

Um método de refinamento local de T-splines conformes para análise

Um método de refinamento local de T-splines conformes para análise

Gabriel Carvalho Sanches Rocha

Dissertação apresentada à Faculdade de
Computação da Universidade Federal de
Mato Grosso do Sul como parte dos requi-
sitos para obtenção do título de Mestre em
Ciência da Computação.

Orientador: Prof. Dr. Paulo Aristarco Pagliosa

Campo Grande - MS

2025

Agradecimentos

Agradeço aos meus pais, Thaís e Ivan, por sempre me incentivarem a estudar e buscar conhecimento, suas lutas permitiram que eu chegasse até aqui. Agradeço também à minha noiva Millena, que esteve ao meu lado durante todos os desafios e obstáculos, por sua paciência, companheirismo e apoio.

Estendo meus agradecimentos aos meus queridos amigos, sempre dispostos a ajudar, apesar de seus próprios percalços.

Contemplo também em meus agradecimentos, o Prof. Dr. Paulo A. Pagliosa, por toda sua paciência e compreensão. Serei eternamente grato por esse período de aprendizagem, e me sinto muito feliz por ser seu colega durante a pesquisa. Muito obrigado também ao Dr. Márcio A. Peres por numerosas ajudas em meio a suas outras obrigações, seu auxílio também foi primordial no desenvolvimento deste trabalho.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Este trabalho está inserido no contexto de um projeto de pesquisa **Aplicações de Análise Isogeométrica em GPU Usando Representações Herméticas de Superfícies**, patrocinado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico – Brasil (CNPq) – sob concessão #422465/2021-6.

Resumo

Rocha, G.C.S. *Um método de refinamento local de T-splines conformes para análise*. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Mato Grosso do Sul, 2025.

No contexto de simulação computacional de problemas físicos, a *análise isogeométrica* (IGA) de um sólido é aquela em que o modelo de análise tem a mesma ordem de continuidade do modelo geométrico do sólido. A representação geométrica mais utilizada na literatura de IGA é baseada em retalhos de superfícies NURBS (*non-uniform rational B-splines*), visto ser essa a forma de representação mais comumente empregada na indústria de desenho e manufatura assistida por computador (CAD/CAM). Uma superfície NURBS, e, em IGA, o modelo de análise dela derivado, tem sua geometria definida por funções de interpolação racionais que atuam em uma malha estruturada de *pontos de controle*. Uma alternativa recentemente adotada em softwares de CAD/CAM são *T-splines*. Estas generalizam superfícies NURBS admitindo formação de superfícies herméticas com malhas de pontos de controle não estruturadas e com junções em T, chamadas *T-malhas*. Em adição, permitem *refinamento local*, resultante da adição de vértices e arestas e da subdivisão de faces de uma T-malha. Embora diversos trabalhos atestem a viabilidade de T-splines em IGA, ainda existem limitações quanto às características das T-splines utilizadas. Essas restrições motivam a definição de um subconjunto de T-splines ditas *conformes para análise*. Neste trabalho, exploram-se métodos de refinamento de T-splines conformes para análise sob malhas não estruturadas. Para tanto, emprega-se uma estrutura de dados hierárquica para representação computacional de T-malhas, chamada HBDS (*handle-body data structure*). Os objetivos desse trabalho incluem desenvolver uma aplicação gráfica interativa capaz de apresentar T-splines, manipular seus elementos topológicos, refinar sua T-malha e prescrever arestas de vinco, além de permitir montagem flexível de cenas, com T-splines geradas pela aplicação desenvolvida, ou importação de malhas prontas de terceiros.

Palavras-chave: *superfícies paramétricas herméticas; T-splines; T-splines conformes para análise; refinamento local*

Abstract

Rocha, G.C.S. *A local refinement method for analysis-suitable T-splines*. Thesis (Master's in Computer Science) – Federal University of Mato Grosso do Sul, 2025.

In the context of computational simulation of physical problems, *isogeometric analysis* (IGA) of a solid is one in which the analysis model has the same order of continuity as the geometric model of the solid. The most commonly used geometric representation in the IGA literature is based on NURBS (*non-uniform rational B-splines*) surface patches, as this is the most commonly employed representation in the computer-aided design and manufacturing (CAD/CAM) industry. A NURBS surface, and consequently the analysis model derived from it in IGA, has its geometry defined by rational interpolation functions acting on a structured mesh of *control points*. A recently adopted alternative in CAD/CAM software are *T-splines*. These generalize NURBS with watertight surfaces that allow unstructured control point meshes with T-junctions, called *T-meshes*. Additionally, they enable *local refinement*, which results in the addition of vertices and edges and the subdivision of faces in a T-mesh. Although various studies confirm the feasibility of T-splines in IGA, there are still limitations regarding the characteristics of the T-splines used. These restrictions motivate the definition of a subset of T-splines referred to as *analysis-suitable*. This work explores refinement methods for analysis-suitable T-splines under unstructured meshes. To this end, a hierarchical data structure for the computational representation of T-meshes, called HBDS (*handle-body data structure*), is employed. The objectives of this work include developing an interactive graphical application capable of presenting T-splines, manipulating their topological elements, refining their T-mesh, prescribing crease edges, besides flexible scene composition, either with T-splines generated by the developed application or by importing pre-existing meshes from third-party sources.

Keywords: *watertight parametric surfaces; T-splines; analysis-suitable T-splines; local refinement*

Conteúdo

Lista de Figuras	vii
1 Introdução	1
1.1 Motivação e justificativa	1
1.2 Objetivos	5
1.3 Organização do texto	6
2 Fundamentos e trabalhos relacionados	7
2.1 Considerações iniciais	7
2.2 T-splines	8
2.2.1 Extração de Bézier para faces regulares	15
2.2.2 Extração de Bézier para faces irregulares	19
2.2.3 Pontos de tangência	22
2.3 Refinamento local de T-splines	24
2.3.1 Inserção de nós na T-malha	25
2.3.2 Algoritmo de refinamento local	26
2.4 HBDS	30
2.4.1 Componentes topológicos da HBDS	30
2.4.2 Operadores topológicos	32
2.5 Considerações finais	34
3 Desenvolvimento	36
3.1 Considerações iniciais	36
3.2 Criação e importação de T-malhas	37

3.2.1	<i>Quadball</i>	38
3.3	Geração de T-splines	39
3.4	Funcionalidades de edição e refinamento	43
3.5	Algoritmo de resolução de inconsistências	46
3.6	Considerações finais	48
4	Resultados	50
4.1	Considerações iniciais	50
4.2	Exemplos iniciais	50
4.3	Edição e refinamento em T-splines	52
4.4	Considerações finais	59
5	Conclusão	60
5.1	Obstáculos e limitações	61
5.2	Trabalhos futuros	61
	Referências Bibliográficas	63

Lista de Figuras

2.1	T-malha não estruturada.	9
2.2	T-malha com arestas de vinco.	10
2.3	T-malha estendida.	12
2.4	Extração de Bézier.	14
2.5	Funções de forma de Bernstein para elemento de Bézier bidimensional.	14
2.6	T-malha estendida: configuração válida.	16
2.7	Cômputo dos vetores de nós locais.	17
2.8	Funções de forma a partir de vetores de nós estendidos.	17
2.9	Relação de um elemento com diversos domínios.	18
2.10	Funções de forma de Bézier após inserção de nós.	18
2.11	Pontos de controle em T-splines.	20
2.12	Extração de Bézier para faces irregulares com vinco.	22
2.13	Pontos de tangência sobre arestas e faces de um vértice incidente em aresta de vinco.	24
2.14	Inserção de nó na T-malha.	26
2.15	Efeito da inserção de nós nas funções de forma.	27
2.16	T-malhas estendidas e a geração do grafo de extensão acoplada.	29
2.17	Componentes topológicos da estrutura de dados HBDS.	31
2.18	Aplicação do operador <i>glue</i>	33
2.19	Inserção de vértice em malha HBDS	33
3.1	Interface completa da aplicação gráfica interativa.	37

3.2	Diagrama UML das classes envolvidas com a representação da classe <i>TMesh</i> , que modela a T-malha.	38
3.3	Diagrama de classes UML representando a relação entre as principais classes envolvidas na geração de T-Spline.	40
3.4	Determinação de região de faces irregulares via topologia.	41
3.5	Inserção topológica de uma extensão de face na T-malha.	42
3.6	Organização dos pontos de tangência vinculados a um vértice.	43
3.7	Diagrama de classes UML dos comandos que modificam T-splines.	43
3.8	Subdivisão de uma aresta sob peso t	44
3.9	Inserção topológica de aresta na T-malha.	44
3.10	Subdivisão topológica de face na T-malha.	45
3.11	Cruzamentos de extensões de T-junção na T-malha.	47
3.12	Inconsistências ocasionadas por extensões de T-junção na região 3-anel de face de um ponto extraordinário.	48
4.1	Exemplos de T-malhas produzidas pela aplicação.	51
4.2	Cilindro perfurado importado de arquivo TSM.	51
4.3	Edição de forma T-spline planar.	52
4.4	T-spline planar com face subdividida em duas.	53
4.5	T-spline planar com face subdividida em quatro.	53
4.6	T-spline cúbica antes e depois da prescrição de vincos.	54
4.7	Refinamento uniforme de região irregular de T-spline cúbica.	54
4.8	Outro refinamento uniforme de região irregular de T-spline cúbica.	55
4.9	Preparação de T-malha cúbica para inserção de aresta.	57
4.10	T-malha admissível gerada pelo algoritmo de resolução de inconsistências.	58
4.11	Repetidas subdivisões da face central de uma malha planar.	58

Capítulo 1

Introdução

1.1 Motivação e justificativa

Aplicações de computação gráfica estão presentes em diversas áreas, tais como ciências e engenharias, indústria de entretenimento e jogos digitais. Em todas essas, a escolha de um modelo geométrico — representação da forma e das dimensões dos corpos — adequado está atrelada às necessidades particulares de cada aplicação, diferenciando-se na precisão visual e numérica de suas representações, bem como a eficiência computacional e versatilidade dos modelos geométricos.

Aplicações de efeitos especiais e animações têm ênfase na factibilidade visual, ou seja, o resultado produzido deve ser o mais indistinguível possível aos olhos humanos do esperado conforme a experiência cotidiana, por esse motivo, toleram-se representações menos acuradas numericamente em comparação a aplicações para ciências e engenharias. Similarmente, para aplicações interativas, como jogos digitais, a maior restrição é o processamento em tempo real. Isso implica sacrificar, em certo grau, precisão numérica e visual para obter resultados mais rápidos, geralmente relacionados a manipulação do modelo geométrico, e então proporcionar uma experiência fluida aos usuários. Por fim, aplicações de ciências e engenharias necessitam de modelos exatos, muitas vezes associados a restrições de sua forma, como continuidade da superfície, visto que esses modelos geralmente são aplicados em problemas de análise.

Dentre as opções de modelos geométricos para superfícies, existem opções como malhas de triângulos, em que a superfície é representada exata ou aproximadamente por um conjunto de triângulos. Um dos problemas com esse modelo é a dificuldade de representação de regiões curvas e suaves, já que para tanto, pode ser necessária uma malha densa que, dependendo da aplicação, pode prejudicar a eficiência computacional.

As representações baseadas em superfícies paramétricas constituem modelos geométricos de elevada versatilidade, capazes de descrever superfícies suaves. Geralmente, essas superfícies são definidas por um conjunto de pontos de controle, que são associados em uma malha de pontos de controle e também funções de forma (ou funções de base). Superfícies NURBS (*Non-Uniform Rational B-Splines*, ou B-Splines racionais não uniformes) [24] são o modelo geométrico mais amplamente utilizado nas indústrias de *design* e manufatura assistida por computador (CAD e CAM, respectivamente). Em particular, as geometrias em projetos de CAD dificilmente podem ser descritas por uma única superfície NURBS, por isso utiliza-se uma composição de retalhos de NURBS, as *trimmed* NURBS [25].

Apesar de sua versatilidade, a utilização de retalhos de NURBS como modelo geométrico envolve diversos desafios, em especial a introdução de frestas no agrupamento desses retalhos. Outras representações incluem superfícies de subdivisão [37] e T-Splines [33]. As superfícies de subdivisão, atual padrão da indústria de animação, são contempladas na API aberta da Pixar denominada OpenSubdiv [26], posteriormente adotada pela DreamWorks [11]. T-splines são uma generalização de NURBS que continuam ganhando espaço na indústria CAD/CAM; essa geometria está presente no programa Autodesk Fusion 360 [1].

Outro aspecto relacionado é a representação computacional da malha de pontos de controle das superfícies paramétricas. Usualmente, são utilizadas estruturas de dados capazes de representar e manipular os componentes topológicos que constituem a malha em questão, que no caso de NURBS e T-splines, são vértices, arestas e faces. DeFloriani e Hui [12] compilam diversas representações possíveis, apresentando valiosas comparações entre elas. Dentre elas destaca-se a estrutura de dados de semi-arestas (*half-edges*) [19], estendida por Castelo et al. na estrutura de dados *handle-edge* [8], mais detalhes encontram-se no Capítulo 2.

No contexto de ciências e engenharia tem-se a área de simulação de fenômenos físicos, onde emprega-se além do modelo geométrico, um modelo matemático que descreve o comportamento dos corpos em questão. Tais modelos matemáticos são geralmente descritos por equações diferenciais parciais e suas condições iniciais e de contorno. Soluções analíticas para esses modelos frequentemente não podem ser obtidas, e portanto, utilizam-se métodos numéricos que forneçam soluções aproximadas em versões discretas das superfícies e/ou volumes dos corpos simulados. Ao solucionar as equações em uma versão discreta do corpo, obtém-se valores de grandezas como deslocamentos e tensões em pontos do sistema discreto, cujos valores podem ser interpolados para pontos fora da discretização.

Quanto a quais métodos numéricos são utilizados para simulação de corpos deformáveis, pode-se citar o *método dos elementos finitos* (MEF) [38] e o *método dos elementos de contorno* (MEC) [14]. O MEF submete o corpo a uma discretização de todo o seu volume em uma núvem de pontos, também chamados de pontos nodais, conectados por entidades conhecidas como elementos finitos, por exemplo tetraedros, no caso tridimensional. Como consequência dessa discretização, as equações diferenciais do modelo matemático, agora aplicadas a um meio discreto, transformam-se em um sistema de equações algébricas cuja solução fornece os deslocamentos (e por consequência, as deformações) em cada ponto nodal. É possível obter os deslocamentos em pontos internos dos elementos finitos por meio da interpolação dessa grandeza com as funções de forma.

Já o MEC, que como o nome sugere, emprega discretização da superfície do corpo através de uma malha composta por elementos de superfície, como por exemplo triângulos. Uma notável diferença em relação ao MEF, é que no MEC, as soluções para pontos no volume do corpo fazem referência às soluções em pontos nodais sobre o contorno e portanto, são calculadas com uso destas. Ainda em comparação com o MEF, uma grande vantagem do MEC é a redução da dimensão do problema, pela discretização da superfície apenas, que impacta significativamente na redução do número de incógnitas, e portanto diminui o consumo de recursos computacionais.

Um processo comum para ambos MEC e MEF é a geração de malhas de superfície ou de volume. Estas são versões discretas do domínio contínuo, onde utilizam-se as superfícies paramétricas, no qual as equações diferenciais do modelo matemático são definidas. Além disso, carregam a definição dos elementos, nós e seus relacionamentos, a base dos métodos numéricos baseados em malha. A geração de malhas, por ser um processo de amostragem, pode ser um fator limitante da qualidade dos resultados dos métodos numéricos. Por isso é de suma importância que as malhas empregadas possuam qualidade suficiente para representação precisa dos resultados do fenômeno em estudo.

Além das preocupações com a qualidade das malhas, também faz-se necessário analisar o seu processo de geração, que deve resultar em um modelo conforme para análise, que ainda representa nos dias de hoje um grande gargalo, principalmente na produção e prototipação industrial. Como consequência, torna-se necessário tratar os corpos antes da análise (industrialmente conhecida como CAE, do inglês, *computer aided engeneering*), processo que dificilmente pode ser automatizado para todas as geometrias e casos de simulação, como discutido por Bazilevs et al. [2].

Tendo em vista o cenário de separação entre design e análise, em 2005, Hughes et al. [13] propuseram a chamada *análise isogeométrica* (IGA, do inglês *isogeometric analysis*), uma abordagem alternativa aos métodos dos elementos finitos e de contorno. O conceito principal de IGA consiste em aplicar as mesmas funções de forma do modelo geométrico do objeto na interpolação dos valores desconhecidos do problema físico, sendo assim geometricamente exato. Além disso, a integração entre modelo geométrico e análise também reduz a comunicação com o ambiente de design, visto que refinamentos necessários para melhorar a qualidade da malha podem ser feitos a partir da malha inicial. Vê-se como consequência a redução do tempo de análise direcionado para geração de malhas. Em [13], Hughes et al. empregaram NURBS como modelo geométrico em problemas de mecânica dos fluidos e análise estrutural.

Recentemente, Peres [23] empregou T-splines como modelo geométrico de seu estudo sobre método dos elementos de contorno aplicado à análise isogeométrica para o problema elastostático de sólidos deformáveis com vincos. Como resultado deste trabalho, o autor codificou em C++ e MATLAB[®] um arcabouço para a integração de IGA com MEC, em que os elementos de contorno (ou retalhos de superfície, partindo de uma visão geométrica) utilizados para análise são provenientes de um processo denominado *extração de Bézier*.

O emprego de T-splines em IGA apesar de crescente, entretanto, não é irrestrito. Em 2010, no trabalho de Buffa et al. [6], destacou-se que nem todas as T-splines são diretamente aplicáveis a IGA. Com isso, é possível delimitar um subconjunto de T-splines que são conformes para análise (em inglês, *Analysis-suitable T-splines*, ou AST-splines). As características que definem tais superfícies conformes para análise são discutidas em maiores detalhes posteriormente no texto.

Constatou-se não só no trabalho de Peres, mas na literatura de T-splines e análise isogeométrica, a inabilidade de definir funções escalares ou vetoriais sob regiões arbitrárias da superfície. No contexto de IGA, essas funções podem ser usadas para especificar condições de contorno, por exemplo, deslocamentos e/ou forças de superfície prescritas em um subconjunto dos elementos de contorno. Esse problema ocorre pela natureza interpolante das representações por NURBS ou T-splines, citadas anteriormente. Portanto, um desafio geométrico concernente à área de IGA é a definição de condições de contorno à uma seleção de elementos de contorno, que quando extrapoladas aos pontos de controle não “derramem” essas condições em elementos fora da seleção inicial.

1.2 Objetivos

Este trabalho tem como objetivo geral a produção de uma aplicação capaz de criar, exibir e que permita manipular, com processos de refinamento local, superfícies T-splines. Complementa o objetivo geral deste trabalho, disponibilizar as capacidades supracitadas em aplicação gráfica interativa, resultando em um editor de T-splines e suas malhas de pontos de controle, ou *T-malhas*, como observado na aplicação Autodesk Fusion 360 [1].

Espera-se que o resultado deste trabalho possa ser estendido e utilizado em aplicações de IGA com AST-splines, como a desenvolvida por Peres [23], auxiliando na resolução de problemas reais envolvendo dinâmica de corpos deformáveis.

De acordo com o exposto acima, os objetivos específicos deste trabalho são:

- O1** Revisitar a *extração de Bézier* em T-splines, utilizada para definir a forma da superfície do sólido, permitindo seu uso em aplicações gráficas interativas;
- O2** Implementar uma estrutura de dados topológica que permita representação, edição e refinamento de T-malhas de superfícies T-splines;
- O3** Implementar, com base na estrutura de dados adotada, operadores de refinamento da malha;
- O4** Formular um algoritmo que, automaticamente, corrija violações das restrições de T-splines conformes para análise.

Apesar de existir uma ferramenta CAD tal como Autodesk Fusion 360 [1], que permite a geração, modelagem e refinamento de T-splines de maneira interativa. É além de seu escopo que essas T-splines sejam conformes para análise. Com isso, a principal contribuição deste trabalho é a implementação de refinamento local de T-splines que garante a conformidade para análise, onde as verificações envolvidas neste processo são feitas diretamente na estrutura de dados topológica subjacente, adotada para representar a T-malha. Diferentemente dos processos existentes na literatura, que se apoiam em estruturas de dados adicionais [29], por se basear unicamente na topologia, essa abordagem se mostra especialmente vantajosa quando empregada em aplicações interativas, como a que será apresentada posteriormente neste trabalho.

Cabe salientar que embora termos referentes a IGA como *condições e elementos de contorno*, *conformidade para análise*, dentre outros, tenham sido utilizados no texto até aqui, IGA ou métodos de análise em geral, não constituem os objetivos deste

trabalho. De fato, os problemas abordados advêm de IGA, mas tratam-se de problemas geométricos.

1.3 Organização do texto

O restante do texto é organizado como segue. O Capítulo 2 contém fundamentos e trabalhos relacionados compatíveis. No Capítulo 3, discute-se o programa de computador produzido neste trabalho, suas funcionalidades e como foram alcançadas, bem como aspectos relacionados a implementação. A apresentação dos resultados obtidos dá-se no Capítulo 4. Por fim, no Capítulo 5, tem-se uma discussão acerca dos objetivos, como, e se, foram alcançados, abordando as dificuldades enfrentadas, além de citar temas a serem desenvolvidos em trabalhos futuros.

Capítulo 2

Fundamentos e trabalhos relacionados

2.1 Considerações iniciais

No Capítulo 1 foram apresentados os principais objetivos deste trabalho, que incluem processamento geométrico de T-splines, refinamento local e implementação de uma estrutura de dados topológica para representação da malha de controle das T-splines. Neste capítulo são abordados os fundamentos teóricos referentes a T-splines, como sua formulação matemática, elementos topológicos constituintes, operadores de extração, refinamento, bem como definições acerca de conformidade para análise, além de apresentar uma estrutura de dados para representação das malhas. Esses assuntos são tratados neste capítulo como segue.

Na Seção 2.2 é apresentado um resumo sobre T-splines, conceituando seus fundamentos e componentes topológicos, necessários para o entendimento do processo de *extração de elementos de Bézier*, empregado na geração dos retalhos de T-spline. Estes assuntos não são abordados minuciosamente, tendo em vista a farta literatura sobre eles [33, 32, 5, 31, 29, 30, 23]. O resumo apresentado na Seção 2.2 baseia-se no trabalho de Peres [23]. Após a introdução de T-splines, nas Subseções 2.2.1 e 2.2.2 são apresentados os processos de extração de elementos de Bézier para faces regulares e irregulares, respectivamente. Concluindo os conceitos fundamentais de T-splines, na Subseção 2.2.3 é exposto o procedimento que determina os pontos de tangência, essenciais para permitir T-splines com vinco.

Na Subseção 2.3 apresenta-se o processo de refinamento local de T-splines, destacando sua motivação e diferenças em relação as NURBS, juntamente do tratamento

distinto para o refinamento de regiões irregulares. Em contrapartida, para regiões regulares, refinamento tem como base a inserção de nós na T-malha, apresentado na Subseção 2.3.1. Essa fundamentação permite definir, na Subseção 2.3.2, um algoritmo de refinamento local capaz de produzir T-splines válidas e conformes para análise.

Enfim, na Seção 2.4 é introduzida a estrutura de dados topológica que representará as malhas de pontos de controle de T-splines, são abordadas a sua origem, componentes e capacidade, no que diz respeito a edição da malha por ela representada. Na Subseção 2.4.1 introduz-se os componentes topológicos da estrutura de dados. Operadores sob a estrutura de dados são discutidos na Subseção 2.4.2.

2.2 T-splines

Definidas por Sederberg et al. [33], T-splines são superfícies paramétricas que generalizam NURBS [24]. O uso e popularização de T-splines se devem ao fato dessa representação contornar diversas limitações de NURBS. Em especial, T-splines são capazes de representar geometrias complexas com apenas uma superfície hermética, evitando as frestas geradas por retalhos de NURBS e também com número reduzido de pontos de controle, como evidenciado por Sederberg et al. [32]. Além disso, T-splines permitem refinamento local das superfícies.

Os componentes topológicos de T-splines assemelham-se aos de NURBS, contando com pontos de controle e as associações entre eles em uma malha de controle, conhecida por T-malha. Uma *T-malha*, $\mathcal{M}$, é uma malha poligonal não estruturada contendo as mesmas informações topológicas da malha de controle espacial. Ela é composta de faces, arestas e vértices ($\mathcal{F}$, $\mathcal{A}$ e $\mathcal{V}$, respectivamente), que permitem *T-junções*, isto é, vértices com 3 arestas incidentes, lembrando o formato da letra “T”. Os vértices $v_i \in \mathcal{V}$ são pontos de controle com posição $P_i \in \mathbb{R}^3$ e peso $w_i^p \in \mathbb{R}^+$. Arestas $a_i \in \mathcal{A}$ são segmentos de linha fechados que conectam dois vértices, sem atravessar nenhum outro, à elas atribui-se também um valor real não negativo, w_i^a . Por sua vez, faces $f_i \in \mathcal{F}$ são regiões com quatro lados, delimitadas por arestas. Define-se como valência de um vértice, μ_i , a quantidade de arestas incidentes no vértice v_i .

Vértices, arestas e faces são *componentes topológicos* da T-malha, e serão discutidos em detalhes na Seção 2.4, mediante introdução da estrutura de dados adotada para representação da T-malha.

A existência de vértices (ou pontos) *extraordinários* na T-malha é o que a torna não estruturada. Com isso, caracteriza-se um vértice extraordinário como um vértice

interno — aquele que não incide em arestas de borda — que não é uma T-junção e tem valência diferente de quatro, ou um vértice de borda com valência maior que quatro. Arestas incidentes em vértices extraordinários chamam-se arestas *spoke*.

