



# Universidade Federal de Alagoas

## Programa de Pós-Graduação em Matemática

### DISSERTAÇÃO DE MESTRADO

#### Aproximação Isotópica Suave de Curvas e Superfícies Implícitas

Leonardo de Oliveira Carvalho

Maceió  
dezembro de 2008

Rio São Francisco

**Leonardo de Oliveira Carvalho**

Dissertação de Mestrado

**Aproximação Isotópica Suave de Curvas  
e Superfícies Implícitas**

Dissertação apresentada ao Programa de Pós-Graduação em Matemática da Universidade Federal de Alagoas como parte dos requisitos para obtenção do título de Mestre em Matemática.

**Orientador: Prof. Dr. Adelailson Peixoto da Silva**

Maceió  
dezembro de 2008

**Catálogo na fonte**  
**Universidade Federal de Alagoas**  
**Biblioteca Central**  
**Divisão de Tratamento Técnico**  
**Bibliotecária Responsável: Helena Cristina Pimentel do Vale**

C331a Carvalho, Leonardo de Oliveira.  
Aproximação isotópica suave de curvas e superfícies implícitas / Leonardo de Oliveira Carvalho. – Maceió, 2008.  
51f. : il.

Orientador: Adailson Peixoto.  
Dissertação (mestrado em Matemática) - Universidade Federal de Alagoas.  
Instituto de Matemática. Maceió, 2008.

Bibliografia: f. 50-51.

1. Superfícies implícitas (Matemática). 2. Curvas implícitas (Matemática).  
3. Aproximação isotópica (Matemática). I. Título.

CDU: 519.65



Universidade Federal de Alagoas  
Programa de Pós-Graduação em Matemática

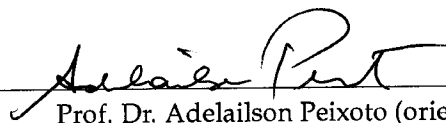
Dissertação de Mestrado

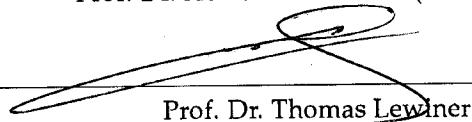
**Aproximação Isotópica Suave de Curvas  
e Superfícies Implícitas**

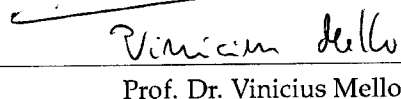
**Leonardo de Oliveira Carvalho**

Dissertação apresentada ao Programa  
de Pós-Graduação em Matemática da  
Universidade Federal de Alagoas como  
parte dos requisitos para obtenção do  
título de Mestre em Matemática.

Banca examinadora:

  
Prof. Dr. Adailson Peixoto (orientador)

  
Prof. Dr. Thomas Lewiner

  
Prof. Dr. Vinicius Mello

Maceió  
dezembro de 2008

# Agradecimentos

À minha família, meus pais Neudson e Inêz, e minhas irmãs Larissa e Luciana, por todo amor e carinho em minha vida.

Ao professor Adelailson Peixoto, pela grande contribuição na minha vida acadêmica, da iniciação científica ao mestrado.

Ao professor Vinícius Mello, por todas as contribuições e sugestões que permitiram o desenvolvimento deste trabalho.

Ao professor Thomas Lewiner pela disponibilidade em participar da banca examinadora, e pelas sugestões de melhorias para este trabalho.

Ao professor Dimas Martinez, por sua participação como suplente na banca examinadora, pela análise do trabalho e pelas sugestões dadas.

Àqueles que foram meus professores durante o mestrado: Adelailson, Vinícius, Ediel e Hilário, pois cada disciplina cursada ajudou bastante a minha formação.

Aos meus colegas do núcleo de computação gráfica da UFAL Allan, Allyson Cabral, Douglas, Fabiane, Leandro e Renata, pela amizade, apoio e descontração durante este tempo. Em especial à Fabiane pela valiosa revisão do texto.

Aos meus colegas de mestrado Alex, Arlyson, Carlos, Daniel, Darliton, Erikson, Everson, Fábio Boia, Gregório, Isadora, Isnaldo, José Borges, Leandro Favacho e Priscila, pelo companheirismo e ajuda em diversos momentos.

Ao convênio CAPES/FAPEAL, pelo auxílio financeiro.

## Resumo

Esta dissertação contém um estudo a respeito de curvas planas e superfícies. São vistas as duas formas mais usuais de se definirem estes elementos: a definição paramétrica e a implícita, com ênfase nesta última. São analisadas algumas formas de representação de curvas planas e superfícies, o que vem a ser uma tarefa relativamente simples ao se utilizar a definição paramétrica, porém com a definição implícita isto exige um maior número de operações.

São apresentados alguns métodos para encontrar aproximações de curvas e superfícies definidas implicitamente que mantenham a sua topologia e que geram objetos suaves o suficiente. Isto é feito basicamente subdividindo-se o plano (respectivamente o espaço), que é utilizado para aproximar a curva (respectivamente a superfície) de forma linear por partes, e então subdivide-se essa aproximação para que o resultado seja suave. No caso das superfícies a saída é uma malha triangular. São realizados também tratamentos para aumentar a qualidade desta malha.

## **Abstract**

This dissertation contains a study about plane curves and surfaces. The two most common way to define this elements are reviewed: the parametric and the implicit definition, with emphasis on the latter. An analysis of some methods to represent plane curves and surfaces is made. One notices that this job is relatively simple when the parametric definition is used, however with the implicit definition this requires a larger number of operations.

This works also presents some methods to find approximations of curves and surfaces implicitly defined that preserves the topology and that generate objects smooth enough. This is achieved basically by a subdivision of the plane (respectively the space), which is used to find a piecewise linear approximation of the curve (respectively the surface), then this approximation is subdivided to make the result smooth. In the case of surfaces the output is a triangular mesh. Some treatments are also made to improve the quality of the mesh.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>4</b>
1.1	Estrutura do trabalho . . . . .	5
<b>2</b>	<b>Curvas e superfícies</b>	<b>6</b>
2.1	Curvas planas . . . . .	6
2.1.1	Definição paramétrica . . . . .	6
2.1.2	Definição implícita . . . . .	7
2.1.3	Isotopia . . . . .	9
2.2	Superfícies . . . . .	10
2.2.1	Definição paramétrica . . . . .	10
2.2.2	Definição implícita . . . . .	12
2.2.3	Isotopia . . . . .	13
<b>3</b>	<b>Representação de curvas e superfícies</b>	<b>14</b>
3.1	Representação de curvas . . . . .	14
3.2	Curvas implícitas . . . . .	15
3.2.1	Rasterização . . . . .	15
3.2.2	Aproximação poligonal . . . . .	16
3.2.3	Enumeração adaptativa . . . . .	18
3.3	Representação de superfícies . . . . .	24
3.4	Representação de superfícies implícitas . . . . .	24
3.4.1	Renderização de superfícies implícitas . . . . .	24
3.4.2	Extração de superfícies implícitas . . . . .	25
<b>4</b>	<b>Aproximação suave</b>	<b>28</b>
4.1	Curvas planas . . . . .	28
4.1.1	Refinamento . . . . .	30
4.1.2	Resultados . . . . .	33
4.2	Superfícies implícitas . . . . .	38
4.2.1	Eliminação de <i>slivers</i> . . . . .	38
4.2.2	<i>Flip</i> de arestas . . . . .	40
4.2.3	Refinamento da malha . . . . .	40
4.2.4	Resultados . . . . .	41
<b>5</b>	<b>Conclusão e trabalhos futuros</b>	<b>48</b>



# Lista de Figuras

1.1	Superfícies implícitas obtidas a partir de dados de tomografias. . . . .	4
2.1	Curva paramétrica. . . . .	6
2.2	Circunferência definida parametricamente. . . . .	7
2.3	Exemplos de curvas paramétricas. . . . .	7
2.4	Exemplos de curvas implícitas. . . . .	8
2.5	Vetor gradiente. . . . .	9
2.6	Isotopia entre uma curva suave e sua aproximação. . . . .	10
2.7	Esfera definida parametricamente. . . . .	11
2.8	Exemplos de superfícies paramétricas. . . . .	11
2.9	Exemplos de Superfícies implícitas. . . . .	12
3.1	Representação de curvas paramétricas. . . . .	14
3.2	Um método para rasterização de curvas implícitas. . . . .	15
3.3	Método de enumeração. . . . .	16
3.4	Amostragem por <i>ray-casting</i> . . . . .	17
3.5	Órbitas e amostragem final. . . . .	17
3.6	Subdivisão adaptativa espacial. . . . .	19
3.7	Subdivisão adaptativa geométrica. . . . .	20
3.8	Isotopia local entre curva e aproximação. . . . .	22
3.9	Alterações topológicas após a subdivisão. . . . .	23
3.10	Aproximação da curva $x^2(1 - x^2) - y^2 + 0.01 = 0$ . . . . .	23
3.11	Representação de superfícies paramétricas. . . . .	24
3.12	Superfícies implícitas renderizadas por <i>ray-tracing</i> . . . . .	25
3.13	Casos possíveis no algoritmo <i>Marching Cubes</i> . . . . .	25
3.14	Comparação entre algoritmos de <i>Marching Cubes</i> . . . . .	26
3.15	Casos utilizados do algoritmo <i>Marching Cubes</i> . . . . .	26
3.16	Aproximação isotópica. . . . .	27
4.1	Descontinuidade na aproximação de curvas. . . . .	28
4.2	<i>Quadtree</i> e <i>DualQuadtree</i> . . . . .	29
4.3	Funções recursivas para obtenção do <i>DualQuadtree</i> . . . . .	29
4.4	Utilização do <i>DualQuadtree</i> na aproximação de curvas. . . . .	30
4.5	Subdivisão de arestas. . . . .	31
4.6	Situação onde não se forma um segmento ativo. . . . .	31
4.7	Suavização de curva pelo comprimento das arestas. . . . .	32

4.8	Ângulo entre arestas. . . . .	32
4.9	Suavização pelo ângulo entre as arestas. . . . .	32
4.10	$x^2 + y^2 - 1 = 0$ . . . . .	34
4.11	$(y - x^2 + 1)^4 + (x^2 + y^2)^4 - 1 = 0$ . . . . .	35
4.12	$y^2 - x^3 + x - 0.25 = 0$ . . . . .	36
4.13	$x^2 - xy + y^4 + 0.0001 = 0$ . . . . .	37
4.14	Irregularidades na malha encontrada. . . . .	38
4.15	Eliminação de <i>slivers</i> . . . . .	39
4.16	Eliminação de <i>slivers</i> com a maior tolerância possível. . . . .	39
4.17	Operação de <i>flip</i> de aresta. . . . .	40
4.18	Aplicação do algoritmo de Delaunay. . . . .	40
4.19	Método de subdivisão. . . . .	41
4.20	Subdivisão aplicada à aproximação da esfera. . . . .	41
4.21	$(x^2 + y^2 + z^2 + R^2 - r^2)^2 - 4R^2(x^2 + y^2) = 0$ . . . . .	42
4.22	$x^2 + y^2 + z^2 - 1 = 0$ . . . . .	43
4.23	$(x^2 + y^2 + z^2 - 23.75)^2 - 0.8((z - 5)^2 - 2x^2)((z + 5)^2 - 2y^2) = 0$ . . . . .	44
4.24	$0.4(\sin(5x) + \sin(5y) + \sin(5z)) + 0.1x^2 + 0.3y^2 + 0.2z^2 - 0.5 = 0$ . . . . .	45
4.25	Histograma 1. . . . .	46
4.26	Histograma 2. . . . .	46
4.27	Histograma 3. . . . .	47
4.28	Histograma 4. . . . .	47

## Introdução

**C**URVAS PLANAS E SUPERFÍCIES são elementos importantes, tanto do ponto de vista puramente matemático quanto do ponto de vista de aplicações. Em computação gráfica, por exemplo, estes elementos podem ser utilizados em modelagem, descrevendo a forma de objetos, e em animação, descrevendo trajetórias.

Há duas formas comuns de se descrever uma curva ou superfície: a descrição paramétrica, onde a partir de um parâmetro é possível encontrar um ponto sobre a curva ou superfície, este parâmetro pode ser um valor real, no caso das curvas, ou uma dupla de valores reais, no caso das superfícies; já a descrição implícita define curvas e superfícies como conjuntos de níveis de funções reais definidas no plano e no espaço, respectivamente.

A descrição paramétrica possui a vantagem de facilidade de se extrair uma aproximação poligonal, porém é difícil saber se algum ponto está sobre a curva ou superfície. Com a descrição implícita a situação se inverte, é difícil extrair uma aproximação poligonal, porém é bastante simples saber quando um ponto está sobre a curva ou superfície, e isto se torna bastante útil em aplicações onde é preciso realizar operações sobre conjuntos, como união, interseção e diferença, que podem ser utilizadas, por exemplo, em ambientes de modelagem CSG. A forma implícita ainda oferece a vantagem de descrever objetos de topologia arbitrária, o que não acontece com a forma paramétrica.

Em algumas aplicações, curvas ou superfícies descritas implicitamente surgem naturalmente. A partir de dados produzidos em tomografias e scanners em aplicações médicas, por exemplo, podem-se extrair superfícies implícitas representando, por exemplo, a parte externa de um osso. A Figura 1.1 ilustra alguns exemplos de superfícies obtidas desta maneira pelo método descrito em [9].

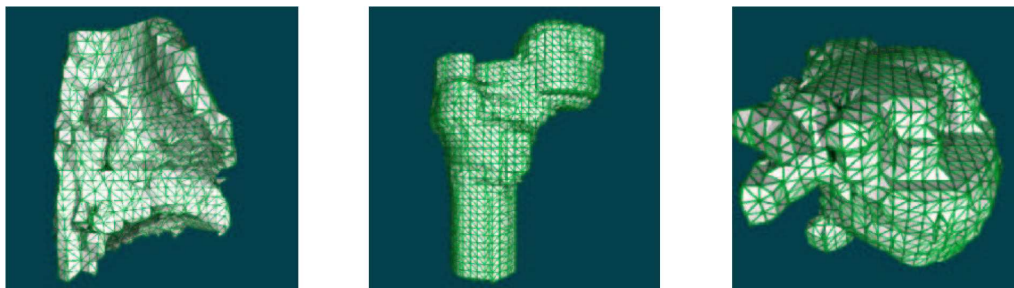


Figura 1.1: Superfícies implícitas (poligonizadas) obtidas a partir de dados de tomografias, da esquerda para a direita: um pulmão, um fêmur e um coração.

Esta dissertação apresenta um estudo de definições e formas de representação de curvas planas e superfícies, dando um enfoque maior para a definição implícita. São apresentados também alguns métodos para obter aproximações destes elementos mantendo suas topologias e com um determinado nível de suavidade. Estes métodos foram formados a partir de outros trabalhos, com algumas alterações de forma a obter o resultado desejado.

## 1.1 Estrutura do trabalho

Este trabalho está dividido da seguinte forma:

**Capítulo 2:** São vistas definições no contexto de curvas planas e superfícies.

**Capítulo 3:** São vistas diversas formas de representação de curvas planas e superfícies.

**Capítulo 4:** São apresentados alguns métodos para a obtenção de aproximações isotópicas suaves de curvas e superfícies definidas implicitamente.

**Capítulo 5:** Conclusão e trabalhos futuros.

# Curvas e superfícies

NESTE CAPÍTULO são vistas as principais definições para curvas e superfícies, além de alguns conceitos relacionados que serão úteis nos próximos capítulos.

## 2.1 Curvas planas

Curvas planas podem ser entendidas intuitivamente como figuras desenhadas a lápis em uma folha de papel. Nesta seção são vistas definições que formalizam este conceito.

### 2.1.1 Definição paramétrica

Nesta definição (tirada de [1]) uma curva plana é vista como uma aplicação contínua no plano que mapeia um intervalo na reta real em um conjunto de pontos no plano.

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

**Definição 2** Seja  $\alpha$  uma curva contínua no plano  $\mathbb{R}^2$ , o traço de  $\alpha$  é o conjunto  $C$  dado por:

$$C = \{\alpha(t) = (x(t), y(t)), t \in I\}.$$

Diz-se que  $\alpha$  é uma parametrização de  $C$ , e  $t$  é o parâmetro de  $\alpha$ . A Figura 2.1 ilustra geometricamente a definição.

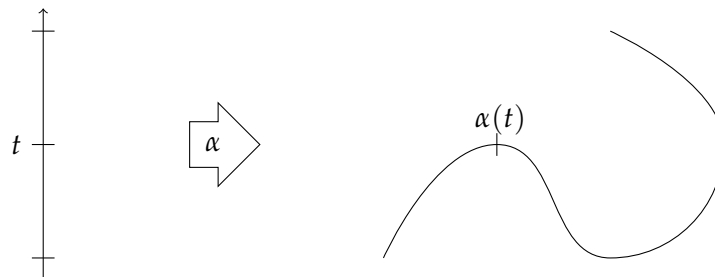


Figura 2.1: Curva paramétrica.

Como exemplo: seja  $C$  a circunferência de raio  $r$  centrada na origem, ou seja,  $C$  é o conjunto de pontos no plano cuja distância à origem  $(0,0)$  é igual a  $r$ .  $C$  pode ser visto como o traço da curva  $\alpha$  definida por  $\alpha(t) = (r \cos(t), r \sin(t))$ . A Figura 2.2 ilustra geometricamente o significado do parâmetro  $t$ , que é o ângulo que o vetor  $\alpha(t)$  forma com o eixo- $x$ .

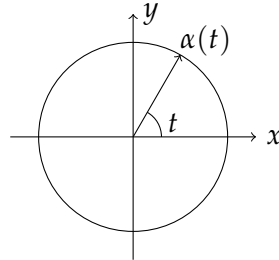
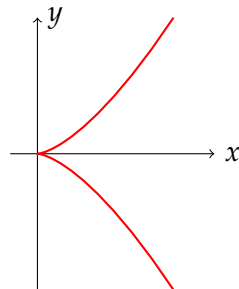
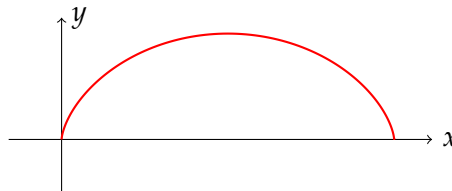


Figura 2.2: Circunferência definida parametricamente.

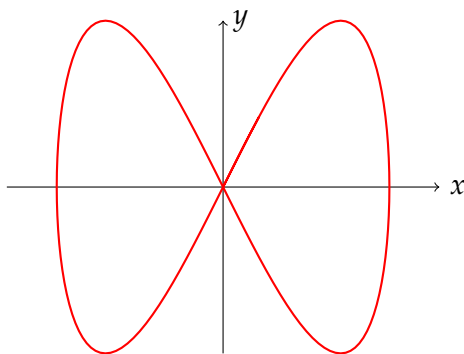
A Figura 2.3 ilustra alguns outros exemplos de curvas paramétricas.



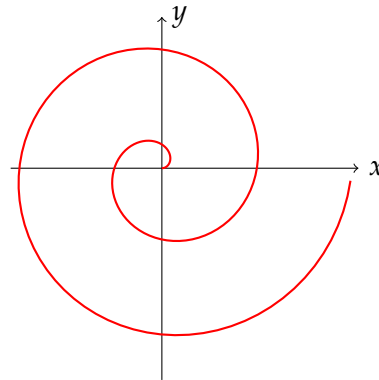
(a)  $\alpha(t) = (t^2, t^3)$



(b)  $\alpha(t) = (t - \sin(t), 1 - \cos(t))$



(c)  $\alpha(t) = (\sin(t), \sin(2t))$



(d)  $\alpha(t) = (t \cos(t), t \sin(t))$

Figura 2.3: Exemplos de curvas paramétricas.

### 2.1.2 Definição implícita

Neste tipo de descrição, as curvas planas são vistas com o conjunto de zeros de uma função real definida no plano.

**Definição 3** Dada uma função  $F : \mathbb{R}^2 \rightarrow \mathbb{R}$  ao menos  $C^0$  contínua, uma curva implícita é o

conjunto:

$$S = \{p = (x, y) \in \mathbb{R}^2; F(p) = 0\}$$

Pode-se também utilizar a notação  $S = F^{-1}(0)$  para representar curvas implícitas. A Figura 2.4 ilustra alguns exemplos de curvas implícitas.

