

Dissertação para obtenção do grau de Mestre em Matemática pelo
**INSTITUTO NACIONAL DE MATEMÁTICA PURA E
APLICADA**

**RASTERIZAÇÃO DE CURVAS IMPLÍCITAS USANDO
APROXIMAÇÕES DE DISTÂNCIA**

por
FRANCISCO PETRÚCIO CAVALCANTE JUNIOR

27 de Fevereiro de 2003

Agradecimentos

Aqui está minha tentativa de agradecer àqueles que de alguma forma contribuíram para a realização deste meu pequeno trabalho. Peço desculpas pelas muitas omissões.

- ao Professor Manfredo do Carmo, meu orientador acadêmico no Mestrado, pelas conversas e pelo estímulo dado durante esse tempo;
- ao Professor Luís Henrique, meu orientador nesta dissertação, pela sugestão do trabalho, pelas ajudas na hora de implementar o algoritmo de Taubin, e sobretudo pela paciência e contínuo bom humor;
- aos professores Luís Velho, Hélio Lopes e Paulo César, membros da banca na defesa dessa dissertação, pelas críticas e sugestões para a melhora do texto e da implementação;
- aos professores, funcionários e colegas, pelo ambiente fraterno e matemático proporcionado por eles;
- aos meus pais, Francisco Petrócio Cavalcante e Maria das Neves de Almeida Cavalcante, por todo incentivo que sempre me deram, desde muito cedo;
- aos amigos do Projeto Universidades Renovadas (www.pur.com.br), que tão bem me acolheram quando cheguei no Rio de Janeiro;
- à comunidade do software livre (em especial a Richard Stallman, criador do projeto GNU, www.gnu.org), que proporcionou as ferramentas necessárias para programar e escrever esse texto (entre as quais, o `gcc` e o `LATEX`);
- a todos os meus amigos e parentes que estão em Alagoas.

Sumário

1	Introdução	1
2	Rasterização de Curvas Implícitas	4
2.1	O algoritmo básico	4
2.2	Aproximação de primeira ordem	6
2.3	Aproximações de ordem superior	10
2.4	Pontos singulares	14
3	Exemplos	17
4	Conclusão	23
A	Equações	24
	Bibliografia	28

Lista de Figuras

1.1	Rasterização de um círculo.	2
2.1	Rasterização utilizando o Algoritmo 2.	7
2.2	Rasterização utilizando a aproximação δ_1	8
2.3	Rasterização correta utilizando a aproximação δ_1	9
2.4	Rasterização utilizando a aproximação δ_2	13
2.5	Casos típicos de pontos singulares isolados.	16
3.1	Curvas algébricas de graus 18 e 32.	18
3.2	Logotipo do projeto Pisces e “sorriso do palhaço”	19
3.3	Coração e pera.	20
3.4	Curvas racionais.	21
3.5	Curvas irracionais.	22

Capítulo 1

Introdução

Existem dois métodos principais para representar matematicamente curvas planas: a representação implícita e a representação paramétrica. Uma curva paramétrica é uma função $\mathbf{x}: I \subset \mathbb{R} \rightarrow \mathbb{R}^2$. Intuitivamente identificamos uma tal curva com seu *traço* $\mathbf{x}(I) = \{\mathbf{x}(\mathbf{t}) \in \mathbb{R}^2; \mathbf{t} \in I\}$. Uma curva implícita plana é o conjunto $Z(f)$ dos zeros de uma função $f: \mathcal{D} \subset \mathbb{R}^2 \rightarrow \mathbb{R}$, definida num subconjunto \mathcal{D} do plano, ou seja,

$$Z(f) = \{(x, y) \in \mathcal{D}; f(x, y) = 0\}.$$

Neste trabalho, o domínio \mathcal{D} será sempre um retângulo. Existem muitos métodos eficientes para desenhar o traço de uma curvas paramétrica. No entanto, o mesmo não acontece para curvas implícitas. Decorre do Teorema da Função Inversa que para cada ponto regular \mathbf{p} de uma curva implícita $Z(f)$, isto é, um ponto $\mathbf{p} = (x, y)$ tal que $f(\mathbf{p}) = 0$ e $\nabla f(\mathbf{p}) \neq \mathbf{0}$, existe uma parametrização local de $Z(f)$ numa vizinhança de \mathbf{p} . Usando esse fato, uma abordagem que muitos pesquisadores seguem é aproximar uma curva dada implicitamente por curvas paramétricas e desenhar as aproximações obtidas usando os métodos existentes [5, 7, 12]. O problema existente com essa abordagem é que curvas implícitas podem ter múltiplas componentes conexas e pontos singulares, de maneira que, para desenhar corretamente tais curvas, um algoritmo deve identificar os pontos singulares e as componentes conexas [1]. Porém, tais algoritmos são de modo geral complexos, computacionalmente caros e só lidam com singularidades isoladas.

Há ainda algoritmos que calculam aproximações lineares por partes [8]. A idéia é dividir o domínio retangular \mathcal{D} uma única vez numa malha regular de retângulos ou triângulos e em seguida visitar todas as células ocupadas

pela curva, a começar de uma dessas células. Dentro de uma célula ocupada, a curva é aproximada por um segmento de reta. Essa abordagem possui dois problemas. O primeiro é como determinar uma célula ocupada para começar o processo. O segundo problema é que, para garantir que todas as componentes conexas serão desenhadas, a priori todas as células deveriam ser visitadas [6]. Além dos métodos que fazem aproximação paramétrica, existem outros que seguem a idéia de *rasterização*. Rasterizar uma curva plana significa obter uma imagem que represente o traço da curva num certo domínio retangular \mathcal{D} , isto é, uma imagem representando um retângulo \mathcal{D} discretizado numa malha regular na qual apenas os blocos que intersectam a curva são preenchidos, como mostra a Figura 1.1.

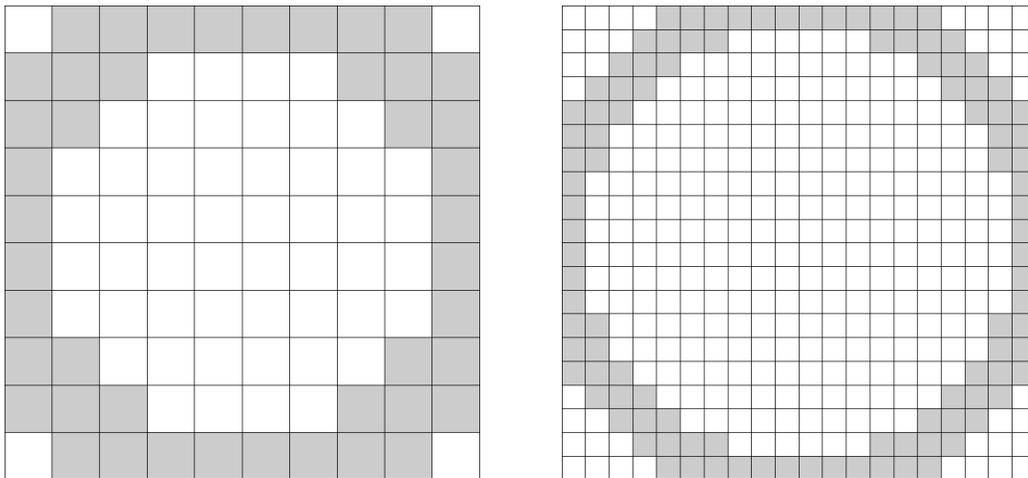


Figura 1.1: Rasterização de um círculo.

Entre os métodos para rasterizar curvas, destacamos aqueles que utilizam *subdivisão espacial*, que consideram o retângulo base \mathcal{D} como um *pixel* em baixa resolução, descartando-o caso não intersecte a curva, subdividindo-o em regiões menores, caso contrário, e aplicam este procedimento recursivamente. O problema se reduz, então, a decidir quando um retângulo intersecta a curva. Há vários trabalhos na literatura que seguem essa abordagem usando a Aritmética Intervalar [3], apoiados no fato que o conceito de *extensão intervalar* de funções fornece automaticamente um modo de decidir se 0 pertence ou não a $f(\mathcal{D})$, isto é, uma maneira correta de descartar blocos que não intersectam a curva dada. O problema restante, então, é refinar o conjunto não descartado para que tenha uma certa largura desejada (ver Algoritmo 1).

Nosso principal objetivo é apresentar um algoritmo de Taubin [9] para resolver esse problema através de aproximações da função distância euclidiana de um ponto do plano a uma curva implícita. De certa forma, o método desenvolvido por Taubin é uma técnica intervalar [4] especializada para polinômios. Porém, esta última não é correta para curvas quaisquer (no sentido que pode descartar erroneamente blocos que intersectam a curva em questão) como os métodos intervalares citados, mas para curvas algébricas mostraremos um algoritmo exato (ver Seção 2.3), que é a principal contribuição do trabalho de Taubin.

