4
Cálculo de Estruturas Afins para Isossuperfícies

No capítulo anterior vimos como calcular as estruturas afins de uma superfície implícita $\{p \in \mathbb{R}^3, f(p) = 0\}$ no ponto regular p a partir das derivadas de f até a quarta ordem. Neste capítulo, discutiremos as principais ferramentas que usamos para aplicar esses cálculos em uma superfície implícita extraída de uma grade regular: como aproximar as derivadas, como incorporar o cálculo de estrutura afim no algoritmo *Marching Cubes* (25) e como medir a qualidade dos resultados. Vimos que as fórmulas para as curvaturas afins envolvem todas as derivadas parciais de f até a quarta ordem. Estas 34 derivadas podem ser muito sensíveis ao ruído numérico em f, especialmente aquelas de alta ordem. Quando f é amostrada em uma grade regular, que é o caso comum para descrever objetos geométricos implicitamente, uma escolha comum para obter tais derivadas depende de convoluções discretas.

4.1
Aproximação das Derivadas Discretas

Para calcularmos as derivadas de f usamos uma aproximação da identidade. Dessa maneira, o cálculo das derivadas se reduz a um produto de convoluções.

Definição 8 *A convolução de duas funções* $f, g : \mathbb{R} \to \mathbb{R}^d$ *é a função* $f * g : \mathbb{R} \to \mathbb{R}^d$ *definida por*

$$(f * g)(x) = \int_{\mathbb{R}^d} f(x - y)g(y)dy$$

Seja $\phi(x)$ uma função $C^\infty(\mathbb{R}^d)$ com suporte na bola unitária $||x|| \leq 1$ e $\int \phi(x)dx = 1$.

Definição 9 *Chamamos* aproximação da identidade *a família de funções*

$$\phi_\epsilon(x) = \epsilon^{-d}\phi(x/\epsilon),\ 0 < \epsilon \leq 1.$$

O teorema seguinte justifica o uso da aproximação da identidade no cálculo das derivadas parciais. A demonstração pode ser encontrada em (23).

Teorema 6 *Para toda $f \in L^1(\mathbb{R}^d)$ as funções $f_\epsilon = f * \phi_\epsilon \in L^1 \cap C^\infty$ convergem em norma de L^1 para f quando ϵ tende a zero.*

Observação 6

– *Seja f uma função contínua com suporte compacto em $\mathbb{R}^d$. As funções $f_\epsilon = f * \phi_\epsilon \in C_0^\infty$ e convergem uniformemente em $\mathbb{R}^d$ para f quando ϵ tende a zero.*

– *Uma aproximação da identidade pode ser gerada por*

$$\phi_0(x) = 0, ||x|| > R, \phi_n(x) = n^d \phi_0(nx) \text{ e } \int \phi_0(x)dx = 1.$$

Notemos que se ϕ é uma identidade aproximada e f é uma função, então $\frac{\partial}{\partial x_j}(f * \phi) = f * \frac{\partial \phi}{\partial x_j}$. Além disso, $f * \frac{\partial \phi}{\partial x_j} \rightarrow \frac{\partial f}{\partial x_j}$. Dessa forma, utilizamos em nossos experimentos uma função spline $\sigma(x, y, z)$ como aproximação da identidade (30, 32) definida pela expressão $\sigma(x, y, z) = \sigma_1(x)\,\sigma_1(y)\,\sigma_1(z)$ (ver figura 4.1), onde

$$\sigma_1(x) = \frac{1}{120} \begin{cases} (3 - |x|)^5 - 6(2 - |x|)^5 + 15(1 - |x|)^5 & , \quad 0 \leq |\,x\,| < 1 \\ (3 - |x|)^5 - 6(2 - |x|)^5 & , \quad 1 \leq |\,x\,| < 2 \\ (3 - |x|)^5 & , \quad 2 \leq |\,x\,| < 3 \\ 0 & , \quad |\,x\,| > 3. \end{cases}$$

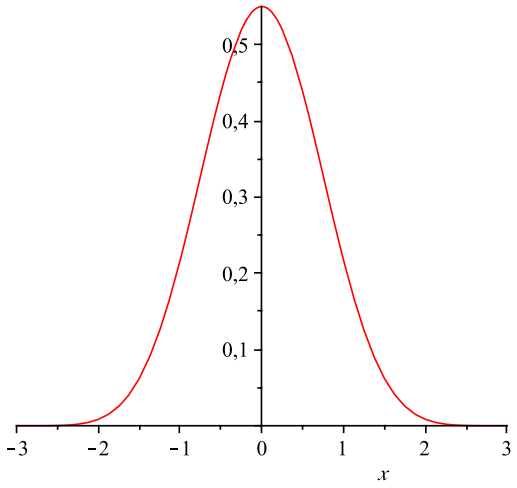

Figura 4.1: Função spline σ_1 de grau 5 em uma variável.

