2
Aproximação por curvas implícitas e partição da unidade

Este capítulo expõe alguns conceitos básicos necessários para o entendimento deste trabalho.

2.1
Curvas Algébricas

Um subconjunto $\mathcal{O} \subset \mathbb{R}^2$ é chamado de uma *curva implícita* se existe uma função $F : U \to \mathbb{R}$, $\mathcal{O} \subset U$, e $c \in \mathbb{R}$ tal que $\mathcal{O} = F^{-1}(c)$.

Uma curva implícita $F^{-1}(c)$ é *regular* se F é diferenciável e satisfaz a condição que em cada ponto $\mathbf{x} \in F^{-1}(c)$ o gradiente de F em $\mathbf{x}$ não se anula.

Um *polinômio de grau d* definido no plano é uma função $P_d : \mathbb{R}^2 \to \mathbb{R}$ especificada pela seguinte expressão:

$$P_d(x, y) = \sum_{0 \leq j+k \leq d} a_{j,k} x^j y^k. \tag{2-1}$$

Dá-se o nome de *curva algébrica de grau d* ao conjunto $P_d^{-1}(0)$.

Como a curva algébrica é o objeto matemático mais utilizado nessa dissertação, é conveniente adotar uma representação adequada para P_d, que é a mesma adotada por Tasdizen et al. em [20]:

$$P_d(x, y) = \mathbf{v}^t \mathbf{a}, \tag{2-2}$$

onde

$$\mathbf{a} = \begin{bmatrix} a_{0,0} & a_{1,0} & \ldots & a_{d,0} & a_{d-1,1} & \ldots & a_{0,1} & \ldots & a_{0,d} \end{bmatrix}^t$$

e

$$\mathbf{v} = \begin{bmatrix} 1 & x & \ldots & x^d & x^{d-1}y & \ldots & y & \ldots & y^d \end{bmatrix}^t.$$

O vetor $\mathbf{a} \in \mathbb{R}^l$ representa os coeficientes $(a_{j,k})_{0 \leq j; 0 \leq k; 0 \leq j+k \leq d}$ e o vetor $\mathbf{v}$ representa os monômios do polinômio P_d. Como P_d possui grau d é fácil verificar que o número de termos que ele possui (que corresponde à dimensão de $\mathbf{a}$ ou à dimensão de $\mathbf{v}$) é exatamente $l = \frac{(d+1)(d+2)}{2}$.

PUC-Rio - Certificação Digital Nº 0410413/CA

Coeficientes do Polinômio P_d	
Grau d da curva $\mathcal{C}$	Número de coeficientes l do polinômio P_d
$d = 2$	$l = \frac{(2+1)(2+2)}{2} = 6$
$d = 3$	$l = \frac{(3+1)(3+2)}{2} = 10$
$d = 4$	$l = \frac{(4+1)(4+2)}{2} = 15$
$d = 5$	$l = \frac{(5+1)(5+2)}{2} = 21$
$d = 6$	$l = \frac{(6+1)(6+2)}{2} = 28$
$d = 7$	$l = \frac{(7+1)(7+2)}{2} = 36$
$d = 8$	$l = \frac{(8+1)(8+2)}{2} = 45$
$d = 9$	$l = \frac{(9+1)(9+2)}{2} = 55$
$d = 10$	$l = \frac{(10+1)(10+2)}{2} = 66$

Tabela 2.1: Tabela do número de coeficientes do polinômio

2.2 Aproximações algébricas por mínimos quadrados

Considere os dados de entrada como sendo um conjunto de pontos $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_q\} \in \mathbb{R}^2$ amostrados de uma curva planar $\mathcal{C}$, onde $\mathbf{p}_i = (x_i, y_i)$. Considere também o conjunto de vetores normais unitários correspondentes $\mathcal{N} = \{\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_q\}$.

Apresenta-se agora diversos métodos de como obter uma curva que aproxima $\mathcal{C}$.