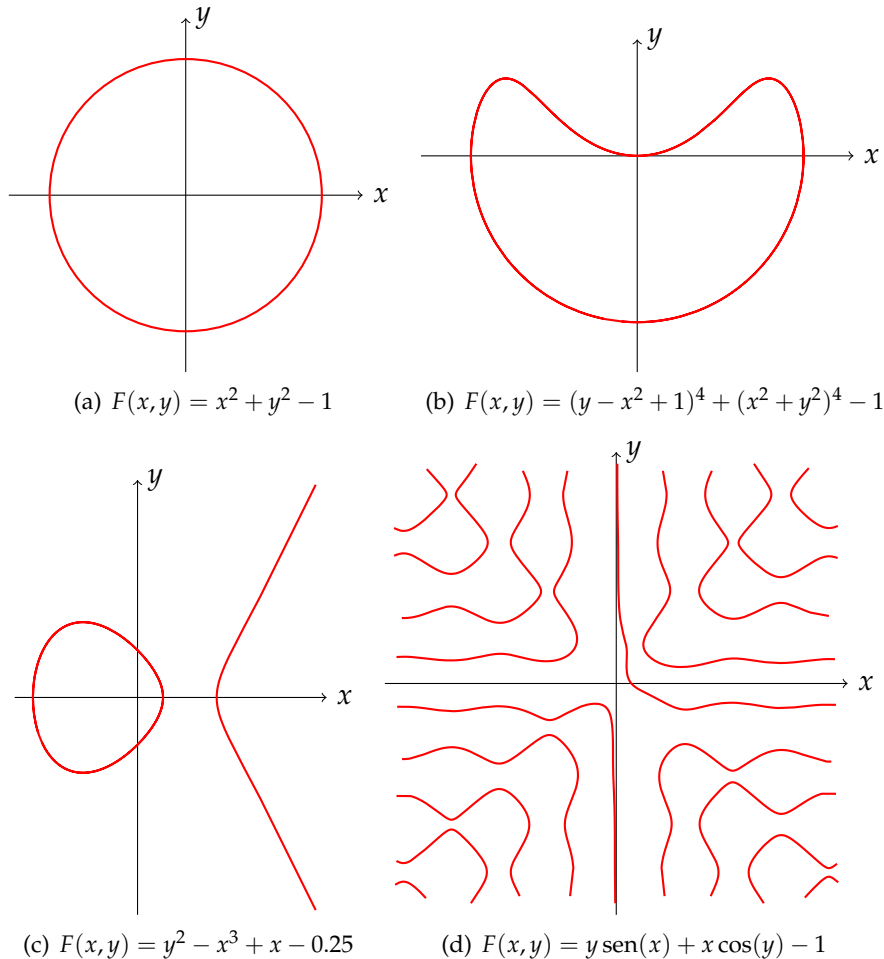


Figura 2.4: Exemplos de curvas implícitas  $S = F^{-1}(0)$  (os gráficos estão em diferentes escalas).

Os pontos  $p \in \mathbb{R}^2$  que não pertencem à curva  $S$  podem ser classificados de acordo com o sinal de  $F(p)$ :

**Definição 4** O ponto  $p = (x, y) \in \mathbb{R}^2$  que satisfaz  $F(p) > 0$  é chamado ponto exterior à curva  $S = F^{-1}(0)$ .

**Definição 5** O ponto  $p = (x, y) \in \mathbb{R}^2$  que satisfaz  $F(p) < 0$  é chamado ponto interior à curva  $S = F^{-1}(0)$ .

Pode-se perceber que nem toda curva descrita implicitamente pode ser descrita parametricamente. De fato curvas implícitas podem ser compostas por mais de uma componente conexa (como na Figura 2.4(c)-(d)), o que não ocorre em curvas paramétricas. Neste trabalho, a definição utilizada predominantemente será a definição implícita. A seguir serão vistos outros conceitos importantes relacionados a curvas implícitas.

Uma definição importante para este trabalho é a noção de gradiente de uma função:

**Definição 6** Seja  $F : \mathbb{R}^2 \rightarrow \mathbb{R}$  uma função ao menos  $C^1$  contínua, o vetor gradiente de  $F$  no ponto  $(x, y)$  é dado por:

$$\nabla F(x, y) = \left( \frac{\partial}{\partial x} F(x, y), \frac{\partial}{\partial y} F(x, y) \right).$$

$\nabla F(x, y)$  indica a direção de maior crescimento da função  $F$  no ponto  $(x, y)$ . A Figura 2.5(a) ilustra o gradiente da função  $F(x, y) = x(y + 1)$ , dado por  $\nabla F(x, y) = (y + 1, x)$ , calculado em pontos amostrados nos intervalos  $x \in [-0.8, 0.4]$  e  $y \in [-0.9, 0.2]$  (aplicou-se uma escala de 0.1 aos vetores gradiente para facilitar a visualização). Uma característica interessante do gradiente é sua relação com as curvas implícitas dadas por  $F^{-1}(0)$  (mais geralmente com qualquer curva de nível  $F^{-1}(c)$ ,  $c \in \mathbb{R}$ ), onde pode-se verificar que se  $p \in S = F^{-1}(0)$ , então  $\nabla F(p)$  é perpendicular a  $S$ . A Figura 2.5(b) ilustra essa propriedade para a função  $F(x, y) = y^2 - x^3 + x - 0.25$  (foi aplicada uma escala de forma que o comprimento de cada vetor gradiente seja igual a 0.25).

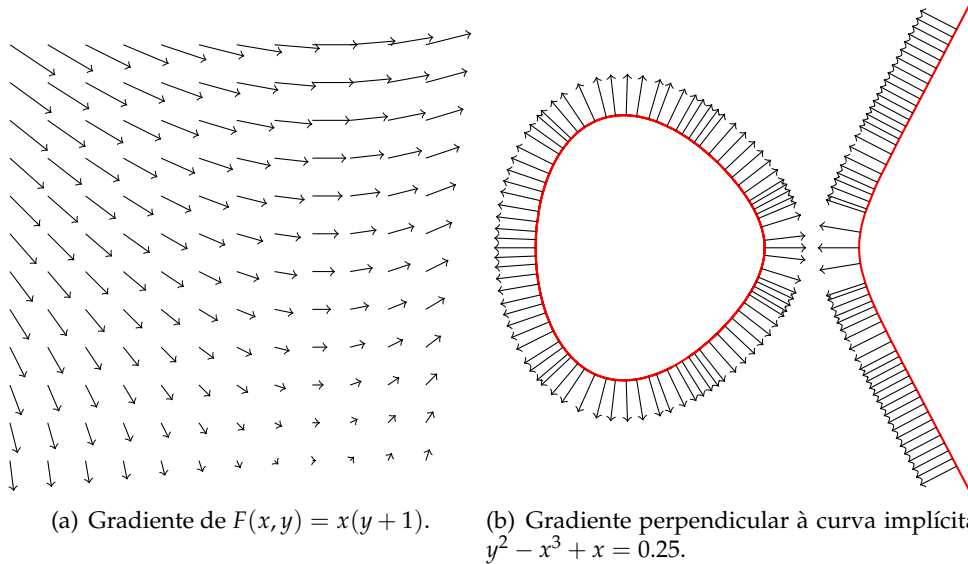


Figura 2.5: Vetor gradiente.

### 2.1.3 Isotopia

O conceito de isotopia é uma ferramenta utilizada para fazer comparações entre conjuntos. É bastante útil para comparar um conjunto com alguma aproximação deste. Isto será utilizado nos capítulos seguintes, onde serão feitas aproximações de curvas, requerendo-se que estas sejam isotópicas.

Em [4] apresenta-se a seguinte definição de isotopia<sup>1</sup>:

**Definição 7** Uma isotopia entre duas curvas planas  $S, S' \subset \mathbb{R}^2$  é uma aplicação contínua

$$\gamma : S \times [0, 1] \rightarrow \mathbb{R}^2$$

onde, para cada  $t \in [0, 1]$  fixo,  $\gamma(\cdot, t)$  é um homeomorfismo de  $S$  em sua imagem, ou seja, é uma aplicação contínua que possui uma inversa contínua, e que deforma continuamente  $S$  em  $S'$ :  $\gamma(S, 1) = S'$ .

<sup>1</sup>A definição de [4] se refere a superfícies, aqui foi adaptada ao caso de curvas planas.



A Figura 2.6 ilustra uma isotopia entre uma curva e uma aproximação desta mesma. A isotopia está representada através de segmentos de reta perpendiculares às arestas da aproximação poligonal, onde cada segmento corresponde à aplicação  $\gamma(p, t)$ , dado um  $p$  fixo sobre a curva e variando-se  $t$  no intervalo  $[0, 1]$ .

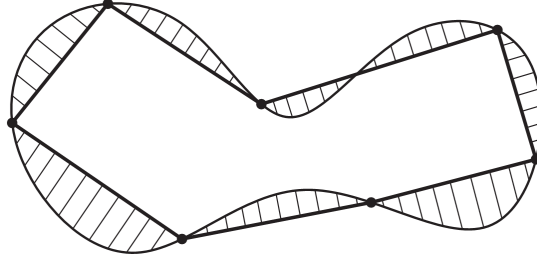


Figura 2.6: Isotopia entre uma curva suave e sua aproximação.

## 2.2 Superfícies

Uma superfície é um subconjunto bidimensional imerso em um espaço tridimensional. Nesta seção serão vistas definições para superfícies.

### 2.2.1 Definição paramétrica

Em [7] definem-se superfícies regulares da seguinte forma:

**Definição 8** Um subconjunto  $S \subset \mathbb{R}^3$  é uma superfície regular se, para cada  $p \in S$ , existe uma vizinhança  $V \in \mathbb{R}^3$  e uma aplicação  $\mathbf{x} : U \rightarrow V \cap S$  de um aberto  $U \subset \mathbb{R}^2$  em  $V \cap S \subset \mathbb{R}^3$  tal que:

1.  $\mathbf{x}$  é diferenciável, ou seja, se for escrito:

$$\mathbf{x}(u, v) = (x(u, v), y(u, v), z(u, v)), \quad (u, v) \in U,$$

as funções  $x(u, v)$ ,  $y(u, v)$ ,  $z(u, v)$  possuem derivadas parciais contínuas de todas as ordens em  $U$ .

2.  $\mathbf{x}$  é um homeomorfismo, ou seja,  $\mathbf{x}$  possui uma inversa  $\mathbf{x}^{-1} : V \cap S \rightarrow U$  que é contínua.
3. Condição de regularidade: Em cada  $q \in U$ , o produto vetorial  $\frac{\partial}{\partial u}\mathbf{x}(q) \wedge \frac{\partial}{\partial v}\mathbf{x}(q) \neq 0$ .

A aplicação  $\mathbf{x}$  é chamada de parametrização de  $S$ .

Como exemplo, seja  $S^2$  a esfera de raio unitário, definida como o conjunto de pontos cuja distância à origem é igual a 1. Dado  $V = \{(\theta, \phi); 0 < \theta < \pi, 0 < \phi < 2\pi\}$  a aplicação  $\mathbf{x} : V \rightarrow \mathbb{R}^3$  dada por

$$\mathbf{x}(\theta, \phi) = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)$$

é uma parametrização de  $S^2$ .  $\theta$  é chamado colatitude, enquanto  $\phi$  é conhecido como longitude. A Figura 2.7 ilustra geometricamente o significado destes parâmetros.

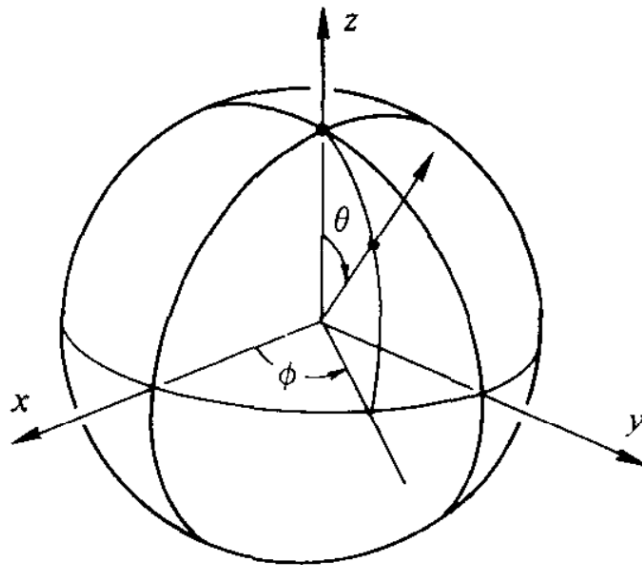


Figura 2.7: Esfera definida parametricamente.

A Figura 2.8 ilustra outros exemplos de superfícies paramétricas. 2.8(a) é a superfície dada por

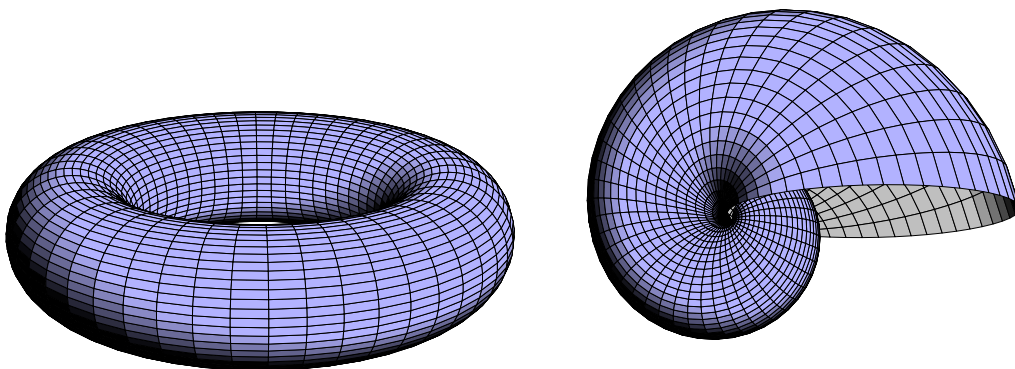
$$\mathbf{x}(u, v) = (\cos(v)(3 + \cos(u)), \sin(v)(3 + \cos(u)), \sin(u)),$$

com  $u, v \in [0, 2\pi]$ .

2.8(b) é dada por

$$\mathbf{x}(u, v) = (1.21^v(\sin(u) \cos(u)), 1.21^v(\sin(u)^2 \sin(v)), 1.21^v(\sin(u)^2 \cos(v))),$$

com  $u \in [0, \pi]$ ,  $v \in [-5\pi/2, \pi/4]$ .



(a)

(b)

Figura 2.8: Exemplos de superfícies paramétricas.

### 2.2.2 Definição implícita

**Definição 9** Dada uma função  $F : \mathbb{R}^3 \rightarrow \mathbb{R}$  ao menos  $C^0$  contínua, uma superfície implícita é o conjunto:

$$S = \{p = (x, y, z) \in \mathbb{R}^3; F(p) = 0\}$$

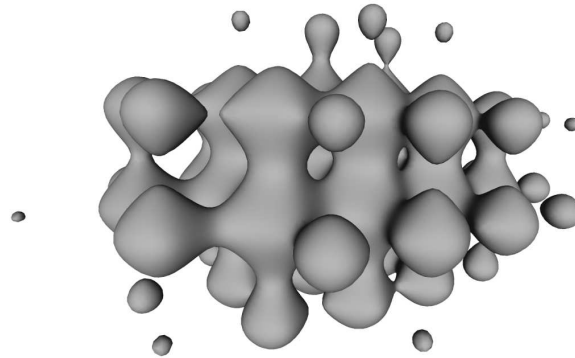
As superfícies implícitas também podem ser denotadas por  $S = F^{-1}(0)$ . Caso 0 seja um valor regular de  $F$ , ou seja, se as derivadas parciais  $\frac{\partial F}{\partial x}$ ,  $\frac{\partial F}{\partial y}$ ,  $\frac{\partial F}{\partial z}$  não se anulam simultaneamente em pontos sobre  $S$ , então pode-se provar que  $S$  é uma superfície regular (prova em [7]).

Do mesmo modo que foi feito para curvas, os pontos  $p \in \mathbb{R}^2$  que não pertencem à superfície  $S$  podem ser classificados de acordo com o sinal de  $F(p)$ :

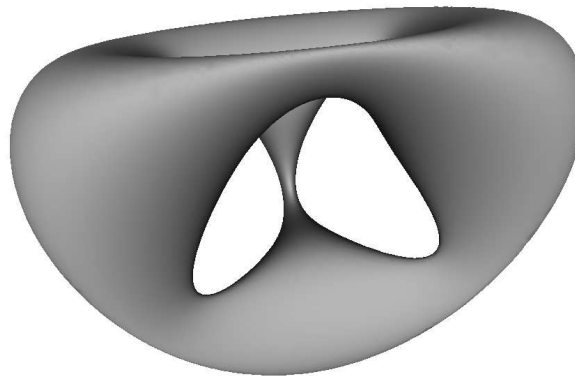
**Definição 10** O ponto  $p = (x, y, z) \in \mathbb{R}^3$  que satisfaz  $F(p) > 0$  é chamado ponto exterior à superfície  $S = F^{-1}(0)$ .

**Definição 11** O ponto  $p = (x, y, z) \in \mathbb{R}^3$  que satisfaz  $F(p) < 0$  é chamado ponto interior à superfície  $S = F^{-1}(0)$ .

A Figura 2.9 ilustra alguns exemplos de superfícies implícitas. 2.9(a) representa a superfície dada por  $0.4(\sin(5x) + \sin(5y) + \sin(5z)) + 0.1x^2 + 0.3y^2 + 0.2z^2 - 0.5 = 0$ . 2.9(b) é a superfície dada por  $(x^2 + y^2 + z^2 - 23.75)^2 - 0.8((z - 5)^2 - 2x^2)((z + 5)^2 - 2y^2) = 0$ .



(a)



(b)

Figura 2.9: Exemplos de Superfícies implícitas.

De forma análoga à utilizada na seção de curvas implícitas, pode-se definir o gradiente de uma função definida no espaço  $\mathbb{R}^3$ :

**Definição 12** *Seja  $F : \mathbb{R}^3 \rightarrow \mathbb{R}$  uma função ao menos  $C^1$  contínua, define-se  $\nabla F : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  da seguinte forma:*

$$\nabla F(x, y, z) = \left( \frac{\partial}{\partial x} F(x, y, z), \frac{\partial}{\partial y} F(x, y, z), \frac{\partial}{\partial z} F(x, y, z) \right).$$

$\nabla F(x, y, z)$  é chamado vetor gradiente de  $F$  no ponto  $(x, y, z)$ .

### 2.2.3 Isotopia

O mesmo conceito de isotopia visto para curvas pode ser utilizado em superfícies, fazendo-se as alterações necessárias:

**Definição 13** *Uma isotopia entre duas superfícies  $S, S' \subset \mathbb{R}^3$  é uma aplicação contínua*

$$\gamma : S \times [0, 1] \rightarrow \mathbb{R}^3$$

*onde, para cada  $t \in [0, 1]$  fixo,  $\gamma(\cdot, t)$  é um homeomorfismo de  $S$  em sua imagem, e que deforma continuamente  $S$  em  $S'$ :  $\gamma(S, 1) = S'$ .*

## Representação de curvas e superfícies

ESTE CAPÍTULO traz um estudo a respeito de diversas formas de se obter aproximações de curvas e superfícies. Serão analisados diversos métodos e trabalhos relacionados, com ênfase a métodos de representação de curvas e superfícies descritas implicitamente.

### 3.1 Representação de curvas

O problema de representação consiste em se obter uma estrutura que aproxima a curva. Esta tarefa é relativamente simples de ser executada quando a curva é descrita parametricamente, pois basta tomar amostras do domínio do parâmetro da função  $\alpha$ , formando uma seqüência de valores  $t_1 < t_2 < \dots < t_n$ , a partir dos quais forma-se uma seqüência de pontos  $\alpha(t_1), \alpha(t_2), \dots, \alpha(t_n)$ , a curva é aproximada através de algum tipo de interpolação destes pontos, como por exemplo por interpolação linear, ligando cada dois pontos consecutivos  $\alpha(t_i)$  e  $\alpha(t_{i+1})$  por segmentos de reta, conforme ilustra a Figura 3.1(a). A representação da Figura 3.1(b) utiliza os mesmos pontos amostrados da Figura 3.1(a), porém utiliza um método de interpolação que torna a representação mais suave.

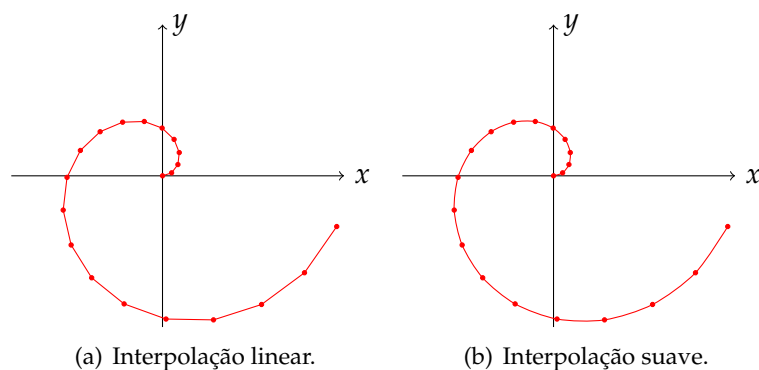


Figura 3.1: Representação de curvas paramétricas.

No caso de curvas descritas implicitamente a tarefa de representação torna-se mais trabalhosa, pois é preciso analisar pontos no plano, inferindo de alguma forma as regiões por onde a curva passa. Na próxima seção serão vistas diversas formas de se realizar esta tarefa.

