PDE-based gradient limiting. **Engineering with Computers**, Springer-Verlag, v. 22, n. 2, p. 95–109, sep 2006.
- 111 BARROSO, E. S.; EVANS, J. A.; CAVALCANTE NETO, J. B.; VIDAL, C. A.; Parente Junior, E. An algorithm for automatic discretization of isogeometric plane models. In: **XL Iberian Latin-American Congress on Computational Methods in Engineering**. Natal, RN, Brasil: ABMEC, 2019.
- 112 FREY, P. J.; MARÉCHAL, L. Fast Adaptive Quadtree Mesh Generation. In: **Proc. 7Th International Meshing Roundtable**. Dearborn, MI, USA: Sandia National Laboratories, 1998. p. 211—224.
- 113 PINTO, F. D. M.; FREITAS, C. M. D. S. Fast medial axis transform for planar domains with general boundaries. In: **Proceedings of SIBGRAPI 2009 - 22nd Brazilian Symposium on Computer Graphics and Image Processing**. Rio de Janeiro, RJ, Brasil: IEEE, 2009. p. 96–103.
- 114 XIE, Z. Q.; SEVILLA, R.; HASSAN, O.; MORGAN, K. The generation of arbitrary order curved meshes for 3D finite element analysis. **Computational Mechanics**, Springer-Verlag, v. 51, n. 3, p. 361–374, mar 2013.

- 115 ABGRALL, R.; DOBRZYNSKI, C.; FROEHLI, A. A method for computing curved meshes via the linear elasticity analogy, application to fluid dynamics problems. **International Journal for Numerical Methods in Fluids**, John Wiley & Sons, Ltd, v. 76, n. 4, p. 246–266, oct 2014.
- 116 MOXEY, D.; EKELSCHOT, D.; KESKIN, U.; SHERWIN, S. J.; PEIRÓ, J. High-order curvilinear meshing using a thermo-elastic analogy. **CAD Computer Aided Design**, Elsevier, v. 72, p. 130–139, mar 2016.
- 117 GONDEGAON, S.; VORUGANTI, H. K. An efficient parametrization of planar domain for isogeometric analysis using harmonic functions. **Journal of the Brazilian Society of Mechanical Sciences and Engineering**, Springer Berlin Heidelberg, v. 40, n. 10, p. 493, oct 2018.
- 118 HINZ, J.; MÖLLER, M.; VUIK, C. Elliptic grid generation techniques in the framework of isogeometric analysis applications. **Computer Aided Geometric Design**, Elsevier, v. 65, p. 48–75, oct 2018.
- 119 HABER, R.; SHEPHARD, M. S.; ABEL, J. F.; GALLAGHER, R. H.; GREENBERG, D. P. A general two-dimensional, graphical finite element preprocessor utilizing discrete transfinite mappings. **International Journal for Numerical Methods in Engineering**, John Wiley & Sons, Ltd, v. 17, n. 7, p. 1015–1044, jul 1981.
- 120 MIRANDA, A. C. O.; MARTHA, L. F. Mapeamento transfinito tridimensional. In: **Proceedings of the XX Iberian Latin-American Congress on Computational Methods in Engineering**. São Paulo: ABMEC, 1999. p. 1–13.
- 121 GONÇALVES, M. A. B. **Geração de Malhas Bidimensionais de Elementos Finitos Baseada em Mapeamentos Transfinitos**. 97 p. Tese (Dissertação de Mestrado) — Universidade Federal de Minas Gerais, Belo Horizonte, 2004.
- 122 LASSER, D. Triangular subpatches of rectangular Bézier surfaces. **Computers and Mathematics with Applications**, Pergamon, v. 55, n. 8, p. 1706–1719, apr 2008.
- 123 BARROSO, E. S.; CAVALCANTE NETO, J. B.; VIDAL, C. A.; Parente Junior, E. Geração de malhas isogeometricas utilizando mapeamento transfinito. In: **Proceedings of the XXXVIII Iberian Latin American Congress on Computational Methods in Engineering**. Florianópolis: ABMEC, 2017.
- 124 SHEPHARD, M. S.; FLAHERTY, J. E.; JANSEN, K. E.; LI, X.; LUO, X.; CHEVAUGEON, N.; REMACLE, J. F.; BEALL, M. W.; O'BARA, R. M. Adaptive mesh generation for curved domains. **Applied Numerical Mathematics**, Elsevier, v. 52, n. 2-3 SPEC. ISS., p. 251–271, feb 2005.
- 125 CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Introduction to Algorithms**. 3rd. ed. Cambridge, MA, USA: The MIT Press, 2009. 1292 p.
- 126 CAVALCANTE-NETO, J. B.; MARTHA, L. F.; WAWRZYNEK, P. A.; INGRAFFEA, A. R. A back-tracking procedure for optimization of simplex meshes. **Communications in Numerical Methods in Engineering**, John Wiley & Sons, Ltd, v. 21, n. 12, p. 711–722, sep 2005.

