

3

Geração Robusta de Pontos para Superfícies Implícitas no \mathbb{R}^4

Neste capítulo serão propostos métodos para a geração de superfícies implícitas no \mathbb{R}^4 . Como resultado da combinação destes métodos apresenta-se uma aproximação da superfície implícita $S = f^{-1}(0,0)$ determinada pela função $f : \Omega \subseteq \mathbb{R}^4 \rightarrow \mathbb{R}^2$. Para tal, assume-se que f é continuamente diferenciável.

3.1

Descrição Geral do Algoritmo

Considere $f : \Omega \subseteq \mathbb{R}^4 \rightarrow \mathbb{R}^2$ uma função continuamente diferenciável. O algoritmo proposto calcula uma aproximação robusta topologicamente e adaptativa geometricamente para a superfície implícita $S = f^{-1}(0,0)$.

Considere um domínio $[\mathbf{h}] \subseteq \mathbb{R}^4$ que é uma hipercaixa $[x_{min}, x_{max}] \times [y_{min}, y_{max}] \times [z_{min}, z_{max}] \times [w_{min}, w_{max}]$. Inicialmente o algoritmo subdivide $[\mathbf{h}]$ usando uma *KD-tree*. O processo de subdivisão usa a *Aritmética Intervalar* para descartar as células da *KD-tree* que são garantidas **não** interceptar a superfície, levando a identificação de todas as células que contém em seu interior pontos pertencentes a superfície. A *Aritmética Intervalar* também promove uma estimativa intervalar para a derivada da função f , sendo assim possível localizar regiões de alta curvatura e ambiguidades topológicas que também serão utilizadas como critérios para subdivisão da *KD-tree*.

Após a subdivisão do domínio $[\mathbf{h}]$ o *Método de Newton* é utilizado para encontrar uma aproximação dos pontos pertencentes a superfície que se encontram sobre as arestas das hipercaixas que não foram descartados no processo de geração da *KD-tree*. Esses pontos formam um conjunto de sementes que serão usadas para o refinamento da visualização da superfície. A geração de novos pontos ocorre sobre os planos tangentes a superfície nas sementes, que depois serão projetadas sobre a superfície fazendo uso de um *operador de projeção aproximado*.

Nas próximas seções serão descritos os detalhes de cada uma dessas etapas.

3.2

Construção da KD-tree

O algoritmo proposto inicia com o já citado domínio $[\mathbf{h}]$ o qual é atribuído a hipercaixa raiz da KD-tree. A cada passo verifica se a célula recorrente n , cujo domínio é a hipercaixa $[\mathbf{h}_n]$, deverá ser subdividida. Em caso positivo, o algoritmo divide a hipercaixa $[\mathbf{h}_n]$ na direção de um dos eixos, X , Y , Z ou W , alternadamente. O algoritmo continua o processo recursivamente sobre as subdivisões da célula n .

Nas próximas subseções será apresentado como é feito o uso da Aritmética Intervalar, os critérios de subdivisão da KD-tree e como esses critérios são combinados em um teste que dará origem ao algoritmo de subdivisão da KD-tree.

3.2.1

Aritmética Intervalar

Considere uma função $f : \Omega \subseteq \mathbb{R}^4 \rightarrow \mathbb{R}^2$ e uma hipercaixa retangular $[\mathbf{h}] \subseteq \Omega$, onde suas dimensões são representadas pelos intervalos $[\mathbf{x}]$, $[\mathbf{y}]$, $[\mathbf{z}]$ e $[\mathbf{w}]$, nas direções dos eixos X , Y , Z e W , respectivamente. Define-se através da aritmética intervalar (2) a *representação intervalar de uma função* (22):

Considere a função $F : \mathbb{IR}^m \rightarrow \mathbb{IR}^n$ como uma extensão intervalar da função $f : \mathbb{R}^m \rightarrow \mathbb{R}^n$, extensão essa feita sobre cada coordenada da f .

A partir disso faz-se possível dizer que:

$$F([\mathbf{h}]) = F([\mathbf{x}], [\mathbf{y}], [\mathbf{z}], [\mathbf{w}]) = (F_1([\mathbf{x}], [\mathbf{y}], [\mathbf{z}], [\mathbf{w}]), F_2([\mathbf{x}], [\mathbf{y}], [\mathbf{z}], [\mathbf{w}])) \subseteq \mathbb{IR}^2.$$

Assim, $F([\mathbf{h}])$ são intervalos que estimam o conjunto de valores resultantes da avaliação da função f na hipercaixa $[\mathbf{h}]$ no sentido que:

$$F([\mathbf{h}]) \supseteq f([\mathbf{h}]) = \{f(x, y, z, w) = (f_1(x, y, z, w), f_2(x, y, z, w)) : (x, y, z, w) \in [\mathbf{h}]\}.$$

Neste caso, pode-se dizer que a aritmética intervalar gera um vetor de intervalos $F([\mathbf{h}])$ que permite analisar se a hipercaixa $[\mathbf{h}]$ não intercepta pontos da superfície implícita $S = f^{-1}(0, 0)$, isto é:

$$(0, 0) \notin F([\mathbf{h}]) \implies (0, 0) \notin f([\mathbf{h}]). \quad (3-1)$$

Este fato é utilizado como um dos principais testes de subdivisão da KD-tree, como poderá ser visto na seção 3.2.2.

Todos os cálculos feitos na avaliação intervalar da função f são baseados nas funções estendidas para intervalos apresentadas na seção 2.0.6.

É necessário observar que a implicação 3-1 garante que a hipercaixa $[\mathbf{h}]$ não intercepta a superfície, mas a recíproca nem sempre é verdadeira, ou seja,

o fato de $(0, 0) \in F([\mathbf{h}])$ não garante que $(0, 0) \in f([\mathbf{h}])$, pois $F([\mathbf{h}])$ pode ser estritamente maior que $f([\mathbf{h}])$.

3.2.2

Critérios de Subdivisão

Para que uma célula seja subdividida ela precisará satisfazer quatro critérios: *critério da componente conexa*, *critério topológico*, *critério geométrico* e *critério de nível máximo*, onde os três primeiros foram adaptados de Paiva et al. (21).