## 3.2 Representação de curvas implícitas

Há diversas maneiras de se representar curvas implícitas. Há métodos que tratam da rasterização de curvas, isto é, geram visualizações em forma de imagens bidimensionais. Outros métodos extraem estruturas que podem ser usadas como aproximações das curvas, geralmente na forma de curvas poligonais. Este tipo de método pode, normalmente, ser dividido em duas etapas: primeiramente faz-se uma *amostragem* de pontos sobre as curvas; em seguida realiza-se uma etapa de *estruturação*, onde os dados da amostragem são utilizados para se formar uma estrutura que representa a curva, mantendo (idealmente) a geometria e a topologia da curva.

### 3.2.1 Rasterização

Entre os métodos de rasterização, uma forma simples, porém pouco prática, é calcular a distância euclidiana entre pontos em uma grade regular e a curva desejada, onde cada ponto é relacionado a um pixel de uma imagem, e cada pixel só será pintado caso a distância euclidiana deste ponto à curva seja menor do que um determinado limiar. Este método não é prático pois se  $n$  for o número de pixels num lado de um quadrado, serão necessários  $n^2$  cálculos de distância para aproximar a curva tendo este quadrado como domínio, porém o número esperado de pixels que serão pintados é apenas  $\mathcal{O}(n)$ , ou seja, muitos cálculos desnecessários são efetuados.

[17] descreve um algoritmo que utiliza um esquema de subdivisão através de recursões como em *Quadrees*, evitando cálculos desnecessários. A distância euclidiana de um ponto à curva foi verificada utilizando-se uma distância aproximada, que é assintoticamente equivalente à distância euclidiana. Foi descrito também um teste suficiente para uma função polinomial de grau  $k$  não possuir raízes em um círculo. A Figura 3.2 ilustra um resultado de rasterização obtido por [17].

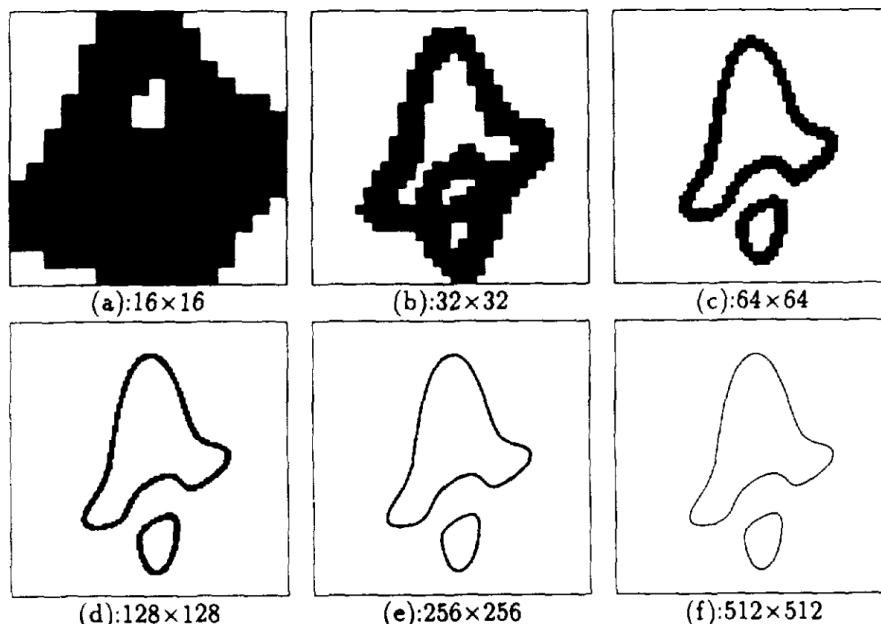


Figura 3.2: Um método para rasterização de curvas implícitas.

### 3.2.2 Aproximação poligonal

A forma mais comum de se obter uma aproximação poligonal de curvas implícitas é através de métodos de *enumeração*, onde o domínio é decomposto em uma grade (normalmente retangular ou triangular). A amostragem é feita calculando-se os pontos de interseção da curva com cada aresta da grade. A estruturação é feita ligando-se os pontos que pertençam a uma mesma célula, como ilustra a Figura 3.3.

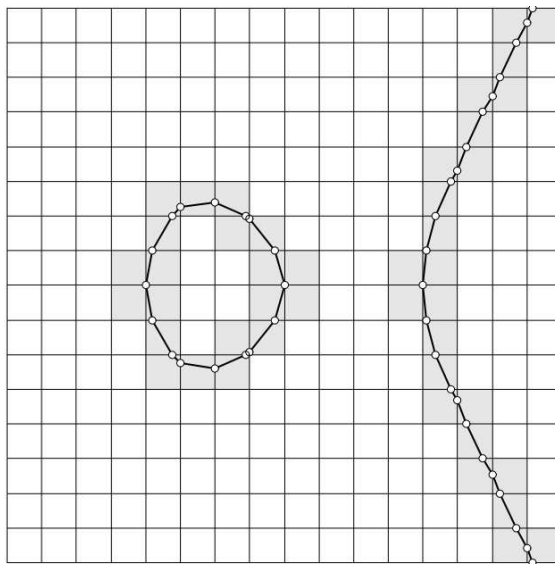


Figura 3.3: Método de enumeração aplicado à curva implícita  $y^2 - x^3 + x = 0$ .

Um problema deste método é a escolha da resolução da grade, isto é, como escolher o tamanho das células de forma a não perder informações importantes sobre a topologia da curva. Outro problema é como encontrar as interseções da curva com as arestas da grade, o que pode ser feito avaliando-se o sinal da função em cada vértice. Se em uma aresta os vértices possuem sinais opostos, então há algum ponto de interseção na aresta, que pode ser obtido de forma aproximada através de interpolação linear, ou, se for exigida uma maior precisão, através de algum método numérico para encontrar raízes de funções, como bisseção, por exemplo. Porém não se pode descartar arestas que não possuam variação de sinal nos vértices, pois a curva pode também cortar a aresta duas vezes (ou qualquer número par de vezes). Para evitar estes problemas, a solução mais simples é utilizar uma grade de alta resolução, porém, esta solução torna os algoritmos mais custosos, exigindo desnecessariamente operações em pontos de muitas regiões por onde a curva não passa. Uma outra solução mais eficiente é utilizar uma *enumeração adaptativa*, onde as células são maiores em regiões distantes da curva, e menores em regiões próximas. A maior tarefa neste caso é como saber quando uma célula está mais próxima ou mais afastada da curva em questão. Mais adiante serão analisadas algumas formas de se realizar tal tarefa.

Há diversas outras formas para se fazer amostragem de pontos sobre curvas implícitas, como por exemplo através de algoritmos de *ray-casting*, onde são calculadas interseções da curva com uma família de retas, chamadas de *raios*, como ilustra a Figura 3.4.

Outra forma de se obter amostragem é por *continuação*, onde a partir de um ponto  $p$  sobre a curva, utiliza-se um método para prever a posição de outro ponto  $q$  próximo a  $p$ , como o método de Euler, que utiliza um vetor perpendicular ao gradiente da função em  $p$

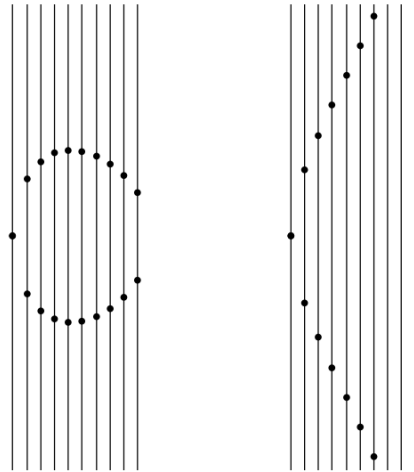


Figura 3.4: Amostragem por *ray-casting*.

para aproximar  $q$ , em seguida aplica-se o método de Newton para levar  $q$  a um ponto sobre a curva, este processo é repetido a partir de cada novo ponto da curva encontrado, até que se tenham pontos suficientes sobre a curva. A estruturação pode ser feita simplesmente ligando-se os pontos na seqüência em que são encontrados. Uma desvantagem deste método é a necessidade de se conhecer a priori um ponto pertencente à curva, e para se obter possíveis componentes disjuntas da curva é preciso haver inicialmente um ponto de cada uma destas componentes.

Há ainda métodos de amostragem baseados em física de partículas, como o apresentado por [8], que inicia com uma amostragem aleatória no plano, cada ponto da amostra se move em órbitas que tendem à curva, utilizando-se para este movimento um campo gradiente modificado. A Figura 3.5 mostra um exemplo deste método, ilustrando as órbitas destes pontos juntamente com a amostragem final.

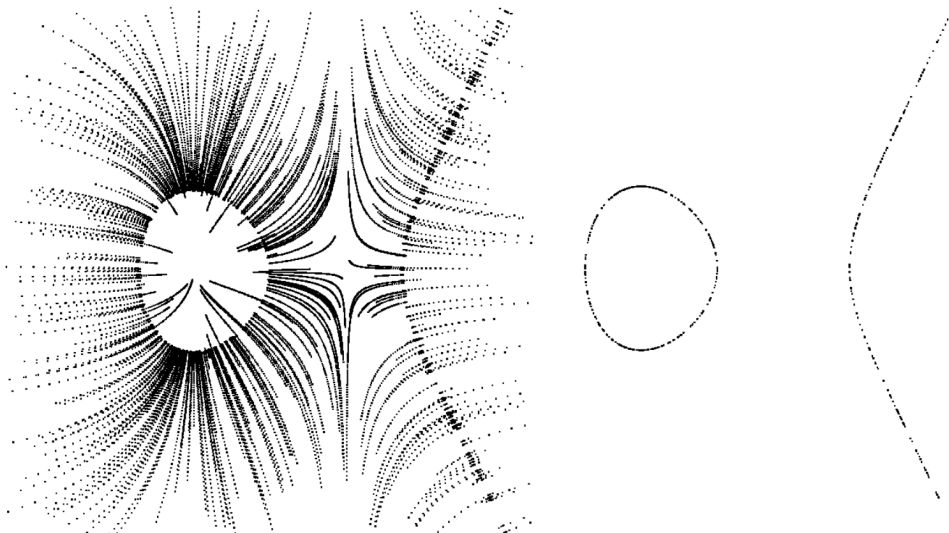


Figura 3.5: Órbitas e amostragem final.



### 3.2.3 Enumeração adaptativa

Os métodos de enumeração adaptativa, como já foi visto, podem ser usados para encontrar aproximações de curvas implícitas a partir de grades cujas células possuem um tamanho variável, de forma a evitar cálculos desnecessários em regiões por onde a curva não passa. Nesta seção serão analisados dois trabalhos nesta linha.

O trabalho de [13] apresenta um método de enumeração adaptativa que utiliza aritmética intervalar e diferenciação automática (descritos brevemente a seguir) para encontrar uma aproximação robusta de curvas definidas implicitamente.

#### Aritmética intervalar

A aritmética intervalar [6] consiste em realizar operações básicas sobre intervalos, como somas, subtrações, multiplicações, etc. Para cada função  $F(x, y, \dots)$  operando sobre valores reais, há uma função correspondente  $\bar{F}(\bar{x}, \bar{y}, \dots)$  operando sobre intervalos, cujo resultado é um intervalo (preferencialmente o mínimo) que contém todos os valores de  $F(x, y, \dots)$ , onde  $x, y, \dots$  pertencem aos intervalos  $\bar{x}, \bar{y}, \dots$ , respectivamente.

Por exemplo, seja a função  $F(x) = x^2$ , pode-se definir  $\bar{F}$  por

$$\bar{F}([a, b]) = \begin{cases} [\min(a^2, b^2), \max(a^2, b^2)] & \text{se } ab \geq 0, \\ [0, \max(a^2, b^2)] & \text{se } ab < 0. \end{cases}$$

#### Diferenciação automática

Em diversas aplicações, é preciso calcular o valor de derivadas de funções reais, há algumas formas de se realizar tal tarefa. Uma possibilidade é através da *diferenciação simbólica*, que manipula a expressão de uma função  $F$  para encontrar uma outra expressão referente à derivada de  $F$  (em relação a uma determinada variável). Outra forma é através da *diferenciação numérica*, que calcula aproximações das derivadas utilizando métodos numéricos. Ambas as possibilidades são simples de se implementar, porém a diferenciação automática pode gerar expressões muito longas, o que torna o cálculo das derivadas lento, enquanto que grandes erros de aproximação podem surgir ao se utilizar diferenciação numérica. Existe uma outra técnica chamada *diferenciação automática*, que une a velocidade da diferenciação numérica com a precisão da diferenciação simbólica. Há diversos trabalhos relacionados, como em [18, 10, 11].

Na diferenciação automática, dada uma função  $F : \mathbb{R}^n \rightarrow \mathbb{R}$ , avaliam-se tuplas de valores  $(u_0, u_1, \dots, u_n)$ , onde  $u_0$  é o valor da função, enquanto  $u_i$  é sua derivada parcial em relação à  $i$ -ésima variável. As operações elementares são definidas entre tuplas, de acordo com a regra da cadeia e fórmulas elementares do cálculo. Com isto, calculam-se automaticamente as derivadas de funções dadas por expressões complicadas através da aplicação das regras definidas para cada operação elementar presente nas expressões.

Como exemplo, para  $n = 2$  as operações de soma, multiplicação e função seno são dadas respectivamente por:

$$\begin{aligned} (u_0, u_1, u_2) + (v_0, v_1, v_2) &= (u_0 + v_0, u_1 + v_1, u_2 + v_2) \\ (u_0, u_1, u_2) \cdot (v_0, v_1, v_2) &= (u_0 v_0, u_0 v_1 + u_1 v_0, u_0 v_2 + u_2 v_0) \\ \text{sen}(u_0, u_1, u_2) &= (\text{sen } u_0, u_1 \cos u_0, u_2 \cos u_0) \end{aligned}$$

### Adaptatividade espacial

Utilizando os recursos da aritmética intervalar, pode-se descrever um algoritmo simples de enumeração adaptativa, como através desta função:

---

**Função**  $\text{Explore}(B)$ .

---

**Entrada:** uma função  $F : \mathbb{R}^2 \rightarrow \mathbb{R}$ , um retângulo

$B = \{(x, y) \in \mathbb{R}^2; x \in [x_0, x_1], y \in [y_0, y_1]\}$ , e um parâmetro  $\epsilon > 0$ .

**Saída:** uma subdivisão adaptativa espacial de  $B$ .

**início**

**se**  $0 \notin \bar{F}(B)$  **então** descarte  $B$ ;

**senão se**  $\text{diam}(B) < \epsilon$  **então** retorne  $B$ ;

**senão**

        Divida  $B$  em pedaços menores  $B_i$ ;

**para cada**  $i$  **faça**  $\text{Explore}(B_i)$

**fim**

**fim**

---

Esta função cria uma subdivisão adaptativa de  $B$ , geralmente divide-se  $B$  em quatro pedaços iguais, gerando assim uma *Quadtree*.  $\text{diam}(B) < \epsilon$  significa que  $\max(x_1 - x_0, y_1 - y_0) < \epsilon$ , para um determinado  $\epsilon$ . Todas as células que são retornadas possuem o mesmo tamanho, diz-se então que esta é uma subdivisão adaptativa espacial. A Figura 3.6 ilustra um exemplo deste tipo de subdivisão, as células pintadas de cinza são as retornadas pelo algoritmo.

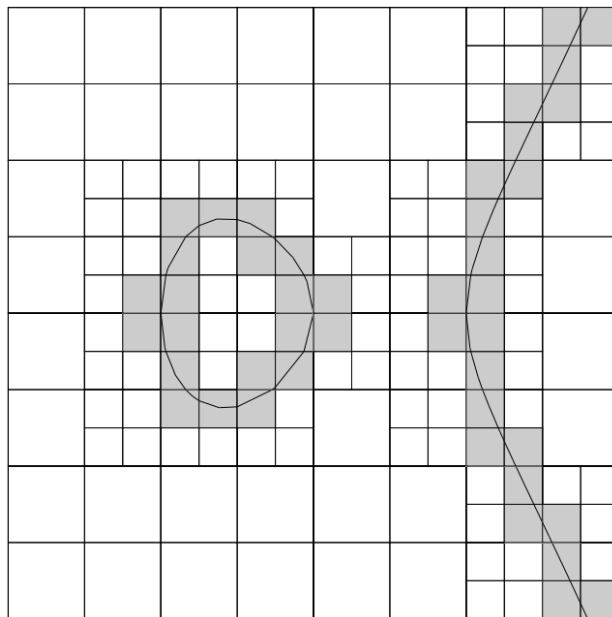


Figura 3.6: Subdivisão adaptativa espacial.

Uma desvantagem deste método de subdivisão é que pequenas componentes da curva podem não ser detectadas, neste caso o valor de  $\epsilon$  precisa ser menor. Além disto, não há uma adaptação à curvatura da curva, ou seja, a mesma quantidade de pontos são amostrados em regiões tanto de alta como de baixa curvatura, o que pode prejudicar a geometria da curva.

### Adaptatividade geométrica

A adaptatividade geométrica precisa obter informações sobre a curvatura de curvas implícitas a fim de ajustar o tamanho das células da subdivisão a estas informações. Para isto, [13] utiliza o gradiente de  $F$  (que pode ser obtido através de diferenciação automática). O valor de  $\overline{\nabla F}$  é analisado em cada célula, como o gradiente possui duas componentes, este valor é composto por dois intervalos, um indicando a variação de  $\frac{\partial F}{\partial x}$  e outro indicando a variação de  $\frac{\partial F}{\partial y}$ . Se o diâmetro destes intervalos for um valor pequeno (i.e. abaixo de uma tolerância  $\delta$ ) então há uma pequena variação do gradiente na célula, conseqüentemente há também uma pequena variação na direção da curva dentro da célula. Assim, um algoritmo de subdivisão espacial pode utilizar a função abaixo:

---

**Função**  $\text{Explore}(B)$ .

---

**Entrada:** uma função  $F : \mathbb{R}^2 \rightarrow \mathbb{R}$ , um retângulo

$B = \{(x, y) \in \mathbb{R}^2; x \in [x_0, x_1], y \in [y_0, y_1]\}$ , e parâmetros  $\epsilon > 0$  e  $\delta > 0$ .

**Saída:** uma subdivisão adaptativa geométrica de  $B$ .

**início**

**se**  $0 \notin \overline{F}(B)$  **então** descarte  $B$ ;

**senão se**  $\text{diam}(B) < \epsilon$  **ou**  $\text{diam}(\overline{\nabla F}(B)) < \delta$  **então**  $\text{approx}(B)$ ;

**senão**

        Divida  $B$  em pedaços menores  $B_i$ ;

**para cada**  $i$  **faça**  $\text{Explore}(B_i)$

**fim**

**fim**

---

A diferença entre esta função e a anterior é, além da presença da verificação da condição  $\text{diam}(\overline{\nabla F}(B)) < \delta$ , a chamada à função  $\text{approx}(B)$ , que aproxima a curva através de segmentos de reta dentro da célula  $B$ . A Figura 3.7 ilustra um exemplo da aplicação deste algoritmo.

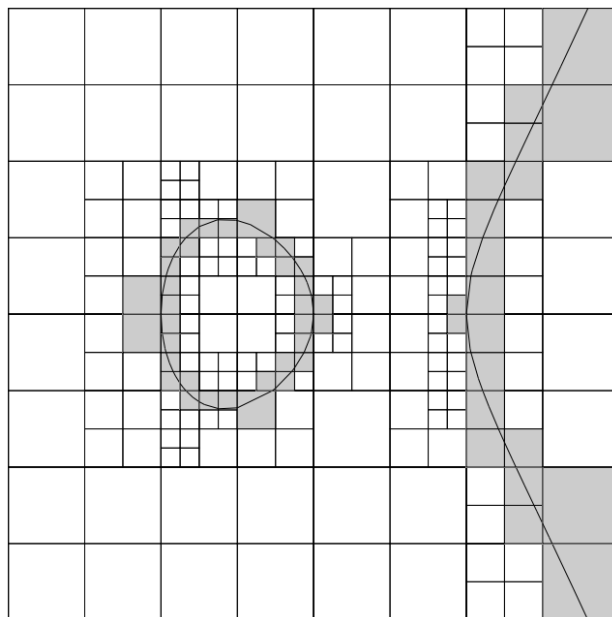


Figura 3.7: Subdivisão adaptativa geométrica.

Já no trabalho de Plantinga e Vegter [15] descreve-se um algoritmo de aproximação isotópica de curvas e superfícies implícitas, avaliando o valor da função e seu gradiente em intervalos do domínio onde a curva será aproximada, utilizando para tal avaliação aritmética intervalar.