Capítulo 2

Rasterização de Curvas

Implícitas usando

Aproximações de distância

Neste capítulo descrevemos de maneira detalhada o algoritmo de Taubin [9] sobre rasterização de curvas implícitas planas. Na Seção 2.1, descrevemos um algoritmo básico de rasterização baseado na distância euclidiana e em seguida apresentamos uma versão recursiva do mesmo, utilizando subdivisão espacial para diminuir a complexidade. Na Seção 2.2, apresentamos uma aproximação de distância de primeira ordem, no sentido de utilizar derivadas de primeira ordem. Na Seção 2.3, generalizamos a mesma para ordem qualquer, mostrando que a aproximação de ordem k é capaz de rasterizar corretamente curvas algébricas de grau k . Na Seção 2.4, comentamos seu desempenho em torno de pontos singulares.

2.1 O algoritmo básico

Um algoritmo simples para rasterizar uma curva implícita com largura aproximadamente L consiste em discretizar o domínio numa grade regular de n^2 unidades (“blocos”) suficientemente fina e preencher os blocos cujo centro dista da curva no máximo a metade da largura desejada para a rasterização, como mostra o Algoritmo 1.

Algoritmo 1 DrawCurve(n, L)

```
for i = 1 to n do
  for j = 1 to n do
    p = center(Block(i, j))
    if ( $\delta(p, Z(f)) < L/2$ ) then
      draw Block(i, j)
    end if
  end for
end for
```

Embora o número de blocos numa grade $n \times n$ que interceptam uma curva suave seja de ordem $\mathcal{O}(n)$, o algoritmo acima executa necessariamente n^2 testes de distância, o que é claramente um desperdício computacional. A fim de contornar esta questão, vamos modificar o algoritmo acima aproveitando a geometria do problema, fazendo subdivisões sucessivas do espaço.

Iniciamos considerando o domínio retangular onde a curva será rasterizada como um único bloco (ou um “pixel em baixa resolução”). Se este bloco satisfaz o teste de distância para sua resolução, que consiste em seu centro distar da curva menos que metade de sua diagonal mais metade da largura da curva, então este bloco é dividido em quatro blocos iguais e repete-se o processo até que se atinja a largura desejada. Ver o Algoritmo 2.

A Figura 2.1 ilustra o uso deste algoritmo para rasterizar o círculo da Figura 1.1. Com isto diminui-se significativamente a complexidade computacional do algoritmo original. Resta-nos o problema de avaliar a distância euclidiana de um ponto a uma curva implícita, um problema matematicamente difícil e sem solução geral. Ao invés de tentar avaliar a distância euclidiana exata, usaremos aproximações que serão cotas inferiores para a distância euclidiana no caso de curvas algébricas, o que nos dará algoritmos práticos para rasterizar tais curvas. Nas próximas seções mostraremos que essas aproximações são assintoticamente equivalentes à distância euclidiana em torno de pontos regulares e analizaremos seu comportamento em torno de pontos singulares.

Algoritmo 2 RecursiveDrawCurve(B, L)

```
p = center(B)
H = half-diagonal(B)
if ( $\delta(p, Z(f)) < H + L/2$ ) then
  if  $H < L/2$  then
    draw(B)
  else
     $B = B_1 \cup B_2 \cup B_3 \cup B_4$ 
    for  $i = 1$  to 4 do
      RecursiveDrawCurve( $B_i, L$ )
    end for
  end if
end if
```

2.2 Aproximação de primeira ordem

Vamos definir uma aproximação para a distância euclidiana de um ponto a uma curva implícita $Z(f)$ que faz uso apenas de derivadas parciais de primeira ordem de f . Consideremos uma função $f: \mathcal{D} \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ de classe \mathcal{C}^1 e \mathbf{p} um ponto regular de f no domínio retangular \mathcal{D} . Será suficiente considerar apenas pontos do domínio da curva (e não pontos quaisquer do \mathbb{R}^2), uma vez que rasterizar a curva $Z(f)$ em \mathcal{D} equivale a obter $Z(f) \cap \mathcal{D}$.

Desenvolvendo f em série de Taylor em torno de \mathbf{p} , obtemos

$$f(\mathbf{q}) = f(\mathbf{p}) + \nabla f(\mathbf{p})^T(\mathbf{p} - \mathbf{q}) + \mathcal{O}(\|\mathbf{p} - \mathbf{q}\|^2),$$

onde \mathbf{q} é um ponto de \mathcal{D} suficientemente próximo de \mathbf{p} . Aplicando a desigualdade triangular e a desigualdade de Cauchy-Schwarz, temos

$$\begin{aligned} |f(\mathbf{q})| &\approx |f(\mathbf{p}) + \nabla f(\mathbf{p})^T(\mathbf{p} - \mathbf{q})| \\ &\geq |f(\mathbf{p})| - |\nabla f(\mathbf{p})^T(\mathbf{p} - \mathbf{q})| \\ &\geq |f(\mathbf{p})| - \|\nabla f(\mathbf{p})\| \|\mathbf{p} - \mathbf{q}\|. \end{aligned}$$

Se \mathbf{p} é um ponto suficientemente próximo da curva, a expressão acima será válida para os pontos \mathbf{q} próximos de \mathbf{p} pertencentes à curva. Em particular

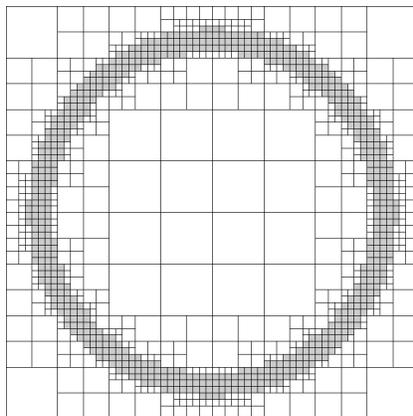


Figura 2.1: Rasterização utilizando o Algoritmo 2.

para um ponto \mathbf{q}^* que realize a distância euclidiana de \mathbf{p} a $Z(f) \cap \mathcal{D}$, temos

$$\delta(\mathbf{p}, Z(f)) = \inf\{\|\mathbf{p} - \mathbf{q}\| ; \mathbf{q} \in \mathcal{D}, f(\mathbf{q}) = 0\} = \|\mathbf{p} - \mathbf{q}^*\|$$

(o ínfimo será atingido pelo fato de \mathcal{D} ser compacto). Assim, se $f(\mathbf{q}) = 0$, com \mathbf{q} nas condições acima, temos

$$|f(\mathbf{p})| \leq \|\nabla f(\mathbf{p})\| \|\mathbf{p} - \mathbf{q}\| + |f(\mathbf{q})| = \|\nabla f(\mathbf{p})\| \|\mathbf{p} - \mathbf{q}\|,$$

donde

$$\frac{|f(\mathbf{p})|}{\|\nabla f(\mathbf{p})\|} \leq \|\mathbf{p} - \mathbf{q}\|.$$

Em particular,

$$\|\mathbf{p} - \mathbf{q}^*\| = \min_{\mathbf{q} \in \Lambda} \|\mathbf{p} - \mathbf{q}\| \geq \frac{|f(\mathbf{p})|}{\|\nabla f(\mathbf{p})\|}, \quad (2.1)$$

onde $\Lambda = B_\varepsilon(\mathbf{p}) \cap Z(f)$ é uma vizinhança da curva $Z(f)$ em torno de \mathbf{p} . Com base nessas observações, definimos a aproximação de primeira ordem para a distância do ponto \mathbf{p} à curva $Z(f)$ por

$$\delta_1(\mathbf{p}, Z(f)) = \frac{|f(\mathbf{p})|}{\|\nabla f(\mathbf{p})\|}.$$

Pela desigualdade (2.1), esta aproximação é uma *cota inferior* para a distância euclidiana de pontos regulares próximos da curva em questão. Embora esta

propriedade seja interessante, o desenvolvimento acima é apenas uma motivação para a definição de δ_1 . Esta aproximação é muito interessante por sua simplicidade computacional, uma vez que só envolve o cálculo da função f e seu gradiente ∇f num ponto p . Naturalmente, ela não resolve todos os problemas. Como nossa análise foi feita para pontos (em um certo sentido) próximos da curva, nada nos garante que um bloco de resolução baixa não seja descartado erroneamente no Algoritmo 2. A Figura 2.2 demonstra o uso do Algoritmo 2 com a aproximação δ_1 para rasterizar uma curva algébrica de grau 4 (sua equação, assim como as demais equações usadas neste trabalho, encontra-se no Apêndice A). Note que um bloco grande que intersecta a curva foi incorretamente descartado. Para melhorar a eficiência computacional do Algoritmo 2, a aproximação δ_1 sempre será aplicada antes das aproximações de ordem superior, discutidas a seguir. Há casos, porém, em que a aproximação δ_1 dá bons resultados, como mostra a Figura 2.3.