2.2.1 Minimizando a distância algébrica

Uma maneira simples de se obter uma curva algébrica $P_d^{-1}(0)$ que aproxima a curva $\mathcal{C}$ é o de minimizar a distância algébrica sobre o conjunto de pontos dados, isto é, determinar os coeficientes de $P_d^{-1}(0)$ tal que a soma da distância algébrica de cada um dos q pontos à curva, denotada por

$$e_{total} = \sum_{i=1}^{q} (P_d(x_i, y_i))^2 \tag{2-3}$$

seja a menor possível.

Utilizando a representação vetorial de P_d em (2-2), e_{total} pode ser escrito como:

$$e_{total} = \mathbf{a}^t (\sum_{i=1}^{q} \mathbf{v}_i \mathbf{v}_i^t) \mathbf{a}, \tag{2-4}$$

onde

$$\mathbf{v}_i = \begin{bmatrix} 1 & x_i & \ldots & x_i^d & x_i^{d-1} y_i & \ldots & y & \ldots & y_i^d \end{bmatrix}^t.$$

Para melhorar a notação, defina as matrizes $\mathbf{M}$ de tamanho $l \times q$ e $\mathbf{S}$ de

PUC-Rio - Certificação Digital Nº 0410413/CA

tamanho $l \times l$ da seguinte maneira:

$$\mathbf{M} = \left[\begin{array}{cccc} \mathbf{v}_1 & \mathbf{v}_2 & ... & \mathbf{v}_q \end{array} \right],$$

e

$$\mathbf{S} = \mathbf{M}\mathbf{M}^t = \sum_{i=1}^{q} \mathbf{v}_i \mathbf{v}_i^t.$$

Considere o problema de minimização como sendo

$$min_{\mathbf{a}}\{\mathbf{a}^t\mathbf{S}\mathbf{a}\},$$

sujeito a restrição $\|\mathbf{a}\|^2 = 1$.

Com o uso do multiplicador de Lagrange λ, o problema de otimização condicionado torna-se:

$$min_{\mathbf{a}}\{\mathbf{a}^t\mathbf{S}\mathbf{a} + \lambda(\mathbf{a}^t\mathbf{a} - 1)\}. \quad (2\text{-}5)$$

A solução desse problema é dada pelo autovetor unitário de S associado ao seu menor autovalor [21].

Em resumo, o método para a minimização da distância algébrica restrito à condição $\|\mathbf{a}\|^2 = 1$ consiste simplesmente em encontrar o autovetor de $\mathbf{S}$ associado ao seu menor autovalor em módulo usando, por exemplo, o método da potência em $\mathbf{S}^{-1}$ [18]. Esse método será chamado de *classical least squares*.

Embora este método seja invariante por transformações afins [21], ele apresenta alguns problemas. Seus resultados são extremamente sensíveis às pequenas pertubações nos dados de entrada. Além disso, a curva algébrica $P_d^{-1}(0)$ obtida na aproximação não leva em consideração a continuidade do conjunto de dados de entrada, podendo gerar novas componentes conexas ou juntar componentes que eram originalmente separadas.

Uma técnica simples para tornar a aproximação por esse método mais estável é aplicar uma padronização nos conjuntos de dados, que consiste em colocar a origem do domínio no centróide do conjunto de dados e então escalonar os pontos dividindo-os pela média dos autovalores da matriz $\mathbf{S}$.

Para maiores detalhes sobre esse método consulte [19, 20, 21].

2.2.2 Aproximação considerando o gradiente

Para evitar os problemas de inconsistência de continuidade e de sensibilidade à pequenas pertubações do conjunto de dados de entrada, foi proposto um outro método que considera os vetores unitários normais à curva [20]. Esses

PUC-Rio - Certificação Digital Nº 0410413/CA

vetores são fornecidos como dados de entrada do sistema através do conjunto $\mathcal{N}$.

Define-se o conjunto $\mathcal{T} = \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_q\}$ como sendo o conjunto dos vetores unitários tangentes à curva $\mathcal{C}$, onde $\mathbf{t}_i$ é obtido aplicando uma rotação de $\pi/2$ no sentido anti-horário ao vetor $\mathbf{n}_i$.

