

Universidade Federal de Alagoas  
Centro de Ciências Exatas e Naturais  
Departamento de Matemática

**Reconstrução de superfícies paramétricas  
triangulares usando uma técnica de SPR  
adaptada**

Thales Miranda de Almeida Vieira  
thales@pos.mat.ufal.br

Orientadores:  
Adelailson Peixoto  
William W. M. Lira

Maceió, Setembro de 2005

# Resumo

Este trabalho apresenta um procedimento computacional para a reconstrução de superfícies paramétricas triangulares. Essas superfícies são caracterizadas por possuírem três curvas delimitando o seu contorno e são usualmente utilizadas na modelagem geométrica de problemas reais de Engenharia. A representação de superfícies triangulares usando técnicas convencionais de modelagem como, por exemplo, NURBS (*Non Uniform Rational B-Splines*) não é uma tarefa trivial, pois a descrição do espaço paramétrico normalmente requer domínios retangulares, os quais não se aplicam no tratamento de superfícies triangulares. A técnica proposta neste trabalho é baseada na integração de três módulos. O primeiro é responsável pela discretização das curvas do contorno da superfície e pela geração de uma malha de elementos triangulares que é utilizada como base para a representação da superfície. O segundo módulo consiste na utilização de um algoritmo para a determinação de um espaço paramétrico discreto associado à malha. O último módulo refere-se a utilização de uma técnica para recuperação de valores de coordenadas cartesianas e derivadas associadas às coordenadas paramétricas, garantindo uma transição suave no espaço paramétrico da superfície.

# Abstract

This work presents a computational procedure to reconstruct triangular parametric surfaces. These surfaces are defined by three boundary curves, and they are usually used in geometric modeling of engineering real problems. The representation of triangular surfaces using conventional methods, like NURBS (*Non Uniform Rational B-Splines*), is not a trivial task, because the description of the parametric space normally requires rectangular domains, which cannot be applied to treatment of triangular surfaces. The procedure proposed in this work is divided into three parts. The first is related to the surface's boundary curves discretization and the generation of a triangular mesh that is used for surface representation. The second part uses an algorithm to establish a discrete parametric space associated to the mesh. The last part uses a method to recover Cartesian coordinates and derivatives values associated with any parametric coordinate given, assuring a smooth transition in the surface's parametric space.

# Agradecimentos

Ao Programa Institucional de Bolsas de Iniciação Científica (PIBIC/CNPq), pela bolsa de iniciação científica concedida no período de Agosto de 2004 à Julho de 2005, sob orientação do Prof. Hilário Alencar, à Fundação de Amparo à Pesquisa do Estado de Alagoas (FAPEAL) e ao Instituto do Milênio: Avanço Global e Integrado da Matemática Brasileira (AGIMB) pelos recursos didáticos e computacionais concedidos.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Modelagem de Curvas e Superfícies . . . . .	1
1.2	Objetivos . . . . .	2
1.3	Estrutura do Trabalho . . . . .	3
<b>2</b>	<b>Modelagem de Curvas</b>	<b>4</b>
2.1	Introdução . . . . .	4
2.1.1	Técnicas de Descrição de Curvas . . . . .	4
2.1.2	Derivação de Curvas Paramétricas . . . . .	5
2.1.3	Classes de Funções . . . . .	5
2.2	Curvas de Bézier . . . . .	6
2.2.1	Definição . . . . .	6
2.2.2	Propriedades . . . . .	7
2.2.3	Derivadas . . . . .	8
2.3	Funções Básicas B-spline . . . . .	9
2.3.1	Definição . . . . .	9
2.3.2	Propriedades . . . . .	10
2.3.3	Derivadas . . . . .	11
2.4	Curvas B-spline . . . . .	11
2.4.1	Definição . . . . .	11
2.4.2	Propriedades . . . . .	12
2.4.3	Derivadas . . . . .	12
2.5	Curvas NURBS . . . . .	14
2.5.1	Definição . . . . .	14
2.5.2	Propriedades . . . . .	15
2.5.3	Derivadas . . . . .	15
<b>3</b>	<b>Modelagem de Superfícies</b>	<b>16</b>
3.1	Introdução . . . . .	16
3.1.1	Técnicas de Descrição de Superfícies . . . . .	16
3.1.2	Derivação de Superfícies . . . . .	18
3.2	Superfícies de Bézier . . . . .	18
3.3	Superfícies B-Spline . . . . .	18
3.3.1	Definição . . . . .	18
3.3.2	Propriedades . . . . .	19
3.3.3	Derivadas . . . . .	20
3.4	Superfícies NURBS . . . . .	20
3.4.1	Definição . . . . .	21

---

3.4.2	Propriedades . . . . .	21
3.4.3	Derivadas . . . . .	21
<b>4</b>	<b>Estratégia Adotada</b>	<b>23</b>
4.1	Introdução . . . . .	23
4.2	Mapeamentos Transfinitos . . . . .	24
4.3	Geração do Espaço Paramétrico . . . . .	27
4.4	Recuperação de Valores e Derivadas . . . . .	28
4.4.1	Recuperação Usando Coordenadas Baricêntricas . . . . .	29
4.4.2	Recuperação Usando SPR . . . . .	31
<b>5</b>	<b>Exemplos</b>	<b>36</b>
<b>6</b>	<b>Conclusão e Trabalhos Futuros</b>	<b>38</b>

# Lista de Figuras

1.1	Espaço Paramétrico e Superfície Paramétrica. . . . .	1
1.2	Superfície Paramétrica Triangular. . . . .	2
2.1	Exemplo de uma curva. . . . .	4
2.2	Círculo unitário e derivadas. . . . .	6
2.3	Curvas de Bézier. . . . .	8
2.4	Polinômios de Bernstein de grau 3. . . . .	8
2.5	Funções básicas não nulas de grau 2 com $U = \{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$ . . . . .	10
2.6	Curva B-spline e seu polígono de controle correspondente a $u \in [u_3, u_4)$ com $U = \{0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1\}$ . . . . .	13
2.7	Curva NURBS com variação do peso associado ao ponto de controle $P_1$ . . . . .	15
3.1	Exemplo de uma Superfície: Torus. . . . .	16
3.2	Esfera de raio unitário. . . . .	17
4.1	Procedimento para geração de uma superfície paramétrica triangular. . . . .	24
4.2	Mapeamento transfinito. . . . .	25
4.3	Discretização das curvas de contorno e geração da malha. . . . .	27
4.4	Malha de triângulos e espaço paramétrico gerado usando a PFS. . . . .	28
4.5	Recuperação de valores usando coordenadas baricêntricas. . . . .	29
4.6	Superfície triangular recuperada por coordenadas baricêntricas. . . . .	30
4.7	<i>Patch</i> gerado por um nó do campo. . . . .	32
4.8	Espaço paramétrico triangulado dividido em <i>patches</i> . . . . .	33
4.9	Determinação de <i>patches</i> influentes para cálculo das médias $M_i$ . . . . .	34
4.10	Superfície triangular recuperada por SPR. . . . .	35
5.1	Discretização das curvas de contorno. . . . .	36
5.2	Geração da malha de elementos triangulares e espaço paramétrico obtido. . . . .	37
5.3	Superfícies reconstruídas. . . . .	37

# Capítulo 1

## Introdução

### 1.1 Modelagem de Curvas e Superfícies

A Computação Gráfica é definida atualmente como a área que estuda métodos e técnicas para processamento e visualização de objetos gráficos. Do ponto de vista matemático, um objeto gráfico é um subconjunto  $S \subset \mathbb{R}^m$  e uma função  $f: S \subset \mathbb{R}^m \rightarrow \mathbb{R}^n$ , onde  $S$  é chamado suporte geométrico e  $f$  é chamada de função de atributos do objeto gráfico (Gomes & Velho 2003). A parte da Computação Gráfica que se preocupa com a representação e estruturação de dados associados a objetos gráficos é denominada Modelagem Geométrica.

Mais especificamente, a Modelagem Geométrica trata do problema da criação e manipulação da geometria e topologia dos objetos gráficos, chamados normalmente de modelos. As *curvas e superfícies paramétricas* são exemplos de modelos de extrema importância para a modelagem em diversas áreas da ciência e da indústria.

As curvas paramétricas são descritas por aplicações contínuas  $\alpha: I \rightarrow \mathbb{R}^3$ , onde  $I \subset \mathbb{R}$ . Já as superfícies paramétricas são definidas por parametrizações  $\varphi: D \rightarrow \mathbb{R}^3$  que mapeiam  $(u, v) \rightarrow (x^1(u, v), x^2(u, v), x^3(u, v))$ , onde  $D \subset \mathbb{R}^2$  é um conjunto aberto denominado espaço paramétrico (Figura 1.1). Estes objetos serão estudados nos capítulos 2 e 3.

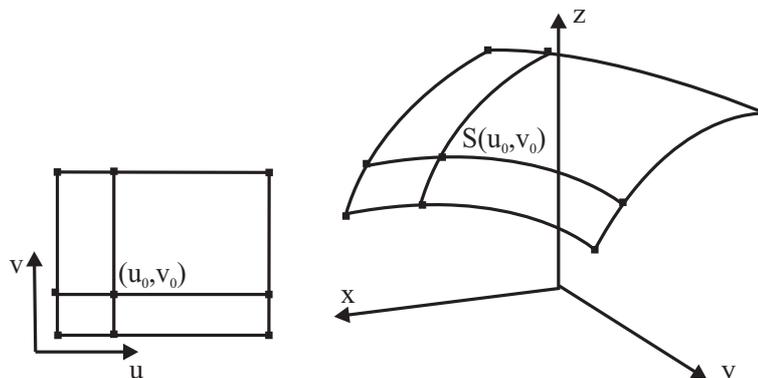


Figura 1.1: Espaço Paramétrico e Superfície Paramétrica.

Entre as diversas aplicações da modelagem de curvas e superfícies, destacam-se o desenvolvimento de aplicações em Engenharia Civil, Geologia, Medicina, Biologia e Química, além do seu uso na indústria do entretenimento como, por exemplo, o desenvolvimento de jogos e mundos virtuais e, finalmente, aplicações CAD/CAM (*Computer Aided Design / Computer Aided Manufacturing*).

## 1.2 Objetivos

Dentre as superfícies paramétricas mais utilizadas na modelagem geométrica de problemas reais de Engenharia estão as *superfícies triangulares* (Figura 1.2). Estas apresentam certas particularidades que dificultam suas modelagens usando técnicas convencionais, como NURBS (*Non Uniform Rational B-Splines*), e são caracterizadas por possuírem três curvas de bordo delimitando seu contorno.

Este trabalho apresenta uma técnica para reconstrução de superfícies paramétricas triangulares. A técnica foi desenvolvida em três etapas. A primeira etapa é responsável pela discretização das curvas de contorno e geração de uma malha de elementos triangulares, apresentada por Gordon & Hall (1973). Na segunda etapa é criado um espaço paramétrico discreto para a malha obtida no primeiro módulo (Floater 1997). A última etapa realiza a recuperação de pontos da superfície e apresenta uma adaptação da técnica SPR (*Superconvergent Patch Recovery*, Zienkiewicz & Zhu (1994)), de modo a manter a suavidade da superfície reconstruída.

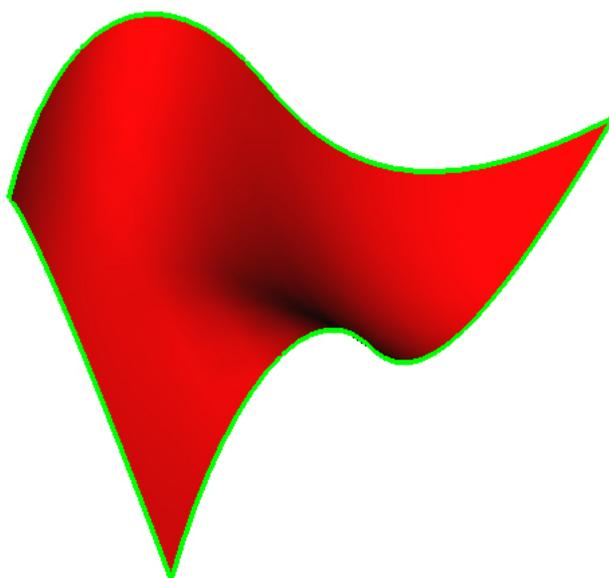


Figura 1.2: Superfície Paramétrica Triangular.

Assim, as principais contribuições deste trabalho estão na integração das três etapas descritas anteriormente e na adaptação da técnica SPR de modo a manter a suavidade da superfície.

Além disso, foi implementado na terceira etapa um procedimento para recuperação de pontos da superfície usando coordenadas baricêntricas, com o objetivo de comparar os resultados com o SPR.

Em termos de implementação, foi desenvolvida uma biblioteca para modelagem de superfícies triangulares, usando a linguagem C++ (Stroustrup 1991) e a API OpenGL (Shreiner et al. 2003). Essa biblioteca promove a integração dos três módulos citados acima, possibilitando a recuperação usando coordenadas baricêntricas e a recuperação usando SPR. Além disso, foi desenvolvido uma aplicação para visualização e modelagem de superfícies. Tal aplicação faz uso da biblioteca desenvolvida neste trabalho.