Critério da Componente Conexa

Esse critério avalia se a célula recorrente n pode conter pontos da superfície $S = f^{-1}(0, 0)$. Para tal, utiliza-se a avaliação intervalar $F([\mathbf{h}_n])$ da função f na hipercaixa $[\mathbf{h}_n]$. Como visto na seção 3.2.1, se $(0, 0) \notin F([\mathbf{h}_n])$ pode-se garantir que não há pontos da superfície na hipercaixa recorrente $[\mathbf{h}_n]$. Isto significa dizer que não será descartada nenhuma hipercaixa que contenha pontos da superfície S , pois:

$$(0, 0) \notin F([\mathbf{h}_n]) \supseteq f([\mathbf{h}_n]) \implies (0, 0) \notin f([\mathbf{h}_n]) \implies (0, 0) \neq f(x, y, z, w),$$

$$\forall (x, y, z, w) \in [\mathbf{h}_n]$$

Por outro lado, se $(0, 0) \in F([\mathbf{h}_n])$ há possibilidade da superfície interceptar a caixa $[\mathbf{h}_n]$, logo o teste a ser feito é:

$$\text{Se } (0, 0) \in F([\mathbf{h}_n]) \text{ então } \textit{subdivide } [\mathbf{h}_n]$$

Ou seja, se os intervalos resultantes da avaliação intervalar $F([\mathbf{h}_n]) = (F_1([\mathbf{h}_n], F_2([\mathbf{h}_n]))$ contêm o zero, então deve-se subdividir a célula n . Este teste, fazendo uso da Aritmética Intervalar na avaliação da função, garante a robustez topológica da superfície, isto é, garante que nenhum componente da superfície será perdido.

Critério Topológico

O *critério da componente conexa*, citado anteriormente, não garante sozinho a preservação da topologia de uma superfície implícita que contenha túneis ou buracos. Mas, se aplicado conjuntamente com o *critério topológico* pode-se garantir que as partes vazias desses túneis ou buracos serão descartadas.

Se uma hipercaixa $[\mathbf{h}_n]$ contém um túnel ou buraco em seu interior então os vetores gradiente n_1 e n_2 das funções f_1 e f_2 no ponto (x, y, z, w) irão variar

para $-n_1$ e $-n_2$ no ponto oposto a (x, y, z, w) , fazendo com que cada um dos intervalos que correspondem as coordenadas de $\nabla F_1([\mathbf{h}_n])$ e $\nabla F_2([\mathbf{h}_n])$, resultante da avaliação intervalar do gradiente da f_1 e da f_2 na hipercaixa $[\mathbf{h}_n]$, contenham o zero. Portanto tem-se:

Se $[(0, 0, 0, 0) \in \nabla F_1([\mathbf{h}_n]) \text{ e } (0, 0, 0, 0) \in \nabla F_2([\mathbf{h}_n])]$ **então** *subdivide* $[\mathbf{h}_n]$

Critério Geométrico

Para aproximar corretamente a geometria da superfície implícita S , com uma boa amostragem de pontos, foi preciso estudar a curvatura da superfície na hipercaixa recorrente $[\mathbf{h}_n]$. Para tal, estimou-se a curvatura pela variação do gradiente de cada componente, F_1 e F_2 , nesta hipercaixa. Se a curvatura naquela região for alta então o comprimento dos intervalos correspondentes as coordenadas de $\nabla F_1([\mathbf{h}_n])$ ou $\nabla F_2([\mathbf{h}_n])$ serão grandes, logo deve-se subdividir tal hipercaixa buscando uma melhor amostragem de pontos. O critério escolhido foi:

$$K = \max\{\text{Diam}([\mathbf{x}_1], [\mathbf{y}_1], [\mathbf{z}_1], [\mathbf{w}_1]), \text{Diam}([\mathbf{x}_2], [\mathbf{y}_2], [\mathbf{z}_2], [\mathbf{w}_2])\}$$

$$= \max\{\max\{|\mathbf{x}_1|, |\mathbf{y}_1|, |\mathbf{z}_1|, |\mathbf{w}_1|\}, \max\{|\mathbf{x}_2|, |\mathbf{y}_2|, |\mathbf{z}_2|, |\mathbf{w}_2|\}\},$$

onde $([\mathbf{x}_1], [\mathbf{y}_1], [\mathbf{z}_1], [\mathbf{w}_1])$ e $([\mathbf{x}_2], [\mathbf{y}_2], [\mathbf{z}_2], [\mathbf{w}_2])$ são as coordenadas intervalares de $\frac{\nabla F_1([\mathbf{h}_n])}{\max\{\nabla F_1([\mathbf{h}_n])\}}$ e $\frac{\nabla F_2([\mathbf{h}_n])}{\max\{\nabla F_2([\mathbf{h}_n])\}}$, respectivamente. O limiar da curvatura, K_{max} , é um parâmetro de entrada do usuário. Portanto, o critério geométrico é:

Se $K > K_{max}$ **então** *subdivide* $[\mathbf{h}_n]$

Para que a geometria da superfície seja preservada e a amostragem dos pontos seja adequada é necessário ajustar o parâmetro K_{max} de acordo com a superfície, ou seja, busca-se um parâmetro adequado para que em regiões de alta curvatura a KD-tree realize mais subdivisões e a amostragem dos pontos seja mais densa.

Em superfícies no \mathbb{R}^4 pode ser difícil identificar visualmente regiões de alta curvatura. Nos resultados obtidos nas figuras 3.1 e 3.2, as superfícies ganham um maior destaque nessas regiões com uma maior concentração de pontos proporcionada pelo ajuste do parâmetro K_{max} .