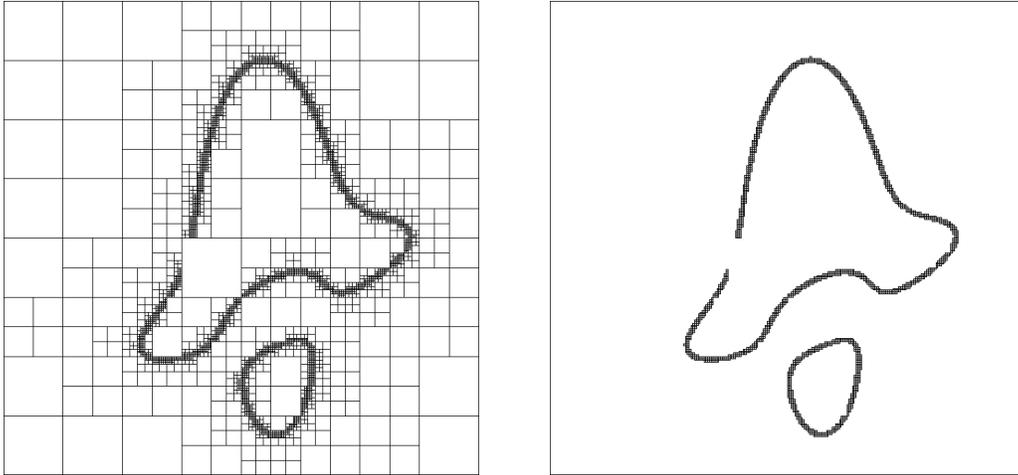


Figura 2.2: Rasterização utilizando a aproximação δ_1 .

A proposição abaixo mostra que a aproximação δ_1 é assintoticamente equivalente à distância euclidiana para pontos regulares.

Proposição 1 *Seja $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ uma função de classe \mathcal{C}^2 numa vizinhança de um ponto regular p_0 de $Z(f)$. Sejam w um vetor normal unitário a $Z(f)$ em p_0 e $p_t = p_0 + tw$, para $t \in \mathbb{R}$. Então,*

$$\lim_{t \rightarrow 0} \frac{\delta_1(p_t, Z(f))}{\delta(p_t, Z(f))} = 1.$$

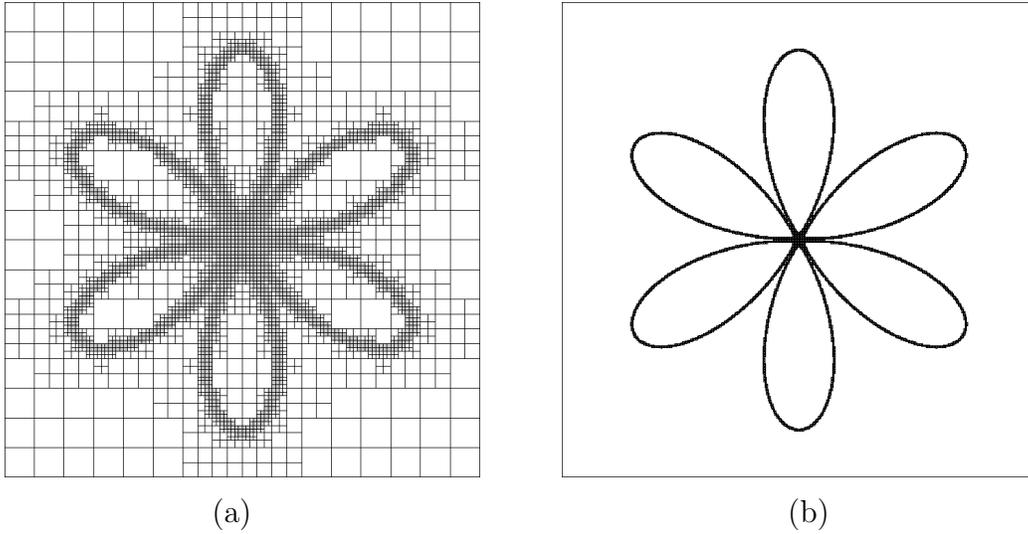


Figura 2.3: Rasterização correta utilizando a aproximação δ_1 .

Prova. Das hipóteses da proposição e por [11, capítulo 16], existe uma vizinhança de \mathbf{p}_0 na qual a distância de \mathbf{p}_t à curva $Z(f)$ é exatamente $|t|$,

$$\delta(\mathbf{p}_t, Z(f)) = \|\mathbf{p}_t - \mathbf{p}_0\| = |t|.$$

Além disso, como \mathbf{p}_0 é um ponto regular de $Z(f)$, temos $\|\nabla f(\mathbf{p}_0)\| > 0$ e, por continuidade, $\|\nabla f(\mathbf{p}_t)\|$ é limitado para t suficientemente próximo de 0. Como \mathbf{w} é normal unitário à curva, temos $\nabla f(\mathbf{p}_0) = \pm \|\nabla f(\mathbf{p}_0)\| \mathbf{w}$. Por continuidade da segunda derivada de f e pelos fatos acima, temos

$$\begin{aligned} \nabla f(\mathbf{p}_t) &= \nabla f(\mathbf{p}_0) + \mathcal{O}(\|\mathbf{p}_t - \mathbf{p}_0\|) \\ &= \pm \|\nabla f(\mathbf{p}_0)\| \mathbf{w} + \mathcal{O}(|t|) \end{aligned}$$

e, portanto,

$$\begin{aligned} 0 = f(\mathbf{p}_0) &= \nabla f(\mathbf{p}_t) + \nabla f(\mathbf{p}_t)^\top (\mathbf{p}_0 - \mathbf{p}_t) \mathcal{O}(\|\mathbf{p}_0 - \mathbf{p}_t\|^2) \\ &= f(\mathbf{p}_t) \pm \|\nabla f(\mathbf{p}_t)\| |t| + \mathcal{O}(|t|^2). \end{aligned}$$

Dividindo por $\|\nabla f(\mathbf{p}_t)\| |t|$,

$$\frac{1}{|t|} \frac{f(\mathbf{p}_t) - f(\mathbf{p}_0)}{\|\nabla f(\mathbf{p}_t)\|} = \frac{\delta_1(\mathbf{p}_t, Z(f))}{\delta(\mathbf{p}_t, Z(f))} = 1 + \mathcal{O}(|t|),$$

como queríamos demonstrar. ■

2.3 Aproximações de ordem superior

Nesta seção vamos assumir que f é de classe \mathcal{C}^{k+1} numa vizinhança de $\mathbf{p} \in \mathcal{D}$ e $f(\mathbf{p}) \neq 0$ (do contrário, \mathbf{p} pertenceria à curva $Z(f)$). Expandindo f em série de Taylor de ordem k em torno de $\mathbf{p} = (\mathbf{u}, \mathbf{v})$, temos

$$f(\mathbf{q}) = \sum_{i=1}^k \sum_{j=1}^k f_{ij}(\mathbf{x} - \mathbf{u})^i (\mathbf{x} - \mathbf{y})^j + \mathcal{O}(\|\mathbf{p} - \mathbf{q}\|^{k+1}),$$

onde $\mathbf{q} = (\mathbf{x}, \mathbf{y})$ e

$$f_{ij} = \frac{1}{i!j!} \frac{\partial^{i+j} f(\mathbf{p})}{\partial \mathbf{x}^i \partial \mathbf{y}^j}.$$

Vamos denotar o polinômio de Taylor de ordem k de f em \mathbf{p} por $T^k f_{\mathbf{p}}$. Nosso objetivo é obter uma cota inferior para a distância de \mathbf{p} a $Z(T^k f_{\mathbf{p}})$. Para isso, inicialmente vamos reescrever o polinômio $T^k f_{\mathbf{p}}$ como uma soma de polinômios homogêneos:

$$T^k f_{\mathbf{p}}(\mathbf{q}) = \sum_{h=0}^k \left\{ \sum_{i+j=h} f_{ij}(\mathbf{x} - \mathbf{u})^i (\mathbf{x} - \mathbf{y})^j \right\} = \sum_{h=0}^k f_{\mathbf{p}}^h(\mathbf{q}).$$

Aplicando a desigualdade triangular, obtemos

$$|T^k f_{\mathbf{p}}(\mathbf{q})| \geq |f_{\mathbf{p}}^0| - \sum_{h=1}^k |f_{\mathbf{p}}^h(\mathbf{q})|. \quad (2.2)$$

Agora vamos analisar cada parcela separadamente. Temos

$$((\mathbf{x} - \mathbf{u})^2 + (\mathbf{y} - \mathbf{v})^2)^h = \sum_{i+j=h} \binom{h}{i} (\mathbf{x} - \mathbf{u})^{2i} (\mathbf{y} - \mathbf{v})^{2j} = \|\mathbf{X}_h\|^2, \quad (2.3)$$

onde \mathbf{X}_h é um elemento de \mathbb{R}^{h+1} de coordenadas $\binom{h}{i}^{1/2} (\mathbf{x} - \mathbf{u})^i (\mathbf{y} - \mathbf{v})^j$, para todos i e j tais que $i+j = h$. Dessa forma podemos reescrever o polinômio $f_{\mathbf{p}}^h$ como um produto interno em \mathbb{R}^{h+1} :

$$f_{\mathbf{p}}^h = \sum_{i+j=h} \frac{f_{ij}}{\binom{h}{i}^{1/2}} \cdot \binom{h}{i}^{1/2} (\mathbf{x} - \mathbf{u})^i (\mathbf{y} - \mathbf{v})^j = \mathbf{F}_h^T \mathbf{X}_h,$$

onde F_h^T é um elemento de \mathbb{R}^{h+1} de coordenadas $f_{ij}/\binom{h}{i}^{1/2}$, para todos i e j tais que $i+j=h$, se $h \geq 1$ e $F_0 = f_p^0 = f_{00} = f(p)$. Aplicando a desigualdade de Cauchy-Schwarz e usando (2.3), obtemos