- 127 BAZILEVS, Y.; BEIRÃO Da VEIGA, L.; COTTRELL, J. A.; HUGHES, T. J.; SANGALLI, G. Isogeometric analysis: Approximation, stability and error estimates for h-refined meshes. **Mathematical Models and Methods in Applied Sciences**, World Scientific Publishing Company, v. 16, n. 7, p. 1031–1090, jul 2006.
- 128 POYA, R.; SEVILLA, R.; GIL, A. J. A unified approach for a posteriori high-order curved mesh generation using solid mechanics. **Computational Mechanics**, Springer Berlin Heidelberg, v. 58, n. 3, p. 457–490, sep 2016.
- 129 PERSSON, P.-O.; PERAIRE, J. Curved Mesh Generation and Mesh Refinement using Lagrangian Solid Mechanics. In: **47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition**. Reston, VA, USA: American Institute of Aeronautics and Astronautics, 2013.
- 130 KNUPP, P. Label-invariant mesh quality metrics. In: **Proceedings of the 18th International Meshing Roundtable**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 139–155.
- 131 KNUPP, P. M. Algebraic mesh quality metrics for unstructured initial meshes. **Finite Elements in Analysis and Design**, Elsevier, v. 39, n. 3, p. 217–241, jan 2003.
- 132 TRISTANO, J. R.; OWEN, S. J.; CANANN, S. A. Advancing Front Surface Mesh Generation in Parametric Space Using a Riemannian Surface Definition. In: **Proc. 7Th International Meshing Roundtable**. Dearborn, MI, USA: Sandia National Laboratories, 1998. p. 429–445.
- 133 MIRANDA, A. C. O.; MARTHA, L. F. Mesh Generation on High-Curvature Surfaces Based on a Background Quadtree Structure. In: **Proc. 11Th International Meshing Roundtable**. Ithaca, NY, USA: Sandia National Laboratories, 2002. p. 333—342.
- 134 MIRANDA, A. C.; MARTHA, L. F.; WAWRZYNEK, P. A.; INGRAFFEA, A. R. Surface mesh regeneration considering curvatures. **Engineering with Computers**, Springer, v. 25, n. 2, p. 207–219, dec 2009.
- 135 BARROSO, E. S.; RIBEIRO, L. G.; MAIA, M. A.; ROCHA, I. B. C. M. da; PARENTE, E.; MELO, A. M. C. de. BIOS: an object-oriented framework for Surrogate-Based Optimization using bio-inspired algorithms. **Structural and Multidisciplinary Optimization**, Springer, v. 65, n. 7, p. 203, jul 2022.
- 136 RIBEIRO, L. G.; MARTINS, J. M. F. C.; Parente Jr., E.; MELO, A. M. C. de. Surrogate Based Optimization of Functionally Graded Plates using PSO and DE. In: **XLI Ibero-Latin American Congress on Computational Methods in Engineering**. Foz do Iguaçu, PR, Brasil: ABMEC, 2020.
- 137 BARROSO, E. S. **Análise e otimização de estruturas laminadas utilizando formulação isogeométrica**. Tese (Dissertação de Mestrado) — Universidade Federal do Ceará, Brasil, Ceará, 2015.
- 138 TRAN, L. V.; LEE, J.; NGUYEN-VAN, H.; NGUYEN-XUAN, H.; WAHAB, M. A. Geometrically nonlinear isogeometric analysis of laminated composite plates based on higher-order shear deformation theory. **International Journal of Non-Linear Mechanics**, Pergamon, v. 72, p. 42–52, jun 2015.