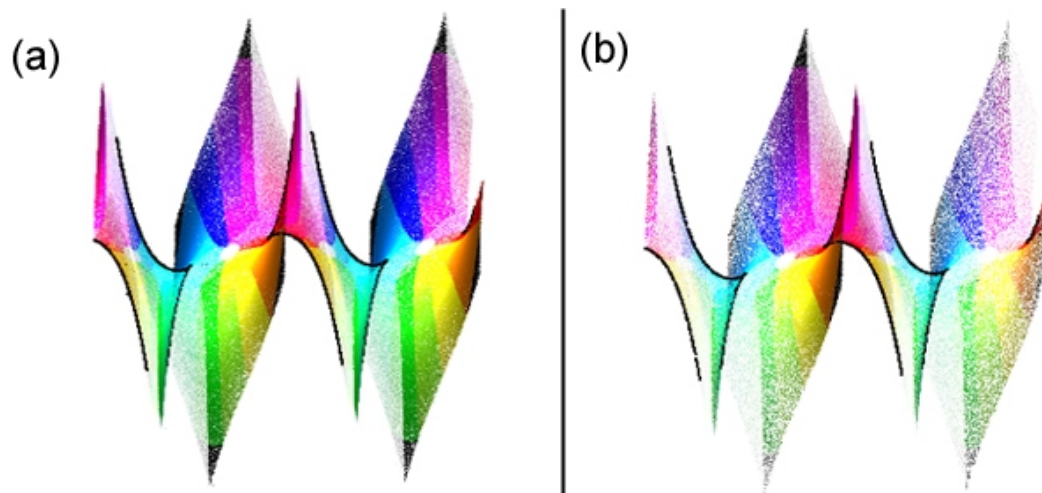


Figura 3.1: Amostragem da variação do parâmetro K_{max} para superfície implícita definida pela equação complexa $\mathbf{w} = \cos(\mathbf{z})$. (a) $K_{max} = 0.5$, (b) $K_{max} = 1.2$.

Critério de Nível Máximo

Os critérios citados anteriormente podem fazer com que ocorram um grande número de subdivisões, até mesmo infinitas. Além disso, a avaliação intervalar $F([\mathbf{h}_n])$ contém a imagem da $f([\mathbf{h}_n])$ com um erro numérico de arredondamento, o que pode gerar subdivisões desnecessárias em regiões vazias, isto é, que não contenham componentes da superfície, quando a precisão numérica diminui. Para evitar este fato é estabelecido um nível máximo $depth_{max}$ para a KD-tree, ou seja, o algoritmo subdivide a KDtree enquanto o nível da hipercaixa recorrente $depth_n$, for menor que o nível máximo, como abaixo:

Se $depth_n < depth_{max}$ então subdivide $[\mathbf{h}_n]$

3.2.3

Algoritmo de Subdivisão

No algoritmo proposto, os critérios de subdivisão são combinados em um teste que garantirá quando a hipercaixa recorrente $[\mathbf{h}_n]$ não possui a componente da superfície implícita S , fazendo com que ocorra uma boa aproximação da mesma. Tal fato garante a robustez topológica da superfície e permite parar a subdivisão antes do nível máximo $depth_{max}$, caso a aproximação esperada seja alcançada, o que acontece em todos os exemplos apresentados neste trabalho, como pode ser visto na tabela A.

O teste de subdivisão é:

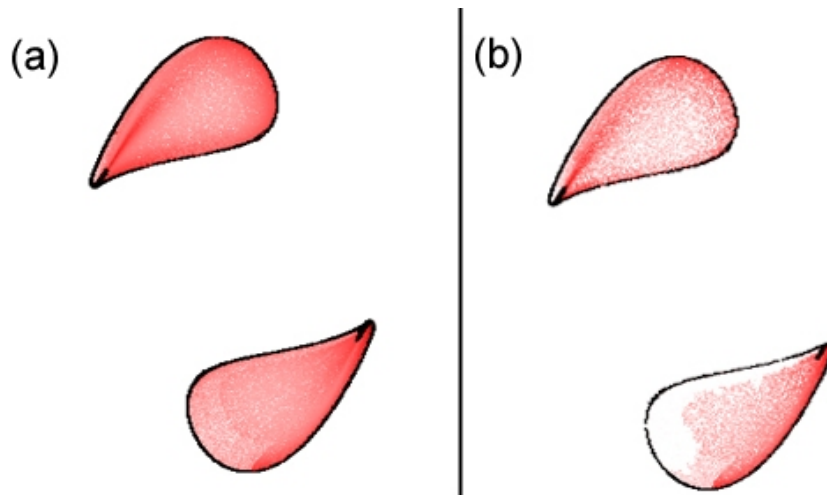


Figura 3.2: Amostragem da variação do parâmetro K_{max} para a superfície implícita definida pela função $f(x, y, z, w) = ((y - 0.2w)^2 + z^2 - 1, x^2 + y^2 + (z + w)^2 - 0.49)$. (a) $K_{max} = 0.5$, (b) $K_{max} = 1.2$.

Se

$$(0, 0) \in F([\mathbf{h}_n]) \text{ e}$$

$$[(0, 0, 0, 0) \in dF([\mathbf{h}_n]) \text{ ou } K > K_{max}] \text{ e}$$

$$depth_n < depth_{max}$$

então

subdivide $[\mathbf{h}_n]$

Uma vez passado no teste acima é necessário definir como será subdividida a hipercaixa recorrente $[\mathbf{h}_n]$, para tal faz-se uso da profundidade $depth_n$ da hipercaixa e são realizados os seguintes testes:

Se $(depth_n \% 4) = 0$ então *subdivide* $[\mathbf{h}_n]$ ao meio ortogonalmente ao eixo X

Se $(depth_n \% 4) = 1$ então *subdivide* $[\mathbf{h}_n]$ ao meio ortogonalmente ao eixo Y

Se $(depth_n \% 4) = 2$ então *subdivide* $[\mathbf{h}_n]$ ao meio ortogonalmente ao eixo Z

Se $(depth_n \% 4) = 3$ então *subdivide* $[\mathbf{h}_n]$ ao meio ortogonalmente ao eixo W

Desta forma, a hipercaixa raiz, que está na profundidade 0 da KD-tree, é dividida ao meio por um plano ortogonal ao eixo X; os filhos da hipercaixa raiz, que se encontram na profundidade 1, ambos são divididos ao meio por um plano ortogonal ao eixo Y; os filhos dos filhos se encontrando na profundidade 2 são divididos ao meio por um plano ortogonal ao eixo Z; no próximo nível de profundidade o plano de divisão será ortogonal ao eixo W, no seguinte retornará um plano ortogonal ao eixo X, e assim por diante, como mostra a figura 3.3.