O conjunto de faces incidentes em v_i é chamado vizinhança 1-anel de face, dada por $\mathcal{R}_i^1$. As faces incidentes em vértices no 1-anel de face de v_i , excluindo as próprias faces em $\mathcal{R}_i^1$, compreendem a vizinhança 2-anel de face, $\mathcal{R}_i^2$, deste vértice. Da mesma maneira obtém-se $\mathcal{R}_i^n$, a vizinhança n -anel de face de v_i . A vizinhança n -disco de face de v_i , $\mathcal{D}_i^n$, é composta pela união de todas as faces em 1-, 2-, $\dots$, n -anel de face. Toda face é dita *regular* se não faz parte da vizinhança 2-disco de face de um vértice extraordinário, caso contrário, será uma face *irregular*. Além disso, os vértices que incidem nas faces da vizinhança 1-anel de face de um vértice extraordinário são também vértices *irregulares*, o restante sendo *regular*. A Figura 2.1 ilustra uma malha com vértices extraordinários e explicita os vértices e faces regulares e irregulares conforme definidos acima.

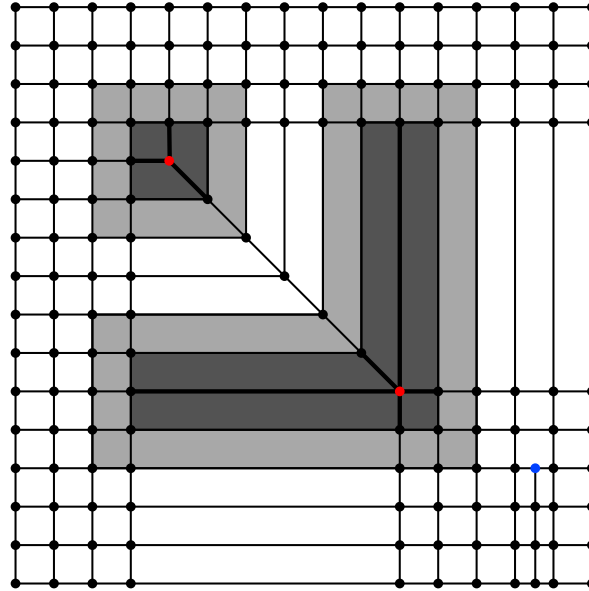


Figura 2.1: T-malha não estruturada. Os pontos em vermelho são vértices extraordinários, uma T-junção é ilustrada por um ponto em azul. Arestas *spoke* são denotadas por linhas pretas grossas, notadamente aquelas incidentes nos vértices extraordinários. As regiões cinza-escuro e cinza-claro correspondem às vizinhanças 1-anel de face e 2-anel de face dos vértices extraordinários, respectivamente.

Embora um retalho de superfície associado a uma face da T-malha possua continuidade C^∞ , podem existir regiões descontínuas em uma T-spline, como nas bordas de superfícies abertas. Essas descontinuidades acontecem em arestas da T-malha chamadas *arestas de vinco*. Para entender o conceito de aresta de vinco, é necessário antes apresentar o conceito de *ponto de tangência*. Um ponto de tangência é tal qual um ponto de controle, possui posição espacial e peso associados, sua diferença dá-se por ser relacionado a um vértice regular na T-malha por um segmento com intervalo

de nó nulo. Vale ressaltar que, de fato, a ligação de um ponto de tangência com seu vértice não é dada por uma aresta da T-malha, justamente por apresentar um valor nulo. Denota-se χ o conjunto de pontos de tangência associados a um vértice regular, a continuidade da curva no valor na coordenada paramétrica de um vértice regular é reduzida pela cardinalidade do conjunto χ . Dito isso, entende-se que arestas de vinco são aquelas que ligam vértices de continuidade local C^0 , gerando um vinco na superfície. Casos triviais são as bordas de uma superfície aberta, enquanto outras ocorrências são relacionadas a redução de continuidade causada pelos pontos de tangência.

A Figura 2.2(a) ilustra uma superfície com vinco, enquanto Figura 2.2(b) apresenta uma visão dos componentes topológicos da T-malha desta superfície, com uma aresta de vinco, arestas de borda e seus pontos de tangência. Como o peso da aresta entre pontos de tangência e os vértices que referenciam é nulo, essa relação foi ilustrada por segmentos pontilhados. Mesmo que tenham arestas de vinco incidentes, vértices irregulares não possuem pontos de tangência.

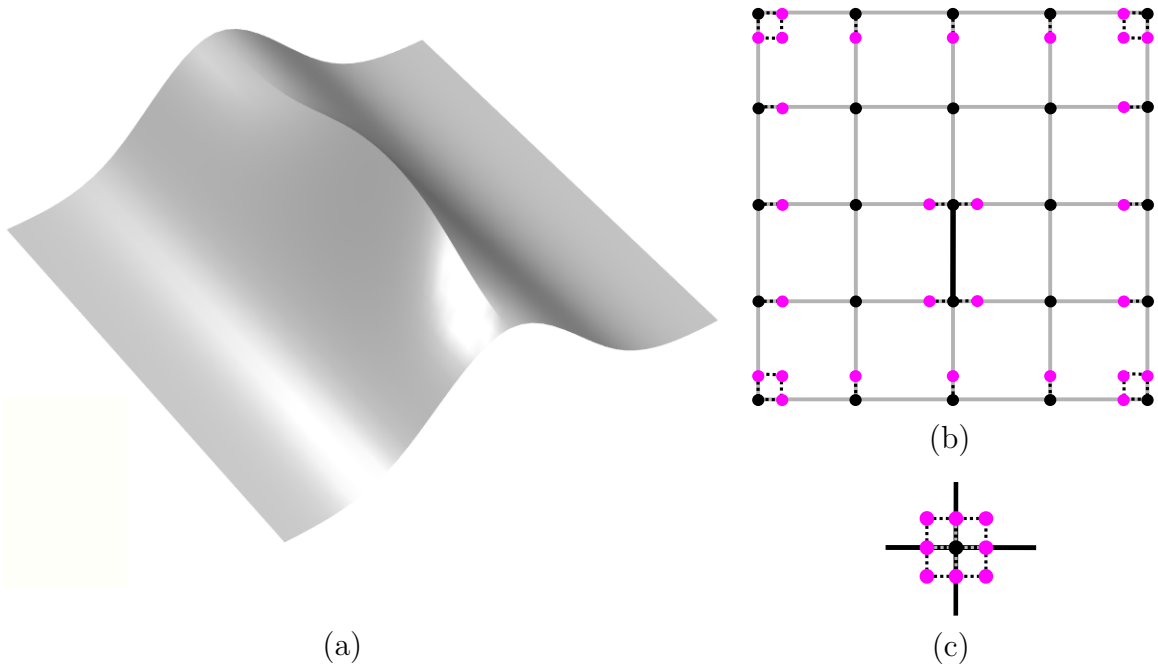


Figura 2.2: T-malha com arestas de vinco (adaptado de [23]). (a) Superfície com descontinuidade geométrica pela presença de arestas de vinco. (b) T-malha da superfície em (a), com arestas de vinco nas bordas e também na aresta de cor preta. Vértices são representados por círculos pretos e pontos de tangência na cor magenta. (c) Vértice regular com 4 arestas de vinco incidentes, resultando em 8 pontos de tangência, o número máximo para T-splines bicúbicas.

Como adotado por Peres [23], neste trabalho consideram-se as T-splines com função de forma de grau 3, ou bicúbicas, visto sua flexibilidade e compatibilidade com ferramentas CAD/CAM, como no caso do Autodesk Fusion 360 [1].

Apesar deste trabalho não tratar de análise isogeométrica, por ter sua motivação relacionada à IGA, faz-se necessário introduzir o conceito de T-spline *conforme para análise*, do inglês *analysis-suitable T-spline*, ou AST-spline. Conforme para análise diz respeito a flexibilidade de um modelo em representar geometrias arbitrárias, bem como a posse de propriedades matemáticas que assegurem análise robusta. AST-splines são definidas sobre T-malhas *admissíveis*, que então garantem tais propriedades por meio de restrições topológicas na T-malha, que resultam na manutenção de propriedades matemáticas básicas de NURBS em suas funções de forma, fazendo com que sejam todas não negativas e preservem partição de unidade e independência linear. Esse assunto é de suma importância para aplicabilidade de T-splines em IGA e métodos de análise, cuja extensa literatura continua a ser desenvolvida [17, 30, 3, 7, 35, 36].

A descrição da topologia de AST-splines necessita do conceito de *extensões de T-junção*. *Extensões de face* consistem em um ou dois segmentos de linha originários de uma T-junção na direção onde não há aresta até que no máximo duas arestas perpendiculares sejam atingidas. Há ainda a chamada *extensão de aresta*, compreendida pela aresta já existente na T-junção na direção oposta das extensões de face. As *extensões de face e de aresta*, juntas, compõem as extensões de uma T-junção. Chamamos de *primeira extensão de face*, o primeiro segmento das extensões de face de uma T-junção, isto é, aquele que alcança a face a qual a T-junção toca.

Os vértices resultantes da interseção das extensões de face com arestas da T-malha, *vértices de extensão*, são apenas topológicos, e portanto não possuem ponto de controle. Chamamos de *T-malha estendida* (ou *T-malha elementar*) a T-malha com adição de todas as suas extensões de T-junção. A Figura 2.3 apresenta a T-malha estendida da T-malha na Figura 2.1, evidenciando em linhas tracejadas as extensões de face.

Mediante a definição de T-malha estendida, torna-se possível detalhar as restrições topológicas aplicadas às malhas de AST-splines. Inicialmente, os primeiros estudos sobre o tema apresentavam restrições mais rígidas [6]; no entanto, esses requisitos foram progressivamente flexibilizados em trabalhos posteriores [17, 3, 35], conforme o conceito foi sendo aprimorado na literatura. Neste contexto, adota-se o conjunto de restrições definido por Scott et al. [30]. Assim, AST-splines são aquelas cuja T-malha estendida é dita *admissível*, e portanto respeita as seguintes restrições:

- Não existe interseção entre extensões de T-junção perpendiculares;
- Nenhuma primeira extensão de face alcança uma face na região 3-anel de face de um vértice extraordinário; e

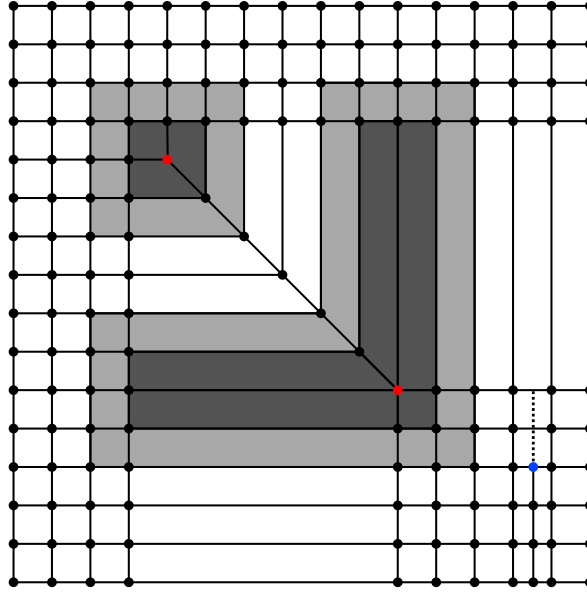


Figura 2.3: T-malha estendida, formada pela adição das extensões de face (destacadas com linhas tracejadas na cor preta).

- Nenhum vértice extraordinário incide em faces na região 3-disco de face de outro vértice extraordinário.

Uma face da T-malha pode estar associada a mais de um *elemento T-spline*, isto é, mais de um retalho de superfície T-spline. Essa característica, no entanto, não se replica para faces da T-malha estendida (ou T-malha elementar), as quais possuem relação um para um com elementos T-spline da superfície. Pelo restante do texto, utiliza-se de forma intercambiável os termos T-spline e AST-spline a menos que especificado o contrário, visto que o contexto deste trabalho envolve o uso de T-splines conformes para análise.

Com isso, tem-se que uma superfície T-spline é composta pela união de seus elementos T-spline. Sendo $\mathcal{M}_e$ a T-malha elementar, a superfície de um elemento e é dada por

$$S^e(u, v) = \sum_{i=1}^{n^e} R_i^e(u, v) P_i^e, \quad (2.1)$$

onde (u, v) são as coordenadas associadas ao domínio paramétrico do elemento $[-1, 1]$ e n^e é o número de funções de forma racionais, $R_i^e(u, v)$, bem como o número de pontos de controle, P_i^e , que influenciam o elemento e . A função $R_i^e(u, v)$ é definida como

$$R_i^e(u, v) = \frac{w_i^v N_i^e(u, v)}{W^e(u, v)} = \frac{w_i^v N_i^e(u, v)}{\sum_{j=1}^{n^e} w_j^v N_j^e(u, v)}, \quad (2.2)$$

em que $N_i^e(u, v)$ é a i -ésima função de forma polinomial e w_i^v é o i -ésimo peso, ambos influenciando o elemento e .

Apesar da flexibilidade apresentada por T-splines, fora de um contexto de CAD ou CAM, T-splines dificilmente são utilizadas sua forma natural. Aplicações de IGA [31, 30], algoritmos de renderização [28], dentre outros, desejam obter os pontos de controle, pesos e funções de forma que influenciam cada elemento T-spline da superfície. Para tanto, emprega-se uma técnica chamada extração de elementos de Bézier. Nesse contexto, elemento de Bézier é a representação de um elemento T-spline por uma superfície de Bézier. Em NURBS, a extração é realizada pela inserção repetida de nós em cada direção paramétrica, até que todos tenham multiplicidade igual ao grau da curva [5]. A presença de vértice extraordinários na T-malha não estruturada, assim como a característica local dos vetores de nós, tornam o processo mais complexo para T-splines.

O operador de extração de Bézier mapeia as funções de base polinomiais de *Bernstein* dos elementos de Bézier para as funções T-spline globais. Em outras palavras, a extração de Bézier determina a representação exata das funções de forma de T-spline para cada elemento e em termos de um conjunto de polinômios de Bernstein, $\mathbf{B}$. Esse mapeamento é dado por uma transformação linear $\mathbf{C}^e$, onde e é um elemento T-spline. Formalmente, essa transformação é definida por

$$\mathbf{N}^e = \mathbf{C}^e \mathbf{B}. \quad (2.3)$$

Ao transpor a matriz do operador de extração de Bézier, obtém-se uma matriz que mapeia cada um dos pontos de controle P_i^e que influenciam no elemento e (organizados matricialmente em $\mathbf{P}^e$) para os pontos de controle de Bézier em forma matricial, $\mathbf{Q}^e$. Essa transformação é dada por

$$\mathbf{Q}^e = \mathbf{C}^{e^T} \mathbf{P}^e. \quad (2.4)$$

A Figura 2.4 ilustra o uso do operador de extração de Bézier para o elemento e_3 de uma curva B-Spline, tanto para obter as funções de base polinomiais de Bernstein, quanto para obter os pontos de controle de Bézier.

Assim como adotado por Peres [23], as bases de polinômios de Bernstein unidimensionais serão definidas sob um intervalo paramétrico $-1 \leq u, v \leq 1$, com propósito de facilitar a quadratura gaussiana, conforme enunciado por Scott et al. [31]. As bases de polinômios de Bernstein bidimensionais de grau p são dadas pelo produto tensorial de duas bases de Bernstein unidimensionais, considerando u e v as coordenadas no plano paramétrico $[-1, 1] \times [-1, 1]$, tem-se:

$$B_{i,j}(u, v) = B_i(u)B_j(v), \quad 0 \leq i, j, \leq p. \quad (2.5)$$

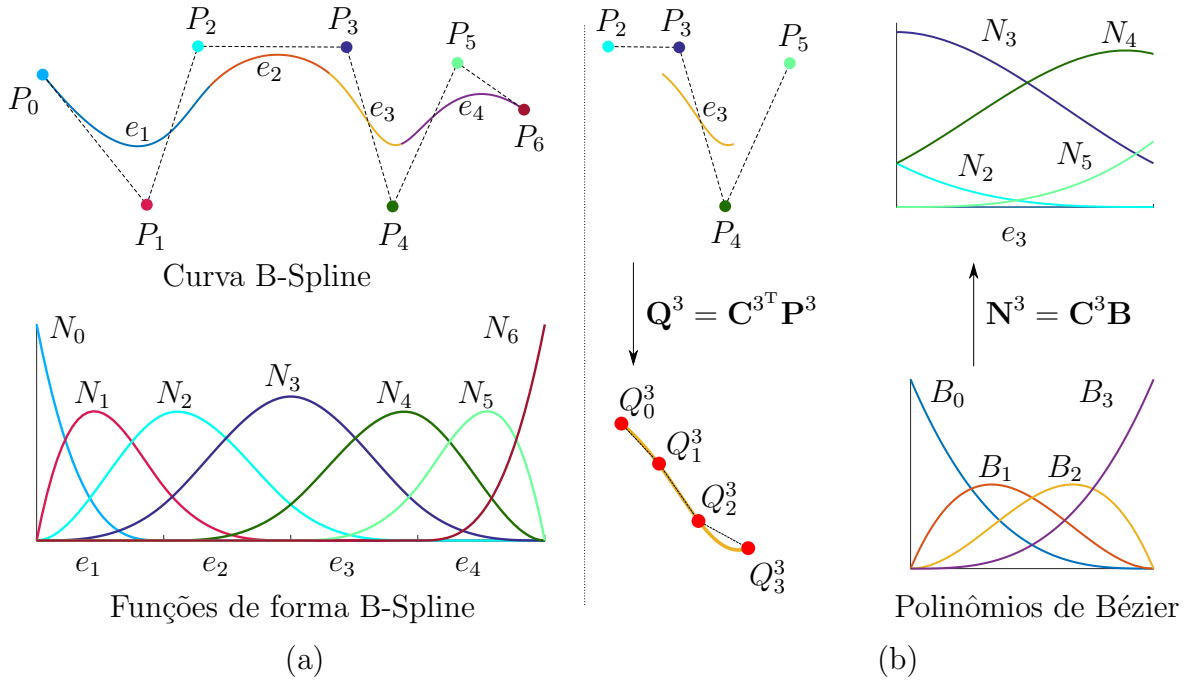


Figura 2.4: Extração de Bézier (adaptado de [23]). (a) Curva B-Spline com seus elementos destacados. A um ponto de controle P_i associa-se uma função de forma N_i , essa relação também é destacada pelas cores. (b) Mapeamento do elemento e_3 da B-Spline em (a) por meio do operador de extração de Bézier $\mathbf{C}^3$. Os pontos de controle e funções de forma de B-Spline que influenciam em e_3 são dados, em forma matricial, por $\mathbf{P}^3$ e $\mathbf{N}^3$, respectivamente.

A Figura 2.5 ilustra o caso onde $p = 3$, nesse caso tem-se 4 funções unidimensionais, que multiplicadas duas a duas totalizam 16 funções bidimensionais. Dessa maneira, considerando n^e a quantidade de funções de forma T-splines que influenciam no elemento e , a matriz de transformação $\mathbf{C}^e$ tem ordem $n^e \times 16$.

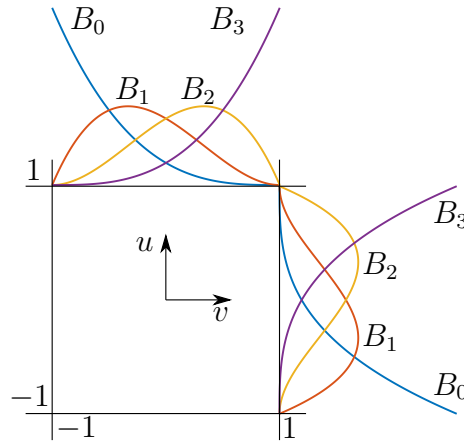


Figura 2.5: Funções de forma de Bernstein para elemento de Bézier bidimensional (adaptado de [23]).

Conforme comentado anteriormente, a presença de vértices extraordinários torna o processo de extração de Bézier mais complexo. Por isso, esse procedimento é separado em dois casos distintos, o primeiro trata de elementos provenientes de faces regulares, enquanto o segundo compreende os elementos associados a faces irregulares.

2.2.1 Extração de Bézier para faces regulares

A extração de Bézier para faces regulares necessita da definição de *ancoragem*. Âncoras são associações de vetores de nós locais com uma posição na T-malha. A localização de uma âncora na T-malha depende do grau da função T-spline, caso seja par, as âncoras situam-se no centro das faces. Caso seja ímpar, estão nos vértices. Juntar par e ímpar (em cada direção paramétrica) posiciona as âncoras nas arestas. Visto que este trabalho adota exclusivamente T-splines bicúbicas, isto é, aquelas as quais associam-se polinômios de grau três em cada direção paramétrica, as âncoras residem nos vértices. A pontos extraordinários não associam-se âncoras nem funções de forma de T-splines, já que tais pontos não respeitam as direções paramétricas da T-malha [30]. A extração de Bézier para faces pertencentes a vizinhança 2-anel de face de um ponto extraordinário é tratada na próxima seção. Uma discussão mais abrangente sobre âncoras é apresentada por Bazilevs et al. [2].

Exclusivamente em regiões regulares da T-malha, os pesos associados às arestas podem ser interpretados como intervalos nodais, isto é, a diferença dos valores nodais dos vértices que nela incidem. Com isso, os vetores de nós locais são inferidos da topologia da T-malha, observando os valores nas arestas. Tem-se uma valoração correta das arestas da T-malha, se e somente se, a soma dos intervalos nodais em lados opostos de cada face regular for igual. Assim, os vetores de nós definem o domínio paramétrico das âncoras associadas aos vértices, e conseqüentemente as funções de forma. Portanto, definidos os domínios paramétricos, é possível estabelecer quais funções de forma exercem influência (isto é, são não nulas) em cada um dos elementos da T-malha. A associação de vetores de nós locais com funções de forma é proveniente de PB-splines [33] (do inglês, *Point-Based Splines*), e é essa característica que confere às T-malhas a capacidade de refinamento local.

A Figura 2.6 representa um pedaço de uma T-malha estendida, de forma que a i -ésima aresta é dada por a_i e seu peso, ou nesse caso, intervalo nodal, é dado por w_i^a . Dizer que esta T-malha estendida tem configuração válida implica que $w_{42}^a = w_{43}^a + w_{44}^a$ e $w_{59}^a = w_{57}^a + w_{58}^a$.

Uma âncora, α , possui dois vetores de nós associados, chamados $\Xi_\alpha = \{U_\alpha, V_\alpha\}$, um para cada direção paramétrica (u e v respectivamente). O cômputo desses vetores de nós dependem de dois vetores de intervalos de nós, $\Delta\Xi_\alpha = \{\Delta U_\alpha, \Delta V_\alpha\}$, obtidos com os pesos das arestas na T-malha. Contando com as coordenadas paramétricas de uma âncora, o vetor ΔU_α é composto pelos intervalos nodais das $p+1 = 4$ arestas mais próximas na direção paramétrica u . No caso de T-junções, tomam-se os valores das extensões de face, que por sua vez são iguais aos intervalos nodais das arestas paralelas

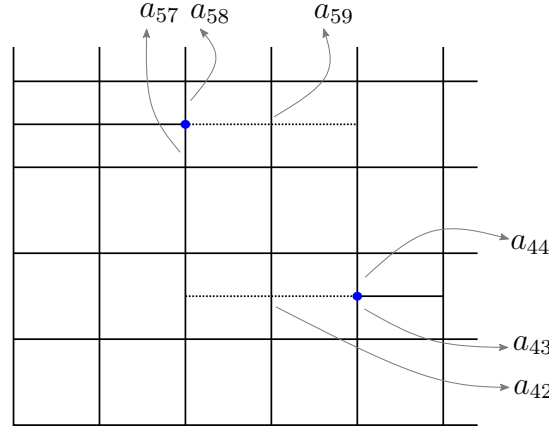


Figura 2.6: T-malha estendida: configuração válida com os intervalos nodais representados pelos pesos das arestas destacadas. T-junções são ilustradas por pontos azuis, enquanto as suas extensões de face são dadas por segmentos pontilhados.

a elas. Caso se trate da borda da T-malha, o intervalo nodal zero é utilizado. O vetor ΔV_α é obtido de maneira análoga.

Peres et al. [22] tratam a determinação de vetores de nós e funções de forma em T-malhas com arestas de vinco. Durante a determinação do vetor de intervalos de nós para uma âncora (ΔU_α , por exemplo), caso uma aresta de vinco seja encontrada, adiciona-se intervalo de nó nulo, que representa o intervalo nulo de um ponto de tangência até o vértice associado. Na determinação do vetor de nós locais de uma âncora associada a um vértice incidente em uma aresta de vinco perpendicular, o domínio daquela âncora é dividido em dois, acarretando em dois vetores de nós separados, um para cada lado da aresta de vinco. Este trabalho adota a extensão do método de extração de Bézier apresentada anteriormente. Mais detalhes são encontrados em [23, 22].

Em posse do vetor $\Delta U_\alpha = \{\Delta u_0, \Delta u_1, \Delta u_2, \Delta u_3\}$, considerando que $\Delta u_i = u_{i+1} - u_i$ para $i \geq 0$ e $u_0 = 0$, tem-se $U_\alpha = \{u_0, u_1, u_2, u_3, u_4\}$, o vetor de nós local de α na direção paramétrica u . Considera-se o mesmo procedimento para a direção paramétrica v . Por fim, o domínio paramétrico Ω_α é determinado pela combinação dos domínios dos vetores de nós U_α e V_α . A Figura 2.7 ilustra o processo para obter os vetores de nós locais para duas âncoras na T-malha estendida da Figura 2.6.