As derivadas são obtidas pela convolução de f com a derivada de normalização do spline $f \approx f * \sigma$ e $\partial^\alpha f \approx f * \left(\frac{1}{c}\partial^\alpha \sigma\right)$. A constante c é determinada para cada ordem de derivação a fim de compensar a escala entre

o domínio $]-3,3[^3$ de σ e o domínio real de f e garantir que as derivadas de monômios de grau α sejam corretamente estimadas (17).[1]

Proposição 5 *Os estimadores obtidos dos vetores co-normal e normal afins e das curvaturas Gaussiana e média afins convergem uniformemente para as funções ν, ξ, $\mathcal{K}$ e $\mathcal{H}$ (em σ_1 tem suporte compacto) quando f é amostrada com densidade indo para infinito, desde que f, f', f'', f''' e $f^{(4)}$ sejam funções integráveis.*

4.2 Implementação dentro do Marching Cubes

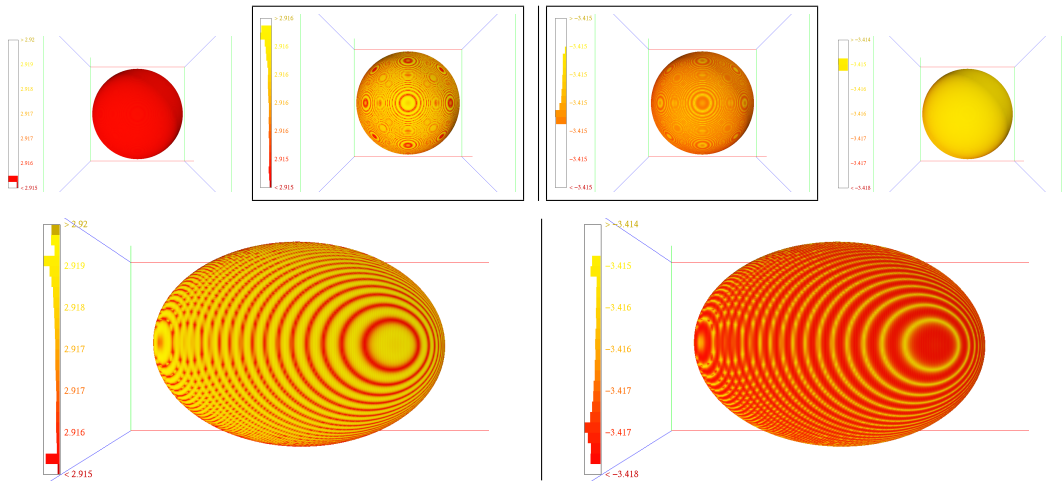

Figura 4.2: Incorporando os estimadores dentro do *Marching Cubes* revela o padrão não-invariante da grade baseado na estimação das derivadas. As curvaturas Gaussiana afim $\mathcal{K}$ (à esquerda) e média afim $\mathcal{H}$ (à direita) antes (em cima) com um aumento da escala e depois (em baixo) com a transformação afim $((0.9,0,0.9),(0,2,0),(1.1,0,0.6))$.

Marching Cubes (25) é o algoritmo base para extração de superfícies implícitas. Ele opera em cada voxel de uma grade regular e, eventualmente, gera alguns triângulos no interior dos voxels. Os vértices dos triângulos são calculados por interpolação linear ao longo das bordas do voxel, gerando 0 ou 1 em cada vértice dos lados, dependendo se os valores da função implícita nas extremidades da borda têm sinais iguais ou diferentes, respectivamente.

Podemos avaliar diretamente as derivadas através da convolução discreta apenas nos vértices do voxel, onde podem ser calculadas as estruturas afins ν, ξ, $\mathcal{K}$ e $\mathcal{H}$ nos cantos do *voxel* (fora da superfície) e interpolar linearmente as estruturas ao longo da borda, ou interpolar as derivadas ao longo da borda e

[1] Ver também http://www.cs.duke.edu/courses/spring03/cps296.1/handouts/Image%20Processing.pdf.

calcular a estrutura afim nos vértices do *Marching Cubes* a partir das derivadas interpoladas. A primeira opção tem a desvantagem do cálculo da estrutura afim fora da superfície, e sem restrição de f, esta estrutura pode ser diferente daquela na superfície.

A segunda opção usa a interpolação linear para todas as derivadas, esta última opção não é totalmente consistente: a interpolação linear de uma derivada de quarta ordem significa interpolar a função em si como um polinômio de grau 4, enquanto que a função é interpolada como polinômio de grau 1. Isto é visível na esfera de figura 4.2, onde há variações de ruídos muito pequenos e estão correlacionados com a estrutura da grade.