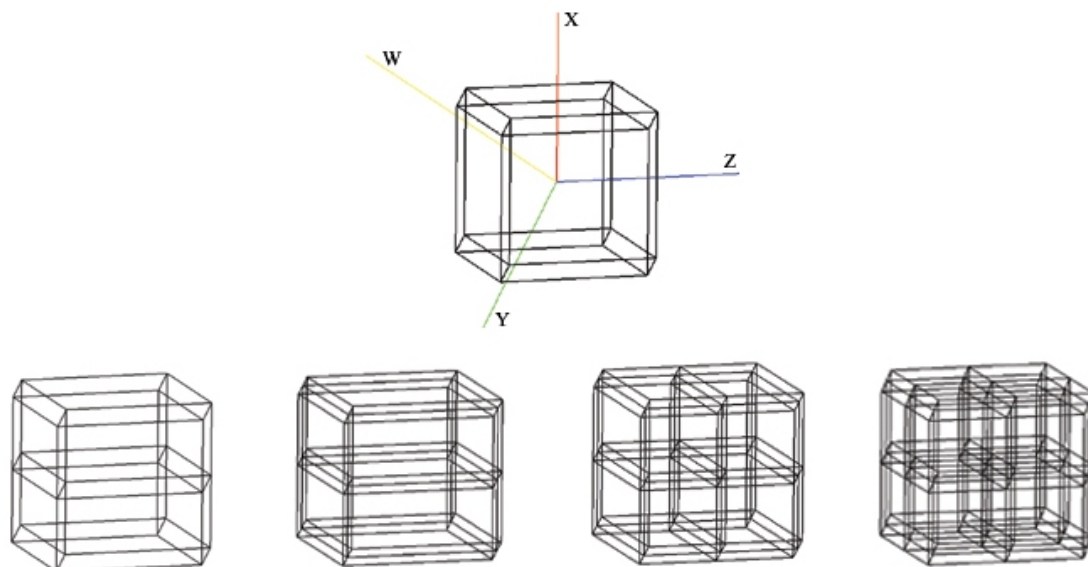


Figura 3.3: Hipercaixas projetadas sobre o plano XY e originadas da subdivisão da KD-tree nos níveis de profundidade 0 (acima), 1, 2, 3 e 4 (abaixo), respectivamente.

Este algoritmo conduz a um equilíbrio da KD-tree, em que cada hipercaixa que se encontra na profundidade máxima está aproximadamente a mesma distância da hipercaixa raiz.

3.3 Geração dos Pontos

Uma vez construída a KD-tree, a próxima etapa é o processamento das folhas que tenham passado no Critério da Componente Conexa, visto na seção 3.2.2. Como estas hipercaixas não foram descartadas durante a construção da árvore, então existe a possibilidade de conterem pontos pertencentes a superfície implícita $S = f^{-1}(0, 0)$. Nesta seção será descrito como foi encontrado um conjunto inicial de pontos (sementes) e a geração de novos pontos sobre a superfície.

3.3.1 Geração das Sementes

A geração dos pontos sementes é importante para definir a qualidade da visualização da superfície em questão, já que a geração dos demais pontos será proveniente da geração destes.

A estratégia usada para encontrar as sementes é achar uma aproximação, através do *Método de Newton*, para os pontos da superfície que se encontram sobre as arestas das hipercaixas, ou seja, os pontos de interseção da superfície

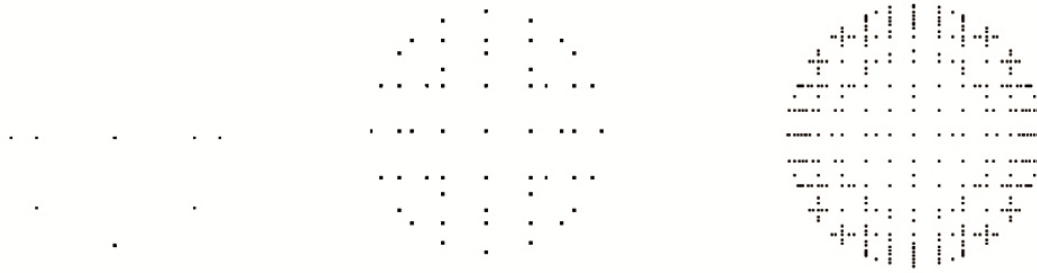


Figura 3.4: Geração dos pontos sementes da superfície implícita $f(x, y, z, w) = (x^2 + y^2 + z^2 + w^2 - 4, x^2 + y^2 + z^2 + (w - 1)^2 - 4)$ nos níveis de profundidade da KD-tree 5, 10 e 15, respectivamente.

implícita com essas hipercaixas. Assim, a quantidade de sementes é proporcional a profundidade máxima da KD-tree, como pode ser visto na figura 3.4. Nos experimentos feitos neste trabalho, em geral, a profundidade máxima varia em torno de 20.

Em seguida será descrito como é feita a determinação dos vértices e arestas de uma hipercaixa e como o Método de Newton é aplicado para a determinação dos pontos sementes.

Determinação dos Vértices e Arestas de uma Hipercaixa

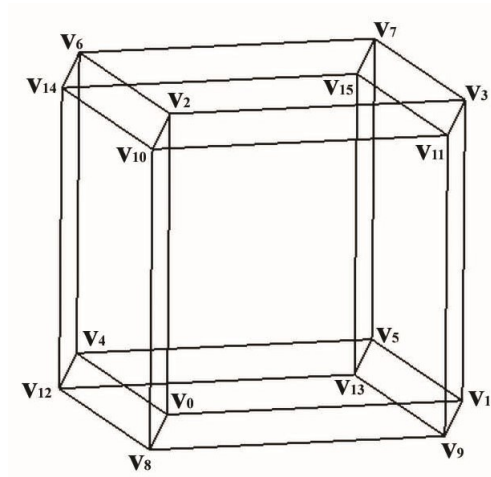


Figura 3.5: Hipercaixa com indicação dos vértices e projetada sobre o plano XY

Uma hipercaixa retangular $[\mathbf{h}] \subseteq \mathbb{R}^4$ é composta por 16 vértices e 32 arestas, onde suas dimensões são representadas pelos intervalos $[\mathbf{x}]$, $[\mathbf{y}]$, $[\mathbf{z}]$ e $[\mathbf{w}]$. Considere $\inf([\mathbf{x}])$, $\inf([\mathbf{y}])$, $\inf([\mathbf{z}])$ e $\inf([\mathbf{w}])$, os ínfimos dos intervalos $[\mathbf{x}]$, $[\mathbf{y}]$, $[\mathbf{z}]$ e $[\mathbf{w}]$, respectivamente, e $\sup([\mathbf{x}])$, $\sup([\mathbf{y}])$, $\sup([\mathbf{z}])$ e $\sup([\mathbf{w}])$, os

supremos dos intervalos $[\mathbf{x}]$, $[\mathbf{y}]$, $[\mathbf{z}]$ e $[\mathbf{w}]$, respectivamente. A matriz $V_{16 \times 4}$ lista as 4 coordenadas dos 16 vértices da hipercaixa $[\mathbf{h}]$.

