Caminhando sobre uma Interseção de Superfícies com Passos Circulares

LENIMAR NUNES ANDRADE
WU, SHIN - TING

UNICAMP – Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Departamento de Engenharia de Computação e Automação Industrial (DCA)
Grupo de Computação de Imagens (GCI)
C.P. 6101
13083-970 Campinas, SP, Brasil
lenimar,ting@dca.fee.unicamp.br

Abstract. This paper presents an alternative way to calculate the next approximate point in marching techniques for the computation of the intersection of two parametric surfaces. Differing from the classical methods, the algorithm is based on the approximate osculating circle instead of tangent vector to estimate the next point. It provides closer estimation and a larger marching step in each iteration with relatively low computational cost.

1 Introdução

A determinação da interseção entre duas superfícies é um importante e difícil problema em Modelagem Geométrica. Se as superfícies forem definidas por equações paramétricas

$$\begin{aligned} F(u,v) &= (f_1(u,v), f_2(u,v), f_3(u,v)) \quad \text{e} \\ G(u,v) &= (g_1(u,v), g_2(u,v), g_3(u,v)) \end{aligned}$$

então teoricamente a interseção de F e G corresponde à solução do sistema não-linear de 3 equações a 4 variáveis

$$\begin{cases} f_1(u,v) = g_1(r,s) \\ f_2(u,v) = g_2(r,s) \\ f_3(u,v) = g_3(r,s) \end{cases}$$

Este sistema pode não ter solução (quando as duas superfícies não se interceptarem), ter uma única solução (no caso em que elas se tangenciam em um ponto) ou uma infinidade de soluções (cuja interpretação geométrica pode ser um conjunto de pontos isolados, uma curva ou parte de uma superfície).

Existem alguns métodos básicos para a determinação da interseção de F e G quando essa interseção for uma curva. Dois métodos eficientes que fornecem pontos aproximados para este tipo de problema são o Método da Subdivisão (*subdivision*) e o Método da Caminhada (*marching*). O Método da Subdivisão é um método de natureza global que consiste na subdivisão de todo o domínio das parametrizações, de modo que pequenas partes das superfícies sejam aproximadas por pequenos planos. Desse modo, a interseção superfície/superfície reduz-se a muitas interseções plano/plano [Houghton–Emnett (1985)]. O Método da Caminhada é de natureza local e consiste em se obter um ponto da interseção e uma direção de onde provavelmente vai estar outro ponto próximo da interseção, e a partir dessa direção, inicia-se uma caminhada sobre os pontos da curva interseção [Barnhill et al. (1987)], [Müllenheim (1990)], [Müllenheim (1991)]. Técnicas eficientes para o cálculo da interseção surgem quando são combinados esses métodos básicos, como apresentadas em [Barnhill–Kersey (1990)] e em [Kopakar (1991)].

O Método da Caminhada é uma técnica eficiente e rápida, desde que sejam garantidos que os vetores tangentes não se anulem nos pontos da curva interseção e que a estimativa inicial de cada ponto ao longo da caminhada seja boa. A técnica clássica para essa estimativa utiliza a direção do vetor tangente do ponto predecessor. Recentemente novas sugestões foram apresentadas para obter uma melhor aproximação inicial [Barnhill–Kersey (1990)], [Stoyanov (1992)] a preço de um maior custo computacional. Neste artigo é apresentada uma nova proposta com base no conceito de círculo osculador. Através dos exemplos testados observamos que a nossa proposta oferece bons resultados sem comprometer a simplicidade do algoritmo.

Este artigo é organizado em 6 seções. Na seção 2 apresentamos sucintamente o Método da Caminhada clássico; na seção 3 propomos uma nova maneira de calcular o passo da caminhada; na seção 4 uma forma detalhada de implementação desta nova proposta e na seção 5, fazemos algumas comparações entre a caminhada com passo calculado da forma clássica e a caminhada de acordo com a nossa proposta. Por fim, na seção 6 apresentamos algumas conclusões.

2 Método da Caminhada

Esta seção apresenta uma breve descrição do Método da Caminhada.

