

Universidade Federal de Campina Grande  
Centro de Ciências e Tecnologia  
Programa de Pós-Graduação em Matemática  
Curso de Mestrado em Matemática

# Uso da Aplicação Normal de Gauss na Poligonização de Superfícies Implícitas

por

Thiciany Matsudo Iwano

sob orientação da

Prof<sup>a</sup>. Dr<sup>a</sup>. Rosana Marques da Silva

Dissertação apresentada ao Corpo Docente do Programa de Pós-Graduação em Matemática - CCT - UFCG, como requisito parcial para obtenção do título de Mestre em Matemática.

Campina Grande - PB

Outubro/2005

# Uso da Aplicação Normal de Gauss na Poligonização de Superfícies Implícitas

por

Thiciany Matsudo Iwano

Dissertação apresentada ao Corpo Docente do Programa de Pós-Graduação em Matemática - CCT - UFCG, como requisito parcial para obtenção do título de Mestre em Matemática.

Área de Concentração: Matemática Aplicada - Modelagem Geométrica

Aprovada por:

---

Prof<sup>a</sup>. Dr<sup>a</sup>. Wu Shin-Ting

---

Prof. Dr. Vânio Fragoso de Melo

---

Prof<sup>a</sup>. Dr<sup>a</sup>. Rosana Marques da Silva  
Orientadora

Universidade Federal de Campina Grande  
Centro de Ciências e Tecnologia  
Programa de Pós-Graduação em Matemática  
Curso de Mestrado em Matemática

Outubro/2005

# Agradecimentos

A professora Rosana pela orientação e paciência.

Aos companheiros de trabalho do DME, pelas suas palavras de apoio e incentivo.

Aos amigos da pós-graduação, em especial, Cícero, Fernando, Luis Paulo e Orlando.

A meu namorado Daniel, pela compreensão e carinho nos momentos críticos.

Aos professores Adonai e Wilbert do Departamento de Matemática da Universidade Federal de Alagoas.

# Dedicatória

Aos meus pais.

# Resumo

Neste trabalho apresentamos um estudo das principais técnicas de geração de malhas poligonais, a partir de superfícies descritas matematicamente por funções implícitas, isto é, superfícies definidas pelo conjunto

$$S = f^{-1}(0) = \{X \in \mathbb{R}^3 \mid f(X) = 0\},$$

onde  $f : \mathbb{R}^3 \rightarrow \mathbb{R}$  e  $f$  é, pelo menos, de classe  $\mathcal{C}^2$ . Mostramos um método para obter as curvaturas gaussianas e médias dessas superfícies a partir do vetor  $\nabla f$  para cada ponto de  $S$ . Abordamos questões como a preservação de características geométricas e topológicas do objeto gráfico.

Dentre os métodos estudados, ressaltamos o algoritmo *Marching Triangles*, que gera uma malha a partir de um ponto arbitrário  $p$  sobre a superfície  $S$  e um referencial local, usando a abordagem do avanço de frentes. Em sua implementação, usamos o raio de curvatura, calculado a partir da curvatura normal máxima absoluta da superfície em cada ponto  $p$  pertencente a  $S$ , para adaptar o comprimento das arestas da malha triangular à geometria local da superfície  $S$ .

# Abstract

In this work we present a study about the main techniques of surfaces meshes generation, described by implicit functions, that is, surfaces defined by the set

$$S = f^{-1}(0) = \{X \in \mathbb{R}^3 \mid f(X) = 0\},$$

where  $f : \mathbb{R}^3 \rightarrow \mathbb{R}$  and  $f$  is, at least,  $\mathcal{C}^2$ . We discuss aspects involving his preservation of graphic object's geometry and topology.

As special method we cite the Marching Triangles that generates a mesh starting from an arbitrary point  $p$  on surface  $S$  and a local referencial, using advancing fronts approach. In our implementation, we use the radius of curvature, calculated from surface's absolute maximum normal curvature in each point  $p$  in  $S$  and the triangular mesh, to adapt the edges length of the mesh to the local geometry.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
1.1	Organização do Trabalho . . . . .	5
<b>2</b>	<b>Resultados Importantes</b>	<b>6</b>
2.1	Superfícies Regulares . . . . .	6
2.1.1	Aplicação Normal de Gauss . . . . .	9
2.2	Aplicação de Weingarten . . . . .	12
<b>3</b>	<b>Métodos de Poligonização de Superfícies Implícitas</b>	<b>15</b>
3.1	Introdução . . . . .	15
3.2	Particionamento Celular ou Volumétrico . . . . .	16
3.2.1	Subdivisão Adaptativa de Cubos . . . . .	20
3.2.2	O Uso das Células Tetraédricas . . . . .	20
3.2.3	Ambigüidades Topológicas . . . . .	22
3.3	Rastreio de Superfície . . . . .	24
3.3.1	Continuação Linear por Partes . . . . .	24
3.3.2	Continuação Preditor-Corretor . . . . .	25
3.3.3	<i>Marching Triangles</i> . . . . .	25
3.4	Pseudo-Físicos . . . . .	27
3.4.1	Sistema de Partículas . . . . .	27
3.4.2	Sistema Massa-mola . . . . .	29
<b>4</b>	<b>O Algoritmo <i>Marching Triangles</i></b>	<b>31</b>
4.1	Determinação da Frente Inicial . . . . .	32

	1
4.2 Cálculo dos Ângulos Frontais . . . . .	35
4.3 Verificação de intra e extra colisões . . . . .	36
4.4 Construindo a Expansão da Frente . . . . .	37
<b>5 Implementação</b>	<b>39</b>
5.1 Considerações Gerais . . . . .	39
5.1.1 O Algoritmo . . . . .	39
5.2 Estruturas de Dados . . . . .	41
5.3 Imagens . . . . .	42
<b>6 Conclusão</b>	<b>46</b>
<b>A Aplicações Lineares Auto - Adjuntas</b>	<b>48</b>
<b>B Superfícies Trianguláveis</b>	<b>50</b>
<b>C Métodos Numéricos para Determinação de Raízes Reais</b>	<b>54</b>
C.1 Método da Bissecção . . . . .	54
C.2 Método de Euler . . . . .	55
C.3 Método de Newton . . . . .	56
<b>D Aritmética Intervalar e Afim</b>	<b>58</b>
D.1 Aritmética Intervalar . . . . .	58
D.2 Aritmética Afim . . . . .	59
D.3 Conversão entre IA e AA . . . . .	61
<b>Bibliografia</b>	<b>62</b>



# Capítulo 1

## Introdução

Aproximações poligonais de objetos geométricos são comumente usadas para representar uma versão discreta desses objetos em meios computacionais. Essa representação discreta deve contemplar as características geométricas (relações métricas) e topológicas (relações de conectividade) do objeto geométrico em questão, ou seja, a aproximação deve capturar de forma mais fiel possível as informações contidas nos dados amostrados. O processo de construir uma aproximação poligonal de objetos geométricos no computador é denominado de poligonalização.

Uma aproximação poligonal ou poligonização de superfícies pode ser obtida através da interpolação de pontos amostrados ou a aproximação de representações exatas, formando uma malha, conhecida como superfície poliédrica que, por sua vez, é composta por um conjunto de faces planares, geralmente triangulares, delimitadas por vértices e arestas.

Métodos de poligonização de superfícies devem satisfazer, no mínimo, os seguintes requisitos [5]: consistência topológica com o modelo original, tempo de execução aceitável (complexidade), conectividade automática dos vértices para a formação de cada face, e também, um número limitado de faces da malha para permitir aplicações iterativas.

Uma superfície no espaço euclidiano pode ser representada de diversas formas: podemos ter um conjunto de pontos, obtido pela varredura de um objeto real ou por uma simulação; uma fórmula matemática ou por *métodos procedurais*. Em nosso tra-

balho, os objetos geométricos de interesse são superfícies descritas implicitamente por fórmulas matemáticas, também conhecidas como superfícies implícitas.

Os objetos geométricos podem ser descritos matematicamente de duas formas: descrição paramétrica e descrição implícita [7]: paramétrica e implícita. A descrição paramétrica é a mais popular nos processos gráficos. Nesta descrição os pontos que constituem o objeto de interesse são dados diretamente por uma parametrização local que define uma aplicação do espaço dos parâmetros no espaço do objeto. Na descrição implícita, o conjunto de pontos que pertencem a uma superfície é determinado pelo conjunto de pontos do espaço  $S = f^{-1}(0) = \{X \in \mathbb{R}^3 \mid f(X) = 0\}$ , onde  $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ . Em outras palavras, o conjunto de pontos pertencentes a um objeto implícito é dado de forma indireta através de uma função que define a relação das coordenadas dos pontos do objeto.

A partir de meados dos anos 80, com os trabalhos de Allgower e Bloomenthal [4, 34], a representação implícita de objetos geométricos passou a receber muita atenção na comunidade de computação gráfica devido às facilidades que oferece para modelar objetos complexos, com formas suaves e topologia arbitrária [17]. Atualmente, a literatura especializada apresenta vários métodos de poligonização de superfícies.

Os diversos métodos de poligonização de superfícies entre si pela forma como são realizadas as etapas de amostragem e estruturação, onde a amostragem consiste na determinação dos pontos que definem o objeto geométrico e a estruturação trata da criação de uma estrutura para os pontos, de forma a garantir as adjacências, e representar a aproximação poligonal, ou seja, a estruturação estabelece uma organização combinatorial consistente com o objeto amostrado. Podemos dizer que a estruturação está intimamente relacionada com a topologia do objeto.

Esses métodos, em sua grande maioria, empregam a partição do espaço ambiente, isto é, o espaço onde o objeto geométrico (superfície) está imerso é particionado em células, com a forma de cubos ou de tetraedros, e procuram gerar uma malha poligonal a partir da determinação dos pontos de interseção dessas células com a superfície dada, caracterizando um processo de amostragem, e da interpolação desses pontos, configurando o processo de estruturação.

Existem algumas tentativas de classificar os métodos de poligonização de superfícies implícitas. Bloomenthal [4,6] propõe uma classificação de acordo com a abordagem

usada pelo algoritmo para determinar os pontos do objeto geométrico e o uso, ou não, de critérios adaptativos a partir da extração de informações do objeto, de modo a realizar etapas de refinamento durante a obtenção de pontos. Figueiredo [7, 14] sugere a classificação dos algoritmos em função das duas etapas: amostragem e estruturação.

Araújo e Jorge [30] apresentam uma classificação semelhante a de Bloomenthal, onde o critério considerado é a forma como os pontos da superfície são calculados: a partir do particionamento do espaço ou a partir de um ponto conhecido (ou suficientemente) próximo do objeto e Carvalho [33] classifica os métodos de acordo com o tipo da malha obtida em métodos simpliciais (malhas triangulares) e métodos não simpliciais.

Em nosso trabalho, faremos uma classificação, de acordo com a forma como é realizada a amostragem em métodos geométricos e em métodos pseudo-físicos.

Os métodos geométricos tem duas abordagens que denominamos particionamento celular ou volumétrico e rastreamento de superfícies. Nos métodos volumétricos, a amostragem dos pontos sobre a superfície é feito considerando o espaço onde o objeto está imerso, e em rastreamento de superfícies a amostragem é realizada diretamente sobre o objeto que está sendo modelado.

Os métodos Pseudo-físicos, também chamados de métodos baseados em física, modelos matemáticos do movimento, com ou sem amortecimento, são usados para simular o movimento de partículas, que representarão os pontos sobre a superfície considerada. Esses modelos matemáticos são equações diferenciais que governam o movimento de cada ponto. A solução do sistema de equações diferenciais, que modelam o movimento das partículas, buscam encontrar a posição de equilíbrio e esta deve ocorrer sobre a superfície, determinando, assim, uma amostra dos pontos sobre a superfície dada.

Dentre os métodos estudados, entendemos que a obtenção de um algoritmo que satisfaça os requisitos básicos ainda é um desafio. Os algoritmos analisados exigem que os objetos geométricos sejam regulares, do ponto de vista da Geometria Diferencial, no mínimo de classe  $C^1$ .

Nosso trabalho objetiva apresentar um estudo das diversas técnicas para construção de uma aproximação poligonal conveniente a partir de objetos geométricos bidimensionais (superfícies) definidos de forma implícita. Com esse intuito, apresentamos uma revisão dos métodos de poligonização mais conhecidos e, posteriormente, uma

implementação do método *marching triangles* [30,31], onde uma aproximação polinomial, adaptativa e robusta, é obtida pela adaptação da geometria do objeto implícito, usando a curvatura local exata para controlar o comprimento da aresta da malha em cada ponto amostrado, e a abordagem de avanço de frentes para a geração da malha. Com isso, obtemos uma malha razoavelmente uniforme, com a geometria aproximada e a topologia exata do objeto amostrado utilizando uma estruturação (ou estrutura de dados) bastante simples.

## 1.1 Organização do Trabalho

No Capítulo 2 apresentamos resultados matemáticos importantes para o entendimento dos métodos analisados. Dentre esses resultados destacamos o estudo da Aplicação Normal de Gauss para superfícies implícitas e do Operador de Weingarten em  $\mathbb{R}^3$ .

No Capítulo 3 apresentamos uma revisão dos métodos usados para resolver o problema de poligonização de superfícies implícitas.

No Capítulo 4 discutimos, em detalhes, o método *Marching Triangles* dependente de curvatura e a implementação proposta.

No Capítulo 5 apresentamos os detalhes de implementação e os resultados obtidos.

Finalmente, no Capítulo 6 fazemos os comentários finais, com as conclusões e possíveis trabalhos futuros.

# Capítulo 2

## Resultados Importantes

Neste capítulo apresentamos, de uma forma sucinta, os principais resultados matemáticos usados nos métodos de poligonização estudados.

### 2.1 Superfícies Regulares

Segundo Gomes e Velho [18], um objeto gráfico é um subconjunto  $S \subset \mathbb{R}^m$ , juntamente com uma função  $f : S \subset \mathbb{R}^m \rightarrow \mathbb{R}^n$ , onde  $S$  é chamado de suporte geométrico e determina a geometria (as relações métricas) e a topologia (relações de conectividade) do objeto. A função  $f$  é chamada de função de atributos e especifica as propriedades do objeto gráfico. Os objetos gráficos são classificados de acordo com a dimensão do seu suporte geométrico. Os mais relevantes objetos para a Computação Gráfica são: ponto (dimensão zero), curvas (unidimensionais), superfícies e imagens (bidimensionais), volumes ou objetos espaciais (tridimensionais).

As superfícies são objetos gráficos bidimensionais que "moram" no espaço tridimensional. Nos processos gráficos, a descrição matemática mais popular desses objetos é a paramétrica, onde os pontos que constituem o objeto gráfico de interesse são dados diretamente por uma parametrização local que define uma aplicação do espaço dos parâmetros no espaço do objeto. Com o desenvolvimento e sofisticação crescente na área na modelagem de objetos tridimensionais, a partir dos anos 80, o interesse pela descrição implícita dos objetos gráficos tem crescido significativamente, com o surgimento de vários trabalhos nesse tema. Nesta formulação, os pontos pertencentes ao

objeto são dados indiretamente através de uma função que define a relação dos pontos do espaço tridimensional com o objeto.

Segundo DoCarmo [21] uma superfície regular é uma figura geométrica onde faz sentido falar em plano tangente em todos os seus pontos, definida parametricamente por:

**Definição 2.1** Um subconjunto  $S$  do Espaço Euclidiano  $\mathbb{R}^3$  é uma superfície regular se, para cada ponto  $p \in S$ , existe uma vizinhança  $V$  de  $p$  em  $\mathbb{R}^3$  e uma aplicação  $f : U \rightarrow V \cap S$  de um aberto  $U \subset \mathbb{R}^2$  sobre  $V \cap S \subset \mathbb{R}^3$  tal que (Figura 2.1)

- (i) A aplicação  $f$  é diferenciável, isto é, dados  $f(u, v) = (x(u, v), y(u, v), z(u, v))$ ,  $(u, v) \in U$ , todas as derivadas parciais das funções  $x(u, v)$ ,  $y(u, v)$  e  $z(u, v)$  existem e são contínuas em  $U$ .
- (ii)  $f$  é um homeomorfismo, isto é,  $f$  é bijetiva e contínua, e possui inversa contínua.
- (iii) Para todo  $q \in U$  a diferencial  $df_q : \mathbb{R}^2 \rightarrow \mathbb{R}^3$  é injetiva, ou seja, existe o plano tangente a  $S$  para todo  $q = (u, v)$  tal que  $p = f(q)$ .

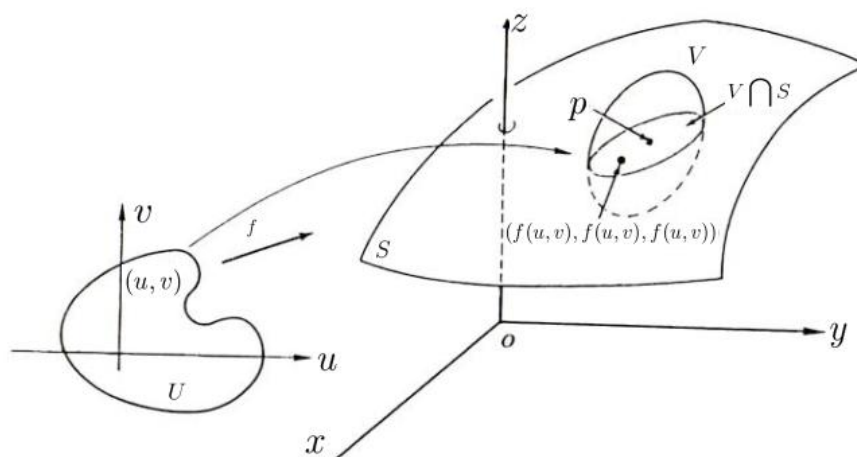


Figura 2.1: Superfície regular [21].