$$V = \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \\ v_{10} \\ v_{11} \\ v_{12} \\ v_{13} \\ v_{14} \\ v_{15} \end{pmatrix} = \begin{pmatrix} \inf([\mathbf{x}]) & \inf([\mathbf{y}]) & \inf([\mathbf{z}]) & \inf([\mathbf{w}]) \\ \sup([\mathbf{x}]) & \inf([\mathbf{y}]) & \inf([\mathbf{z}]) & \inf([\mathbf{w}]) \\ \inf([\mathbf{x}]) & \inf([\mathbf{y}]) & \sup([\mathbf{z}]) & \inf([\mathbf{w}]) \\ \sup([\mathbf{x}]) & \inf([\mathbf{y}]) & \sup([\mathbf{z}]) & \inf([\mathbf{w}]) \\ \inf([\mathbf{x}]) & \sup([\mathbf{y}]) & \inf([\mathbf{z}]) & \inf([\mathbf{w}]) \\ \sup([\mathbf{x}]) & \sup([\mathbf{y}]) & \inf([\mathbf{z}]) & \inf([\mathbf{w}]) \\ \inf([\mathbf{x}]) & \sup([\mathbf{y}]) & \sup([\mathbf{z}]) & \inf([\mathbf{w}]) \\ \sup([\mathbf{x}]) & \sup([\mathbf{y}]) & \sup([\mathbf{z}]) & \inf([\mathbf{w}]) \\ \inf([\mathbf{x}]) & \inf([\mathbf{y}]) & \inf([\mathbf{z}]) & \sup([\mathbf{w}]) \\ \sup([\mathbf{x}]) & \inf([\mathbf{y}]) & \inf([\mathbf{z}]) & \sup([\mathbf{w}]) \\ \inf([\mathbf{x}]) & \inf([\mathbf{y}]) & \sup([\mathbf{z}]) & \sup([\mathbf{w}]) \\ \sup([\mathbf{x}]) & \inf([\mathbf{y}]) & \sup([\mathbf{z}]) & \sup([\mathbf{w}]) \\ \inf([\mathbf{x}]) & \sup([\mathbf{y}]) & \inf([\mathbf{z}]) & \sup([\mathbf{w}]) \\ \sup([\mathbf{x}]) & \sup([\mathbf{y}]) & \inf([\mathbf{z}]) & \sup([\mathbf{w}]) \\ \inf([\mathbf{x}]) & \sup([\mathbf{y}]) & \sup([\mathbf{z}]) & \sup([\mathbf{w}]) \\ \sup([\mathbf{x}]) & \sup([\mathbf{y}]) & \sup([\mathbf{z}]) & \sup([\mathbf{w}]) \end{pmatrix} \quad (3-2)$$

Porém, é preciso saber também quais as ligações que estes vértices fazem para formar as arestas da hipercaixa $[\mathbf{h}]$. Analisando a figura 3.5 é possível determinar a matriz $A_{32 \times 2}$ que lista as ligações vértice-vértice realizadas para a formação das 32 arestas da hipercaixa $[\mathbf{h}]$:

$$A = \begin{pmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \\ A_8 \\ A_9 \\ A_{10} \\ A_{11} \\ A_{12} \\ A_{13} \\ A_{14} \\ A_{15} \\ A_{16} \\ A_{17} \\ A_{18} \\ A_{19} \\ A_{20} \\ A_{21} \\ A_{22} \\ A_{23} \\ A_{24} \\ A_{25} \\ A_{26} \\ A_{27} \\ A_{28} \\ A_{29} \\ A_{30} \\ A_{31} \end{pmatrix} = \begin{pmatrix} v_0 & v_1 \\ v_0 & v_2 \\ v_0 & v_4 \\ v_0 & v_8 \\ v_1 & v_3 \\ v_1 & v_5 \\ v_1 & v_9 \\ v_2 & v_3 \\ v_2 & v_6 \\ v_2 & v_{10} \\ v_3 & v_7 \\ v_3 & v_{11} \\ v_4 & v_5 \\ v_4 & v_6 \\ v_4 & v_{12} \\ v_5 & v_7 \\ v_5 & v_{13} \\ v_6 & v_7 \\ v_6 & v_{14} \\ v_7 & v_{15} \\ v_8 & v_9 \\ v_8 & v_{10} \\ v_8 & v_{12} \\ v_9 & v_{11} \\ v_9 & v_{13} \\ v_{10} & v_{11} \\ v_{10} & v_{14} \\ v_{11} & v_{15} \\ v_{12} & v_{13} \\ v_{12} & v_{14} \\ v_{13} & v_{15} \\ v_{14} & v_{15} \end{pmatrix} \quad (3-3)$$

Método de Newton

Uma vez determinados os vértices e arestas de uma folha da KD-tree determina-se os pontos de interseção da superfície com as arestas.

Para cada aresta a da hipercaixa, com vértices representados por A_0 e A_1 , considere $g : \mathbb{R} \rightarrow \mathbb{R}^4$ definida por $g(t) = A_0 + t(A_1 - A_0)$. Seja $h : \mathbb{R} \rightarrow \mathbb{R}^2$ a composição de f com g , tal que:

$$h(t) = f(g(t)) = (f_1(g(t)), f_2(g(t))) = (f_1(A_0 + t(A_1 - A_0)), f_2(A_0 + t(A_1 - A_0))),$$

onde $t \in [0, 1]$.

Para encontrar o ponto que a superfície $S = f^{-1}(0, 0)$ intercepta a aresta a é preciso que:

$$h(t) = 0 \iff (f_1(A_0 + t(A_1 - A_0)), f_2(A_0 + t(A_1 - A_0))) = (0, 0).$$

Para encontrar a raiz da função h optou-se por usar o *Método de Newton*, pois a função implícita proporciona calcular facilmente as suas derivadas parciais. Logo, obtém-se:

$$t_{k+1} = t_k - J_h^{-1}(t_k) \cdot h(t_k),$$

onde J_h é a matriz *Jacobiana* de h e J_h^{-1} é a pseudo-inversa de J_h .