### 1.3 Estrutura do Trabalho

Este trabalho está estruturado da seguinte forma:

**Capítulo 2.** Contém uma introdução ao estudo de curvas, além de um breve resumo da literatura existente envolvendo as *curvas de Bézier*, *B-spline* e *NURBS*.

**Capítulo 3.** Exibe uma introdução ao estudo das superfícies e descreve as principais características das *superfícies de Bézier*, *B-spline* e *NURBS*, tornando claras as possibilidades e limitações dessas superfícies.

**Capítulo 4.** Faz um levantamento dos principais trabalhos envolvidos nas etapas de reconstrução das superfícies triangulares e apresenta a estratégia adotada neste trabalho.

**Capítulo 5.** Contém exemplos e ilustrações de todos os passos da técnica descrita neste trabalho para reconstrução de superfícies triangulares.

**Capítulo 6.** Apresenta as conclusões sobre o trabalho desenvolvido e idéias para trabalhos futuros.



qualquer no plano é única, ou seja, uma curva é bem definida por uma e somente uma equação.

Um exemplo interessante é a circunferência de raio unitário, que é bem definida pela equação  $x^2 + y^2 - 1 = 0$ .

A descrição implícita tem como maior limitação a impossibilidade de representar curvas com mais de duas dimensões, se restringindo apenas à representação de curvas planas. Suas vantagens são a possibilidade de representação de formas infinitas (e.g. uma linha reta dada pela equação  $ax + by + c = 0$ ), e a facilidade para avaliar se um ponto pertence a uma curva.

### Descrição paramétrica

**Definição 2.1.1.** *Uma curva contínua no espaço  $\mathbb{R}^n$  é uma aplicação contínua  $\alpha : I \rightarrow \mathbb{R}^n$ , definida num intervalo  $I \subset \mathbb{R}$ . A aplicação  $\alpha$ , dada por  $\alpha(t) = (\alpha^1(t), \alpha^2(t), \dots, \alpha^n(t))$ , é contínua, se cada função coordenada  $\alpha^i : I \rightarrow \mathbb{R}$  é uma função contínua.*

O círculo unitário definido implicitamente (Figura 2.2), é também definido pela função paramétrica  $\alpha(t) = (\sin(t), \cos(t)), 0 \leq t \leq 2\pi$ .

Exemplos de curvas paramétricas são as curvas de Bézier, B-splines e NURBS, funções paramétricas que serão abordadas mais adiante e fornecem métodos intuitivos para design, além de estabilidade numérica. Portanto, de agora em diante, este trabalho foca-se na descrição paramétrica de curvas.

### 2.1.2 Derivação de Curvas Paramétricas

Uma curva  $\alpha(t)$  é diferenciável se cada função  $\alpha^i(t) : \mathbb{R} \rightarrow \mathbb{R}$  for diferenciável. De modo geral, a  $n$ -ésima derivada de uma curva é dada por

$$\alpha^{(n)}(t) = \left( \frac{d^n \alpha^1}{dt^n}, \frac{d^n \alpha^2}{dt^n}, \frac{d^n \alpha^3}{dt^n} \right). \quad (2.1)$$

Geometricamente,  $\alpha'(t)$  representa o vetor tangente à curva. A Figura 2.2 exibe o círculo de raio unitário com suas derivadas.

### 2.1.3 Classes de Funções

A descrição de curvas através de funções paramétricas fornece uma infinidade de possibilidades para modelagem. Entretanto, para a otimização da modelagem computacional de curvas, deve-se restringir essas funções a uma classe de funções que cumpram alguns requisitos importantes. Basicamente, uma classe de funções interessante deve ter flexibilidade para representar todos os tipos de curva que interessem ao usuário,

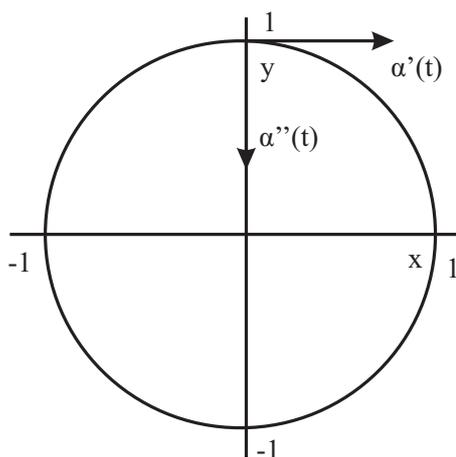


Figura 2.2: Círculo unitário e derivadas.

ser simples e matematicamente bem compreendida, além de prover facilidade, eficiência e precisão para o processamento no computador. Mais especificamente, o computador deve ser capaz de calcular pontos e derivadas com eficiência e ser invulnerável a problemas de arredondamento. Além disso, a descrição dessas funções no computador deve requerer pouca memória.

Seguindo essa idéia, uma classe de funções que se encaixa parcialmente nesses requisitos são os polinômios, que são usados para a construção das Curvas de Bézier. Entretanto, essas curvas não podem representar todas os tipos de formas geométricas, como é o caso das cônicas. Para tanto, é necessário o uso de uma classe de funções mais poderosa: as funções racionais. Essa classe de funções é definida como a divisão de dois polinômios e é usada para construir os conceitos de Curvas de Bézier Racionais.

Além dessas duas classes de funções, existem classes mais poderosas que fornecem mais precisão e melhor performance computacional para desenvolver algumas formas geométricas. São os chamados polinômios por partes e polinômios racionais por partes, que são funções definidas de acordo com intervalos. Usando essas classes de funções, foram desenvolvidas as Curvas B-spline e as NURBS. Detalhes sobre algumas destas representações são mostradas nas seções seguintes, tendo como referência Piegl & Tiller (1997).

## 2.2 Curvas de Bézier

### 2.2.1 Definição

As Curvas de Bézier são definidas por um conjunto de pontos de controle  $\{P_0, P_1, \dots, P_n\}$ , tal que  $n$  é o grau da curva. Os pontos  $P_0$  e  $P_n$  definem o início e o final da curva, e os pontos intermediários influenciam o caminho da curva, de modo que a alteração de qualquer desses pontos influencia toda a curva, sendo seu controle global. Além dos pontos

de controle, as Curvas de Bézier usam um conjunto de funções básicas, denominadas *polinômios de Bernstein*.

Uma Curva de Bézier de grau  $n$  é definida por

$$C(u) = \sum_{i=0}^n B_{i,n}(u)P_i \quad 0 \leq u \leq 1, \quad (2.2)$$

onde  $B_{i,n}(u)$  são os polinômios de Bernstein de grau  $n$ , e  $P_i$  são os pontos de controle. Os polinômios de Bernstein são dados por

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}. \quad (2.3)$$

### 2.2.2 Propriedades

- o polígono formado pelos seus pontos de controle  $\{P_0, P_1, \dots, P_n\}$ , chamada de poligonal de controle, aproxima a forma da curva (Figura 2.3(a));
- os pontos de controle das extremidades são também extremidades da curva, ou seja:  $P_0 = C(0)$  e  $P_n = C(1)$ ;
- a direção do vetor tangente da curva nas extremidades é paralelo aos vetores  $P_1 - P_0$  e  $P_n - P_{n-1}$ ;
- a curva está contida no fecho convexo formado pelos pontos de controle  $\{P_0, P_1, \dots, P_n, P_0\}$  (Figura 2.3(b));
- nenhum plano intercepta a curva mais vezes do que intersecta o polígono de controle, consolidando a primeira propriedade;
- transformações de rotação, translação e escala podem ser aplicados diretamente nos pontos de controle, tendo o mesmo efeito de transformações aplicadas diretamente à curva.

Os polinômios de Bernstein possuem as seguintes propriedades:

- não negatividade:  $B_{i,n}(u) \geq 0$  para todo  $i, n$  e  $0 \leq u \leq 1$ ;
- partição da unidade:  $\sum_{i=0}^n B_{i,n}(u) = 1$  para todo  $0 \leq u \leq 1$ ;
- $B_{0,n}(0) = B_{n,n}(1) = 1$ ;
- $B_{i,n}(u)$  tem exatamente um máximo no intervalo  $[0, 1]$ , quando  $u = \frac{i}{n}$ ;
- simetria: para todo  $n$ ,  $B_{i,n}(u)$  é simétrico em relação a  $u = \frac{1}{2}$ ;

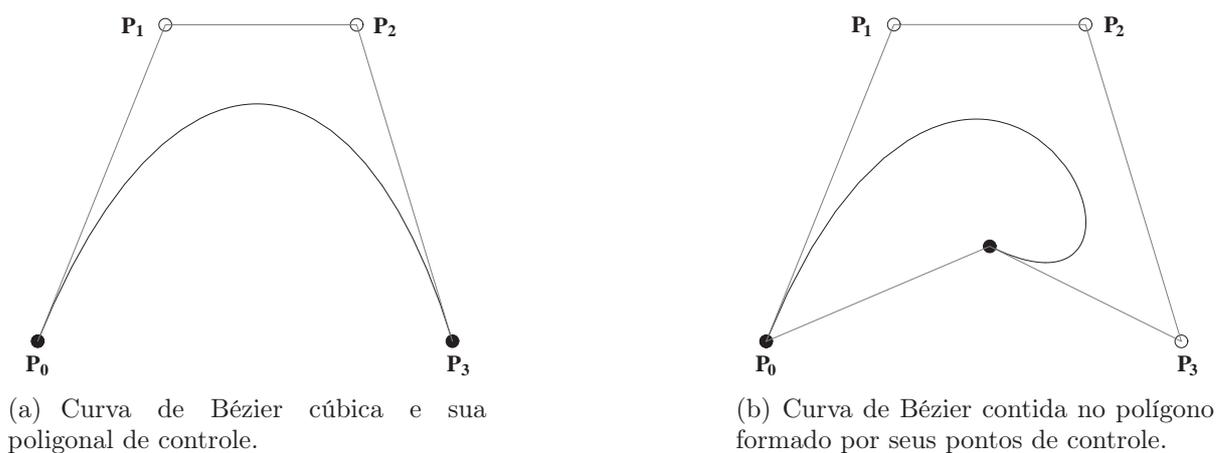


Figura 2.3: Curvas de Bézier.

- definição recursiva: Os polinômios de Bernstein podem ser calculados com algoritmos mais simples com a equação

$$B_{i,n}(u) = (1 - u)B_{i,n-1}(u) + uB_{i-1,n-1}(u) \quad (2.4)$$

com  $B_{i,n}(u) \equiv 0$  se  $i < 0$  ou  $i > n$ .

A Figura 2.4 mostra os polinômios de Bernstein de grau 3.

### 2.2.3 Derivadas

As derivadas das Curvas de Bézier são obtidas a partir das derivadas dos polinômios de Bernstein.

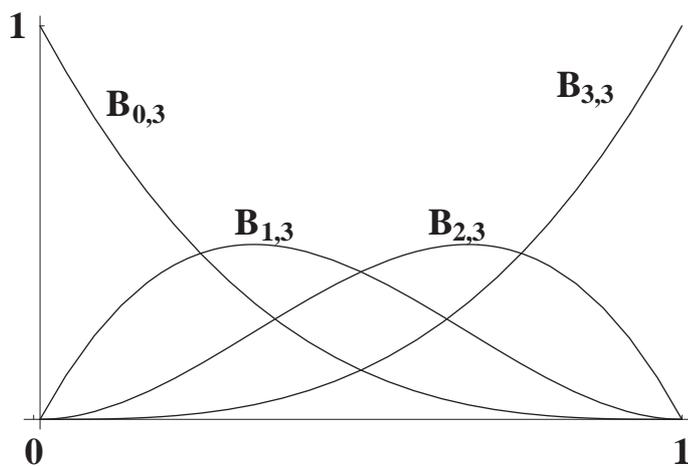


Figura 2.4: Polinômios de Bernstein de grau 3.

A derivada de um polinômio de Bernstein é dada por

$$B'_{i,n}(u) = \frac{dB_{i,n}(u)}{du} = n(B_{i-1,n-1}(u) - B_{i,n-1}(u)), \quad (2.5)$$

com  $B_{-1,n-1}(u) \equiv B_{n,n-1}(u) \equiv 0$ .

Derivando-se a equação da Curva de Bézier e substituindo-se a derivada do polinômio de Bernstein pela Equação (2.5), chega-se com facilidade à derivada da Curva de Bézier, que é dada por

$$C'(u) = n \sum_{i=0}^{n-1} B_{i,n-1}(u)(P_{i+1} - P_i). \quad (2.6)$$

## 2.3 Funções Básicas B-spline

Devido às limitações das funções polinomiais, foi necessário o desenvolvimento de técnicas mais poderosas para modelagem de curvas e superfícies. Nesse sentido, foram desenvolvidas as *Curvas B-spline* e posteriormente as *Curvas NURBS (Non Uniform Rational B-Splines)*.