O Método da Caminhada requer um ponto da interseção e uma direção da caminhada para chegar ao próximo ponto sobre a interseção de superfícies. Esse processo é repetido seguidamente até percorrer toda a curva interseção. É interessante observar que, na aplicação desta técnica, em momento algum precisamos efetivamente resolver sistemas não-lineares [Barnhill et al. (1987)]. Ele requer, porém, que as funções envolvidas tenham pelo menos as primeiras derivadas contínuas.

Suponhamos que seja dado $P_0 = (p_1, p_2, p_3)$, um ponto próximo da interseção que corresponda aproximadamente a (u_0, v_0) no domínio de F e a (r_0, s_0) no domínio de G. Estes valores iniciais podem ser bem estimados se for usado antes, por exemplo, o Método da Subdivisão [Barnhill–Kersey (1990)] ou algum algoritmo específico como em [Müllenheim (1991)].

A partir de P_0 podemos definir uma seqüência de pontos que converge para um ponto da interseção [Barnhill-Kersey (1990)], [Müllenheim (1990)]. Determinamos o ponto A da superfície F que está mais próximo de P_0, e o ponto B da superfície G que está mais próximo de P_0. O ponto A é calculado como sendo $A = F(u, v)$ onde (u, v) é o limite da seqüência:

$$\begin{bmatrix} u_{i+1} \\ v_{i+1} \end{bmatrix} = \begin{bmatrix} u_i \\ v_i \end{bmatrix} + \left[J^t J\right]^{-1} J^t \begin{bmatrix} p_1 - f_1(u_i, v_i) \\ p_2 - f_2(u_i, v_i) \\ p_3 - f_3(u_i, v_i) \end{bmatrix},$$

onde $J = \begin{bmatrix} \frac{\partial f_1}{\partial u}(u_i, v_i) & \frac{\partial f_1}{\partial v}(u_i, v_i) \\ \frac{\partial f_2}{\partial u}(u_i, v_i) & \frac{\partial f_2}{\partial v}(u_i, v_i) \\ \frac{\partial f_3}{\partial u}(u_i, v_i) & \frac{\partial f_3}{\partial v}(u_i, v_i) \end{bmatrix}$.

De forma análoga, pode-se determinar o ponto B.

Conhecidos A e B, determinamos o vetor normal $\overrightarrow{n_1}$ à superfície F no ponto A e o vetor normal $\overrightarrow{n_2}$ a G em B (Fig. 1). Usando estes vetores normais e os pontos A e B, obtemos as equações dos planos tangentes $T_F(A)$ e $T_G(B)$ às superfícies F e G nos pontos A e B, respectivamente. Estes planos podem ser paralelos, coincidentes ou podem se interceptar segundo uma reta R. Caso os planos não se interceptem segundo uma reta, deve-se voltar ao início do algoritmo e escolher outro ponto inicial P_0.

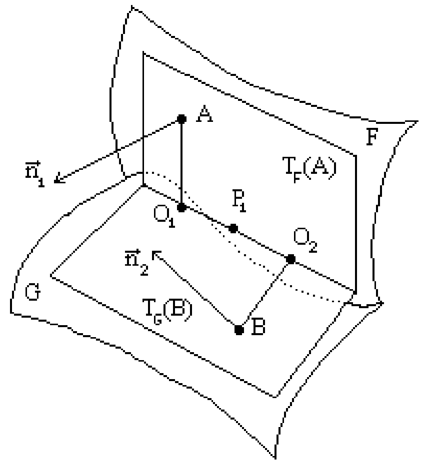

Figure 1: Vetores normais aos planos tangentes

Em seguida, calculamos a projeção ortogonal O_1 de A na reta R, a projeção ortogonal O_2 de B em R, e o ponto médio P_1 do segmento de reta O_1O_2. Substituimos P_1 por P_0 e repetimos todo esse procedimento para obter um novo ponto P_1, e assim sucessivamente até que P_1 seja suficientemente próximo de P_0 (dentro de uma aproximação desejada previamente definida, por exemplo, menor do que 10^{-6}), ou seja, que o ponto P_1 possa ser considerado pertencente à interseção de F e G.