Geometricamente, uma superfície ser regular significa que não possui "bicos" e auto-interseções (Figura 2.2), que é a condição necessária para a existência de plano tangente em todos os pontos da superfície, sem ambigüidades.

Para ampliarmos a definição acima para superfícies com bordo também devemos considerar  $f$  como uma aplicação do semi-disco unitário

$$D = \{(x, y) \in \mathbb{R}^2; x^2 + y^2 < 1 \text{ e } y \geq 0\}$$

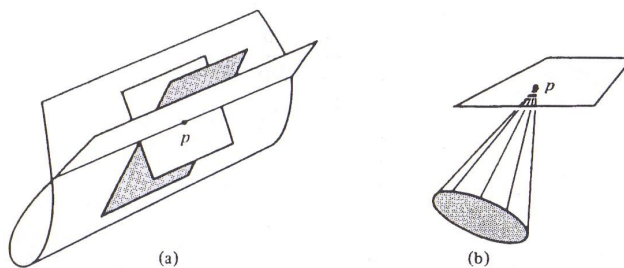


Figura 2.2: auto-interseções e bicos [21].

do semi-plano  $\{(x, y) \in \mathbb{R}^2; y \geq 0\}$ . Nesse caso nos pontos do bordo de  $S$ , a vizinhança coordenada  $V \cap S$  é homeomorfa ao semi-disco  $D$  e nos pontos interiores de  $S$  a vizinhança coordenada  $V \cap S$  é homeomorfa ao conjunto aberto  $U$ .

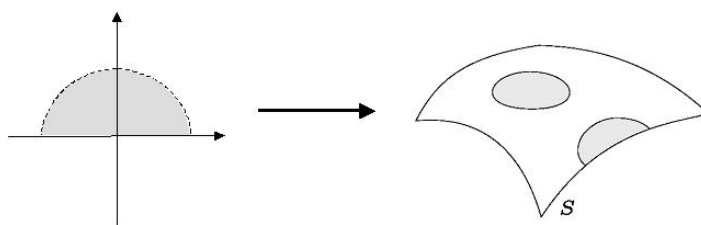


Figura 2.3: Homeomorfismo em semi-disco [18]

Os vetores tangentes  $f_u$  e  $f_v$ , em relação aos parâmetros  $u$  e  $v$ , e o vetor normal unitário  $N$  da superfície em um ponto qualquer  $p = f(u, v)$  são dados, respectivamente, por:

$$f_u(u, v) = \left( \frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial z}{\partial u} \right), \quad f_v(u, v) = \left( \frac{\partial x}{\partial v}, \frac{\partial y}{\partial v}, \frac{\partial z}{\partial v} \right),$$

$$N(u, v) = \frac{f_u(u, v) \times f_v(u, v)}{\|f_u(u, v) \times f_v(u, v)\|},$$

onde os vetores  $f_u(u, v)$  e  $f_v(u, v)$  controlam o lugar geométrico da superfície  $S$  e os vetores normais  $N(u, v)$  a suavidade de  $S$ . A função  $f$  é uma parametrização ou um sistema de coordenadas locais em uma vizinhança de  $p$  de  $S$ .

Uma outra forma de descrever uma superfície é usando a forma implícita, ou seja, uma superfície  $S \subseteq \mathbb{R}^3$  é descrita pela imagem inversa de uma função diferenciável  $f : W \subset \mathbb{R}^3 \rightarrow \mathbb{R}$ , onde  $W$  é um conjunto aberto que contém  $S$ .

$$S = f^{-1}(a) = \{(x, y, z) \in W \mid f(x, y, z) = a\}$$

Para preservar as suas características geométricas (superfície regular) o gradiente de  $f$  não deve se anular nos pontos de  $S$ , ou seja:

$$\nabla f = \text{grad}(f) = \left( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right) \neq 0$$

e, neste caso, dizemos que  $a \in f(W)$  é um valor regular de  $f$ .

O gradiente de  $f$  é uma função de  $W \subset \mathbb{R}^3$  em  $\mathbb{R}^3$ , assim seu diferencial  $d(\nabla f)$  é a matriz  $3 \times 3$ , conhecida como matriz Hessiana de  $f$ , onde:

$$d(\nabla f) = Hf(x_1, x_2, x_3) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_1 \partial x_3} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \frac{\partial^2 f}{\partial x_2 \partial x_3} \\ \frac{\partial^2 f}{\partial x_3 \partial x_1} & \frac{\partial^2 f}{\partial x_3 \partial x_2} & \frac{\partial^2 f}{\partial x_3 \partial x_3} \end{bmatrix}.$$

A seguinte proposição, cuja demonstração encontra-se em [21], nos dá as condições necessárias para que uma função  $f : U \subset \mathbb{R}^3 \rightarrow \mathbb{R}$  seja uma superfície regular.

**Proposição 2.2** *Se  $f : U \subset \mathbb{R}^3 \rightarrow \mathbb{R}$  é uma função diferenciável e  $a \in f(U)$  é um valor regular de  $f$ , então  $f^{-1}(a)$  é uma superfície regular de  $\mathbb{R}^3$ .*

Uma das características desta descrição é a facilidade na determinação de ponto-conjunto, uma vez que a função  $f$ , assim definida, divide os pontos do espaço onde está definida, em três conjuntos. Por exemplo, se considerarmos

$$S = f^{-1}(0), \quad f : W \subset \mathbb{R}^3 \rightarrow \mathbb{R},$$

uma superfície fechada, podemos dizer que o conjunto  $\{p \in W; f(p) < 0\}$  é formado pelos pontos  $p \in W$  que estão no interior da superfície  $S$ ,  $\{p \in W; f(p) = 0\} = S$  e  $\{p \in W; f(p) > 0\}$  é o conjunto dos pontos exteriores a superfície  $S$ .

Na próxima seção, apresentaremos alguns resultados que permitem a captura da geometria local das superfícies regulares. Eles estão intimamente relacionados com a questão de orientabilidade e garantem a existência de um plano tangente em qualquer ponto de  $S$  [21].

### 2.1.1 Aplicação Normal de Gauss

A Aplicação Normal de Gauss fornece uma medida de quão rapidamente o vetor normal de uma superfície regular  $S$  varia em uma vizinhança de um ponto  $p \in S$ . Essa medida está diretamente relacionada com a curvatura da superfície  $S$  em  $p$ .



Uma superfície que admite uma aplicação diferenciável  $N : U \subset S \rightarrow \mathbb{R}^3$  que associa a cada  $q \in U$  um vetor normal unitário  $N(q) \in \mathbb{R}^3$  a superfície  $S$  no ponto  $q$ , é dita orientável e a aplicação  $N$  denominada de campo diferenciável de vetores normais no aberto  $U \subset S$ .

A escolha desse campo  $N$  é chamada de orientação de  $S$ . A existência de vetores normais a superfície em todo ponto  $p \in S$ , determina um campo diferenciável sobre  $S$ , garante a existência de um plano tangente a superfície em cada ponto  $p \in S$ , denotado por  $T_p(S)$ , e induz a sua orientação [21].

**Definição 2.3** *Seja  $S \subset \mathbb{R}^3$  uma superfície com uma orientação  $N$ . A aplicação  $N : S \rightarrow \mathbb{R}^3$  toma seus valores na esfera unitária*

$$S^2 = \{(x, y, z) \in \mathbb{R}^3, x^2 + y^2 + z^2 = 1\}$$

*é chamada de Aplicação Normal de Gauss da Superfície  $S$ .*

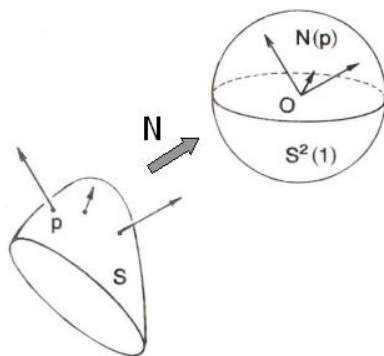


Figura 2.4: Aplicação Normal de Gauss [21].

A Aplicação Normal de Gauss é diferenciável e sua diferencial, denotada por  $dN_p$ , é uma aplicação linear que leva vetores do plano tangente da superfície  $S$  em  $p$ , em vetores no plano tangente da esfera unitária  $S^2$  em  $N(p)$ , ou seja,

$$dN_p : T_p(S) \rightarrow T_{N(p)}(S^2)$$

Como os planos  $T_p(S)$  e  $T_{N(p)}(S^2)$  são paralelos, considera-se

$$dN_p : T_p(S) \rightarrow T_p(S).$$

A aplicação  $dN_p : T_p(S) \rightarrow T_p(S)$  opera como segue: Para cada curva parametrizada diferenciável  $\alpha(t)$  sobre a superfície  $S$ , com  $\alpha(0) = p$  e  $\alpha'(0) = v \in T_p(S)$ , considere uma curva parametrizada  $N \circ \alpha(t) = N(t)$  na esfera  $S^2$  (Figura 2.5).

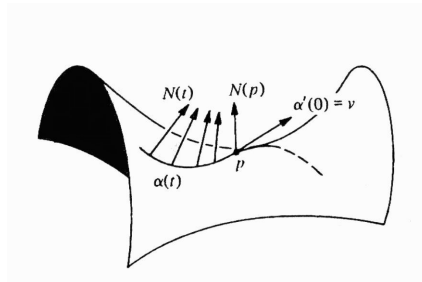


Figura 2.5: Caminho diferenciável sobre a sela [21].

O vetor tangente  $N'(0) = dN_p(\alpha'(0))$  em  $T_pS$  mede a taxa de variação do vetor normal unitário  $N$ , restrito à curva  $\alpha(t)$  em  $t = 0$ . Geometricamente, a função  $dN_p$  mede quanto  $N$  "afasta-se" de  $N(p)$  em uma vizinhança de  $p$ . A aplicação linear  $dN_p$  independe da escolha da curva  $\alpha(t) \in S$ , isto é,  $dN_p(v)$  depende apenas das coordenadas do vetor  $v = \alpha'(0)$  e não do comportamento da curva nas vizinhanças de  $p = \alpha(0)$ .

Observe que

$$N'(0) = \frac{d}{dt}(N \circ \alpha(t))_{t=0} = dN_p(\alpha'(0)) = dN_p \circ \alpha'(0)$$

cuja notação matricial é

$$DN_p(\alpha'(0)) = DN_{\alpha(0)} \cdot \alpha'(0) = DN_p \cdot v.$$

**Definição 2.4** *Seja  $C$  uma curva regular em  $S$  que passa por um ponto  $p \in S$ ,  $k$  a curvatura de  $C$  em  $p$  e  $\cos \theta = \langle n, N \rangle$ , onde  $n$  é o vetor normal unitário a  $C$  e  $N$  é o normal unitário de  $S$  em  $p$ . O número*

$$k_n = k \cdot \cos \theta$$

*denomina-se curvatura normal de  $C \subset S$  em  $p$ .*

**Proposição 2.5 (Meusnier)** *Todas as curvas sobre uma superfície  $S$  que possuem uma mesma reta tangente em  $p \in S$  tem a mesma curvatura normal em  $p \in S$ .*

A proposição acima nos permite falar da curvatura normal ao longo de uma dada direção em  $p$ , ou seja, dado um vetor unitário  $v \in T_p(S)$ , a interseção de  $S$  com o plano que contém  $v$  e  $N(p)$  é chamada de seção normal de  $S$  em  $p$  na direção de  $v$ , portanto, a proposição acima afirma que o valor absoluto da curvatura normal em  $p$  de uma curva  $\alpha(t) \in S$ , tal que  $v = \alpha'(0)$ , é igual à curvatura da seção normal de  $S$  em  $p$  na direção de  $v$ .

**Proposição 2.6** *A diferencial  $dN_p : T_p S \rightarrow T_p S$  da Aplicação Normal de Gauss é uma aplicação auto-adjunta.*

Uma vez que  $dN_p$  é linear e auto-adjunta (veja apêndice A), podemos associá-la a uma forma quadrática  $Q(v) = -\langle dN_p(v), v \rangle$ , onde  $v \in T_p S$  e que para cada  $p \in S$  existe uma base ortonormal  $\{b_1, b_2\}$  de  $T_p(S)$  tal que a matriz associada a  $dN_p$  relativa a esta base é diagonal e  $dN_p(b_1) = -k_1 b_1$  e  $dN_p(b_2) = -k_2 b_2$ , onde  $-k_1$  e  $-k_2$  são os autovalores e  $b_1$  e  $b_2$  são os autovetores associados. Além disso, os elementos da diagonal são os negativos máximo e o mínimo da forma quadrática correspondente restrita ao círculo unitário de  $T_p(S)$  [22].

**Definição 2.7** *A curvatura normal máxima e a curvatura normal mínima, denotadas respectivamente por  $k_1$  e  $k_2$ , são denominadas curvaturas principais em  $p$ . Suas direções correspondentes, dadas pelos autovetores  $b_1$  e  $b_2$  são chamadas de direções principais em  $p$ .*

O determinante da matriz associada a aplicação  $dN_p$  é o produto  $(-k_1)(-k_2) = k_1 k_2$  das curvaturas principais e é chamado de Curvatura Gaussiana  $K$  de  $S$  em  $p$ . Já o negativo da metade do traço de  $dN_p$ ,  $\frac{k_1 + k_2}{2}$ , é chamado de curvatura Média  $H$  de  $S$  em  $P$ , ou seja,

$$K = k_1 k_2 \quad \text{e} \quad H = -\frac{k_1 + k_2}{2}. \quad (2.1)$$

Maiores detalhes sobre este item, assim como as demonstrações das proposições citadas encontram-se em DoCarmo [21].

## 2.2 Aplicação de Weingarten

Os vetores normais unitários a superfície  $S$  em  $p$   $N(p)$  e  $\nabla f(p)$  são ortogonais a  $T_p(S)$ , onde  $p \in S = f^{-1}(0)$ . Assim, podemos definir uma função  $\alpha : W \subset \mathbb{R}^3 \rightarrow \mathbb{R}$  de forma que  $N(p) = \alpha(p) \nabla f(p)$ , ou seja,  $\alpha(p) = \frac{1}{\|\nabla f(p)\|}$ .

Consideremos as funções  $\alpha : W \subset \mathbb{R}^3 \rightarrow \mathbb{R}$  e  $\tilde{N} : W \subset \mathbb{R}^3 \rightarrow \mathbb{R}^3$ , de tal forma que  $\tilde{N}(p) = N(p)$  para todo  $p \in S$ . O campo vetorial  $\tilde{N}$  é uma extensão de  $N$ , definido no mesmo domínio de  $f$  [23, 32]. Logo podemos escrever

$$\tilde{N} : W \subset \mathbb{R}^3 \rightarrow \mathbb{R}^3$$

$$\tilde{N}(p) = \alpha(p) \cdot \nabla f(p).$$

Seja  $L$  uma aplicação de  $T_p(S)$  em  $T_p(S)$  definida por  $L(v) = d\tilde{N}_p(v)$ , chamada de aplicação de Weingarten [32]. Como  $\tilde{N}(p) = N(p)$  para todo  $p \in S$  temos  $L(v) = dN_p(v)$  para todo  $p \in S$ .

Os elementos  $l_i^j$  associada a transformação  $L$  com relação a uma base ortonormal,  $\{b_1, b_2\}$ , do plano tangente à superfície  $S$  em  $p$  são dados pela expressão:

$$l_{ij} = \langle b_i, L(b_j) \rangle = b_i^t L(b_j),$$

onde  $b_i^t$  denota a transposta do vetor  $b_i$  e

$$l = \begin{bmatrix} b_1^t L(b_1) & b_1^t L(b_2) \\ b_2^t L(b_1) & b_2^t L(b_2) \end{bmatrix}.$$

O determinante e a metade do traço da matriz  $l$ , definida no parágrafo anterior, determinam as curvatura Gaussiana e média, de  $S$  em  $p$ , respectivamente [21, 32].