$$|f_p^h(q)| = |F_h^T X_h| \leq \|F_h\| \|X_h\| = \|F_h\| \delta^h, \quad (2.4)$$

onde $\delta = \sqrt{(x-u)^2 + (y-v)^2} = \|p-q\|$. Aplicando as desigualdades (2.4) em (2.2), concluimos que

$$|T^k f_p(q)| \geq |F_0| - \sum_{h=1}^k \|F_h\| \delta^h.$$

Como o polinômio $F_p^k(\delta) = |F_0| - \sum_{h=1}^k \|F_h\| \delta^h$ é uma função decrescente para $\delta > 0$ e positiva em 0, pois por hipótese $F_0 = f(p) \neq 0$, este polinômio possui uma única raiz real δ_k , que é uma *cota inferior* para a distância euclidiana de p a $Z(T^k f_p)$. De fato, para $\delta < \delta_k$, tem-se

$$|T^k f_p(q)| \geq |F_0| - \sum_{h=1}^k \|F_h\| \delta^h > 0,$$

donde, se $q = (x, y)$ é um ponto da curva $Z(T^k f_p)$ que realiza a distância a p , então $\delta = \|p-q\| \geq \delta_k$. Assim, se f é um polinômio de grau menor ou igual a k , temos $f \equiv T^k f_p$ e δ_k é uma cota inferior para a distância de p a $Z(f)$. Uma vez que, para nossos objetivos, não precisamos saber o valor de δ_k mas apenas comparar um $\delta > 0$ dado com δ_k , apenas avaliamos o polinômio F_p^k em δ e determinamos o sinal deste valor. Se for positivo, a distância de p a $Z(T^k f_p)$ é maior que δ , o que para polinômios de grau menor ou igual a k significa que o disco centrado em p de raio δ não intersecta a curva $Z(f)$ e, portanto, pode ser descartado no Algoritmo 2.

Se, por alguma razão, necessita-se saber o valor de δ_k , pode-se obtê-lo usando um algoritmo padrão de cálculo de raiz [2]. Por exemplo, o método de Newton funcionaria muito bem. Para ver isso, note que as aproximações δ_k formam uma seqüência decrescente pois

$$F_p^k(\delta) = |F_0| - \sum_{h=1}^k \|F_h\| \delta^h \leq |F_0| - \sum_{h=1}^{k-1} \|F_h\| \delta^h = F_p^{k-1}(\delta),$$

para todo $\delta > 0$. Portanto, como F_p^k é decrescente em \mathbb{R}_+ , temos $\delta_k \leq \delta_{k-1}$. Além disso, δ_1 coincide com a aproximação de primeira ordem desenvolvida

na Seção 2.2. Donde, como $0 \leq \delta_k \leq \delta_1$, temos um intervalo inicial para aplicar o método de Newton. Para uma cota superior melhor, pode-se usar a segunda aproximação de distância caso $\|F_2\| \neq 0$. Neste caso, a única raiz positiva de $F_p^2(\delta)$ é dada por

$$\delta_2 = \sqrt{\frac{\|F_1\|^2}{4\|F_2\|^2} + \frac{\|F_0\|}{\|F_2\|}} - \frac{\|F_1\|}{2\|F_2\|}.$$

Temos então $\delta_k \in [0, \delta_2]$ para todo $k \in \mathbb{N}$.

Embora tenhamos mostrado que a aproximação δ_k garante a rasterização correta de curvas algébricas de grau k , Taubin [9] relata que, em seus experimentos, a aproximação de segunda ordem tem sido suficiente para rasterizar corretamente curvas algébricas de alto grau (em nosso caso, temos tido êxito também com curvas não algébricas; ver Capítulo 3). A Figura 2.4 mostra a curva algébrica de grau 4 da Figura 2.2 rasterizada corretamente utilizando a aproximação δ_2 . Se, num certo ponto sendo testado, as derivadas de primeira e segunda ordem se anularem, situação em que não teríamos nenhuma informação da aproximação δ_2 , poderíamos usar a aproximação de terceira ordem, e assim por diante.

A proposição abaixo mostra que a aproximação δ_k é assintoticamente equivalente à distância euclidiana para pontos regulares.

Proposição 2 *Seja $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ uma função de classe \mathcal{C}^{k+1} numa vizinhança de um ponto regular p_0 de $Z(f)$. Sejam w um vetor normal unitário a $Z(f)$ em p_0 e $p_t = p_0 + tw$, para $t \in \mathbb{R}$. Então,*

$$\lim_{t \rightarrow 0} \frac{\delta_k(p_t, Z(f))}{\delta(p_t, Z(f))} = 1.$$

Prova. Podemos supor que $k > 1$, pois o caso $k = 1$ já foi analisado na Proposição 1. Dado $t \in \mathbb{R}$, vamos denotar por $\delta_k(t)$ a aproximação de ordem k da distância do ponto p_t à curva $Z(f)$, isto é, $\delta_k(t)$ é a única raiz positiva da equação

$$F_0(t) - \sum_{h=1}^k \|F_h(t)\| \delta_k(t)^h = 0,$$

onde os coeficientes são funções contínuas de t . Para t suficientemente pequeno temos $|F_0(t)| \neq 0$ (se $t \neq 0$) e $\|F_1(t)\| = \|\nabla f(p_t)\| \neq 0$. Dividindo

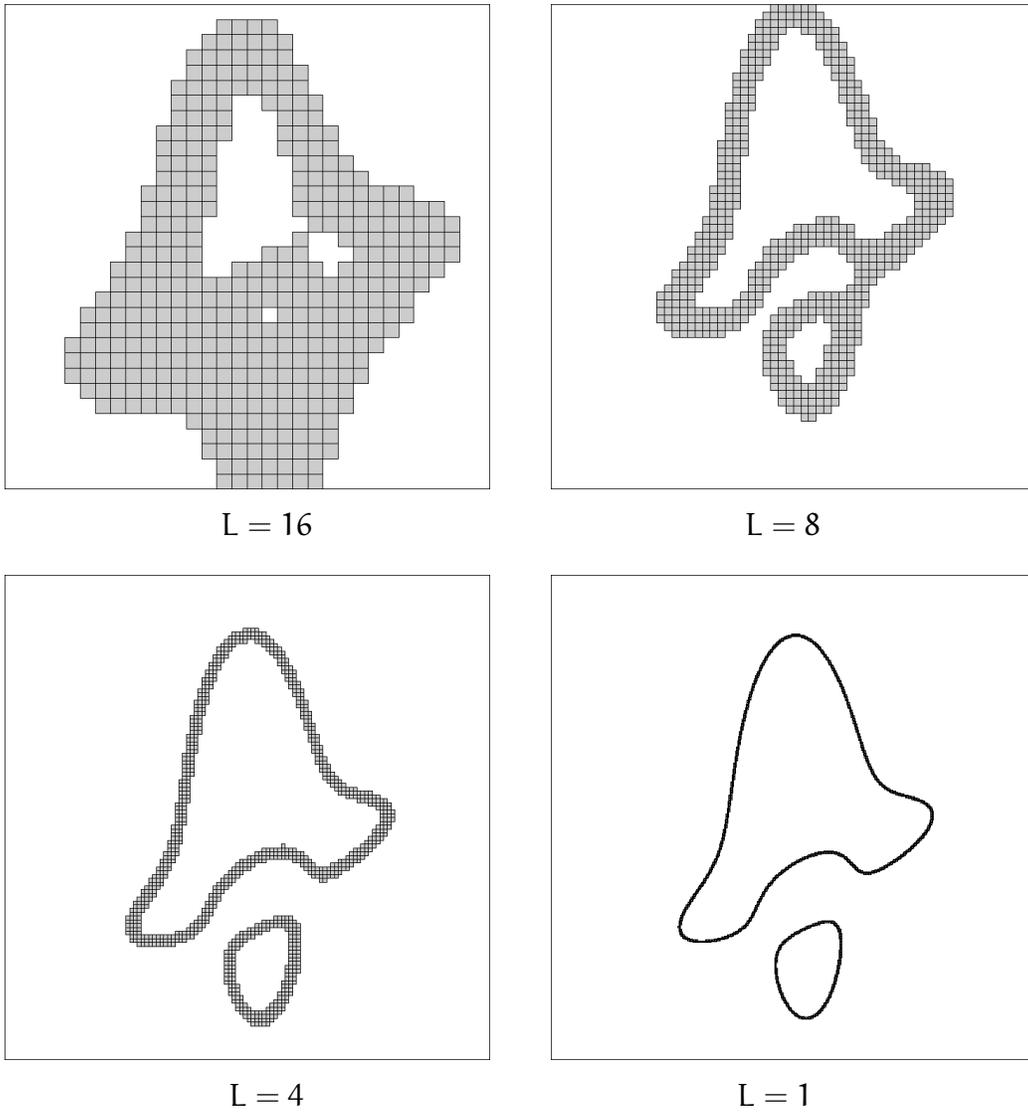


Figura 2.4: Rasterização utilizando a aproximação δ_2 .

ambos os membros da equação acima por $\delta_1(t) = |F_0(t)|/\|F_1(t)\|$, obtemos

$$1 - \frac{\delta_k(t)}{\delta_1(t)} = \frac{\delta_k(t)}{\delta_1(t)} \left\{ \sum_{h=2}^k \frac{\|F_h(t)\|}{\|F_1(t)\|} \right\} \delta_k(t).$$

Observe que os três fatores à direita da igualdade são positivos. O primeiro é limitado por 1, uma vez que $\delta_k(\mathbf{t}) \leq \delta_1(\mathbf{t})$ para todo $k > 1$; o segundo é limitado por

$$\sum_{h=2}^k \frac{\|F_h(\mathbf{t})\|}{\|F_1(\mathbf{t})\|},$$

pois $\delta_1(\mathbf{t}) < 1$, e por conseguinte é limitado para \mathbf{t} próximo de zero; e o último fator é limitado por $\delta_1(\mathbf{t})$. Assim,

$$\frac{\delta_k(\mathbf{t})}{\delta_1(\mathbf{t})} = 1 + \mathcal{O}(\delta_1(\mathbf{t})),$$

o que, pela Proposição 1, conclui a prova. ■

2.4 Pontos singulares

Até aqui analisamos o comportamento das aproximações propostas numa vizinhança de pontos regulares. Vamos agora estabelecer os conceitos de pontos múltiplos, retas tangentes e transversas em um ponto para em seguida estudar como se comportam as aproximações de distância em torno de pontos singulares.