Tanto as Curvas B-Spline como as Curvas NURBS utilizam *Funções Básicas B-spline*, que serão descritas a seguir.

### 2.3.1 Definição

Existem diversas definições para as Funções Básicas B-spline. Porém, este documento apresenta apenas a definição recursiva, que é a mais interessante para implementação computacional.

Seja  $U = \{u_0, \dots, u_m\}$  uma sequência não-decrescente de números reais. Para cada  $i = 0, \dots, m - 1$ , chamaremos  $u_i$  de nó e  $U$  de vetor de nós. A  $i$ -ésima Função Básica B-spline de grau  $p$  é dada por

$$N_{i,0}(u) = \begin{cases} 1 & \text{se } u_i \leq u < u_{i+1} \\ 0 & \text{em outro caso,} \end{cases}$$

onde

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \quad (2.7)$$

Observa-se que as Funções Básicas B-spline são polinômios por partes que só tem valor não nulo quando  $u \in [u_i, u_{i+1})$ . Dessa forma, apenas quando  $u$  pertencer a esse intervalo, o valor final da função terá seu valor alterado, visto que, de acordo com a Equação (2.7), o valor final é a combinação linear de várias funções  $N_{i,p}$ , exceto quando  $p = 0$ . Portanto, o vetor de nós desempenha papel fundamental para definir qual intervalo será afetado. O caráter local dessas funções pode ser claramente observado na Figura 2.5.

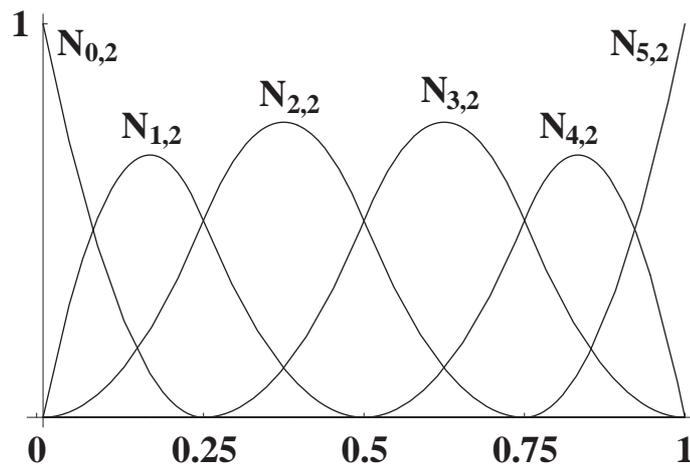


Figura 2.5: Funções básicas não nulas de grau 2 com  $U = \{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$ .

Define-se como *multiplicidade de um nó*, a quantidade de vezes que ele aparece no vetor de nós. Na Figura 2.5, o nó 0 do vetor  $U$  tem multiplicidade 3, enquanto o nó 0.5 tem multiplicidade 1. O vetor de nós pode ser classificado como não periódico quando ele tem a forma

$$U = \{\underbrace{a, \dots, a}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{b, \dots, b}_{p+1}\} \quad (2.8)$$

ou periódico caso contrário. Pode-se também classificá-lo como uniforme, quando os nós interiores forem igualmente espaçados, e não uniforme, quando essa condição não for satisfeita.

As Funções Básicas B-splines generalizam os polinômios de Bernstein. Isso pode ser facilmente deduzido fazendo-se  $U = \{\underbrace{a, \dots, a}_{p+1}, \underbrace{b, \dots, b}_{p+1}\}$ .

Um outro detalhe importante da definição é que frequentemente chega-se à divisão  $0/0$ . Nesse caso, por definição, o resultado da divisão é zero.

### 2.3.2 Propriedades

- $N_{i,p}$  é nula fora do intervalo  $[u_i, u_{i+p+1})$  (Figura 2.5);
- para qualquer intervalo  $[u_i, u_{i+1})$ , apenas as funções  $N_{i-p,p}, \dots, N_{i,p}$  podem ser não nulas;
- não negatividade:  $N_{i,p}(u) \geq 0$  para todo  $i, p$  e  $u$ ;
- partição da unidade: para cada intervalo  $[u_i, u_{i+1}]$ ,  $\sum_{i=j-p}^j N_{i,p}(u) = 1$  para todo  $u \in [u_i, u_{i+1}]$ ;
- No interior de cada intervalo  $[u_i, u_{i+1})$ , todas as derivadas de  $N_{i,p}(u)$  existem. Em um nó,  $N_{i,p}(u)$  é  $p - k$  continuamente diferenciável, onde  $k$  é a multiplicidade

do nó. Portanto, aumentando o grau das funções, aumentamos a continuidade, e aumentando a multiplicidade do nó, a continuidade diminui.

### 2.3.3 Derivadas

A derivada da Função Básica B-spline é dada por

$$N'_{i,p}(u) = \frac{p}{u_{i+p} - u_i} N_{i,p-1}(u) - \frac{p}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \quad (2.9)$$

A Equação (2.9) pode ser diferenciada continuamente, chegando a uma forma geral para a  $k$ -ésima derivada de  $N_{i,p}(u)$ , dada por

$$N_{i,p}^{(k)}(u) = p \left( \frac{N_{i,p-1}^{(k-1)}}{u_{i+p} - u_i} - \frac{N_{i+1,p-1}^{(k-1)}}{u_{i+p+1} - u_{i+1}} \right). \quad (2.10)$$

Além destas duas equações, existe uma outra equação que generaliza a Equação (2.9), bastante usada em algoritmos, definida por

$$N_{i,p}^k = \frac{p!}{(p-k)!} \sum_{j=0}^k a_{k,j} N_{i+j,p-k}, \quad (2.11)$$

onde

$$\begin{aligned} a_{0,0} &= 1 \\ a_{k,0} &= \frac{a_{k-1,0}}{u_{i+p-k+1} - u_i} \\ a_{k,j} &= \frac{a_{k-1,j} - a_{k-1,j-1}}{u_{i+p+j-k+1} - u_{i+j}} j = 1, \dots, k-1 \\ a_{k,k} &= \frac{-a_{k-1,k-1}}{u_{i+p+1} - u_{i+k}}. \end{aligned}$$

Na Equação (2.11),  $k$  não deve exceder  $p$ . Para  $k > p$ , a derivada se anula.

## 2.4 Curvas B-spline

### 2.4.1 Definição

As Curvas B-Spline são formadas por  $n+1$  pontos de controle e um conjunto de Funções Básicas B-spline de grau  $p$ , que são definidas por um vetor não periódico e não uniforme, com  $m+1$  nós (Eq. (2.8)). Uma Curva B-spline é definida por

$$C(u) = \sum_{i=0}^n N_{i,p}(u) P_i \quad a \leq u \leq b. \quad (2.12)$$

### 2.4.2 Propriedades

- sendo  $p$  o grau da curva,  $n + 1$  a quantidade de pontos de controle e  $m + 1$  o número de nós, podemos relacionar essas variáveis com a equação

$$m = n + p + 1; \quad (2.13)$$

- os pontos de controle das extremidades são também extremidades da curva, ou seja:  $P_0 = C(0)$  e  $P_n = C(1)$ ;
- uma transformação afim pode ser aplicada diretamente aos pontos de controle com o mesmo efeito de transformações aplicadas diretamente à curva;
- esquema de modificação local: uma alteração em um ponto de controle  $P_i$  altera a curva  $C(u)$  apenas no intervalo  $[u_i, u_{i+p+1}]$ , pois  $N_{i,p}(u) = 0$  para  $u \notin [u_i, u_{i+p+1}]$ ;
- um ponto  $C(u)$  da curva está contido no fecho convexo formado pelos pontos de controle que lhe afetam diretamente, ou seja, se  $u \in [u_i, u_{i+1})$ , o ponto está contido no fecho convexo formado pelos pontos  $P_{i-p}, \dots, P_i$ . A Figura 2.6 mostra uma curva B-spline cúbica e o fecho convexo formado pelos pontos de controle  $P_0, P_1, P_2$  e  $P_3$ , de modo que o pedaço da curva correspondente a  $u \in [u_3, u_4)$  está contido neste fecho convexo;
- o polígono de controle dá uma aproximação linear da curva, de modo que essa aproximação é aperfeiçoada com procedimentos de inserção de nó e elevação de grau (ver Piegl & Tiller 1997);
- nenhum plano intersepta a curva mais vezes do que intersepta o polígono de controle;
- a continuidade da curva está diretamente relacionada com a continuidade das Funções Básicas B-splines, sendo  $C(u)$  infinitamente diferenciável no interior dos intervalos  $[u_i, u_{i+1})$  e pelo menos  $p - k$  continuamente diferenciável em um nó de multiplicidade  $k$ .

### 2.4.3 Derivadas

Seja  $C(u)$  uma curva B-spline de grau  $p$  e vetor de nós  $U$ . A  $k$ -ésima derivada de  $C(u)$  será representada com a notação  $C^{(k)}(u)$ . As  $k$ -ésimas derivadas das Funções Básicas B-spline podem ser obtidas através da Equação (2.10) para chegar à fórmula

$$C^{(k)}(u) = \sum_{i=0}^n N_{i,p}^{(k)}(u) P_i. \quad (2.14)$$

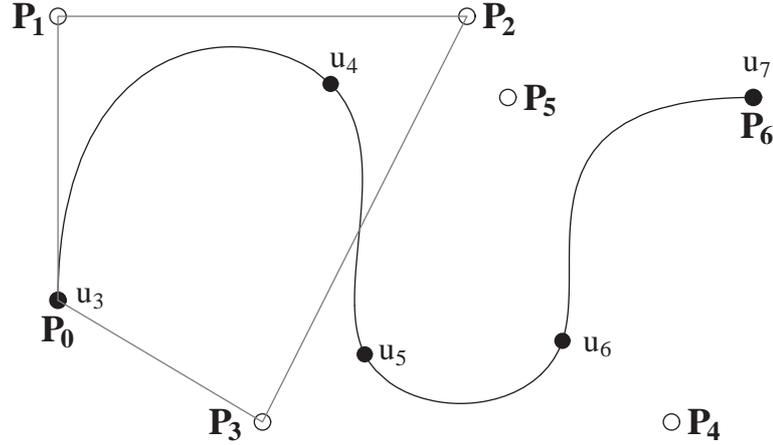


Figura 2.6: Curva B-spline e seu polígono de controle correspondente a  $u \in [u_3, u_4)$  com  $U = \{0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1\}$ .

Diferenciando-se a Equação (2.12), chega-se a uma nova curva  $C'(u)$  de grau  $p-1$ . Essa nova curva deve ter  $n$  pontos de controle e um vetor de nós que é obtido desprezando-se o primeiro e o último elemento do vetor de nós  $U$ . Então

$$U' = \underbrace{\{a, \dots, a\}}_p, u_{p+1}, \dots, u_{m-p-1}, \underbrace{\{b, \dots, b\}}_p, \quad (2.15)$$

de modo que  $U'$  terá  $m-1$  nós. Os pontos de controle de  $C'(u)$  são obtidos pela equação

$$Q_i = p \frac{P_{i+1} - P_i}{u_{i+p+1} - u_{i+1}}, \quad (2.16)$$

onde  $P_i$  é ponto de controle da curva original. Finalmente, a derivada da curva  $C(u)$  é representada pela equação

$$C'(u) = \sum_{i=0}^{n-1} N_{i,p-1}(u) Q_i. \quad (2.17)$$

Derivando-se repetidamente a Equação (2.17), chega-se finalmente à forma geral dada pela equação

$$C^{(k)}(u) = \sum_{i=0}^{n-k} N_{i,p-k}(u) P_i^{(k)}, \quad (2.18)$$

onde os pontos de controle  $P_i^k$  são dados por

$$P_i^k = \begin{cases} P_i & k = 0 \\ \frac{p-k+1}{u_{i+p+1} - u_{i+k}} (P_{i+1}^{(k-1)} - P_i^{(k-1)}) & k > 0, \end{cases} \quad (2.19)$$

e o vetor de nós pela equação

$$U^{(k)} = \underbrace{\{a, \dots, a\}}_{p-k+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{\{b, \dots, b\}}_{p-k+1} \quad (2.20)$$

Provas podem ser encontradas em Piegl & Tiller (1997).

## 2.5 Curvas NURBS

As Curvas NURBS (*Non Uniform Rational B-Splines*), são assim nomeadas por se tratarem de curvas definidas por Funções Básicas Racionais que utilizam vetores de nós não uniformes. Dentre suas vantagens, destacam-se as técnicas para modelar círculos e cônicas em geral. Para tanto, é necessária a introdução do conceito de peso. As curvas NURBS têm associado a cada ponto de controle um valor real que determina o peso daquele ponto de controle. Dessa forma, quanto maior o peso de um ponto de controle, mais influência este ponto exerce sobre o pedaço da curva no qual atua.