Até então, os vetores de nós obtidos podem ser *fechados*, isto é, o primeiro e último nó possuem multiplicidade menor que $p + 1$. Multiplicidade denota a quantidade de ocorrências de um nó em seu vetor de nós. Por conta disso, computa-se o vetor de nós estendido, $\bar{\Xi}_\alpha = \{\bar{U}_\alpha, \bar{V}_\alpha\}$, formado a partir de Ξ_α , porém, repetindo o primeiro e último nó até que tenham multiplicidade $p + 1$, em ambos $\bar{U}_\alpha$ e $\bar{V}_\alpha$. A adição desses nós não muda a parametrização da função associada à Ξ_α , mas inclui mais funções nesse espaço paramétrico. Por conta disso, faz-se necessário identificar exatamente

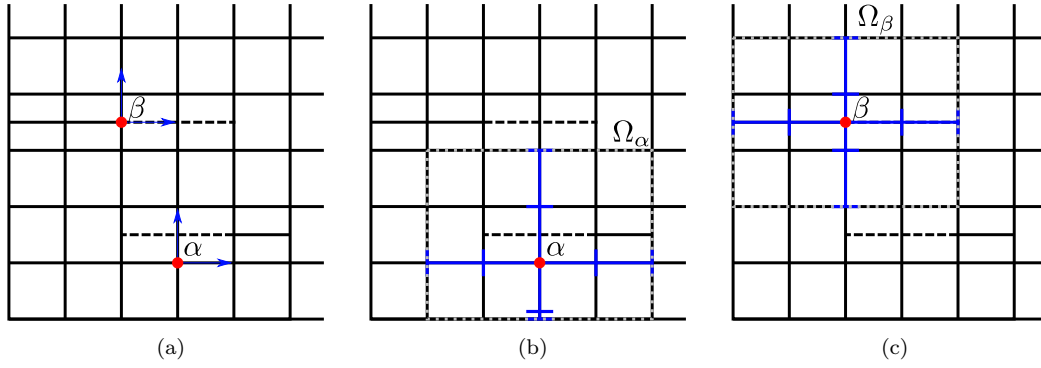


Figura 2.7: Cômputo dos vetores de nós locais (adaptado de [23]). (a) âncoras α e β (pontos vermelhos) juntamente de suas direções paramétricas (setas azuis); (b) e (c) processo para definir os vetores de intervalos de nós $\Delta\Xi_\alpha$ e $\Delta\Xi_\beta$ a partir dos intervalos nodais das arestas. Os espaços delimitados pelas linhas pontilhadas são os domínios paramétricos das funções de forma associadas às âncoras α e β , respectivamente.

qual função de forma está associada à âncora α (e conseqüentemente ao vetor Ξ_α). Na direção u , isso é alcançado pelo índice $n_{u,\alpha}$, que é igual a quantidade de nós adicionados no início de $\bar{U}_\alpha$. Emprega-se o mesmo raciocínio para a direção v .

A Figura 2.8 ilustra as funções cúbicas de T-spline (na direção v) das âncoras α e β sob seus vetores de nós estendidos. Na Figura 2.8(a) considera-se o vetor de nós $V_\alpha = \{0, 0, 1, 2, 3\}$ e vetor de nós estendido $\bar{V}_\alpha = \{0, 0, 0, 0, 1, 2, 3, 3, 3, 3\}$, o índice da função de forma de α é $n_{v,\alpha} = 2$, visto que 2 nós foram adicionados para formar um vetor aberto. Enquanto na Figura 2.8(b), o vetor de nós e vetor de nós estendido são $V_\beta = \{2, 3, 3.5, 4, 5\}$ e $\bar{V}_\beta = \{2, 2, 2, 2, 3, 3.5, 4, 5, 5, 5, 5\}$, respectivamente. Por fim, o índice da função de forma de β é $n_{v,\beta} = 3$.

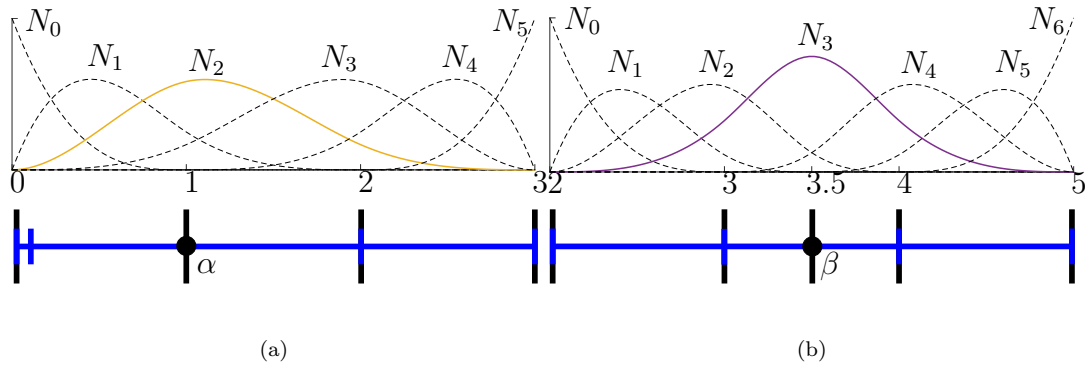


Figura 2.8: Funções de forma a partir de vetores de nós estendidos das âncoras α e β , respectivamente (adaptado de [23]).

Um elemento T-spline e geralmente é compreendido em domínios de diferentes âncoras, por conseqüência, essas são as âncoras cujas funções de forma influenciam nesse elemento. A Figura 2.9 ilustra um elemento e que está no domínio de múltiplas âncoras, dentre elas α , β e γ , destacando que as funções de forma associadas a essas âncoras pertencem a $\mathbf{N}^e$.

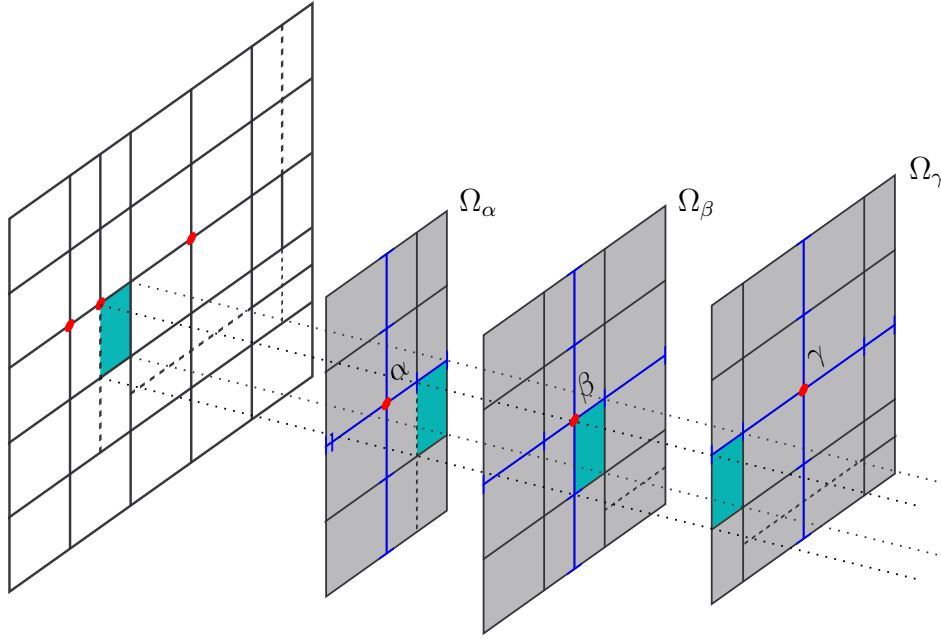


Figura 2.9: Relação de um elemento (em azul) com os domínios Ω_α , Ω_β , Ω_γ de funções de forma que influenciam nesse elemento (adaptado de [23]).

Conceitualmente, cada linha do operador de extração $\mathbf{C}^e$, associada a uma função de forma T-spline, é formada pela inserção de nós no vetor de nós estendidos, até que todos possuam multiplicidade $p + 1$, de maneira similar ao que ocorre com NURBS [5]. A título de ilustração, considerando que o vetor de nós estendido $\bar{V}_\beta$ fosse associado a uma curva NURBS, a extração de Bézier produziria as funções de forma expostas na Figura 2.10, nesse caso sendo as mesmas funções produzidas na extração da T-spline. Ao executar esse processo para todas as funções de T-splines cujos domínios contenham o elemento e , obtém-se a matriz de extração de elementos de Bézier $\mathbf{C}^e$ completa.

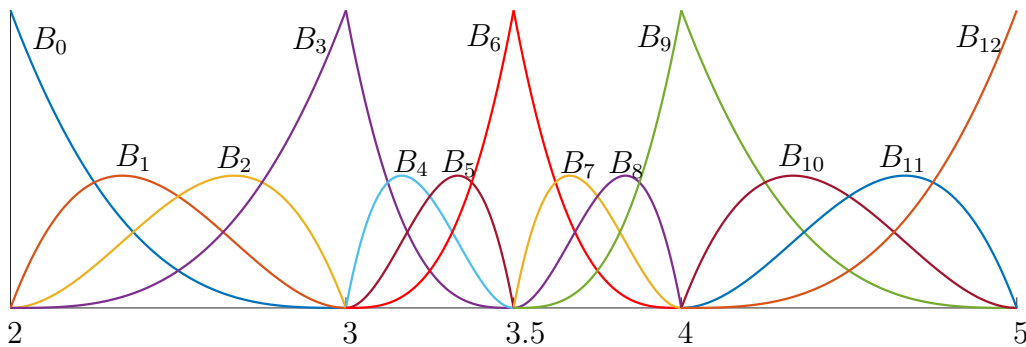


Figura 2.10: Funções de forma de Bézier após inserção de nós no vetor $\bar{V}_\beta$, proveniente da Figura 2.8(b).

Dessa maneira, a diferença encontra-se no fato de que a inserção de nós em NURBS é propagada na direção paramétrica perpendicular, o que necessita do cômputo completo do operador de extração. Já para T-splines, apenas uma linha do operador é computada para cada função de base de T-spline que exerce influência em um dado

elemento. O cômputo é feito pelo algoritmo apresentado por Scott et al. [31], com a adição proposta por Peres et al. [23], para tratar casos provenientes de arestas de vinco.

2.2.2 Extração de Bézier para faces irregulares

Como as T-malhas abordadas nesse trabalho são não estruturadas, isto é, permitem vértices extraordinários, passam a existir regiões em torno dos pontos extraordinários onde perde-se a possibilidade de definir direções paramétricas consistentes, como existe em regiões estruturadas. Neste caso, empregar o mesmo método de extração de Bézier para faces regulares acarretaria em buracos na superfície, conforme relatado por Toshniwal et al. [34]; com isso, faz-se necessário adotar um método de extração distinto. Na literatura, abordagens baseadas em retalhos de superfície são comumente empregadas [30, 34, 35, 36]. Elas operam por definir tratamento especial na determinação da forma de faces no m -disco de face de pontos extraordinários, a fim de contemplar restrições de continuidade desejáveis tanto para modelagem, quanto para análise.

A solução adotada neste trabalho foi proposta por Scott et al. [30], com a extensão de Peres et al. [22] para tratamento de arestas de vinco. Essa solução consiste na geração de um mapeamento linear entre as posições dos pontos de controle de um elemento da T-spline e as posições dos pontos de controle de um elemento de Bézier, dados em forma matricial por $\mathbf{P}$ e $\mathbf{Q}$, respectivamente. Esse mapeamento busca escrever os pontos em $\mathbf{Q}$ como combinações lineares dos pontos em $\mathbf{P}$. Os coeficientes dessas combinações são funções polinomiais racionais dos intervalos de nós das arestas incidentes no elemento. Os pontos em $\mathbf{Q}$ são divididos em pontos de face, pontos de aresta e pontos de vértice, $\mathbf{Q}^f$, $\mathbf{Q}^a$ e $\mathbf{Q}^v$, respectivamente. Enfatiza-se que todos os pontos de controle em $\mathbf{P}$ e $\mathbf{Q}$ pertencem ao espaço tridimensional euclidiano.

A Figura 2.11 compila a relação entre os pontos de controle da T-spline, $\mathbf{P}$, e os pontos de controle de Bézier, $\mathbf{Q}$. Na Figura 2.11(a) destaca-se a grade de pontos de controle de Bézier para uma face da T-spline. Mostra-se, na Figura 2.11(b), que os pontos de controle $\mathbf{Q}^f$ são determinados por combinações dos pontos de controle da T-spline, P_A , P_B , P_C , P_D . Esses pontos estão associados aos vértices da face do elemento e também aos intervalos de nós de a até f . Após determinação dos pontos $\mathbf{Q}^f$ é possível calcular $\mathbf{Q}^a$ e $\mathbf{Q}^v$, as relações desses pontos com os pontos de face estão ilustradas nas Figuras 2.11(c) e (d), respectivamente.

Assumindo $x = a + b + c$ e $y = d + e + f$, os pontos $\mathbf{Q}^f$ são descritos em forma matricial por na Equação (2.6)

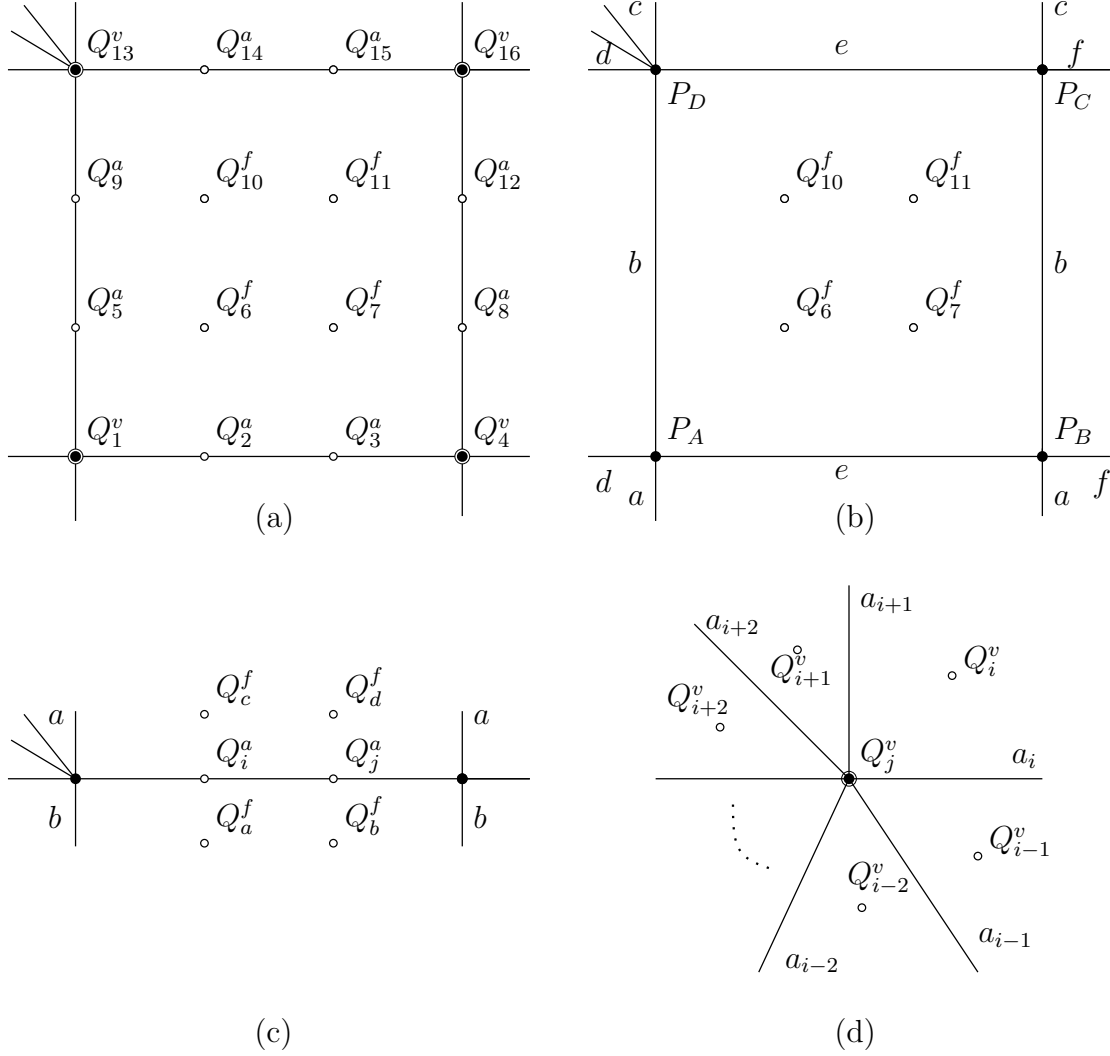


Figura 2.11: Pontos de controle em T-splines (adaptado de [30]). Q^v , Q^a e Q^f , representados por círculos vazados pretos, denotam pontos de controle dos elementos de Bézier, enquanto P_A , P_B , P_C e P_D (pontos pretos preenchidos) denotam pontos de controle da T-spline. Os pontos quando organizados matricialmente, formam os vetores $\mathbf{Q}^v$, $\mathbf{Q}^a$, $\mathbf{Q}^f$ e $\mathbf{P}$. $a, \dots, f$ denotam os pesos das arestas na T-malha.

$$\begin{bmatrix} Q_5^f \\ Q_6^f \\ Q_9^f \\ Q_{10}^f \end{bmatrix} = \begin{bmatrix} \left(\frac{b+c}{x} \frac{e+f}{y} \right) & \left(\frac{b+c}{x} \frac{d}{y} \right) & \left(\frac{a}{x} \frac{d}{y} \right) & \left(\frac{a}{x} \frac{e+f}{y} \right) \\ \left(\frac{b+c}{x} \frac{f}{y} \right) & \left(\frac{b+c}{x} \frac{d+e}{y} \right) & \left(\frac{a}{x} \frac{d+e}{y} \right) & \left(\frac{a}{x} \frac{f}{y} \right) \\ \left(\frac{c}{x} \frac{e+f}{y} \right) & \left(\frac{c}{x} \frac{d}{y} \right) & \left(\frac{a+b}{x} \frac{d}{y} \right) & \left(\frac{a+b}{x} \frac{e+f}{y} \right) \\ \left(\frac{c}{x} \frac{f}{y} \right) & \left(\frac{c}{x} \frac{d+e}{y} \right) & \left(\frac{a+b}{x} \frac{d+e}{y} \right) & \left(\frac{a+b}{x} \frac{f}{y} \right) \end{bmatrix} \begin{bmatrix} P_A \\ P_B \\ P_C \\ P_D \end{bmatrix}. \quad (2.6)$$

Os pontos de controle de arestas, Q_i^a e Q_j^a , presentes na Figura 2.11 são calculados na Equação (2.7) pela interpolação linear de pontos de face na qual a aresta a incide.

$$Q_i^a = \left(\frac{a}{a+b} \right) Q_a^f + \left(\frac{b}{a+b} \right) Q_d^f, \quad Q_j^a = \left(\frac{a}{a+b} \right) Q_b^f + \left(\frac{b}{a+b} \right) Q_c^f. \quad (2.7)$$

O cômputo de um vértice de elemento de Bézier sob um ponto de controle de um vértice v_j , ilustrado na Figura 2.11(d), é dado por,

$$Q_j^v = \begin{cases} \sum_{i=1}^{\mu_j} \frac{a_{i-1}a_{i+2}}{(a_{i-1}+a_{i+1})(a_i+a_{i+2})} Q_i^f & \text{se } \mu_j = 4, \\ \frac{1}{\mu_j} \sum_{i=1}^{\mu_j} Q_i^f & \text{caso contrário.} \end{cases} \quad (2.8)$$

No caso de pontos de arestas e vértices que estão na borda da vizinhança 2-anel de face de um vértice extraordinário, isto é, na interface de uma face irregular com uma face regular, utiliza-se a transposta do operador de extração de Bézier ($\mathbf{C}^T$), da Equação (2.4), para seu cômputo, ao invés das Equações (2.7) e (2.8).

A presença de arestas de vinco nas regiões irregulares solicita um tratamento distinto para determinar os pontos de controle de Bézier, de maneira que tanto continuidade quanto aspecto visual sejam contemplados. A Figura 2.12 ilustra o esquema apresentado em [22] para determinar pontos de controle de Bézier $\mathbf{Q}^e$ e $\mathbf{Q}^v$, quando arestas de vinco estão presentes. Tem-se, na Figura 2.12(a), que os pontos de controle de Bézier associados a arestas de vinco são determinados pela combinação da posição dos vértices v_a e v_b , incidentes na aresta, da seguinte forma

$$\begin{aligned} Q_i^a &= \left(\frac{d+e}{d+e+f} \right) P_A + \left(\frac{f}{d+e+f} \right) P_B, \\ Q_j^a &= \left(\frac{d}{d+e+f} \right) P_A + \left(\frac{e+f}{d+e+f} \right) P_B. \end{aligned} \quad (2.9)$$

Já na Figura 2.12(b) é posto o esquema utilizado para determinar o ponto de controle de Bézier associado a um vértice irregular v_a . Nele, considera-se o caso onde ζ_a — o número de arestas de vinco incidentes em v_a — é igual a dois, resultando em uma combinação dos pontos $\mathbf{Q}^e$ das arestas de vinco envolvidas. Matematicamente, essa relação é descrita por

$$Q_a^v = \frac{a}{a+b} Q_k^e + \frac{b}{a+b} Q_l^e. \quad (2.10)$$

Caso $\zeta_a > 2$, então

$$Q_a^v = P_a. \quad (2.11)$$

Caso contrário, Q_a^v é determinado pela Equação (2.8).

O desenvolvimento das Equações (2.7)-(2.11) permite a escrita de uma equação matricial que relaciona os pontos de controle de Bézier com os pontos de controle de T-spline sob um elemento e , por meio de uma matriz de fatores, chamada $\mathbf{F}^e$. Essa equação é

$$\mathbf{Q}^e = \mathbf{F}^e \mathbf{P}^e, \quad (2.12)$$

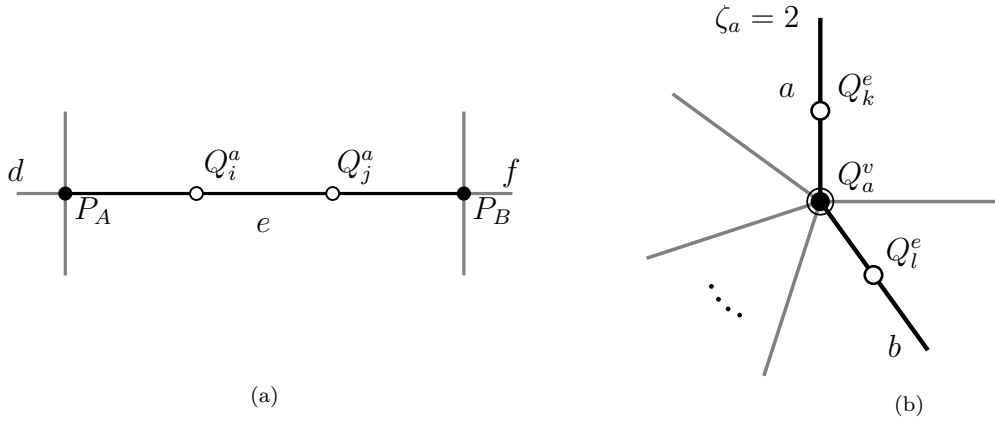


Figura 2.12: Extração de Bézier para faces irregulares com arestas de vinco (adaptado de [22]). (a) Pontos de controle de Bézier associados a arestas de vinco (linhas pretas) são calculados com base na posição dos vértices que incidem, e os pesos d , e e f . (b) Vértice irregular com exatamente duas arestas de vinco incidentes. A posição do ponto de controle de Bézier é computada com base nas posições dos pontos de controle de Bézier das arestas que incide, bem como os pesos a e b .

onde $\mathbf{Q}^e$ é o vetor que contém os 16 pontos de controle do elemento de Bézier e $\mathbf{P}^e$ é o vetor contendo os pontos de controle da T-spline utilizados na combinação linear. Vale ressaltar que substituindo a Equação (2.3) na Equação (2.12) determina-se que $\mathbf{C}^e = \mathbf{F}^e \mathbf{T}$.

2.2.3 Pontos de tangência

Recapitulando o exposto na Seção 2.2, pontos de tangência são tais quais outros pontos de controle da T-malha, possuem posição espacial e peso, associam-se a vértices regulares da T-malha por segmentos de intervalos nodais nulos.

A capacidade de representar vincos em T-splines existe desde sua formulação [33], entretanto, pouca ênfase se deu aos pontos que acompanham os pontos de controle incidentes nas arestas de vinco, tratando-os como pontos de controle ordinários. Uma forma mais moderna de suportar vincos em T-splines, adotada por exemplo, em ferramentas como Autodesk Fusion 360 [1] e trabalhos como os de Peres et al. [23, 22], impõe uma definição controlada da posição espacial de tais pontos de controle, nesse contexto, pontos de tangência. Isso é alcançado por determinar suas posições espaciais como combinação linear das posições espaciais de outros pontos de controle. Nessa conjuntura, os pontos de tangência são convenientemente denominados *pontos de tangência vinculados* (em inglês, *linked tangency handles*) ou LTHs.