Para o chute inicial do Método de Newton foi escolhido $t = 0$, isto é, a iteração começa sobre o vértice A_0 .

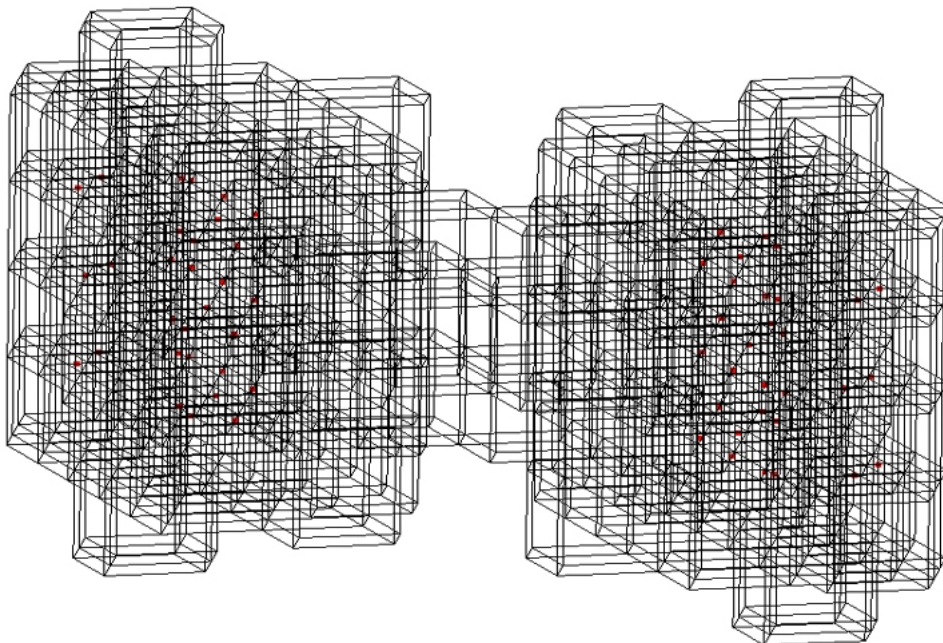


Figura 3.6: Amostragem das sementes sobre uma hipercaixa folha da KD-tree.

O processo é repetido sobre cada aresta e recursivamente sobre todas as hipercaixas que são folhas da KD-tree, gerando um conjunto de sementes sobre as arestas, como mostra a figura 3.6.

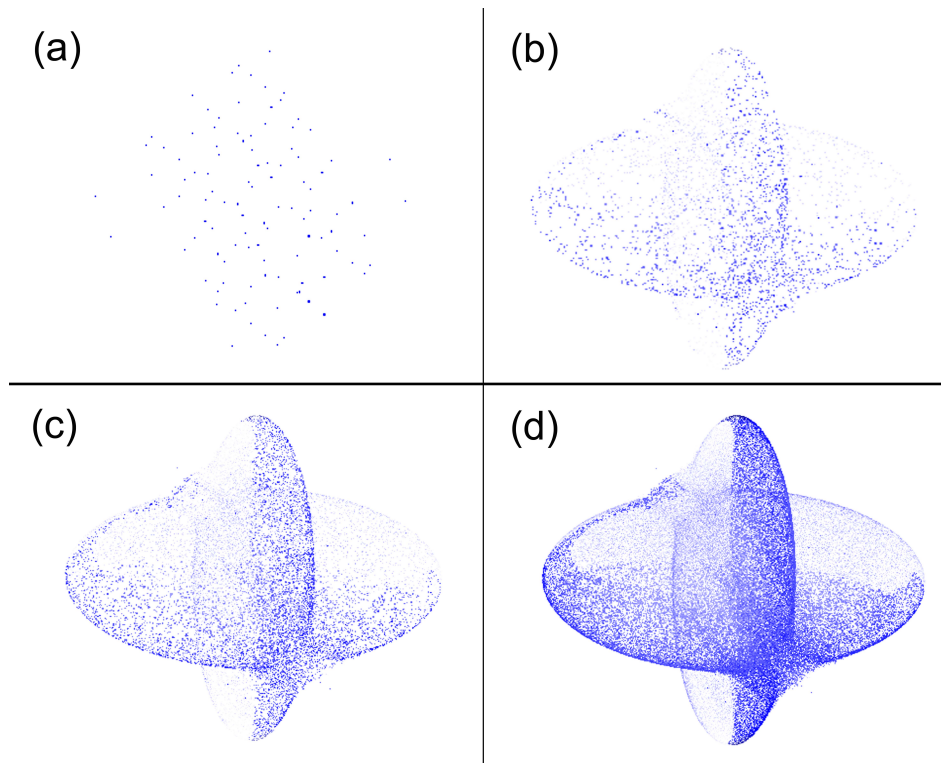


Figura 3.7: Comparação da superfície definida pela função implícita $f(x, y, z, w) = (x^2 + z^2 + w^2 - 0.64, y^2 + z^2 + w^2 - 0.64)$ com diversos refinamentos. (a) pontos sementes; (b) visualização da superfície com 1 refinamento; (c) dois refinamentos; (d) 3 refinamentos.

3.3.2

Refinamento

Após a geração das sementes, prossegue-se com um processo de refinamento que consiste em gerar novos pontos a partir do posicionamento das sementes fazendo uso da técnica *Stippling points* e do *Operador de projeção aproximado*, adaptados de Brazil et al. (3), como pode ser visto na figura 3.7.