### 2.5.1 Definição

Uma curva NURBS de grau  $p$  é definida por

$$C(u) = \frac{\sum_{i=0}^n N_{i,p}(u)w_i P_i}{\sum_{i=0}^n N_{i,p}(u)w_i} \quad a \leq u \leq b \quad (2.21)$$

onde  $\{P_i\}$  são os pontos de controle da curva,  $\{N_{i,p}\}$  são Funções Básicas B-spline de grau  $p$  e cada valor real  $w_i$  é um peso associado ao ponto de controle  $P_i$ .

As Funções Básicas Racionais são definidas pelo quociente dado pela equação

$$R_{i,p}(u) = \frac{N_{i,p}(u)w_i}{\sum_{j=0}^n N_{j,p}(u)w_j}. \quad (2.22)$$

Finalmente, a equação da curva NURBS pode ser reescrita, substituindo-se a Equação (2.22) na Equação (2.21), alcançando a forma

$$C(u) = \sum_{i=0}^n R_{i,p}(u)P_i \quad a \leq u \leq b. \quad (2.23)$$

Usando o conceito de coordenadas homogêneas, é possível chegar a uma definição eficiente para as curvas NURBS. Uma curva NURBS racional contida no espaço  $\mathbb{R}^3$  será representada como uma curva polinomial no espaço  $\mathbb{R}^4$ , de modo que os pontos de controle  $P_i = (x_i, y_i, z_i)$  serão usados para construir pontos de controle com peso, dados pelas coordenadas  $P_i^w = (w_i x_i, w_i y_i, w_i z_i, w_i)$ . Dessa forma, a Equação (2.23) é usada para

chegar à forma

$$C^w(u) = \sum_{i=0}^n N_{i,p}(u) P_i^w \quad a \leq u \leq b. \quad (2.24)$$

Naturalmente, qualquer ponto da curva no espaço  $\mathbb{R}^3$  pode ser obtido dividindo-se as três primeiras coordenadas de  $C^w(u)$  pela quarta.

Para mostrar que as curvas NURBS generalizam as curvas B-spline, é suficiente definir  $w_i = 1$  para todo  $i$ . Consequentemente, também generalizam as curvas de Bézier.

### 2.5.2 Propriedades

As curvas NURBS herdam todas as propriedades das curvas B-spline. Porém, além da edição local de pontos de controle, existe agora a possibilidade de alteração de pesos, o que implica uma nova ferramenta para modelagem de curvas. A Figura 2.7 mostra o que acontece com a curva quando um ponto de controle tem seu peso alterado.

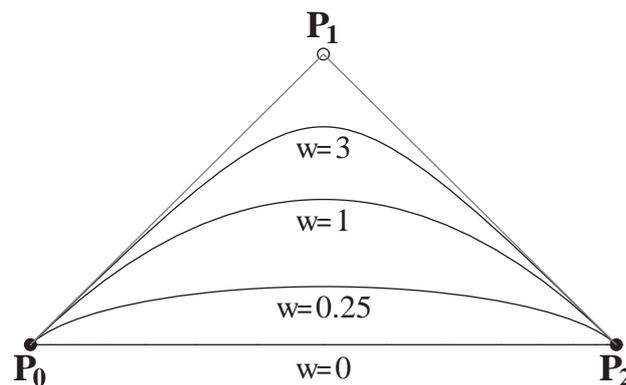


Figura 2.7: Curva NURBS com variação do peso associado ao ponto de controle  $P_1$

### 2.5.3 Derivadas

De acordo com a definição dada pela Equação (2.23), as curvas NURBS podem ser tratadas como curvas não racionais no espaço  $\mathbb{R}^4$ . Desse modo, as derivadas de  $C^w(u)$  correspondem às derivadas das curvas B-spline, dadas pelas Equações (2.14) à (2.20).

# Capítulo 3

## Modelagem de Superfícies

### 3.1 Introdução

Este capítulo trata dos conceitos que envolvem a modelagem de superfícies. Serão abordados principalmente os conceitos das superfícies paramétricas, além de três técnicas para modelagem de superfícies que derivam dos conceitos de modelagem de curvas já apresentados.

Neste trabalho, serão consideradas as superfícies com dimensão 2. Uma bolha de sabão, uma folha de papel com qualquer deformação, canos e balões são apenas alguns exemplos de superfícies. A Figura 3.1 exibe uma superfície denominada Torus.

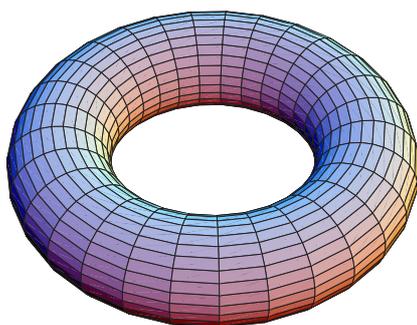


Figura 3.1: Exemplo de uma Superfície: Torus.

#### 3.1.1 Técnicas de Descrição de Superfícies

Assim como as curvas, as superfícies podem ser descritas implícita ou parametricamente.

### Descrição Implícita

Uma superfície é descrita implicitamente por equações da forma  $f(x, y, z) = 0$ , e é definida como o conjunto solução desta equação, ou seja, cada ponto  $p \in \mathbb{R}^3$  que satisfaz a equação implícita pertence à superfície. Um exemplo desta descrição é ilustrada pela equação implícita da esfera de raio unitário, dada por  $x^2 + y^2 + z^2 - 1 = 0$  (Figura 3.2).

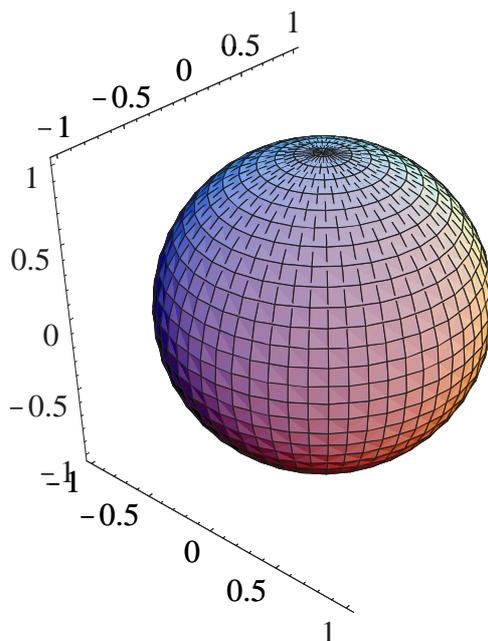


Figura 3.2: Esfera de raio unitário.

### Descrição Paramétrica

A descrição paramétrica de superfícies é definida por uma aplicação

$$x : D \rightarrow \mathbb{R}^3(u, v) \rightarrow (x^1(u, v), x^2(u, v), x^3(u, v)), \quad (3.1)$$

onde  $D \subset \mathbb{R}^2$  é um conjunto aberto. Dessa forma, a aplicação  $x$  retorna as coordenadas de cada ponto da superfície.

A esfera de raio unitário pode ser descrita pela função paramétrica  $x(u, v) = (\cos(u) \cos(v), \sin(u) \cos(v), \sin(v))$ , com  $0 \leq u \leq 2\pi$  e  $-\pi/2 \leq v \leq \pi/2$ .

É interessante observar que, fixando-se  $u = u_0$ , tem-se que  $x(u_0, v)$  descreve uma curva. O mesmo acontece fixando-se o parâmetro  $v = v_0$ .

As funções paramétricas serão adotadas neste trabalho para a descrição de superfícies.

### 3.1.2 Derivação de Superfícies

Sendo  $x(u, v)$  uma superfície paramétrica, é possível calcular suas derivadas parciais em relação à  $u$  e  $v$ . Então, fixando-se  $v$ , tem-se  $x_u = \left( \frac{dx^1}{du}, \frac{dx^2}{du}, \frac{dx^3}{du} \right)$ . Analogamente, fixando-se  $u$ , tem-se  $x_v = \left( \frac{dx^1}{dv}, \frac{dx^2}{dv}, \frac{dx^3}{dv} \right)$ .

Essas derivadas representam o vetor tangente às curvas  $x_u(v)$  e  $x_v(u)$ , respectivamente. O vetor normal à superfície pode ser encontrado através do produto vetorial dos vetores  $x_u$  e  $x_v$ .

## 3.2 Superfícies de Bézier

As superfícies de Bézier estendem o conceito de Curva de Bézier. Uma superfície de Bézier é definida tomando-se uma rede bidirecional de pontos de controle e o produto dos polinômios de Bernstein associados. Então,

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_{i,n}(u) B_{j,m}(v) P_{i,j} \quad 0 \leq u, v \leq 1. \quad (3.2)$$

Fixando-se  $u = u_0$ , tem-se que  $S(u_0, v) = C_{u_0}(v)$ , de modo que  $C_{u_0}$  é uma Curva de Bézier denominada isocurva, que fica totalmente contida na superfície.

As propriedades das Superfícies de Bézier são semelhantes às propriedades das Curvas de Bézier.

## 3.3 Superfícies B-Spline

### 3.3.1 Definição

As superfícies B-Spline são obtidas a partir de uma rede bidirecional de pontos de controle, dois vetores de nós que representam cada uma das duas direções, e o produto das funções B-Spline associadas. Assim, as superfícies B-Spline são definidas pela equação:

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) P_{i,j} \quad (3.3)$$

com

$$U = \left\{ \underbrace{0, \dots, 0}_{p+1}, u_{p+1}, \dots, u_{r-p-1}, \underbrace{1, \dots, 1}_{p+1} \right\}$$

$$V = \left\{ \underbrace{0, \dots, 0}_{q+1}, u_{q+1}, \dots, u_{s-q-1}, \underbrace{1, \dots, 1}_{q+1} \right\}$$

de modo que  $r = n + p + 1$  e  $s = m + q + 1$ .

Também é possível obter isocurvas da mesma maneira apresentada na seção sobre superfícies de Bézier.

### 3.3.2 Propriedades

As propriedades do produto das funções básicas são provenientes das propriedades das funções básicas com uma variável:

- não negatividade:  $N_{i,p}(u)N_{j,q}(v) \geq 0$  para todo  $i, j, p, q, u, v$ ;
- partição da unidade:  $\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u)N_{j,q}(v) = 1$  para todo  $(u, v) \in [0, 1] \times [0, 1]$ ;
- $N_{i,p}(u)N_{j,q}(v) = 0$  se  $(u, v)$  está fora do retângulo  $[u_i, u_{i+p+1}] \times [v_j, v_{j+q+1}]$ ;
- em qualquer retângulo  $[u_{i_0}, u_{i_0+1}] \times [v_{j_0}, v_{j_0+1}]$  apenas as funções  $N_{i,p}(u)N_{j,q}(v)$  com  $i_0 - p \leq i \leq i_0$  e  $j_0 - q \leq j \leq j_0$  podem ser não nulas;
- No interior de cada retângulo formado pelas retas definidas pelos valores  $u$  e  $v$  dos vetores de nós, todas as derivadas de  $N_{i,p}(u)N_{j,q}(v)$  existem. Em um nó  $u$  ou  $v$ , a função é  $p - k$  ou  $q - k$  continuamente diferenciável na direção de  $u$  ou  $v$ , onde  $k$  é a multiplicidade do nó.

Já as superfícies B-Spline têm as seguintes características:

- interpolação dos quatro pontos de controle nos cantos:  $S(0, 0) = P_{0,0}, S(1, 0) = P_{n,0}, S(0, 1) = P_{0,m}, S(1, 1) = P_{n,m}$ ;
- uma transformação afim pode ser aplicada diretamente aos pontos de controle com o mesmo efeito de transformações aplicadas diretamente à superfície;
- se  $(u, v) \in [u_{i_0}, u_{i_0+1}] \times [v_{j_0}, v_{j_0+1}]$ , então  $S(u, v)$  está no paralelepípedo formado pelos pontos de controle  $P_{i,j}, i_0 - p \leq i \leq i_0$  e  $j_0 - q \leq j \leq j_0$ .
- esquema de modificação local: uma alteração em um ponto de controle  $P_{i,j}$  afeta a superfície apenas no retângulo  $[u_i, u_{i+p+1}] \times [v_j, v_{j+q+1}]$ ;
- a continuidade e diferenciabilidade da superfície está diretamente relacionada com a continuidade das Funções Básicas B-splines.  $S(u, v)$  é  $p - k$  ou  $q - k$  continuamente diferenciável na direção de  $u$  ou  $v$  em um nó  $u$  ou  $v$  de multiplicidade  $k$ .