- 139 YU, T. T.; YIN, S.; BUI, T. Q.; HIROSE, S. A simple FSDT-based isogeometric analysis for geometrically nonlinear analysis of functionally graded plates. **Finite Elements in Analysis and Design**, Elsevier, v. 96, n. C, p. 1–10, apr 2015.
- 140 OESTERLE, B.; RAMM, E.; BISCHOFF, M. A shear deformable, rotation-free isogeometric shell formulation. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 307, p. 235–255, aug 2016.
- 141 COTTRELL, J. A.; HUGHES, T. J. R.; REALI, A. Studies of refinement and continuity in isogeometric structural analysis. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 196, n. 41-44, p. 4160–4183, sep 2007.
- 142 VUONG, A. T.; GIANNELLI, C.; JÜTTLER, B.; SIMEON, B. A hierarchical approach to adaptive local refinement in isogeometric analysis. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 200, n. 49-52, p. 3554–3567, dec 2011.
- 143 SCHILLINGER, D.; DEDÈ, L.; SCOTT, M. A.; EVANS, J. A.; BORDEN, M. J.; RANK, E.; HUGHES, T. J. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 249-252, n. 0, p. 116–150, dec 2012.
- 144 GARAU, E. M.; VÁZQUEZ, R. Algorithms for the implementation of adaptive isogeometric methods using hierarchical B-splines. **Applied Numerical Mathematics**, Elsevier, v. 123, p. 58–87, jan 2018.
- 145 COTTRELL, J. A.; EVANS, J. A.; LIPTON, S.; SCOTT, M. A.; SEDERBERG, T. W. Isogeometric analysis using T-splines. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 199, n. 5-8, p. 229–263, jan 2010.
- 146 LIU, Z.; CHENG, J.; YANG, M.; YUAN, P.; QIU, C.; GAO, W.; TAN, J. Isogeometric analysis of large thin shell structures based on weak coupling of substructures with unstructured T-splines patches. **Advances in Engineering Software**, Elsevier, v. 135, p. 102692, sep 2019.
- 147 LUDWIG, T.; HÜHNE, C.; De Lorenzis, L. Rotation-free Bernstein–Bézier elements for thin plates and shells — development and validation. **Computer Methods in Applied Mechanics and Engineering**, Elsevier B.V., v. 348, p. 500–534, may 2019.
- 148 XIA, S.; WANG, X.; QIAN, X. Continuity and convergence in rational triangular Bézier spline based isogeometric analysis. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 297, p. 292–324, dec 2015.
- 149 ZAREH, M.; QIAN, X. Kirchhoff–Love shell formulation based on triangular isogeometric analysis. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 347, p. 853–873, apr 2019.
- 150 DUNAVANT, D. A. High degree efficient symmetrical Gaussian quadrature rules for the triangle. **International Journal for Numerical Methods in Engineering**, John Wiley & Sons, Ltd, v. 21, n. 6, p. 1129–1148, jun 1985.
- 151 FISH, J.; BELYTSCHKO, T. **A First Course in Finite Elements**. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2007. 1–320 p.

- 152 JUNIOR, J. a. B. M. S. **Auto-Adaptação e Otimização de Forma em Cascas**. Tese (Doutorado) — Pontifícia Universidade Católica do Rio de Janeiro, 2000.
- 153 DORNISCH, W.; KLINKEL, S.; SIMEON, B. Isogeometric Reissner–Mindlin shell analysis with exactly calculated director vectors. **Computer Methods in Applied Mechanics and Engineering**, v. 253, p. 491–504, 2013.
- 154 UHM, T.-K.; YOUN, S.-K. T-spline finite element method for the analysis of shell structures. **International Journal for Numerical Methods in Engineering**, John Wiley & Sons, Ltd, v. 80, n. 4, p. 507–536, oct 2009.
- 155 BATHE, K. **Finite Element Procedures**. 2. ed. Watertown, MA, USA: Englewood Cliffs, N.J. Prentice Hall, 2014. 1073 p.
- 156 ZIENKIEWICZ, O. C.; TAYLOR, R. L.; ZHU, J. Z. **The Finite Element Method: Its Basis and Fundamentals**. 6. ed. London, England: Butterworth-Heinemann, 2005.
- 157 KANG, P.; YOUN, S. K. Isogeometric shape optimization of trimmed shell structures. **Structural and Multidisciplinary Optimization**, Springer Berlin Heidelberg, v. 53, n. 4, p. 825–845, apr 2016.
- 158 KANG, P.; YOUN, S. K. Isogeometric topology optimization of shell structures using trimmed NURBS surfaces. **Finite Elements in Analysis and Design**, Elsevier B.V., v. 120, p. 18–40, nov 2016.
- 159 BELYTSCHKO, T.; STOLARSKI, H.; LIU, W. K.; CARPENTER, N.; ONG, J. S. Stress projection for membrane and shear locking in shell finite elements. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 51, n. 1-3, p. 221–258, sep 1985.
- 160 SCORDELIS, A. C.; LO, K. S. Computer Analysis of Cylindrical Shells. **ACI Journal Proceedings**, v. 61, n. 5, 1964.
- 161 LI, J.; SHI, Z.; LIU, L. A scaled boundary finite element method for static and dynamic analyses of cylindrical shells. **Engineering Analysis with Boundary Elements**, Elsevier Ltd, v. 98, p. 217–231, jan 2019.
- 162 DVORKIN, E. N.; BATHE, K. A continuum mechanics based four-node shell element for general non-linear analysis. **Engineering Computations**, MCB UP Ltd, v. 1, n. 1, p. 77–88, jan 1984.
- 163 MACNEAL, R. H.; HARDER, R. L. A proposed standard set of problems to test finite element accuracy. **Finite Elements in Analysis and Design**, v. 1, n. 1, p. 3–20, 1985.
- 164 LEI, Z.; GILLOT, F.; JEZEQUEL, L. An isogeometric Reissner–Mindlin shell element based on mixed grid. **Advances in Mechanical Engineering**, SAGE Publications Inc., v. 10, n. 4, apr 2018.