Geração de novos pontos

Para gerar novos pontos a partir de um ponto semente P , utiliza-se plano tangente a superfície nesse ponto. Para isso foi utilizada uma base de vetores ortogonais ao plano normal a superfície, obtida através da decomposição QR da matriz $N_{4 \times 2}$ composta pelos vetores $n1$ e $n2$ normais à superfície no ponto semente:

$$N = \begin{pmatrix} n1_x & n2_x \\ n1_y & n2_y \\ n1_z & n2_z \\ n1_w & n2_w \end{pmatrix} = Q_{4 \times 4} R_{4 \times 2}$$

As duas últimas colunas da matriz Q correspondem a base ortogonal desejada, que serão denotadas por u e v . Em seguida, para cada ponto semente P são gerados novos pontos P_1, P_2, P_3 e P_4 , como segue abaixo:

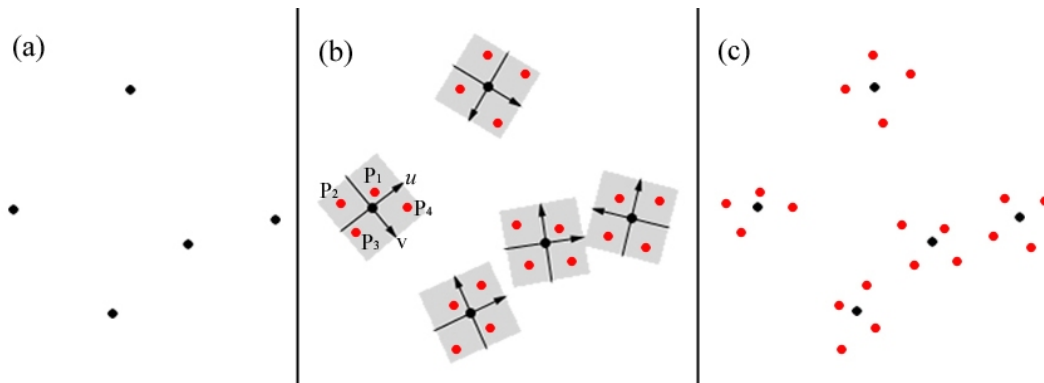


Figura 3.8: Representação da geração de novos pontos: **(a)** Pontos sementes, **(b)** em vermelho, pontos gerados; quadrado, representação de parte dos planos tangentes a superfície nas sementes e **(c)** resultado de um de refinamento.

$$\begin{aligned} P_1 &= P + R\mu_{11}u + R\mu_{12}v \\ P_2 &= P - R\mu_{21}u + R\mu_{22}v \\ P_3 &= P - R\mu_{31}u - R\mu_{32}v \\ P_4 &= P + R\mu_{41}u - R\mu_{42}v \end{aligned}$$

onde R é a medida da diagonal principal da hipercaixa em que o ponto P se encontra e $\mu_{ij} \sim U(0, 1)$, ou seja, μ_{ij} é uma variável aleatória com distribuição uniforme no intervalo $[0, 1]$, com $i = 1, 2, 3, 4$ e $j = 1, 2$.

A ideia central é selecionar um ponto em cada quadrante do plano tangente. A figura 3.8 ilustra esse processo.

Projeção sobre a superfície

Após a colocação de novos pontos é necessário projetá-los sobre a superfície, para isso o algoritmo proposto faz uso de uma adaptação do *Operador de projeção aproximado* (3), que usa um método iterativo baseado no *steepest descent* com redução do tamanho do passo, como ilustra a figura 3.9. Diante disso buscou-se:

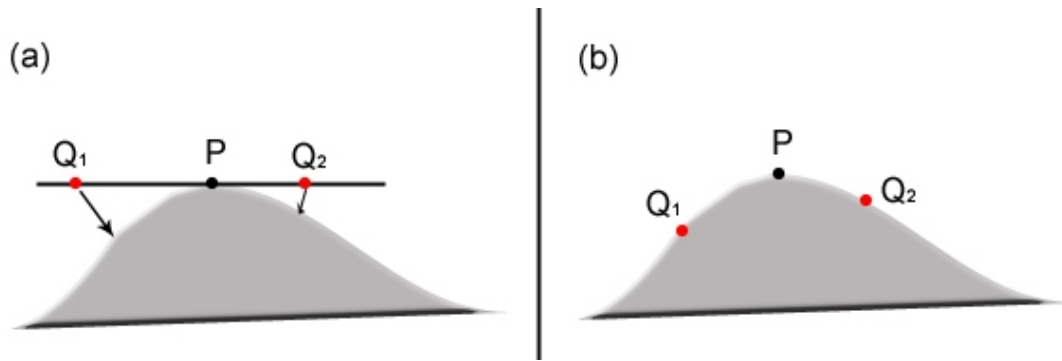


Figura 3.9: Representação da projeção dos novos pontos gerados sobre a superfície: **(a)** Em preto, ponto semente; em vermelho pontos sobre o plano tangente e antes da projeção, **(b)** em vermelho, pontos depois da projeção.

$$\min \bar{f}(x, y, z, w)^2, \text{ onde } \bar{f} = \|f\| = \sqrt{f_1^2 + f_2^2}.$$

Para isso, dado um ponto $Q \in \mathbb{R}^4$ sobre o plano tangente aplica-se:

$$Q^0 = Q \\ Q^{k+1} = Q^k - \delta^{i_k} \frac{\bar{f}(Q^k)}{\|\nabla \bar{f}(Q^k)\|^2} \nabla \bar{f}(Q^k),$$

onde i_k é o menor inteiro não negativo tal que:

$$\bar{f}^2(Q^{k+1}) \leq \bar{f}^2(Q^k)[1 - 2\sigma\delta^{i_k}]$$

e, $\delta \in (0, 1)$ e $\sigma \in (0, \frac{1}{2})$ são dois parâmetros.

No algoritmo proposto neste trabalho usou-se $\delta = 0.1$ e $\sigma = 0.01$ e a iteração é encerrada quando $|\bar{f}(Q^k)| < 10^{-6}$ ou $k \geq 16$, ou seja, quando o ponto for projetado sobre a superfície com uma precisão de 10^{-6} . Quando a sequência superar 16 iterações optou-se por descartar o ponto, fato que não aconteceu em nenhum dos exemplos implementados neste trabalho.

A figura 3.10 mostra o resultado da projeção obtida em dois exemplos.