Sejam função real $f : \mathbf{U} \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ suficientemente diferenciável e \mathbf{p} um ponto em \mathbf{U} . A *multiplicidade* de \mathbf{p} como zero de f é definida por

$$m(\mathbf{p}; f) = \min\{h \in \mathbb{N}; f_p^h \neq 0\},$$

onde o símbolo f_p^h foi introduzido na Seção 2.3, ou ∞ se todas as derivadas parciais de f se anulam em \mathbf{p} .

Tem-se naturalmente que

$$Z(f) = \{\mathbf{p} \in \text{Dom}(f); m(\mathbf{p}; f) > 0\}.$$

Um ponto $\mathbf{p} \in Z(f)$ é dito *simples* ou *regular* se tem multiplicidade 1 e *múltiplo* ou *singular* caso contrário. O comportamento das aproximações de distância em torno de um ponto singular depende não apenas da multiplicidade do ponto em questão mas também da estrutura do *espaço tangente* do mesmo ponto. Em um ponto simples \mathbf{p} , a curva $Z(f)$ possui uma única reta tangente, que é dada pela equação

$$\{\mathbf{q} \in \mathbb{R}^2; \nabla f(\mathbf{p})^T(\mathbf{q} - \mathbf{p}) = f_p^1(\mathbf{q}) = 0\}.$$

Por sua vez, um ponto múltiplo possui múltiplas tangentes. Infelizmente, não há um resultado geral sobre quando as aproximações de distância subestimam ou superestimam a distância euclidiana, como observa Taubin [9]. Este problema foi resolvido por Taubin em [10]. Vamos apresentar alguns dos casos típicos por meio de exemplos.

A Figura 2.5 mostra várias curvas algébricas com pontos singulares isolados. Nas curvas em (a), (b), (c), (f), (g), (h) e (i), a origem é o único ponto singular, o qual é um ponto duplo, ou seja, $m(\mathbf{p}; f) = 2$, em (a), (b), (c) e (i); um ponto quádruplo em (f) e um ponto sêxtuplo em (g) e (h). Em (d) há quatro pontos duplos, todos interseção de pares de retas. Em (e) há quatro pontos duplos, dois pontos triplos e dois pontos quádruplos. Em (i) a origem é um ponto isolado de $Z(f)$, o qual é um ponto duplo. Em (a) e (b) a origem possui duas tangentes simples. Em (c) a origem possui uma tangente dupla. Em (d) os quatro pontos singulares possuem duas tangentes simples, cada, as quais estão contidas na curva. Em (e) os dois pontos duplos e os dois pontos triplos possuem tangentes simples, enquanto os pontos quádruplos possuem uma tangente dupla e duas tangentes simples, cada. A origem possui uma tangente quádrupla em (f) e três tangentes duplas em (g). Em (h) a origem possui uma tangente dupla e quatro tangentes imaginárias e em (i), duas tangentes imaginárias.

O Algoritmo 2 foi usado para rasterizar as curvas na Figura 2.5, com a aproximação de distância de segunda ordem δ_2 em lugar da distância euclidiana. Pode-se notar nessa figura que os pontos múltiplos com tangentes simples são desenhados corretamente, assim como também alguns casos onde há tangentes múltiplas. Porém, em torno de pontos em que a multiplicidade das tangentes é alta como em (h), a aproximação de distância subestima a distância euclidiana exata, produzindo traços de largura superior à largura da curva.

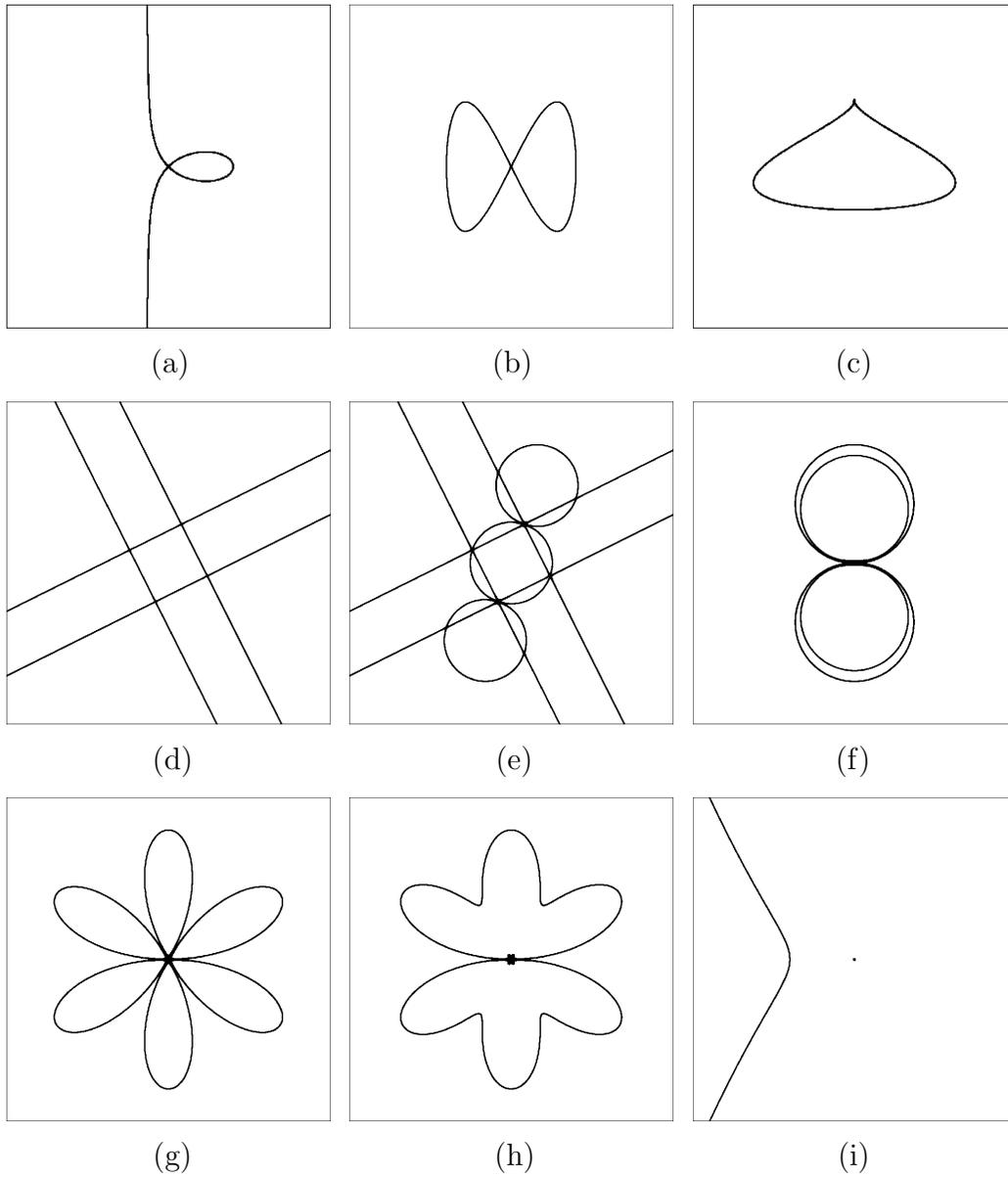


Figura 2.5: Casos típicos de pontos singulares isolados.

Capítulo 3

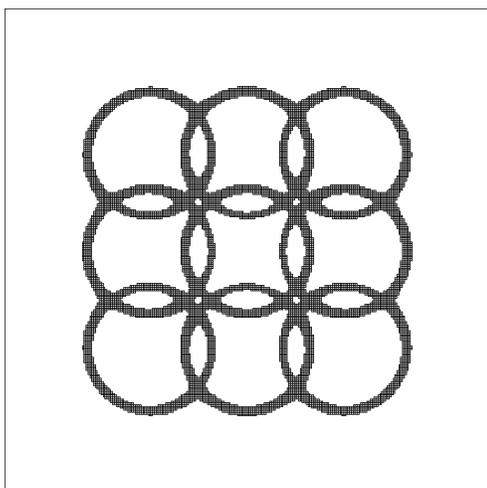
Exemplos

Neste capítulo apresentamos vários exemplos de curvas rasterizadas utilizando a aproximação δ_2 , bem como alguns testes feitos em [9].