### 3.3.3 Derivadas

Assim como as curvas B-Spline, é possível calcular as derivadas parciais das superfícies B-Spline derivando-se as funções básicas. Desse modo, pode-se obter a derivada  $k$ -ésima na direção de  $u$  e  $l$ -ésima na direção de  $v$  usando a equação

$$\frac{\delta^{k+l}}{\delta^k u \delta^l v} S(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}^{(k)} N_{j,q}^{(l)} P_{i,j}. \quad (3.4)$$

Para chegar-se a uma representação formal para a derivada de uma superfície, é necessário encontrar  $S_u(u, v)$  e  $S_v(u, v)$ , para se obter

$$S_{uv}(u, v) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} N_{i,p-1}(u) N_{j,q-1}(v) P_{i,j}^{(1,1)}, \quad (3.5)$$

onde

$$P_{i,j}^{(1,1)} = q \frac{P_{i,j+1}^{(1,0)} - P_{i,j}^{(1,0)}}{v_{j+q+1} - v_{j+1}}$$

$$U^{(1)} = \left\{ \underbrace{0, \dots, 0}_p, u_{p+1}, \dots, u_{r-p-1}, \underbrace{1, \dots, 1}_p \right\}$$

$$V^{(1)} = \left\{ \underbrace{0, \dots, 0}_q, v_{q+1}, \dots, v_{s-q-1}, \underbrace{1, \dots, 1}_q \right\}$$

A partir da equação acima, é possível chegar à forma geral abaixo:

$$\frac{\delta^{k+l}}{\delta^k u \delta^l v} S(u, v) = \sum_{i=0}^{n-k} \sum_{j=0}^{m-l} N_{i,p-k}(u) N_{j,q-l}(v) P_{i,j}^{(k,l)} \quad (3.6)$$

onde

$$P_{i,j}^{(k,l)} = (q - l + 1) \frac{P_{i,j+1}^{(k,l-1)} - P_{i,j}^{(k,l-1)}}{v_{j+q+1} - v_{j+l}}$$

## 3.4 Superfícies NURBS

As superfícies NURBS apresentam características semelhantes às superfícies B-Splines. Assim como as curvas NURBS, é possível definir essas superfícies usando coordenadas homogêneas, como será mostrado na seção abaixo.

### 3.4.1 Definição

Uma superfície NURBS de grau  $p$  na direção de  $u$  e grau  $q$  na direção de  $v$  é definida através de uma rede bidirecional de pontos de controle  $P_{i,j}$  com seus respectivos pesos  $w_{i,j}$  e funções básicas B-Spline da seguinte maneira:

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j} P_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j}} \quad 0 \leq u, v \leq 1. \quad (3.7)$$

A Equação (3.7) pode ser reescrita como

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,j}(u, v) P_{i,j} \quad (3.8)$$

onde

$$R_{i,j}(u, v) = \frac{N_{i,p}(u) N_{j,q}(v) w_{i,j}}{\sum_{k=0}^n \sum_{l=0}^m R_{i,j}(u, v) P_{i,j}} \quad (3.9)$$

Finalmente, é possível representar uma superfície NURBS usando a mesma idéia de coordenadas homogêneas usada para representar curvas NURBS, como na Equação (2.24). Então

$$S^w(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) P_{i,j}^w. \quad (3.10)$$

É importante salientar que a forma expressa pela Equação (3.10) é a mais utilizada nos algoritmos computacionais.

### 3.4.2 Propriedades

As Funções Básicas Racionais apresentam as mesmas propriedades das Funções Básicas B-Spline. Consequentemente, as propriedades das superfícies NURBS são as mesmas das superfícies B-Spline.

É importante observar que as superfícies NURBS generalizam as superfícies de Bézier e superfícies B-Spline.

### 3.4.3 Derivadas

Primeiramente, é possível calcular as derivadas das superfícies NURBS na forma  $S^w(u, v)$  usando-se as Equação (3.4) e (3.6), definidas para as superfícies B-Spline. A partir dessas equações, é possível obter uma representação formal para as derivadas das superfícies NURBS na forma  $S(u, v)$ .

A superfície  $S(u, v)$  pode ser representada da seguinte maneira:

$$S(u, v) = \frac{w(u, v)S(u, v)}{w(u, v)} = \frac{A(u, v)}{w(u, v)}, \quad (3.11)$$

onde  $A(u, v)$  é o numerador da Equação (3.7).

A derivada de  $S(u, v)$  em um sentido  $\alpha$  é dada pela fórmula

$$S_\alpha(u, v) = \frac{A_\alpha(u, v) - w_\alpha(u, v)S(u, v)}{w(u, v)}. \quad (3.12)$$

Após algumas manipulações, chega-se, finalmente, à forma geral para calcular a derivada  $k$ -ésima no sentido de  $u$  e  $l$ -ésima no sentido de  $v$ :

$$S^{(k,l)} = \frac{1}{w} \left( A^{(k,l)} - \sum_{i=1}^k \binom{k}{i} \sum_{j=1}^l \binom{l}{j} w^{(i,j)} S^{(k-i,l-j)} \right). \quad (3.13)$$

# Capítulo 4

## Estratégia Adotada

### 4.1 Introdução

Este capítulo apresenta um procedimento para representação de superfícies paramétricas triangulares. O capítulo passado apresentou uma breve introdução ao estudo de superfícies e a descrição de algumas das principais técnicas para representação desses objetos gráficos. Essas técnicas são limitadas à modelagem de superfícies paramétricas retangulares.

Esta limitação se deve ao fato de que normalmente o espaço paramétrico de uma superfície NURBS é o quadrado  $[0, 1] \times [0, 1]$ , que é topologicamente compatível apenas com superfícies limitadas por quatro curvas de bordo. Isto confere às superfícies NURBS características topológicas que inviabilizam a modelagem de superfícies triangulares. Portanto, é necessário gerar um espaço paramétrico com suporte à topologia triangular.

Uma abordagem para resolver esse problema é apresentada por Qin & Terzopoulos (1997), e consiste no uso das *NURBS triangulares*. Usando conceitos de *patches triangulares*, é feita uma construção a partir dos *triângulos de Bézier*, utilizando coordenadas baricêntricas, para então definir as funções *Simplex Splines*. Essas funções são definidas sobre um conjunto de pontos no  $\mathbb{R}^2$  denominados nós da *Simplex Spline*. Esses pontos são calculados numa região do  $\mathbb{R}^2$ , e o resultado é um número real. Finalmente, essas funções *Simplex Splines* são utilizadas na construção dos fundamentos das *B-splines triangulares (DMS-splines)* e das NURBS triangulares, abordadas pelos autores.

Este capítulo descreve um novo procedimento para obtenção de uma superfície paramétrica a partir de três curvas de bordo. Esse procedimento é baseado em três etapas. A primeira refere-se à geração de uma malha triangular a partir de discretizações das três curvas de bordo. Isso é feito usando-se um algoritmo de mapeamento transfinito (Gordon & Hall 1973).

A segunda etapa é responsável pela geração de um espaço paramétrico discreto associado à malha triangular obtida na etapa anterior. Esse processo é baseado no trabalho de Floater (1997) e foi implementado por Miranda & Martha (2004).

A última etapa apresenta uma técnica utilizada para recuperar, a partir das coordenadas paramétricas, os valores e derivadas correspondentes na superfície. Esta nova técnica é uma adaptação da técnica SPR (*Superconvergent Patch Recovery*, Zienkiewicz & Zhu (1994)). Além disso, foi implementada a recuperação usando coordenadas baricênticas com o objetivo de comparar as duas técnicas. A Figura 4.1 ilustra os três passos do procedimento.

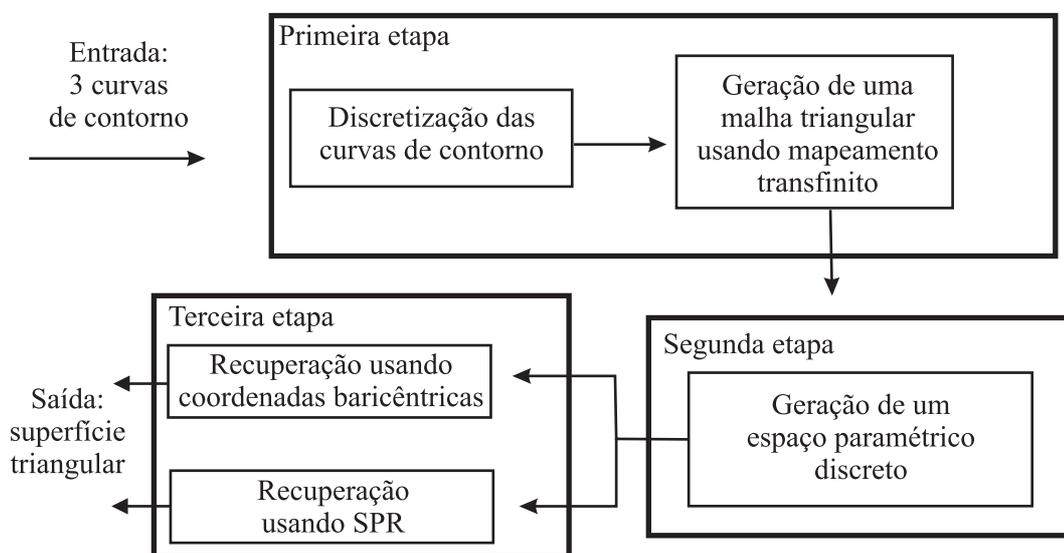


Figura 4.1: Procedimento para geração de uma superfície paramétrica triangular.

## 4.2 Mapeamentos Transfinitos

As técnicas de mapeamentos transfinitos são uma classe de métodos para estabelecer sistemas de coordenadas curvilíneas definidos pelo contorno de domínios arbitrários. O primeiro método associado a essas técnicas foi desenvolvido inicialmente por Gordon & Hall (1973). Neste trabalho, foram abordadas especificamente as regiões triangulares limitadas por três curvas de contorno.

Para descrever este mapeamento, é necessário introduzir o conceito de *operador linear idempotente* e *projektor*.

**Definição 4.2.1.** Um operador linear  $\psi: A \rightarrow A$  é dito idempotente se  $\psi(\psi(A)) = \psi(A)$ .

**Definição 4.2.2.** Um projetor  $P$  é qualquer operador linear idempotente, que mapeia uma superfície real em uma superfície aproximada, sujeito a certas restrições de interpolação.

Neste trabalho, para a obtenção de regiões triangulares, é utilizado o projetor *lofting* linear. Esse projetor efetua uma interpolação linear entre duas curvas de contorno  $\alpha(u)$  e  $\beta(u)$ , criando uma região  $F$ :

$$P[F] = P(u, v) = (1 - v)\alpha(u) + v\beta(u), \quad 0 \leq u \leq 1, \quad 0 \leq v \leq 1, \quad (4.1)$$

onde  $u$  é uma coordenada paramétrica normalizada ao longo de  $\alpha$  e  $\beta$  e  $v$  é uma coordenada normalizada que vale 0 em  $\alpha$  e 1 em  $\beta$ .

Uma região triangular  $T$  é delimitada por três curvas  $\alpha$ ,  $\beta$  e  $\psi$  (Figura 4.2(a)). Sejam  $u, v$  e  $w$  o sistema de cordenadas de área triangular normalizada, estabelecido sobre o interior da região, sujeito à seguinte restrição:  $u + v + w = 1$ .

Inicialmente, define-se um conjunto de 3 projetores lineares:

$$N_1(u, v, w) = \left(\frac{u}{1-v}\right)\beta(v) + \left(\frac{w}{1-v}\right)\psi(1-v) \quad (4.2)$$

$$N_2(u, v, w) = \left(\frac{v}{1-w}\right)\psi(w) + \left(\frac{u}{1-w}\right)\alpha(1-w) \quad (4.3)$$

$$N_3(u, v, w) = \left(\frac{w}{1-u}\right)\alpha(u) + \left(\frac{v}{1-u}\right)\beta(1-u) \quad (4.4)$$

O projetor  $N_1$  é ilustrado na Figura 4.2(b). Na verdade, cada projetor  $N_j$  é uma interpolação Lagrangeana linear entre duas das três curvas de contorno. Além disso, o projetor produto  $N_1N_2$  tem o aspecto mostrado na Figura 4.2(c).

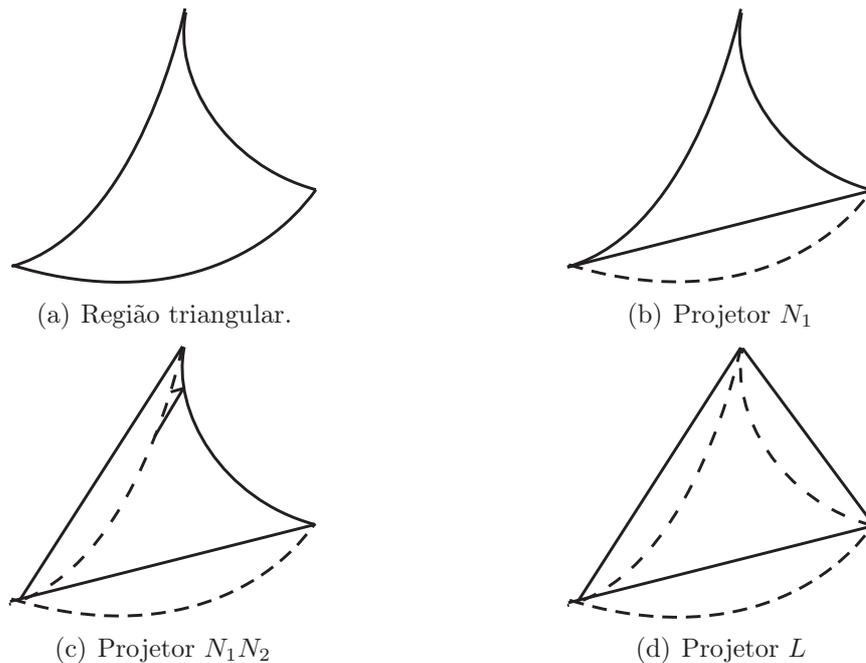


Figura 4.2: Mapeamento transfinito.