## APÊNDICE A – DESCRIÇÃO DOS EXEMPLOS

Este apêndice apresenta os dados geométricos (coordenadas e pesos dos pontos de controle e vetores de knots) dos exemplos abordados na presente tese. Os exemplos com geometrias planas são disponibilizados no endereço eletrônico <<https://github.com/lmcv-ufc/PMGen>>. Os exemplos com modelos de superfícies, com e sem curvas de aparos, são apresentados a seguir.

Tabela 25 – Pontos de controle e vetor de *knots* de um quarto da casca de Scordelos-Lo.

id	x	y	z	w
1	+0.000000e+00	+0.000000e+00	+5.848889e+00	+1.000000e+00
2	+9.099256e+00	+0.000000e+00	+5.848889e+00	+9.396930e-01
3	+1.606969e+01	+0.000000e+00	+0.000000e+00	+1.000000e+00
4	+0.000000e+00	+2.500000e+01	+5.848889e+00	+1.000000e+00
5	+9.099256e+00	+2.500000e+01	+5.848889e+00	+9.396930e-01
6	+1.606969e+01	+2.500000e+01	+0.000000e+00	+1.000000e+00

$\Xi = [0\ 0\ 0\ 1\ 1\ 1], H = [0\ 0\ 1\ 1]$

Fonte: Elaborada pelo autor.

Tabela 26 – Pontos de controle e vetor de *knots* da casca de Scordelos-Lo.

id	x	y	z	w
1	+0.000000e+00	+0.000000e+00	+0.000000e+00	+1.000000e+00
2	+1.606969e+01	+0.000000e+00	+1.348407e+01	+7.660444e-01
3	+3.213938e+01	+0.000000e+00	+0.000000e+00	+1.000000e+00
4	+0.000000e+00	+5.000000e+01	+0.000000e+00	+1.000000e+00
5	+1.606970e+01	+5.000000e+01	+1.348407e+01	+7.660444e-01
6	+3.213938e+01	+5.000000e+01	+0.000000e+00	+1.000000e+00

$\Xi = [0\ 0\ 0\ 1\ 1\ 1], H = [0\ 0\ 1\ 1]$

Fonte: Elaborada pelo autor.

Tabela 27 – Pontos de controle e vetor de *knots* do furo central da casca de Scordelos-Lo.

id	r	s	w
1	+3.939340e-01	+4.259521e-01	+1.000000e+00
2	+2.878680e-01	+5.000000e-01	+7.071068e-01
3	+3.939340e-01	+5.740479e-01	+1.000000e+00
4	+5.000000e-01	+6.480957e-01	+7.071068e-01
5	+6.060660e-01	+5.740479e-01	+1.000000e+00
6	+7.121320e-01	+5.000000e-01	+7.071068e-01
7	+6.060660e-01	+4.259521e-01	+1.000000e+00
8	+5.000000e-01	+3.519043e-01	+7.071068e-01
9	+3.939340e-01	+4.259521e-01	+1.000000e+00

$\Xi = [0\ 0\ 0\ .25\ .25\ .5\ .5\ .75\ .75\ 1\ 1\ 1]$

Fonte: Elaborada pelo autor.

Tabela 28 – Pontos de controle e vetor de *knots* do oitavo de cilindro.

id	x	y	z	w
1	+0.000000e+00	+0.000000e+00	+3.000000e+02	+1.000000e+00
2	+3.000000e+02	+0.000000e+00	+3.000000e+02	+7.071070e-01
3	+3.000000e+02	+0.000000e+00	+0.000000e+00	+1.000000e+00
4	+3.000000e+02	+3.000000e+02	+0.000000e+00	+1.000000e+00
5	+3.000000e+02	+3.000000e+02	+3.000000e+02	+7.071068e-01
6	+0.000000e+00	+3.000000e+02	+3.000000e+02	+1.000000e+00

$\Xi = [0\ 0\ 0\ 1\ 1\ 1], H = [0\ 0\ 1\ 1]$

Fonte: Elaborada pelo autor.