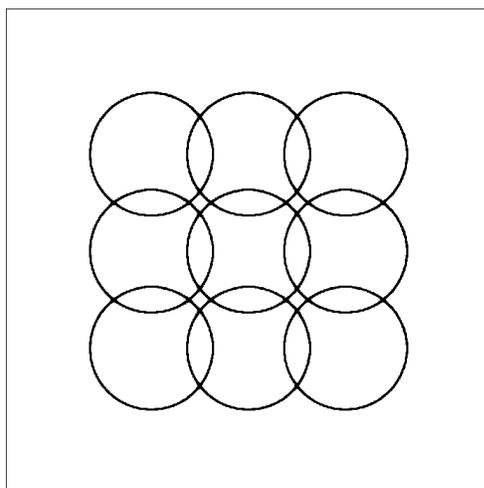
Como mostramos na Seção 2.3, a substituição da distância euclidiana pela aproximação δ_k no Algoritmo 2 produz rasterizações corretas de curvas algébricas de grau k . Porém, aplicar este método diretamente não é o mais indicado, pois o custo computacional do cálculo de derivadas necessárias para a aproximação δ_k é muito alto até mesmo para valores pequenos de k , dependendo da largura escolhida para a curva.

A Figura 3.1 mostra curvas algébricas de graus 18 (com 190 coeficientes) e 32 (com 561 coeficientes), respectivamente, a saber, uniões de círculos distribuídos regularmente num retângulo, rasterizadas corretamente usando a aproximação δ_2 . Com isso espera-se que, na maioria dos casos práticos, essa aproximação produza bons resultados.

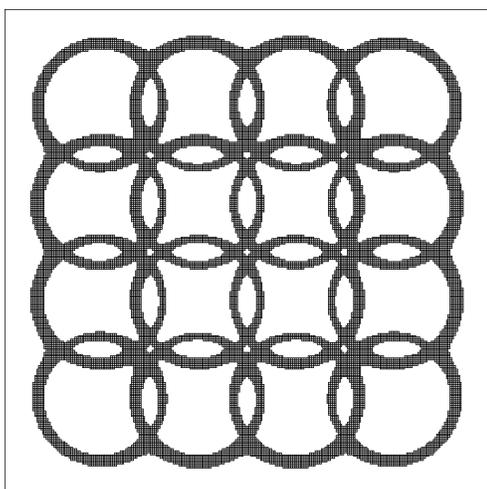
Recentemente, Martin et al. [13] fez uma análise comparativa detalhada de diversos métodos intervalares de rasterização de curvas algébricas, entre os quais uma versão um pouco melhorada [10] do trabalho de Taubin descrito aqui [9], que mostra que o Algoritmo de Taubin possui um bom desempenho se comparado aos principais métodos intervalares existentes para rasterização de curvas algébricas ficando, em quase todas as comparações (custo computacional, número de cálculos, resultados práticos), entre os mais eficientes.



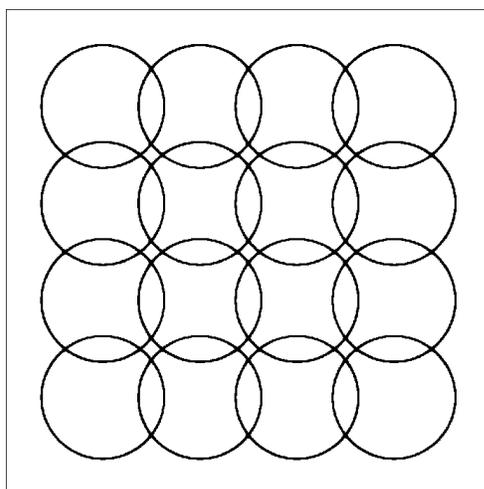
(a) $L = 4$



(b) $L = 1$

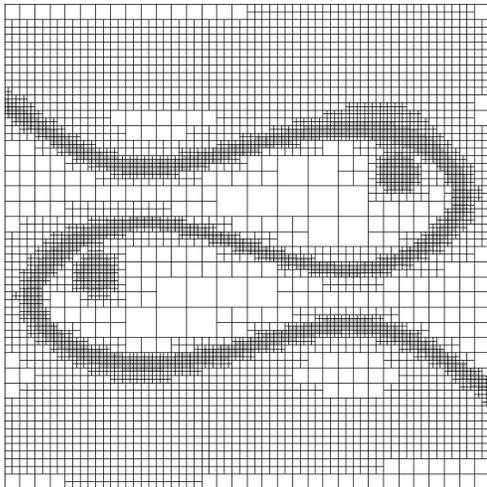


(c) $L = 4$

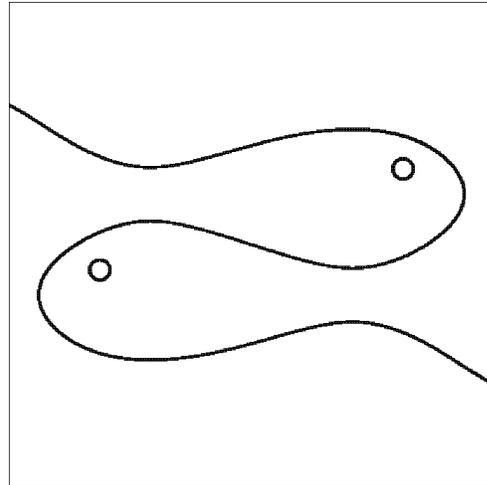


(d) $L = 1$

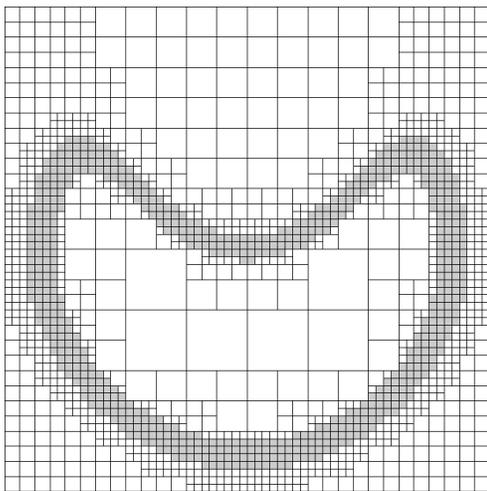
Figura 3.1: Curvas algébricas de graus 18 e 32.



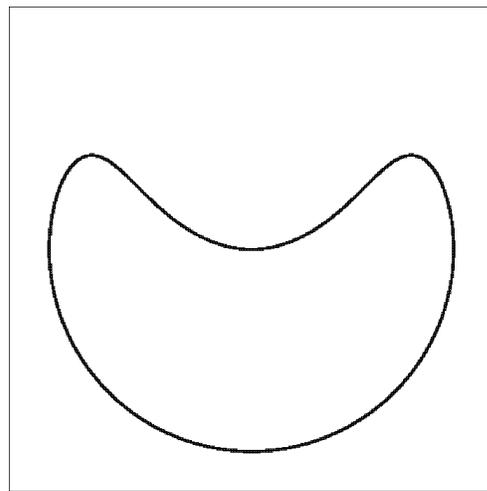
(a) $L = 5$



(b) $L = 1$

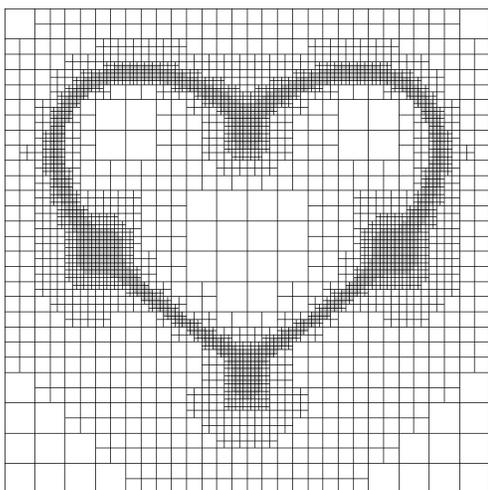


(c) $L = 10$

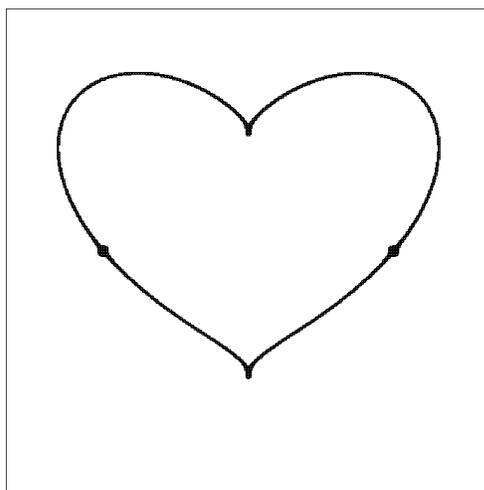


(d) $L = 1$

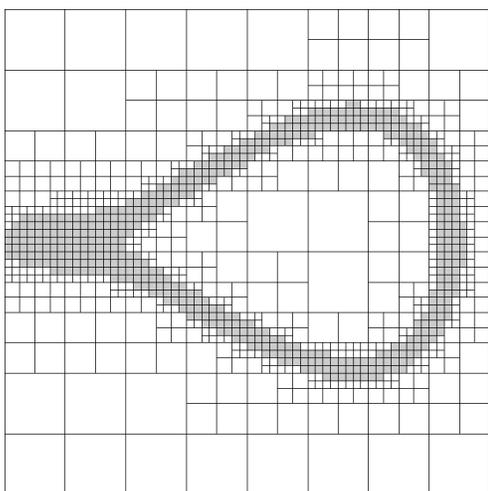
Figura 3.2: As figuras (a) e (b) mostram o logotipo do projeto Pisces (Platform for Implicit Surfaces and Curves and the Exploration of Singularities, mais informações no site <http://www.geom.umn.edu/fjw/pisces/>). As Figuras (c) e (d) mostram o “sorriso do palhaço”.