Esta última opção de interpolar as derivadas no *Marching Cubes* e em seguida calcular a estrutura afim leva a melhores resultados na prática. Apesar dessa melhoria com a segunda possibilidade o problema ainda não está totalmente resolvido, como foi discutido no parágrafo anterior.

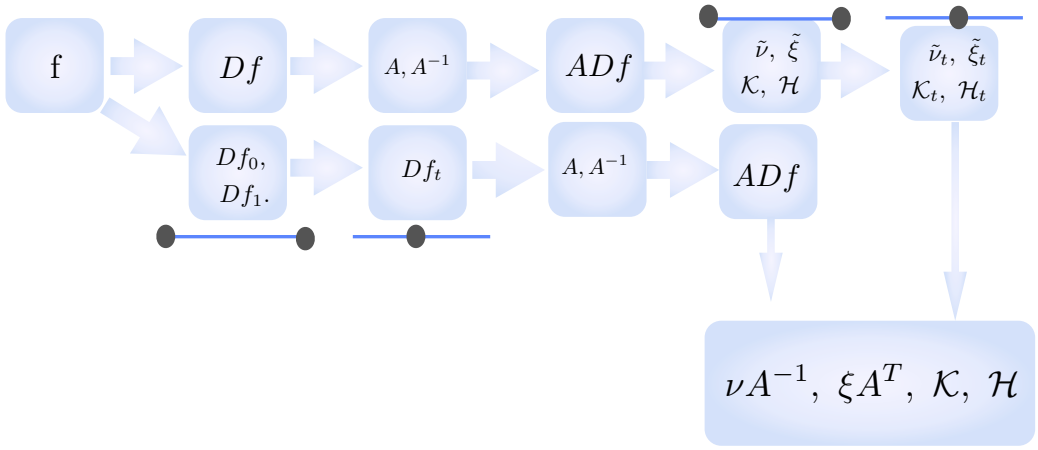

Figura 4.3: Implementação dentro do Marching Cubes. Na primeira linha temos a primeira tentativa e na segunda linha a última tentativa que resultou melhores resultados.

4.3 Estabilidade Numérica

O primeiro passo para usarmos o algoritmo *Marching Cubes* e construírmos o nosso algoritmo é o cálculo das derivadas até a quarta ordem. Feito isso, se formos usar o método *direto* então usamos as fórmulas obtidas diretamente a partir do teorema da função implícita. Por outro lado, ao usarmos o *método com transformação* teremos que calcular inicialmente a transformação A, ou seja, devemos calcular as derivadas de f até a segunda ordem, depois calculamos as transformadas das derivadas (ver equações 3-5), com isso podemos supor que o gradiente é $(0, 0, 1)$ e $\tilde{f}_{xy} = 0$, ver teorema 5. Agora aplicamos

esse resultado nas expressões obtidas do método direto, obtemos as estruturas afins com transformação e por fim usando a contravariância do co-normal afim ν, a covariância do normal afim ξ e a invariância das curvaturas Gaussiana $\mathcal{K}$ e média $\mathcal{H}$ obtemos as expressões mais simplificadas (ver tabela 4.1 e a subseção 3.3.3). As principais etapas estão descritas no algoritmo 1.

Algoritmo 1: Implementação dentro do *Marching Cubes.*

Entrada: Superfície
Saída: Invariantes afins: $\nu, \xi, \mathcal{K}, \mathcal{H}$
bool compute_affine_struct_direct(Df, v)

bool compute_affine_structure_reduction(Df, v)

compute_derivate(Df)

compute_transf(Df, A,Ainv)

transf_derive(Df, A, ADf)

compute_affine_struct_transf(ADf, v)

transform_back_affine_struct(A, Ainv,v)

As fórmulas simplificadas são mais estáveis que o cálculo direto sem transformação, como confirmado em nossos experimentos. Usamos o software Maple para otimizar ambas as fórmulas diretas e com a transformação, visando reduzir o número de operações. A comparação do número de operações nas fórmulas diretas e simplificadas mostra claramente o ganho de estabilidade das fórmulas simplificadas (ver tabela 4.1).

	Matrizes A, A^{-1}	Aplicações de $\partial(\tilde{f})$	Fórmulas simplificadas	Total
Simplificada	749	7.335	1.783	9.867
Direta				23.690

Tabela 4.1: Número de operações de cada passo do estimador para um único ponto. As fórmulas simplificadas são muito mais concisas e são mais intensas computacionalmente nas operações de mapear as derivadas.

A principal ferramenta de derivação para obter as fórmulas das estruturas afins de superfícies implícitas vem do teorema da função implícita, onde todas as derivadas de g são obtidas através de uma divisão por f_z. Portanto, qualquer implementação numérica pode sofrer quando o gradiente é quase zero. Nas fórmulas simplificadas essa instabilidade é confinada na transformação A (em especial na escala não-uniforme S).