### Algoritmo de subdivisão

Seja  $S = F^{-1}(0)$  uma curva implícita, onde  $F : \mathbb{R}^2 \rightarrow \mathbb{R}$  é uma função suave e 0 é um valor regular de  $F$ , ou seja,  $\nabla F$  não se anula nos pontos da curva. O seguinte algoritmo forma uma subdivisão do plano:

---

**Algoritmo 3:** Subdivisão do plano.

---

**Entrada:** uma função  $F : \mathbb{R}^2 \rightarrow \mathbb{R}$  e um retângulo

$$B = \{(x, y) \in \mathbb{R}^2; x \in [x_0, x_1], y \in [y_0, y_1]\}.$$

**Saída:** um *Quadtree*  $\mathcal{T}$ .

Inicialize o *Quadtree*  $\mathcal{T}$  com o retângulo  $B$ ;

Subdivide  $\mathcal{T}$  até que em todos os nós folhas  $C$  tenha-se

$$0 \notin \bar{F}(C) \vee \langle \nabla \bar{F}(C), \nabla \bar{F}(C) \rangle > 0;$$


---

Para cada nó folha do *Quadtree* resultante, se a primeira condição ( $0 \notin \bar{F}(C)$ ) for verdadeira, então garante-se que este nó não contém pontos da curva a ser aproximada. Se a segunda condição ( $\langle \nabla \bar{F}(C), \nabla \bar{F}(C) \rangle > 0$ ) for verdadeira, então o ângulo formado entre dois vetores gradientes em pontos do nó folha não ultrapassa  $90^\circ$ , como  $\nabla F$  em pontos da curva possui a mesma direção do vetor normal à curva, então conseqüentemente o ângulo formado entre dois vetores normais à curva não ultrapassa  $90^\circ$ . Além disto, se  $\langle \nabla \bar{F}(C), \nabla \bar{F}(C) \rangle > 0$ , então ao menos um dos dois termos  $\bar{F}_x(C) \cdot \bar{F}_x(C)$  e  $\bar{F}_y(C) \cdot \bar{F}_y(C)$  (onde  $F_x = \frac{\partial F}{\partial x}$ ) não contém zero, ou seja  $F$  é estritamente crescente ou decrescente na direção  $x$  ou  $y$ , portanto a função  $F$  é localmente parametrizável numa das direções dos eixos.

### Aproximação linear por partes

O algoritmo apresentado por [15] para a aproximação linear por partes de curvas verifica cada nó folha do *Quadtree* resultante do algoritmo 3, criando vértices nos pontos médios das arestas destes nós sempre que houver uma alteração no sinal de  $F$  nos pontos extremos destas arestas, e em seguida ligando estes vértices, construindo assim a aproximação desejada. Como o objetivo do trabalho é obter uma aproximação isotópica, não houve preocupação com a geometria desta aproximação, por isto foi utilizado o ponto médio das arestas, e não algum ponto mais próximo à curva.

Este algoritmo forma uma aproximação isotópica da curva implícita. A prova desta afirmação colocada em [15] e será resumida aqui. A prova é dividida em três partes:

- Constrói-se uma aproximação a partir de uma grade regular, com determinadas restrições, e prova-se que esta aproximação é isotópica a curva  $S = F^{-1}(0)$ .
- Removem-se restrições.
- Mostra-se que a aproximação obtida a partir da grade regular é isotópica à criada a partir do *Quadtree*.

### Grade regular

Seja  $G$  uma grade regular, tal que em cada célula  $C$  é satisfeita a condição  $0 \notin \bar{F}(C) \vee \langle \nabla \bar{F}(C), \nabla \bar{F}(C) \rangle > 0$ . Assume-se  $F \neq 0$  nos vértices de  $G$ , e que  $S$  intersecta cada aresta de  $G$  no máximo uma vez. Pela restrição do produto interno,  $S$  é parametrizável na direção de um dos eixos, logo não é possível haver sinais alternados nos vértices de  $C$ , pois  $F$  seria crescente ao longo de uma aresta e decrescente ao longo da aresta paralela. Assim  $S$  intersecta no máximo duas arestas de  $C$ . Como  $S$  é localmente parametrizável, não pode haver *loops* na célula, e como  $0$  é um valor regular de  $F$ , também não pode haver auto-interseções. Logo, se em uma célula  $C$  existem dois pontos de interseção de  $S$  com as arestas de  $C$ , então a curva  $S$  dentro da célula  $C$  é isotópica a um segmento de reta. De fato, seja  $S$  parametrizável na direção  $y$ . Se  $S$  intersectar as arestas esquerda e direita, pode-se simplesmente utilizar interpolação linear na direção  $y$  para mover de forma contínua esta parte de  $S$  para o segmento de reta da aproximação (que liga os pontos médios das arestas esquerda e direita), conforme ilustra a Figura 3.8(a). Se a curva  $S$  não intersectar as arestas esquerda e direita, pode-se aplicar uma escala na direção  $x$  de forma que a curva intersecte a aresta horizontal de  $C$  no mesmo ponto da aproximação, e pode-se então mover de forma contínua os pontos da curva sobre os pontos da aproximação na direção  $y$ , conforme ilustra a Figura 3.8(b).

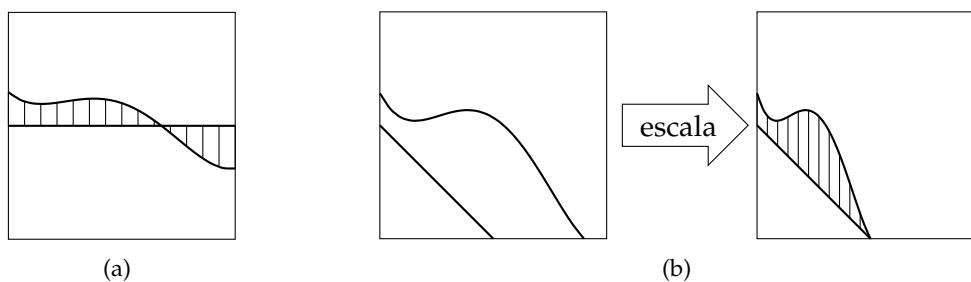


Figura 3.8: Isotopia local entre curva e aproximação em uma célula.

### Remoção de restrições

As restrições aplicadas à curva  $S$  são retiradas nesta etapa da prova. Se  $S$  intersecta uma aresta da grade mais de uma vez, ela será parametrizável na direção perpendicular a esta aresta nas duas células adjacentes. Pelo teorema do valor médio é fácil ver que a restrição do produto interno evita  $S$  de ser “muito” curvada (i.e. acima de  $90^\circ$ ) entre os dois pontos de interseção, e portanto  $S$  não pode sair das duas células que contêm os pontos de interseção. Pode-se mover continuamente a curva entre os pontos de interseção na direção parametrizável projetando-a sobre a aresta, e pode-se mover esta projeção de forma a eliminar os dois pontos de interseção. Fazendo-se isto para todos os pares de pontos de interseção, verifica-se que  $S$  é isotópica à curva que intersecta cada aresta no máximo uma vez, sendo portanto isotópica à aproximação poligonal.

Se  $S$  passar por um vértice da grade, pode-se deformar  $S$  movendo-a continuamente para a curva  $F + \epsilon = 0$ . Para  $\epsilon$  pequeno isto também forma uma isotopia. Pode-se tomar  $\epsilon$  arbitrariamente pequeno com uma perturbação simbólica, considerando  $F$  estritamente positiva num vértice onde  $F \geq 0$ .

**Quadrees**

Nesta terceira etapa prova-se que a aproximação usando *Quadtree* é isotópica à encontrada usando grade regular. Dado um nó folha  $C$  do *Quadtree*, com suas arestas da aproximação. Ao subdividir  $C$ , a restrição do produto interno continua válida para seus quatro nós filhos. Como  $S$  é parametrizável em  $C$ , também será nos nós filhos. As únicas alterações topológicas possíveis ocorrem quando a subdivisão forma dois novos vértices em uma aresta de  $C$ , conforme ilustra a Figura 3.9.

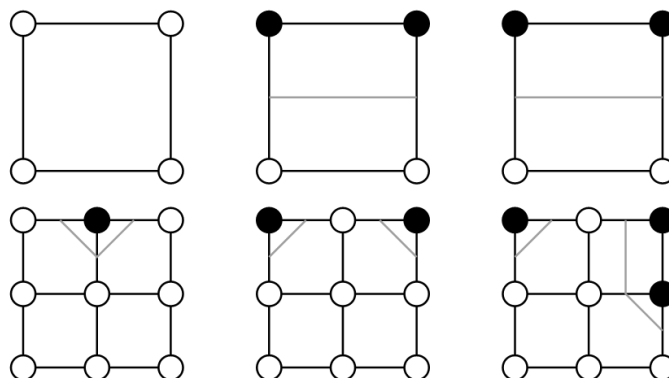


Figura 3.9: Alterações topológicas após a subdivisão.

Os dois novos vértices serão detectados quando a célula vizinha for subdividida, e serão conectados a dois outros vértices já existentes. Esta alteração corresponde a “empurrar” parte da curva através da aresta, nas duas células adjacentes a esta aresta a aproximação a isotopia se mantém. Ao subdividir todas as folhas do *Quadtree* chega-se a um *Quadtree* completo, com uma aproximação isotópica à encontrada na grade regular. Portanto a aproximação encontrada é isotópica à curva  $S$ .

A Figura 3.10 mostra o resultado da execução deste algoritmo para aproximação da curva  $S = F^{-1}(0)$ , onde  $F(x, y) = x^2(1 - x^2) - y^2 + 0.01$ .

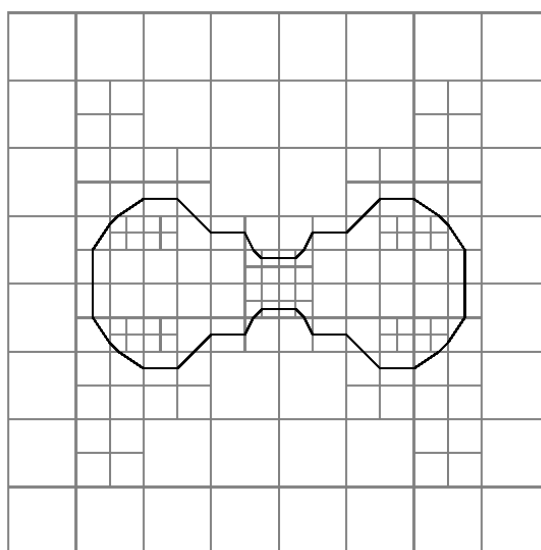


Figura 3.10: Aproximação da curva  $x^2(1 - x^2) - y^2 + 0.01 = 0$ .

### 3.3 Representação de superfícies

De forma análoga ao descrito para curvas, o problema de representação de superfícies consiste em se obter uma estrutura que aproxima a superfície. Quando as superfícies são descritas parametricamente, elas são relativamente simples de serem representadas. Uma forma de se fazer isto é formando uma grade quadrangular no domínio dos parâmetros. Para cada célula da grade obtém-se um quadrilátero sobre a superfície. Ao unir todos os quadriláteros obtém-se a aproximação desejada, conforme ilustra a Figura 3.11.

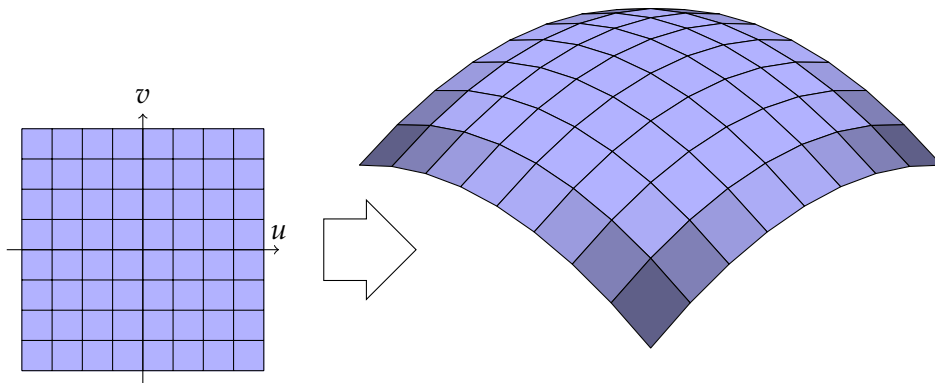


Figura 3.11: Representação de superfícies paramétricas.

Quando a superfície é descrita implicitamente, entretanto, a tarefa de representação não é tão simples, pois é preciso percorrer o espaço  $\mathbb{R}^3$  tomando informações necessárias para encontrar as regiões por onde a superfície passa. Na próxima seção são vistos alguns métodos de representação de superfícies descritas implicitamente.

### 3.4 Representação de superfícies implícitas

Existem várias abordagens para a tarefa de representar superfícies implícitas, há aquelas focadas na renderização de superfícies, cujo objetivo é gerar uma imagem ilustrando a superfície. Há também aquelas cujo principal objetivo é a extração de superfícies, onde estas são aproximadas através de malhas poligonais.

#### 3.4.1 Renderização de superfícies implícitas

Há algumas formas possíveis de se renderizar superfícies implícitas. Um forma possível é simplesmente usar um método de extração de superfícies (como será visto na próxima seção) para obter uma malha poligonal, que pode ser trivialmente rasterizada através de algoritmos *Z-buffer*. Outra possibilidade é amostrar a superfície utilizando outros objetos que a aproximam, e usar algum outro método para renderizá-los. Pode-se ainda utilizar algoritmos de *ray-tracing*, onde são simulados raios de luz através de retas, uma reta  $\alpha(t)$  intersecta uma superfície  $S = F^{-1}(0)$  nos pontos que satisfazem  $F(\alpha(t)) = 0$ , logo o problema se resume a encontrar as raízes da função  $G(t) = F(\alpha(t))$ . A Figura 3.12 ilustra alguns resultados obtidos por [5], que resolve esta tarefa utilizando operações sobre intervalos através da aritmética afim (mais informações em [6]).

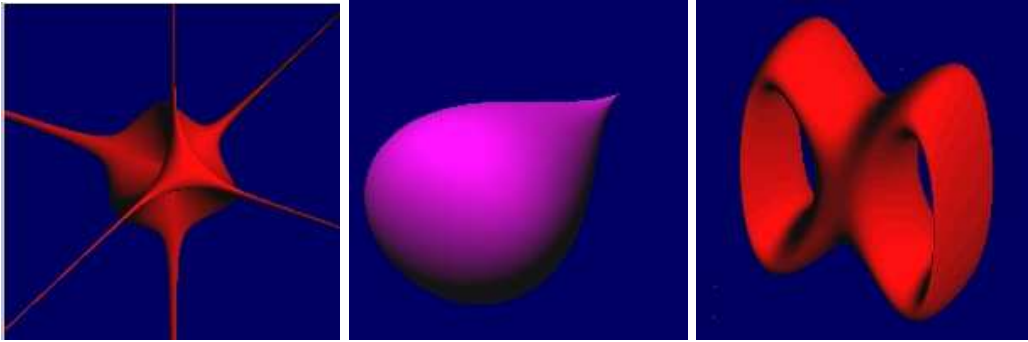


Figura 3.12: Superfícies implícitas renderizadas por *ray-tracing*.

### 3.4.2 Extração de superfícies implícitas

O método de extração de superfícies implícitas mais conhecido na literatura é o algoritmo de *Marching Cubes*, desenvolvido por Lorensen e Cline [14]. Neste algoritmo, o espaço de onde se quer extrair a superfície é subdividido, formando uma grade regular composta por paralelepípedos. Cada paralelepípedo é analisado, verificando-se o valor da função em cada um de seus oito vértices. Quando houver variação no sinal da função, a superfície é aproximada dentro do paralelepípedo através de polígonos cujos vértices estão sobre as arestas do paralelepípedo onde há variação no sinal. Estes polígonos são obtidos através de uma tabela que indica, para cada configuração de vértices, como formar os polígonos (uma forma comum é definir o polígono como a união de triângulos). Existem 256 casos possíveis (8 vértices, cada um podendo ser positivo ou não-positivo), porém, pode-se observar que diversos casos diferem apenas por rotações ou por inversões de sinais. Agrupando estes casos o número de possibilidades diminui para apenas 15, e os polígonos podem ser definidos com base nestes casos, conforme indica a Figura 3.13.

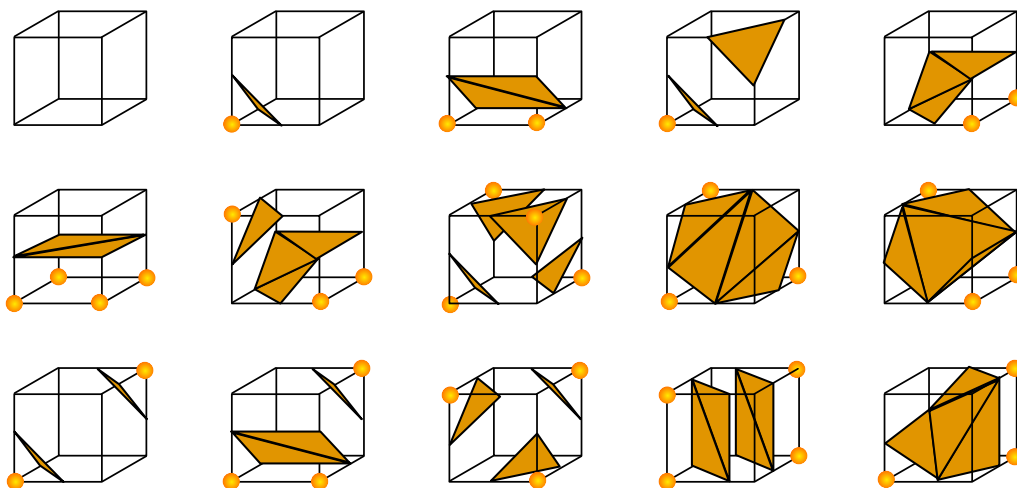


Figura 3.13: Casos possíveis no algoritmo *Marching Cubes*.

Um problema neste algoritmo é a possível presença de casos ambíguos, isto é, quando há mais de uma forma possível de se gerarem os polígonos, a forma escolhida pode não corresponder à correta, no sentido de não manter a topologia da superfície. Há diversos trabalhos que modificam este algoritmo de forma a tratar os casos ambíguos mantendo a



topologia, como por exemplo em [12], que utiliza uma técnica para garantir um resultado topologicamente correto. A Figura 3.14 mostra uma comparação da malha gerada pelo algoritmo clássico com a gerada pelo algoritmo de [12].

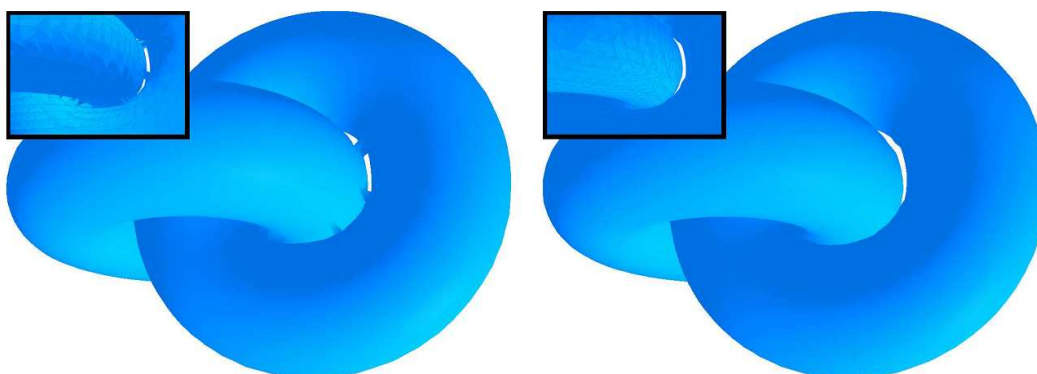


Figura 3.14: Comparação entre o algoritmo de *Marching Cubes* clássico e o de [12].

Um outro trabalho que estende o algoritmo clássico de *Marching Cubes* é [15], já citado na Seção 3.2.3 no caso de curvas implícitas. O Algoritmo 3 também pode ser utilizado para tratar de superfícies implícitas, fazendo as devidas alterações. Neste caso é gerado um *Octree*, ao invés de um *Quadtree*. As células do *Octree* são analisadas da mesma forma que as células da grade regular do algoritmo clássico, o número de casos possíveis cai para 9, conforme a Figura 3.15, diminuição causada pela forma como é feita a subdivisão. Alguns casos ambíguos (casos 4 e 6 da Figura 3.15) são tratados de forma a manter a topologia da superfície.

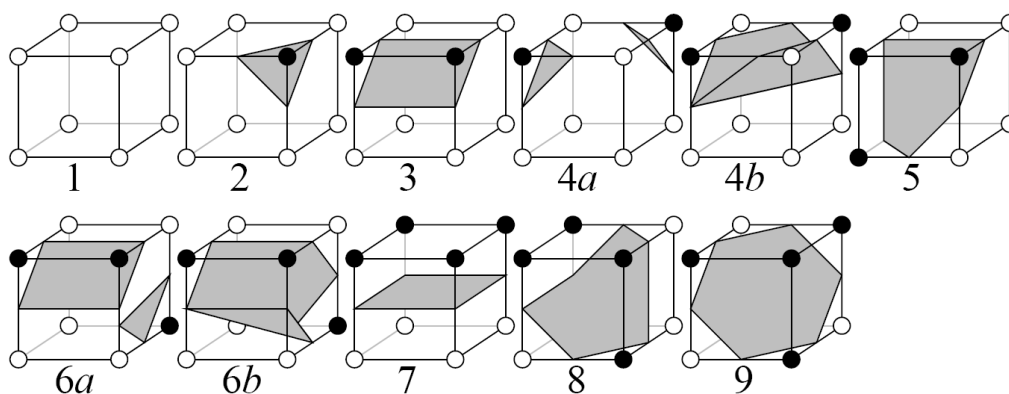


Figura 3.15: Casos utilizados do algoritmo *Marching Cubes*.