Como, por hipótese, as funções F e G têm suas primeiras derivadas contínuas nos pontos da interseção, então o vetor tangente $\overrightarrow{T}$ à interseção de F e G no ponto que corresponda a (u_0, v_0) no domínio de F e a (r_0, s_0) no domínio de G, pode ser calculado como sendo $\overrightarrow{T} = \overrightarrow{N_1} \times \overrightarrow{N_2}$, onde

$$\begin{aligned} \overrightarrow{N_1} &= \frac{\partial F}{\partial u}(u_0, v_0) \times \frac{\partial F}{\partial v}(u_0, v_0) \\ \overrightarrow{N_2} &= \frac{\partial G}{\partial r}(r_0, s_0) \times \frac{\partial G}{\partial s}(r_0, s_0) \end{aligned}$$

Uma vez calculado o ponto da interseção e o vetor tangente $\overrightarrow{T}$, estima-se o ponto inicial da seqüência que converge para o próximo ponto da interseção a partir dos pontos conhecidos, procurando-se caminhar na direção do vetor tangente. O método clássico de caminhada estima o próximo ponto como sendo o ponto

$$P + L\frac{\overrightarrow{T}}{||\overrightarrow{T}||}$$

onde L é o comprimento do passo da caminhada. L pode ser constante (convenientemente pequeno) [Müllenheim (1990)], ou então ser dado em função da curvatura da curva interseção no ponto P [Barnhill–Kersey (1990)].

Alternativamente, podemos adotar outra direção de caminhada. Ao invés de considerarmos que a aproximação do próximo ponto esteja sobre a reta tangente ao ponto atual, podemos supor que o próximo ponto aproximado esteja sobre outro tipo de curva. Em [Stoyanov (1992)] é considerado que o próximo ponto aproximado esteja sobre uma parábola que aproxima a curva interseção no ponto corrente.

Depois que forem obtidos pontos isolados da interseção com seus respectivos vetores tangentes, pode-se traçar a curva usando-se, por exemplo, uma interpolação com polinômios de Hermite cúbicos.

Vale ressaltar ainda que, quando o vetor tangente se anula em algum ponto Q, ficamos sem ter uma direção para prosseguir na caminhada. Neste caso, entre outros poucos procedimentos que podem ser adotados, pode-se usar o vetor tangente que foi calculado antes de Q para obter a direção do próximo ponto [Müllenheim (1990)], [Barnhill–Kersey (1990)].

O Método da Caminhada assim descrito é genérico. Existem trabalhos que exploram as propriedades geométricas de certas classes específicas de superfícies para aumentar a eficiência do método [Patrikalakis (1993)], [Lasser (1986)] ou [Kriezis et al. (1990)].

3 Um Passo Circular

Quando duas curvas parametrizadas contínuas $\phi(t)$ e $\varphi(t)$ têm uma interseção em um ponto P tal que $P = \phi(t_1) = \varphi(t_2)$, podemos supor que $t_1 = t_2$; caso contrário, reparametrizamos uma das curvas fazendo a mudança de variável $s = t + t_1 - t_2$. Da definição de função contínua temos que se t for escolhido próximo de t_1, então o ponto $\phi(t)$ será próximo de $\varphi(t)$. Portanto, no caso particular em que $\phi(t)$ for uma circunferência tangente à curva $\varphi(t)$ em um ponto Q, a curva $\varphi(t)$ pode ser aproximada por uma circunferência $\phi(t)$. O círculo que passa pelos pontos $\varphi(t)$, $\varphi(t+h)$, $\varphi(t+k)$ quando h e k tendem a 0 é um círculo chamado *círculo osculador* em t cujo raio é denominado *raio de curvatura* em t. O círculo osculador tem um maior raio nos pontos de menor curvatura e um menor raio onde a curvatura for mais acentuada [Carmo (1971)].