Para facilitar os cálculos, Hartmann [31] sugere considerar as bases  $\{c_1, c_2\}$ ,  $\{c_2, c_3\}$  e  $\{c_1, c_3\}$ , onde

$$\begin{aligned} c_1 &= \nabla f(p) \times e_1 = (0, f_z, -f_y) \\ c_2 &= \nabla f(p) \times e_2 = (-f_z, 0, f_x) \\ c_3 &= \nabla f(p) \times e_3 = (f_y, -f_x, 0) \end{aligned}$$

as quais podem ser ortonormalizadas usando, por exemplo, o processo de Gram-Schmidt obtendo

$$\begin{aligned} b_1 &= \frac{c_1}{\|c_1\|} = \frac{1}{\|c_1\|} (0, f_z, -f_y), \\ b_2 &= \frac{c_2}{\|c_2\|} = \frac{1}{\|c_2\|} (-f_z, 0, f_x) \quad \text{e} \\ b_3 &= \frac{c_3}{\|c_3\|} = \frac{1}{\|c_3\|} (f_y, -f_x, 0) \end{aligned}$$

A partir das curvaturas Gaussiana e média podemos determinar o valor das curvaturas principais através da relação dada em (2.1) que correspondem à soma e ao produto das raízes de uma equação do segundo grau:

$$x^2 - 2Hx + K = 0$$

cujas raízes são:

$$k_1 = H + \sqrt{H^2 - 4K} \quad (2.2)$$

$$k_2 = H - \sqrt{H^2 - 4K}$$

# Capítulo 3

## Métodos de Poligonização de Superfícies Implícitas

### 3.1 Introdução

Neste capítulo, discutiremos os principais métodos utilizados para a poligonização de superfícies definidas implicitamente por equações matemáticas. Como já mencionamos, obter uma representação poligonal significa obter uma lista de vértices e suas interconexões. Os algoritmos variam de acordo com essas duas etapas conceituais, ou seja, de acordo com a forma como os vértices são obtidos (amostragem) e como esses vértices serão interligados de forma a gerar os contornos das faces da representação poligonal (estruturação), como ilustra a Figura 3.1(a) e (b). Os polígonos que representam as faces da superfície poliédrica, podem não ser planares, mas sempre é possível, através de uma triangulação, transformar as faces não planares em faces planares (Apêndice B).

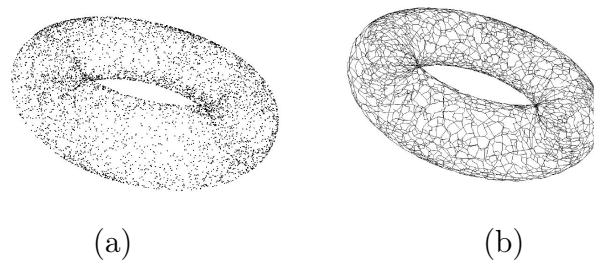


Figura 3.1: Amostragem de pontos (a) e estruturação (b) [7].

Em nosso estudo dos métodos de poligonização de superfícies, procuramos dar uma abordagem histórica, iniciando com os trabalhos de Bloomenthal [3–6] que foi um dos pioneiros no estudo dos métodos poligonizadores para superfícies implícitas. Neles, a obtenção dos pontos e sua estruturação são discutidas indistiguivelmente, focando a execução do algoritmo. Nos trabalhos de Luiz Henrique de Figueiredo [7, 8, 14], tais métodos são chamados de "*métodos clássicos*" e sua análise é feita em dois estágios distintos: fase de amostragem e fase de estruturação. Essa separação tem por objetivo identificar os problemas particulares a cada fase [7].

A seguir passaremos a descrever os métodos estudados seguindo a classificação dada por nós, a saber:

- Métodos Geométricos: particionamento celular ou volumétrico e rastreamento de superfície; e
- Métodos Pseudo-físicos.

## 3.2 Particionamento Celular ou Volumétrico

Tais métodos caracterizam-se pela decomposição do espaço ambiente onde o objeto está imerso, em porções menores, denominadas de células ou *voxels*. A partir dessas células, são obtidas as aproximações poligonais através da interseção de cada célula com o objeto geométrico em questão. As células que possuem partes do objeto geométrico são denominadas de células transversas [4, 17, 19].

Uma vez que trabalha-se com o particionamento do espaço ambiente, torna-se necessário estimar os parâmetros que definem o espaço ocupado pelo objeto e assim determinar um envoltório inicial, geralmente um paralelepípedo, chamado de célula raiz (*root cell*) [4], como mostra a Figura 3.2. A célula raiz é decomposta em células menores e, a partir delas, por meio de suas arestas e vértices, uma malha é gerada.

A decomposição celular em cubos [4, 15, 16] surgiu com os primeiros algoritmos desenvolvidos e ainda são muito utilizados. Nesta abordagem, a geração dos polígonos, que constituirão a malha, é feita a partir das arestas e vértices das células seguindo, basicamente, os seguintes passos: Cálculo das polaridades em cada vértice; cálculo dos pontos da superfície e geração do polígono.

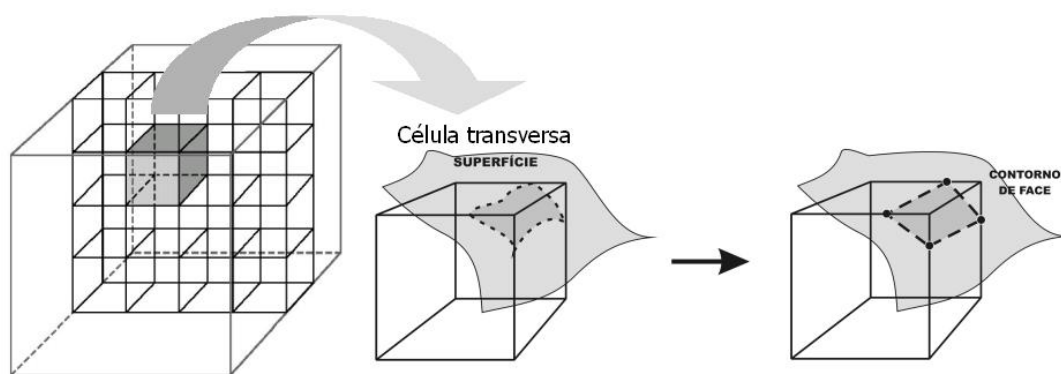


Figura 3.2: Particionamento em células cúbicas [4].

Para o cálculo das polaridades em cada vértice  $v = (x, y, z)$ , de cada uma das células, é aplicada a função  $f(x, y, z) = 0$  que define implicitamente a superfície, pela equação  $f(x, y, z) = 0$ , associando, a esses vértices, um valor real, que pode ser positivo, negativo ou zero, relacionado a orientação da superfície. Aos valores, positivo ou negativo, que  $f(x, y, z)$  pode assumir, damos o nome de polaridade. Se as arestas das células possuírem pontos extremos (vértices) com polaridades (sinais) distintas, então a célula é transversa, isto é, a célula é atravessada pela superfície a ser imageada, desta forma é obtida uma amostra de pontos da superfície.

O uso das polaridades dos vértices para a identificação de células transversas podem apresentar alguns problemas de ambigüidades topológicas na geração da malha, os quais serão discutidas mais tarde (seção 3.2.3).

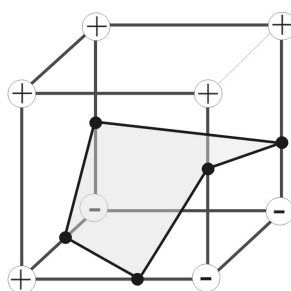


Figura 3.3: Particionamento em células cúbicas

No cálculo dos pontos da superfície destacamos o método da bissetção [24] e a utilização da Análise Intervalar através da Aritmética Intervalar e da Aritmética Afim [2, 10, 11](Apêndice D).

Na geração do polígono, os pontos encontrados são ligados de maneira a criar um



polígono, possivelmente não-planar. Esses polígonos podem ser decompostos em vários triângulos. Nesta etapa, a ordenação dos pontos infere uma orientação a este trecho da superfície.

A forma como os voxels são analisados sugerem duas abordagens distintas, a saber: Enumeração exaustiva e Subdivisão ou enumeração hierárquica.

Na enumeração exaustiva, a célula raiz é particionada em um reticulado regular e os voxels são analisados um a um em busca de células transversas [3–5,8] e na subdivisão ou enumeração hierárquica, excetuando-se a célula raiz, a subdivisão ocorre recursivamente somente nas células transversas. Os critérios para a realização da subdivisão, ou a sua parada, também envolvem a definição de um número máximo de subdivisões possíveis ou se o tamanho mínimo de célula (voxel) foi atingido, entre outros [4].

O *octree* é a técnica mais usada na subdivisão de células (Figura 3.4) e consiste em uma estrutura de dados hierárquica onde uma célula transversa subdivide-se em 8 novas células até que seja atingido o tamanho mínimo possível [19,27]. Outras formas de subdivisão conhecidas são: *KD-tree* e *bintree* [3].

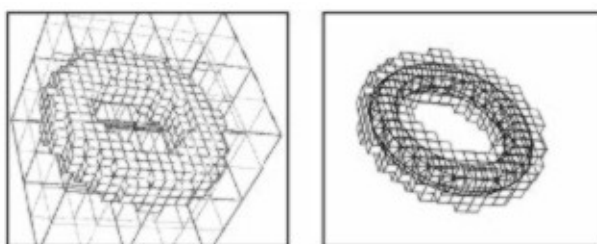


Figura 3.4: Subdivisão aplicada a um toro [6].

A ordenação dos pontos e, conseqüentemente, a criação do polígono exige cuidados adicionais uma vez que, nestes casos, as células apresentam tamanhos distintos. Segundo [4], os seguintes critérios podem ser usados para a subdivisão de uma célula:

- (i) Se a aresta da célula (cor preta) possui dois vértices com mesma polaridade e o ponto médio desta mesma aresta possuir polaridade oposta (Figura 3.5) então a célula em questão deve ser subdividida, bem como as células adjacentes (que compartilham a mesma aresta).
- (ii) Se as arestas da célula conectam vértices de mesmo sinal e o baricentro possui sinal diferente (Figura 3.6), então, neste caso, somente a face vizinha será



Figura 3.5: Vértices com polaridades opostas ao ponto médio [4].

subdividida;



Figura 3.6: Vértices da face com sinal oposto ao baricentro [4].

Bloomenthal [4] apresenta um algoritmo, utilizando octree que, além de solucionar as anomalias que aparecem nas Figuras 3.5 e 3.6, também resolvem o problema ilustrado na Figura 3.7, que ocorre quando dois cubos que compartilham a mesma face e em um deles houve a subdivisão e o outro permaneceu inalterado (Figura 3.7(b)). A ordenação dar-se-á a partir de um ponto pertencente ao bordo da face comum as duas células adjacentes. A partir do ponto (1) (Figura 3.7(a)), seguindo orientação horária ao longo da face mais subdividida até encontrar o próximo ponto (2). O processo segue até chegar ao ponto (4) que está no bordo da face original.

O uso de células cúbicas, apesar de ser uma solução natural para particionamento do espaço, apresenta problemas em cada uma das fases acima descritas tais como a subdivisão excessiva das células e a não garantia da eliminação das ambigüidades topológicas.

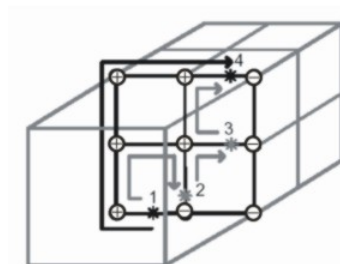


Figura 3.7: (a)

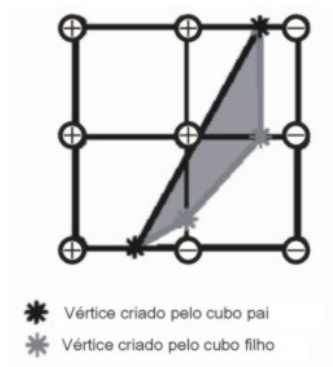


Figura 3.8: (b)

### 3.2.1 Subdivisão Adaptativa de Cubos

Os melhores resultados do método de enumeração são obtidos a partir de uma *abordagem adaptativa* [4, 5]. A poligonização adaptativa resulta de melhorias sobre os algoritmos de subdivisão. Acrescentando um critério mais seletivo para que detalhes da superfície, geralmente relacionados a características locais, como curvatura e tangência, sejam mais precisos sem comprometer o desempenho do algoritmo [10].

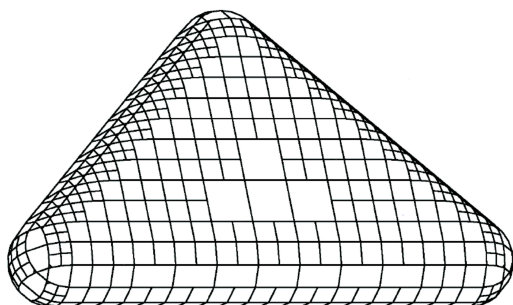


Figura 3.9: Malha adaptativa [4].

### 3.2.2 O Uso das Células Tetraédricas

Os problemas de ambigüidades topológicas encontrados no uso de células cúbicas foram superados através da decomposição do espaço em tetraedros [4, 17]. Nesta abordagem vértices de sinais opostos podem ser separados por meio de um plano [17]. Isso não diminui a importância da decomposição do espaço em células cúbicas, já que é a partir

dessa decomposição que os tetraedros são obtidos. Segundo [18] é possível decompor um cubo em 5, 6 e 12 tetraedros, conforme ilustram as Figuras 3.10(a), 3.10(b) e 3.10(c), respectivamente.

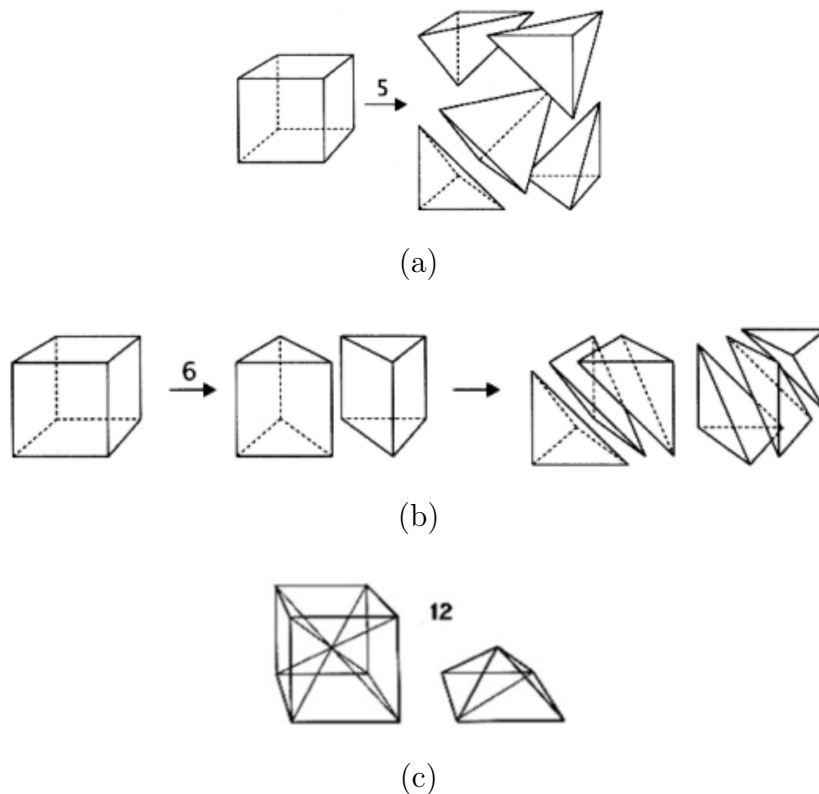
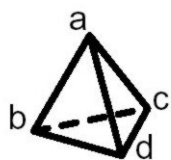


Figura 3.10: Decomposição do cubo em tetraedros [28].

A partir da decomposição do espaço em tetraedros, segundo Ning e Bloomenthal [5] os polígonos podem ser gerados através do uso de uma lista de vértices (tabela) que contém todas as 16 combinações possíveis de polaridades (Figura 3.2.2).

A triangulação feita a partir de um cubo decomposto em seis tetraedros é conhecida como triangulação de *Coxeter-Freudentahl-Kuhn* ou triangulação CFK.

A decomposição de um cubo em cinco ou seis tetraedros não exige o cálculo de pontos extras, como no caso da decomposição em 12 tetraedros, mas a ordem em que os vértices, das faces triangulares, são ligados são decisivos para a obtenção de malhas que representem exatamente a topologia das superfícies amostradas. Para a decomposição em cinco tetraedros é preciso impor a cada vértice do tetraedro uma orientação indicando os vértices do topo (*top*), direita (*right*), esquerda (*left*), próximo (*near*) e longe (*far*) como mostra a Figura 3.12.



Vértices Positivos	Ordem de Vértices
caso $\{\}$	$\{\}$
caso $\{a\}$	$\{ab, ad, ac\}$
caso $\{b\}$	$\{ab, bc, bd\}$
caso $\{c\}$	$\{ac, cd, bc\}$
caso $\{d\}$	$\{bd, cd, ad\}$
caso $\{ab\}$	$\{ad, ac, bc, bd\}$
caso $\{ac\}$	$\{ab, ad, cd, bc\}$
caso $\{ad\}$	$\{ab, bd, cd, ac\}$
caso $\{bc\}$	$\{ab, ac, cd, bd\}$
caso $\{bd\}$	$\{ad, ab, bc, cd\}$
caso $\{cd\}$	$\{ad, bd, bc, ac\}$
caso $\{abc\}$	$\{bd, ad, cd\}$
caso $\{acd\}$	$\{ab, bd, bc\}$
caso $\{abd\}$	$\{cd, ac, bc\}$
caso $\{bcd\}$	$\{ab, ac, ad\}$
caso $\{abcd\}$	$\{\}$

Figura 3.11: Lista de vértices [5].