Como o objetivo de [15] é obter uma aproximação isotópica, não houve tratamento específico para a geometria das superfícies, de forma que a aproximação encontrada é bastante grosseira. A Figura 3.16 ilustra o resultado obtido por [15] da aproximação da superfície dada pelo zero da função  $F(x, y, z) = x^4 - 5x^2 + y^4 - 5y^2 + z^4 - 5z^2 + 10$ .

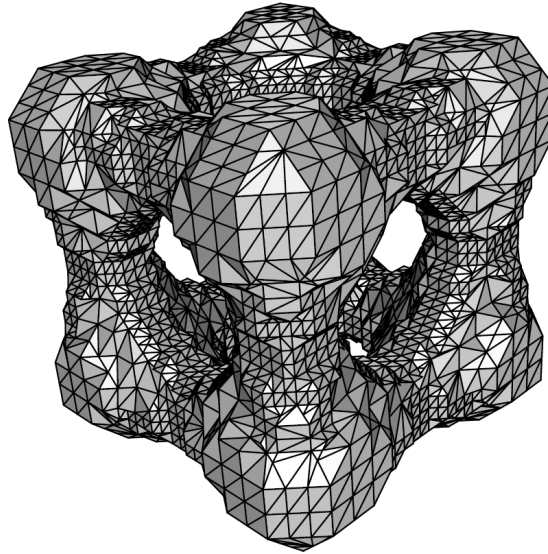


Figura 3.16: Aproximação isotópica da superfície  $x^4 - 5x^2 + y^4 - 5y^2 + z^4 - 5z^2 + 10 = 0$ .

Outro trabalho relacionado à aproximação isotópica de superfícies é [3], que utiliza teoria de Morse para obter uma aproximação linear por partes. É fornecido um critério para garantir a isotopia, a partir do qual é deduzido um algoritmo para obter uma malha isotópica à superfície real.

# Aproximação suave de curvas e superfícies implícitas

NESTE CAPÍTULO são apresentados métodos para aproximação de curvas e superfícies implícitas, criados a partir de outros métodos conhecidos, com algumas modificações. O objetivo destes métodos é obter aproximações lineares por partes, isotópicas e que apresentem um determinado nível de suavidade. No caso de curvas obtém-se uma aproximação poligonal, isto é, um conjunto de segmentos de reta suficientemente próximos à curva. No caso de superfícies o resultado são malhas triangulares, onde os triângulos devem estar suficientemente próximos à superfície.

## 4.1 Curvas planas

Dada uma curva implícita  $S = F^{-1}(0)$ , onde 0 é um valor regular de  $F$ , pode-se utilizar o algoritmo de [15] (descrito na seção 3.2.3) para construir uma estrutura que tem a mesma topologia da curva a ser representada. Desta forma um dos objetivos é alcançado, porém é preciso realizar algum tipo de processamento sobre esta estrutura a fim de se melhorar a geometria da aproximação, para torná-la mais suave.

Uma primeira alteração a ser feita no algoritmo de [15] está relacionada à forma de divisão do plano. Alguns nós folhas do *Quadtree* podem ser vizinhos e estarem em níveis diferentes, é preciso verificar quando essa situação ocorre para evitar a formação de descontinuidades na aproximação da curva, conforme ilustra a Figura 4.1.

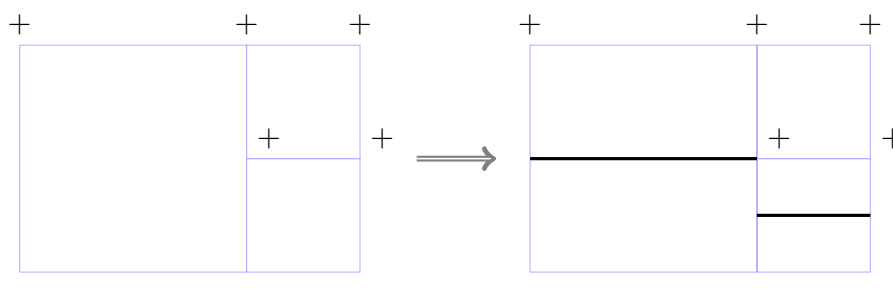


Figura 4.1: Nós vizinhos em níveis diferentes do *Quadtree* gerando uma descontinuidade na aproximação da curva (em preto).

Para abolir a presença de discontinuidades na aproximação de curvas, pode-se utilizar uma estrutura que é topologicamente dual à estrutura *Quadtree*, como a utilizada em [16], que será chamada de *DualQuadtree*. Nesta estrutura, cada célula do *DualQuadtree* corresponde a um vértice do *Quadtree*, cada aresta do *DualQuadtree* corresponde a duas células vizinhas do *Quadtree*, e cada vértice do *DualQuadtree* corresponde a uma célula do *Quadtree*, conforme ilustra a Figura 4.2.

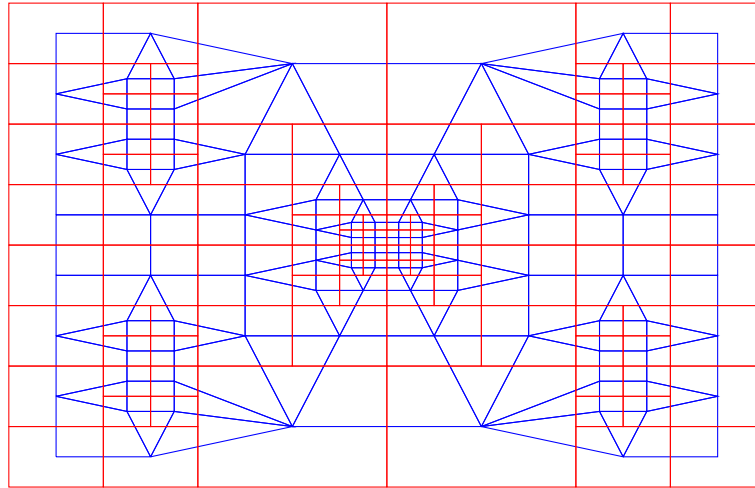


Figura 4.2: *Quadtree* (em vermelho) e *DualQuadtree* (em azul).

Uma forma de se obter o *DualQuadtree* a partir do *Quadtree* é utilizando-se as funções recursivas *faceProc*, *edgeProc* e *vertProc*. A função *faceProc* recebe como entrada um nó do *Quadtree*, *edgeProc* recebe dois nós vizinhos, e *vertProc* recebe quatro nós do *Quadtree* que compartilham um vértice, as chamadas recursivas estão indicadas na Figura 4.3. O *DualQuadtree* é obtido através de uma chamada à função *faceProc*, passando-se como entrada o nó raiz do *Quadtree*. Se em alguma chamada da função *vertProc* os quatro nós passados sejam folhas, cria-se então uma nova célula do *DualQuadtree*, cujos vértices são os pontos centrais de cada um destes quatro nós.

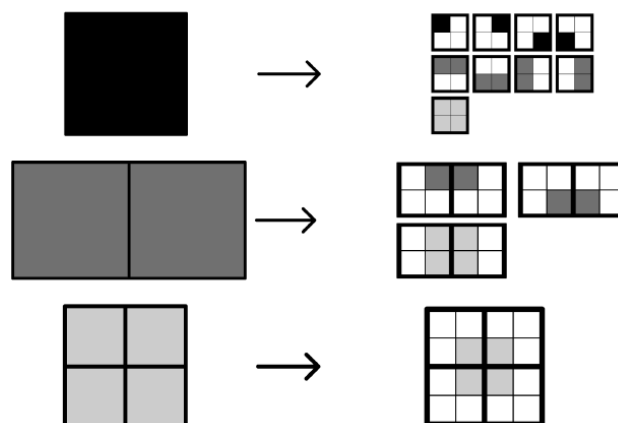


Figura 4.3: Funções recursivas para obtenção do *DualQuadtree*: *faceProc* (preto), *edgeProc* (cinza escuro) e *vertProc* (cinza claro).

O mesmo procedimento utilizado para aproximar a curva nas células do *Quadtree* pode ser utilizado agora nas células do *DualQuadtree*, sem haver problema de descon-

tinuidade na aproximação obtida. A Figura 4.4 ilustra o resultado deste procedimento utilizando a mesma curva da Figura 3.10. Uma outra alteração no algoritmo, já utilizada na Figura 4.4, é a alteração do cálculo dos pontos que estão sobre a curva. Ao invés de se utilizar o ponto médio das arestas do *DualQuadtree* onde há variação no sinal da função, pode-se aplicar algum método numérico como o de biseção, que encontra um ponto  $p$  sobre a aresta, tal que  $|F(p)| < \epsilon$ , para um determinado  $\epsilon$ .

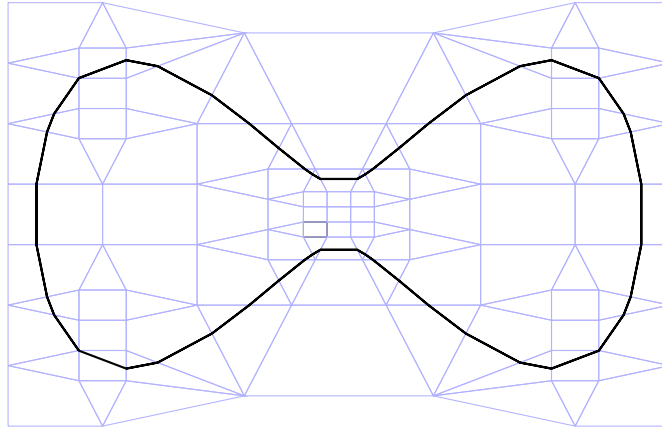


Figura 4.4: Aproximação da curva  $x^2(1 - x^2) - y^2 + 0.01 = 0$ , utilizando o *DualQuadtree*.

Em geral, a topologia da aproximação obtida desta maneira equivale à encontrada através do *Quadtree*. Porém pode haver casos onde as células do *DualQuadtree* não satisfazem as mesmas condições das células do *Quadtree*, nestes casos uma forma de garantir a isotopia é subdividindo as células do *DualQuadtree*, de forma que cada parte desta subdivisão satisfaça as condições que garantem a aproximação isotópica. Por exemplo, pode-se tomar as interseções da célula do *DualQuadtree* com as células do *Quadtree*, extraindo as aproximações a partir destas interseções.

#### 4.1.1 Refinamento

A aproximação obtida até aqui obedece ao critério de manter a topologia, porém ainda não pode ser considerada suave. Pode-se aplicar um método de refinamento para suavizar as aproximações de curvas. Uma outra forma simples de se obter tal refinamento é alterar o critério de subdivisão do *Quadtree* do Algoritmo 3, de forma a continuar a subdivisão até que a aproximação seja suave o suficiente. Uma forma de se refinar a curva é apresentada a seguir: primeiramente é necessário utilizar uma estrutura diferente para representar a curva. Dados dois pontos  $p_1$  e  $p_2 \in \mathbb{R}^2$ , caso  $F(p_1)$  e  $F(p_2)$  possuam sinais diferentes, o segmento  $e = p_1p_2$  é chamado *segmento ativo*. O vértice de  $e$  que é externo à curva  $F^{-1}(0)$  é representado por  $p_e$ , o vértice interno à curva é representado por  $n_e$ , e o ponto do segmento ativo intersectado pela curva é representado por  $z_e$  (este ponto pode ser encontrado numericamente utilizando um método de biseção).

Seja  $E$  um conjunto de segmentos ativos, formado inicialmente pelas arestas de células do *DualQuadtree* que sejam segmentos ativos. Dados  $e_1, e_2 \in E$ , se  $p_{e_1}p_{e_2}$  ou  $n_{e_1}n_{e_2}$  for uma aresta do *DualQuadtree* então o segmento  $z_{e_1}z_{e_2}$  é uma aresta da aproximação da curva  $S = F^{-1}(0)$ . A aproximação obtida desta forma equivale à obtida anteriormente (a partir do *DualQuadtree*).

Uma vez criados o conjunto  $E$  e as arestas da aproximação da curva, pode-se então refinar esta aproximação através de subdivisões de cada aresta. Isto pode ser realizado

da seguinte maneira: dados  $e_1, e_2 \in E$ , onde  $z_{e_1} z_{e_2}$  é uma aresta da aproximação da curva, calcula-se o ponto  $p_{e_3}$  como sendo o ponto médio entre  $p_{e_1}$  e  $p_{e_2}$ . Da mesma maneira o ponto  $n_{e_3}$  é calculado como o ponto médio entre  $n_{e_1}$  e  $n_{e_2}$ , formando assim um novo segmento ativo  $e_3$  que está situado entre  $e_1$  e  $e_2$ , como ilustra a Figura 4.5.

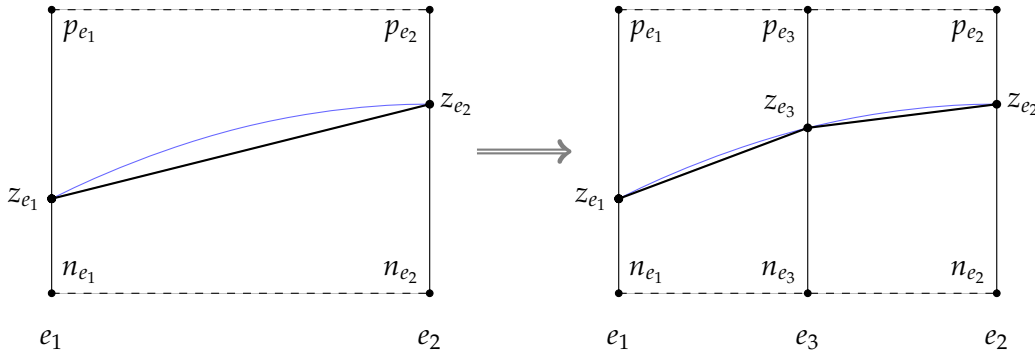


Figura 4.5: Subdivisão de arestas.

Nesta etapa de subdivisão, é possível que o segmento  $e_3$  não seja um segmento ativo (quando  $p_{e_3} < 0$  ou  $n_{e_3} > 0$ ). Neste caso não é possível utilizar o método de bisseção para encontrar  $z_{e_3}$ , uma vez que a raiz de  $F$  não está situada entre  $n_{e_3}$  e  $p_{e_3}$ . Ainda assim, pode-se encontrar um ponto da curva através do método de Newton para aproximação de raízes de funções, utilizando o ponto mais próximo da curva dentre  $n_{e_3}$  e  $p_{e_3}$  como sendo a primeira aproximação da raiz da função. O método de Newton se encarrega de “atrair” esta primeira aproximação até a curva. A Figura 4.6 ilustra uma situação onde isto ocorre.

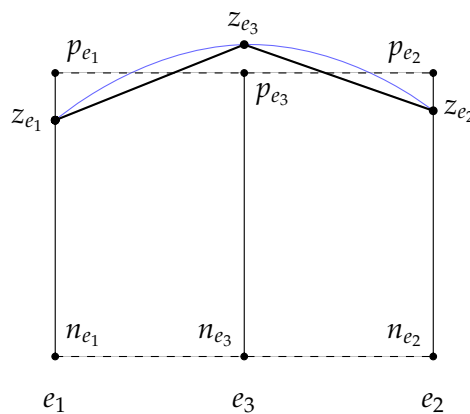


Figura 4.6: Situação onde não se forma um segmento ativo na etapa de subdivisão de arestas, o ponto  $z_{e_3}$  foi calculado pelo método de Newton a partir de  $p_{e_3}$  (ponto mais próximo à curva).

Já se sabe até agora como subdividir uma aresta. Falta ainda decidir o critério de subdivisão, ou seja, em que situação uma determinada aresta deve ser subdividida. Um critério simples é verificar o comprimento da aresta, ou seja a distância entre os seus vértices, caso este comprimento ultrapasse um determinado valor, a aresta é subdividida. A Figura 4.7 ilustra um exemplo de resultado desta operação aplicada à aproximação encontrada na Figura 4.4.

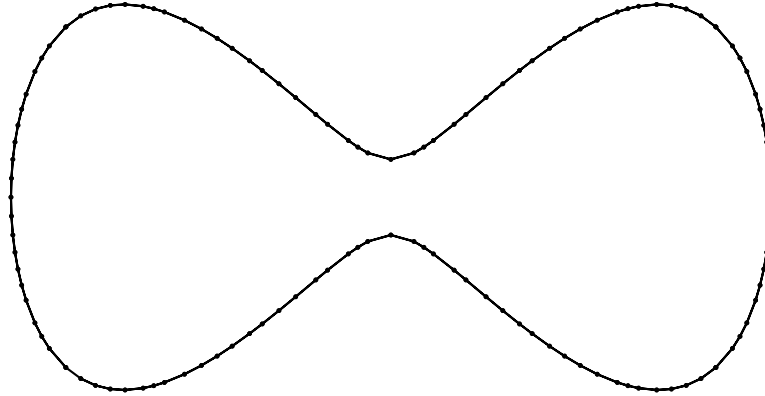


Figura 4.7: Suavização de curva obtida subdividindo-se as arestas pelos seus comprimentos.

O uso deste critério possui a desvantagem de não ser adaptativo, isto é, subdividem-se da mesma forma regiões da curva que possuam alta ou baixa curvatura, o que causa uma amostragem com muitos pontos desnecessários em regiões de baixa curvatura, e com um número insuficiente de pontos em regiões de alta curvatura, como pode ser observado na Figura 4.7.

Outro critério que pode ser utilizado é a verificação do ângulo entre arestas: dadas duas arestas adjacentes  $z_{e_1}z_{e_2}$  e  $z_{e_2}z_{e_3}$ , seja  $\theta$  o ângulo entre os vetores  $z_{e_2} - z_{e_1}$  e  $z_{e_3} - z_{e_2}$ , conforme ilustra a Figura 4.8. Subdivide-se a aresta  $z_{e_1}z_{e_2}$  quando  $\theta > \delta$ , para um determinado  $\delta$ . Este critério possui a vantagem de ser adaptativo, como pode ser observado na Figura 4.9.

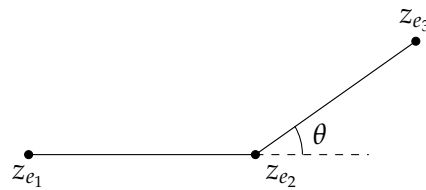


Figura 4.8: Ângulo entre arestas.

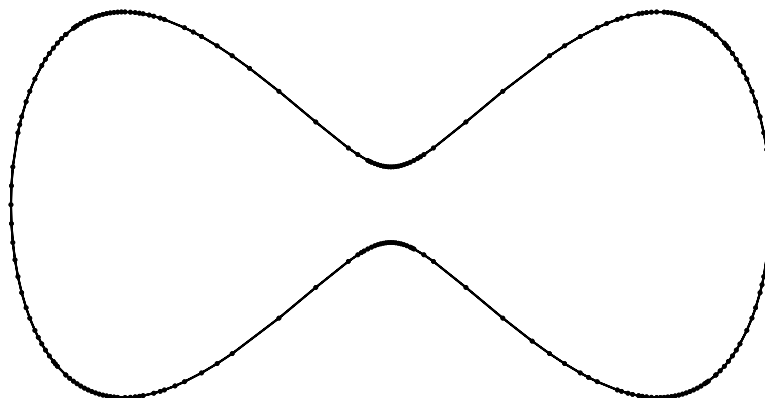


Figura 4.9: Suavização de curva subdividindo-se pelo ângulo entre as arestas, observa-se uma alta concentração de pontos em regiões de alta curvatura.

### 4.1.2 Resultados

O método descrito foi implementado em C++. Para obter os resultados desta seção foi utilizado  $\epsilon = 10^{-8}$  como sendo a tolerância no cálculo numérico das raízes das funções, isto é, obtém-se  $|F(p)| < \epsilon$  em cada vértice  $p$  das aproximações. Como critério de subdivisão verificou-se o ângulo entre as arestas, com a tolerância  $\delta = 0.075$ .

As Figuras 4.10 até 4.13 ilustram alguns exemplos da aplicação do algoritmo de extração de curvas implícitas. Cada figura foi dividida em cinco partes. A parte (a) mostra o *Quadtree* encontrado pelo algoritmo de subdivisão do plano. O *DualQuadtree* obtido a partir deste *Quadtree* pode ser visto em azul na parte (b). A parte (c) mostra a aproximação linear por partes obtida a partir do *DualQuadtree*. A parte (d) mostra como a aproximação se altera após o refinamento. O resultado final é mostrado em (e), é a mesma curva de (d), porém sem destacar os pontos encontrados.