O método *Gradient One Fitting* a ser discutido foi proposto por Lei et al. em [13]. Ele faz uso do gradiente do polinômio P_d em (x_i, y_i), que é denotado por

$$\nabla P_d(x_i, y_i) = \begin{bmatrix} \frac{\partial P_d}{\partial x}(x_i, y_i) \\ \frac{\partial P_d}{\partial y}(x_i, y_i) \end{bmatrix}, \tag{2-6}$$

Vale lembrar que se o vetor gradiente de P_d em (x_i, y_i) não é nulo, então ele é perpendicular ao vetor tangente da curva de nível de P_d que passa por (x_i, y_i).

A proposta do método proposto por Lei et al. é novamente um problema de mínimos quadrados. Ele adiciona dois termos novos à função objetivo e retira a condição $\|\mathbf{a}\|^2 = 1$. Esses novos termos são funções do gradiente do polinômio P_d. O primeiro deles é $\sum_{i=1}^{q}(\mathbf{n}_i^t \nabla P_d(x_i, y_i) - 1)^2$, cujo objetivo é o de ajustar P_d de tal forma que o seu gradiente em (x_i, y_i) tenda a ficar alinhado com $\mathbf{n}_i$. O segundo termo é $\sum_{i=1}^{q}(\mathbf{t}_i^t \nabla P_d(x_i, y_i))^2$, que tem por objetivo forçar ainda mais o ajuste fazendo com que o vetor gradiente de P_d em (x_i, y_i) tenda a ficar perpendicular à $\mathbf{t}_i$. A função a ser minimizada fica, portanto, definida da seguinte forma:

$$e_{grad} = \sum_{i=1}^{q}\{(P_d(x_i, y_i))^2 + \mu(\mathbf{n}_i^t \nabla P_d(x_i, y_i) - 1)^2 + \mu(\mathbf{t}_i^t \nabla P_d(x_i, y_i))^2\} \tag{2-7}$$

onde μ é o peso que se dá aos dois novos termos.

Com o objetivo de seguir a representação vetorial para o polinômio P_d, as seguintes matrizes e vetores são definidas:

– A matriz $\mathbf{D}_i$ de tamanho $l \times 2$:

$$\mathbf{D}_i = \begin{bmatrix} \frac{\partial \mathbf{v}_i}{\partial x} & \frac{\partial \mathbf{v}_i}{\partial y} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 2x_i & 0 \\ \vdots & \vdots \\ dx_i^{d-1} & 0 \\ (d-1)x_i^{d-2} \cdot y_i & x_i^{d-1} \\ \vdots & \vdots \\ 0 & 1 \\ \vdots & \vdots \\ y_i^{d-1} & (d-1)x_i \cdot y_i^{d-2} \\ 0 & d \cdot y_i^{d-1} \end{bmatrix}. \tag{2-8}$$

PUC-Rio - Certificação Digital Nº 0410413/CA

– O vetor gradiente ∇P_d:

$$\nabla P_d = \nabla(\mathbf{v}_i^t \mathbf{a}) = (\mathbf{D}_i)^t \mathbf{a}. \tag{2-9}$$

– A matriz $\mathbf{S_N}$ de tamanho $l \times l$:

$$\mathbf{S_N} = \sum_{i=1}^{q} \mathbf{D}_i \mathbf{n}_i \mathbf{n}_i^t \mathbf{D}_i^t.$$

– A matriz $\mathbf{S_T}$ de tamanho $l \times l$:

$$\mathbf{S_T} = \sum_{i=1}^{q} \mathbf{D}_i \mathbf{t}_i \mathbf{t}_i^t \mathbf{D}_i^t.$$

– O vetor $\mathbf{g_N}$ de tamanho l:

$$\mathbf{g_N} = \sum_{i=1}^{q} \mathbf{D}_i \mathbf{n}_i.$$

O problema de otimização para o método é:

$$min_{\mathbf{a}}\{e_{grad}\} = min_{\mathbf{a}}\{\mathbf{a}^t\mathbf{S}\mathbf{a} + \mu\mathbf{a}^t\mathbf{S_N}\mathbf{a} + \mu\mathbf{a}^t\mathbf{S_T}\mathbf{a} - 2\mu\mathbf{a}^t\mathbf{g_N} + \mu q\}$$