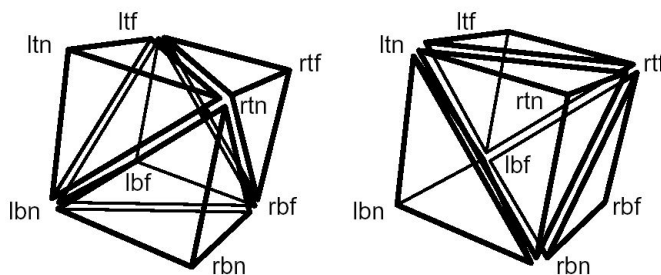


Figura 3.12: Orientação dos triângulos de um tetraedro [5].

### 3.2.3 Ambigüidades Topológicas

Como mencionado no início desta seção, o uso de células cúbicas pode gerar contornos de face inconsistentes, onde uma mesma configuração, de pontos de interseção de uma célula com o objeto, pode gerar trechos de superfícies com geometria ou topologias distintas (Figura 3.13).

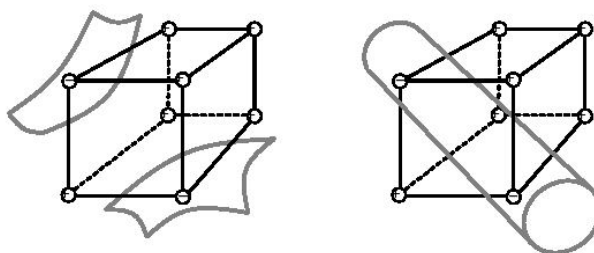


Figura 3.13: Interpretação distinta para uma mesma configuração de vértices. [5]

Em outras palavras, o modelo de superfície gerado a partir da amostra obtida com as células cúbicas podem conter fendas ou quinas, ou outras configurações inexistentes no objeto que está sendo modelado, esses casos são denominados de ambigüidades topológicas.

Bloomenthal e Ning [5] descrevem diversas estratégias para eliminar esses artefatos. Algumas já foram mencionadas em seções anteriores e serão brevemente descritas abaixo. Mencionaremos as três técnicas mais usadas:

**Decomposição celular.** Procura-se subdividir as células ambíguas em outras menores, que não apresentem ambigüidades. Tais células podem ser cúbicas ou tetraédricas. As primeiras podem ter um custo computacional alto, a depender do número de cubos gerados pelo polgonizador, e não resolver definitivamente a questão da ambigüidade. O uso de células tetraédricas é bem mais eficiente embora seu uso implique no aumento da complexidade da implementação e no número de faces na malha gerada.

**Polaridade Preferida.** A idéia consiste em solucionar o problema apresentado na Figura 3.13, inserindo um bom critério de separação dos vértices de mesmo sinal. Em [4] os pontos da superfície são ligados seguindo a direção do vértice positivo da aresta, no sentido horário ao longo da face (Figura 3.14).

**Inferência topológica.** Alguns métodos tentam inferir a topologia correta da superfície a partir do cálculo de alguns valores adicionais e assim gerar polígonos que constituam uma representação da superfície. Destacam-se os métodos:

- **Reamostragem de Centro/Face:** Como o próprio nome já diz, calcula-se a polaridade do centro de cada face da célula. Como mostra a Figura 3.15 o método pode, em alguns casos, não apresentar bom resultado [5].
- **Contorno Bilinear:** Usa um interpolador bilinear em cada face do cubo. O con-

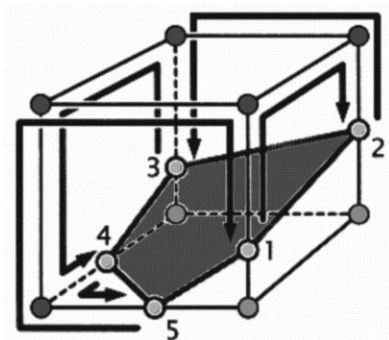


Figura 3.14: Determinação da polaridade dos vértices [4].



Figura 3.15: Reamostragem do centro de uma face (falha) [5]

torno da face é hiperbólico e suas direções são facilmente obtidas por amostras de face. Este método produz bons resultados se o interpolador bilinear for uma boa estimativa da função entre os pontos calculados [5, 16].

### 3.3 Rastreio de Superfície

O foco destes métodos deixa de ser a decomposição celular do espaço onde está imersa a superfície [4] e passa a ser a própria superfície. Fixa-se um ponto, ou uma pequena célula que contém pontos da superfície, e os demais pontos (ou demais células) são obtidos mediante operações realizadas a partir deste ponto inicial. Descreveremos a seguir duas abordagens: Continuação (Linear por Partes e Preditor-Corretor) e o *Marching Triangles* com avanço de frentes.

#### 3.3.1 Continuação Linear por Partes

No método da continuação linear por partes [4, 6] cada objeto gráfico disjunto deve possuir uma pequena célula transversa denominada de célula semente (*seed cell*). Novos cubos, com o mesmo tamanho da célula semente e contendo partes da superfície, são

criados ao longo das faces transversas até que toda superfície esteja contida nestas células (Figura 3.16). As células devem ser pequenas o suficiente para manter os detalhes do objeto a ser modelado. A malha poderá ser refinada através de métodos de subdivisão já mencionados nas seções anteriores.

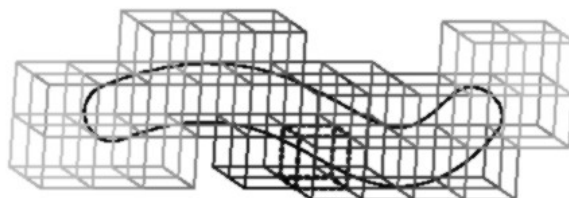


Figura 3.16: Continuação Linear por Partes [4].

### 3.3.2 Continuação Preditor-Corretor

Como o próprio nome sugere, este método [4, 6] encontra os pontos do objeto geométrico por meio de um ponto inicial. As aproximações são obtidas a partir do plano tangente a superfície via iteração de Newton (veja Apêndice B). Este método apresenta as seguintes dificuldades [6]:

- Os vértices gerados não estão intrinsecamente ordenados, isto é, não existe *a priori* a informação de como gerar o contorno de face;
- A detecção de sobreposições, pois todos os pontos, com exceção da primeira célula semente (*seed cell*), são aproximações.

### 3.3.3 *Marching Triangles*

Em [31], Hartmann apresenta um algoritmo preditor-corretor chamado *Marching Triangles* para poligonização de superfícies implícitas. Nele a obtenção dos pontos sobre a superfície é feita através do método da iteração de Newton e o problema da ordenação dos vértices é solucionado localmente.

Este método usa o avanço de frentes para obter uma malha, ou seja, a malha é construída progressivamente gerando novas faces a partir do seu bordo. Este método exige um especial cuidado com as junções ou separações de frentes.



A partir de um ponto  $q$  localizado nas proximidades da superfície determina-se um ponto correspondente  $p_1$  sobre a mesma (Figura 4.2).

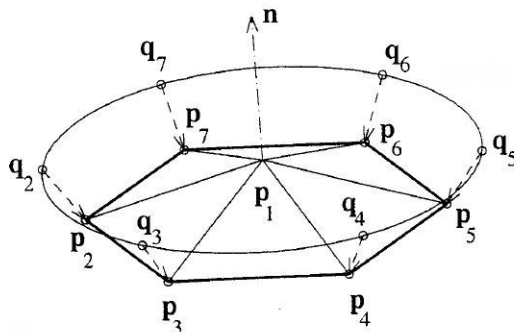


Figura 3.17: Plano tangente a superfície no ponto  $p_1$  [31].

Os demais pontos, sobre a superfície, são obtidos com o auxílio do plano tangente a superfície em  $p_1$ . A discussão detalhada de como os pontos da superfícies são calculados a partir de outros pontos determinados sobre o plano tangente e a propagação de frentes será apresentada no Capítulo 4.

Este método apresenta as seguintes vantagens em relação aos demais métodos estudados:

- A estrutura de dados utilizada é simples;
- Torna possível a geração de uma malha poligonal coerente para interseção de retalhos de superfícies.
- Além de superfícies implícitas, também pode ser usado para imagear superfícies paramétricas.

Em [30], são adicionados ao *Marching Triangles* critérios adaptativos utilizando o raio de curvatura [21] da superfície, como um critério de cálculo para o comprimento das arestas dos triângulos. Desta forma, os triângulos gerados em regiões onde a superfície apresenta maior curvatura são menores que aqueles gerados em áreas mais planas (curvaturas menores).

## 3.4 Pseudo-Físicos

Nos métodos Pseudo-Físicos [7, 8, 14], usados na modelagem de curvas e superfícies, a amostragem é feita via modelos físicos discretos. Esses Métodos surgiram da necessidade de modelar fenômenos que ocorrem na natureza, tais como cachoeiras, chamas, neblina, etc. O foco principal desses métodos está na integração (resolução) das equações diferenciais ordinárias relacionadas ao gradiente da função implícita  $f$  que define a superfície, esses métodos usam sistemas de partículas e o modelo massa-mola.

### 3.4.1 Sistema de Partículas

Em [8] o sistema de partículas usado para amostragem provém de uma função potencial  $|F|$ , onde  $F : \mathbb{R}^n \rightarrow \mathbb{R}$  tal que  $F(x, y, z) = 0$  define a superfície implícita. As partículas, ou pontos sobre a superfície, irão buscar a posição de equilíbrio sobre  $S = F^{-1}(0)$ , onde a energia potencial é mínima. O uso de  $\nabla F$  como campo vetorial sobre  $S$  implica na seguinte equação do movimento:

$$\frac{d^2x}{dt^2} + \gamma \frac{dx}{dt} + \text{sign}(F)\nabla F = 0,$$

onde  $x$  representa a posição das partículas em um instante  $t$  e  $\gamma$  representa o coeficiente de amortecimento.

O algoritmo para determinar a posição da partícula em cada instante usa o método de Euler descrito abaixo:

**Algorithm 3.1** *Equação Movimento(Dinâmica)*

- 1:  $y \leftarrow h(x)$
- 2:  $v \leftarrow v + \delta(F(x) - \gamma v)$  ▷ *Preditor de Euler para velocidade*
- 3:  $x \leftarrow x + \delta F(x)$  ▷ *Preditor de Euler modificado para posição*
- 4: **se**  $\text{sign}(y) \neq \text{sign}(h(x))$  **então** ▷ *verifica se houve interseção*
- 5:      $\delta \leftarrow \frac{\delta}{2}$  ▷ *controle do passo*
- 6:      $v \leftarrow 0$  ▷ *reinicia velocidade da partícula*
- 7: **fim se**

Considerando o movimento das partículas sem amortecimento, o modelo matemático

é simplificado e dado pela equação [14]

$$\frac{dx}{dt} + \text{sign}(F)\nabla F = 0,$$

neste caso, temos também uma simplificação do algoritmo para obtenção dos pontos da amostra:

**Algorithm 3.2** *Equação Movimento*

- 1:  $y \leftarrow h(x)$
- 2:  $x \leftarrow x + \delta F(x)$
- 3: **se**  $\text{sign}(y) \neq \text{sign}(h(x))$  **então**
- 4:      $\delta \leftarrow \frac{\delta}{2}$
- 5: **fim se**

Este método apresenta uma desvantagem relativa a concentração de amostras obtidas, que pode ser demasiadamente grande, isto se deve ao fato de que muitos pontos podem estar repetidos, concentrados em locais de alta curvatura ou ainda incidindo sobre falsos atratores, ocasionando, com isso, um excesso de informação a ser usado na fase de estruturação. Nesses casos, uma reamostragem pode ser necessária para evitar a alta concentração de pontos em uma mesma região uma vez que os cálculos necessários para a estruturação estão fortemente acopladas.

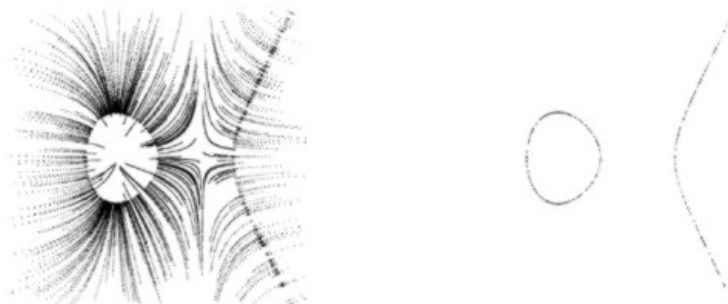


Figura 3.18: Curva gerada por partículas [7].

Uma forma de eliminar o excesso de pontos coincidentes é o uso do *Bucketing Algorithm* [14]. Ele funciona em tempo linear e age da seguinte forma: Divide o fecho de pontos da amostra em *buckets* (regiões) regulares de tamanho  $\delta$ . Todos os pontos do *bucket* são colapsados em um. A escolha realiza-se elegendo o elemento mais representativo ou via baricentro dos pontos do *bucket*.

Figueiredo [7] sugere o uso do método apresentado por Diestel [26] para a estruturação dos pontos.

### 3.4.2 Sistema Massa-mola

Em [8] é usado o sistema massa-mola para obter uma boa distribuição dos pontos. O sistema massa-mola também pode ser visto como um sistema de partículas interligadas por forças representadas pelas molas. A força exercida por estas molas depende da distância entre as partículas.

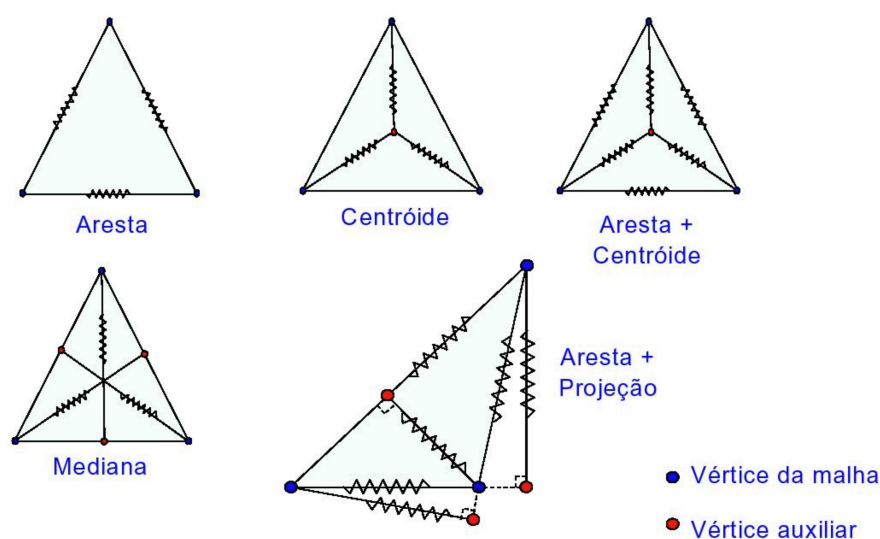


Figura 3.19: Possíveis configurações de molas.

O sistema massa-mola criado em [8] está associado a triangulação de Freudenthal [5] e sujeito a forças de deformação provenientes do campo gradiente associado a superfície implícita.

Para a criação do reticulado massa-mola são dados os seguintes passos:

- (i) A triangulação de Freudenthal é criada sobre uma porção limitada do espaço que contém a superfície;
- (ii) São identificados todos os simplejos (tetraedros quasi-angulares) transversos (aqueles que contém partes de  $S$ ). Eles constituem o *complexo simplicial* [20]
- (iii) O sistema massa-mola é criado associando nós e molas aos vértices e arestas do complexo simplicial.

Após a criação da malha usa-se uma abordagem pseudo-física de modo a obter a triangulação final. [8].

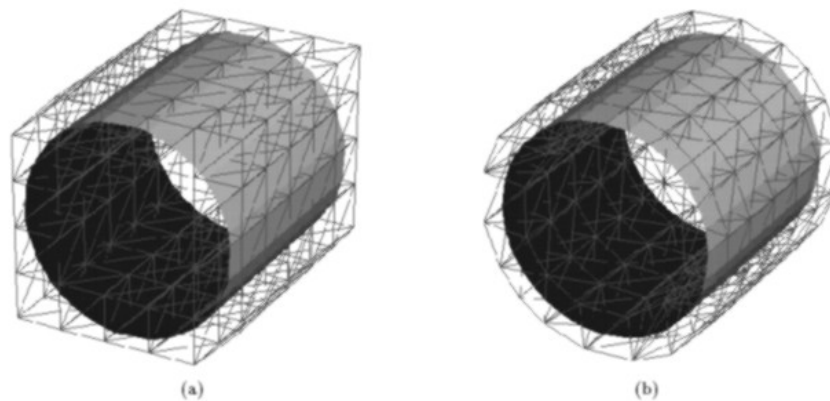


Figura 3.20: Malha antes (a) e depois (b) da deformação [8].

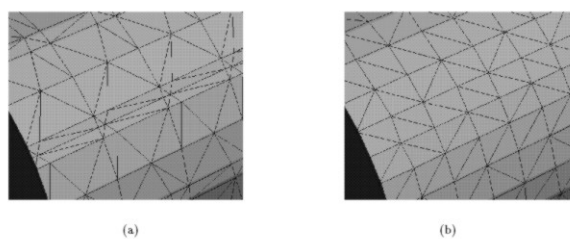


Figura 3.21: Detalhe da malha antes (a) e depois (b) da deformação [8].