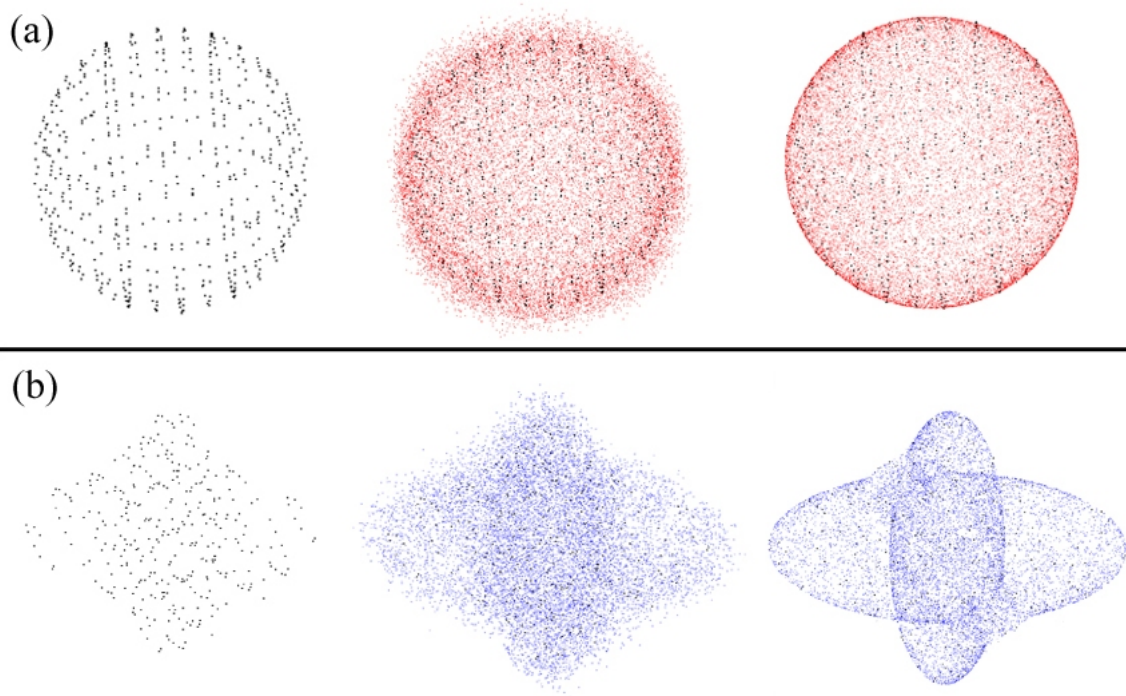


Figura 3.10: Passo a passo do processo de refinamento da superfície. A esquerda, pontos sementes; ao centro, novos pontos colocados antes de serem projetados sobre a superfície; a direita, pontos já projetados sobre a superfície. **(a)** Superfície implícita $f(x, y, z, w) = (x^2 + y^2 + z^2 + w^2 - 4, x^2 + y^2 + z^2 + (w - 1)^2 - 4)$ e **(b)** Superfície implícita $f(x, y, z, w) = (x^2 + z^2 + w^2 - 0.64, y^2 + z^2 + w^2 - 0.64)$

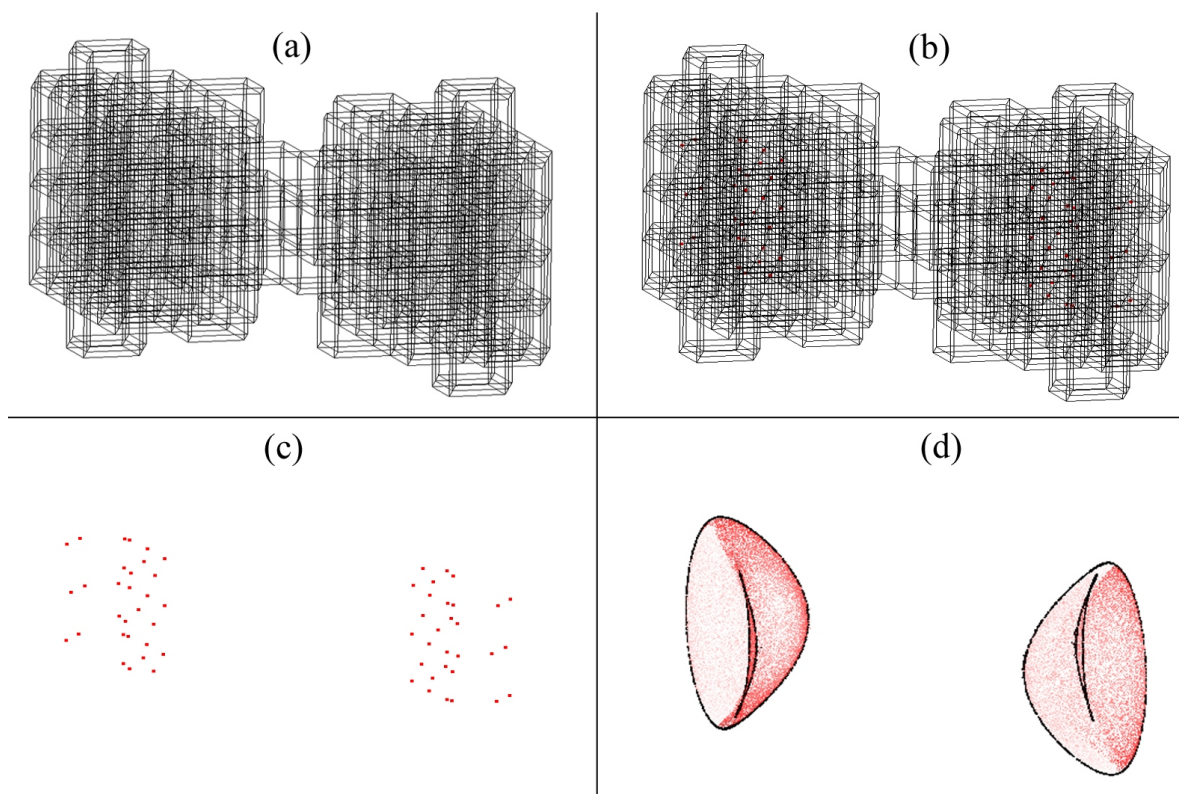


Figura 3.11: Processo de aproximação da superfície implícita $f(x, y, z, w) = ((y - 0.2w)^2 + z^2 - 1, x^2 + y^2 + (z + w)^2 - 0.49)$.

E, a figura 3.11 mostra o processo de aproximação de uma superfície. Primeiramente o algoritmo subdivide o domínio através de 16 passos da KD-tree, como mostrado na imagem **(a)** as hipercaixas da KD-tree que passaram nos testes de subdivisão expostos na seção 3.2.2; após o algoritmo calcula e amostra os pontos (sementes) que estão sobre as arestas de cada hipercaixa, como pode ser visto na imagem **(b)**; a imagem **(c)** retrata todos os pontos sementes da superfície em questão; a imagem **(d)** mostra a superfície após 4 níveis de refinamento.