Um projetor soma quase-Booleana de  $N_i$  e  $N_j$  é definido por

$$N_i \oplus N_j \equiv N_i + N_j - N_i N_j, \quad i \neq j; \quad i, j = 1, 2, 3. \quad (4.5)$$

Além desses elementos, deve-se definir um projetor  $L = N_i N_j N_k$ ,  $i \neq j \neq k$ , que é projetado em um plano passando pelos três vértices de  $T$  (Figura 4.2(d)).

Finalmente, um projetor trilinear é dado por

$$\begin{aligned} S &= \frac{1}{6} \sum_{i \neq j} N_i \oplus N_j \\ &= \frac{1}{2} [(N_i \oplus N_j) + (N_i \oplus N_k)] \\ &= \frac{1}{2} [N_1 + N_2 + N_3 - L]; \quad i \neq j \neq k \end{aligned} \quad (4.6)$$

ou

$$\begin{aligned} S &= S(u, v, w) \\ &= \frac{1}{2} \left[ \left( \frac{u}{1-v} \right) \beta(v) + \left( \frac{w}{1-v} \right) \psi(1-v) + \left( \frac{v}{1-w} \right) \psi(w) + \right. \\ &\quad \left. \left( \frac{u}{1-w} \right) \alpha(1-w) + \left( \frac{w}{1-u} \right) \alpha(u) + \left( \frac{v}{1-u} \right) \beta(1-u) - \right. \\ &\quad \left. w\alpha(0) - u\beta(0) - \psi(0) \right] \end{aligned} \quad (4.7)$$

O conceito de projetor utilizado nessa mapeamento fornece uma maneira natural de criar-se uma malha triangular. Os nós da malha gerada são obtidos pela interseção de curvas geradas a partir de coordenadas constantes do projetor trilinear. Desse modo, as regiões triangulares delimitadas por essas curvas serão justamente os elementos triangulares da malha.

A implementação dessa etapa usa um algoritmo que recebe como entrada uma amostra do contorno e a quantidade  $n$  de pontos que cada lado do contorno possui. Este algoritmo usa o conceito de mapeamento transfinito descrito nessa seção e pode ser dividido em três partes:

1. os pontos do contorno são posicionados em determinadas posições da malha;
2. os pontos interiores são calculados e posicionados de acordo com a técnica de mapeamento transfinito;
3. os elementos triangulares são definidos por um vetor que armazena sequencialmente índices dos pontos, de modo que, a cada três índices, um triângulo da malha fica bem definido.

É interessante observar que esse algoritmo se baseia justamente na idéia de encontrar intersecções de isocurvas do projetor trilinear definido pelas três curvas de contorno. A Figura 4.3 ilustra essa etapa.

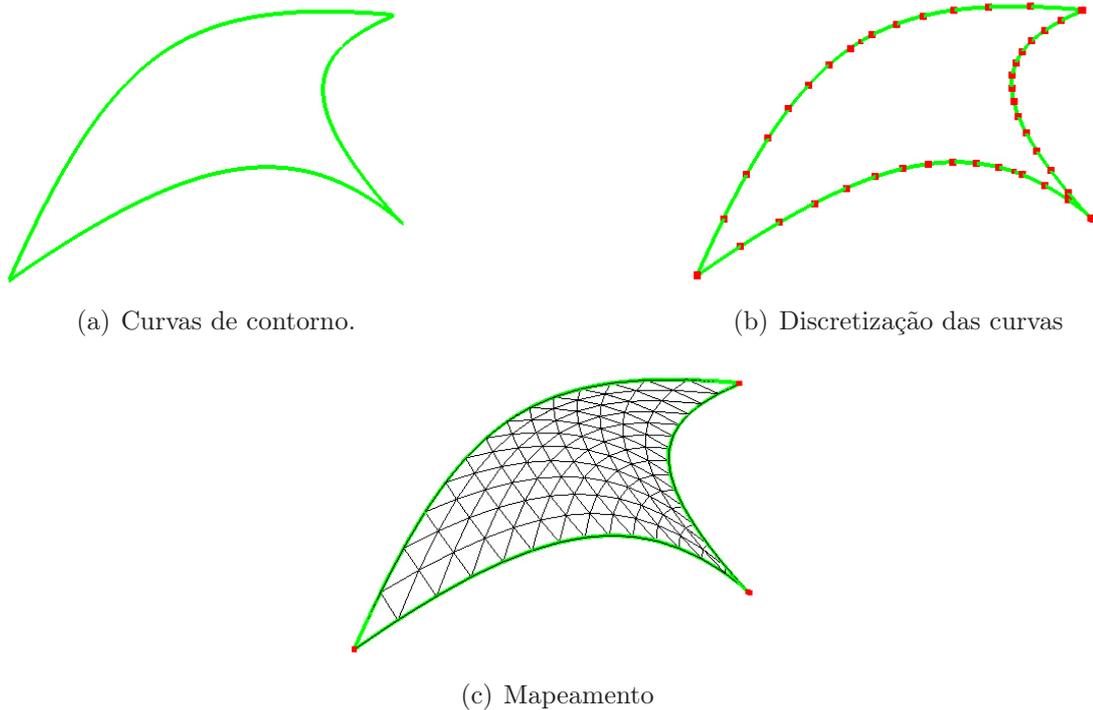


Figura 4.3: Discretização das curvas de contorno e geração da malha.

### 4.3 Geração do Espaço Paramétrico

A técnica utilizada, desenvolvida por Floater (1997) e implementada por Miranda & Martha (2004), se baseia no trabalho apresentado por Tutte (1963) sobre teoria dos grafos. A solução adotada mapeia cada triângulo da triangulação da superfície em um triângulo no espaço paramétrico. Desse modo, cada ponto  $(x, y, z)$  da triangulação é associado a um ponto  $(u, v)$  do espaço paramétrico.

O trabalho de Floater apresenta três formas de obter uma parametrização: uniforme, média ponderada do inverso da distância e preservação de forma. A descrição detalhada dessas formas podem ser encontradas no trabalho de Floater (1997).

A idéia básica da parametrização é gerar um espaço paramétrico triangulado que é topologicamente igual à superfície original triangulada, mas de geometria distinta. Desse modo, cada triângulo da malha terá um triângulo associado no espaço paramétrico.

Neste trabalho, foi utilizada a biblioteca PFS (Miranda & Martha 2004), que apresenta uma implementação de acordo com o trabalho de Floater. Segue abaixo uma descrição dos passos para geração da parametrização da malha da superfície.

Dada uma malha, o procedimento inicial consiste em encontrar um plano de mínimos quadrados com os *Border Stones* da malha da superfície. Definem-se como *Border Stones* os nós localizados em cantos agudos da malha. Esses nós são calculados automaticamente pela biblioteca PFS.

Em seguida, é criado um sistema de coordenadas locais nesse plano, de modo que o eixo  $u$  é orientado do primeiro para o segundo *Border Stone*. Os *Border Stones* são projetados no plano e mapeados para um sistema normalizado nesse mesmo plano.

As curvas de contorno serão mapeadas para retas definidas por *Border Stones*, de modo que os nós do contorno da malha são mapeados respeitando o comprimento do arco ao longo da curva.

A determinação dos valores paramétricos relacionados a nós do interior da malha é feita inicialmente pela projeção desses nós no plano. Porém, valores mais interessantes são obtidos usando alguma forma de parametrização, como citado acima.

A diferença entre as diversas formas de parametrização está no modo de calcular fatores de ponderação relativos a um nó e seus nós adjacentes.

A parametrização usando preservação de forma é utilizada neste trabalho. Ela se baseia no uso de combinações convexas. O trabalho de Tutte apresenta uma técnica para desenho de grafos planos com linhas retas usando combinações convexas. Essas combinações devem seguir algumas regras pré-definidas. Mais informações podem ser encontradas em (Floater 1997). A Figura 4.4 exibe uma malha de elementos triangulares e seu espaço paramétrico correspondente.

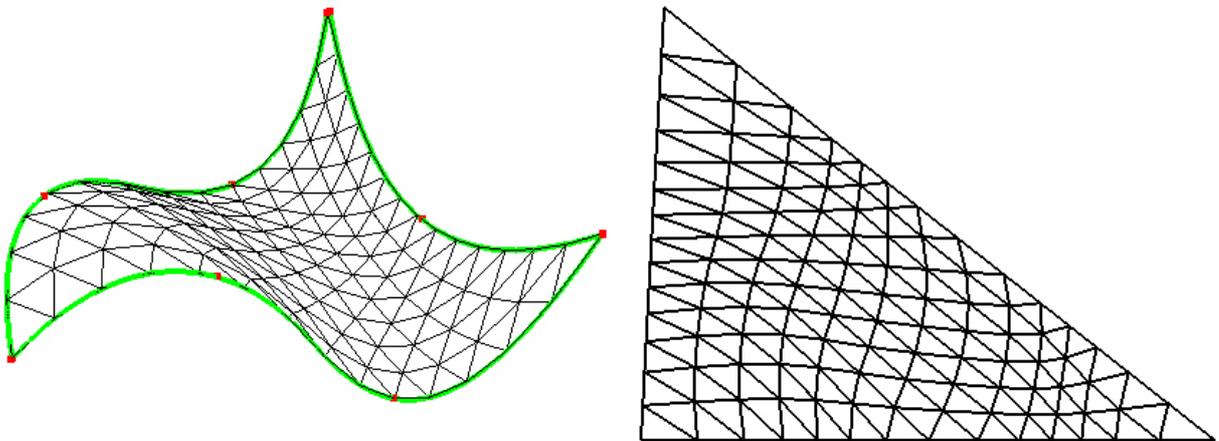


Figura 4.4: Malha de triângulos e espaço paramétrico gerado usando a PFS.

## 4.4 Recuperação de Valores e Derivadas

Esta seção discute uma nova técnica utilizada para recuperar, a partir das coordenadas paramétricas, os valores e derivadas correspondentes na superfície. Também discute uma técnica de recuperação baseada em coordenadas baricêntricas.

### 4.4.1 Recuperação Usando Coordenadas Baricêntricas

Seja  $T$  um triângulo formado pelos vértices  $v_0, v_1, v_2$ . Dado um ponto  $x$  qualquer do plano, podemos representar  $x$  da seguinte maneira:

$$x = av_0 + bv_1 + cv_2, \quad a + b + c = 1, \quad (4.8)$$

onde  $(a, b, c)$  são as coordenadas baricêntricas de  $x$  em relação ao triângulo  $T$ .

Este conceito apresenta as seguintes propriedades:

- a soma das coordenadas baricêntricas sempre é igual a um;
- as coordenadas baricêntricas são sempre positivas no interior do triângulo;
- se o ponto está localizado em uma aresta do triângulo, uma coordenada será zero;
- se o ponto está localizado em um vértice do triângulo, duas coordenadas serão nulas;
- se o ponto está localizado fora do triângulo, pelo menos uma coordenada é negativa.

A Figura 4.5 ilustra o uso de coordenadas baricêntricas para recuperar valores em um triângulo. Neste exemplo, sabe-se que os vértices  $A, B$  e  $C$  têm valores  $f(A), f(B)$  e  $f(C)$  associados. Dado um ponto  $x$  de coordenadas baricêntricas  $(a, b, c)$  no triângulo, é possível achar o valor de  $f(x)$  que é dado por  $af(A) + bf(B) + cf(C)$ .

Usando coordenadas baricêntricas, um ponto qualquer da superfície pode ser obtido da seguinte maneira:

1. dado um ponto  $x = (u, v)$  do espaço paramétrico, determina-se o triângulo  $T$ , formado pelos vértices  $v_0, v_1, v_2$ , do espaço paramétrico, que contém  $x$ ;
2. obtêm-se as coordenadas baricêntricas  $(a, b, c)$  do ponto  $x$  em relação ao triângulo  $T$ .

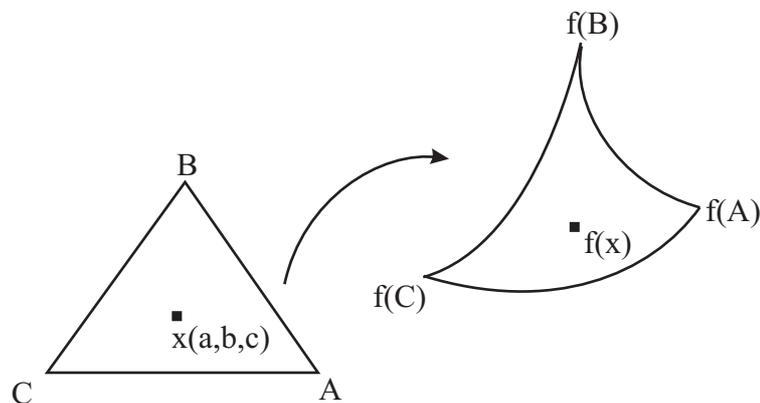


Figura 4.5: Recuperação de valores usando coordenadas baricêntricas.