Neste artigo, com base no conceito de círculo osculador, propomos um algoritmo de caminhada segundo passos circulares. São circulares porque tomamos como próximo ponto aproximado da interseção um ponto A sobre um círculo osculador aproximado, tangente à curva interseção. O círculo osculador é adaptativo no sentido de que ele se adapta ao formato da curva interseção: o seu raio varia ponto a ponto, de acordo com a curvatura da curva interseção em cada ponto (Fig. 2). A existência do círculo osculador é garantida pelo fato de que se pressupõe que a curva interseção seja contínua, com derivadas em todos os pontos.

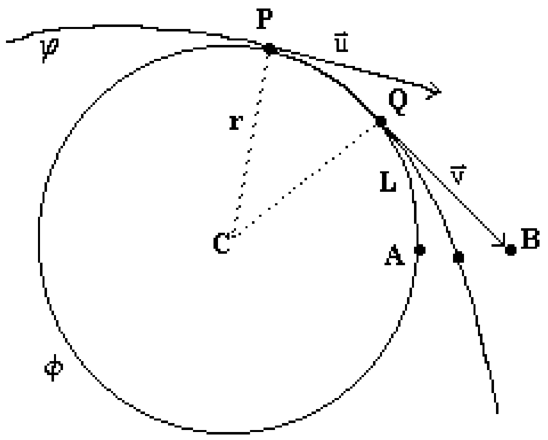

Figure 2: Círculo osculador aproximado

O uso de círculo osculador aproximado no lugar do círculo osculador teórico se deve ao compromisso entre o grau de complexidade da implementação e a precisão dos resultados que procuramos encontrar.

Construtivamente, o círculo osculador aproximado em cada ponto Q é o círculo que passa por ele e pelo seu ponto predecessor P no sentido da caminhada e tangencia os vetores $\vec{u}$ e $\vec{v}$ tangentes à curva interseção nestes pontos (Fig. 3). Para que esta construção seja possível as seguintes condições devem ser observadas;

(1) O ângulo entre os vetores $\vec{u}$ e $\vec{v}$ não deve ser nulo nem muito próximo de 0, para que o centro do círculo osculador possa ser calculado.

(2) O trecho a ser aproximado pela circunferência precisa ser "quase plano", uma vez que o círculo osculador deve estar contido num plano.

(3) A curva interseção φ não deve "oscilar" entre P e Q, para que o círculo construído seja uma melhor aproximação do círculo osculador.

Se a condição (1) não for satisfeita, recomenda-se o uso do passo clássico pontualmente. E se as condições (2) e (3) não forem atendidas deve-se, antes de se prosseguir na caminhada, determinar um outro ponto da interseção que fique entre P e Q de tal forma que (2) e (3) fiquem satisfeitas. A condição (2) pode ser testada, verificando-se se o produto misto dos vetores $\vec{u} \times (\vec{v} \cdot \overrightarrow{CP})$ é próximo de 0.

Tendo sido determinado o círculo osculador aproximado, o próximo ponto aproximado da curva interseção A é obtido caminhando-se L unidades sobre este círculo a partir de Q, no sentido de P para Q. Note-se que se escolhermos valores apropriados para L poderemos evitar que ocorram oscilações entre dois pontos consecutivos.

Segundo testes que realizamos para diversos tipos de superfícies, o ponto A obtido através do nosso círculo osculador aproximado, em geral, é mais próximo da curva interseção φ do que o ponto B construído usando-se a direção do vetor tangente (passo clássico).

A construção detalhada deste círculo é descrita na próxima seção.

4 Uma Implementação Eficiente

Nesta seção apresentamos um algoritmo para implementação da idéia descrita na seção anterior. São usadas apenas operações simples, sendo que a parte mais trabalhosa, são dois produtos de matrizes quadradas de ordem 3. O fato de usar a cada iteração dois pontos da curva e dois vetores tangentes não o torna mais complicado do que o método clássico, que usa somente um ponto e um vetor tangente.