Pode-se fazer uma análise do quão próximo uma curva está de sua aproximação. Uma medida possível é a através distância de Hausdorff entre conjuntos, definida por:

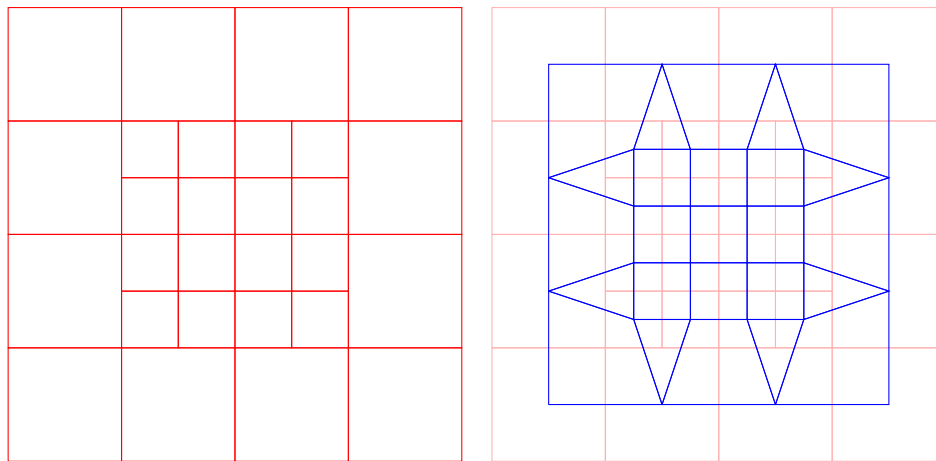
$$d_H(X, Y) = \max\left\{ \sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y) \right\}.$$

Onde  $X$  e  $Y$  são conjuntos, e  $d(x, y)$  é uma função de distância entre pontos  $x$  e  $y$ . No caso da aproximação de curvas, pode-se utilizar  $X$  como sendo a curva  $F^{-1}(0)$ ,  $Y$  a aproximação, e  $d(x, y)$  a distância euclidiana entre pontos no plano. É quase sempre inviável o cálculo exato de  $d_H(X, Y)$ , pois seria preciso calcular  $d(x, y)$  para um número infinito de pontos  $x \in X$  e  $y \in Y$ . O que se pode fazer é obter uma aproximação desta medida, por exemplo calculando-se uma amostragem de pontos sobre as arestas da aproximação. Para cada ponto  $p$  desta amostragem calcula-se um ponto  $\bar{p}$  sobre a curva através do método de Newton,  $d_H(X, Y)$  é aproximado pelo valor máximo entre todos os  $d(p, \bar{p})$  calculados. A tabela abaixo ilustra o resultado desta métrica para algumas aproximações encontradas, ilustrando o valor encontrado antes e depois da suavização, cada aresta foi particionada em cem pontos uniformemente distribuídos.

Curva	Sem suavização	Com suavização
$x^2 + y^2 - 1 = 0$	0.0513113	0.00128388
$(y - x^2 + 1)^4 + (x^2 + y^2)^4 - 1 = 0$	0.0228366	0.00282148
$y^2 - x^3 + x - 0.25 = 0$	0.0697288	0.0013592
$x^2 - xy + y^4 + 0.0001 = 0$	0.00549521	0.00115506

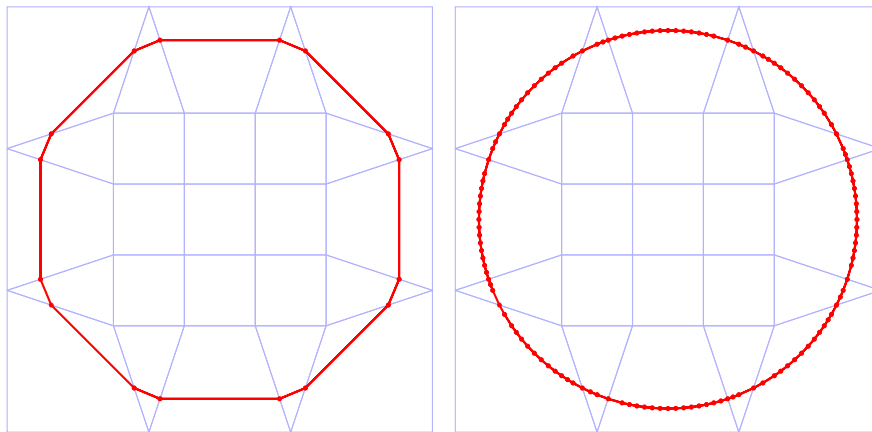
Tabela 4.1: Aproximação da distância de Hausdorff entre a curva e suas aproximações.





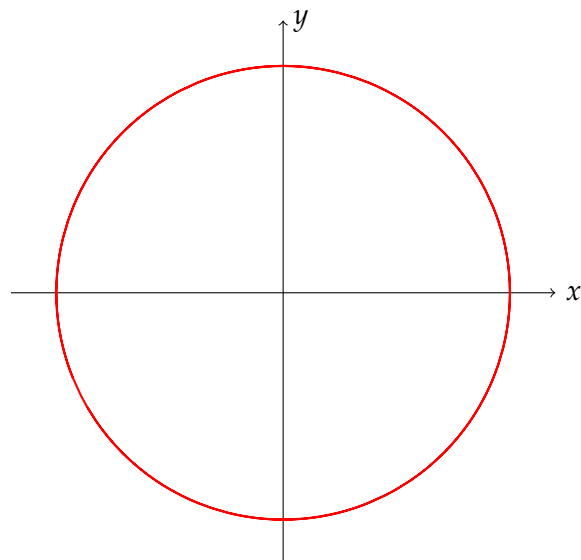
(a) Quadtree.

(b) DualQuadtree.



(c) Aproximação linear.

(d) Suavização.



(e) Aproximação final.

Figura 4.10: Etapas do processo de extração da curva implícita  $x^2 + y^2 - 1 = 0$ .

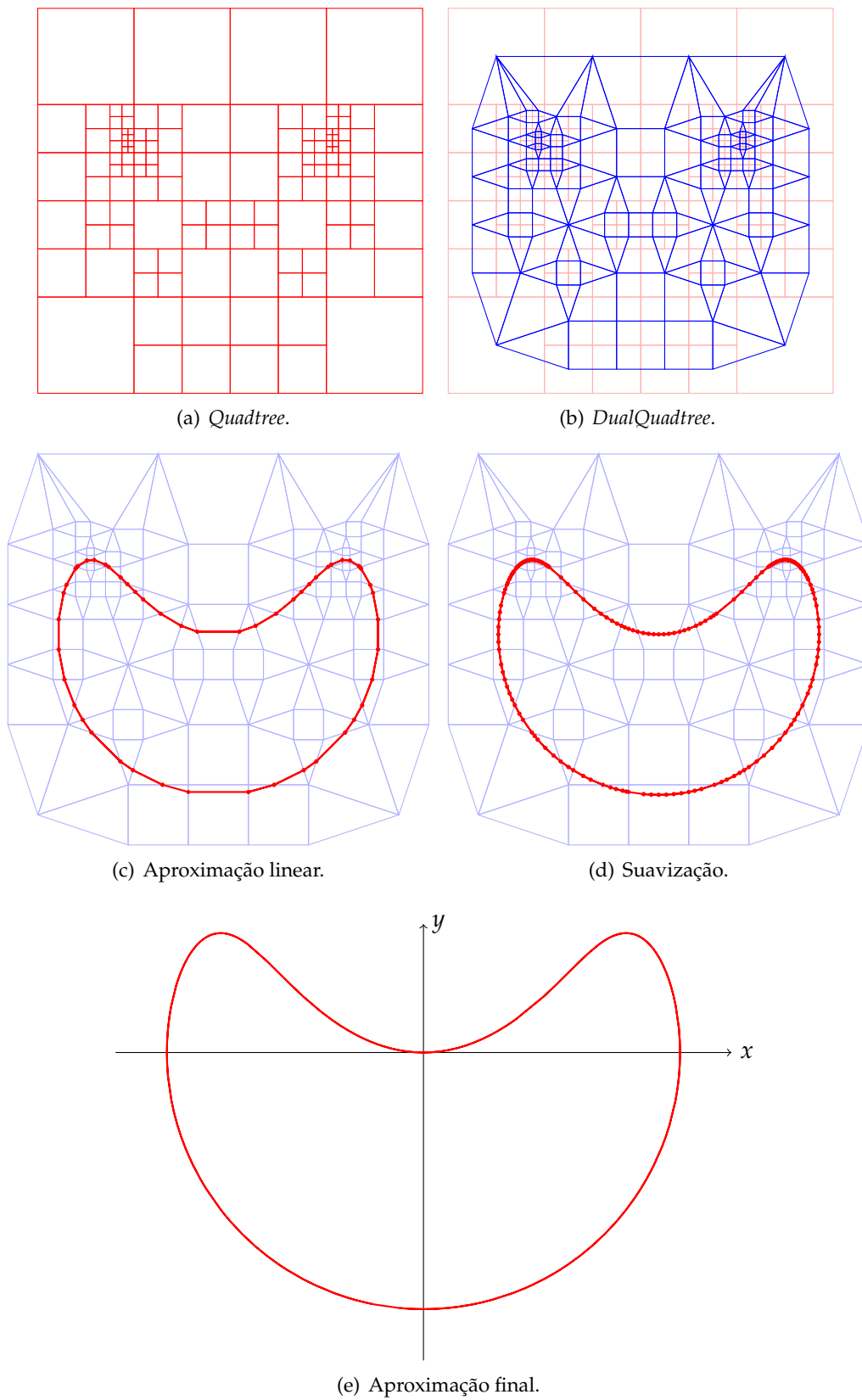


Figura 4.11: Etapas do processo de extração da curva implícita  $(y - x^2 + 1)^4 + (x^2 + y^2)^4 - 1 = 0$ .

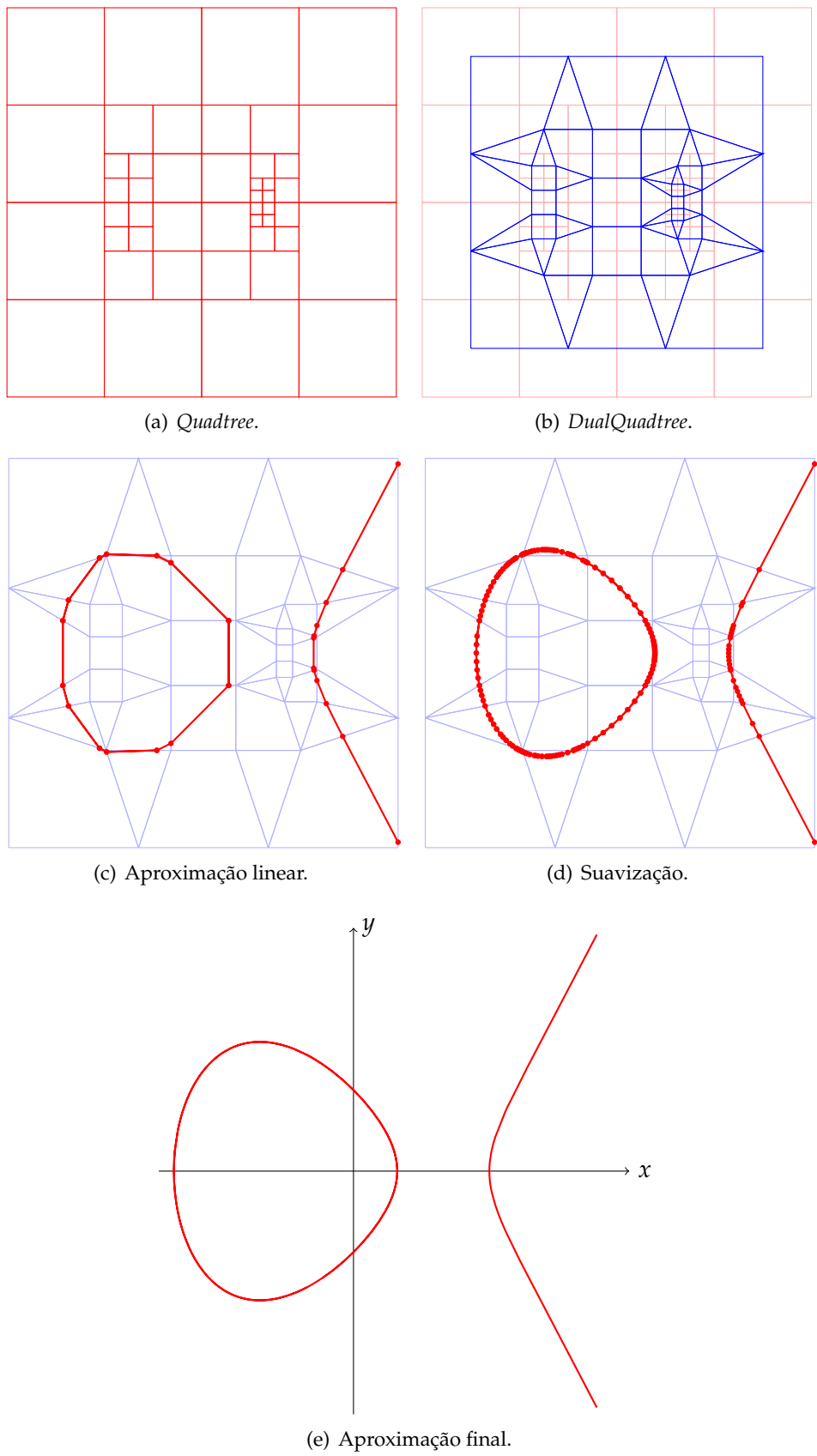


Figura 4.12: Etapas do processo de extração da curva implícita  $y^2 - x^3 + x - 0.25 = 0$ .

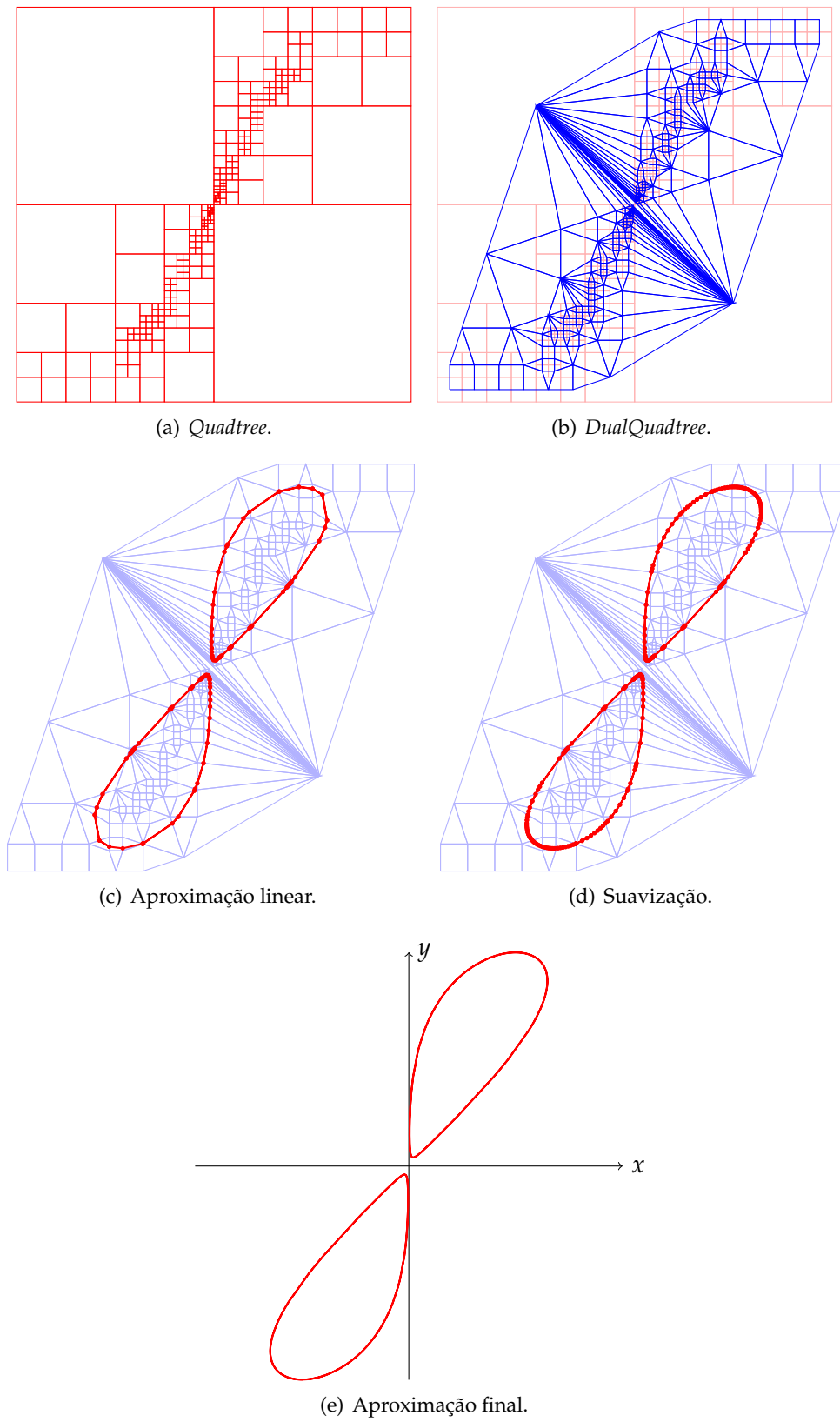


Figura 4.13: Etapas do processo de extração da curva implícita  $x^2 - xy + y^4 + 0.0001 = 0$ .

## 4.2 Superfícies implícitas

Para a geração de malhas suaves aproximando superfícies implícitas foi utilizado inicialmente o algoritmo de [15], que obtém uma aproximação isotópica da superfície. Porém esta aproximação precisa ser tratada para que o resultado seja uma representação suave. A primeira alteração no algoritmo está relacionada à estruturação dos objetos tratados. O mesmo problema que acontece com a aproximação de curvas a partir do *Quadtree* acontece também com aproximação de superfícies a partir do *Octree*: células vizinhas porém em níveis diferentes podem causar descontinuidades na aproximação. A forma como foi isto tratado em [15] possui um inconveniente de não gerar uma malha triangular consistente, isto é, a interseção de dois triângulos pode ser diferente de vazio, de um vértice, ou de uma aresta da triangulação, conforme pode ser visto na Figura 4.14.

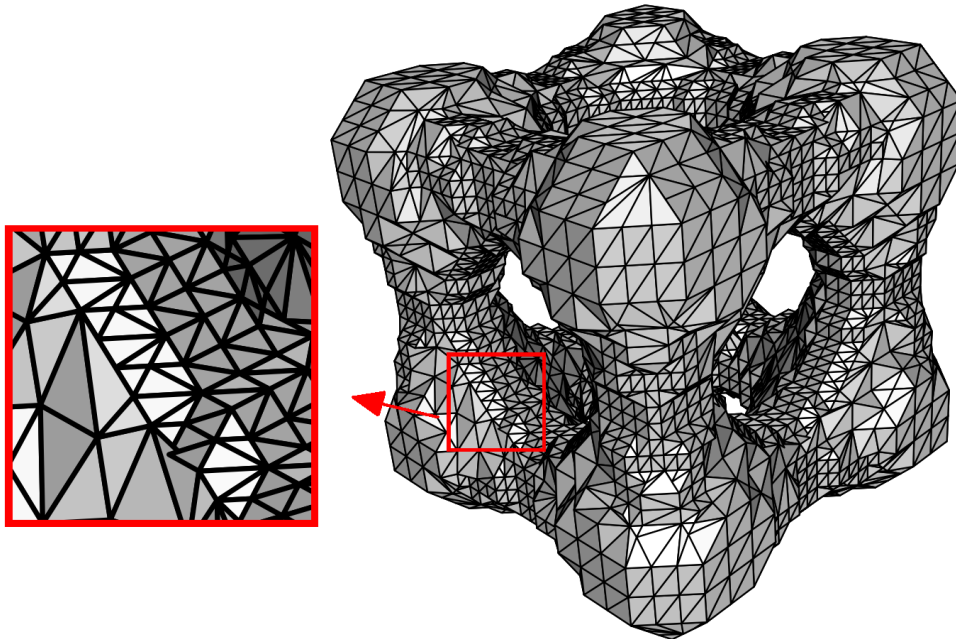


Figura 4.14: Irregularidades na malha encontrada.

Estas irregularidades prejudicam diversas formas de pós-processamento, sendo, portanto, indesejáveis. Para evitá-las pode-se utilizar o algoritmo *Dual Marching Cubes* [16], que processa uma estrutura dual ao *Octree*, que é chamada *DualOctree*. Esta estrutura pode ser encontrada de forma análoga à utilizada para encontrar o *DualQuadtree*, através de chamadas de funções recursivas que constroem o *DualOctree* percorrendo os nós do *Octree* do nó raiz até os nós folhas. Uma vez calculado o *DualOctree*, analisa-se cada célula (que tem a mesma topologia de um cubo) de acordo com os casos do algoritmo clássico *Marching Cubes*, encontrando assim a aproximação desejada. Em geral a aproximação obtida utilizando o *DualOctree* possui a mesma topologia daquela encontrada utilizando o *Octree*, nos casos onde isso não ocorre pode-se subdividir as células do *DualOctree* em partes que satisfazem as condições que garantem a aproximação isotópica.