3. recupera-se os pontos  $f(v_0)$ ,  $f(v_1)$  e  $f(v_2)$  da superfície correspondentes aos pontos  $v_0$ ,  $v_1$  e  $v_2$  do espaço paramétrico;
4. o ponto  $f(x)$  da superfície é calculado usando as coordenadas baricênticas de  $x$  no triângulo formado pelos pontos  $f(v_0)$ ,  $f(v_1)$  e  $f(v_2)$ . Então

$$f(x) = af(v_0) + bf(v_1) + cf(v_2). \quad (4.9)$$

O mesmo procedimento é utilizado para recuperação das derivadas.

É interessante ressaltar que a determinação do triângulo que contém um ponto  $x$  é feita testando-se cada triângulo da triangulação, até que se encontre o triângulo no qual as coordenadas baricênticas sejam todas positivas. Uma maneira de otimizar este processo consiste em armazenar para cada triângulo informações relativas à sua caixa delimitadora, ou seja, as coordenadas do retângulo que envolve o triângulo (*bounding box*). Dessa maneira, comparações simples são realizadas em cada triângulo para determinar a possibilidade de o ponto estar contido na caixa que envolve o triângulo.

Outra forma para otimizar o processo é a utilização de estruturas de dados especiais (árvores) para armazenar as informações de cada triângulo. Esse procedimento ainda não está implementado neste trabalho.

Por se tratar de uma interpolação linear local em cada triângulo, sem haver qualquer processo de suavização, essa técnica não gera superfícies suaves, mas tem baixo custo computacional. A Figura 4.6 exibe uma superfície triangular recuperada usando essa técnica.

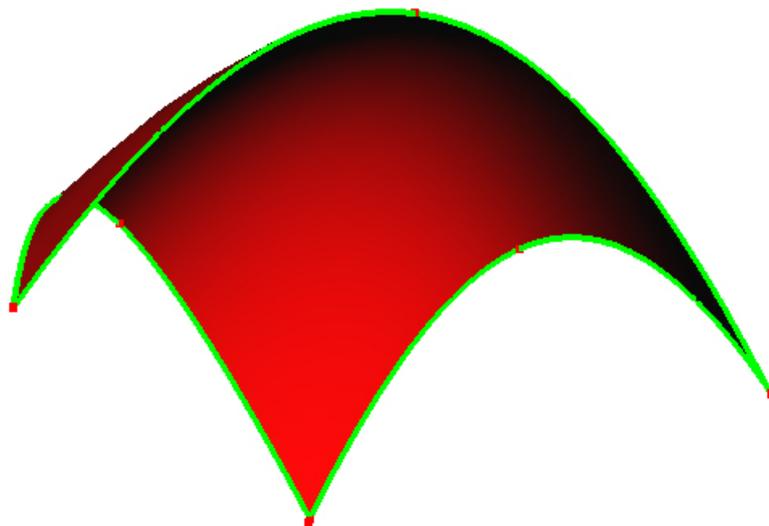


Figura 4.6: Superfície triangular recuperada por coordenadas baricênticas.

### 4.4.2 Recuperação Usando SPR

A recuperação usando SPR é uma técnica que busca obter superfícies mais suaves que as obtidas com a recuperação por coordenadas baricêntricas. Esta subseção contém um breve resumo da técnica original e descreve sua adaptação feita neste trabalho.

#### SPR original

A técnica SPR (*Superconvergent Patch Recovery*) é aplicada em análise numérica pelo Método dos Elementos Finitos. Essa técnica foi desenvolvida inicialmente por Zienkiewicz & Zhu (1994) e ela é bastante utilizada em Engenharia, por exemplo, para se obter um campo de tensões aproximado a partir de pontos de amostragem.

A aproximação de um campo genérico é dada por

$$\psi = Pa, \quad (4.10)$$

onde  $P$  contém termos polinomiais e  $a$  é um conjunto de parâmetros desconhecidos.

Os termos polinomiais devem ser previamente definidos de acordo com a aplicação. Um exemplo simples é o plano representado por  $P = [1, u, v]$ . Nesse caso, é necessário calcular os parâmetros  $a = [a_0, a_1, a_2]$ .

O cálculo do conjunto de parâmetros  $a$  baseia-se em sistemas de mínimos quadrados ponderados. A amostra considerada para esse cálculo é obtida a partir de grupos de elementos denominados *patches*.

Desse modo, dado um nó qualquer do campo, deve-se inicialmente criar um *patch*. No SPR original, a formação de *patches* segue um conjunto de regras e restrições. A Figura 4.7 exibe um *patch* com quatro elementos quadrilaterais formado por um nó central. Cada elemento possui quatro pontos de amostragem denotados por um símbolo  $x$ . Esses pontos possuem, opcionalmente, um peso associado que é determinado pelo inverso da distância do ponto de amostragem ao nó formador de *patch*. São esses pontos de amostragem que serão levados em consideração para gerar a expansão polinomial.

Então, sendo  $(u_i, v_i)$  as coordenadas Cartesianas de um ponto de amostragem  $i$ ,  $\sigma(u_i, v_i)$  o valor associado ao ponto de amostragem e  $w_i$  seus pesos, é necessário minimizar a função

$$F = \sum_{i=1}^m w_i^2 [\sigma(u_i, v_i) - \psi(u_i, v_i)]^2. \quad (4.11)$$

Usando (4.10), chega-se à equação

$$F = \sum_{i=1}^m w_i^2 [\sigma(u_i, v_i) - P(u_i, v_i)a]^2. \quad (4.12)$$

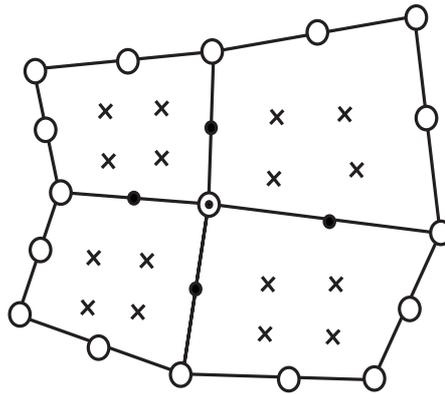


Figura 4.7: *Patch* gerado por um nó do campo.

Para minimizar essa função, é necessário fazer  $\frac{\delta F}{\delta a} = 0$ , chegando-se ao sistema de equações lineares

$$Aa = b, \quad (4.13)$$

onde

$$A = \sum_{i=1}^m w_i^2 P^t(u_i, v_i) P(u_i, v_i) \quad (4.14)$$

$$b = \sum_{i=1}^m w_i^2 P^t(u_i, v_i) \sigma(u_i, v_i). \quad (4.15)$$

Desse modo, é possível calcular o vetor  $a$  correspondente ao conjunto de parâmetros do polinômio, calculando-se a matriz inversa de  $A$ .

Finalmente, as coordenadas do nó podem ser aplicadas à expansão polinomial para se recuperar o valor aproximado.

### SPR adaptado

A SPR original foi adaptada neste trabalho para a recuperação de pontos da superfície a partir de um conjunto de pontos do espaço paramétrico. Existem algumas diferenças fundamentais entre os dois procedimentos.

No caso da SPR original, os pontos formadores de *patch* são nós do campo. Os elementos contidos no *patch* possuem pontos de amostragem que determinam a expansão polinomial dada pela Equação (4.10). Além disso, apenas uma quantidade finita de pontos podem ser determinados, para garantir a convergência dos *patches*.

Porém, no caso da recuperação de pontos da superfície, é necessário se calcular, a partir de um ponto qualquer do espaço paramétrico, suas coordenadas correspondentes na superfície.

Neste trabalho, os patches são criados a partir de alguns nós do espaço paramétrico no início do processo, seguindo uma regra que garante que os *patches* sejam todos disjuntos

e cubram todos os triângulos do espaço paramétrico. A Figura 4.8 exibe um espaço paramétrico triangulado dividido em *patches*.

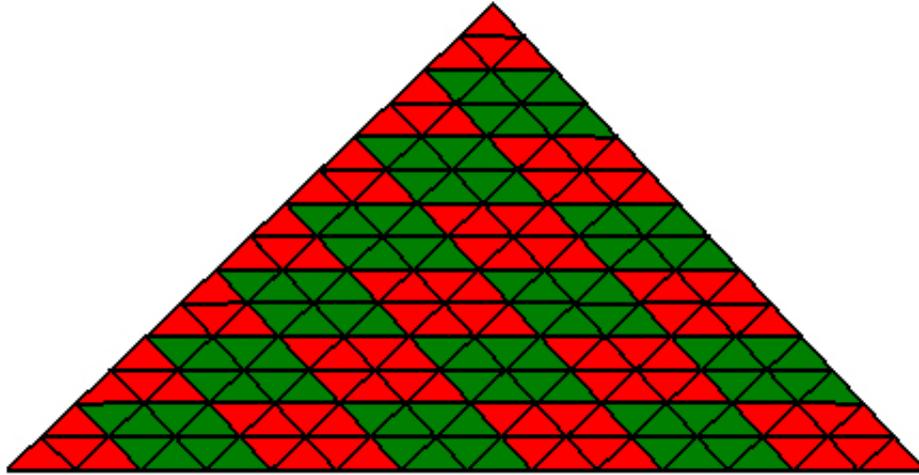


Figura 4.8: Espaço paramétrico triangulado dividido em *patches*.

Cada *patch* do espaço paramétrico tem seu polinômio gerado a partir de uma amostragem formada pelos vértices dos triângulos nele contidos. Neste trabalho, foram considerados polinômios quadráticos com os termos polinomiais  $P = [1, u, v, u^2, uv, v^2]$ , para geração de superfícies suaves.

Além disso, para a obtenção de superfícies suaves, é necessário garantir a transição suave entre os *patches*. Com essa finalidade, algumas adaptações foram desenvolvidas.

Dado um espaço paramétrico dividido em *patches*, o cálculo de um ponto da superfície a partir de um ponto  $(u, v)$  qualquer do espaço paramétrico ocorre da seguinte maneira:

1. determina-se o triângulo  $T$  do espaço paramétrico que contém o ponto  $(u, v)$ ;
2. determina-se o *patch*  $P$  no qual o triângulo está contido;
3. determinam-se todos os triângulos  $T_{i,j}$  que compartilham o vértice  $v_i$  do triângulo  $T$  (inclusive o próprio  $T$ );
4. determinam-se os *patches*  $P_{i,j}$  que contêm cada triângulo  $T_{i,j}$ ;
5. calcula-se, a partir do polinômio  $p_{i,j}$  gerado por cada *patch*  $P_{i,j}$ , o valor  $p_{i,j}(u, v)$ ;
6. para cada  $i$ , é feita uma média  $M_i$  dos valores  $p_{i,j}(u, v)$  usando todos os *patches*  $P_{i,j}$ . Esse cálculo é dado pela equação

$$M_i(u, v) = \frac{1}{n_i} \left( \sum_{j=1}^{n_i} p_{i,j}(u, v) \right), \quad (4.16)$$

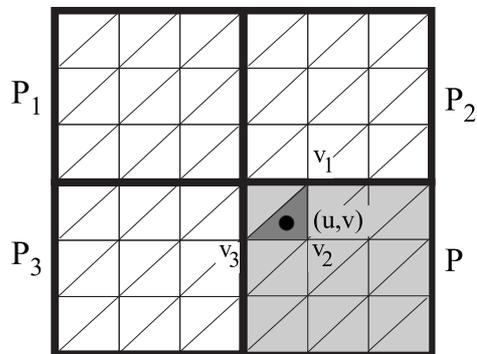
onde  $n_i$  é a quantidade de triângulos que compartilham o vértice  $v_i$ ;

7. calculam-se as coordenadas baricêntricas  $c_1, c_2$  e  $c_3$  relativas aos vértices  $v_1, v_2$  e  $v_3$  do triângulo  $T$ ;
8. o valor final é obtido pela seguinte fórmula:

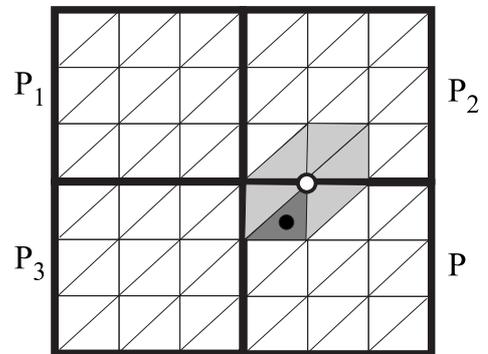
$$V(u, v) = \sum_{i=1}^3 c_i M_i(u, v). \quad (4.17)$$

O procedimento acima é utilizado também para recuperação das derivadas. O procedimento para determinação do triângulo que contém o ponto  $(u, v)$  está descrito na seção de recuperação usando coordenadas baricêntricas. A determinação do *patch* que contém cada triângulo é facilmente executada usando-se estruturas de dados simples.