Tabela 29 – Pontos de controle e vetor de *knots* do oitavo de esfera com corte.

id	x	y	z	w
1	+1.000000e+01	+0.000000e+00	+0.000000e+00	+1.000000e+00
2	+1.000000e+01	+1.000000e+01	+0.000000e+00	+7.071068e-01
3	+0.000000e+00	+1.000000e+01	+0.000000e+00	+1.000000e+00
4	+1.000000e+01	+0.000000e+00	+7.265425e+00	+8.090170e-01
5	+1.000000e+01	+1.000000e+01	+7.265425e+00	+5.720614e-01
6	+0.000000e+00	+1.000000e+01	+7.265425e+00	+8.090170e-01
7	+3.090170e+00	+0.000000e+00	+9.510565e+00	+1.000000e+00
8	+3.090170e+00	+3.090170e+00	+9.510565e+00	+7.071068e-01
9	+0.000000e+00	+3.090170e+00	+9.510565e+00	+1.000000e+00

$\Xi = H = [0\ 0\ 0\ 1\ 1\ 1]$

Fonte: Elaborada pelo autor.

Tabela 30 – Pontos de controle e vetor de *knots* da superfície de forma livre.

id	x	y	z	id	x	y	z
1	+0.000000e+00	+0.000000e+00	+0.000000e+00	19	+1.000000e+01	+5.000000e+00	+0.000000e+00
2	+0.000000e+00	+0.000000e+00	+5.000000e+00	20	+9.703704e+00	+4.444444e+00	+3.444444e+00
3	+0.000000e+00	+2.000000e+00	+7.000000e+00	21	+9.111111e+00	+4.148148e+00	+6.925926e+00
4	+0.000000e+00	+2.000000e+00	+1.000000e+01	22	+8.222222e+00	+3.592593e+00	+1.081481e+01
5	+0.000000e+00	+0.000000e+00	+1.200000e+01	23	+7.629630e+00	+3.444444e+00	+1.311111e+01
6	+0.000000e+00	+0.000000e+00	+1.500000e+01	24	+7.333333e+00	+4.000000e+00	+1.500000e+01
7	+5.000000e+00	+0.000000e+00	+0.000000e+00	25	+1.000000e+01	+0.000000e+00	+0.000000e+00
8	+4.580247e+00	+0.000000e+00	+4.740741e+00	26	+9.975309e+00	+0.000000e+00	+2.925926e+00
9	+3.740741e+00	+1.802469e+00	+6.987654e+00	27	+9.925926e+00	+4.197531e-01	+6.901235e+00
10	+2.481481e+00	+1.987654e+00	+1.013580e+01	28	+9.851852e+00	+1.901235e+00	+1.108642e+01
11	+1.641975e+00	+4.814815e-01	+1.218519e+01	29	+9.802469e+00	+3.851852e+00	+1.348148e+01
12	+1.222222e+00	+6.666667e-01	+1.500000e+01	30	+9.777778e+00	+5.333333e+00	+1.500000e+01
13	+5.000000e+00	+5.000000e+00	+0.000000e+00	31	+1.100000e+01	+0.000000e+00	+0.000000e+00
14	+4.851852e+00	+4.444444e+00	+4.222222e+00	32	+1.100000e+01	+0.000000e+00	+2.666667e+00
15	+4.555556e+00	+4.740741e+00	+6.962963e+00	33	+1.100000e+01	+2.222222e-01	+6.888889e+00
16	+4.111111e+00	+3.629630e+00	+1.040741e+01	34	+1.100000e+01	+1.888889e+00	+1.122222e+01
17	+3.814815e+00	+2.000000e+00	+1.255556e+01	35	+1.100000e+01	+4.333333e+00	+1.366667e+01
18	+3.666667e+00	+2.000000e+00	+1.500000e+01	36	+1.100000e+01	+6.000000e+00	+1.500000e+01

$\Xi = H = [0\ 0\ 0\ 1/3\ 2/3\ 1\ 1\ 1]$

Fonte: Elaborada pelo autor.

Tabela 31 – Pontos de controle e vetor de *knots* da faixa curva.

id	x	y	z
1	0.000000e+00	0.000000e+00	0.000000e+00
2	1.500000e+02	0.000000e+00	1.000000e+02
3	3.000000e+02	0.000000e+00	1.000000e+02
4	4.500000e+02	0.000000e+00	1.000000e+02
5	6.000000e+02	0.000000e+00	0.000000e+00
6	0.000000e+00	2.000000e+02	0.000000e+00
7	1.500000e+02	2.000000e+02	1.000000e+02
8	3.000000e+02	2.000000e+02	1.000000e+02
9	4.500000e+02	2.000000e+02	1.000000e+02
10	6.000000e+02	2.000000e+02	0.000000e+00

$\Xi = [0\ 0\ 0\ 1\ 1\ 1], H = [0\ 0\ 1\ 1]$

(a) Superfície NURBS.

id	r	s
1	0.00	0.80
2	0.20	0.70
3	0.20	0.50
4	0.20	0.30
5	0.00	0.20

$\Xi = [0\ 0\ 0\ 0\ .5\ .5\ 1\ 1\ 1\ 1]$

(b) Curva  $c_1$ .

id	r	s
1	1.00	0.20
2	0.80	0.30
3	0.80	0.50
4	0.80	0.70
5	1.00	0.80

$\Xi = [0\ 0\ 0\ 0\ .5\ .5\ 1\ 1\ 1\ 1]$

(c) Curva  $c_2$ .