Este trabalho incorpora a formulação de Peres et al. [22] para computar a posição espacial dos LTHs. A Figura 2.13 ilustra os casos usuais de LTHs, bem como os vértices, arestas e intervalos nodais envolvidos na determinação de suas posições. Observa-se na Figura 2.13(a) o caso onde um vértice v_A , de posição P_A , incide em uma aresta

de vinco, possui um LTH v_E posicionado sobre aresta a_u cuja direção paramétrica é perpendicular a da aresta de vinco, e também incide no vértice v_B posicionado em P_B . Os valores d_u e d_{u+} denotam os intervalos nodais de a_u e da aresta subsequente, na mesma direção paramétrica de v_A até v_B . Caso não haja tal aresta, em virtude de v_B estar na borda, por exemplo, considera-se $d_{u+} = d_u$. Assim, a posição P_E de v_E é dada por

$$\begin{aligned} P_E &= (2d_u + d_{u+})^{-1} [(d_u + d_{u+})P_A + d_u P_B] \\ &= (1 - f_u)P_A + f_u P_B, \end{aligned} \quad (2.13)$$

onde $f_u = d_u / (2d_u + d_{u+})$. A Equação (2.13) deve ser utilizada para calcular a posição dos LTHs para cada aresta incidente em v_A cuja direção paramétrica é perpendicular, isto é, para cada lado da aresta de vinco. Quando v_A incide em duas arestas de vinco de direções perpendiculares, além dos LTHs sobre as arestas, existe também um LTH na face compartilhada pelas duas arestas, cuja posição, neste caso, é determinada pela interpolação bilinear das posições dos vértices da face. Considerando v_A como o centro do sistema paramétrico local, como exposto na Figura 2.13(b), f_u e f_v como os pesos provenientes da aplicação da Equação (2.13) nas arestas a_u e a_v , respectivamente, tem-se que

$$P_F = g_A P_A + g_B P_B + g_C P_C + g_D P_D, \quad (2.14)$$

onde

$$\begin{aligned} g_A &= (1 - f_u)(1 - f_v) \\ g_B &= f_u(1 - f_v) \\ g_C &= f_u f_v \\ g_D &= (1 - f_u)f_v. \end{aligned} \quad (2.15)$$

A Figura 2.13(b) também explicita os LTHs associados ao vértice v_A , são eles $P_{E,u}$ e $P_{E,v}$ sobre arestas, e P_F sobre a face. Com o uso da Equação (2.13), adotando intervalos nodais unitários, as posições dos pontos de tangência sobre arestas são $P_{E,u} = (1 - f_u)P_A + f_u P_B$ e $P_{E,v} = (1 - f_v)P_A + f_v P_D$, onde $f_u = f_v = \frac{1}{3}$. Enquanto utilizando as Equações (2.14) e (2.15), obtém-se $P_F = \frac{4}{9}P_A + \frac{2}{9}P_B + \frac{1}{9}P_C + \frac{2}{9}P_D$.

É possível utilizar as Equações (2.13)-(2.15) para determinar uma transformação linear $\mathbf{L}^e$, responsável por mapear as posições dos pontos de controle que influenciam em um elemento de Bézier, e , para pontos de controle *reais*. Este termo busca remeter a todos os pontos de controle, exceto os LTHs. O uso de $\mathbf{L}^e$ na Equação (2.3) permite a remoção dos LTHs na T-malha elementar, e por consequência, nos elementos de Bézier. Em referência à Seção 1.2, o presente trabalho limita-se ao problema geométrico, e, portanto, o processo de remoção de LTHs não foi empregado. Peres et al. [22] apresentam

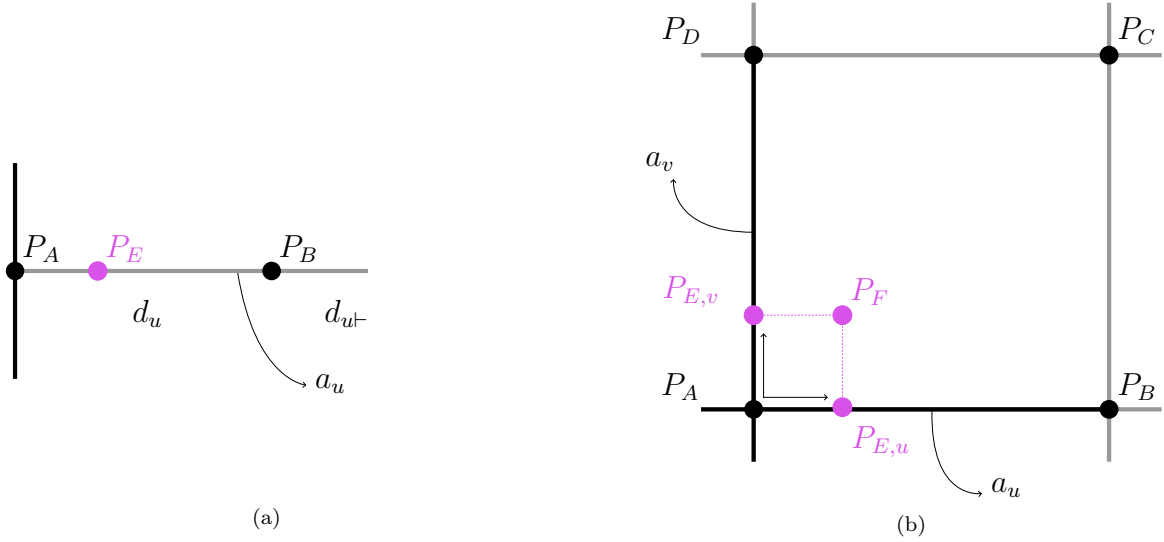


Figura 2.13: Pontos de tangência (círculos roxos) sobre arestas e faces de um vértice incidente em aresta de vinco, dadas por linhas pretas (adaptado de [22]). (a) A posição espacial do ponto de tangência associado a v_A sobre uma aresta perpendicular à aresta de vinco. (b) Posição dos pontos de tangência associados ao vértice v_A , incidente em duas arestas de vinco de direções perpendiculares. Os pontos de tangência são $P_{E,u}$ e $P_{E,v}$ sobre arestas a_u e a_v , respectivamente; e P_F sobre a face.

mais detalhes sobre a aplicação da transformação $\mathbf{L}^e$ para a remoção de LTHs, seguida da confecção do modelo de análise de IGA.

2.3 Refinamento local de T-splines

Conforme introduzido no Capítulo 1, um dos objetivos do trabalho envolve o projeto de um algoritmo capaz de refinar regiões de uma T-spline, para que seja possível empregar condições de contorno na superfície de maneira isolada.

O processo de refinamento local adotado baseia-se no exposto por Scott et al. [29].

De acordo com o enunciado anteriormente, um problema presente em NURBS é a inabilidade de refinamento local, desta forma, a inserção de um nó em uma superfície NURBS (abordada em [4, 24]) acarreta na inserção de diversos outros pontos de controle supérfluos. Esse problema é resolvido naturalmente por T-splines, visto a natureza local dos vetores de nós.

Nesta seção utilizam-se os conceitos enunciados na Subseção 2.2.1, como o cômputo de vetores de intervalos de nós e vetores de nós para vértices da T-malha. Considerando o contexto deste trabalho, o refinamento local discutido prevê que a T-spline resultante do refinamento seja também conforme para análise.

O refinamento local de faces irregulares, isto é, aquelas na vizinhança 2-disco de face de um ponto extraordinário, ainda é um assunto bastante estudado na literatura.

Com isso, este trabalho adota a mesma estratégia topológica apresentada por Toshniwal et al. [34], que em suma, define que se uma face irregular for refinada, todas as faces irregulares de sua região também devem ser. A esse esquema, denota-se refinamento uniforme de regiões irregulares, isto é, a subdivisão de faces irregulares em quatro. Abordagens similares encontram-se em [7, 18, 35]. Delega-se a apresentação do processo de refinamento mencionado para o Capítulo 3. Desta forma, o refinamento local aqui abordado limita-se a regiões regulares da T-malha.

O refinamento de T-splines necessita do conceito de *refinamento de função de base*, este é apresentado em seguida.

2.3.1 Inserção de nós na T-malha

Seja $N(u)$ uma função de forma de T-spline definida sob o vetor de nós $U = [u_0, u_1, u_2, u_3, u_4]$. Considere que deseja-se inserir um novo nó $\tilde{u}$ em U , obtendo o novo vetor de nós $\tilde{U} = [\tilde{u}_0, \tilde{u}_1, \tilde{u}_2, \tilde{u}_3, \tilde{u}_4, \tilde{u}_5]$, onde o índice i indica a posição de $\tilde{u}$ no vetor de nós, ou seja, $\tilde{u} = \tilde{u}_i$. Também considera-se que $N[k_0, \dots, k_m](k)$ denota a função de forma de B-spline unidirecional, $N(k)$, definida sob o vetor de nós $K = [k_0, \dots, k_m]$. Assim, de acordo com Sederberg et al. [32], tem-se que,

$$N(u) = a_i N[\tilde{u}_0, \dots, \tilde{u}_4](u) + b_i N[\tilde{u}_1, \dots, \tilde{u}_5](u), \quad (2.16)$$

onde os coeficientes a_i e b_i são dados, respectivamente, por

$$a_i = \begin{cases} 0 & \text{se } i = 0, \\ \frac{\tilde{u} - \tilde{u}_0}{\tilde{u}_3 - \tilde{u}_0} & \text{se } 1 \leq i \leq 3 \\ 1 & \text{se } 4 \leq i \leq 5, \end{cases} \quad (2.17)$$

e

$$b_i = \begin{cases} 1 & \text{se } 0 \leq i \leq 1 \\ \frac{\tilde{u}_4 - \tilde{u}}{\tilde{u}_4 - \tilde{u}_1} & \text{se } 1 \leq i \leq 4 \\ 0 & \text{se } i = 5. \end{cases} \quad (2.18)$$

Definidas essas relações, percebe-se que caso $i = 0$ ou $i = 5$, a função $N(u)$ não sofre alteração pela inserção do novo nó. Visto que nesses casos, a_i e b_i são zero e um ou vice-versa. Destaca-se que a inserção de um nó na T-malha acompanha a criação de uma nova função de forma da T-spline, com âncora situada no nó inserido.

Embora a explicação tenha sido feita sob a direção paramétrica u , a inserção na direção v realiza-se da mesma maneira. Além disso, a inserção de mais de um nó pode

ser alcançada pela repetição do mesmo método apresentado acima. A Figura 2.14 simula a inserção de um novo nó na T-malha, mostrando o efeito no domínio paramétrico de uma âncora existente. Ilustra-se, na Figura 2.14(a), o domínio paramétrico de uma função de forma associada a uma âncora α . Já na Figura 2.14(b) destaca-se a inserção de um novo nó, a ele associa-se uma nova âncora (ponto verde vazado) na T-malha, destaca-se também o domínio paramétrico da nova função de forma associada a essa âncora.

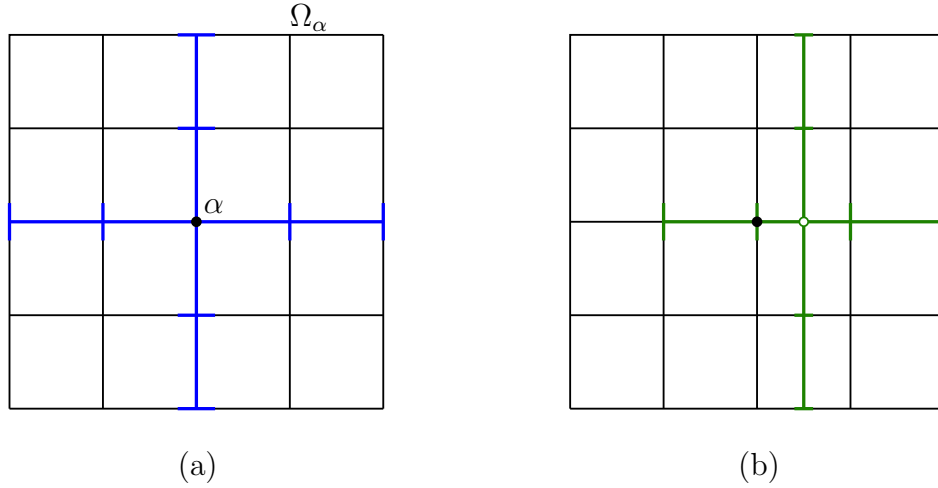


Figura 2.14: Inserção de nó na T-malha. (a) Âncora α (ponto preto) e seu domínio paramétrico (segmentos azuis). (b) Inserção de um novo nó, e posicionamento de uma âncora (ponto verde vazado) em seu lugar. O domínio da nova âncora é dado pelos segmentos em verde.

Embora a Figura 2.14 demonstre o resultado da inserção de nós nas T-malha, é importante frisar o efeito nas âncoras e funções de forma que consideram a aresta onde o novo nó foi inserido. Tal situação é abordada na Figura 2.15. Nela, observa-se uma T-malha em que as âncoras α , β , γ e θ , no cômputo de seus vetores de nós locais, passam pela aresta onde o novo nó $\tilde{u}$ foi inserido. Por isso, a adição de $\tilde{u}$ resulta na modificação das funções de forma dos vértices em vermelho, utilizando a Equação (2.16). Os coeficientes empregados nessa combinação linear das funções de forma são utilizados pelo algoritmo de refinamento local, abordado na Subseção 2.3.2.

2.3.2 Algoritmo de refinamento local

A inserção de nós na T-malha, descrita na Subseção 2.3.1, é um tipo de refinamento genérico, isto é, pode ser aplicado a quaisquer T-splines. Entretanto, é possível descrever um refinamento local mais robusto que leve em conta o contexto atual, que emprega não T-splines genéricas, mas AST-splines. Para o entendimento adequado do algoritmo de refinamento local, é necessário introduzir o conceito de *aninhamento de*

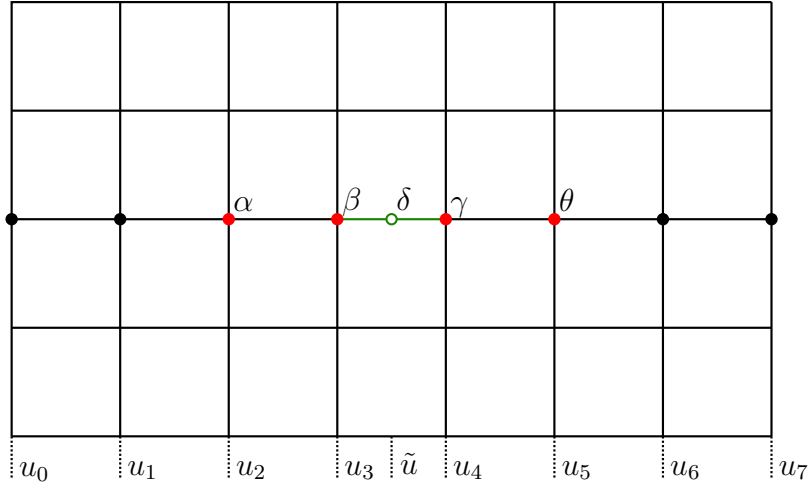


Figura 2.15: Efeito da inserção de nós nas funções de forma. A inserção de um nó $\tilde{u}$ modifica o domínio paramétrico das funções de forma associadas às âncoras α , β , γ e θ (pontos vermelhos). Retrata-se o novo vértice na T-malha com um ponto verde vazado, associado à âncora γ . Os valores $u_0, \dots, u_7$ compõem os intervalos nodais dos vértices da T-malha na direção paramétrica u .

espaços de T-spline. Esse resumo toma como base as definições de Sederberg et al. [29], e suas provas formais por Li et al. [16].

Um espaço de T-spline, dado por $\mathcal{T}$, consiste no conjunto de todas as T-splines que possuem mesma topologia e intervalos nodais na T-malha, $\mathcal{M}$. Se dois espaços de T-spline, $\mathcal{T}^1$ e $\mathcal{T}^2$, são ditos *aninhados*, $\mathcal{T}^1 \subseteq \mathcal{T}^2$, então $\mathcal{T}^2$ é um *refinamento local* de $\mathcal{T}^1$. Ressalta-se que a notação $\mathcal{M}^1 \subseteq \mathcal{M}^2$ indica que a T-malha $\mathcal{M}^2$ é obtida pela inserção de vértices e arestas em $\mathcal{M}^1$.

Da definição anterior, decorre que se dois espaços de T-spline são aninhados, isto é, $\mathcal{T}^1 \subseteq \mathcal{T}^2$, existe um mapeamento linear que descreve as funções de forma de $\mathcal{T}^1$ como combinação linear das funções de forma de $\mathcal{T}^2$. Os pesos envolvidos nesse mapeamento são provenientes da inserção de nós, descrita na Subseção 2.3.1, e podem ser organizados em uma matriz de transformação $\mathbf{M}$ de ordem $|\mathcal{V}^1| \times |\mathcal{V}^2|$. Onde $\mathcal{V}^1$ e $\mathcal{V}^2$ são os conjuntos de vértices das T-malhas $\mathcal{M}^1$ e $\mathcal{M}^2$, respectivamente. Denota-se $\mathbf{M}$ como *operador de refinamento local*.

Enquanto a matriz $\mathbf{M}$ realiza a transformação das funções de forma, sua transposta $\mathbf{M}^T$, mapeia a posição dos pontos de controle de $\mathcal{M}^1$ para $\mathcal{M}^2$, de forma que a T-spline não tenha nenhuma alteração geométrica. Sejam $\mathbf{P}^1$ e $\mathbf{P}^2$, em forma matricial, os pontos de controle de $\mathcal{M}^1$ e $\mathcal{M}^2$, respectivamente. Tal transformação é expressada por,

$$\mathbf{P}^2 = \mathbf{M}^T \mathbf{P}^1. \quad (2.19)$$

Com isso, define-se um algoritmo em que inserções de arestas e vértices em uma T-malha mantenham o aninhamento de espaços citado anteriormente, e portanto configure um *refinamento local*. Partindo de uma T-malha admissível, $\mathcal{M}_{ext}^1$, inserem-se as arestas desejadas, causando subdivisões nos elementos de T-spline e enfim gerando uma nova T-malha estendida. As novas arestas podem causar violações na T-malha, que por sua vez são resolvidas por inserções de outras arestas. Ao resolver todas as violações, o algoritmo termina. Dele obtém-se uma T-malha admissível, $\mathcal{M}_{ext}^2$, contendo as arestas inseridas inicialmente e que é um refinamento local de $\mathcal{M}_{ext}^1$. Por fim, se desejado, computa-se o operador de refinamento $\mathbf{M}^T$ para atualizar as posições dos pontos de controle, conservando a geometria [29].

O algoritmo permite que violações existam, momentaneamente, decorrentes da inserção de novas arestas. Tais violações são,

- V1** Cruzamento de extensões de T-junção em direções paramétricas perpendiculares;
- V2** Uma extensão de face em $\mathcal{M}_{ext}^2$ termina em um ponto interno de uma extensão de face em $\mathcal{M}_{ext}^1$.

A primeira violação diz respeito a conservação da conformidade para análise, enquanto a segunda, relaciona-se com o aninhamento de espaços de T-splines, que busca-se manter durante o refinamento. Em termos menos formais, a violação **V2** ocorre quando uma extensão de face é “encurtada” na malha refinada.

Geralmente, deseja-se que resolução das violações insira a menor quantidade de arestas possíveis. Nesta linha, Scott et al. [29] abordam o problema propondo uma estratégia gulosa para determinar quais arestas inserir. Para tanto, é empregado o *grafo de extensão acoplada* auxiliar que permite identificar as violações em uma T-malha estendida $\mathcal{M}_{ext}^2$, e então selecionar quais arestas inserir. Cada nó deste grafo corresponde ao conjunto de uma T-junção, v_i^t , e suas extensões, tanto de face quanto de aresta, enquanto traçam-se arestas entre nós que representem a violação **V1**. As extensões enquadradas na violação **V2** introduzem laços no grafo, já que afetam a própria extensão.

A Figura 2.16 destaca duas T-malhas estendidas juntamente com a construção do grafo de extensão acoplada para elas, permitindo determinar se existe aninhamento entre seus espaços de T-splines, de forma que seja uma um refinamento local da outra. Na Figura 2.16(a) tem-se uma T-malha admissível, $\mathcal{M}_{ext}^1$, de uma AST-spline. Já na Figura 2.16(b) é mostrada outra T-malha estendida, $\mathcal{M}_{ext}^2$, que desta vez não é admissível. Uma sobreposição das duas T-malhas é apresentada na Figura 2.16(c), evidenciando a violação **V2** na T-junção v_3^t , que pode ser constatada visualmente por

uma extensão de face de $\mathcal{M}_{ext}^1$ que ficou visível na sobreposição das malhas (linha vermelha pontilhada dentro de um retângulo cinza). Por fim, na Figura 2.16(d), traz-se o grafo resultante desse exemplo, ressaltando que $\mathcal{M}_{ext}^2$ não é admissível, por conta da existência de arestas no grafo, e não é um refinamento local de $\mathcal{M}_{ext}^1$, visto que existe um laço no nó associado a v_3^t .

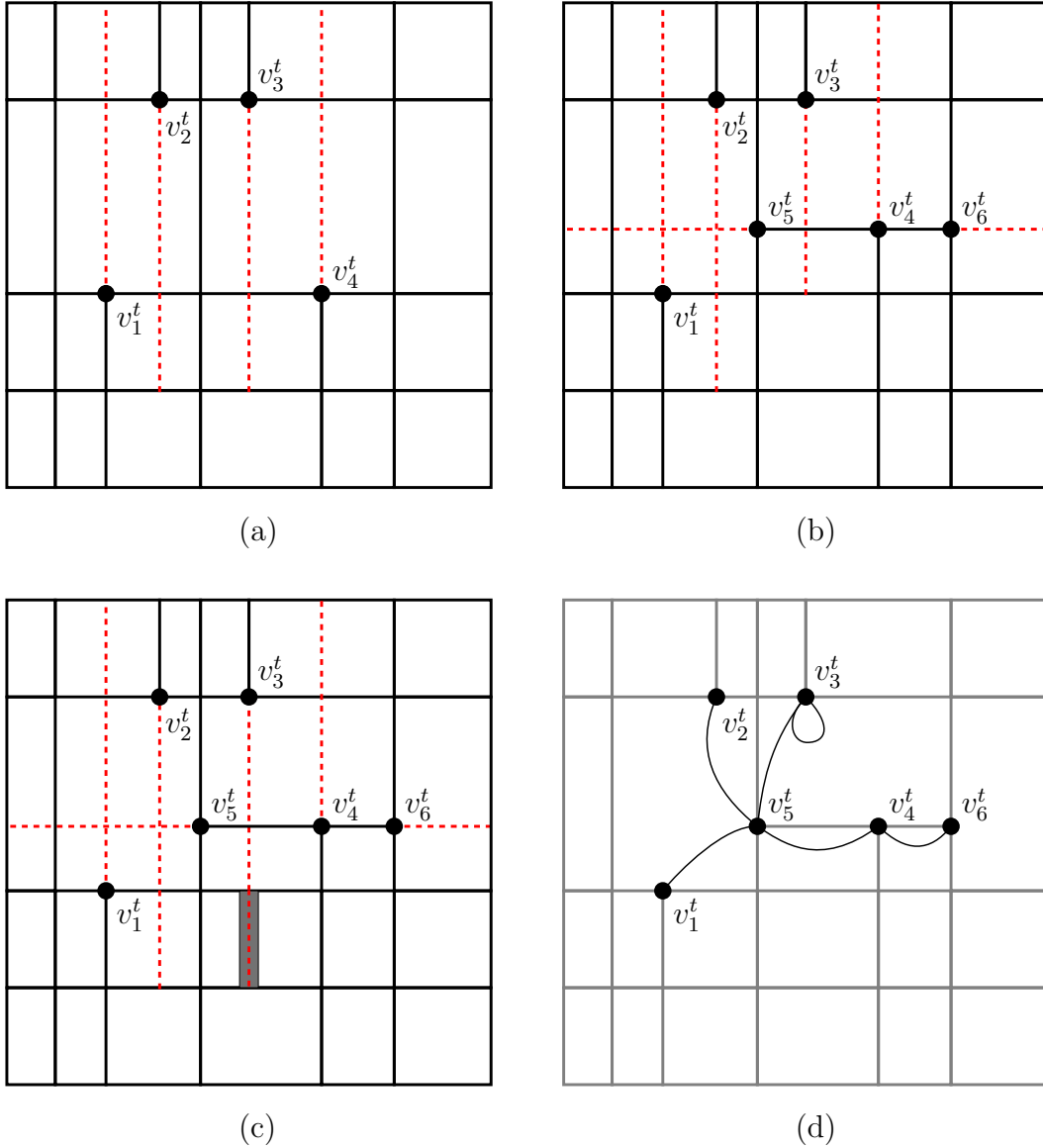


Figura 2.16: T-malhas estendidas e a geração do grafo de extensão acoplada (adaptado de [29]). Destacam-se com círculos pretos as T-junções, enquanto suas extensões de face são linhas vermelhas pontilhadas. (a) Uma T-malha estendida $\mathcal{M}_{ext}^1$ admissível. (b) Uma T-malha estendida $\mathcal{M}_{ext}^2$ que não é admissível, visto que existem diversos cruzamentos entre extensões de T-junção perpendiculares. (c) Uma vista sobreposta das T-malhas $\mathcal{M}_{ext}^2$ sobre a $\mathcal{M}_{ext}^1$, evidenciando com um retângulo cinza uma extensão de face que não foi contemplada. (d) Grafo de extensão acoplada de $\mathcal{M}_{ext}^2$.

Em posse do referido grafo de extensão acoplada, enquanto houverem violações, ou seja, enquanto o grafo possuir arestas, insere-se uma aresta candidata por vez, então atualiza-se o grafo, já que a inserção pode adicionar novas violações. Aresta candidata

nada mais é do que a aresta equivalente à primeira extensão de face da T-junção. Desta forma, tem-se uma aresta candidata por nó do grafo. A escolha da aresta candidata é feita de maneira a minimizar a quantidade de arestas do grafo a cada repetição. Em outras palavras, escolhe-se a aresta candidata cujo nó tem o maior número de arestas. Cessadas as violações, obtém-se uma T-malha estendida admissível $\mathcal{M}_{ext}^2$, que é um refinamento local de $\mathcal{M}_{ext}^1$.

2.4 HBDS

A estrutura de dados adotada para representar T-malhas é a *Handlebody Data Structure* (HBDS), ela é capaz de representar e manipular malhas com ou sem borda e consiste em uma hierarquia de componentes topológicos, além de ser dotada de operadores topológicos (detalhados na Subseção 2.4.2) que a modificam enquanto mantém a integridade topológica. Essas modificações são realizadas por meio de operadores de alto nível chamados de *glue* e *unglue*, esses por sua vez baseiam-se nos *Operadores de Morse* [20], compreendidos pela Teoria *Handlebody*.

Informalmente, a topologia de uma T-malha diz respeito a conectividade de seus *componentes topológicos*. Vértices, arestas e faces são componentes topológicos de dimensionalidade 0, 1 e 2, respectivamente. Assim, essa topologia da T-malha é dada pela conexão desses componentes de maneira a formar as faces que constituem a T-malha. Vale ressaltar que a topologia independe de aspectos geométricos, como a posição espacial dos vértices, por exemplo.