# Capítulo 4

## O Algoritmo *Marching Triangles*

Neste Capítulo, vamos expor, com detalhes, o algoritmo *Marching Triangles* apresentado em Hartmann [31] e a inserção de critérios adicionais para melhoria do algoritmo em questão, apresentado por [30].

Como já mencionamos, o *Marching Triangles* cria uma malha inicial cujo bordo é chamado de frente atual (Figura 4.1), a partir dela os demais pontos são obtidos e novos triângulos são gerados propiciando a expansão da frente. A medida em que a frente se propaga pela superfície podem ocorrer a separação de uma frente em duas ou a união de duas frentes distintas.

Os passos citados nas seções 4.1, 4.2, 4.3 e 4.4 são executados até que a frente consista de apenas 3 pontos e o último triângulo da frente atual é gerado, ou todos os pontos da frente atual atingirem um envoltório limitante da superfície. Se houver uma frente sobrando, resultante da separação da frente atual, ela será promovida a frente atual e os passos serão novamente repetidos. O algoritmo termina quando não existirem frentes sobrando.

O pseudo-código usado como modelo da implementação será descrito e comentado no Capítulo 5.

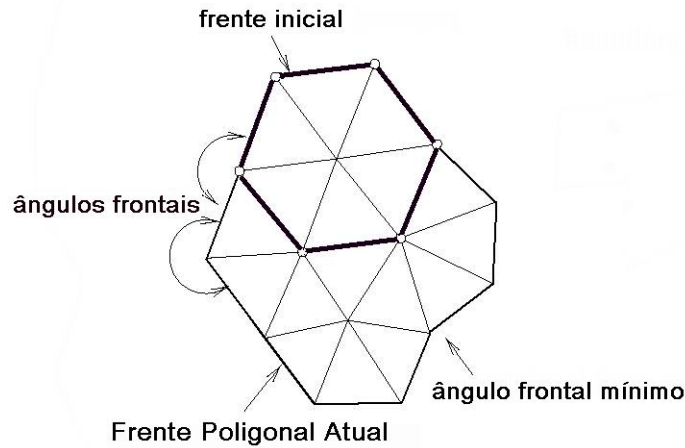


Figura 4.1: Propagação de frentes [31].

## 4.1 Determinação da Frente Inicial

Consideremos uma superfície implícita dada por  $S = f^{-1}(0)$ , onde  $f : \mathbb{R}^3 \rightarrow \mathbb{R}$  e  $f^{-1}(0) = \{(x, y, z) \mid f(x, y, z) = 0\}$ , cujo gradiente  $\nabla f$  existe e não é identicamente nulo para todo ponto  $p \in S$  (condição de regularidade), e um ponto  $q$  arbitrário em uma vizinhança da superfície  $S$ .

Primeiramente determinamos um ponto  $p$  da superfície que esteja próximo ao ponto  $q$  e um referencial ortonormal  $\{p, (n, t_1, t_2)\}$  por meio do procedimento *Surfacepoint*, proposto por Hartmann [31]:

### Algorithm 4.1 *Surfacepoint*

1: **procedimento** SURFACEPOINT

2:  $u_0 = q$  ▷  $u_0$  recebe ponto próximo à superfície

3: **repita** ▷ Passo de Newton

4:  $u_{k+1} \leftarrow u_k - \frac{f(u_k)}{\|\nabla f(u_k)\|^2} \nabla f(u_k)$

5: **até**  $\|u_{k+1} - u_k\| < \epsilon$

6:  $p \leftarrow u_{k+1}$  ▷  $p$  é o ponto da superfície

7:  $n \leftarrow \frac{\nabla f(p)}{\|\nabla f(p)\|}$  ▷ O vetor normal à superfície no ponto  $p$

8: **se**  $n_x > 0,5$  **ou**  $n_y > 0,5$  **então** ▷ Determinando os vetores tangentes  $t_1$  e  $t_2$

- 9:  $t_1 \leftarrow \frac{(n_y, -n_x, 0)}{\|\dots\|}$   
 10: **senão**  
 11:  $t_1 \leftarrow \frac{(n_z, 0, n_x)}{\|\dots\|}$   
 12: **fim se**  
 13:  $t_2 \leftarrow n \times t_1$   
 14: **fim procedimento**

A determinação do vetor  $t_1$  segue critérios empíricos e o vetor  $t_2$  é calculado mediante o produto vetorial entre os vetores  $n$  e  $t_1$ . Em nossa descrição denotaremos os pontos amostrados da superfície, pertencentes a malha por  $p_i$  e os seus vetores associados (base ortonormal) por  $n, t_{i1}$  e  $t_{i2}$ .

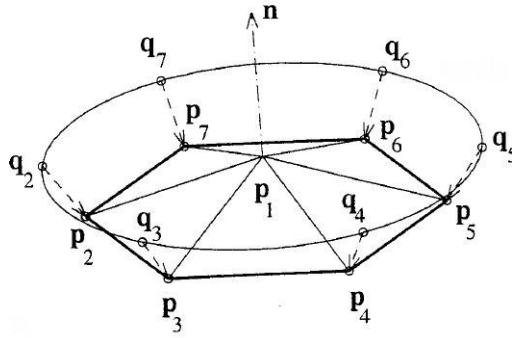


Figura 4.2: Criação da primeira frente [31].

Com o ponto  $p_1$  e seus vetores  $t_{11}$  e  $t_{12}$ , obtemos a expressão abaixo que determina os 6 pontos  $q_i$ ,  $i = 2, \dots, 7$ , calculadas pela equação sobre uma circunferência centrada em  $p_1$  e raio  $\delta_t$ , pertencente ao plano tangente a  $S$  em  $p_1$ . (Figura 4.2)

$$q_i = p_1 + \delta_t \cos \frac{i - 2\pi}{3} t_{11} + \delta_t \sin \frac{i - 2\pi}{3} t_{12}$$

A fórmula apresentada é uma simplificação da matriz de mudança de base, ou seja, para a determinação dos pontos estamos considerando um referencial local com origem em  $p_1$ , e a escolha do fator  $\frac{i - 2\pi}{3}$  visa criar triângulos aproximadamente equiláteros.

Para cada  $q_i$  aplicamos o procedimento *Surfacepoint* e encontramos seus respectivos pontos  $p_i$ ,  $i = 2, \dots, 7$ , sobre a superfície e o referencial associado. Tais pontos constituem a primeira frente poligonal e, nesse instante é denominada frente atual. Os



pontos  $p_i$ ,  $i = 1, \dots, 7$ , constituem os vértices da malha malha inicial que consiste nos 6 triângulos gerados pelos pontos:

$$(p_1, p_2, p_3), (p_1, p_3, p_4), (p_1, p_4, p_5), (p_1, p_5, p_6), (p_1, p_6, p_7), (p_1, p_7, p_2).$$

É importante ressaltar que a ordem de inserção dos vértices do triângulo infere uma orientação para cada polígono da malha.

Podemos ainda perceber que o raio da circunferência  $\delta_t$  está diretamente relacionado com o comprimento das arestas dos triângulos criados. A obtenção de uma malha triangular regular, isto é, uma malha cujo comprimento das arestas sejam semelhantes, dependerá do comprimento do raio  $\delta_t$  da circunferência centrada em  $p_i$ , o controle do comprimento poder ser realizado através do raio de curvatura da superfície nesse ponto. Na nossa implementação, para determinar  $\delta_t$ , usamos a curvatura máxima, dada por  $k = \max\{|k_1|, |k_2|\}$ , onde  $k_1$  e  $k_2$  são as curvaturas principais de  $S$  em  $p_i$  e  $\delta_t = \frac{1}{k}$ .

Para estimar os valores da curvatura máxima, usamos as idéias desenvolvidas na Seção 2.2 com as seguintes considerações:

A transformação:

$$\begin{aligned} L(\alpha) &= d\tilde{N}_p(v) \\ &= \frac{d}{dt}(\alpha(p + tv)\nabla f(p + tv))|_{t=t_0} \\ &= \frac{d}{dt}(\alpha(p + tv))\nabla f(p + tv)|_{t=t_0} + (\alpha(p + tv))\frac{d}{dt}\nabla f(p + tv)|_{t=t_0}. \end{aligned}$$

é tomada como

$$L(v) = \alpha(p)Hf(p).v,$$

onde

$Hf(p) = D(\nabla f(p))$  é a matriz Hessiana de  $f$  em  $p$ , ou seja, estamos considerando que  $d(\alpha(p + tv))\nabla f(p + tv)|_{t=0} = 0$  para todos zero, e neste caso, a matriz associada a transformação  $L$  tem a forma

$$l = \frac{1}{\|\nabla f(p)\|} \begin{bmatrix} b_1^t L(b_1) b_1 & b_1^t L(b_2) b_2 \\ b_2^t L(b_1) b_1 & b_2^t L(b_2) b_2 \end{bmatrix}.$$

considerando a base  $\{c_1, c_2\}$ ,  $\{c_2, c_2\}$  do plano tangente obtemos, respectivamente, as seguintes expressões:

$$\|c_1\|^2\|c_2\|^2K = f_z^2 [f_x^2A + f_y^2B + f_z^2C - 2f_xf_yD - 2f_yf_zE - 2f_xf_zF] \cdot \frac{1}{\|\nabla f(p)\|^2}$$

$$\|c_2\|^2\|c_3\|^2K = f_x^2 [f_x^2A + f_y^2B + f_z^2C - 2f_xf_yD - 2f_yf_zE - 2f_xf_zF] \cdot \frac{1}{\|\nabla f(p)\|^2}$$

$$\|c_1\|^2\|c_3\|^2K = f_y^2 [f_x^2A + f_y^2B + f_z^2C - 2f_xf_yD - 2f_yf_zE - 2f_xf_zF] \cdot \frac{1}{\|\nabla f(p)\|^2}$$

onde

$$A = \begin{vmatrix} f_{yy} & f_{yz} \\ f_{yz} & f_{zz} \end{vmatrix}, B = \begin{vmatrix} f_{zz} & f_{zx} \\ f_{zx} & f_{xx} \end{vmatrix}, C = \begin{vmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{vmatrix}$$

$$D = \begin{vmatrix} f_{xy} & f_{yz} \\ f_{xz} & f_{zz} \end{vmatrix}, E = \begin{vmatrix} f_{yz} & f_{zx} \\ f_{yx} & f_{xx} \end{vmatrix} \text{ e } F = \begin{vmatrix} f_{xz} & f_{yx} \\ f_{zy} & f_{yy} \end{vmatrix}$$

Somando as três expressões encontradas:

$$K((f_x^2 + f_y^2 + f_z^2)^2 + f_y^2f_z^2 + f_y^2f_x^2 + f_x^2f_z^2) = (f_x^2 + f_y^2 + f_z^2)Q \cdot \frac{1}{\|\nabla f(p)\|^2}$$

Como  $\|c_1\|^2 = f_y^2 + f_z^2$ ,  $\|c_2\|^2 = f_x^2 + f_z^2$  e  $\|c_3\|^2 = f_x^2 + f_y^2$ . Encontramos:

$$K = \frac{Q}{\|\nabla f\|^2 + M}, \text{ onde}$$

$$M = f_y^2f_z^2 + f_y^2f_x^2 + f_x^2f_z^2.$$

que é um valor associado a curvatura gaussiana.

O cálculo da metade do traço da matriz  $l$  resulta na expressão que determina um valor relacionado com a curvatura média.

$$H = \frac{1}{2\|\nabla f\|} \left( \frac{1}{\|c_1\|^2} (f_z^2f_{yy} + f_y^2f_{zz}) + \frac{1}{\|c_2\|^2} (f_z^2f_{xx} + f_x^2f_{zz}) \right)$$

e usando as equações 2.2 obtemos os valores relacionados com as curvaturas principais  $k_1$  e  $k_2$ .

## 4.2 Cálculo dos Ângulos Frontais

Para facilitar a identificação dos pontos com as suas respectivas frentes, usaremos a notação  $p_{mi}$ , onde  $m$  representa a frente a qual pertence o ponto e  $i$  o índice do ponto propriamente dito.

Para cada novo ponto  $p_{0i}$  da frente atual, denotada por  $\prod_0$ , é necessário calcular o ângulo frontal  $w$ , bem como de seus pontos adjacentes.

Seguindo a Figura 4.3 vamos definir os pontos vizinhos a  $p_{0i}$ , denotado por  $v_1$  e  $v_2$ .

Para  $v_1$ :

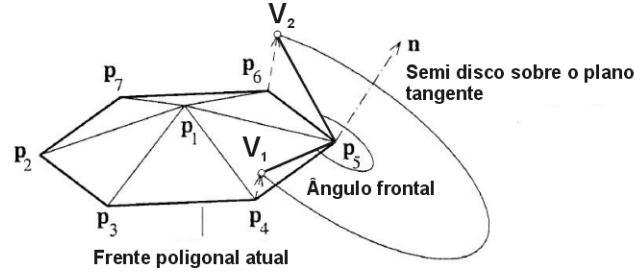


Figura 4.3: Determinação do ângulo frontal [31].

- Se  $i > 1$  então  $v_1 = p_{0,i-1}$
- Se  $i = 1$  então  $v_1 = p_{0,N_0}$ , onde  $N_0$  é o número de pontos da frente atual.

Para  $v_2$ :

- Se  $i < N_0$  então  $v_2 = p_{0,i+1}$
- Se  $i = N_0$  então  $v_2 = p_{01}$

Reescrevendo as coordenadas dos pontos  $v_1 = (x_1, y_1, y_1)$  e  $v_2 = (x_2, y_2, y_2)$  com relação ao referencial  $\{p_{0i}, (n, t_1, t_2)\}$  obtendo as novas coordenadas  $v_1 = (\xi_1, \eta_1, \zeta_1)$  e  $v_2 = (\xi_2, \eta_2, \zeta_2)$ . A partir destas, calculamos os ângulos polares  $w_1$  e  $w_2$  de  $v_1 = (\xi_1, \eta_1)$  e  $v_2 = (\xi_2, \eta_2)$ , respectivamente. Se  $w_2 > w_1$ , então o ângulo frontal será  $w = w_2 - w_1$ , caso contrário  $w = w_2 - w_1 + 2\pi$ .

### 4.3 Verificação de intra e extra colisões

Para evitar a sobreposição de triângulos, caso os bordos de uma frente (ou frentes distintas) se encontrem, (Figuras 4.4 e 4.5) é necessário uma verificação das seguintes distâncias:

- Entre pares de pontos de uma mesma frente poligonal que não sejam vizinhos, ou vizinhos de vizinhos.

Dados dois pontos  $p_{0i}, p_{0j}, i < j$ , que satisfaçam a condição mencionada. Se a distância euclidiana entre eles for menor do que  $\delta_t$  ( $\|p_{0i} - p_{0j}\| < \delta_t$ ), então a frente poligonal atual,  $\Pi_0$ , será dividida em duas: a primeira frente, que continua

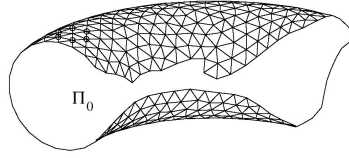


Figura 4.4: Colisão entre pontos da mesma frente [31].

sendo considerada a frente atual,  $(p_{01}, \dots, p_{0i}, p_{0j}, \dots, p_{0N_0},)$  com  $N_0 - (j - i - 1)$  pontos e uma segunda frente  $(p_{0i}, \dots, p_{0j})$  com  $j - i + 1$  pontos.

- (ii) Entre pontos da frente atual,  $\Pi_0$  com todos os pontos de todas as outras frentes poligonais  $\Pi_k, k > 0,.$

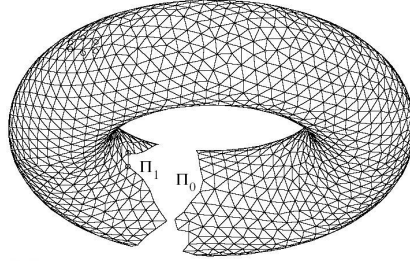


Figura 4.5: Colisão entre pontos de frentes distintas [31].

Se  $p_{0i} \in \Pi_0$  e  $p_{mj} \in \Pi_m$  estão próximos, isto é,  $\|p_{0i} - p_{mj}\| < \delta_t$ , então as frentes

$$\Pi_0 = (p_{01}, \dots, p_{0N_0}) \text{ e } \Pi_m = (p_{m1}, \dots, p_{mN_m})$$

são unidas numa única frente atual com  $N_0 + N_m$  pontos.