### 4.2.1 Eliminação de *slivers*

Uma característica do uso do algoritmo *Dual Marching Cubes* é a presença de triângulos de baixa qualidade conhecidos como *slivers*, que possuem um dos lados muito

menor do que os outros dois. É possível utilizar alguma operação para evitar a presença deste tipo de triângulos, esta operação não é necessária se o objetivo for apenas obter uma malha triangular independente da qualidade dos triângulos, porém triângulos de baixa qualidade dificultam outras operações, pois podem causar erros de instabilidade numérica.

Em [16] a maior parte dos *slivers* são eliminados ao se modificarem as posições de vértices do *DualOctree* da seguinte forma: caso em uma aresta do *DualOctree* que seja cortada pela superfície, um dos vértices estiver muito próximo (dada uma determinada tolerância  $\epsilon$ ) à raiz da função, este vértice é reposicionado para esta raiz. A Figura 4.15 ilustra um exemplo de resultado deste método de eliminação de *slivers*.

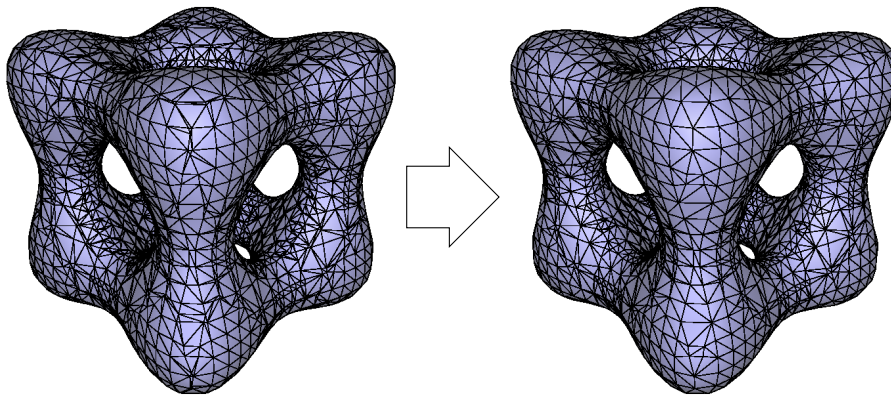


Figura 4.15: Eliminação de *slivers*.

Uma observação útil a ser feita é que quanto maior for a tolerância  $\epsilon$ , mais *slivers* deixam de ser formados. O maior valor possível para  $\epsilon$  é de 50% do tamanho total da aresta, ao se usar este valor todas as arestas por onde a superfície passa são alteradas, de forma que um de seus vértices é movido para a superfície. A Figura 4.16 mostra um exemplo desta forma de se eliminar *slivers*, compare com a Figura 4.15.

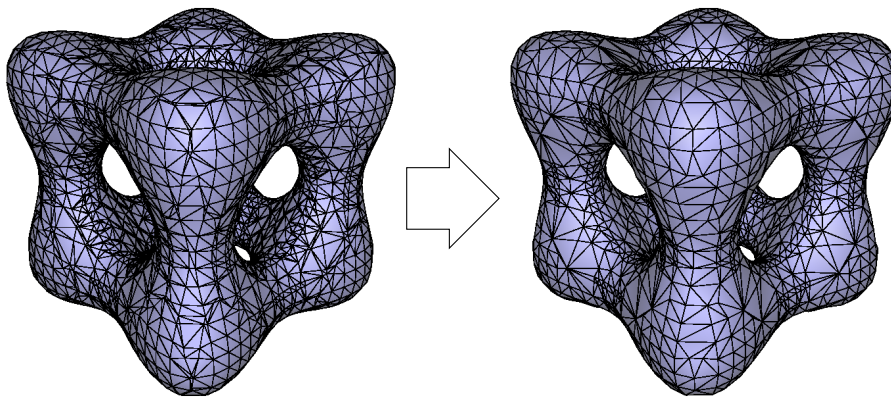


Figura 4.16: Eliminação de *slivers* com a maior tolerância possível.

Apesar de diminuir o número de *slivers*, em alguns casos esta alteração nos vértices das arestas pode formar triângulos sobressalentes, e a triangulação deixa de representar corretamente a topologia da superfície. Nestes casos deve-se buscar uma tolerância menor, para evitar resultados indesejáveis.

### 4.2.2 Flip de arestas

A forma de eliminação de *slivers* vista possui uma desvantagem de formar em alguns casos triângulos de baixa qualidade, ou mesmo degenerados. É preciso tratar estes casos antes de tentar refinar a malha. Para isto foi utilizado um algoritmo de *flip* de arestas, que, ao ser efetuado em arestas de triângulos mal-formados aumenta a qualidade destes triângulos. Dada uma aresta da triangulação, a operação de *flip* nesta aresta consiste em se trocar os vértices da aresta pelos outros vértices dos triângulos adjacentes a ela, conforme ilustra a Figura 4.17.

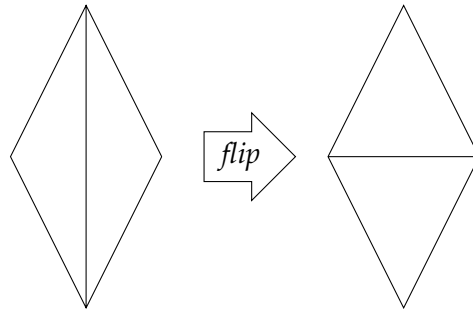


Figura 4.17: Operação de *flip* de aresta.

Os algoritmos de *flip* de arestas executam diversos *flips* até que uma determinada condição seja satisfeita. Um critério que pode ser utilizado é o de Delaunay (mais informações em [2]). A Figura 4.18 ilustra o resultado da aplicação deste algoritmo em uma aproximação da esfera  $x^2 + y^2 + z^2 - 1 = 0$ .

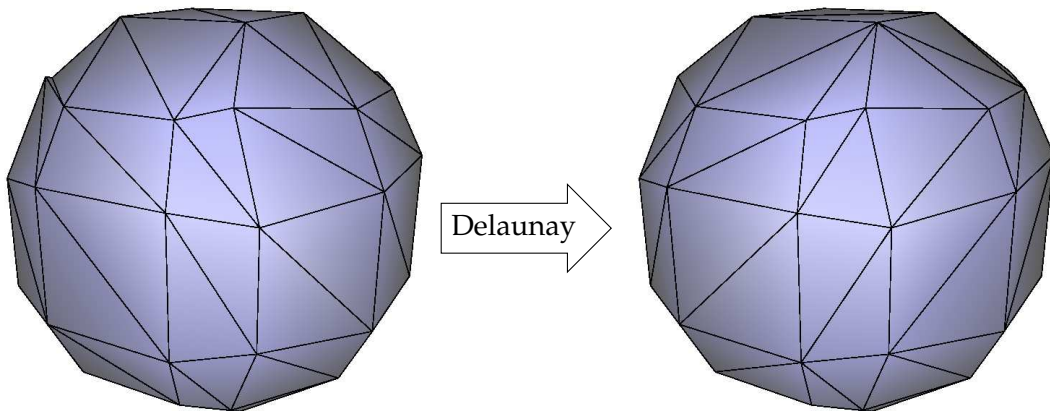


Figura 4.18: Aplicação do algoritmo de Delaunay.

### 4.2.3 Refinamento da malha

Para tornar a malha suave é preciso aplicar algum método de subdivisão, isto foi feito da seguinte forma: dada uma aresta da triangulação a ser subdividida, calcula-se seu ponto médio  $p$ , que é utilizado como primeira aproximação no método de Newton, que encontra um ponto sobre a superfície. Os triângulos adjacentes a esta aresta também são subdivididos, conforme ilustra a Figura 4.19



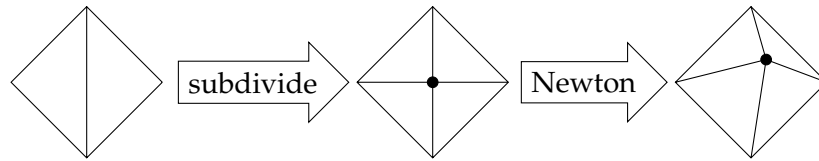


Figura 4.19: Método de subdivisão.

Para subdividir toda a triangulação, aplica-se este esquema de subdivisão a todas as arestas que satisfizerem uma determinada condição, repetindo o processo até que nenhuma aresta satisfaça esta condição. Neste trabalho foi utilizada uma verificação do tamanho da aresta. Uma aresta é subdividida se seu tamanho for superior a um determinado valor. Melhores resultados (em relação à qualidade dos triângulos formados) são obtidos quando se dá prioridade às arestas de maior comprimento. A Figura 4.20 ilustra o resultado desta operação aplicada à aproximação da esfera.

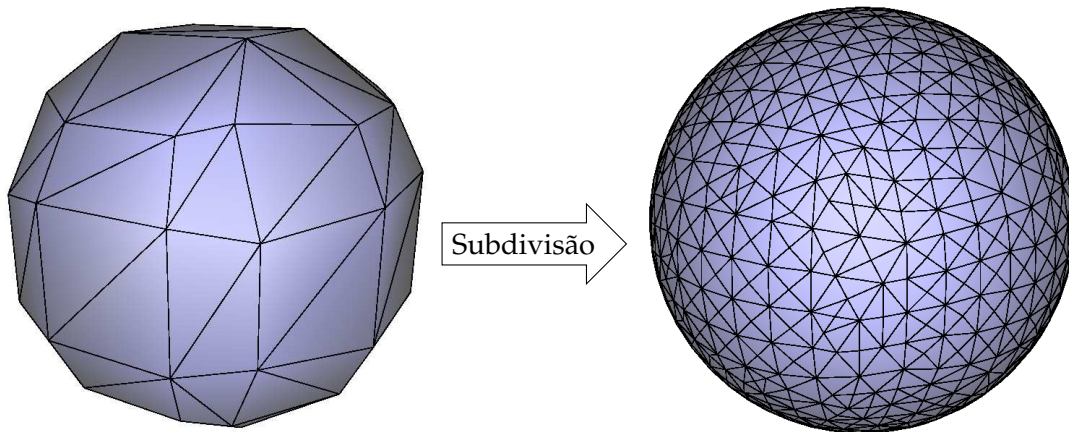


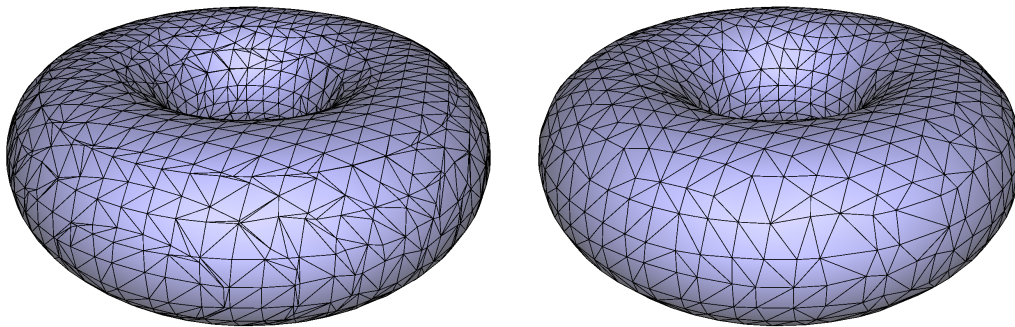
Figura 4.20: Subdivisão aplicada à aproximação da esfera.

#### 4.2.4 Resultados

O método de aproximação de superfícies descrito neste capítulo foi implementado com a linguagem C++. As Figuras 4.21 até 4.24 ilustram resultados obtidos pela aplicação do método de extração de superfícies implícitas. A parte (a) mostra a aproximação sem eliminação de *slivers* nem *flip* de arestas. Em (b) são aplicadas a eliminação de *slivers* e *flip* de arestas, subdividindo a malha em (a) obtém-se (c). E (d) é o resultado da subdivisão aplicada à malha em (b).

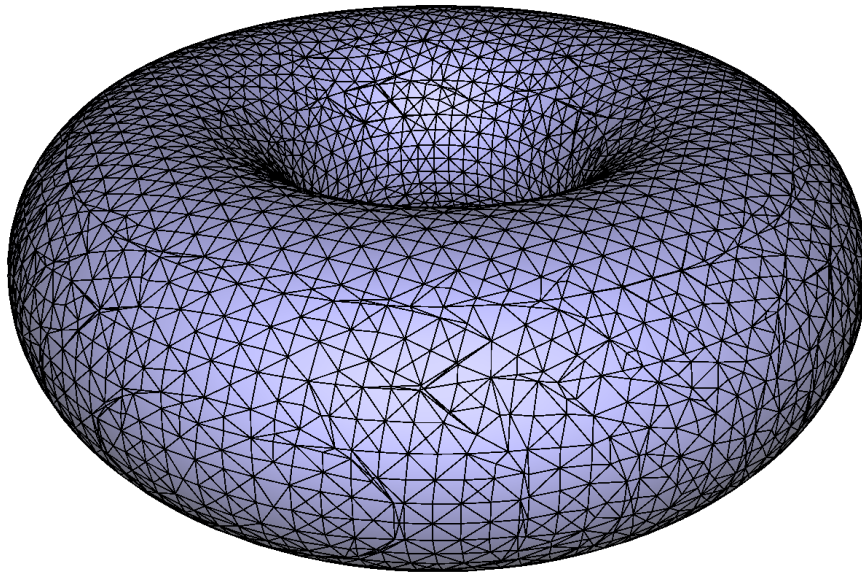
Pode-se fazer uma análise das aproximações de acordo com a qualidade dos triângulos encontrados. Para isto é preciso utilizar uma métrica que classifique os triângulos, como, por exemplo, por meio da razão entre a área do círculo inscrito e a do circuncírculo a cada triângulo. Esta razão vale no mínimo zero (em triângulos degenerados), e possui um valor máximo quando o triângulo é equilátero, podendo ser normalizada para que os valores estejam entre 0 e 1. Utilizando esta métrica, foram gerados os histogramas das Figuras 4.25 até 4.28, que ilustram a distribuição da qualidade dos triângulos das malhas das Figuras 4.21 até 4.24, respectivamente. Observa-se que a operação para eliminação de *slivers*, juntamente com o *flip* de arestas aumenta a qualidade dos triângulos, o que facilita o uso destas malhas triangulares em outras possíveis aplicações.



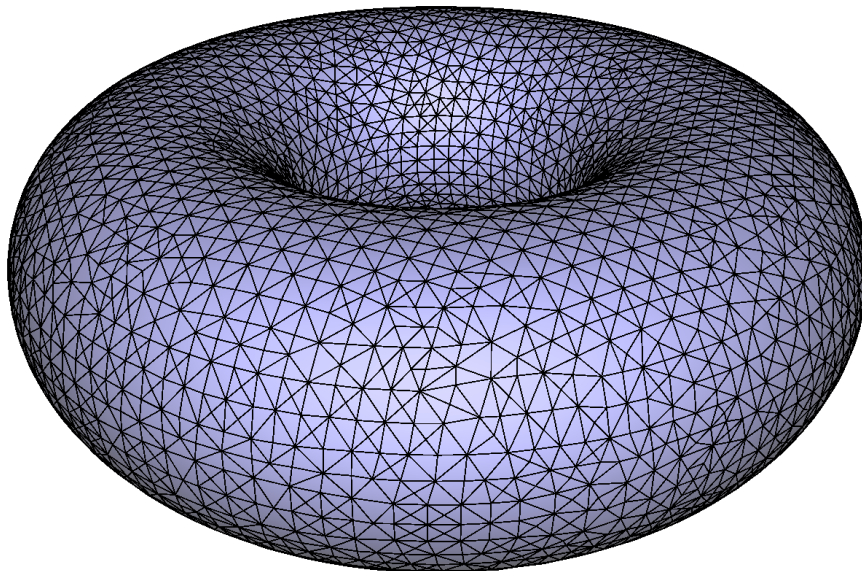


(a)

(b)



(c)



(d)

Figura 4.21: Extração da superfície implícita  $(x^2 + y^2 + z^2 + R^2 - r^2)^2 - 4R^2(x^2 + y^2) = 0$ , com  $R = 4$  e  $r = 2$ .

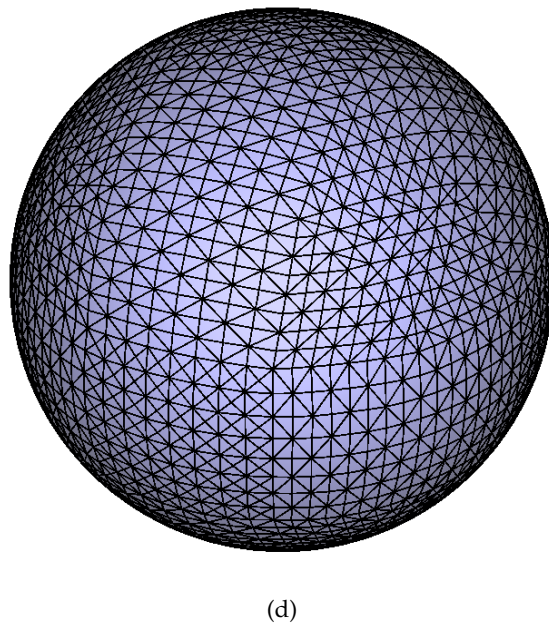
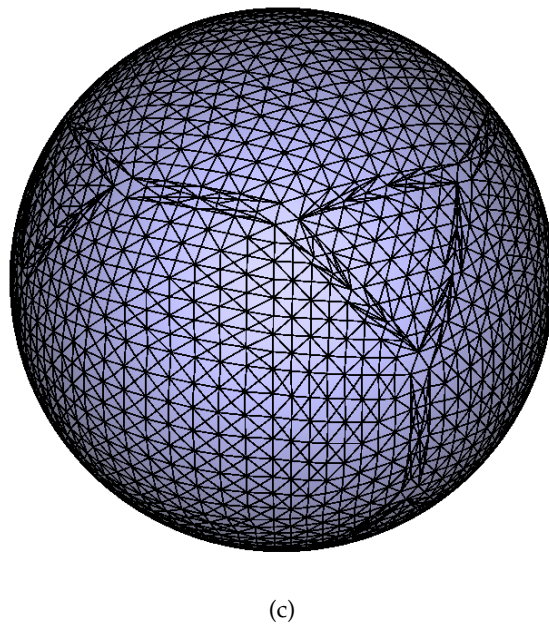
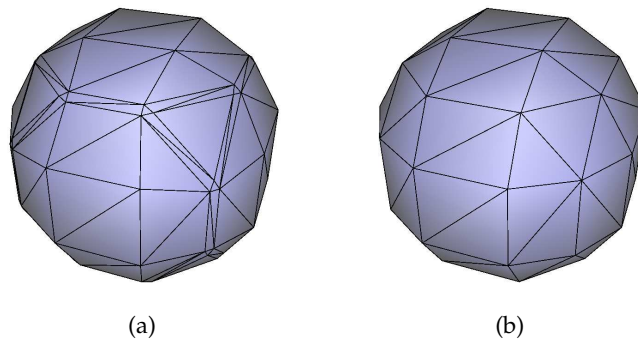
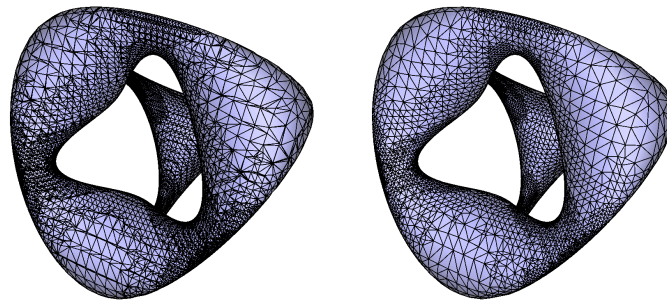
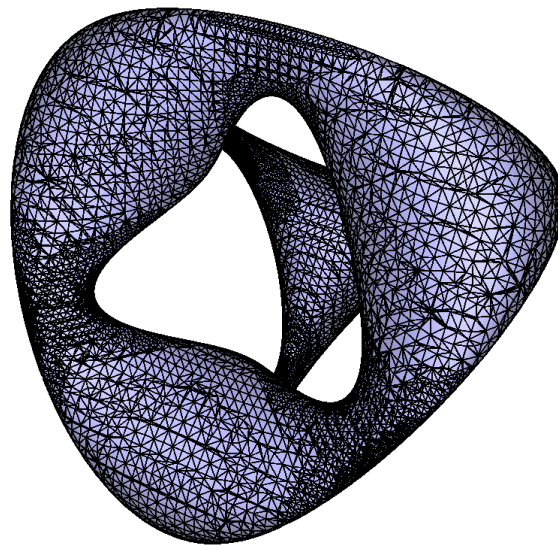


Figura 4.22: Extração da superfície implícita  $x^2 + y^2 + z^2 - 1 = 0$ .