Fonte: Elaborada pelo autor.

Tabela 32 – Pontos de controle e vetor de *knots* da faixa curva modificada (furos adicionais).

id	$r$	$s$	$w$
1	0.50	0.65	1.0
2	0.55	0.65	$\sqrt{2}/2$
3	0.55	0.50	1.0
4	0.55	0.35	$\sqrt{2}/2$
5	0.50	0.35	1.0
6	0.45	0.35	$\sqrt{2}/2$
7	0.45	0.50	1.0
8	0.45	0.65	$\sqrt{2}/2$
9	0.50	0.65	1.0

$\Xi = [0\ 0\ 0\ .25\ .25\ .5\ .5\ .75\ .75\ 1\ 1\ 1]$

(a) Curva  $h_1$ .

id	$r$	$s$	$w$
1	0.430	0.450	1.0
2	0.430	0.300	$\sqrt{2}/2$
3	0.480	0.300	1.0
4	0.480	0.150	1.0
5	0.480	0.050	1.0
6	0.330	0.050	1.0
7	0.200	0.050	1.0
8	0.200	0.150	1.0
9	0.380	0.300	1.0
10	0.400	0.450	1.0
11	0.415	0.450	1.0
12	0.430	0.450	1.0

$\Xi = [0\ 0\ 0\ .2\ .2\ .4\ .4\ .6\ .6\ .7\ .8\ .8\ 1\ 1\ 1]$

(b) Curva  $h_2$ .

id	$r$	$s$	$w$
1	0.570	0.450	1.0
2	0.585	0.450	1.0
3	0.600	0.450	1.0
4	0.620	0.300	1.0
5	0.800	0.150	1.0
6	0.800	0.050	1.0
7	0.670	0.050	1.0
8	0.520	0.050	1.0
9	0.520	0.150	1.0
10	0.520	0.300	1.0
11	0.570	0.300	$\sqrt{2}/2$
12	0.570	0.450	1.0

$\Xi = [0\ 0\ 0\ .2\ .2\ .3\ .4\ .4\ .6\ .6\ .8\ .8\ 1\ 1\ 1]$

(c) Curva  $h_3$ .

id	$r$	$s$	$w$
1	0.430	0.550	1.0
2	0.415	0.550	1.0
3	0.400	0.550	1.0
4	0.380	0.700	1.0
5	0.200	0.850	1.0
6	0.200	0.950	1.0
7	0.330	0.950	1.0
8	0.480	0.950	1.0
9	0.480	0.850	1.0
10	0.480	0.700	1.0
11	0.430	0.700	$\sqrt{2}/2$
12	0.430	0.550	1.0

$\Xi = [0\ 0\ 0\ .2\ .2\ .3\ .4\ .4\ .6\ .6\ .8\ .8\ 1\ 1\ 1]$

(d) Curva  $h_4$ .

id	$r$	$s$	$w$
1	0.570	0.550	1.0
2	0.570	0.700	$\sqrt{2}/2$
3	0.520	0.700	1.0
4	0.520	0.850	1.0
5	0.520	0.950	1.0
6	0.670	0.950	1.0
7	0.800	0.950	1.0
8	0.800	0.850	1.0
9	0.620	0.700	1.0
10	0.600	0.550	1.0
11	0.585	0.550	1.0
12	0.570	0.550	1.0

$\Xi = [0\ 0\ 0\ .2\ .2\ .4\ .4\ .6\ .6\ .7\ .8\ .8\ 1\ 1\ 1]$

(e) Curva  $h_5$ .

Fonte: Elaborada pelo autor.

Tabela 33 – Pontos de controle e vetor de *knots* da casca esférica (parte 1).

id	<i>x</i>	<i>y</i>	<i>z</i>	<i>w</i>	id	<i>r</i>	<i>s</i>
1	-2.474040e+00	-2.474040e+00	-7.062330e-03	1.000000e+00	1	0.9836	0.0286
2	0.000000e+00	-2.474040e+00	6.463229e-01	9.668508e-01	2	0.9886	0.0122
3	2.474040e+00	-2.474040e+00	-7.062330e-03	1.000000e+00	3	0.9523	0.0193
4	-2.635346e+00	0.000000e+00	6.037225e-01	9.689124e-01	4	0.8421	0.0399
5	0.000000e+00	0.000000e+00	1.299708e+00	9.367938e-01	5	0.7582	0.0433
6	2.635346e+00	0.000000e+00	6.037225e-01	9.689124e-01	6	0.8448	0.1161
7	-2.474040e+00	2.474040e+00	-7.062330e-03	1.000000e+00	7	0.8839	0.1552
8	0.000000e+00	2.474040e+00	6.463229e-01	9.668508e-01	8	0.9567	0.2418
9	2.474040e+00	2.474040e+00	-7.062330e-03	1.000000e+00	9	0.9601	0.1579
10					10	0.9836	0.0286

$\Xi = H = [0\ 0\ 0\ 1\ 1\ 1]$

(a) Superfície NURBS.