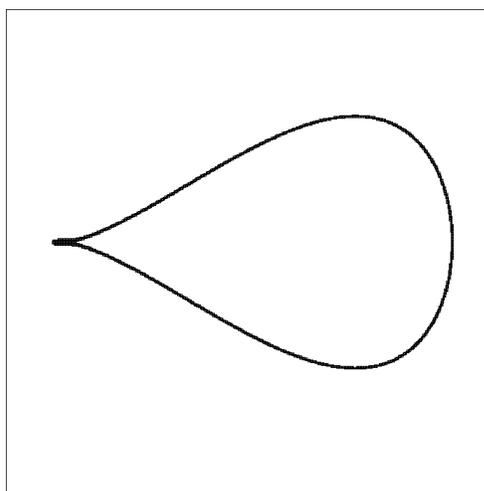
(a) $L = 5$



(b) $L = 1$

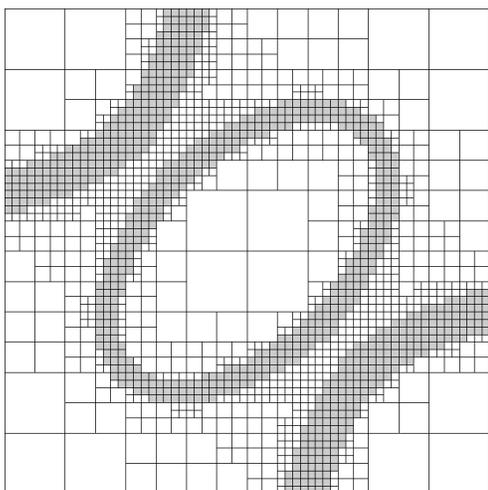


(c) $L = 10$

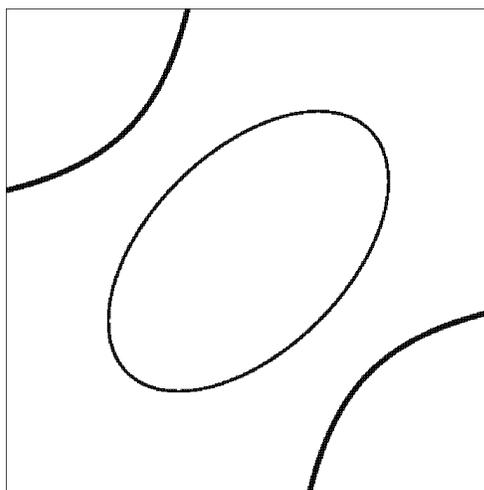


(d) $L = 1$

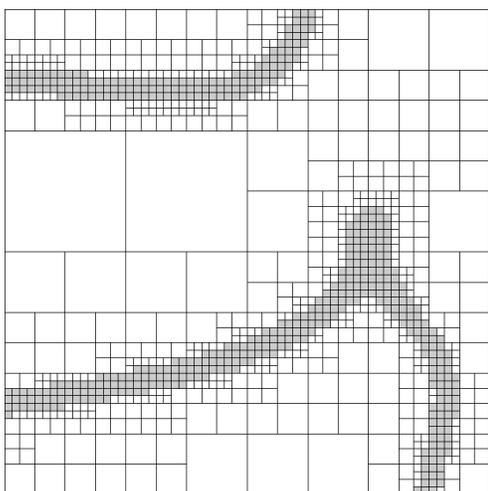
Figura 3.3: Coração e pera.



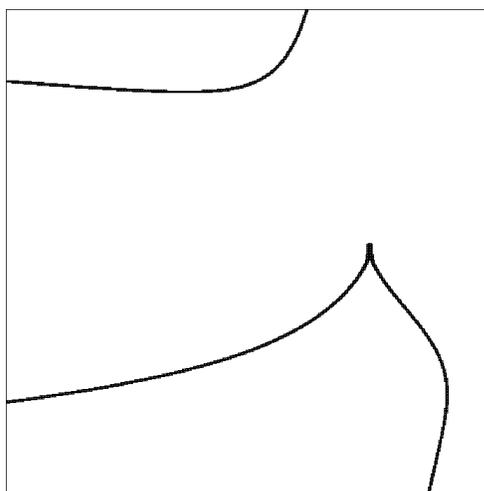
(a) $L = 10$



(b) $L = 1$

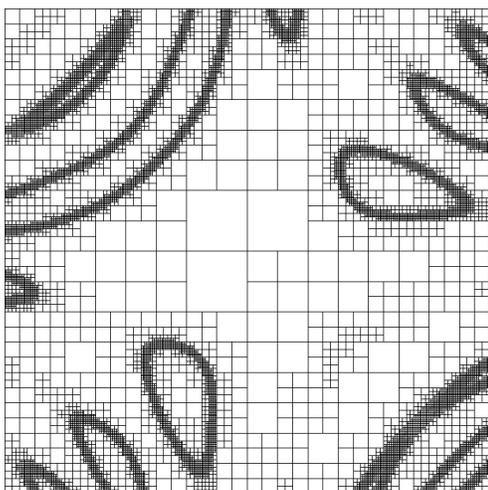


(c) $L = 10$

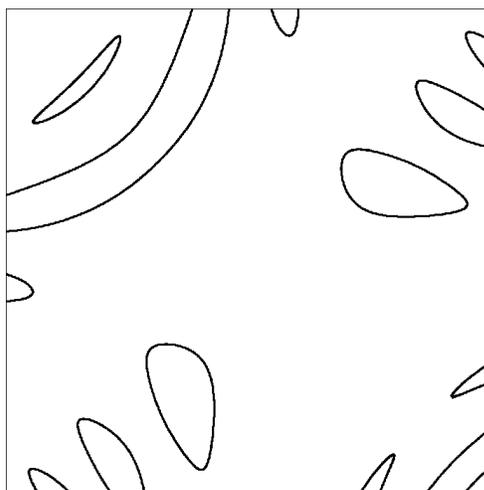


(d) $L = 1$

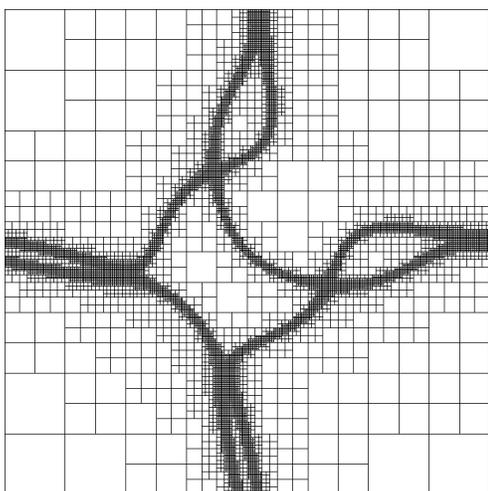
Figura 3.4: Curvas racionais.



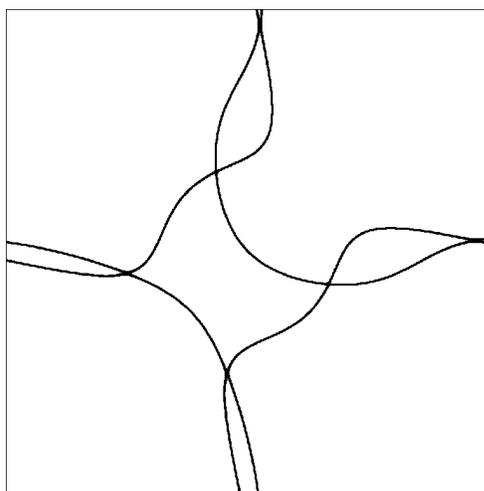
(a) $L = 5$



(b) $L = 1$



(c) $L = 5$



(d) $L = 1$

Figura 3.5: Curvas irracionais.

Capítulo 4

Conclusão

O Algoritmo 2, que utiliza a aproximação δ_2 , é capaz (teoricamente) de gerar traços de curvas implícitas regulares com largura aproximadamente constante, mesmo não algébricas, como mostram as Proposições 1 e 2. Pelos testes vistos em [9] e em [13], e pelos nossos experimentos com esse algoritmo (Capítulo 3), podemos dizer que ele é eficiente, nesse caso. Porém este algoritmo apresenta problemas em torno de pontos singulares, como mostram, por exemplo, as Figuras 3.3(b) e 2.5(h), em que o traço produzido pelo algoritmo é mais largo em torno dos pontos singulares.