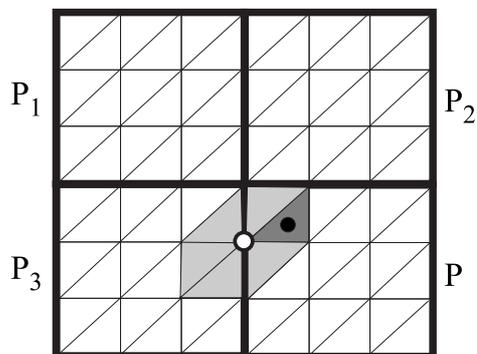
A Figura 4.9 ilustra o processo de determinação dos triângulos e *patches* que influenciam o valor final. A Figura 4.9(a) exibe o triângulo e o *patch*  $P$  que contém um dado ponto  $(u, v)$ . As Figuras 4.9(b), 4.9(c) e 4.9(d) exibem os triângulos que compartilham os vértices  $v_1, v_2$  e  $v_3$ , respectivamente. Os *patches* que contêm cada um desses triângulos serão utilizados no cálculo das médias  $M_i$ . Note que um *patch* pode conter mais que um dos triângulos compartilhados. Neste caso, o valor obtido naquele *patch* é somado mais que uma vez.



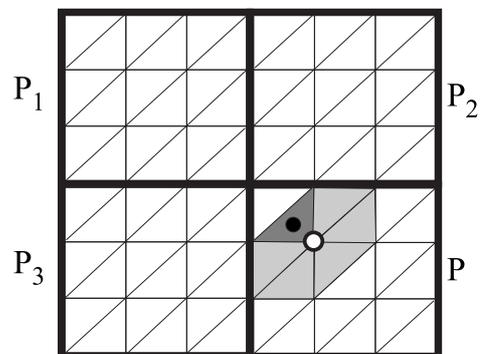
(a) Determinação do triângulo e *patch*  $P$  que contém um ponto  $(u, v)$ .



(b) Determinação dos triângulos que compartilham o vértice  $v_1$ .



(c) Determinação dos triângulos que compartilham o vértice  $v_2$ .



(d) Determinação dos triângulos que compartilham o vértice  $v_3$ .

Figura 4.9: Determinação de *patches* influentes para cálculo das médias  $M_i$ .

A formulação acima garante a transição suave entre os patches, pelo uso de coordenadas baricêntricas. É fácil observar que, à medida que um ponto  $(u, v)$  se aproxima da fronteira do *patch*, a influência dos *patches* vizinhos aumenta, enquanto a influência do *patch* que contém o ponto diminui. A Figura 4.10 exibe uma superfície triangular recuperada usando SPR.

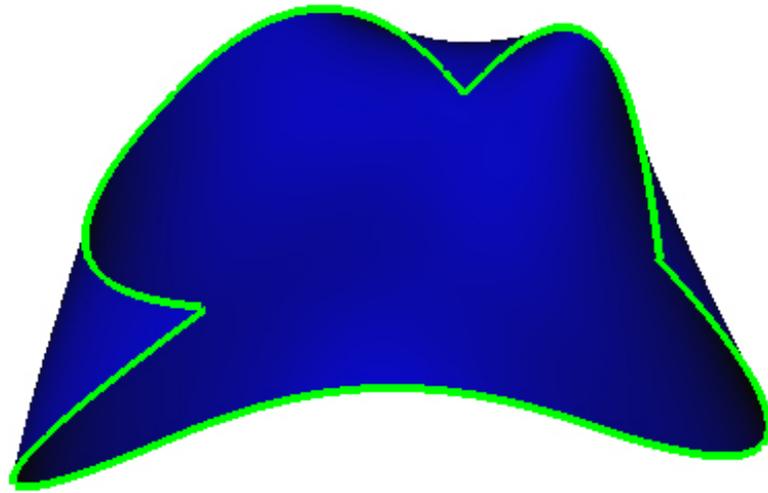


Figura 4.10: Superfície triangular recuperada por SPR.

# Capítulo 5

## Exemplos

Este capítulo ilustra o procedimento para interpolação de superfícies paramétricas triangulares, desde a discretização das curvas de contorno até as duas técnicas de recuperação, usando coordenadas baricêntricas e SPR. É importante ressaltar que todas as imagens foram obtidas do ambiente gráfico desenvolvido neste trabalho para modelagem e visualização de superfícies.

A Figura 5.1 exibe três curvas de contorno e pontos representando a discretização a ser utilizada para a geração da malha de elementos triangulares.

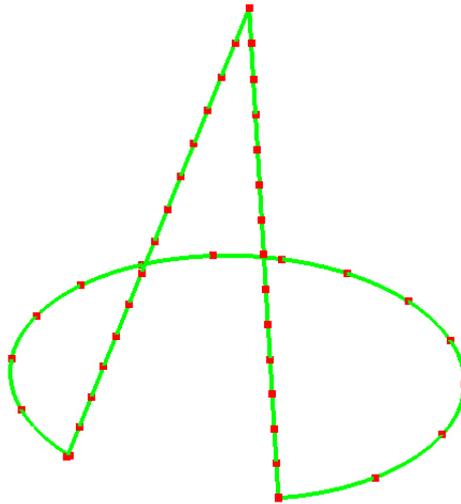


Figura 5.1: Discretização das curvas de contorno.

A malha gerada a partir da discretização acima pode ser vista na Figura 5.2. Essa figura também exibe o espaço paramétrico gerado pela biblioteca PFS.

Finalmente, a Figura 5.3 exibe a superfície reconstruída usando coordenadas baricêntricas e SPR.

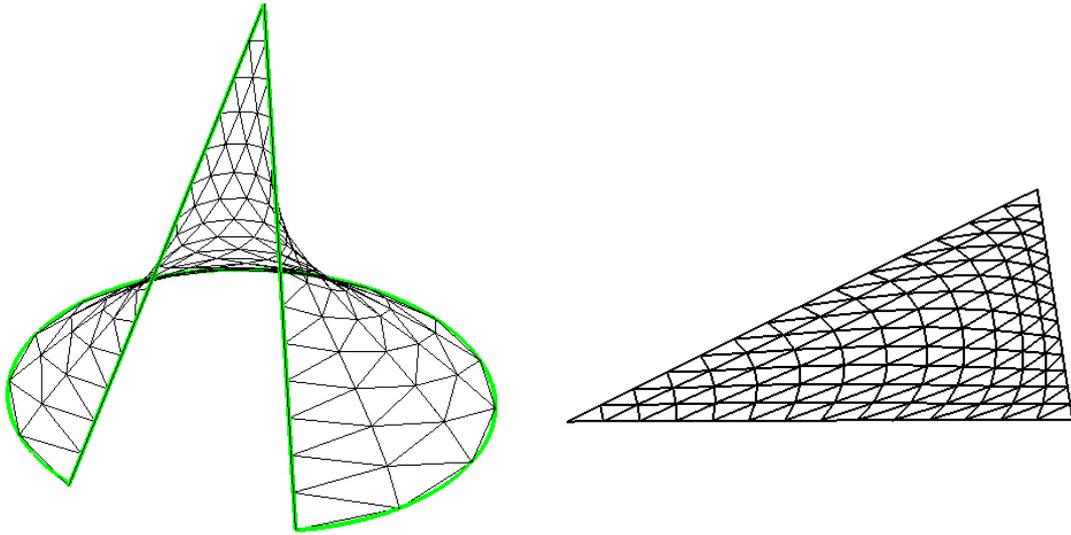
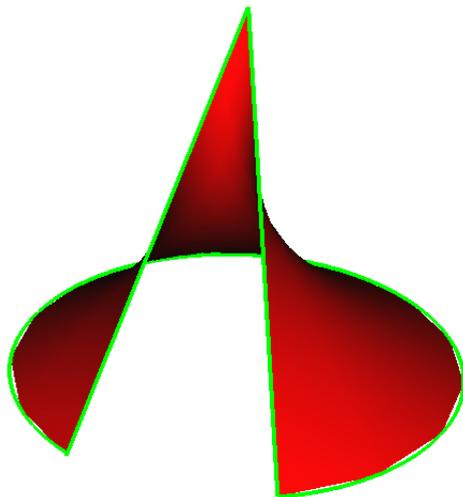
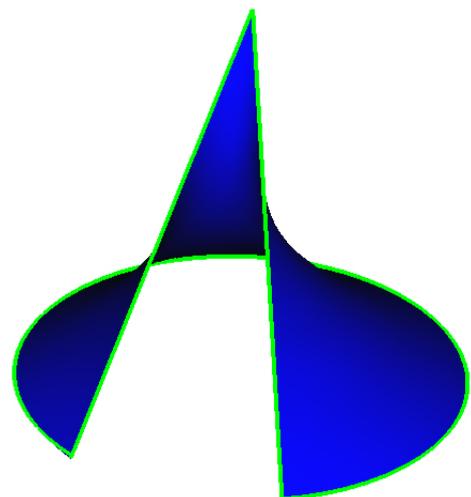


Figura 5.2: Geração da malha de elementos triangulares e espaço paramétrico obtido.



(a) Superfície reconstruída usando coordenadas baricênticas.



(b) Superfície reconstruída usando SPR.

Figura 5.3: Superfícies reconstruídas.

# Capítulo 6

## Conclusão e Trabalhos Futuros

Este trabalho apresentou uma alternativa para a modelagem de superfícies paramétricas triangulares a partir de três curvas de contorno. Esta estratégia resolve a principal dificuldade do uso de NURBS, que está no fato de seu espaço paramétrico ser topologicamente incompatível com as superfícies triangulares.

A técnica proposta é baseada na integração de três módulos. O primeiro é responsável pela discretização das curvas do contorno da superfície e pela geração de uma malha de elementos triangulares que é utilizada como base para a representação da superfície. O segundo módulo consiste na utilização de um algoritmo para a determinação de um espaço paramétrico discreto associado à malha. O último módulo é baseado na utilização de uma técnica para recuperação de valores de coordenadas cartesianas e derivadas associadas às coordenadas paramétricas (técnica SPR), garantindo uma transição suave no espaço paramétrico da superfície.

A técnica de recuperação usando SPR apresentou bons resultados, mas o método de mínimos quadrados usado para geração do polinômio não garante que a superfície gerada irá respeitar todos os pontos da malha de elementos triangulares. Desse modo, o contorno da superfície pode apresentar pequenas diferenças em relação à discretização do contorno.

Também foi implementada uma técnica de recuperação baseada em coordenadas baricêntricas a fim de comparar os resultados com a técnica SPR adaptada.

Os trabalhos futuros incluem o desenvolvimento de um tratamento específico para garantir que as superfícies respeitem as curvas de contorno, um estudo comparativo mais aprofundado entre as superfícies geradas a partir de coordenadas baricêntricas e SPR, e o desenvolvimento de métodos para geração de *patches* em triangulações arbitrárias.

# Referências Bibliográficas

- Carmo, M. P. (2005), *Geometria Diferencial de Curvas e Superfícies*, 1 edn, SBM, Rio de Janeiro.
- Floater, M. S. (1997), ‘Parametrization and smooth approximation of surface triangulations’, *Computer Aided Geometric Design* **14**(4), 231–250.
- Gomes, J. & Velho, L. (2003), *Fundamentos da Computação Gráfica*, 1 edn, IMPA, Rio de Janeiro.
- Gordon, W. & Hall, C. A. (1973), ‘Construction of curvilinear co-ordinate systems and application to mesh generation’, *International Journal for Numerical Methods in Engineering* **7**, 461–477.
- Miranda, A. C. O. & Martha, L. F. (2004), Geração de malhas estruturadas a partir de um espaço paramétrico de triangulações não estruturadas, *in* ‘Anais do XXV CILAMCE: 25nd Iberian Latin-American Congress on Computational Methods in Engineering’.
- Piegl, L. & Tiller, W. (1997), *The NURBS Book*, 2 edn, Springer, Berlin.
- Qin, H. & Terzopoulos, D. (1997), ‘Triangular nurbs and their dynamic generalizations’, *Computer Aided Geometric Design* **14**(4), 325–347.
- Shreiner, D., Woo, M., Neider, J. & Davis, T. (2003), *OpenGL Programming Guide: The Official Guide to Learning OpenGL*, 4 edn, Addison-Wesley, Massachusetts.
- Stroustrup, B. (1991), *The C++ Programming Language*, 2 edn, Addison-Wesley, Massachusetts.
- Tutte, W. T. (1963), ‘How to draw a graph’, *Proc. London Math. Soc.* **3**(13), 743–768.
- Zienkiewicz, O. & Zhu, J. (1994), ‘The SPR recovery and boundaries’, *International Journal for Numerical Methods in Engineering* **37**, 3195–3196.