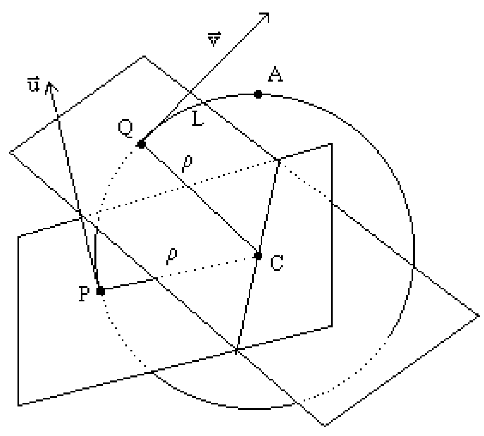

Figure 3: Computação do círculo osculador aproximado

Consideremos 2 pontos da curva interseção P e Q com seus respectivos vetores tangentes $\vec{u}$ e $\vec{v}$. Fixado o comprimento do passo da caminhada L e construída a circunferência descrita nesta seção, o próximo ponto aproximado da interseção pode ser estimado caminhando-se L unidades sobre a circunferência a partir de Q, no sentido de P para Q (Fig. 3).

Para cada par de pontos $P = (p_1, p_2, p_3)$ e $Q = (q_1, q_2, q_3)$ da curva interseção com seus respectivos vetores tangentes $\vec{u} = (u_1, u_2, u_3)$ e $\vec{v} = (v_1, v_2, v_3)$, podemos achar a equação do plano que passa por P e tem $\vec{u}$ como vetor normal, e também a equação do plano que passa por Q e tem $\vec{v}$ como vetor normal. Estas equações são dadas, respectivamente, por

$$\begin{aligned} u_1x + u_2y + u_3z &= u_1p_1 + u_2p_2 + u_3p_3 \\ v_1x + v_2y + v_3z &= v_1q_1 + v_2q_2 + v_3q_3 \end{aligned}$$

Se os vetores $\vec{u}$ e $\vec{v}$ não forem paralelos, podemos determinar a reta interseção $r(t)$ destes planos resolvendo o sistema linear formado pelas duas equações anteriores. Se esta reta $r(t)$ não for paralela ao plano $z = 0$, então ela intercepta o plano $z = 0$ em um único ponto R e intercepta também o plano $z = 1$ em um único ponto S. Substituindo $z = 0$ nas duas equações anteriores e resolvendo o sistema linear obtido obtemos um ponto $R = (r_1, r_2, r_3)$ na reta interseção dos planos. Fazendo o mesmo com $z = 1$, obtemos outro ponto $S = (s_1, s_2, s_3)$ da mesma interseção. Se a reta interseção destes planos for paralela ao plano $z = 0$, então calculamos os pontos R e S de forma análoga: substituímos $y = 0$ e depois $y = 1$ nas equações anteriores, ou então substituímos $x = 0$ e depois $x = 1$. Logo, a reta interseção dos planos tem equação paramétrica da forma $r(t) = R + t(S - R)$. Podemos então determinar o único ponto desta reta que é equidistante dos pontos P e Q, bastando para isso resolver uma equação do primeiro grau na variável t. A solução desta equação é $t = a/b$, onde

$$\begin{aligned} a &= ||P||^2 - ||Q||^2 - 2(P - Q).S \quad \text{e} \\ b &= 2(P - Q) \cdot (S - R). \end{aligned}$$

Não é possível calcular este valor de t somente quando $b = 0$. Neste caso, todos os pontos da reta $r(t)$ serão equidistantes de P e Q e podemos tomar o valor de t da iteração anterior.

O ponto equidistante de P e Q determinado anteriormente é o centro da circunferência e o raio ρ é a distância de P a C.

Quando for calculado o centro C, então calculamos o vetor normal ao plano da circunferência que é $\vec{n} = (P - C) \times (Q - C) = (n_1, n_2, n_3)$.