Por outro lado, pode-se utilizar essa característica do algoritmo para detectar pontos singulares e componentes múltiplas de curvas algébricas. Outra característica interessante do algoritmo é sua facilidade de implementação.

Apêndice A

Equações

A seguir listamos as equações $f(x, y)$ das curvas mostradas no texto com seus domínios B e larguras L , usados no Algoritmo 2, quando não especificadas.

- Figuras 2.2 e 2.4

$$\begin{aligned}f(x, y) &= 0.004 + 0.110x - 0.177y - 0.174x^2 + 0.224xy - 0.303y^2 \\ &\quad - 0.168x^3 + 0.327x^2y - 0.087xy^2 - 0.013y^3 + 0.235x^4 \\ &\quad - 0.667x^3y + 0.745x^2y^2 - 0.029xy^3 + 0.072y^4. \\ B &= [-2.5, 2.5] \times [-2, 3].\end{aligned}$$

- Figura 2.5

$$\begin{aligned}\text{(a)} \quad f(x, y) &= x^3 + 3xy^2 - x^2 + y^2 \\ B &= [-2.5, 2.5] \times [-2.5, 2.5]; L = 1. \\ \text{(b)} \quad f(x, y) &= 4x^4 - 4x^2 + y^2 \\ B &= [-2.5, 2.5] \times [-2.5, 2.5]; L = 1. \\ \text{(c)} \quad f(x, y) &= -3y^4 + 5y^3 - x^2 \\ B &= [-2.5, 2.5] \times [-3.5, 1.5]; L = 1. \\ \text{(d)} \quad f(x, y) &= (2y - x - 1)(2y - x + 1)(2x + y + 1)(2x + y - 1) \\ B &= [-2.5, 2.5] \times [-2.5, 2.5]; L = 1.\end{aligned}$$

(e)

$$f(x, y) = \frac{(2y - x - 1)(2y - x + 1)(2x + y + 1)(2x + y - 1)}{((5x - 2)^2 + (5y - 6)^2 - 10)((5x + 2)^2 + (5y + 6)^2 - 10)(25(x^2 + y^2) - 10)}$$

$$B = [-2.5, 2.5] \times [-2.5, 2.5]; L = 1.$$

(f)

$$f(x, y) = \frac{((y + 1)^2 + x^2 - 1)((y - 1)^2 + x^2 - 1)}{((y + 1.1)^2 + x^2 - 1.21)((y - 1.1)^2 + x^2 - 1.21)}$$

$$B = [-3, 3] \times [-3, 3]; L = 1.$$

(g) $f(x, y) = (3x^2 - y^2)^2 y^2 - (x^2 + y^2)^4$

$$B = [-1.25, 1.25] \times [-1.25, 1.25]; L = 1.$$

(h) $f(x, y) = (8x^4 - 4x^2 y^2 + y^4) y^2 - (x^2 + y^2)^4$

$$B = [-1.25, 1.25] \times [-1.25, 1.25]; L = 1.$$

(i) $f(x, y) = x^2 + y^2 + x^3$

$$B = [-2.5, 2.5] \times [-2.5, 2.5]; L = 1.$$

• Figura 3.1

(a) e (b)

$$f(x, y) = \frac{((x + 1)^2 + (y + 1)^2 - 0.4)((x + 1)^2 + y^2 - 0.4)}{((x + 1)^2 + (y - 1)^2 - 0.4)(x^2 + (y + 1)^2 - 0.4)} \cdot \frac{(x^2 + y^2 - 0.4)(x^2 + (y - 1)^2 - 0.4)((x - 1)^2 + (y + 1)^2 - 0.4)}{((x - 1)^2 + y^2 - 0.4)((x - 1)^2 + (y - 1)^2 - 0.4)}$$

$$B = [-3, 3] \times [-3, 3].$$

(c) e (d)

$$f(x, y) = \frac{((x - 1.5)^2 + (y + 1.5)^2 - 0.4)((x - 0.5)^2 + (y + 1.5)^2 - 0.4)}{((x + 0.5)^2 + (y + 1.5)^2 - 0.4)((x + 1.5)^2 + (y + 1.5)^2 - 0.4)} \cdot \frac{((x - 1.5)^2 + (y + 0.5)^2 - 0.4)((x - 0.5)^2 + (y + 0.5)^2 - 0.4)}{((x + 0.5)^2 + (y + 0.5)^2 - 0.4)((x + 1.5)^2 + (y + 0.5)^2 - 0.4)} \cdot \frac{((x - 1.5)^2 + (y - 0.5)^2 - 0.4)((x - 0.5)^2 + (y - 0.5)^2 - 0.4)}{((x + 0.5)^2 + (y - 0.5)^2 - 0.4)((x + 1.5)^2 + (y - 0.5)^2 - 0.4)} \cdot \frac{((x - 1.5)^2 + (y - 1.5)^2 - 0.4)((x - 0.5)^2 + (y - 1.5)^2 - 0.4)}{((x + 0.5)^2 + (y - 1.5)^2 - 0.4)((x + 1.5)^2 + (y - 1.5)^2 - 0.4)}$$

$$B = [-3, 3] \times [-3, 3].$$

- Figura 3.2

(a) e (b)

$$\begin{aligned}
 f(x, y) &= \text{Peixe} \cdot \text{olho}_1 \cdot \text{olho}_2, \text{ onde:} \\
 \text{Peixe} &= x^3/3 - x + \alpha y(-1 + (\alpha y)^2 \\
 &\quad (1/6 + (\alpha y)^2(-1/120 + (\alpha y)^2/5040))) \\
 \text{olho}_1 &= (x - a)^2 + (y - .5 - b)^2 - r^2 \\
 \text{olho}_2 &= (x + a)^2 + (y + .5 - b)^2 - r^2 \\
 r &= 0.1; \alpha = 3.14; a = 1.5; b = 0.25
 \end{aligned}$$

$$B = [-2.4, 2.4] \times [-2.4, 2.4].$$

(c) e (d) $f(x, y) = (y - x^2 + 1)^4 + (x^2 + y^2)^4 - 1$; $B = [-1.2, 1.2] \times [-1.2, 1.2]$.

- Figura 3.3

(a) e (b) $f(x, y) = y^2(a^2 - x^2) - (x^2 + 2\alpha y - a)^2$, $a = 0.75$;
 $B = [-1.1, 1.1] \times [-1.1, 1.1]$.

(c) e (d) $f(x, y) = 4y^2 - (x + 1)^3(1 - x)$;
 $B = [-1.3, 1.2] \times [-1.3, 1.2]$.

- Figura 3.4

1. racionais

- Figura 3.5

(a) e (b) $f(x, y) = \cos(xy) - \text{sen}(x - y) + 1$;
 $B = [-4.4, 4.4] \times [-4.4, 4.4]$

(c) e (d) $f(x, y) = (xy + \cos(x + y))(xy + \text{sen}(x + y))$;
 $B = [-4, 4] \times [-4, 4]$.

Referências Bibliográficas

- [1] D. S. Arnon. Topologically reliable display of algebraic curves. *Computer Graphics*, 17(3):219–227, July 1983.
- [2] K. Atkinson. *An introduction to numerical analysis*. Wiley, 1978.
- [3] Luís Henrique de Figueiredo & Hélio Lopes & João Batista Oliveira. Robust adaptive approximation of implicit curves. In *Proceedings of SIBGRAPI 2001*, pages 10–17, 2001.
- [4] J. Stolfi & L. H. de Figueiredo. *Self-Validated Numerical Methods and Applications*. 21^o Colóquio Brasileiro de Matemática, 1997.
- [5] Y. De Mountaadouin. Resolution of $p(x, y) = 0$. *Computer Aided Design*, 23(9):653–654, Nov. 1991.
- [6] E. L. Allgower & S. Gnutzmann. Simplicial pivoting for mesh generation of implicitly defined surfaces. *Computer Aided Geometric Design*, 8(4):305–325, Oct. 1991.
- [7] J. D. Hobby. Rasterization of nonparametric curves. *ACM Transactions on Graphics*, 9(3):162–277, July 1990.
- [8] E. L. Allgower & P. H. Schmidt. An algorithm for piecewise-linear approximation of an implicitly defined manifold. *SIAM Journal*, 22(2):322–346, April 1985.
- [9] G. Taubin. Distance approximations for rasterizing implicit curves. *ACM Transactions on Graphics*, 13(1):3–42, 1994.
- [10] G. Taubin. Rasterizing algebraic curves and surfaces. *IEEE Computer Graphics and Applications*, (14):14–23, March 1994.

- [11] J. A. Thorpe. *Elementary Topics in Differential Geometry*. Springer-Verlag, New York, 1979.
- [12] M. Van Aken, J. & Novak. Curve-drawing algorithms for raster displays. *ACM Transactions on Graphics*, 4(2):147–169, 1979.
- [13] R. Martin , H. Shou , I. Voiculescu , A. Bowyer & G. Wang. Comparison of interval methods for plotting algebraic curves. *Computer Aided Geometric Design*, (19):553–587, 2002.