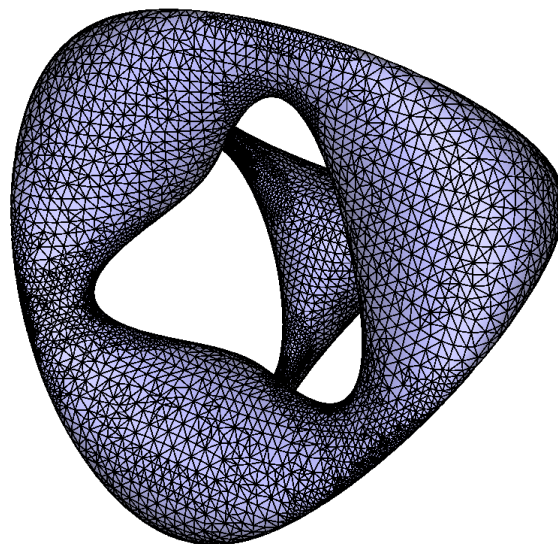


(a)

(b)



(c)



(d)

Figura 4.23: Extração da superfície implícita  $(x^2 + y^2 + z^2 - 23.75)^2 - 0.8((z - 5)^2 - 2x^2)((z + 5)^2 - 2y^2) = 0$ .



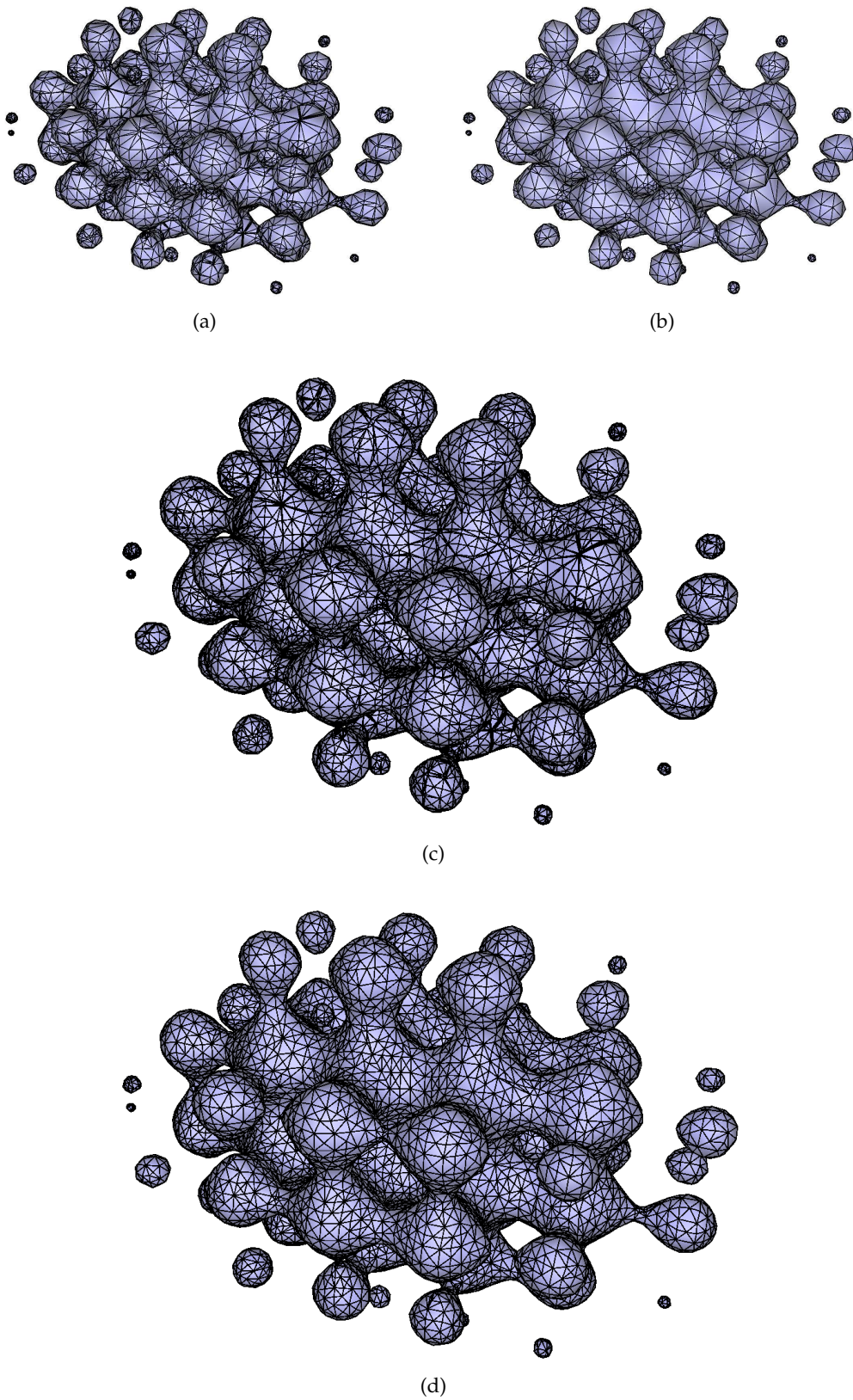


Figura 4.24: Extração da superfície implícita  $0.4(\sin(5x) + \sin(5y) + \sin(5z)) + 0.1x^2 + 0.3y^2 + 0.2z^2 - 0.5 = 0$ .

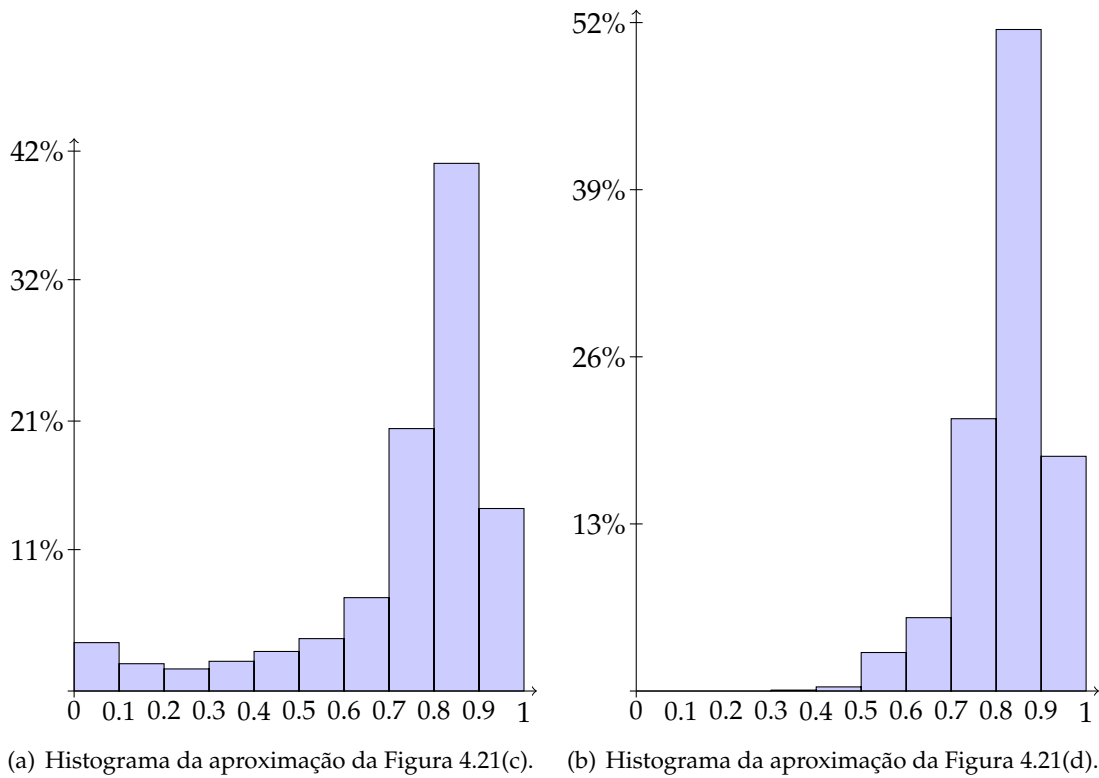


Figura 4.25: Histogramas indicando a qualidade dos triângulos nas aproximações da superfície  $(x^2 + y^2 + z^2 + R^2 - r^2)^2 - 4R^2(x^2 + y^2) = 0$ .

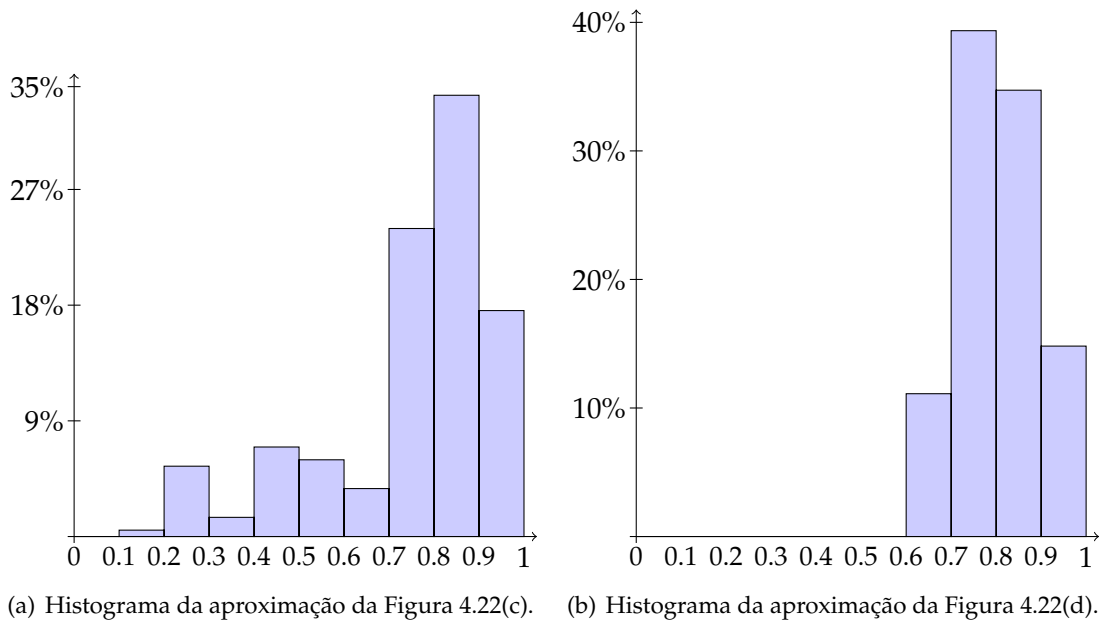


Figura 4.26: Histogramas indicando a qualidade dos triângulos nas aproximações da superfície  $x^2 + y^2 + z^2 - 1 = 0$ .

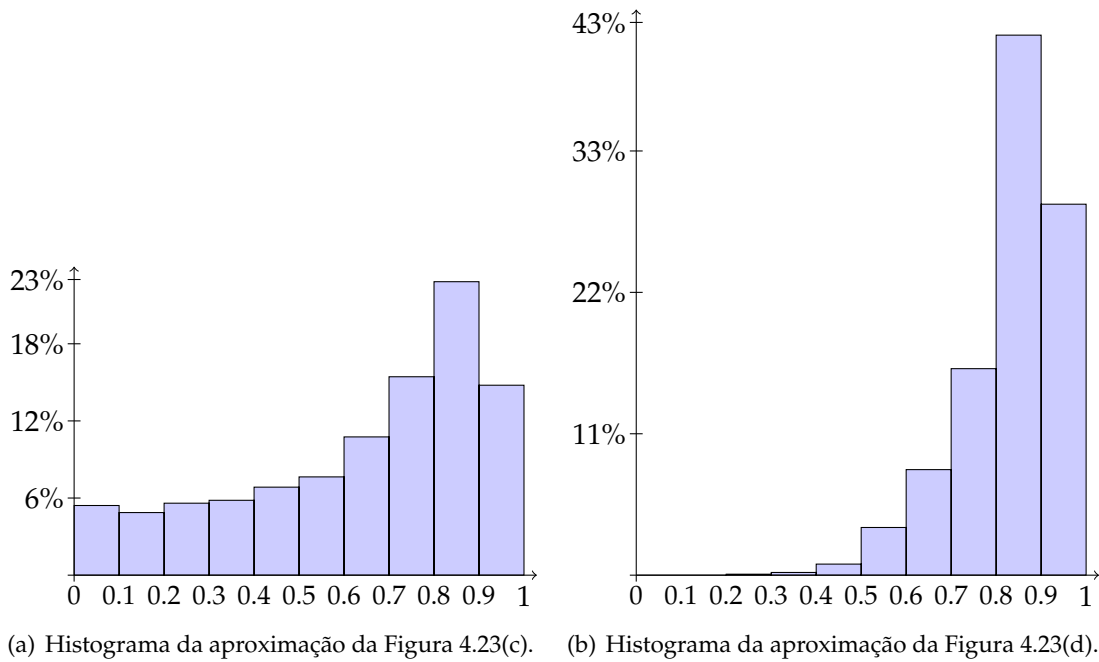


Figura 4.27: Histogramas indicando a qualidade dos triângulos nas aproximações da superfície  $(x^2 + y^2 + z^2 - 23.75)^2 - 0.8((z - 5)^2 - 2x^2)((z + 5)^2 - 2y^2) = 0$ .

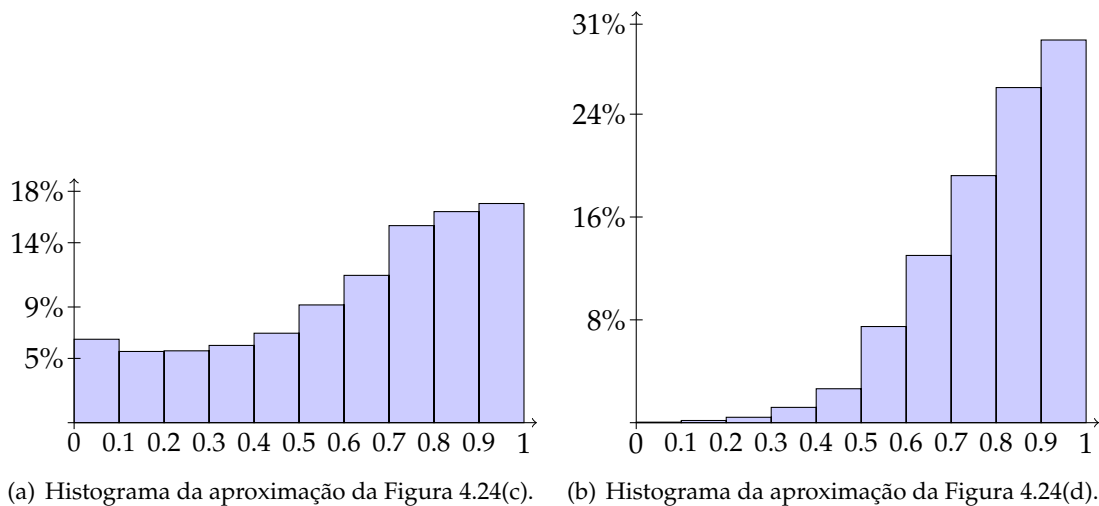


Figura 4.28: Histogramas indicando a qualidade dos triângulos nas aproximações da superfície  $0.4(\text{sen}(5x) + \text{sen}(5y) + \text{sen}(5z)) + 0.1x^2 + 0.3y^2 + 0.2z^2 - 0.5 = 0$ .

## Conclusão e trabalhos futuros

ESTE TRABALHO discutiu definições e diversas formas de representação de curvas planas e superfícies. Observou-se que, a partir da definição paramétrica, torna-se de certa forma simples a representação da curva ou superfície. Porém, quando a definição é implícita esta tarefa se torna mais laboriosa. Foram analisados diversos métodos cujo objetivo é a visualização ou extração de estruturas representando curvas e superfícies implícitas.

Apresentou-se também um método de extração de curvas e superfícies implícitas que garante que o resultado seja uma aproximação isotópica e que mantém um determinado nível de suavidade. Para manter a topologia, o plano (respectivamente espaço) foi subdividido formando um *Quadtree* (respectivamente *Octree*) de acordo com o método descrito em [15]. Para evitar problemas de descontinuidade, trabalhou-se com estruturas duais ao *Quadtree* e ao *Octree*, de onde pode-se tirar uma primeira aproximação das curvas e superfícies, respectivamente. Apesar de manter a topologia, esta primeira aproximação precisa ser tratada para que sua geometria seja mais próxima da geometria da curva ou superfície. Foram utilizados alguns métodos para subdividir estas aproximações de forma a aumentar a suavidade dos resultados finais, melhorando assim a geometria. No caso das curvas planas, isto foi feito subdividindo-se as arestas de acordo com seus comprimentos ou pelo ângulo entre as arestas. Para o caso das superfícies, cujo resultado obtido é uma malha triangular, a suavização é obtida através de diversas subdivisões nas arestas e triângulos, realizadas de acordo com o comprimento das arestas. Foi proposta também uma forma de aumentar a qualidade dos triângulos gerados, através de alterações na estrutura de onde são extraídos os triângulos, juntamente com uma operação de *flip* de arestas. Estas operações aumentaram bastante a qualidade das malhas triangulares, conforme pôde ser constatado pelas figuras e histogramas no Capítulo 4.

Entre possíveis trabalhos futuros destacam-se:

- Adaptar o algoritmo para encontrar aproximações para curvas em 3 dimensões.
- Utilizar uma aritmética mais precisa para se trabalhar com intervalos, como por exemplo a aritmética afim, uma vez que a precisão da aritmética intervalar tradicional tende a diminuir ao serem avaliadas expressões maiores. A aritmética afim também pode ser útil para verificar se as células do *DualQuadtree* e do *DualOctree* satisfazem as condições que garantem a aproximação isotópica.
- Utilizar, como critério de subdivisão nas aproximações de superfícies, a curvatura da superfície, aproximada possivelmente por algum operador diferencial discreto.

Isto tornaria as aproximações mais adaptativas, subdividindo apenas onde for necessário, ainda mantendo a suavidade.

- Aplicar outro método mais eficiente e seguro para eliminação de *slivers*. Eficiente no sentido de eliminar o maior número possível de triângulos deste tipo. E seguro no sentido de manter a topologia em qualquer circunstância.
- Estudar outras aplicações para as aproximações encontradas.



# Referências Bibliográficas

- [1] H. Alencar and W. Santos. *Geometria Diferencial das Curvas Planas*. 2002.
- [2] A. I. Bobenko and B. A. Springborn. A discrete laplace-beltrami operator for simplicial surfaces, Feb 2006.
- [3] J.-D. Boissonnat, D. Cohen-Steiner, and G. Vegter. Isotopic implicit surface meshing. In *Proc. 36 th ACM Symp. on the Theory of Computing*, pages 301–309, 2004.
- [4] J.-D. Boissonnat and M. Teillaud, editors. *Effective Computational Geometry for Curves and Surfaces*. Springer-Verlag, Mathematics and Visualization, 2006.
- [5] A. de Cusatis Junior, L. H. de Figueiredo, and M. Gattass. Interval methods for ray casting implicit surfaces with affine arithmetic. In *SIBGRAPI '99: Proceedings of the XII Brazilian Symposium on Computer Graphics and Image Processing*, pages 65–72, Washington, DC, USA, 1999. IEEE Computer Society.
- [6] L. H. de Figueiredo and J. Stolfi. *Self-Validated Numerical Methods and Applications*. Brazilian Mathematics Colloquium monographs. IMPAÇNPq, Rio de Janeiro, Brazil, 1997.
- [7] M. P. Do-Carmo. *Differential Geometry of Curves and Surfaces*. Prentice Hall, February 1976.
- [8] L. H. Figueiredo and J. Gomes. Sampling implicit objects with physically-based particle systems. *Computer & Graphics*, 20(3), 1996.
- [9] E. Galin and S. Akkouche. Fast surface reconstruction from contours using implicit surfaces. In *Implicit Surfaces '98 proceedings*, pages 139–144, 1998.
- [10] M. Iri, K. Tanabe, K. Academic, A. Griewank, A. Griewank, and A. Griewank. On automatic differentiation. In *in Mathematical Programming: Recent Developments and Applications*, pages 83–108. Kluwer Academic Publishers, 1989.
- [11] H. Kagiwada, R. Kalaba, N. Rasakhoo, and S. Karl. *Numerical Derivatives and Nonlinear Analysis*, volume 31 of *Mathematical Concepts and Methods in Science and Engineering*. Plenum Press, Inc., New York, NY, USA, 1985.
- [12] T. Lewiner, H. Lopes, A. W. Vieira, and G. Tavares. Efficient implementation of marching cubes' cases with topological guarantees. *journal of graphics tools*, 8(2):1–15, 2003.

- [13] H. Lopes, J. B. S. de Oliveira, and L. H. de Figueiredo. Robust adaptive polygonal approximation of implicit curves. *Computers & Graphics*, 26(6):841–852, 2002.
- [14] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, volume 21, pages 163–169, New York, NY, USA, July 1987. ACM Press.
- [15] S. Plantinga and G. Vegter. Isotopic approximation of implicit curves and surfaces. In *Symposium on Geometry Processing*, pages 251–260, 2004.
- [16] S. Schaefer and J. Warren. Dual marching cubes: Primal contouring of dual grids. In *PG '04: Proceedings of the Computer Graphics and Applications, 12th Pacific Conference on PG '04*, pages 70–76, 2004.
- [17] G. Taubin. Distance approximations for rastering implicit curves. *ACM Transactions on Graphics*, 13(1):3–42, January 1994.
- [18] R. E. Wengert. A simple automatic derivative evaluation program. *Commun. ACM*, 7(8):463–464, 1964.