## 4.4 Construindo a Expansão da Frente

Seja  $p_{0m}$  um ponto da frente atual com ângulo mínimo  $w$ . A triangulação em torno de  $p_{0m}$  é obtida através dos seguintes passos: Primeiro, determinando seus pontos vizinhos  $v_1$  e  $v_2$ ; Segundo, calculando o número de triângulos a serem criados usando a seguinte fórmula:

$$n_t = \text{trunc} \left( \frac{\pi}{3w} \right);$$

Terceiro, calculamos o valor, em radianos, do ângulo  $\Delta w = w/n_t$  formado pelos vetores  $v_1$  e  $v_2$ . A depender da distância do ponto  $p_{0m}$  aos seus pontos vizinhos,  $v_1$  e  $v_2$ ,

será necessária realizar algumas alterações (Figura 4.6) a fim de obter triângulos mais adequados. A seguir serão listados alguns casos extremos onde o triângulo gerado será alterado:

- Se  $(\Delta w < 0.8$  e  $n_t > 1)$ , então  $n_t = n_t - 1$  e  $\Delta w = \frac{w}{n_t}$  (Caso a).
- Se  $n_t = 1$  e  $\Delta w > 0.8$  e  $\|v_1 - v_2\| > 1.2\delta_t$ , então  $n_t = 2$  e  $\Delta w = \frac{w}{2}$  (Caso b).
- $w < 3$  e  $\|v_1 - p_{0m}\| \leq 0.5\delta_t$ , então  $n_t = 1$  (Caso c).

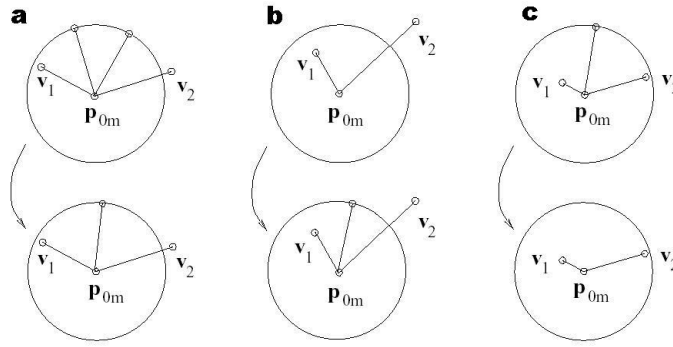


Figura 4.6: Casos especiais na geração dos triângulos [31].

Se  $n_t = 1$ , então temos apenas o triângulo  $(v_1, v_2, p_{0m})$ . Caso contrário temos que executar um procedimento semelhante ao descrito em 4.1 com a diferença de se tratar de um semi disco sobre o plano tangente a superfície no ponto  $p_{0m}$ .

Os pontos  $q_0$  e  $q_n$  são obtidos através das projeções ortogonais de  $v_1$  e  $v_2$  sobre plano tangente a  $S$  em  $p_{0m}$ . Os demais pontos  $q_i, i = 1, \dots, n_t - 1$ , resultam da seguinte rotação

$$q_i = p_{0m} + \delta_t [\cos i\Delta w + \text{sen } i\Delta w],$$

onde um ângulo de  $i\Delta w$  é um ângulo em torno do eixo  $n$ , que é normal a  $S$  em  $p_{0m}$ . Se existir um envoltório limitante (*bounding box*), o segmento  $\overline{p_{0m}q_i}$  é truncado. O procedimento *Surfacepoint* é usado em cada  $q_i$  e, assim, obtemos os novos pontos da superfície  $S$ :  $p_{N+i}, i = 1, \dots, n_t - 1$ , onde  $N$  é o número de pontos já existentes. Os novos triângulos são:

$$(v_1, p_{N+1}, p_{0m}), (p_{N+1}, p_{N+2}, p_{0m}), \dots, (p_{N+n_t-1}, v_2, p_{0m}).$$

# Capítulo 5

## Implementação

### 5.1 Considerações Gerais

O método escolhido para a implementação foi o *Marching Triangles* (Capítulo 4) melhorado com o uso da curvatura [30] para a determinação do comprimento das arestas.

A implementação foi feita utilizando a linguagem de programação C padrão ANSI e foram usados os programas de computador: ambiente integrado de programação DEVCC++ versão 4.9.9.2, para ambiente Windows 2000 e o compilador gcc do Linux. As bibliotecas gráficas usadas foram o glu e o glut para ambiente Linux, ambos pacotes do Mesa ([www.mesa.org](http://www.mesa.org)).

As estruturas de dados básicas necessárias para a criação do programa, foram listas duplamente ligadas alocadas dinamicamente.

#### 5.1.1 O Algoritmo

Implementação foi feita em duas etapas: gera uma lista de pontos da superfície, posteriormente usamos os comandos do mesa para que a superfície seja desenhada na tela. Esta separação das etapas gerou um código fonte que não usa, de forma maciça, os recursos oferecidos pela biblioteca mesa.

Abaixo, listaremos o algoritmo implementado, que foi sugerido por Araújo [30]:

**Algorithm 5.1** *Poligonizador*

```

1: procedimento MAIN
2:    $pt0 \leftarrow \text{CalculaPontoInicial}$ 
3:    $f0 \leftarrow \text{CriaFrenteVazia}$ 
4:    $\text{Triangula\c{c}\~{a}oInicial}(pt0, f0, malha)$ 
5:   enquanto  $\text{PilhaNaoVazia}(S)$  fa\c{c}a
6:      $f0 \leftarrow \text{PopFrente}(S);$ 
7:     enquanto  $\text{TamFrente}(f0) > 3$  fa\c{c}a
8:        $\text{AtualizaAngulo}(f0)$ 
9:        $pt \leftarrow \text{RetornaPontoAnguloMinimo}(f0)$ 
10:       $ptI \leftarrow \text{AutoIntersecao}(f0)$ 
11:      se  $ptI = \text{NULL}$  ent\~{a}o
12:         $f1 \leftarrow \text{ColisaoFrentes}(pt, f0, S)$ 
13:        se  $f1 = \text{NULL}$  ent\~{a}o
14:           $\text{ExpandirFrente}(pt, f0, malha)$ 
15:          sen\~{a}o
16:             $\text{UnirFrente}(f0, f1, S)$ 
17:          fim se
18:        sen\~{a}o
19:           $\text{SeparaFrente}(f0, pt1, S)$ 
20:        fim se
21:      fim enquanto
22:       $\text{Triangula}(f0, malha)$ 
23:       $\text{Delete}(f0)$ 
24:    fim enquanto
25:     $\text{Retorne}(malha)$ 
26: fim procedimento

```

## 5.2 Estruturas de Dados

O modelo da estrutura de dados para a gerenciar a malha gerada foi uma lista duplamente encadeada contendo os vértices de cada face poligonal gerada. As frentes são controladas por uma pilha simulada por uma lista duplamente ligada. A seguir vamos explicitar as estruturas e tipos de dados utilizados no programa para controlar e armazenar os pontos da superfície.

O tipo de dados básico que representa um ponto (ou vetor) do  $\mathbb{R}^3$  é o `Tvetor3`:

```
typedef struct Vetor3{
    double x,y,z;
}TVetor3;
```

Os pontos que constituem as frentes sobre a superfície são representados segundo a estrutura:

```
typedef struct P{
    TVetor3 coord,normal,t_1, t_2;
    int frente_atual;
    int angulo_mudou, pto_borda;
    double angulo;
} TPto_Sup;
```

Hartmann [31] sugere esta estrutura para todos os pontos sobre a superfície mas optamos por mantê-la apenas para os pontos da frente. A lista simplesmente encadeada que guarda os triângulos, e suas coordenadas, é do tipo `Tnos`, onde:

```
typedef struct nos{
    TVetor3 v1,v2,v3;
    struct nos *proximo;
} Tnos;
```

Além da inserção e remoção de nós, uma frente pode ser dividida em duas ou duas frentes podem ser unidas. Por esta razão decidimos representar a frente por uma lista duplamente encadeada circular cuja estrutura é dada por:



```

struct Frente{
    int num;
    TPto_Sup aponta;
    struct Frente *prox, *ant;
}typedef TFrente;

```

Por fim, a estrutura da pilha que armazena as frentes resultantes da separação da frente atual:

```

struct pilha{
    int tamfrente;
    TFrente *frente;
    struct pilha *prox, *ant;
}typedef TPilha;

```

As operações com vetores do  $\mathbb{R}^3$  foram inseridas seguindo Velho & Gomes [25] e são precedidas do sufixo *v3\_*.

O comprimento aproximado da aresta é dado pela variável *delta*. Inicialmente este valor é escolhido pelo usuário, mas com a inserção da curvatura, mais precisamente do raio de curvatura, o valor de *delta* é calculado para cada ponto da frente atual.

A orientação dos triângulos depende da inserção correta de cada vértice dos triângulos que compõem a malha. Adotamos o sentido horário (regra da mão direita) como a orientação da superfície.

### 5.3 Imagens

A Figura 5.1 mostra a propagação de uma frente na geração de uma malha triangular que aproxima uma esfera de equação  $x^2 + y^2 + z^2 = 2$ ; usando o algoritmo *Marching Triangles*.

Podemos observar que a malha gerada apresenta alguns efeitos indesejados, como os triângulos não serem aproximadamente equiláteros, isto se deve ao fato de nossa implementação não ter contemplado os casos especiais mostrados na seção 4.4.

O tamanho das arestas da malha triangular são determinadas a partir da curvatura em cada ponto. Em nossa implementação consideramos uma curvatura máxima permitida *dmax*, e nos pontos onde a curvatura calculada é menor o valor é ajustado

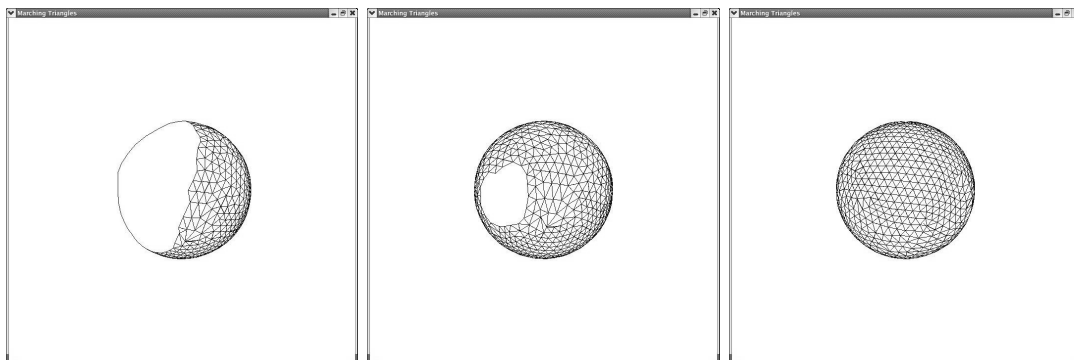


Figura 5.1: Propagação de um frente sobre a esfera

no caso da esfera de raio 2, tomamos  $dmax = 0,25$ .

A figura 5.2 mostra a geração da malha que aproxima um toro representado pela equação  $(x^2 + y^2 + z^2 + r^2 - a^2)^2 - 4r^2(x^2 + y^2) = 0$ , onde  $r = 2$  e  $a = 0.9$ . A Figura 5.3 ilustra a separação de uma frente em duas novas frentes e a Figura 5.4 ilustra a junção de duas frentes em uma nova frente.

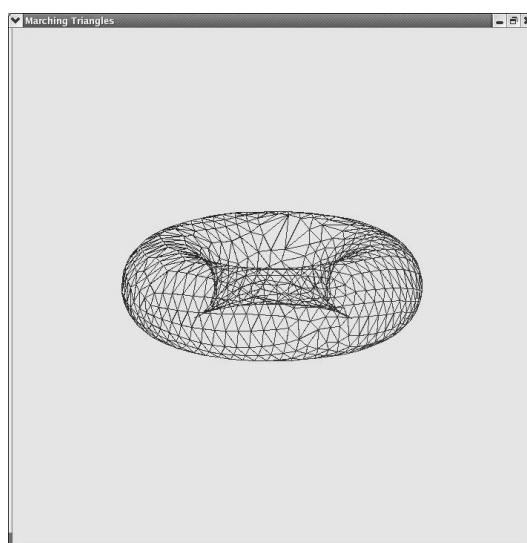


Figura 5.2: Toro

A Figura 5.3 mostra uma malha cilíndrica a partir da equação  $x^2 + y^2 = 2$ , onde o truncamento da imagem é ocasionada por uma caixa limitante.

As Figuras 5.3, 5.3 e 5.3 ilustram o comportamento da implementação do algoritmo *marching triangles* em outras superfícies, com diversas curvaturas.

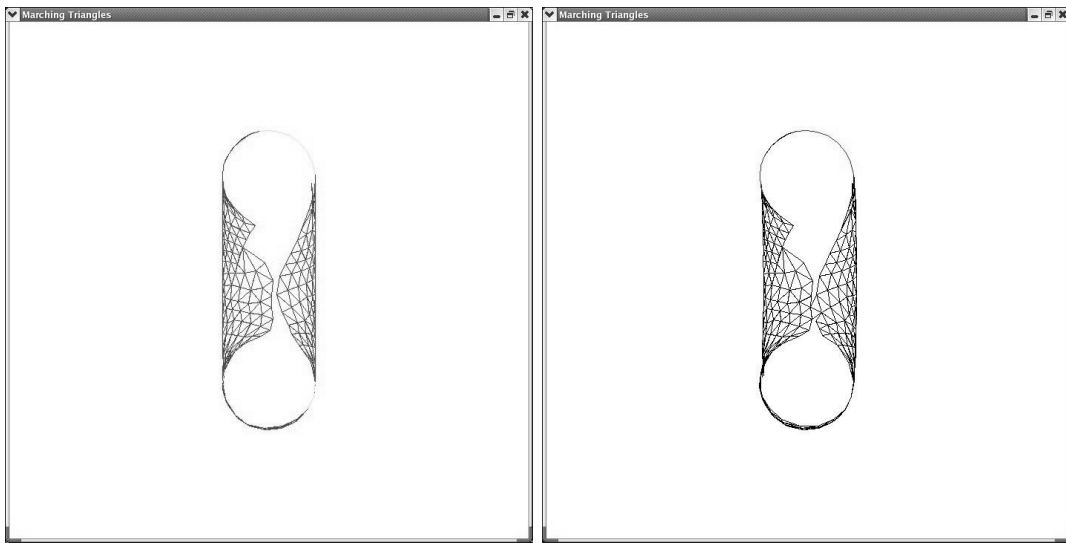


Figura 5.3: Separação de frentes

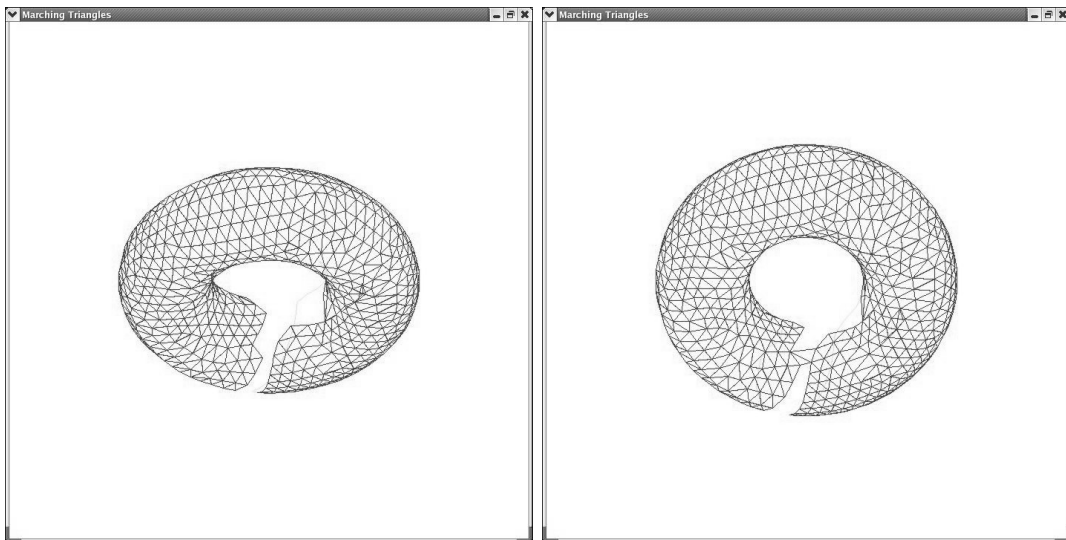


Figura 5.4: União de frentes

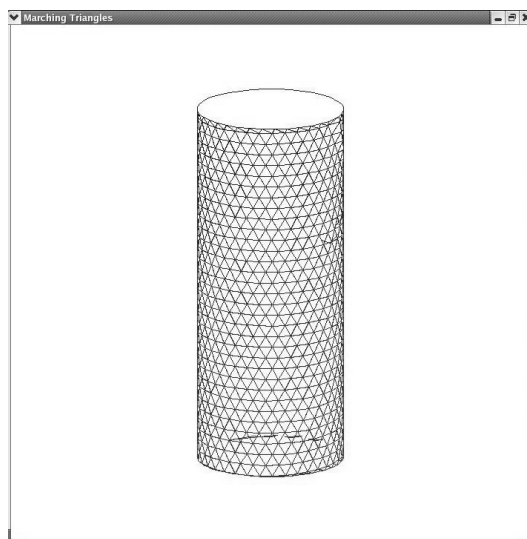


Figura 5.5: Cilindro

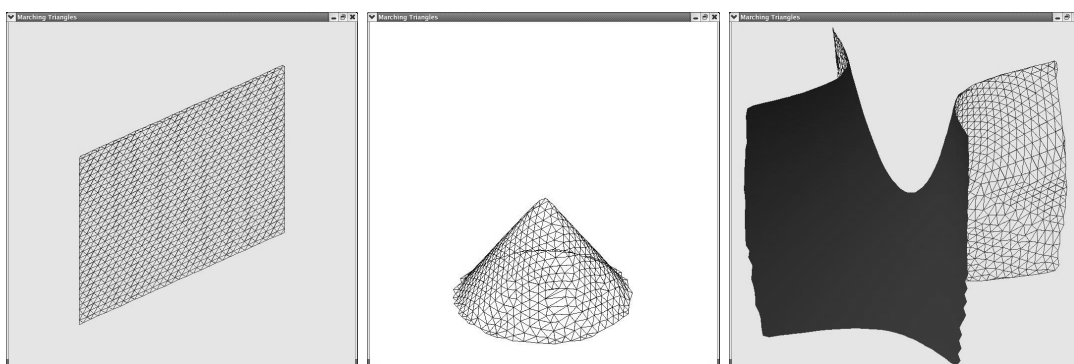


Figura 5.6: Plano

Figura 5.7: Cone

Figura 5.8: Sela

# Capítulo 6

## Conclusão

Nosso trabalho, além de apresentar um estudo dos métodos de poligonização de superfícies descritas implicitamente por uma equação matemática, apresentam, também, uma implementação do método *marching triangles* [30,31], que gera uma aproximação polinomial adaptativa, através do avanço de frentes, a partir do ângulo frontal mínimo, usando o módulo da curvatura máxima exata da superfície, em cada ponto amostrado, para controlar o comprimento das arestas da malha.