Levando em conta o escopo geométrico desse trabalho, definições formais acerca de topologia, operadores topológicos e integridade topológica estão fora do escopo deste trabalho, e portanto são omitidos, essas definições são provenientes da Teoria de Euler-Poincaré [27].

2.4.1 Componentes topológicos da HBDS

No caso da HBDS, os componentes topológicos provêm da estrutura de dados *Handle-Edge* [8], uma extensão da estrutura de dados *Half-Edge* [19]. Dessa representação, em adição aos componentes topológicos da T-malha, adotam-se os seguintes componentes:

- *Aresta* – componente que relaciona dois vértices bem como duas semi-arestas, segmentos orientados que podem ser compreendidos como as duas direções as

quais a aresta pode seguir. Caso uma das semi-arestas não esteja associada a uma face, trata-se de uma aresta de borda;

- *Semi-aresta* – componente de uma aresta no qual existe orientação, a ela associa-se um vértice, chamado de vértice de origem;
- *Curva de borda* – ciclo de semi-arestas que delimitam uma borda na malha.

A Figura 2.17 apresenta exemplos malhas HBDS, evidenciando os componentes topológicos citados acima. Na Figura 2.17(a) tem-se uma face interior, isto é, que não tem nenhuma aresta na borda. Enquanto na Figura 2.17(b) destaca-se uma face na borda, que tem duas arestas de borda (em vermelho), as semi-arestas em vermelho não possuem face associada. Vale ressaltar que a orientação adotada para as faces utiliza a regra da mão direita, como destacado na Figura 2.17(a).

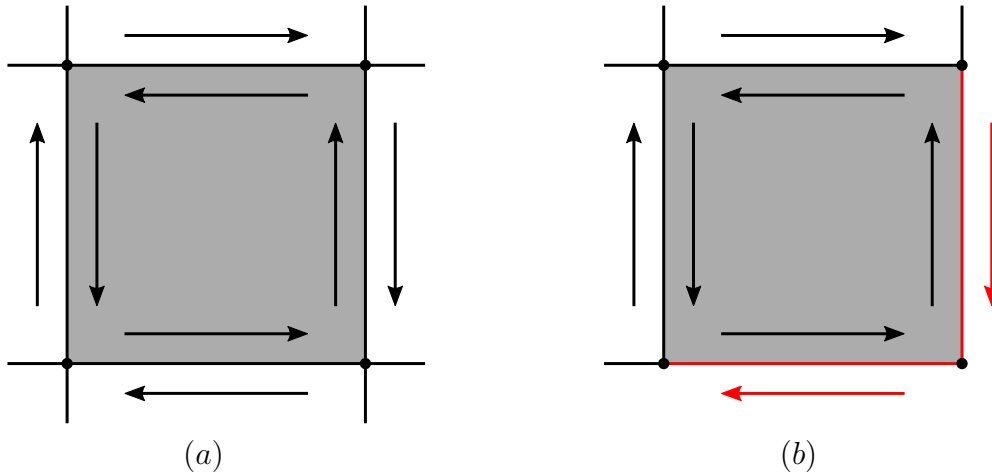


Figura 2.17: Componentes topológicos da estrutura de dados HBDS. Em cinza temos faces da malha, delimitadas por arestas (linhas pretas), que por sua vez conectam dois vértices (círculos pretos). Arestas se associam a até duas semi-arestas, que possuem uma direção e podem estar associadas a uma face. (a) Uma face interna, ressaltando a sequência de semi-arestas em seu interior seguindo a regra da mão direita. (b) Uma face na borda, destacando em vermelho as arestas e semi-arestas na borda.

Para essa representação, uma implementação computacional utilizando lista ligada mostra-se adequada, pois permite atravessamento de faces e bordas, dadas por ciclos de semi-arestas, de maneira especialmente eficiente. Além disso, torna-se possível implementar consultas nessas malhas, como vizinhança de vértices, arestas ou faces. Essas consultas são essenciais em inúmeras situações, notadamente, a determinação de vetores de nós locais, faces irregulares e também o algoritmo de refinamento local, definido no Capítulo 3.

2.4.2 Operadores topológicos

Partindo da definição informal de topologia da Seção 2.4, entende-se operador topológico como aquele que altera a topologia da malha conservando sua integridade topológica, isto é, modifica a conectividade de seus componentes topológicos independente de quaisquer imposições geométricas. Na teoria Handlebody [19], essas alterações são feitas por meio da ligação de *handles* à borda uma malha. Nota-se que não há como ligar nenhum *handle* à uma malha fechada, visto que ela não possui borda. A definição formal de *handles*, da maneira como são ligados às malhas e integridade topológica, necessitam de conceitos de topologia provenientes da Teoria de Euler-Poincaré [8, 15, 19, 20, 27] que não estão no escopo deste trabalho, e portanto não são apresentados.

Para a HBDS, a ligação dos *handles* foi classificada em casos e generalizada em duas operações de alto nível, chamadas *glue* e *unglue*, uma sendo a inversa da outra. A primeira liga duas arestas de borda formando uma aresta interna, enquanto a segunda toma uma aresta interna e a separa em duas arestas de borda. Destaca-se que as operações de *glue* só podem ser realizadas caso a orientação possa ser preservada. A Figura 2.18 ilustra alguns casos de uso do operador *glue*, o lado esquerdo compreende a configuração anterior e o lado direito após o *glue*, em todos os exemplos o *glue* é aplicado nas arestas em vermelho. O efeito do *unglue* pode ser observado considerando as figuras da direita para esquerda. Destacam-se, na Figura 2.18(a), duas faces com orientações compatíveis (pelas semi-arestas, flechas em preto), o operador *glue* só pode ser aplicado quando a orientação das faces é compatível. Na Figura 2.18(b) destaca-se o *glue* de duas arestas pertencentes a mesma borda, o *glue* resulta em duas bordas, uma interna a malha e outra na região externa. Por fim, na Figura 2.18(c) é ilustrado o *glue* de duas arestas que compartilham um vértice.

Enquanto operadores os operadores de *glue* e *unglue* comandam a junção ou separação de faces, é possível também manipular os componente topológicos para operações de refinamento, isto é, de forma a incluir componentes topológicos na malha. Este processo geralmente é expressado por uma subdivisão de aresta ou face, que por diminuir a generalidade da operação, permite simplificar seu funcionamento. A Figura 2.19 expõe os casos referidos. Nela, ilustra-se a inserção de um vértice V no interior de uma face da HBDS, são destacadas as alterações necessárias para manter a consistência da estrutura, como a criação de arestas e semi-arestas. Iniciando na Figura 2.19(a) ilustra-se a face na qual deseja-se inserir o vértice V e suas semi-arestas, para deixar clara a configuração inicial da face. Na Figura 2.19(b) identifica-se o próprio vértice V e também dois vértices de suporte que serão inseridos em arestas

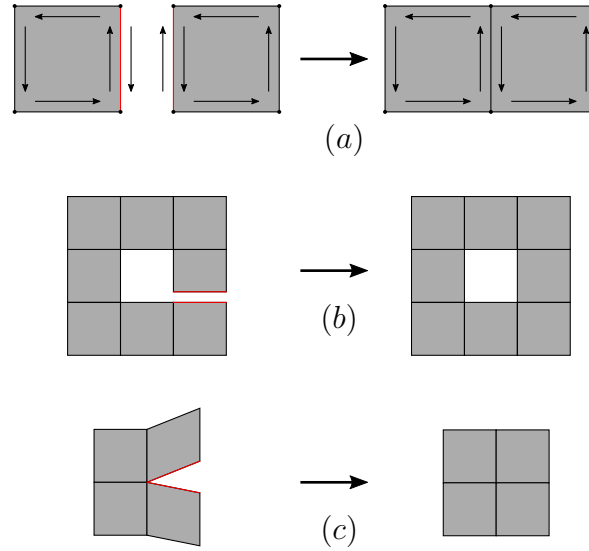


Figura 2.18: Aplicação do operador *glue*. (a) Ligação de duas arestas em malhas diferentes, juntando as duas malhas. (b) *Glue* de duas arestas que não compartilham nenhum vértice, pertencentes a mesma malha. O resultado é a criação de uma nova borda. (c) *Glue* de duas arestas que compartilham um vértice.

opostas da face, V_a e V_b respectivamente. A adição dos vértices V_a e V_b e de semi-arestas que eles incidem é ilustrada nas Figuras 2.19(c) e (d), respectivamente. Demonstra-se na Figura 2.19(e) a inserção de uma nova aresta que divide a face em duas, sendo importante notar que as semi-arestas que a acompanham devem manter a orientação da face. Por fim, na Figura 2.19(f) tem-se a adição de V na face.

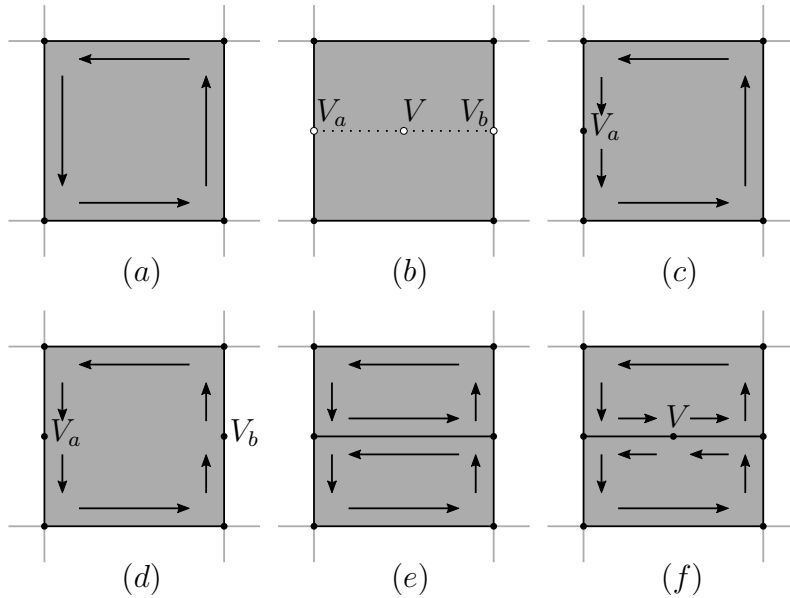


Figura 2.19: Inserção de vértice em malha HBDS. (a) Configuração inicial de uma face da malha, ilustrando também suas semi-arestas. (b) Deseja-se inserir o vértice V no centro da face, mas como vértices só podem ser inseridos sob arestas, é necessário inserir a aresta destacada com linha pontilhada. Para dar suporte a essa aresta, é necessário inserir dois vértices para ficarem nas pontas dessa aresta, notadamente V_a e V_b . (c) Inserção do primeiro vértice, V_a , e alteração nas semi-arestas. (d) Inserção de V_b , alteração nas semi-arestas. (e) Inserção da aresta destacada em (b). (f) Por fim, inserção do vértice V .

2.5 Considerações finais

Neste capítulo foram expostos os conceitos e teorias que fundamentam este trabalho, superfícies representadas por T-splines não estruturadas, AST-splines, extração de elementos de Bézier, refinamento local de T-splines e uma estrutura de dados para representar e manipular as T-malhas (HBDS).

A discussão iniciou-se pela apresentação de T-splines, representação de superfícies paramétricas que generaliza NURBS, permite refinamento local e também gera superfícies herméticas, indicando a inexistência de frestas nas superfícies T-splines. Foram definidos os elementos constituintes de uma T-spline, vértices ligados por arestas formando uma malha de faces quadrilaterais, a T-malha, que permite T-junções, vértices associados a arestas formando a forma da letra “T”. Então foi abordada a característica da T-malha que a torna não estruturada, a existência de vértices extraordinários, os vértices com valência diferente de 4 que não são T-junções. Vértices extraordinários criam uma região de faces irregulares em sua vizinhança 2-disco de face, o restante das faces são regulares. Tratou-se então dos pontos de tangência e arestas de vinco, aquelas que geram descontinuidade do vetor tangente da superfície. Na T-malha, pontos de tangência são associados aos vértices incidentes nas arestas de vinco por meio de arestas com peso nulo. Em sequência foi abordado o conceito de T-spline conforme para análise (ou AST-spline), que leva à apresentação da T-malha estendida, uma T-malha acrescida de extensões de aresta e de face de todas as suas T-junções. Discutiu-se como a T-malha estendida traz uma associação um para um dos elementos de T-spline para com os elementos de Bézier, bem como uma maneira de determinar esses elementos de Bézier, via operador de extração de Bézier.

Com isso, foi apresentada a determinação do operador de extração de Bézier para faces regulares. Para tanto, foram introduzidos os conceitos de âncora, vetor de intervalos de nós, vetor de nós locais e domínio paramétrico de âncoras. Esses, por sua vez são combinados para determinar as linhas do operador de extração de Bézier. Em seguida, foi apresentada a determinação do operador de extração de Bézier para faces irregulares, onde a presença de vértices extraordinários impede a determinação de um sistema de coordenadas paramétricas, e portanto, a determinação dos elementos de Bézier é realizada pela transformação dos pontos de controle de Bézier das faces irregulares.

Na seção seguinte, discutiu-se sobre refinamento local de T-splines, processo generalizado a partir da inserção de nós na T-malha. O algoritmo de refinamento local permite inserir arestas na T-malha de maneira que a geometria não sofra alteração.

Além de solucionar possíveis violações nas restrições impostas às AST-splines por meio de outras inserções de arestas. Enquanto para regiões irregulares, pontuou-se que o refinamento deve ser feito de maneira uniforme, isto é, todas as faces de uma região irregular devem ser subdivididas ao mesmo tempo.

Na seção seguinte, discutiu-se a estrutura de dados topológica empregada para representar T-malhas não estruturadas, bem como manipulá-la, essa é a HBDS (*Handlebody Data Structure*). Destacou-se a origem da HBDS, a junção da teoria Handlebody com a estrutura de dados *Handle-edge*, e também seus componentes topológicos, vértices, arestas, semi-arestas, faces e bordas. Proveniente da teoria Handlebody, evidenciou-se os operadores topológicos *glue*, *unglue*, e exemplos de subdivisões dos elementos topológicos, além de como eles são utilizados para manipular a estrutura.

Capítulo 3

Desenvolvimento

3.1 Considerações iniciais

Neste capítulo apresenta-se a implementação orientada a objetos desenvolvida, que partindo das formulações e algoritmos apresentados anteriormente, resulta em uma aplicação gráfica interativa de manipulação e edição de T-splines, constituindo uma contribuição tecnológica deste trabalho. Especifica-se também, a medida que forem necessários, aspectos de implementação, como classes e suas hierarquias. A aplicação gráfica desenvolvida é baseada em OpenGL e foi implementada contando com a base de código da biblioteca Ds [21], que por sua vez utiliza DearImGui [10] para produção de interfaces gráficas do usuário. Ao iniciar a aplicação, depara-se com uma janela, com a renderização de uma cena, vista a partir de uma câmera virtual. Cenas são objetos de uma classe da implementação chamada *Scene*, e são formados por atores e luzes, os quais possuem posição espacial, rotação e escala. A Figura 3.1 ilustra a janela da aplicação com a T-spline de um torus, juntamente de sua T-malha. Nas laterais da interface, encontram-se os controles de câmera, de cena e inspetor de comandos, todos dos quais serão apresentados neste capítulo.

O restante do capítulo é organizado como segue. A Seção 3.2 introduz o ponto inicial de uso da aplicação, a criação ou importação de T-malhas. O processo de geração de T-spline até a renderização em tela são descritos na Seção 3.3. Na sequência, são apresentadas na Seção 3.4 as funcionalidades implementadas, pelos quais as T-splines da aplicação são editadas ou refinadas. Por fim, apresenta-se na Seção 3.5 a principal contribuição deste trabalho, um algoritmo para resolução de inconsistências nas T-malhas que não necessita de estruturas de dados adicionais. De forma que após manipulações as T-malhas permaneçam admissíveis, e portanto gerem T-splines conformes para análise.

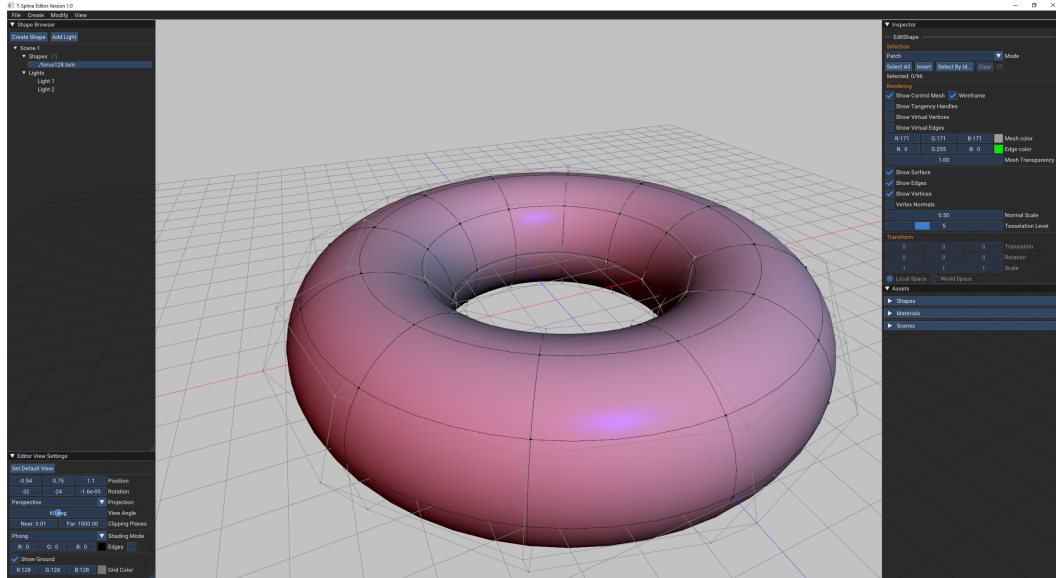


Figura 3.1: Interface completa da aplicação gráfica interativa. Contando com o exemplo de um torus, colorido pela inserção de luzes na cena.

3.2 Criação e importação de T-malhas

Para efetivamente utilizar a aplicação, precisamos de T-malhas, que podem ser ou produzidas pela própria aplicação, ou importadas de ferramentas CAD. A aplicação desenvolvida é capaz de gerar T-malhas para formas mais simples, são elas: plano, cubo, cilindro aberto, *quadball* e torus. Por sua vez, a importação de malhas mais complexas, isto é, aquela que demandam maior robustez na sua modelagem, toma como entrada arquivos no formato TSM (*T-Spline Mesh*) gerados por T-splines modeladas na aplicação de CAD Autodesk Fusion 360 [1], utilizada com licença educacional. Tanto a exportação da T-malha da ferramenta CAD, quanto o *parser* de arquivos TSM foram adotados com base no arcabouço de Peres [23]. Mais detalhes sobre o formato do arquivo TSM encontram-se no mesmo trabalho.

T-malhas na implementação são representadas pela classe *TMesh*, que possui uma malha HBDS subjacente, representando sua topologia. HBDS permite que seus componentes topológicos, identificados na Seção 2.4, sejam parametrizados com tipos definidos por quem visa utilizar a estrutura. Em termos gerais, isso é alcançado com uso de metaprogramação com modelos de classes em C++. A relação entre essas classes é apresentada na Figura 3.2.

A classe *TMeshTraits* empregada no modelo de classe *Complex* (que é a malha HBDS), visa decorar a topologia com informações suficientes para sua interpretação como uma T-malha. Os vértices carregam em *TVertex* informação do seu tipo, que pode ser um de: regular, extraordinário, T-junção ou L-junção. Os três primeiros fazem

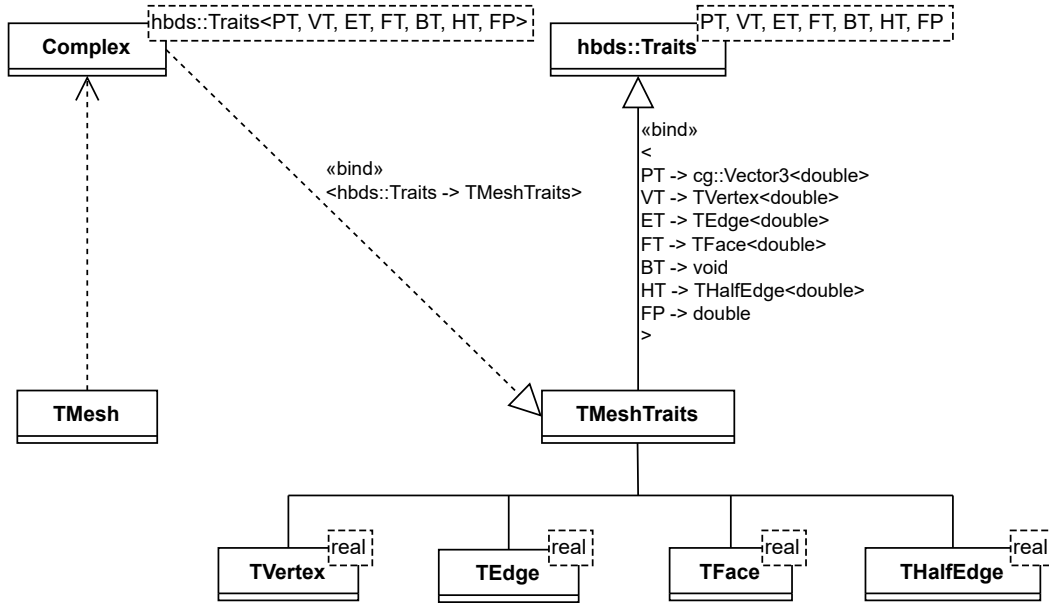


Figura 3.2: Diagrama UML das classes envolvidas com a representação da classe *TMesh*, que modela a T-malha. *Complex* é o modelo de classe C++ que representa a malha HBDS.

referência direta aos tipos apresentados no Capítulo 2, enquanto L-junção trata-se de um vértice de valência dois que está no canto de uma malha aberta. Além do tipo, também encontram-se nos vértices indicações se são virtuais ou pontos de tangência, bem como um vetor de pontos de tangência. Componentes virtuais e detalhes sobre pontos de tangência encontram-se na Seção 3.3. Arestas armazenam em *TEdge* seu peso e booleanos que indicam se são arestas de vinco ou virtuais. Enfim, faces armazenam em *TFace* os vértices que influenciam em sua forma e a sua matriz de extração de Bézier local, ambos explicados anteriormente, além de possuírem um estado que representa se são regulares ou irregulares.

3.2.1 Quadball

Dos modelos citados anteriormente, destaca-se a T-malha de uma *quadball*, uma forma que aproxima a esfera, mas cujo modelo emprega apenas faces quadrilaterais, tal como o exigido para T-malhas. Na aplicação, a geração de *quadballs* é alcançada partindo de uma T-malha cúbica simples com seis faces, submetendo-a a sucessivas iterações do algoritmo de subdivisão de Catmull-Clark [9]. Em virtude do vasto reconhecimento e uso do algoritmo de subdivisão supracitado, apenas seus pontos principais são descritos.

Partindo de $\mathcal{M}_i$, i -ésima malha do processo, onde se $i = 0$ tem-se uma malha cúbica inicial, a malha $\mathcal{M}_{i+1}$ é obtida pelo cômputo, seguido da conexão de pontos

de face, aresta e vértice. Um ponto de face, P_k^f , é dado pela média de seus vértices, v_1, v_2, v_3 e v_4 , seguindo a equação

$$P_k^f = \frac{v_1 + v_2 + v_3 + v_4}{4}. \quad (3.1)$$

Já um ponto de aresta, P_k^a , é calculado como a média dos vértices incidentes e dos pontos de face das faces que a aresta toca, P_l^f e P_m^f . Considerando v_1 e v_2 como os vértices da k -ésima aresta, tem-se que

$$P_k^a = \frac{v_1 + v_2 + P_l^f + P_m^f}{4}. \quad (3.2)$$

Por último, um vértice v_k é associado a um ponto de vértice P_k^v , cuja posição é calculada por

$$P_k^v = \frac{\bar{P}^f + 2\bar{R} + (\deg v_k - 3)v_k}{\deg v_k}, \quad (3.3)$$

onde $\bar{P}^f$ é a média dos pontos de face em que o vértice incide, $\bar{R}$ é a média dos pontos médios de todas as arestas em que o vértice incide, e $\deg v_k$ denota a valência do vértice.

Posteriormente, as arestas e faces da malha $\mathcal{M}_{i+1}$ são obtidas conectando cada ponto de aresta aos pontos de face em que a aresta incide, e também aos pontos de vértice que incidem nessa aresta.

3.3 Geração de T-splines

Um dos objetivos específicos deste trabalho, como visto em Seção 1.2, consiste em revisar o processo de extração de Bézier e integrá-lo a um *pipeline* adequado para uma aplicação interativa, tal como a aqui proposta. Neste ponto, deseja-se partir de uma T-malha e determinar um conjunto de elementos de T-spline que então podem ser renderizados na aplicação. A classe que representa uma T-spline é *TSpline*, que possui uma T-malha e é capaz de gerar os elementos de T-spline, objetos da classe *TSplinePatch*, pelo uso do método `TSpline::generatePatches()`, que por sua vez, o faz por meio de métodos de *TMesh*. As relações descritas constam na Figura 3.3, por meio de um diagrama UML que descreve a relação entre as classes supracitadas.

Os métodos em *TMesh* podem ser organizados em um *pipeline* com a finalidade de gerar a T-spline, sendo eles:

P1 `TMesh::setIrregularRegion();`

P2 `TMesh::extendTMesh();`

```

P3 TMesh::checkInconsistencies();

P4 TMesh::computeLTHs();

P5 TMesh::computeRegularExtraction();

P6 TMesh::computeExtraordinaryElementsExtraction().