E a solução desse problema é obtido resolvendo o seguinte sistema de equações lineares:

$$(\mathbf{S} + \mu(\mathbf{S_N} + \mathbf{S_T}))\mathbf{a} = \mu\mathbf{g_N}. \tag{2-10}$$

2.2.3 Técnica para reduzir a instabilidade numérica

Quando a matriz $\mathbf{SS} = (\mathbf{S} + \mu(\mathbf{S_N} + \mathbf{S_T}))$ não tiver posto máximo ou mostrar instabilidade numérica para resolver o sistema (2-10) pode-se usar a técnica denominada *ridge regression*, e será denotada por RR. Essa técnica é normalmente usada pelos estatísticos para remover a colinearidade dos dados de entrada. A primeira proposta para obter melhores curvas algébricas usando essa técnica foi feita por Tasdizen et al. em [20].

A técnica RR basicamente modifica o problema de otimização adicionando um novo termo:

$$min_{\mathbf{a}}\{\mathbf{a}^t\mathbf{S}\mathbf{a} + \mu\mathbf{a}^t\mathbf{S_N}\mathbf{a} + \mu\mathbf{a}^t\mathbf{S_T}\mathbf{a} - 2\mu\mathbf{a}^t\mathbf{g_N} + \mu q + \kappa\mathbf{a}^t\mathbf{\Delta}\mathbf{a}\},$$

Onde $\mathbf{\Delta}$ é uma matriz diagonal de tamanho $l \times l$ e a constante real κ determina o peso que se dá ao novo termo na aproximação.

A solução para esse problema é obtido da seguinte forma:

PUC-Rio - Certificação Digital Nº 0410413/CA

$$\mathbf{a} = \mu(\mathbf{SS} + \kappa\mathbf{\Delta})^{-1}\mathbf{g_N} \tag{2-11}$$

Tasdizen et al. em [20] sugerem a seguinte escolha para matriz $\mathbf{\Delta}$.

$$\mathbf{\Delta}_{vv} = \frac{i!j!}{(i+j)!} \sum_{k,l\geq 0; k+l=i+j} \frac{(k+l)!}{k!l!} \sum_{m=1}^{q} x_m^{2k} y_m^{2l}, \tag{2-12}$$

para $i, j \geq 0$; $i + j \leq d$, onde $v = j + \frac{(i+j+1)(i+j)}{2}$.

Essa sugestão para $\mathbf{\Delta}$ preserva a invariância por rotação do método descrito na seção 2.2.2 [20].

A seguir, a figura 2.1 [20] mostra uma comparação entre os três métodos apresentados. As linhas pontilhadas são os dados de entrada (pontos) e as cheias são as aproximações globais dos dados.

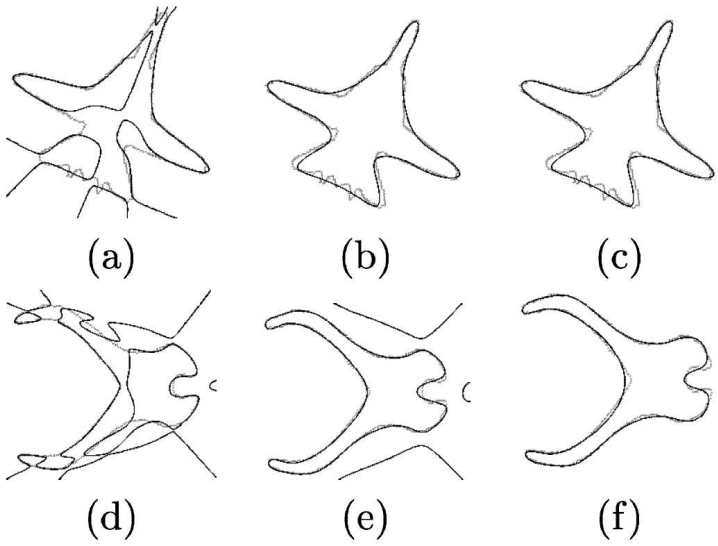

Figura 2.1: As figuras (a) e (d) ilustram aproximações obtidas usando o método *classical least squares*. Já nas figuras (b) e (e) foi utilizado o método *gradient one fitting*, e nas figuras (c) e (f) o método *ridge regression*. As linhas pontilhadas são os dados de entrada e as linhas cheias são as aproximações obtidas. O grau do polinômio foi escolhido como sendo 6 para o alicate e 8 para o avião.