Aplicando a esta circunferência uma translação $T(X) = X - C$ seguida de uma rotação R que leve o vetor $\vec{n}$ a ficar paralelo ao vetor $\vec{k} = (0, 0, 1)$ obtemos uma circunferência no plano $z = 0$ com centro na origem $O = (0, 0, 0)$. Neste caso, os pontos P e Q são levados para pontos P' e Q' na circunferência do plano $z = 0$. Usamos P' e Q' para determinarmos um outro ponto A' sobre esta circunferência de tal forma que o arco de circunferência $Q'A'$ tenha comprimento L. É importante observar se o sentido P'

para Q' é horário ou anti-horário. Isto pode ser feito calculando-se os ângulos α e β que os vetores OP' e OQ' formam com o vetor $\overrightarrow{i} = (1,0,0)$, respectivamente.

Para que o arco de circunferência $Q'A'$ tenha comprimento L, é preciso que ele determine na circunferência de raio ρ, um ângulo central de $\theta = L/\rho$ radianos. Se $\alpha < \beta$, então o sentido de P' para Q' será anti-horário e o ponto A' procurado será igual a $(\rho\cos(\beta+\theta), \rho\sin(\beta+\theta))$; caso contrário, o sentido de P' para Q' será horário e o ponto A' será $(\rho\cos(\beta-\theta), \rho\sin(\beta-\theta))$.

Aplicando-se as transformações inversas R^{-1} e T^{-1} a A', obtemos o ponto A desejado na circunferência inicial. O cálculo de T^{-1} é imediato ($T^{-1}(X) = X + C$). Em coordenadas cartesianas, a rotação R é dada por $R(x,y,z) = [x,y,z][R]$, onde

$$\begin{aligned}
[R] &= \begin{bmatrix} \frac{\lambda}{V} & 0 & \frac{n_1}{V} \\ \frac{-n_1 n_2}{\lambda V} & \frac{n_3}{\lambda} & \frac{n_2}{\lambda} \\ \frac{-n_1 n_3}{\lambda V} & \frac{-n_2}{\lambda} & \frac{n_3}{V} \end{bmatrix} \\
\text{e } \lambda &= \sqrt{n_2^2 + n_3^2}, \\
V &= \sqrt{n_1^2 + n_2^2 + n_3^2}.
\end{aligned}$$

Esta matriz representa a composição de duas rotações: uma em torno do eixo x e outra em torno do eixo y.

No caso particular em que $\lambda = 0$, consideramos

$$[R] = \begin{bmatrix} 0 & 0 & \frac{n_1}{|n_1|} \\ 0 & 1 & 0 \\ \frac{-n_1}{|n_1|} & 0 & 0 \end{bmatrix}.$$

Como $[R]$ é produto de rotações, sua matriz será ortogonal, e daí, sua inversa será a matriz transposta $[R]^t$.

5 Exemplos e Comparações

Nesta seção são apresentados cinco exemplos.

Exemplo 1: Consideremos na hélice

$$f(t) = (\cos(t), \sin(t), t)$$

os pontos P = (0.5403, 0.8414, 1.0000) e Q = (-0.4161, 0.9092, 2.0000). Neste caso, pelo algoritmo descrito, obtemos uma circunferência de centro C = (-0.1568, -1.0200, 1.4192) e raio r = 2.031534. Usando um vetor tangente a curva no ponto Q de comprimento igual a 1 obtemos o ponto (-1.0591, 0.5789, 2.8104) que está a uma distância de 0.241830 unidades da hélice. Usando agora a circunferência de centro C e raio r e um arco de comprimento também 1, obtemos o ponto (-1.0459, 0.3994, 2.5686) que está a uma distância de 0.192835 unidades da hélice.

Consideremos agora a curva polinomial-racional

$$f(t) = (\frac{t^3 - t + 1}{t^2+1}, t^2 + 4t + 3, \frac{2t^3}{t^2+1} - 7t - 2)$$

e tomemos sobre esta curva os pontos P = (0.5000, 8.0000, -8.0000) e Q = (1.4000, 15.0000, -12.8000). Com isso, temos uma circunferência de centro C = (-15.4835, 42.7887, 32.1481) e raio r = 55.4761. Usando o vetor tangente em Q de comprimento 0.1, obtemos o ponto (2.5524, 23.5287, -17.6399) que está a uma distância de 0.258819 unidades da curva. Usando o arco da circunferência também de comprimento 0.1, obtemos o ponto (1.4082, 15.0861, -12.8500) que está a 0.003498 unidades de distância da curva.