```

A lista destaca o nome dos métodos em *TMesh* de cada passo do processo, dados por **P1-P6**, cuja ordem foi estabelecida com base nas premissas de cada parte do processo de geração da extração de Bézier, descrito na Seção 2.2. Iniciando com **P1**, determinam-se as regiões irregulares por meio de uma busca na vizinhança 2-disco de face de todos os pontos extraordinários (EPs) da malha, realizada na estrutura topológica HBDS. A Figura 3.4 demonstra o funcionamento deste método, onde, para cada face na qual um EP incide, coleta-se o vértice na diagonal (v_d) para então marcar todas as faces em que v_d incide como faces irregulares. Conforme indicado anteriormente, a classe *TFace*, apresentada na Seção 3.2, armazena o booleano que indica a irregularidade da face.

Na sequência, o passo **P2** realiza a determinação das extensões de T-junções, bem como a inserção delas na T-malha, formando a T-malha estendida (ou elementar). As arestas que representam na malha as extensões de face, bem como os vértices por elas gerados, são ditos componentes *virtuais*, indicados na estrutura de dados por *TEdge::virtual* e *TVertex::virtual*, respectivamente. Componentes virtuais são, topologicamente, tais como quaisquer outras arestas ou vértices, diferenciando-se apenas em processos como **P5**, onde ignoram-se vértices virtuais enquanto arestas virtuais não restringem o domínio paramétrico de um vértice.

A adição das extensões de face resulta na subdivisão das faces adjacentes em duas, incorrendo na alteração adequada da T-malha e sua HBDS subjacente. A Figura 3.5 ilustra um recorte de uma T-malha para demonstrar a adição da segunda extensão de

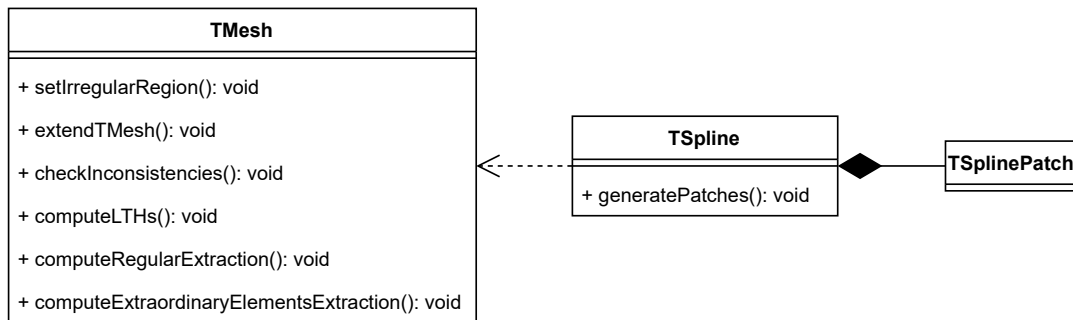


Figura 3.3: Diagrama de classes UML representando a relação entre as principais classes envolvidas na geração de T-Spline.

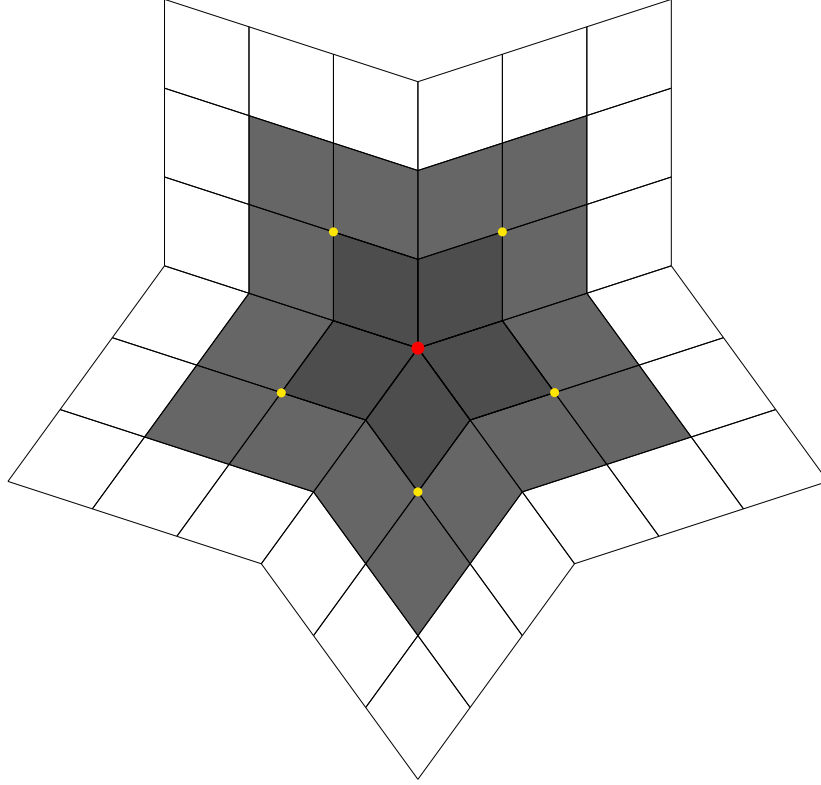


Figura 3.4: Determinação de região de faces irregulares via topologia. O ponto extraordinário é ilustrado como um ponto vermelho, enquanto os vértices na diagonal, v_d , são pontos amarelos. Todas as faces em que os vértices em amarelo incidem são determinadas como irregulares e pintadas em cinza.

face de uma T-junção, focando nas alterações na estrutura de semi-arestas. Na Figura 3.5(a), demonstra-se a primeira extensão de face de uma T-junção. Enquanto na Figura 3.5(b), tem-se a inserção da segunda extensão de face, dividindo a face que ela incide em duas, e reajustando as semi-arestas da HBDS. Nela, as duas extensões de face (em vermelho) e suas extremidades (pontos brancos com borda vermelha) são componentes virtuais. Enfim, embora não tenha sido ilustrado, os pesos das arestas em vermelho e das que foram interceptadas por elas são ajustados para garantir que os lados opostos de cada face somem pesos iguais.

Sejam $a_{i,1}^f$ e $a_{i,2}^f$ a primeira e segunda extensões de face de uma T-junção v_i . Ressalta-se que na implementação, durante o passo **P2**, se a inserção de uma extensão de face do vértice v_j interceptar outra extensão de face topologicamente perpendicular, $a_{i,1}^f$ ou $a_{i,2}^f$, por exemplo; realiza-se imediatamente a inserção de arestas sobre todas as extensões de face da T-junção v_i . Essa abordagem garante que novas extensões de T-junção advindas de refinamentos da malha não “encurtem” extensões existentes, violação **V2** da Seção 2.3, além de simplificar os tratamentos necessários no passo **P3**.

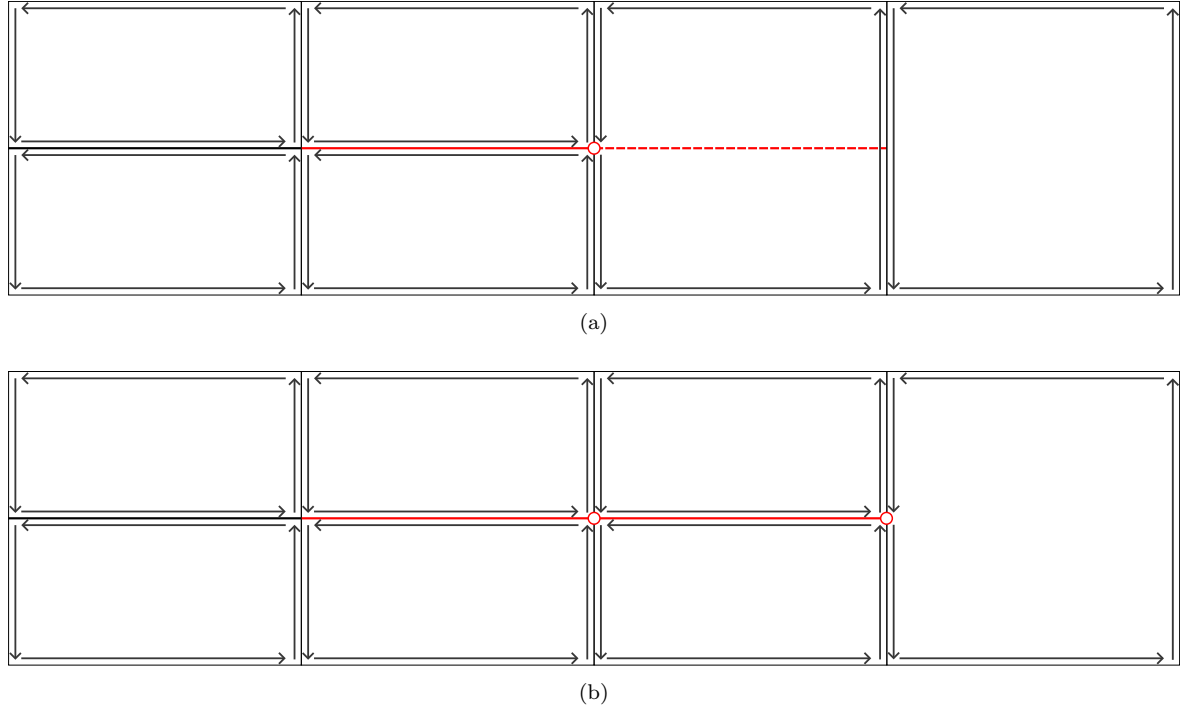


Figura 3.5: Inserção topológica de uma extensão de face na T-malha. As linhas em vermelho são extensões de face, onde as sólidas já foram inseridas na T-malha, enquanto a tracejada ainda será adicionada. A inserção da segunda extensão de face (em vermelho tracejado) causa ajustes na semi-arestas da HBDS (setas em cinza), além da inserção de novos vértices (círculos com borda vermelha). (a) Topologia antes da inserção da segunda extensão. (b) Topologia após a inserção em (a).

O passo **P3** configura a principal contribuição deste trabalho, entretanto seu funcionamento é descrito na Seção 3.5. Considera-se que esse passo produz uma T-malha elementar admissível.

O cômputo dos LTHs é reservado para o passo **P4**, empregando as equações Equações (2.13)-(2.15) apresentadas na Seção 2.2.3. Destaca-se que vértices virtuais não admitem LTHs, mesmo que incidam em arestas de vinco, a representação de vincos em regiões irregulares segue a formulação de Peres et al. [22] e foi detalhada no Capítulo 2. Conforme destacado previamente, os pontos de tangência não estão efetivamente na topologia da T-malha, isto é, não incidem em nenhuma aresta ou face; e portanto, são armazenados no vértice que incidem. A Figura 3.6 explana a organização dos LTHs em um vetor de tamanho oito, definido em `TVertex::tangencyHandles`, de forma que cada posição representa a localização de um LTH. Particularmente, as posições dos LTHs armazenados nos índices 0, 2, 4 ou 6 são determinadas por meio da Equação (2.13), o restante são LTHs nas faces, dados pela Equação (2.14).

Por fim, os passos **P5** e **P6** compreendem o cômputo do operador de extração de Bézier para faces regulares e irregulares, respectivamente. A implementação segue o mesmo apresentado em minúcias nas Seções 2.2.1 e 2.2.2.

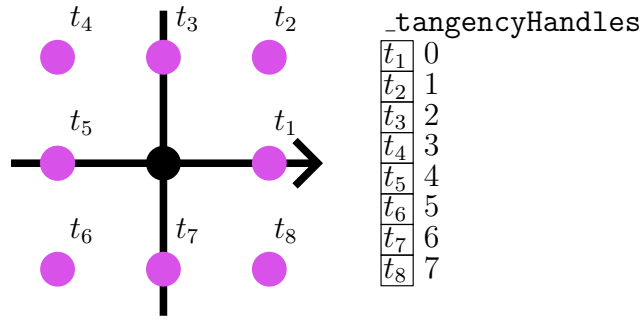


Figura 3.6: Organização dos pontos de tangência, ou LTHs (círculos roxos), associados a um vértice regular de valência quatro (círculo preto). A este vértice, na HBDS, associa-se uma semi-aresta (seta) que serve como indicativo de qual LTH está da HBDS que esse vértice está associado, é com base nele que os LTHs são organizados em `TVertex::_tangencyHandles`.

3.4 Funcionalidades de edição e refinamento

Esta seção apresenta as principais funcionalidades que permitem interagir com as T-splines da aplicação. Tais funcionalidades são chamadas de *comandos*, representados pela classe *Command*, e ilustrados no diagrama da Figura 3.7. Nela, tem-se os comandos de edição e refinamento implementados, *EditShapeCommand* para manipular a posição de vértices, arestas ou retalhos de T-spline; *CreaseCommand* para prescrever arestas de vinco; *SplitCommand* para a subdivisão de face em uma ou duas direções paramétricas; e *RefineIrregularRegionCommand*, capaz de refinar uma região irregular de maneira uniforme.

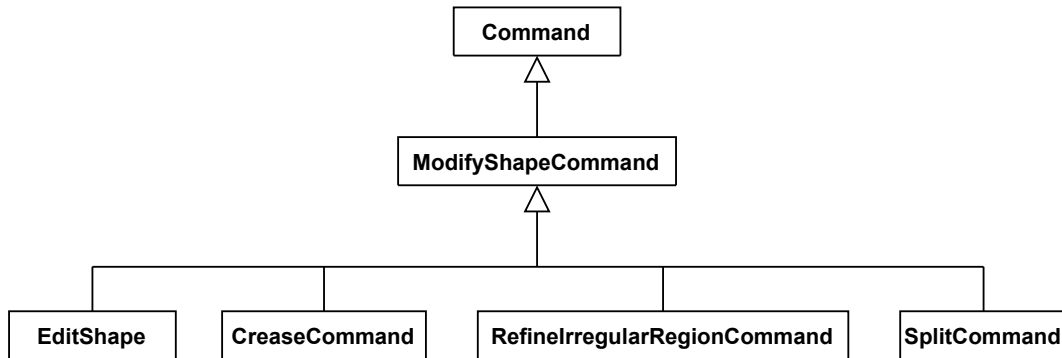


Figura 3.7: Diagrama de classes UML dos comandos que modificam T-splines.

A classe *EditShapeCommand* aplica uma translação nos componentes topológicos selecionados, agregando valor nas capacidades de modelagem da aplicação, enquanto *CreaseCommand* é responsável por prescrever vincos em arestas. A prescrição de vinco se concretiza no passo **P4** da geração de T-splines, onde são calculados os pontos de tangência e armazenados nos vértices que estão relacionados. *SplitCommand* emprega manipulações topológicas na HBDS para subdividir uma face regular da T-malha, resultando na criação de novos vértices, arestas e faces. Esse comando toma como

parâmetro qual direção paramétrica subdividir a face, ou *ambas* para subdividi-la em 4 partes. Para melhor definir o funcionamento do comando de subdivisão de faces de alto nível, convém introduzir dois procedimentos fundamentais de subdivisão na T-malha e sua HBDS subjacente, as subdivisões topológicas de aresta e de face. A primeira é implementada no método `splitEdge(TMesh&, HalfEdgeRef, real)`, que toma uma T-malha, uma semi-aresta e um valor real $t \in (0, 1)$. Seu funcionamento é ilustrado na Figura 3.8, onde busca-se dividir uma aresta de peso w_i^a . Ressalta-se que essa operação não configura um comando válido a ser executado isoladamente em T-splines, visto que seu resultado é a formação de um ponto interno em arestas, chamado junção em I, que faz com que a T-spline não seja conforme para análise [17].

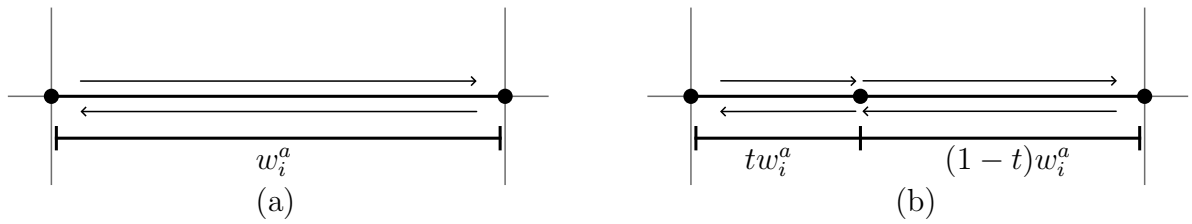


Figura 3.8: Subdivisão de uma aresta (em preto) sob peso $t \in (0, 1)$, de forma a gerar duas arestas com pesos tw_i^a e $(1-t)w_i^a$ que somam o peso da aresta subdivida. Os vértices envolvidos são ilustrados com círculos pretos, enquanto as setas indicam a sequência de semi-arestas da malha HBDS. (a) Topologia da face antes da subdivisão da aresta na base. (b) Após a subdivisão da aresta, destacando os pesos das novas arestas, bem como a conservação da sequência de semi-arestas da face.

Quanto à subdivisão topológica de face, a implementação conta com a função `insertEdge(TMesh&, HalfEdgeRef, real)`, cuja responsabilidade é de inserir uma aresta paralela a aresta associada a semi-aresta fornecida. Na Figura 3.9(a) mostra-se o uso da subdivisão de arestas para produzir vértices (círculos pretos), os quais serão conectados na Figura 3.9(b) por uma nova aresta inserida na malha, causando a divisão da face f_i em duas, f_j e f_k .

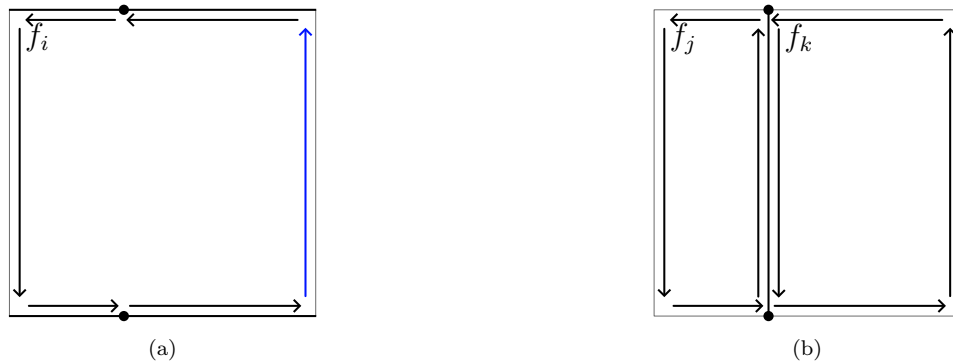


Figura 3.9: Inserção topológica de aresta na T-malha, recebendo a semi-aresta em azul como entrada. (a) Uma face que possui dois vértices inseridos pelo método de subdivisão de aresta. (b) Inserção da aresta que subdivide a face f_i em duas, f_j e f_k .

A subdivisão de uma face nas duas direções paramétricas é exposta, passo a passo, na Figura 3.10. Seu funcionamento emprega a mesma função `insertEdge` da Figura 3.9, repetida três vezes para produzir quatro faces onde antes havia uma.

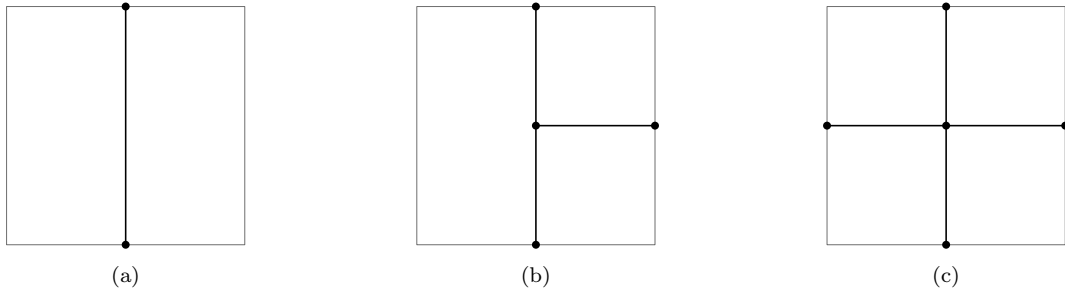


Figura 3.10: Subdivisão topológica de face na T-malha, destacando com círculos pretos os novos vértices e com linhas pretas as novas arestas. (a) Inserção da primeira aresta para subdividir a face. (b) Segunda inserção de aresta, ligando um vértice da aresta central com outro vértice inserido em uma aresta da face. (c) Inserção do último vértice em uma aresta da face original, para então posicionar a última aresta, resultando em uma subdivisão completa.

Destaca-se que o processo de subdivisão de uma face em duas é similar ao que acontece na inserção das extensões de T-junção, mencionadas na Seção 3.3, e também serve como base para o algoritmo de refinamento local introduzido no Capítulo 2.

A classe *RefineIrregularRegionCommand* realiza o refinamento uniforme de uma região irregular inteira, de forma que cada face irregular desta região seja subdividida nas duas direções paramétricas, resultando em 4 faces. Comumente, esta subdivisão ocasiona a criação de T-junções apontadas para fora da região irregular, podendo causar inconsistências na T-malha que serão corrigidas posteriormente. Devido a similaridade com a classe *SplitCommand*, refere-se à Figura 3.10 para compreender as modificações topológicas das faces de uma região irregular. Ressalta-se que as subdivisões são realizadas sempre de forma a dividir o peso das arestas ao meio, similarmente ao adotado por Toshniwal et al. [34].

Enfim, salienta-se que na aplicação, a execução dos comandos de refinamento *SplitCommand* e *RefineIrregularRegionCommand* alteram a topologia da T-malha, e portanto exigem que a mesma seja recomputada, fazendo com que o *pipeline* enunciado na Seção 3.3 seja percorrido novamente. E assim, quaisquer inconsistências introduzidas, bem como mudanças na rotulagem das faces em regulares e irregulares pelo uso dos comandos, serão resolvidas.

3.5 Algoritmo de resolução de inconsistências

Disponibilizadas as capacidades de subdividir elementos da T-spline na aplicação via comandos, apresentadas na Seção 3.4. É necessário então garantir que o resultado dessas operações seja uma AST-spline que é um refinamento da AST-spline original. É este o objetivo do algoritmo de resolução de inconsistências, o passo **P3** do *pipeline* de geração de T-splines. Tendo em vista os objetivos deste trabalho, foi proposto um algoritmo que solucione os problemas da T-malha observando apenas sua topologia, sem o uso de estruturas de dados adicionais, como o proposto por Scott et al. [29]. Em adição, o presente algoritmo também aborda explicitamente casos de extensões de T-junção que invadem regiões 3-anel de face de EPs. A Tabela 3.1 mostra os tipos de inconsistências a serem solucionadas nas T-malhas deste trabalho, derivadas das restrições de AST-splines descritas na Seção 2.2. Convém ressaltar que as inconsistências **I1**, interseções entre extensões de face topologicamente perpendiculares, são tratadas imediatamente no passo **P2**. Dessa forma, o algoritmo mencionado trata os casos restantes.

Tabela 3.1: Inconsistências

Código	Descrição
I1	Cruzamento entre extensões de face
I2	Cruzamento de extensão de face com extensão de aresta
I3	Cruzamento entre extensões de aresta
I4	Extensão de aresta na região 3-anel de face de um EP
I5	Primeira extensão de face na região 3-anel de face de um EP

O algoritmo proposto opera como segue. Partindo de uma T-malha estendida $\mathcal{M}_{ext}$, identificam-se as inconsistências **I2** e **I3** diretamente da topologia da malha. Esta identificação é realizada consultando, para cada T-junção, v_i^t , se arestas perpendiculares às suas extensões de T-junção, compostas por extensões de face e de aresta, são extensão de aresta de outra T-junção ou são virtuais, **I2** e **I3**, respectivamente. Nesses casos, insere-se uma aresta sobre as primeira extensão de face de v_i^t , seguida da produção de uma nova extensão de face que mantenha a malha estendida, até que não haja mais inconsistências **I2** e **I3** nesta T-junção. Para ilustrar tais casos, tem-se na Figura 3.11(a) o cruzamento da extensão de aresta de v_1^t com a extensão de face perpendicular de v_2^t , bem como o cruzamento das extensões de aresta de v_2^t e v_3^t , inconsistências **I2** e **I3**, respectivamente. Na Figura 3.11(b) é apresentada a T-malha após as inserções de aresta nas primeiras extensões de face de v_1^t e v_2^t solucionando as inconsistências existentes, e por consequência gerando as novas T-junções v_4^t e v_5^t . Esta solução não é única e assume que as T-junções são avaliadas na ordem v_1^t , v_2^t e v_3^t .

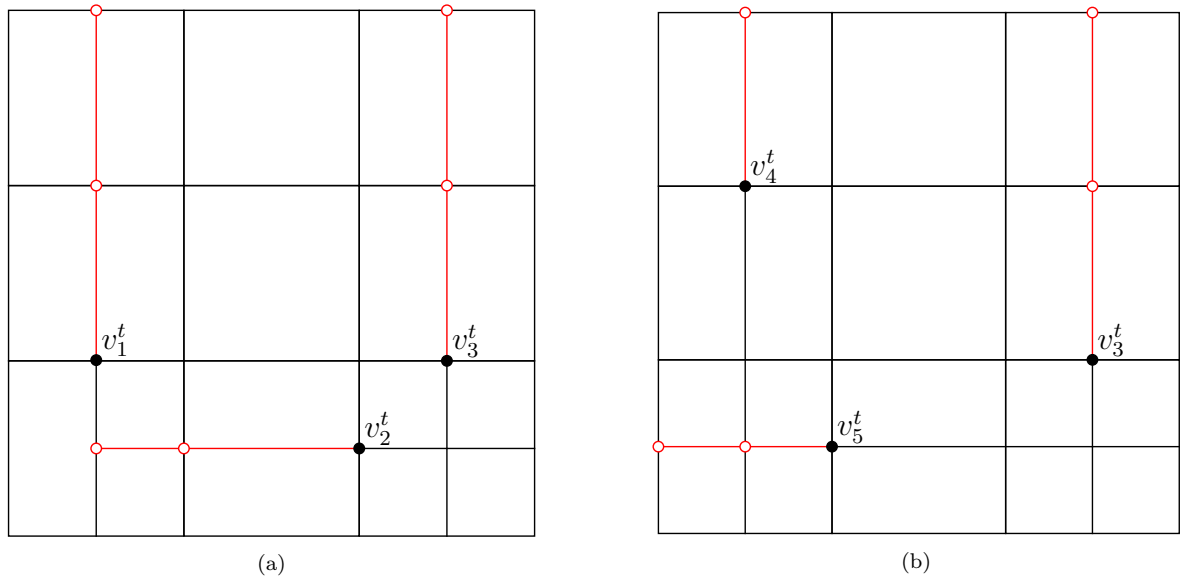


Figura 3.11: Cruzamentos de extensões de T-junção na T-malha, destacando as T-junções $v_1^t, \dots, v_5^t$ (círculos pretos), bem como suas extensões de face (linhas vermelhas) e extremidades (círculos brancos com borda vermelha). (a) T-malha com inconsistências **I2** e **I3**, e que portanto, não é admissível. (b) T-malha com as inconsistências corrigidas por meio de inserções de aresta sob as extensões de face de suas T-junções.