2.3 Partição da Unidade

A *Partição da Unidade* (PU) [5, 10, 11] é uma ferramenta matemática muito útil para combinar aproximações locais a fim de definir uma aproximação global. Importantes propriedades como o erro máximo global e a ordem de convergência podem ser herdadas do comportamento local.

A idéia básica da construção de uma aproximação global por partição da unidade é a seguinte:

PUC-Rio - Certificação Digital Nº 0410413/CA

1. dividir o domínio de interesse em diversas partes,

2. obter uma aproximação local para o subconjunto de dados pertencentes a cada parte,

3. obter uma aproximação global fazendo uma combinação ponderada das soluções locais através de funções suaves não negativas que correspondem aos pesos. Em cada ponto do domínio, a soma dessas funções peso deve ser um.

Mais precisamente, considere um domínio limitado $\Omega \subset \mathbb{R}^2$ e denote por $\{\varphi_i\}$ o conjunto de funções não negativas com suporte compacto tal que:

$$\sum_i \varphi_i(x, y) \equiv 1 \text{ para todo ponto } (x, y) \in \Omega.$$

Chame de $\mathcal{F}_i$ o conjunto de funções com suporte compacto definidas em $supp(\varphi_i)$. Cada função pertencente a $\mathcal{F}_i$ estaria representando uma aproximação local para os pontos do conjunto de entrada $\mathcal{P}$ que pertencem à $supp(\varphi_i)$.

Uma aproximação global para a função $f : \Omega \to \mathbb{R}^2$ poderia ser obtida da seguinte maneira:

$$f(x, y) \approx \sum \varphi_i(x, y) f_i(x, y). \tag{2-13}$$

onde $f_i \in \mathcal{F}_i$.

Considere $\{w_i\}$ um conjunto de funções não-negativas com suporte compacto tal que:

$$\Omega \subset \bigcup_i supp(w_i).$$

As funções partições da unidade φ_i podem ser geradas da seguinte forma:

$$\varphi_i(x, y) = \frac{w_i(x, y)}{\sum_{j=1}^{n} w_j(x, y)} \tag{2-14}$$

A idéia por trás de uma aproximação construída através da partição da unidade pode ser resumida pelas equações (2-13) e (2-14). Essas equações formam a base do algoritmo MPU proposto por Ohtake et al. em [16].

2.4 O método MPU

O método chamado de *Multilevel Partion of Unity* (MPU) foi proposto por Ohtake et al. em [16] originalmente para gerar uma superfície implícita em $\mathbb{R}^3$. Nessa dissertação restringimos o seu uso para curvas implícitas em $\mathbb{R}^2$.

PUC-Rio - Certificação Digital Nº 0410413/CA

Como seu próprio nome diz, o MPU usa a partição da unidade para obter aproximações globais a partir de aproximações locais usando uma estrutura hierárquica para a subdivisão do domínio, que são feitas recursivamente através de uma *Quad-Tree*.

A *Quad-Tree* é um tipo especial de árvore onde todas as células ou são células folha ou têm quatro células filhas. Tendo como utilização principal a representação de uma decomposição espacial adaptativa do domínio através de um processo recursivo.

A figura 2.2 ilustra um exemplo de uma *Quad-Tree* adaptada aos pontos de entrada.