Comparativamente com os demais métodos estudados, salientamos as seguintes vantagens do método *marching triangles*: É conceitualmente simples; as etapas de amostragem e de reestruturação são realizadas concomitantemente; não requer uma estrutura de dados complexa para a sua implementação; a orientação da malha poligonal é inferida naturalmente pela frente angular, durante a sua expansão. Além disso, a abordagem do avanço de frente, a partir do ângulo mínimo, limita a quantidade de pontos da malha onde a verificação da distância será calculada, para a tomada de decisão sobre a estratégia de avanço.

Podemos dizer que uma das contribuições deste trabalho está em mostrar como as técnicas da Geometria Diferencial, na caracterização das propriedades locais de superfícies descritas implicitamente, podem ser usadas eficientemente na representação computacional de objetos geométricos, particularmente na obtenção de malhas poligonais que aproximam superfícies.

Como continuidade deste trabalho propomos melhorar a implementação inserindo os procedimentos que tratam os casos críticos na determinação dos novos triângulos da frente atual, apresentados na seção 4.4; as funcionalidades para identificar componentes não conexos de uma superfície e a determinação dos pontos adequados para a abertura de frentes em cada um desses componentes, como também, as funcionalidades que permitam a sua extensão para superfícies descritas parametricamente e um estudo quantitativo do Algoritmo *marching triangles* em relação aos métodos de poligonização mais conhecidos.

Propomos, também, um estudo de métodos que permitam capturar, além da geometria do objeto descrito implicitamente, a topologia exata desse objeto.

# Apêndice A

## Aplicações Lineares Auto - Adjuntas

Seja  $V$  um espaço vetorial de dimensão 2, munido de um produto interno  $\langle, \rangle$ , uma aplicação linear  $A : V \rightarrow V$  é auto-adjunta se  $\langle Av, w \rangle = \langle v, Aw \rangle$ , para todo  $v, w \in V$ .

Se  $\{e_1, e_2\}$  é uma base ortonormal de  $V$  e  $(\alpha_{ij}), i, j = 1, 2$ , é a matriz de  $A$  relativa a esta base, então

$$\alpha_{ij} = \langle Ae_j, e_i \rangle = \langle e_j, Ae_i \rangle = \langle Ae_i, e_j \rangle = \alpha_{ji},$$

isto é, a matriz  $(\alpha_{ij})$  é simétrica.

Existe uma bijeção entre as formas quadráticas em  $V$  e aplicações lineares auto-adjuntas de  $V$ , onde uma forma quadrática é uma função  $E$  que, em termos das coordenadas de um vetor relativo a uma base de  $E$  se exprime como um polinômio homogêneo do segundo grau. .

Um vetor  $v$ , não nulo, é um autovetor de uma aplicação linear  $A : V \rightarrow V$  se  $Av = \lambda v$ , onde  $\lambda$  é um valor real.  $\lambda$  é denominado autovalor de  $A$ .

**Proposição A.1** *Seja  $A : V \rightarrow V$  uma aplicação linear auto-adjunta. Então existe uma base ortonormal  $\{e_1, e_2\}$  de  $V$  tais que  $A(e_1) = \lambda_1 e_1, A(e_2) = \lambda_2 e_2$ . Nesta base ortonormal  $\{e_1, e_2\}$  a matriz de  $A$  é diagonal e os elementos  $\lambda_1, \lambda_2$ , da diagonal são o máximo e o mínimo, respectivamente, da forma quadrática  $Q(v) = \langle Av, v \rangle$  sobre o círculo unitário  $x^2 + y^2 = 1$  de  $V$ .*

Maiores detalhes podem ser encontrados em [21, 22].



# Apêndice B

## Superfícies Trianguláveis

A razão de trabalharmos com superfícies de faces trianguláveis, ao invés de faces poligonais com um número de arestas diferente de três, se deve ao fato de que qualquer polígono pode ser decomposto em triângulos e, também, que faces triangulares tem a vantagem de serem planas e possuem naturalmente um sistema de coordenadas locais (coordenadas baricêntricas) [18].

Nesta seção apresentaremos alguns resultados, encontrados em DoCarmo [21] que justificam a representação de superfícies regulares por meio de uma malha triangular e que, além das características topológicas, existem resultados que garantem que, nas vizinhanças de um ponto  $p$  sobre a superfície  $S$ , a triangulação também guarda a orientação da superfície que representa.

**Definição B.1** *Seja  $\alpha : [0, l] \rightarrow S$  uma aplicação contínua definida em um intervalo fechado  $[0, l]$  sobre uma superfície regular  $S$ . Dizemos que  $\alpha$  é uma curva parametrizada simples, fechada e regular por partes se:*

- (i)  $\alpha(0) = \alpha(l)$ ;
- (ii)  $t_1 \neq t_2$ , onde  $t_1, t_2 \in [0, l]$ , implica que  $\alpha(t_1) \neq \alpha(t_2)$ ;
- (iii) Existe uma partição  $0 = t_0 < t_1 < \dots < t_k < t_{k+1} = l$  de  $[0, l]$  tal que  $\alpha$  é diferenciável e regular em cada  $[t_i, t_{i+1}]$ ,  $i = 0, \dots, k$ .

A primeira condição indica que  $\alpha$  é uma curva fechada e a segunda garante que a curva não possui auto-interseções. A terceira condição implica que  $\alpha$  deixa de ter uma reta tangente apenas em um número finito de pontos  $(t_0, \dots, t_{k+1})$ .

Os pontos  $\alpha(t_i)$ ,  $i = 0, \dots, k$ , são chamados de vértices de  $\alpha$  e os traços  $\alpha([t_i, t_{i+1}])$  são chamados de arcos regulares de  $\alpha$ . O traço de  $\alpha([0, l])$  é chamado de curva fechada, simples e regular por partes.

Pela condição de regularidade dos arcos, para cada vértice  $\alpha(t_i)$  existem os limites à direita e à esquerda, dados, respectivamente por  $\lim_{t \rightarrow t_i^+} \alpha'(t) = \alpha'(t_i + \xi) \neq 0, t > t_i$  e  $\lim_{t \rightarrow t_i^-} \alpha'(t) = \alpha'(t_i - \xi) \neq 0, t < t_i$  para um  $\xi$  suficientemente pequeno.

Suponha que a superfície regular  $S$  está orientada (Figura B.2) e seja  $|\theta|, 0 < \theta_i \leq \pi$ , a menor determinação do ângulo de  $\alpha'(t_i - \xi)$  a  $\alpha'(t_i + \xi)$ . Se  $|\theta| \neq \pi$ , damos a  $\theta_i$  o sinal do determinante  $(\alpha'(t_i - \xi), \alpha'(t_i + \xi), N)$ . Isso significa que se o vértice  $\alpha(t_i)$  não é uma cúspide, então o sinal de  $\theta_i$  é dado pela orientação de  $S$ . O ângulo com sinal  $\theta_i - \pi < \theta_i < \pi$  é chamado de ângulo externo no vértice  $\alpha(t_i)$  (Figura B.1).

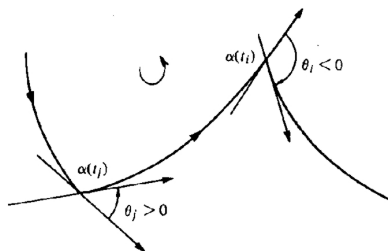


Figura B.1: Ângulos externos [21].

**Definição B.2** *Seja  $S$  uma superfície orientada. Uma região  $R \subset S$  (união de um conjunto aberto conexo com a sua fronteira) é uma região simples se  $R$  é homeomorfa a um disco fechado e a fronteira  $\partial R$  de  $R$  é o traço de uma curva parametrizada simples, fechada e regular por partes  $\alpha : I \rightarrow S$ .*

Dizemos então que  $\alpha$  é orientada positivamente se para cada  $\alpha(t)$  pertencente a um arco regular, a base positiva ortogonal  $\{\alpha'(t), h(t)\}$  satisfaz a condição de que  $h(t)$  "aponta para dentro" de  $R$ ; mais precisamente, para qualquer curva  $\beta : I \rightarrow R$  com  $\beta(0) = \alpha(t)$  e  $\beta'(0) \neq \alpha'(t)$ , temos que  $\langle \beta'(0), h(t) \rangle > 0$ .

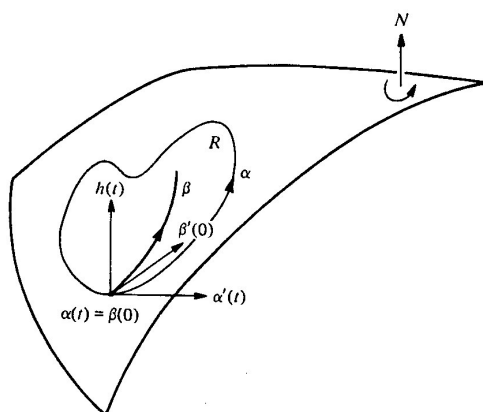


Figura B.2: Uma curva da fronteira orientada positivamente [21].

**Definição B.3** *Seja  $S$  uma superfície regular. Dizemos que uma região conexa  $R \subset S$  é regular se  $R$  é compacta e a sua fronteira  $\partial R$  é uma região finita de curvas regulares por partes fechadas e simples que não se intersectam.*

A Figura B.3 ilustra uma região regular (a) e uma região não regular (b).

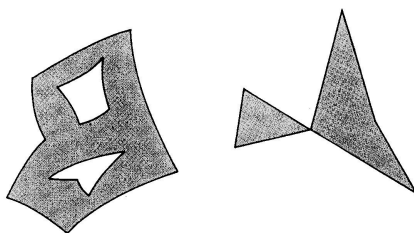


Figura B.3: regiões regular e não-regular [21].

**Definição B.4** *Uma região simples que tem apenas três ângulos externos  $\alpha_i \neq 0$ ,  $i = 1, 2, 3$  é um triângulo.*

**Definição B.5** *Uma triangulação de uma região regular  $R \subset S$  é uma família finita  $\tau$  de triângulos  $T_i, i = 1, 2, \dots, n$  tal que:*

- (i)  $\bigcup_{i=1}^n T_i = R$
- (ii) *Se  $T_i \cap T_j \neq \emptyset$ ,  $i \neq j$ , então  $T_i \cap T_j$  é uma aresta comum de  $T_i$  e  $T_j$  ou um vértice comum de  $T_i$  e  $T_j$ .*

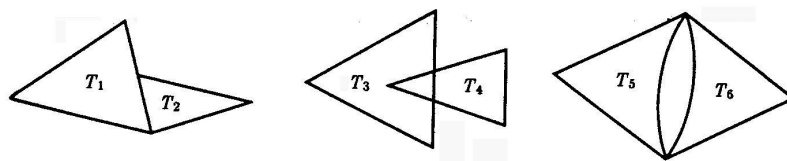


Figura B.4: Triangulações inválidas [29].

As proposições a seguir garantem a existência de uma triangulação para superfícies regulares e fornecem um método para a obtenção de uma malha triangular orientada.

**Proposição B.6** *Toda região regular de uma superfície regular admite uma triangulação.*

**Proposição B.7** *Seja  $S$  uma superfície orientada e  $\{x_\alpha\}, \alpha \in A$ , uma família de parametrizações compatíveis com a orientação de  $S$ . Então existe uma triangulação  $\tau$  de  $R$ , região regular de  $S$ , tal que todo triângulo  $T \in \tau$  está contido em alguma vizinhança coordenada da família  $\{x_\alpha\}$ . Além disso, se a fronteira de todo triângulo de  $T$  for orientada positivamente, triângulos adjacentes determinam orientações opostas na aresta comum a eles (figura B.5).*

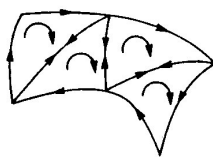


Figura B.5: Orientação de uma região triangular [21].

# Apêndice C

## Métodos Numéricos para Determinação de Raízes Reais

A seguir descrevemos alguns métodos numéricos para a cálculo de raízes reais encontrados em Barroso [24].

### C.1 Método da Bisseccção

Seja  $f$  uma função contínua no intervalo  $[a, b]$ , onde  $f(a)$  e  $f(b)$  possuem sinais contrários, ou seja,  $f(a) \cdot f(b) < 0$ .

Dividindo o intervalo  $[a, b]$  ao meio, obtém-se  $x_0$  e dois subintervalos  $[a, x_0]$  e  $[x_0, b]$ , a serem considerados.

Se  $f(x_0) = 0$ , então,  $x_0$  é um zero de  $f$ , que denotaremos por  $\varepsilon$ ; caso contrário, o zero estará no subintervalo onde a função tem sinais apostos nos pontos extremos. Neste caso, se  $f(a) \cdot f(x_0) < 0$  então  $\varepsilon \in [a, x_0]$ , senão  $\varepsilon \in [x_0, b]$  (Figura C.1).

O novo intervalo  $[a_1, b_1]$  que contém-se  $\varepsilon$  é dividido ao meio e obtém-se o ponto  $x_1$ . O processo se repete até que se obtenha uma aproximação para uma raiz exata  $\varepsilon$  com a tolerância  $\epsilon$  desejada.

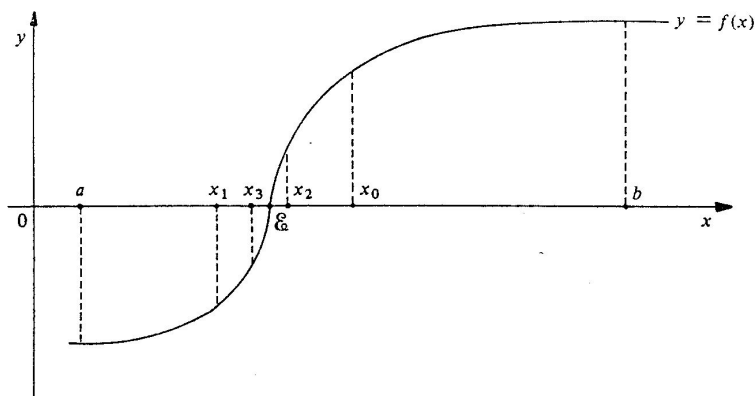


Figura C.1: Método da Bissecção [24]

## C.2 Método de Euler

O método de Euler, também conhecido como método da reta tangente, é usado para resolução de equações diferenciais lineares com problema de valor inicial.

Considere o problema do valor inicial:

$$\begin{cases} y = f(x, y) \\ y(x_0) = y_0 = \eta, \eta \text{ dado,} \end{cases} \quad (\text{C.1})$$

e queremos encontrar as aproximações  $y_1, y_2, \dots, y_m$  para as soluções exatas  $y(x_1), y(x_2), \dots, y(x_m)$  (Figura C.2).

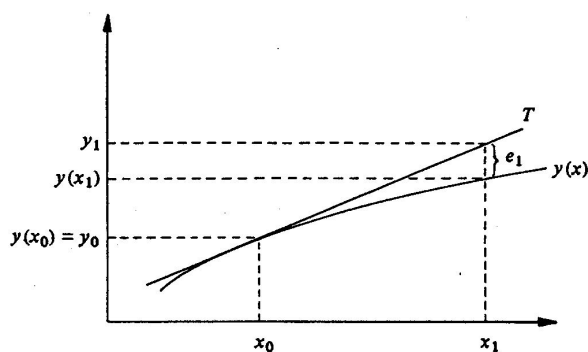


Figura C.2: Método de Euler [24].