As inconsistências **I4** e **I5** são identificadas a partir de verificações em todas as faces na vizinhança 3-anel de face, $\mathcal{R}_i^3$, de todos os EPs da T-malha. A primeira é detectada quando uma T-junção é um dos vértices no canto da face. Enquanto a segunda inconsistência, **I5**, ocorre quando um dos lados de uma dessas faces é delimitado por uma aresta virtual $a_{i,1}^f$, a primeira extensão de face de uma T-junção v_i . A Figura 3.12 resgata a T-malha da Figura 3.4 e insere nela novas arestas, gerando T-junções, v_1^t e v_2^t , que ao serem estendidas, causam inconsistências **I4** e **I5**, respectivamente. Tais problemas são constatados verificando que existem faces no 3-anel de face do EP que foram subdivididas pelas extensões, de aresta ou de face. Topologicamente, verifica-se que v_1^t é um canto de duas das faces em $\mathcal{R}^3$ enquanto a primeira extensão de v_2^t é um dos lados de outras duas faces no mesmo conjunto. Os dois casos são solucionados com o refinamento uniforme da região irregular que violam. Neste momento, quando ao menos um refinamento de região irregular é aplicado, repetem-se os passos **P1** e **P2** do *pipeline* de geração de T-splines.

Embora **I4** não conste no Capítulo 2 como uma violação direta nas condições de AST-splines, o posicionamento de suas extensões de T-junção causaria a subdivisão de pelo menos uma face irregular, que conforme mencionado na Seção 2.3, só podem ser subdivididas de maneira uniforme. Mesmo que fossem suprimidas as restrições de refinamento uniforme delimitadas por Toshniwal et al. [34], a presença de vértices e arestas virtuais na região irregular causaria problemas no cômputo do operador de extração de Bézier para essas faces [30]. É desta forma que o enquadramento de **I4**

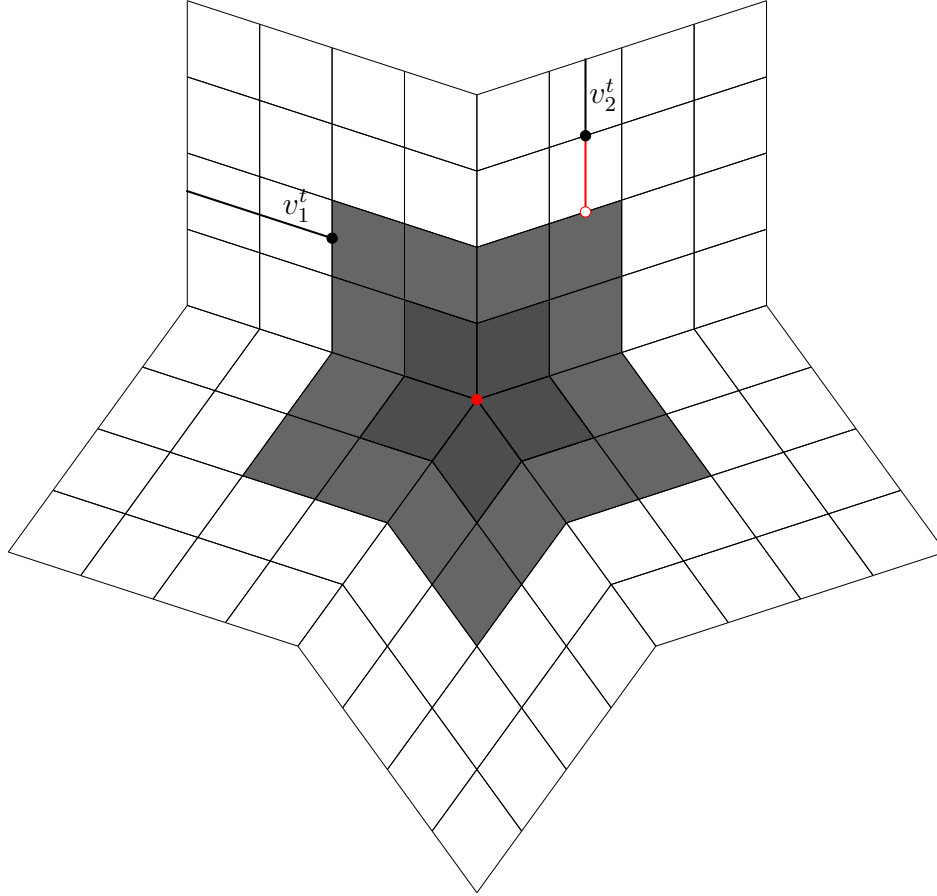


Figura 3.12: Inconsistências ocasionadas por extensões de T-junção na região 3-anel de face de um ponto extraordinário (círculo vermelho). T-junções são dadas por círculos pretos, enquanto extensões de face e suas extremidades são dadas por arestas vermelhas e círculos brancos com borda vermelha, respectivamente.

como inconsistência é justificado. Por fim, as tratativas supracitadas repetem-se até que nenhuma inconsistência seja detectada, então o algoritmo termina.

Não foram realizados testes comparativos para mensurar quanto o algoritmo proposto é melhor ou pior que o apresentado por Scott et al. [29], ou outros da literatura, no que diz respeito a quantidade de arestas e vértices inseridos.

3.6 Considerações finais

Neste capítulo discutiu-se a implementação orientada a objetos desenvolvida, envolvendo as teorias e formulações previamente apresentadas. Iniciou-se abordando as funcionalidades da aplicação, como criação ou importação de T-malhas, realizadas por meio de um arquivo TSM que a descreve. Das T-malhas disponíveis na aplicação, especificou-se em maiores detalhes a *quadball*, obtida a partir da repetida subdivisão de Catmull-Clark. Na sequência, a sequência de passos que culmina na geração das

T-splines foi esmiuçada em detalhes.

Então, os comandos que permitem editar e refinar T-splines na aplicação foram descritos, juntamente com sua organização em uma hierarquia de classes na codificação. Por fim, expôs-se a principal contribuição deste trabalho, um algoritmo de resolução de inconsistências que se baseia completamente na topologia da malha, permitindo corrigir T-malhas para que sejam admissíveis, mesmo quando editadas pelos comandos fornecidos na aplicação.

Capítulo 4

Resultados

4.1 Considerações iniciais

Neste capítulo apresenta-se e discute-se os resultados obtidos com o uso do programa de computador desenvolvido, evidenciando suas capacidades de modelagem e refinamento de T-splines conformes para análise. Os casos de teste reforçam as funcionalidades descritas no Capítulo 3. Além de trazerem comparativos, quando possível, com métodos similares existentes na literatura. Desta forma, inicia-se com exemplos das malhas produzidas pela própria aplicação e importadas na Seção 4.2. Enquanto na Seção 4.3 são ilustradas as possibilidades de manipulação nas T-malhas, envolvendo prescrição de arestas de vinco, edição da forma e refinamento.

Todos os testes realizados nesta seção utilizaram uma máquina *desktop* equipada com um processador AMD Ryzen 9 5900X em 3.7GHz com 32 GB de DRAM, e uma GPU NVIDIA GeForce 960 com 4GB de DRAM.

4.2 Exemplos iniciais

Os testes aqui apresentados almejam demonstrar as possibilidades da aplicação desenvolvida no que tange criação e importação de T-malhas. Para tanto, ilustra-se nas Figuras 4.1(a)-(d) T-malhas criadas pela aplicação desenvolvida; cubo, cilindro, *quadball* e torus, respectivamente. T-malhas planares também são criadas na aplicação, mas são evidenciadas em exemplos posteriores. Ressalta-se que o cubo na Figura 4.1(a) não possui arestas de vinco assim que criado, ficando a cargo do usuário prescrever os vincos. Tal exemplo é apresentado posteriormente no Seção 4.3.

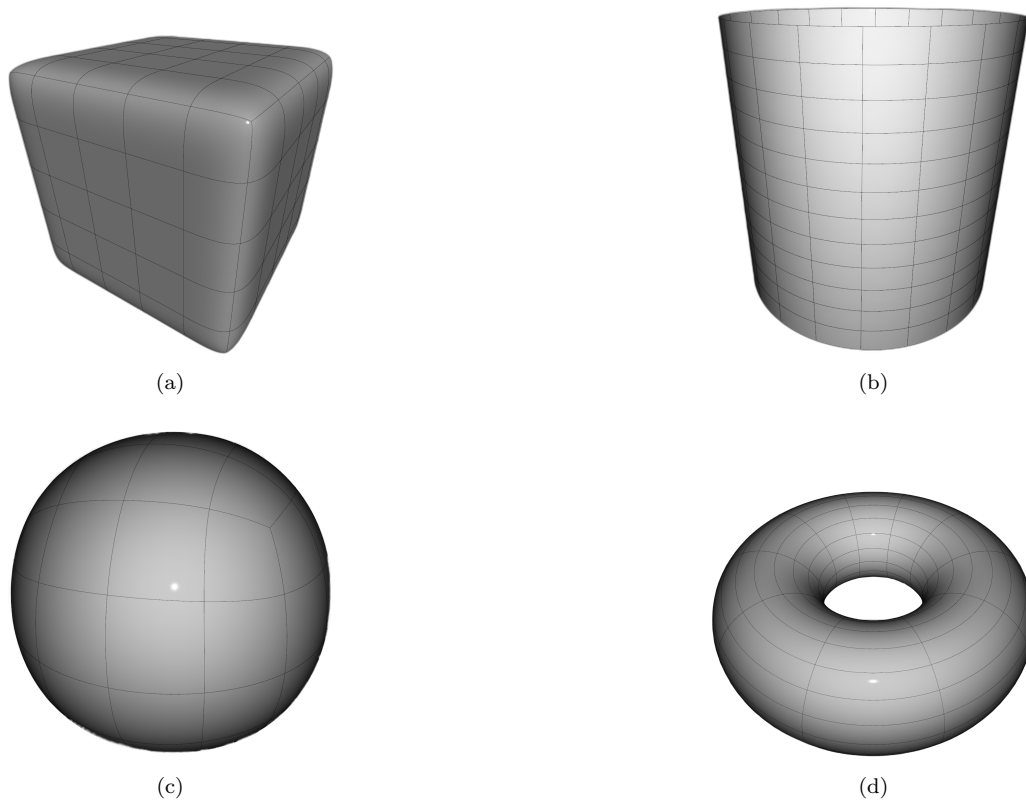


Figura 4.1: Exemplos de T-malhas produzidas pela aplicação. (a) Malha cúbica sem vincos com 5 faces em cada lado. (b) Cilindro aberto. (c) *Quadball* com 2 iterações da subdivisão de Catmull-Clark. (d) Torus.

Para atestar importação de T-malhas por meio de arquivos TSM, mencionada na Capítulo 3, apresenta-se na Figura 4.2 um cilindro perfurado, modelado na ferramenta CAD Autodesk Fusion 360 [1] e importado para a aplicação desenvolvida. Verifica-se que atributos como os vincos no topo e fundo da forma foram preservados na importação.

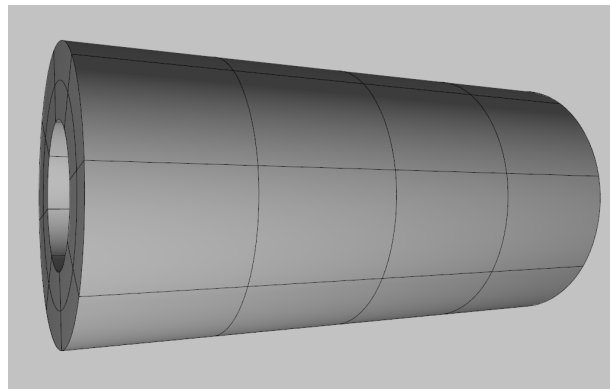


Figura 4.2: Cilindro perfurado modelado na ferramenta Autodesk Fusion 360 e importado para a aplicação desenvolvida.

4.3 Edição e refinamento em T-splines

Após a criação de T-malhas, apresentada na Seção 4.2, torna-se o foco evidenciar os processos de edição e refinamento de T-malhas implementados por meio de comandos, descritos no Capítulo 3. Para evitar ambiguidades, os comandos de edição e refinamento são demonstrados em malhas planares e cúbicas, visto que apresentam diferentes características topológicas, enriquecendo as demonstrações. Evidentemente, malhas planares apresentam bordas com vinco, enquanto cubos apresentam oito pontos extraordinários, um para cada canto do cubo, permitindo explorar casos de vinco tanto em regiões regulares quanto irregulares. A Figura 4.3 ilustra uma T-malha planar cinco por cinco após manipulações na posição dos pontos de controle da sua face central. Ainda em uma malha planar, a fim de pontuar as alterações provenientes do

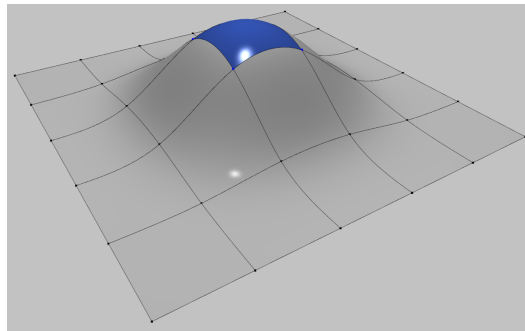


Figura 4.3: Edição de forma T-spline planar. Resultado da translação vertical dos quatro pontos de controle da face central, em azul, de uma T-malha planar cinco por cinco.

refinamento local de faces, traz-se na Figura 4.4 duas vistas da mesma T-spline planar após a subdivisão de sua face central em uma direção paramétrica. Na Figura 4.4(a) tem-se a T-malha, destacando com linhas vermelhas as extensões de face das T-junções geradas, enquanto na Figura 4.4(b) mostra-se a T-malha estendida.

Já na Figura 4.5(a), tem-se a subdivisão da face nas duas direções paramétricas, simultaneamente. Destaca-se que essa subdivisão introduz inconsistências na T-malha que a impedem de ser admissível, são elas o cruzamento de extensões de aresta de T-junções, mencionadas na Seção 3.5. Na Figura 4.5 destaca-se o resultado da subdivisão juntamente com a correção das ditas inconsistências, ocasionando a inserção de duas arestas sobre as primeiras extensões de face das T-junções na horizontal. Esta configura a primeira evidência do funcionamento do algoritmo de correção proposto.

Adotando agora uma malha cúbica com 25 faces em cada lado, isto é, 5 subdivisões em cada direção espacial, apresentam-se na Figura 4.6(a) a T-malha evidenciando as arestas de vinco que serão prescritas. De forma que o resultado desses vincos são ilustrados na Figura 4.6(b).

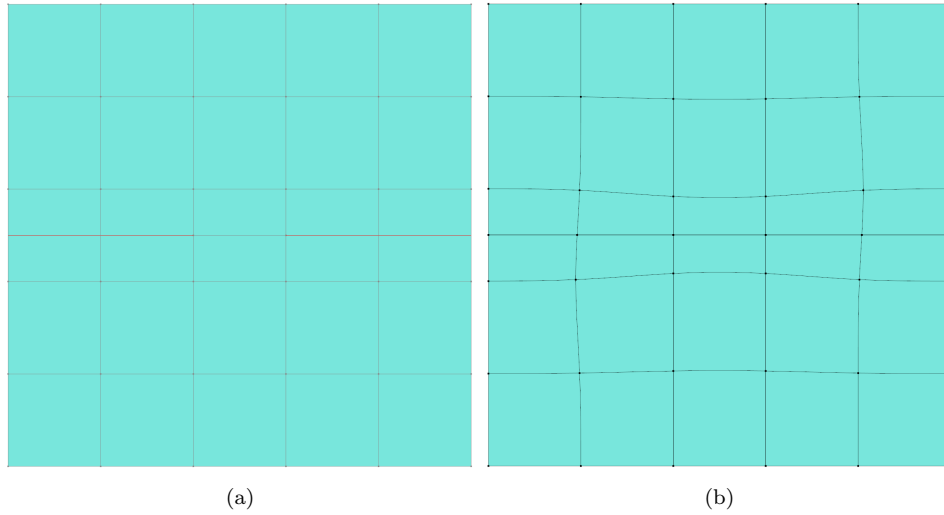


Figura 4.4: T-spline planar com face subdividida em duas. (a) T-malha com face central subdividida, gerando duas extensões de face, em vermelho, para cada T-junção. (b) T-spline após a inserção das extensões de face na T-malha, gerando a T-malha estendida.

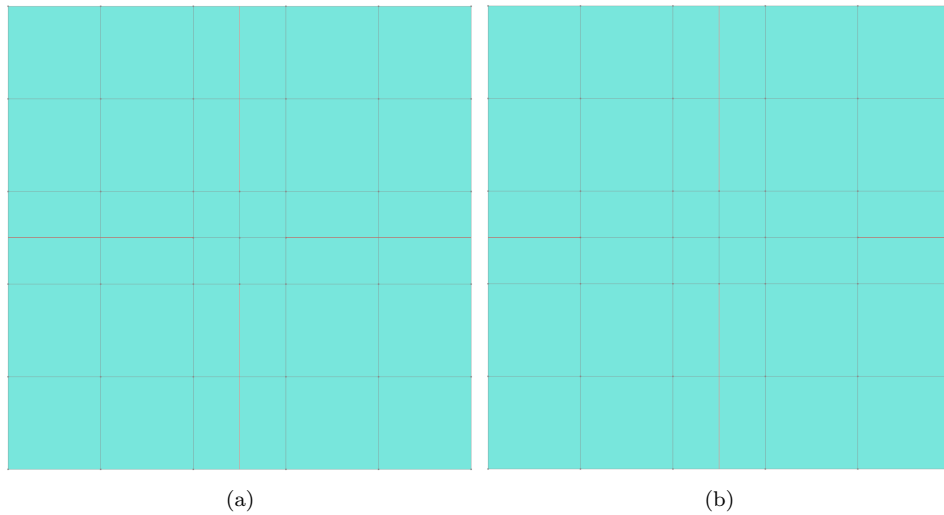


Figura 4.5: T-spline planar com face subdividida em quatro. (a) T-malha estendida não admissível com face central subdividida em duas direções paramétricas, com extensões de aresta perpendiculares se cruzando. Extensões de face estão em vermelho. (b) T-malha de (a) após a execução do algoritmo de resolução de inconsistências, de forma que extensões de face perpendiculares não se cruzam, e portanto a T-malha é admissível.

Para cubo da Figura 4.6(b), utilizou-se o comando *RefineIrregularRegionCommand* para refinar de maneira uniforme uma das regiões irregulares da T-malha. Na Figura 4.7(a) destaca-se o *preview* da subdivisão a ser realizada. Em (b) destaca-se o resultado do comando. Mediante inspeção do restante da T-spline, percebe-se que todas as outras regiões irregulares foram refinadas da mesma forma. Tal refinamento se fez necessário visto que foram introduzidas T-junções cujas extensões de face $a_{i,1}^f$ subdividiram faces no 3-anel de face de outras regiões irregulares, ressaltando que o algoritmo proposto é capaz de lidar com inconsistências **I5**. Resultado similar é ob-

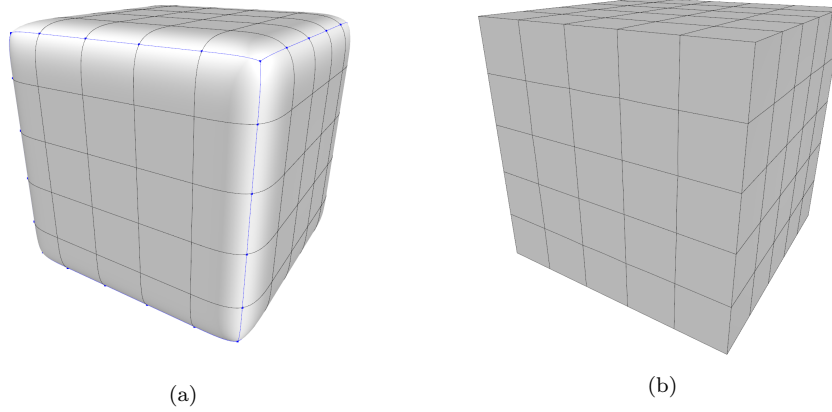


Figura 4.6: T-spline cúbica antes e depois da prescrição de vincos. (a) T-spline cúbica, destacando em azul as arestas em que o vinco será prescrito. (b) T-spline cúbica com vincos nos cantos, resultante a aplicação do comando *CreaseCommand*.

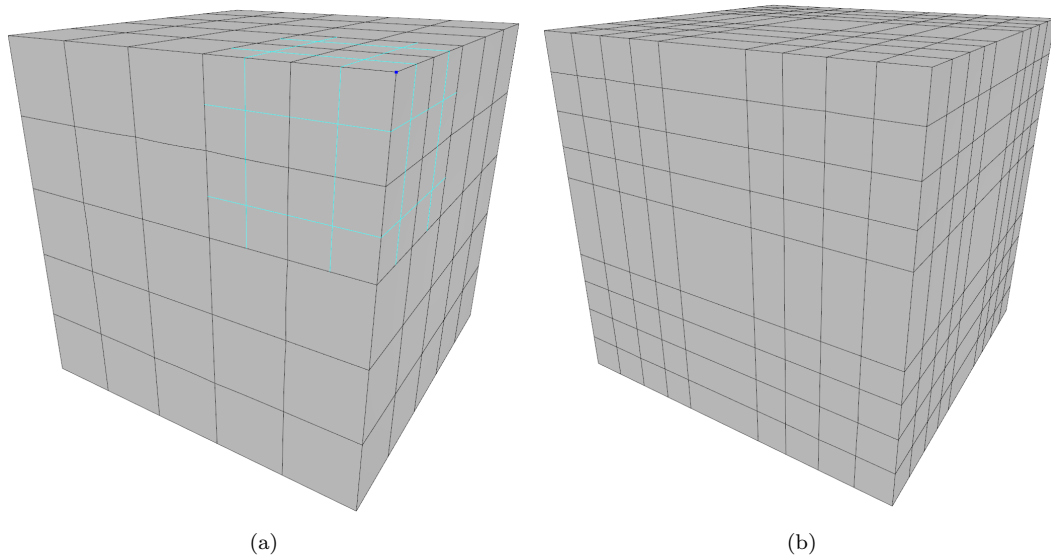


Figura 4.7: Refinamento uniforme de região irregular de T-spline cúbica. (a) T-spline cúbica com vincos, destacando em azul as arestas que serão inseridas para realizar o refinamento uniforme de uma região irregular. (b) T-spline cúbica com vincos após a inserção das arestas destacadas em (a).

tido quando repete-se o teste com uma malha cúbica com quatro subdivisões em cada direção. Nesse caso, expõe-se na Figura 4.8(a) a região irregular que será submetida ao refinamento. Nota-se que as T-junções terão suas extensões de aresta subdividindo uma face no 3-disco de face de outros pontos extraordinários da T-malha, representando inconsistências **I4**. Após o refinamento da região irregular, o algoritmo de resolução detecta os problemas citados e refina as regiões irregulares afetadas, até que não exista mais nenhuma inconsistência. O resultado deste processo é cubo totalmente refinado, destacado na Figura 4.8(b).

Considerando os exemplos apresentados anteriormente, é possível montar cenários mais complexos que demonstrem a completude do algoritmo de correção de inconsistências, englobando correções em regiões regulares e irregulares. A Figura 4.9(a)

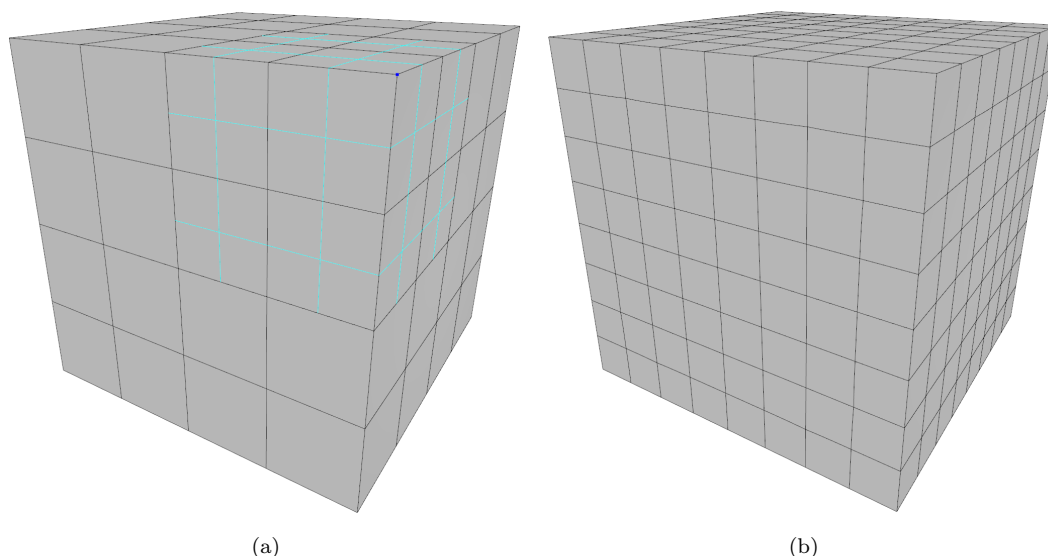


Figura 4.8: Refinamento uniforme de região irregular de T-spline cúbica com 4 subdivisões em cada direção. (a) T-spline cúbica com vincos, destacando em azul as arestas que serão inseridas para realizar o refinamento uniforme de uma região irregular. (b) T-spline cúbica com vincos após a inserção das arestas destacadas na Figura 4.8(a).

mostra a T-malha cúbica submetida a diversas subdivisões de face iniciais. Esses refinamentos foram pensadas para, assim que realizada a operação na Figura 4.9(b), gerem-se inconsistências que uma vez corrigidas, gerem outras, ocasionando diversas iterações do algoritmo de resolução. Na Figura 4.9(c) expõe-se um recorte mais próximo da T-malha, evidenciando a subdivisão a ser realizada, de forma que ao posicionar as extensões de face das T-junções inseridas, será causado um cruzamento de extensão de face com extensão de aresta de outra T-junção abaixo.