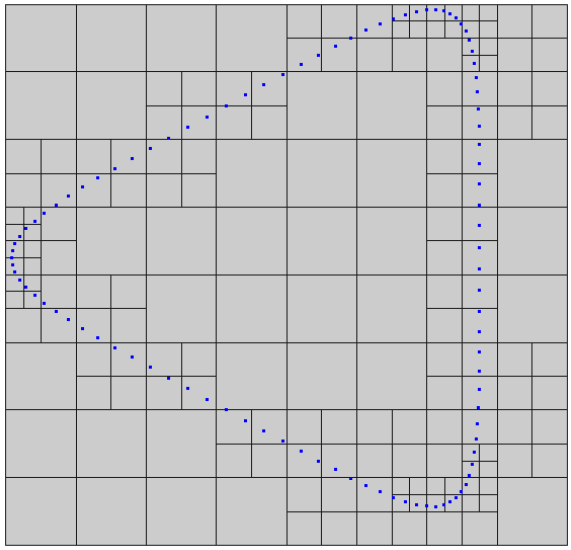

Figura 2.2: Exemplo da construção de uma *Quad-Tree* gerando uma subdivisão do domínio adaptada aos pontos.

PUC-Rio - Certificação Digital Nº 0410413/CA

Segue uma descrição concisa de como o método MPU constrói a função implícita que aproxima globalmente os pontos.

Inicialmente os pontos de $\mathcal{P}$ são transladados de modo a tornar o seu centro de massa a origem do sistema de coordenadas. Depois os pontos são escalonados de tal forma que o quadrado $\Xi = [-1, 1] \times [-1, 1]$ contenha todos eles. Por um abuso de notação, continua-se dando o nome de $\mathcal{P}$ ao conjunto de pontos de entrada após essas duas transformações.

O método cria uma *Quad-Tree* usando um procedimento recursivo controlado pelo erro da aproximação local. O critério de refinamento consiste em identificar localmente o erro cometido pela aproximação e caso ele seja maior que uma determinada tolerância ele subdivide o quadrado em quatro e recursivamente repete o teste para cada um dos seus quadrantes.

Portanto, para cada nó i da *Quad-Tree* existe uma função peso usada na partição da unidade com suporte compacto definido como sendo um círculo de

raio r_i centrado no centro geométrico do nó. Esse raio é escolhido como sendo proporcional ao tamanho da diagonal do quadrado do nó i, denotado por d_i. Geralmente se faz $r_i = 0.75d_i$.

A figura 2.3 ilustra os suportes compactos associados a cada nó da estrutura *Quad-Tree*.

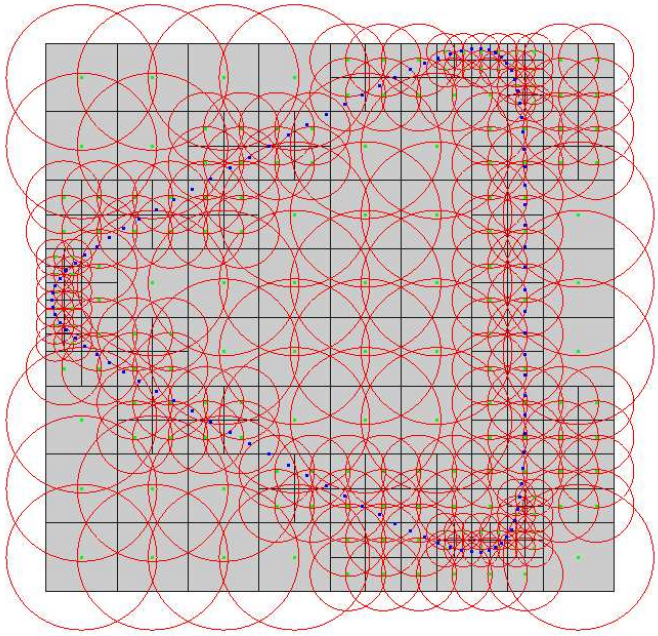

Figura 2.3: Exemplo da região de suporte para cada nó da *Quad-Tree*.