Como o valor de  $y(x_1)$  é desconhecido, considera-se  $y_1$ , como aproximação de  $y(x_1)$

traçando uma reta tangente  $T$ , à curva  $y = y(x)$ , no ponto  $(x_0, y(x_0))$  cuja equação é:

$$y(x) - y(x_0) = (x - x_0)y'(x_0)$$

fazendo  $x = x_1$  e usando o valor inicial  $y(x_0) = y_0$ , obtemos:

$$y(x_1) - y_0 = (x_1 - x_0)y'(x_0)$$

Como  $x_1 - x_0 = h$ ,  $y' = f(x_0, y(x_0))$  e  $y_1 = y(x_1)$ , temos:

$$y_1 = y_0 + hf(x_0, y_0)$$

O erro  $e_1$  cometido na aproximação de  $y(x_1)$  por  $y_1$  é a diferença entre a solução numérica e a exata:  $e_1 = y_1 - y(x_1)$ . Para o cálculo de  $y_2$  avançamos os índices em uma unidade e o erro cometido é  $e_2 = y_2 - y(x_2)$ . Continuando até  $y_n$ , tem-se o algoritmo:

$$y_{j+1} = y_j + hf(x_j, y_j), \quad j = 0, 1, \dots, m - 1.$$

onde o erro é dado pela expressão:

$$e_{j+1} = y_{j+1} - y(x_{j+1}), \quad j = 0, 1, \dots, m - 1.$$

Sintetizando, o Método de Euler consiste em calcular recursivamente a seqüência  $\{y_j\}$  através das fórmulas:

- (i)  $y_0 = y(a) = \eta$
- (ii)  $y_{j+1} = y_j + hf(x_j, y_j), \quad j = 0, 1, \dots, m - 1.$

O método preditor-corretor, apresentado no Capítulo 2, consiste em um método de Euler modificado que avança diretamente um passo  $h$ . Nele, um ponto é calculado (preditor) através da fórmula de Euler, a partir do ponto intermediário a esse calcula-se novamente o valor da função e obtemos o ponto resultante (corretor), calculado via método de Newton, que discutiremos a seguir.

### C.3 Método de Newton

Seja  $f$  uma função contínua de classe  $C^2$  num intervalo aberto  $[a, b]$  e  $\varepsilon$  o seu único zero neste intervalo. Podemos encontrar uma aproximação  $x_n$  para  $\varepsilon$  construindo uma expansão em série de Taylor em torno de  $u = 0$  (Figura C.3).

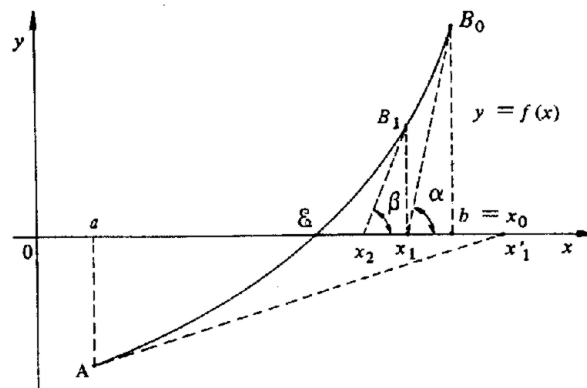


Figura C.3: Interpretação geométrica do método de Newton [24].

$$\begin{aligned}
 f(x) &= f(x_n) + f'(x_n)(x - x_n) \\
 f(x_{n+1}) &= f(x_n) + f'(x_n)(x_{n+1} - x_n) \\
 \frac{-f(x_n)}{f'(x_n)} &= x_{n+1} - x_n \\
 x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, 2, \dots
 \end{aligned}$$

onde  $x_{n+1}$  é uma aproximação para o zero de  $f$ .



# Apêndice D

## Aritmética Intervalar e Afim

### D.1 Aritmética Intervalar

A dificuldade de contornar as ambigüidade topológicas existentes nos métodos que utilizam o particionamento do plano, ou do espaço como estratégia para detecção de pontos ou contornos do objeto geométrico levou ao uso da Aritmética Intervalar (IA), também conhecida como Análise Intervalar. Esta ferramenta foi formalizada por Ramon E. Moore na década de 60 e abordada em diversos trabalhos [12, 13]. Nela, uma quantidade  $x$  é representada por um intervalo  $\hat{x} = [x_{min}, x_{max}]$  de números em ponto flutuante onde podemos efetuar "versões intervalares" das operações aritméticas (soma, subtração, etc).

$$[a, b] + [c, d] = [a + c, b + d],$$

$$[a, b] - [c, d] = [a - d, b - c],$$

$$[a, b] \times [c, d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)],$$

$$[a, b]/[c, d] = [a, b] \times [\frac{1}{d}, \frac{1}{c}], \quad 0 \notin [c, d],$$

Também são válidas as operações união e interseção:

$$A \cup B = \{x | x \in A \text{ e } x \in B\}$$

$$A \cap B = \{x | x \in A \text{ ou } x \in B\}$$

Assim, ao invés de calcularmos o valor de uma função  $f$  para cada valor  $x$  pertencente ao seu domínio podemos estimar um intervalo de valores resultante de  $f$  aplicado

a todo o seu domínio (ou parte dele). Para tanto, é necessário definirmos o que é uma função inclusão.

**Definição D.1** *Seja  $f : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$  uma função. Uma função inclusão  $F$  de  $f$  é aquela definida sobre subconjuntos  $X$  de  $\Omega$  tais que  $F(X) \supseteq f(X) = \{f(x); x \in X\}$ .*

Tal definição nos permite ter uma estimativa dos valores de  $f$  num subintervalo de  $\Omega$ . Se  $n = 2$  então temos que  $\Omega$  é uma plano e  $X$  provavelmente será um retângulo. Assim,  $F(X)$  é uma estimativa, não necessariamente exata, para os valores de  $f$  calculados sobre o conjunto  $X$ .

A função inclusão nos permite implementar um algoritmo que responderá se o zero de  $f$  está em  $X$ : Se  $0 \notin F(X)$ , então  $0 \notin f(X)$ , ou seja, o subconjunto  $X$  não possui pontos do plano por onde passa a curva gerada por  $f$ .

Um grande inconveniente da Aritmética Intervalar é que o intervalo pode ficar maior do que a estimativa exata da imagem da função  $f$ . Por exemplo, considere  $\Omega \subseteq \mathbb{R}^2$ , onde  $f(x) = x^3$  e  $x \in \bar{x} = [2, 5]$ , temos:

$$F(x - x) = F([2 - 5, 5 - 2]) = F([-3, 3]) = [-27, 27],$$

e o intervalo exato da imagem da função  $f$  é  $[0,0]$ .

Como os cálculos são realizados em cadeia, esta superestimação dos valores tende a crescer. Para obter uma estimativa de intervalos mais útil teríamos que particionar um dado intervalo em centenas de subintervalos e repetir a avaliação em cada subintervalo.

Uma forma de contornar o problema da Aritmética Intervalar, que será tratada na próxima seção, é o uso da Aritmética Afim (AA).

## D.2 Aritmética Afim

A Aritmética Afim é uma variante da Aritmética Intervalar e menos suscetível ao problema da superestimação dos intervalos [11]. Nela, uma quantidade  $\hat{x}$  é representada por sua forma afim.

$$\hat{x} = x_0 + x_1\varepsilon_1 + x_2\varepsilon_2 + \dots + x_n\varepsilon_n,$$

onde  $x_i$  são os coeficientes reais e os  $\varepsilon_i$  são variáveis simbólicas cujos valores são desconhecidos mas pertencem ao intervalo  $U = [-1, 1]$ . Cada  $\varepsilon_i$  representa uma origem de erro ou incerteza que contribui para o valor de  $\hat{x}$ . A magnitude desta contribuição é dada pelos coeficientes  $x_i$ .

Para efetuar cálculos usando a aritmética afim, devemos substituir cada operação elementar  $z = \leftarrow f(x, y)$  sobre os reais por uma equivalente  $\hat{z} = \leftarrow \hat{f}(\hat{x}, \hat{y})$ , onde.

$$\hat{x} = x_0 + x_1\varepsilon_1 + x_2\varepsilon_2 + \dots + x_n\varepsilon_n \text{ e}$$

$$\hat{y} = y_0 + y_1\varepsilon_1 + y_2\varepsilon_2 + \dots + y_n\varepsilon_n.$$

Definimos as operações:

$$\hat{x} \pm \hat{y} = (\hat{x} \pm \hat{y}) + (\hat{x} \pm \hat{y})\varepsilon_1 + (\hat{x} \pm \hat{y})\varepsilon_2 + \dots + (\hat{x} \pm \hat{y})\varepsilon_n$$

$$\alpha\hat{x} = (\alpha x_0) + (\alpha x_1)\varepsilon_1 + (\alpha x_2)\varepsilon_2 + \dots + (\alpha x_n)\varepsilon_n$$

$$\hat{x} \pm \alpha = (x_0 \pm \alpha) + x_1\varepsilon_1 + x_2\varepsilon_2 + \dots + x_n\varepsilon_n, \text{ onde } \alpha \in \mathbb{R}.$$

Quando  $f$  não é fruto de uma operação afim,  $\hat{z}$  não pode ser expresso exatamente como uma combinação afim de  $\varepsilon_i$ . Neste caso, introduzimos um termo extra  $z_k\varepsilon_k$  para representar o erro introduzido pela aproximação:

$$\hat{z} = z_0 + z_1\varepsilon_1 + z_2\varepsilon_2 + \dots + z_n\varepsilon_n + z_k\varepsilon_k$$

Assim, a multiplicação entre duas formas afins  $\hat{x}$  e  $\hat{y}$  é dada por  $\hat{z} = \hat{x}\hat{y}$ :

$$z_0 = x_0y_0$$

$$z_i = x_0y_i + y_0x_i \quad (i = 1, \dots, n)$$

$$z_k = uv$$

onde

$$u = \sum_{i=1}^n |x_i|, \quad v = \sum_{i=1}^n |y_i|.$$

Exemplo: Vamos calcular  $z = x(10 - x)$ ,  $x \in [4, 6]$ .

$$\hat{x} = 5 + 1\varepsilon_1$$

$$\hat{u} = 10 - \hat{x} = 5 - 1\varepsilon_1$$

$$\hat{z} = \hat{x}(10 - \hat{x}) = 25 - 0\varepsilon_1 - \varepsilon_2$$

O intervalo de  $z$  dada por sua forma afim  $\hat{z}$  é

$$[\hat{z}] = [25 - 1, 25 + 1] = [24, 26]$$

O cálculo da mesma expressão  $z$  em aritmética intervalar resultaria no intervalo  $[16, 36]$ .

### D.3 Conversão entre IA e AA

Se  $\hat{x} = x_0 + x_1\varepsilon_1 + x_2\varepsilon_2 + \dots + x_n\varepsilon_n$  é a forma afim de uma quantidade  $x$ , então o valor de  $x$  está no intervalo de  $\hat{x}$ , ou seja,

$$[\hat{x}] = [x_0 - \varepsilon \dots x_0 - \varepsilon], \text{ onde } \varepsilon = \sum_{i=1}^n |x_i|.$$

Supondo que cada  $\varepsilon_i$  varia num intervalo  $U = [-1, 1]$ , observe que  $[\hat{x}]$  é o menor intervalo que contém todos os possíveis valores de  $\hat{x}$ .

Reciprocamente, se temos um intervalo  $\bar{x} = [a, b]$  que representa uma quantidade  $x$  na Aritmética Intervalar então a sua forma afim para a mesma quantidade é dada pela expressão  $\hat{x} = x_0 + x_1\varepsilon_1 + \dots + x_k\varepsilon_k$ , onde  $x_0 = \frac{b+a}{2}$  e  $x_k = \frac{b-a}{2}$ .

Maiores detalhes da Análise Intervalar podem ser encontradas em [9–12].

# Bibliografia

- [1] Formella, Arno e Gill, Christian, *Ray tracing: a quantitative analysis and a new practical algorithm*, The Visual Computer, volume 11, number 9, 465-476 (1995).
- [2] Cusatis, A., de Figueiredo, L. e Gattass, M., *Interval methods for ray casting implicit surfaces with affine arithmetic*, Proceedings of SIBGRAP'99, IEEE Press, 65-71 (1999).
- [3] Bloomenthal, J., *Polygonization of Implicit Surfaces*, Computer Aided Geometric Design, 5, 341-355 (1987).
- [4] Bloomenthal, J., *Polygonization of Implicit Surfaces*, Computer Aided Geometric Design, 5, 341-355 (1988).
- [5] Bloomenthal, J. e Ning, P., *An Evaluation of Implicit Surface Tilers*, Computer Graphics and Applications, 45 (1993).
- [6] Bloomenthal, J., *Implicit Surfaces*, Encyclopedia of Computer Science and Technology, (1999).
- [7] Figueiredo, Luiz H. de, *Computational Morphology of Implicit Curves*, IMPA, Rio de Janeiro, 1992.
- [8] Figueiredo, Luiz H. de, Miranda, Jonas de, et al., *Physically-Based Methods for Polygonization of Implicit Surfaces*, Graphics Interface'92, 250-257 (1992).

- [9] Figueiredo, Luiz H. de e Stolfi, Jorge , *Self-Validated Numerical Methods and Applications*, Brazilian Mathematics Colloquium monographs, IMPA/CNPq, Rio de Janeiro, Brazil, 1997.
- [10] Figueiredo, Luiz H. de e Stolfi, J., *Adaptive Enumeration of Implicit Surfaces with Affine Arithmetic*, Computer Graphics Forum, Vol 15, num 5, 287-296 (1996).
- [11] Figueiredo, Luiz H. de e Cusatis, Affonso de, *Interval Methods fo Ray Casting Implicit Surfaces with Affine Arithmetic*, Computer Graphics Forum, Vol 15, num 5, 287-296 (1996).
- [12] Oliveira, Lopes Batista e Figueiredo, Luiz H. de, *Robust Adaptive Polygonal Approximation of Implicit Curves*, Computers and Graphics, 841-852(26), 2002.
- [13] Lopes, H., Oliveira, J. e Figueiredo, L. H. de, *Robust adaptive approximation of implicit curves*, Proceedings of SIBGRAPI.
- [14] Figueiredo, Luiz H. de e Gomes, Jonas, *Sampling implicit objects with physically-based particle systems*, Computers and Graphics, Vol 20, num 3, 365-375, (1996).
- [15] Lorensen, W.E. e Cline, H.E. , *Marching Cubes: A high resolution 3D surface construction algorithm*, ACM Computer Graphics, Vol 21, num 4, 163-169 (1987).
- [16] Natarajan, B.K. , *On Generating topologically Consistent Isosurfaces from Uniform Samples*, The Visual Computer, Vol 11, 52-62 (1994).
- [17] Gomes, Jonas Gomes e Velho, Luiz, *Computação Gráfica Volume 1*, Série Computação e Matemática, SBM/IMPA, Rio de Janeiro, 1998.
- [18] Gomes, Jonas Gomes e Velho, Luiz, *Fundamentos da Computação Gráfica*, IMPA, 2004.
- [19] Watt, Alan, *3D Computer Graphics*, Addison-Wesley, 2000.
- [20] Gomes, Jonas M. e Tavares, Geovan, *Métodos Simpliciais em Computação Gráfica*, Publicações Matemáticas - 17<sup>o</sup> Colóquio Brasileiro de Matemática, IMPA, Rio de Janeiro, 1989.

- [21] Carmo, M.P. do , *Geometria Diferencial de Curvas and Surperfícies*, Sociedade Brasileira de Matemática/SBM - Textos Univeritários, Rio de Janeiro, 2005.
- [22] Lima, Elon Lages, *Álgebra Linear (2;ª ed.)*, Projeto Euclides, IMPA, 1996.
- [23] Lima, Elon Lages *Curso de Análise Vol. 2 (5ª ed.)*, Projeto Euclides, IMPA, 1999.
- [24] Barroso, Leônidas C., et all, *Cálculo Numérico (com aplicações)*, Editora Harbra, 1987.
- [25] Velho, Luiz e Gomes, Jonas, *Sistemas Gráficos 3D*, Série Computação e Matemática, IMPA, 2001.
- [26] Diestel, R., *Graph Theory*, Springer Verlag, 2000.
- [27] Foley, J., *Computer Graphics: Principles and Practice, 2nd edition in C*, Addison-Wesley, 1990.
- [28] Hoff III, Kenneth E., *implicit.ppt*, Internet, September, 1998.
- [29] Massey, W.S., *Algebraic Topology - An Introduction*, Springer - Verlag, New York, 1967.
- [30] Araujo, Bruno Rodrigues de e Jorge, Joaquim A. P., *Curvature Dependent Polygonization of Implicit Surfaces*, Anais of SIBGRAPI 2004, October, (2004).
- [31] Hartmann, Erich, *A marching method for the triangulation of surfaces*, The Visual Computer, Vol 14, num 3, 95-108(1998),
- [32] Hughes, John F., *Differential Geometry of Implicit Surfaces in 3-Space - A Primer*, Brown University, 2003.
- [33] Carvalho, P. C., et all, *Fotografia 3D*, Publicações Matemáticas - 25<sup>o</sup> Colóquio Brasileiro de Matemática, IMPA, Rio de Janeiro, 2005.
- [34] Allgower, Eugene L. e Schmidt, Phillip H., *An algorithm for piecewise-linear approximation of an implicitly defined manifold*, SIAM Journal on Numerical Analysis, Vol 22, num 2, 322-346 (1985).