A partir da correção gerada pela inserção na Figura 4.9(c), inicia-se a sequência de inserções de aresta, resultando na T-malha da Figura 4.10. Evidencia-se que a subdivisão inicial gerou impactos em extensões de T-junção suficientes, para que as inserções de aresta do algoritmo alcancem uma das regiões irregulares da T-malha, causando seu refinamento completo.

Como último exemplo do algoritmo de refinamento proposto, considera-se uma malha planar similar a da Figura 4.3, submetida a um refinamento local recursivo em sua face central, repetido três vezes. Obtém-se uma região refinada com um total de 64 faces, que posteriormente são transladadas verticalmente, como o exposto na Figura 4.11. Nota-se que modificação da superfície por conta da movimentação das faces em azul tem um efeito reduzido, ficando totalmente restrita ao que antes era uma única face central.

Essa figura visa ilustrar um refinamento local que permitisse, em certo grau, isolar a região delimitada pela face central antes de sua divisão. De forma que fosse possível

restringir condições de contorno a essa região, como o representado pela translação vertical no centro. Nota-se que o refinamento gerou faces o suficiente, para impedir que o efeito na superfície “derramasse” para outras partes fora da face central.

Neste ponto, convém adiantar uma discussão presente na conclusão deste texto, acerca de trabalhos futuros que podem derivar deste. Dentre eles, pode-se citar a produção de um algoritmo que pudesse, adaptativamente, refinar T-splines para que condições de contorno impostas, se restringissem às regiões em que fossem aplicadas. Tal algoritmo poderia operar considerando também parametrizações, tal como o nível de subdivisões do refinamento que deseja-se realizar. O refinamento na Figura 4.3 tentou contemplar tal isolamento com sucessivas subdivisões da região. Contudo, essa abordagem carece de maiores fundamentações e testes que constatem sua validade para solucionar o problema em questão. Como pontos negativos, constata-se que mediante o refinamento, geram-se extensões de T-junção para longe da região alvo, dividindo as faces alcançadas em elementos finos e/ou pequenos, conforme visto na Figura 4.11. No contexto de IGA, a avaliação de integrais em elementos muito pequenos pode ser computacionalmente custosa, além de que, o excesso de elementos pequenos pode resultar em problemas de condicionamento de sistemas lineares dos métodos de análise, tornando a solução instável e mais sensível a erros numéricos. Topologicamente, a grande quantidade de extensões de T-junção também torna a região muito propensa a gerar inconsistências, considerando refinamentos posteriores, que também levariam a mais subdivisões de face para corrigi-las.

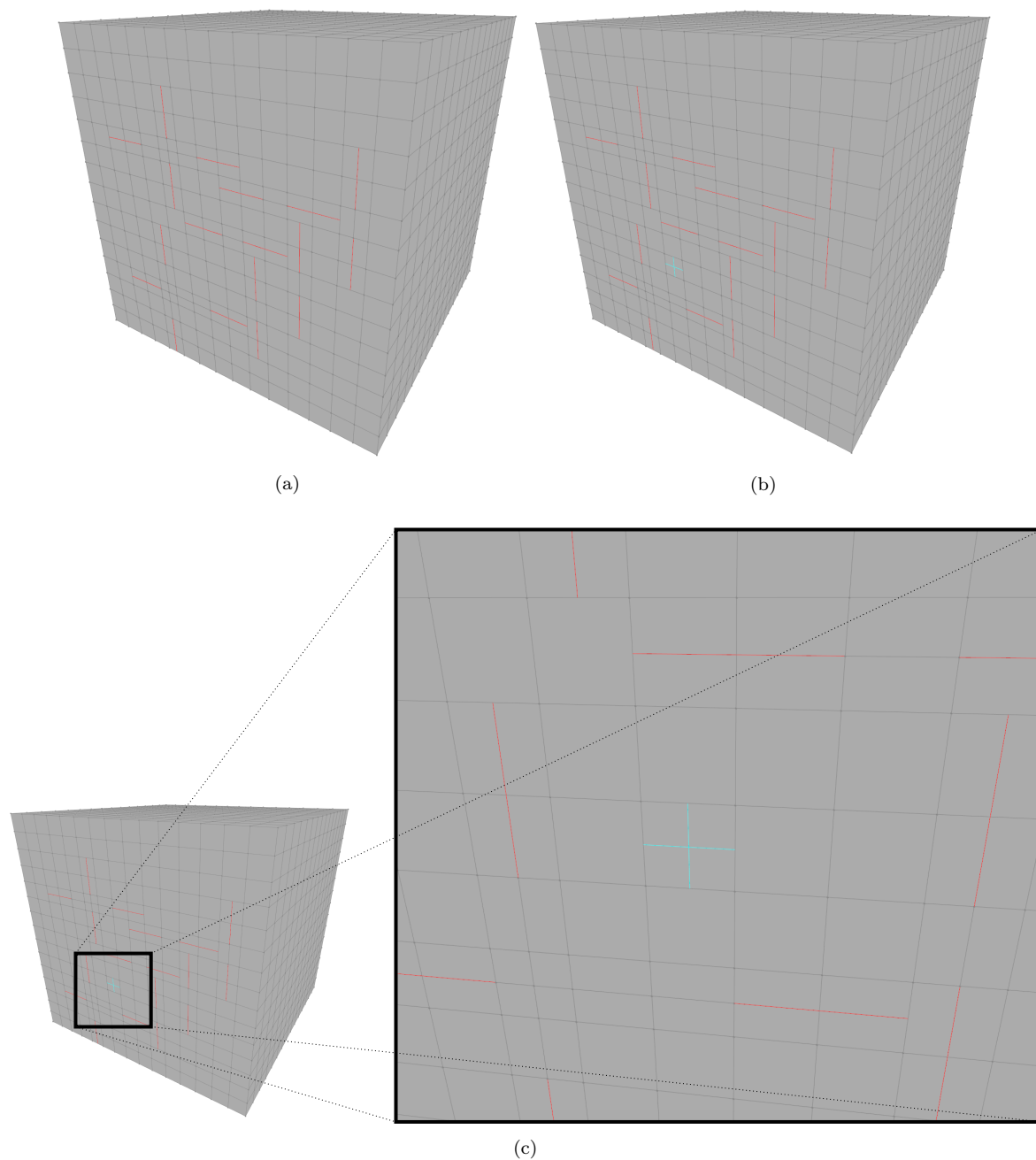


Figura 4.9: Preparação de T-malha cúbica para inserção de aresta. (a) T-malha cúbica após refinamentos variados na região regular. Arestas em vermelho são extensões de face. (b) Seleção de uma face para subdivisão em duas direções, arestas a serem inseridas para a subdivisão estão em azul. (c) Detalhamento do entorno da face a ser subdividida, ressaltando que a inserção, seguida do posicionamento das extensões de face, causará a primeira inconsistência do teste.

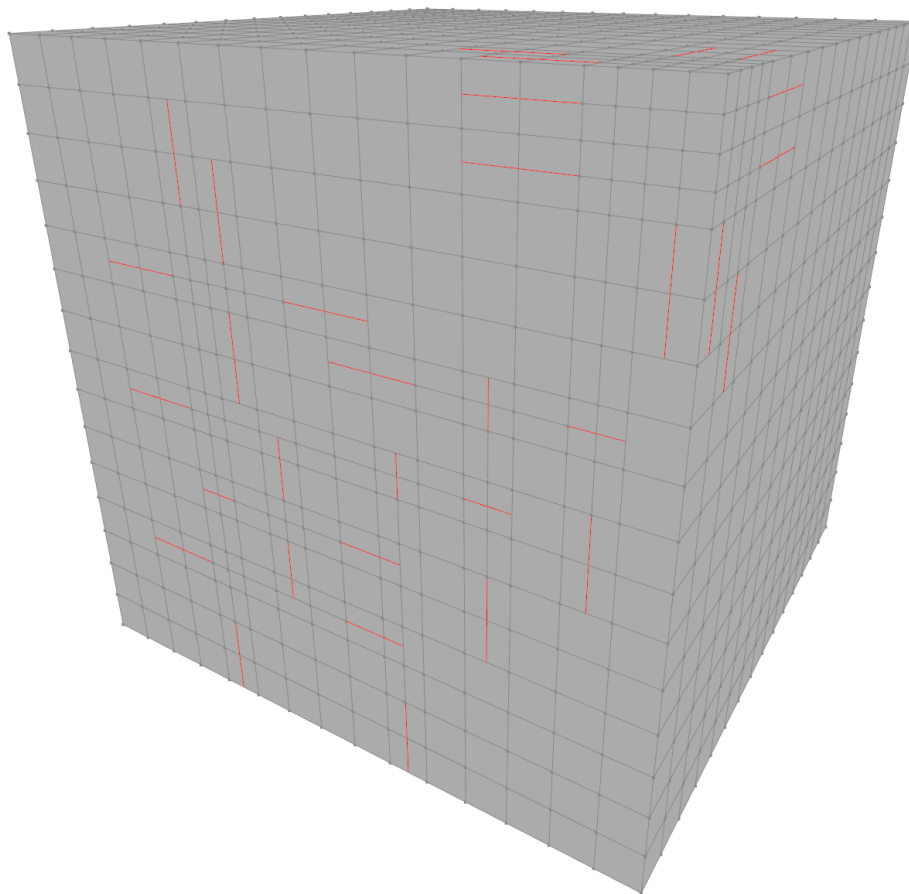


Figura 4.10: T-malha estendida admissível para análise gerada pelo algoritmo de resolução de inconsistências. Arestas em vermelho são extensões de face.

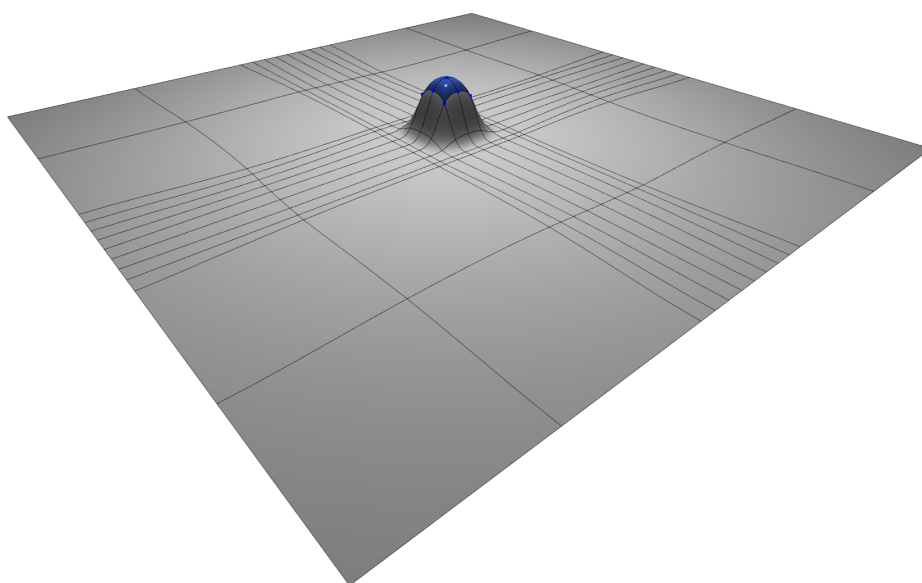


Figura 4.11: Repetidas subdivisões da face central de uma malha planar, seguida de uma translação vertical dos pontos de controle das faces em azul. Ressalta-se que o efeito da edição fica restrito ao que antes era a única face central da malha.

4.4 Considerações finais

O presente capítulo apresentou e discutiu os resultados obtidos por meio do programa da computador desenvolvido. Suas capacidades e funcionalidades, explicitadas no Capítulo 3, foram ilustradas amplamente no decorrer do texto. Exemplos de T-splines geradas pela aplicação e aquelas importadas de fontes externas foram exibidas. Também foi foco deste capítulo a apresentação dos comandos que permitem editar e refinar T-splines e suas T-malhas, sendo eles, edição de forma, refinamento de faces regulares e irregulares e prescrição de vincos. Relacionado aos processos de refinamento, destacou-se o funcionamento do algoritmo de correção da T-malha que a mantém admissível.

Capítulo 5

Conclusão

Neste trabalho foi desenvolvida uma aplicação gráfica interativa para manipular e refinar T-splines não estruturadas conformes para análise, ou AST-splines [17]. Sob a luz os objetivos do trabalho, definidos no Capítulo 1, iniciou-se com uma revisão do processo de *extração de Bézier* em T-splines não estruturadas com vinco [23, 22], que foi codificado de maneira adequada para aplicações interativas. Vincos são alcançados pelo uso de pontos de tangência nas T-malhas. As bases teóricas de T-splines, AST-splines e extração de Bézier encontram-se na Seção 2.2. Com isso, empregou-se uma estrutura de dados topológica HBDS (*Handle-body Data Structure*) para realizar as T-malhas subjacentes, apresentada na Seção 2.4. Com isso, foi possível em estágios iniciais do trabalho, renderizar T-malhas na aplicação. Na sequência, foram implementados operadores de refinamento local em AST-splines, tanto para regiões regulares quanto irregulares. Para que o refinamento não gere T-splines que não são conformes para análise, é necessário corrigir a malha, de forma a remover violações ou inconsistências. Esta teoria foi apresentada na Seção 2.3.

Seguindo para o Capítulo 3, a aplicação desenvolvida foi apresentada em maiores detalhes. Iniciou-se com uma explanação das capacidades de criação e importação de T-malhas, detalhando a construção da T-malha de uma *quadball*. O *pipeline* que gera as T-splines a partir de T-malhas foi discutido passo a passo, acompanhado de explicações das funcionalidades de edição e refinamento, bem como a maneira que foram implementadas. Por fim, a principal contribuição deste trabalho foi apresentada; um algoritmo que resolve inconsistências nas T-malhas sem o uso de estruturas de dados auxiliares, apenas a topologia da T-malha.

No Capítulo 4, todas as capacidades da aplicação são apresentadas, notadamente, criação e importação de T-malhas, edição de forma, prescrição de vincos e diversos cenários de refinamento. Os testes empregaram malhas planares e cúbicas, pois estas

facilitam a visualização dos resultados e também contém características topológicas diversas, como vincos, bordas e regiões irregulares. Tal variedade permitiu a exploração das diferentes funcionalidades do aplicação, em especial, refinamentos e o funcionamento do algoritmo de correção das malhas admissíveis. Tendo em vista o conteúdo previamente exposto, contando com a apresentação da aplicação gráfica interativa, capaz de criar, representar e manipular T-splines conformes para análise, ressalta-se que todos os objetivos específicos deste trabalho foram alcançados.

Por fim, os obstáculos enfrentados no desenvolvimento da aplicação são expostos, seguidos de oportunidades para trabalhos futuros estenderem o que foi aqui apresentado.

5.1 Obstáculos e limitações

A maioria dos obstáculos enfrentados durante a confecção do trabalho surgiram na implementação dos operadores de refinamento e do algoritmo de resolução de inconsistências. A subdivisão topológica de face, componente fundamental para os operadores de refinamento de mais alto nível, envolve complexas operações na malha HBDS subjacente. Além de que tais modificações precisam manter um estado consistente da T-spline, considerando os mais diversos casos, para vértices virtuais ou regulares, arestas de vinco, virtuais ou regulares ou mesmo faces regulares ou irregulares. Isto é, orquestrar alterações topológicas robustas e corretas na HBDS e T-malha se provou um grande desafio.

A aplicação gráfica desenvolvida é mais restrita do que aplicações CAD no que diz respeito a flexibilidade de modelagem e importação de malhas, além dos comandos possíveis para manipular as malhas. Em especial, pode-se citar a inabilidade de gerar T-malhas a partir da importação de outros tipos que superfície, como NURBS. Essa funcionalidade permitia integração com diversas outras aplicações de modelagem, dado que NURBS são amplamente adotadas. No contexto de manipulação de T-splines e T-malhas, não há suporte para remover arestas da T-malha, também não é permitido subdividir faces em mais que 4 outras faces.

5.2 Trabalhos futuros

Este trabalho adotou as restrições topológicas de Scott et al. [30] para T-splines conformes para análise, entretanto, existem avanços na literatura que flexibilizam ainda

mais essas restrições. Em especial, permitindo múltiplos pontos extraordinários por face [35]. Adotar tal flexibilidade permite ainda maiores simplificações no algoritmo de correção de inconsistências, e portanto menos inserções de aresta. Ademais, conforme mencionado na Seção 5.1, novas funcionalidades de edição, como a remoção de arestas em AST-splines, aumentariam as possibilidades de modelagem de T-splines na aplicação, reduzindo a necessidade de utilizar ferramentas CAD externas.

Por fim, apesar de IGA não ser utilizada neste trabalho, ainda compõe o contexto em que ele está inserido. Com isso, seria interessante explorar a remoção de LTHs de Peres et al. [22]. Bem como produzir um algoritmo adaptativo que seja capaz de refinar T-splines automaticamente para que seja possível isolar condições de contorno impostas sob regiões dos modelos de análise. Espera-se que esse método não gere a profusão de elementos muito pequenos e/ou finos, conforme abordado no Capítulo 4. Nessa oportunidade, destacou-se na Figura 4.11 uma proposta de refinamento local que permitiu a aplicação de um deslocamento vertical em uma região sem que seu efeito “derramasse” para outras regiões indesejadas. Entretanto, apesar de possivelmente solucionar o problema a que se propõe, acaba gerando perturbações em aspectos numéricos dos problemas, advindos desses pequenos elementos da T-malha estendida.

Referências Bibliográficas

- [1] AUTODESK. Fusion 360. <https://www.autodesk.com/products/fusion-360/overview>, acessado em 04/02/2025.
- [2] BAZILEVS, Y., CALO, V., COTTRELL, J., EVANS, J., HUGHES, T., LIPTON, S., SCOTT, M., AND SEDERBERG, T. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering* 199 (01 2010), 229–263.
- [3] BEIRÃO DA VEIGA, L., BUFFA, A., SANGALLI, G., AND VÁZQUEZ, R. Analysis-suitable T-splines of arbitrary degree: Definition, linear independence and approximation properties. *Mathematical Models and Methods in Applied Sciences* 23, 11 (2013), 1979–2003.
- [4] BOEHM, W. Inserting new knots into B-spline curves. *Computer-Aided Design* 12, 4 (1980), 199–201.
- [5] BORDEN, M. J., SCOTT, M. A., EVANS, J. A., AND HUGHES, T. J. R. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering* 87, 1-5 (2011), 15–47.
- [6] BUFFA, A., CHO, D., AND SANGALLI, G. Linear independence of the T-spline blending functions associated with some particular T-meshes. *Computer Methods in Applied Mechanics and Engineering* 199, 23 (2010), 1437–1445.
- [7] CASQUERO, H., WEI, X., TOSHNIWAL, D., LI, A., HUGHES, T. J., KIENDL, J., AND ZHANG, Y. J. Seamless integration of design and Kirchhoff–Love shell analysis using analysis-suitable unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering* 360 (2020), 112765.
- [8] CASTELO, A., LOPES, H. C. V., AND TAVARES, G. Handlebody representation for surfaces and Morse operators. In *Curves and Surfaces in Computer Vision and Graphics III* (1992), J. D. Warren, Ed., vol. 1830, International Society for Optics and Photonics, SPIE, pp. 270 – 283.

- [9] CATMULL, E., AND CLARK, J. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design* 10, 6 (1978), 350–355.
- [10] CORNUT, O., AND CONTRIBUTORS. Dear ImGui. <https://github.com/ocornut/imgui>, acessado em 04/02/2025.
- [11] DREAMWORKS. Dreamworks. <https://www.dreamworks.com/>, acessado em 04/02/2025.
- [12] FLORIANI, L. D., AND HUI, A. Shape Representations Based on Simplicial and Cell Complexes. In *Eurographics 2007 - State of the Art Reports* (2007), D. Schmalstieg and J. Bittner, Eds., The Eurographics Association.
- [13] HUGHES, T. J. R., COTTRELL, J. A., AND BAZILEVS, Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* 194 (2005), 4135–4195.
- [14] KATSIKADELIS, J. *Boundary Elements. Theory and Applications*. Elsevier Science & Technology, 2002.
- [15] LEWINER, T., LOPES, H., MEDEIROS, E., TAVARES, G., AND VELHO, L. Topological mesh operators. *Computer Aided Geometric Design* 27, 1 (2010), 1–22.
- [16] LI, X., AND SCOTT, M. A. On the nesting behavior of T-splines, May 2011.
- [17] LI, X., ZHENG, J., SEDERBERG, T. W., HUGHES, T. J., AND SCOTT, M. A. On linear independence of T-spline blending functions. *Computer Aided Geometric Design* 29, 1 (2012), 63–76.
- [18] MAIER, R., MORGENSTERN, P., AND TAKACS, T. Adaptive refinement for unstructured T-Splines with linear complexity, 2021.
- [19] MÄNTYLÄ, M. *An Introduction to Solid Modeling*. Computer Science Press, Inc., USA, 1987.
- [20] MILNOR, J. *Morse Theory*. Princeton University Press, 1963.
- [21] PAGLIOSA, P. A. Ds. <https://github.com/paulo-pagliosa/Ds>, acessado em 04/02/2025., 2021.
- [22] PERES, M., SANCHES, G., PAIVA, A., AND PAGLIOSA, P. Parallel isogeometric boundary element analysis with T-splines on CUDA. *Computer Methods in Applied Mechanics and Engineering* 432 (2024), 117296.

- [23] PERES, M. A. *Elementos de Contorno para Análise Isogeométrica de Sólidos Elásticos com Vincos*. PhD thesis, Universidade Federal de Mato Grosso do Sul, 2021.
- [24] PIEGL, L., AND TILLER, W. *The NURBS Book*, second ed. Springer-Verlag, New York, NY, USA, 1996.
- [25] PIEGL, L. A., AND RICHARD, A. M. Tessellating trimmed nurbs surfaces. *Computer-Aided Design* 27, 1 (1995).
- [26] PIXAR. Open subdiv. <http://graphics.pixar.com/opensubdiv/>, acessado em 11/02/2025.
- [27] POINCARÉ, H. Sur la généralisation d'un théoreme d'euler relatif aux polyedres. *Comptes Rendus de Séances de l'Academie des Sciences* 117 (1893), 144.
- [28] ROCKWOOD, A., HEATON, K., AND DAVIS, T. Real-time rendering of trimmed surfaces. *SIGGRAPH Comput. Graph.* (1989), 107–116.
- [29] SCOTT, M., LI, X., SEDERBERG, T., AND HUGHES, T. Local refinement of analysis-suitable T-splines. *Computer Methods in Applied Mechanics and Engineering* 213-216 (2012), 206–222.
- [30] SCOTT, M., SIMPSON, R., EVANS, J., LIPTON, S., BORDAS, S., HUGHES, T., AND SEDERBERG, T. Isogeometric boundary element analysis using unstructured t-splines. *Computer Methods in Applied Mechanics and Engineering* 254 (2013), 197 – 221.
- [31] SCOTT, M. A., BORDEN, M. J., VERHOOSSEL, C. V., SEDERBERG, T. W., AND HUGHES, T. J. R. Isogeometric finite element data structures based on bézier extraction of t-splines. *International Journal for Numerical Methods in Engineering* 88, 2 (2011), 126–156.
- [32] SEDERBERG, T., CARDON, D., FINNIGAN, G., NORTH, N., ZHENG, J., AND LYCHE, T. T-spline simplification and local refinement. *ACM Transactions on Graphics (TOG)* (2004).
- [33] SEDERBERG, T. W., ZHENG, J., BAKENOV, A., AND NASRI, A. T-splines and T-NURCCs. *ACM Trans. Graph.* (2003).
- [34] TOSHNIWAL, D., SPELEERS, H., AND HUGHES, T. J. Smooth cubic spline spaces on unstructured quadrilateral meshes with particular emphasis on extraordinary

- points: Geometric design and isogeometric analysis considerations. *Computer Methods in Applied Mechanics and Engineering* 327 (2017), 411–458.
- [35] WEI, X., LI, X., QIAN, K., HUGHES, T. J., ZHANG, Y. J., AND CASQUERO, H. Analysis-suitable unstructured T-splines: Multiple extraordinary points per face. *Computer Methods in Applied Mechanics and Engineering* 391 (2022), 114494.
- [36] YANG, J., ZHAO, G., WANG, W., DU, X., AND ZUO, C. Non-uniform C1 patches around extraordinary points with applications to analysis-suitable unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering* 405 (2023), 115849.
- [37] ZHANG, Q., SABIN, M., AND CIRAK, F. Subdivision surfaces with isogeometric analysis adapted refinement weights. *Computer-Aided Design* 102 (2018), 104–114. Proceeding of SPM 2018 Symposium.
- [38] ZIENKIEWICZ, O., TAYLOR, R., AND ZHU, J. *The Finite Element Method: Its Basis and Fundamentals*, seventh edition ed. Elsevier, 2013.