A aproximação local é feita usando uma quádrica e ela só é calculada nos nós que possuirem pontos de $\mathcal{P}$. A função de partição da unidade para cada nó da *Quad-Tree* é uma B-Spline quadrática $b(t)$:

$$\varphi_i(x,y) = b(\frac{3 \mid (x,y) - c_i \mid}{2r_i}) \tag{2-15}$$

onde c_i é centro do quadrado correspondente a um nó da *Quad-Tree*.

Para calcular as quádricas em cada nó, considera-se os pontos de $\mathcal{P}$ que estão na sua região de suporte correspondente. Às vezes (especialmente se a densidade de $\mathcal{P}$ não é uniforme) o círculo de raio r_i de um nó N_i não contém o número de pontos suficiente para uma estimação robusta da quádrica que aproxima os pontos. Se o número de pontos for menor que o número mínimo de pontos suficiente para resolver o problema de mínimos quadrados para a determinação da quádrica, deve-se aumentar o raio do círculo do suporte até que se obtenha a garantia da existência de uma solução (a matriz a ser invertida possuir posto máximo), como ilustra a figura 2.4[16].

Ohtake et al. utilizaram uma função objetivo que é uma média ponderada da distância algébrica de cada ponto à curva, onde a ponderação é feita pela própria função peso φ_i. Mais informações podem ser encontradas em [16].

PUC-Rio - Certificação Digital Nº 0410413/CA

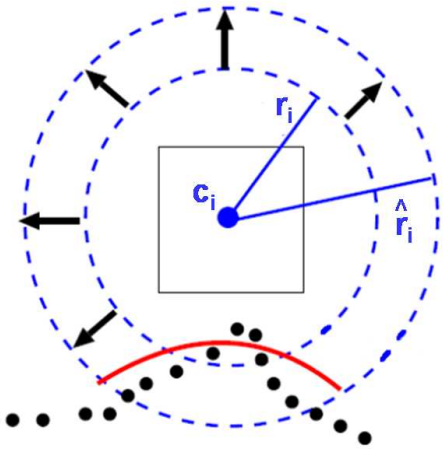

Figura 2.4: Aumentando o raio do círculo do suporte para obter uma garantia de existir solução para o cálculo das quádricas.

As figuras (2.5) e (2.6)[16] exemplificam superfícies implícitas no $\mathbb{R}^3$ obtidas pelo método MPU.

Figura 2.5: Toro: Figura gerada pelo software MPU.

PUC-Rio - Certificação Digital Nº 0410413/CA

PUC-Rio - Certificação Digital Nº 0410413/CA

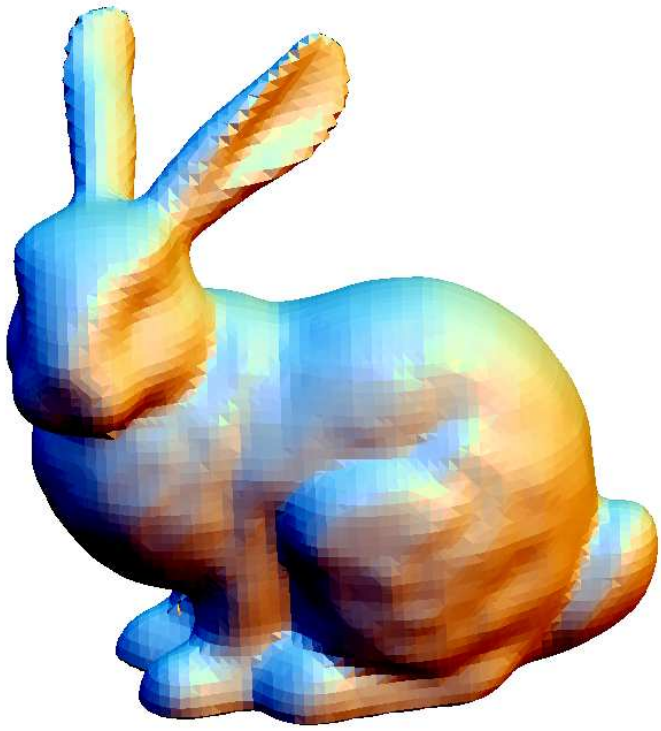

Figura 2.6: Bunny: Figura gerada pelo software MPU.