Além disso, a métrica Berwald-Blaschke degenera $\mathsf{d} = 0$ quando a curvatura Gaussiana Euclidiana é zero. Em particular, as curvaturas afins devem ser

infinito em pontos de sela, que são delicadas de lidar em um contexto numérico. Um tratamento independente dos pontos de inflexão tem sido proposto para as curvas através de uma cuidadosa reamostragem local (13) e poderia ser estendido para as superfícies em trabalhos futuros. Esta instabilidade permanece no interior da fórmulas simplificadas.

4.4 Medidas de Qualidade

No cálculo da estrutura afim de isossuperfícies o critério mais complicado é o da invariância, pois o processo de amostragem da função implícita f em uma grade regular não é invariante sob aplicação afim (ver figura 4.4). A qualidade de um estimador geométrico é geralmente medido através de quatro critérios: invariância sob mapeamento geométrico, erro em comparação com a medida exata geométrica, convergência para a medida contínua, quando a amostragem se torna mais densa e robustez ao ruído.

Além disso, ao verificar a invariância por meio da comparação dos estimadores afins de uma isossuperfície antes e depois de uma transformação afim, os vértices gerados pelo algoritmo *Marching Cubes* não estão em posições correspondentes e não uniformemente distribuídos. Apesar de tentarmos reduzir essa disparidade nos experimentos com a função implícita analítica adaptando o domínio transformado em uma caixa delimitadora da imagem do domínio original, uma medida com invariância global ainda é difícil de implementar.

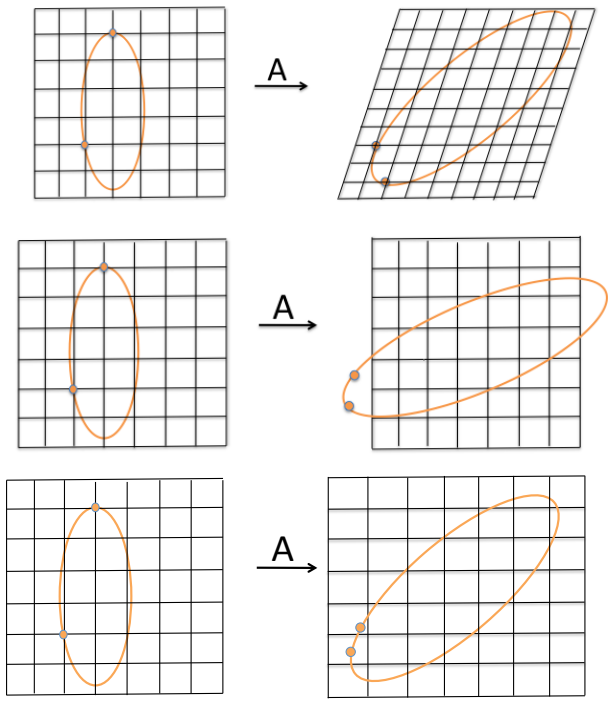

Figura 4.4: Correção do domínio caso bidimensional.

A figura 4.4 ilustra a correção do domínio. Aplicando uma transformação afim na função e no domínio (em cima) modifica o dado de forma invariante, mas gera uma grade não ortogonal. Mantendo uma grade ortogonal modifica o dado de forma não invariante afim (no centro). Aplicando uma transformação afim na função e corrigindo o domínio (em baixo), modifica o dado de forma invariante afim e gera uma grade ortogonal.

Usamos em nosso algoritmo a terceira opção, ou seja, ao aplicarmos uma transformação afim B na superfície corrigimos a caixa delimitadora da superfície calculando os valores máximos e mínimos das coordenadas depois da transformação B. Supomos, sem perda de generalidade, que a caixa inicial delimitadora seja um cubo cujo comprimento das arestas seja 1, ou seja, os valores mínimos e máximos das coordenadas iniciais são: $x[0] = x_{min} = 0$, $y[0] = y_{min} = z[0] = z_{min} = 0$ e $x[1] = x_{max} = y[1] = y_{max} = z[1] = z_{max} = 1$. Denotemos por $P_{ijk} = \{(x[i], y[j], z[k]), (i, j, k) \in \{0, 1\}^3\}$ os pontos do cubo e sejam

$$\begin{aligned} \tilde{W}_{min} &= min_{i,j,k}\{\tilde{P}_{ijk}\}, W = x, y, z \; e \; \tilde{P}_{ijk} = B^{-1}P_{\{ijk\}} \\ \tilde{W}_{max} &= max_{i,j,k}\{\tilde{P}_{ijk}\} \end{aligned}$$

os pontos da nova caixa delimitadora. Dessa forma, obtemos a correção do domínio.