(b) Curva  $h_1$ .

id	<i>r</i>	<i>s</i>									
1	0.9714	0.9836	1	0.0164	0.9714	1	0.0286	0.0164	1	0.6164	0.2288
2	0.9878	0.9886	2	0.0114	0.9878	2	0.0122	0.0114	2	0.6765	0.1175
3	0.9807	0.9523	3	0.0477	0.9807	3	0.0193	0.0477	3	0.7151	0.0385
4	0.9601	0.8421	4	0.1579	0.9601	4	0.0399	0.1579	4	0.6626	0.0259
5	0.9567	0.7582	5	0.2418	0.9567	5	0.0433	0.2418	5	0.5353	0.0141
6	0.8839	0.8448	6	0.1552	0.8839	6	0.1161	0.1552	6	0.5097	0.0107
7	0.8448	0.8839	7	0.1161	0.8448	7	0.1552	0.1161	7	0.5097	0.0336
8	0.7582	0.9567	8	0.0433	0.7582	8	0.2418	0.0433	8	0.5101	0.1045
9	0.8421	0.9601	9	0.0399	0.8421	9	0.1579	0.0399	9	0.5101	0.1838
10	0.9714	0.9836	10	0.0164	0.9714	10	0.0286	0.0164	10	0.5097	0.2715

$\Xi = [0\ 0\ 0\ .125\ .250\ .375\ .5\ .625\ .750\ .875\ 1\ 1\ 1]$

(c) Curva  $h_2$ .

(d) Curva  $h_3$ .

(e) Curva  $h_4$ .

(f) Curva  $h_5$ .

id	<i>r</i>	<i>s</i>	id	<i>r</i>	<i>s</i>	id	<i>r</i>	<i>s</i>
1	0.7712	0.6164	1	0.3836	0.7712	1	0.2288	0.3836
2	0.8825	0.6765	2	0.3235	0.8825	2	0.1175	0.3235
3	0.9615	0.7151	3	0.2849	0.9615	3	0.0385	0.2849
4	0.9741	0.6626	4	0.3374	0.9741	4	0.0259	0.3374
5	0.9859	0.5353	5	0.4647	0.9859	5	0.0141	0.4647
6	0.9893	0.5097	6	0.4903	0.9893	6	0.0107	0.4903
7	0.9664	0.5097	7	0.4903	0.9664	7	0.0336	0.4903
8	0.8955	0.5101	8	0.4899	0.8955	8	0.1045	0.4899
9	0.8162	0.5101	9	0.4899	0.8162	9	0.1838	0.4899
10	0.7285	0.5097	10	0.4903	0.7285	10	0.2715	0.4903
11	0.7132	0.5891	11	0.4109	0.7132	11	0.2868	0.4109
12	0.7712	0.6164	12	0.3836	0.7712	12	0.2288	0.3836

$\Xi = [0\ 0\ 0\ .1\ .2\ .3\ .4\ .5\ .6\ .7\ .8\ .9\ 1\ 1\ 1]$

(g) Curva  $h_6$ .

(h) Curva  $h_7$ .

(i) Curva  $h_8$ .

Fonte: Elaborada pelo autor.

Tabela 34 – Pontos de controle e vetor de *knots* da casca esférica (parte 2).

id	r	s									
1	0.6373	0.2450	1	0.7550	0.6373	1	0.3627	0.7550	1	0.2450	0.3627
2	0.6011	0.3025	2	0.6975	0.6011	2	0.3989	0.6975	2	0.3025	0.3989
3	0.6670	0.3471	3	0.6529	0.6670	3	0.3330	0.6529	3	0.3471	0.3330
4	0.6949	0.3950	4	0.6050	0.6949	4	0.3051	0.6050	4	0.3950	0.3051
5	0.7571	0.3592	5	0.6408	0.7571	5	0.2429	0.6408	5	0.3592	0.2429
6	0.8032	0.3311	6	0.6689	0.8032	6	0.1968	0.6689	6	0.3311	0.1968
7	0.8482	0.3084	7	0.6916	0.8482	7	0.1518	0.6916	7	0.3084	0.1518
8	0.8852	0.2887	8	0.7113	0.8852	8	0.1148	0.7113	8	0.2887	0.1148
9	0.9390	0.2643	9	0.7357	0.9390	9	0.0610	0.7357	9	0.2643	0.0610
10	0.8749	0.1916	10	0.8084	0.8749	10	0.1251	0.8084	10	0.1916	0.1251
11	0.8459	0.1639	11	0.8361	0.8459	11	0.1541	0.8361	11	0.1639	0.1541
12	0.8303	0.1492	12	0.8508	0.8303	12	0.1697	0.8508	12	0.1492	0.1697
13	0.8009	0.1193	13	0.8807	0.8009	13	0.1991	0.8807	13	0.1193	0.1991
14	0.7323	0.0601	14	0.9399	0.7323	14	0.2677	0.9399	14	0.0601	0.2677
15	0.7075	0.1101	15	0.8899	0.7075	15	0.2925	0.8899	15	0.1101	0.2925
16	0.6892	0.1475	16	0.8525	0.6892	16	0.3108	0.8525	16	0.1475	0.3108
17	0.6644	0.1950	17	0.8050	0.6644	17	0.3356	0.8050	17	0.1950	0.3356
18	0.6373	0.2450	18	0.7550	0.6373	18	0.3627	0.7550	18	0.2450	0.3627

$\Xi = [0\ 0\ 0\ .0625\ .125$ $.1875\ .25\ .3125\ .375\ .4375$ $.5\ .5625\ .625\ .6875\ .75$ $.8125\ .875\ .9375\ 1\ 1\ 1]$	$\Xi = [0\ 0\ 0\ .0625\ .125$ $.1875\ .25\ .3125\ .375\ .4375$ $.5\ .5625\ .625\ .6875\ .75$ $.8125\ .875\ .9375\ 1\ 1\ 1]$	$\Xi = [0\ 0\ 0\ .0625\ .125$ $.1875\ .25\ .3125\ .375\ .4375$ $.5\ .5625\ .625\ .6875\ .75$ $.8125\ .875\ .9375\ 1\ 1\ 1]$	$\Xi = [0\ 0\ 0\ .0625\ .125$ $.1875\ .25\ .3125\ .375\ .4375$ $.5\ .5625\ .625\ .6875\ .75$ $.8125\ .875\ .9375\ 1\ 1\ 1]$
--	--	--	--

(a) Curva  $h_8$ .

(b) Curva  $h_{10}$ .

(c) Curva  $h_{11}$ .

(d) Curva  $h_{12}$ .

id	r	s									
1	0.7712	0.3836	1	0.6164	0.7712	1	0.2288	0.6164	1	0.3836	0.2288
2	0.7132	0.4109	2	0.5891	0.7132	2	0.2868	0.5891	2	0.4109	0.2868
3	0.7285	0.4903	3	0.5097	0.7285	3	0.2715	0.5097	3	0.4903	0.2715
4	0.8162	0.4899	4	0.5101	0.8162	4	0.1838	0.5101	4	0.4899	0.1838
5	0.8955	0.4899	5	0.5101	0.8955	5	0.1045	0.5101	5	0.4899	0.1045
6	0.9664	0.4903	6	0.5097	0.9664	6	0.0336	0.5097	6	0.4903	0.0336
7	0.9893	0.4903	7	0.5097	0.9893	7	0.0107	0.5097	7	0.4903	0.0107
8	0.9859	0.4647	8	0.5353	0.9859	8	0.0141	0.5353	8	0.4647	0.0141
9	0.9741	0.3374	9	0.6626	0.9741	9	0.0259	0.6626	9	0.3374	0.0259
10	0.9615	0.2849	10	0.7151	0.9615	10	0.0385	0.7151	10	0.2849	0.0385
11	0.8825	0.3235	11	0.6765	0.8825	11	0.1175	0.6765	11	0.3235	0.1175
12	0.7712	0.3836	12	0.6164	0.7712	12	0.2288	0.6164	12	0.3836	0.2288

$\Xi = [0\ 0\ 0\ .1\ .2\ .3$ $.4\ .5\ .6\ .7\ .8\ .9\ 1\ 1\ 1]$	$\Xi = [0\ 0\ 0\ .1\ .2\ .3$ $.4\ .5\ .6\ .7\ .8\ .9\ 1\ 1\ 1]$	$\Xi = [0\ 0\ 0\ .1\ .2\ .3$ $.4\ .5\ .6\ .7\ .8\ .9\ 1\ 1\ 1]$	$\Xi = [0\ 0\ 0\ .1\ .2\ .3$ $.4\ .5\ .6\ .7\ .8\ .9\ 1\ 1\ 1]$
--	--	--	--

(e) Curva  $h_{13}$ .

(f) Curva  $h_{14}$ .

(g) Curva  $h_{15}$ .

(h) Curva  $h_{16}$ .

Fonte: Elaborada pelo autor.