Exemplo 2: Consideremos as superfícies parametrizadas por

$$\begin{aligned}
F(u,v) &= (u, v, u^3 + v^2 - u - 2v + 1) \\
G(u,v) &= (uv, u+v, 5v^3 - u^2 v + 5).
\end{aligned}$$

Podemos obter pelos algoritmos clássicos os pontos P = (1.4517, 2.5642, 4.0550) e Q = (1.4579, 2.5664, 4.0950) pertencentes à interseção destas superfícies. Estimando-se o próximo ponto da interseção pelo vetor tangente em Q, obtemos como ponto aproximado A = (1.6094, 2.5834, 4.1119). Com 3 iterações do algoritmo clássico, obtemos o ponto I = (1.4639, 2.5686, 4.1342) na interseção. A distância entre A e I é de 0.1479 unidades.

Usando-se a circunferência do algoritmo proposto neste artigo, obtemos como ponto aproximado da interseção $A = (1.6074, 2.6158, 5.0825)$. Com 3 iterações do algoritmo de convergência clássico, obtemos o ponto $I = (1.5926, 2.6237, 5.0839)$ na interseção das superfícies. Neste caso, a distância entre I e A é de 0.0168 unidades. Além disso, no primeiro método a distância entre I e Q é de 0.0397 unidades, enquanto que no segundo método é de 0.9996 unidades.

Exemplo 3: Sejam F e G o parabolóide de revolução e o cilindro circular parametrizados por

$$\begin{aligned}
F(u,v) &= (u, v, u^2 + v^2) \text{ e por} \\
G(u,v) &= (2\cos(u), 2\sin(v), v).
\end{aligned}$$

Os pontos P = (1.3993, 1.4289, 4.0000) e Q = (0.7206, 1.8656, 4.0000) pertencem à interseção

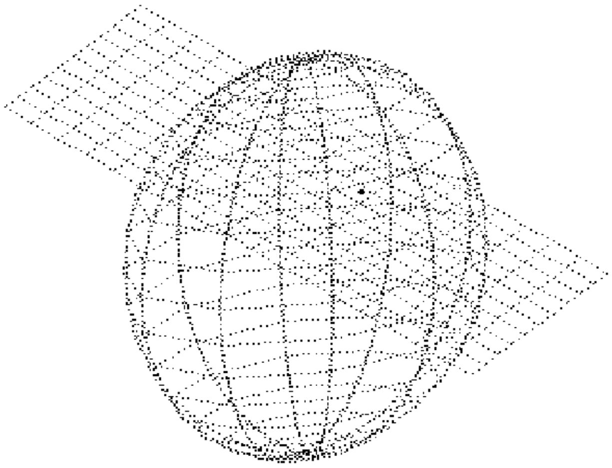

Figure 4: Exemplo 4 - Resultado obtido pelo método tradicional

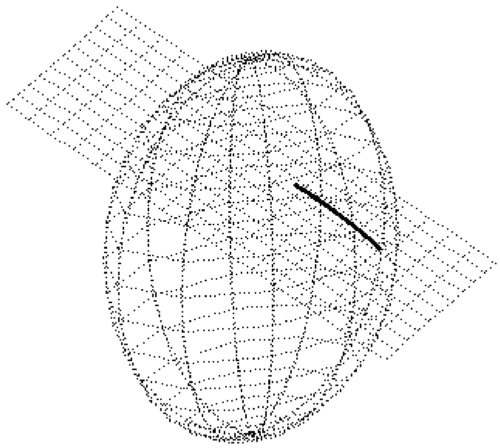

Figure 5: Exemplo 4 - Resultado obtido pelo método proposto

destas superfícies. O vetor tangente em Q fornece como próximo ponto aproximado A = (-0.2121, 1.7595, 3.8939). Com 2 iterações do algoritmo clássico, obtemos I= (-0.2268, 1.9870, 4.0000) na interseção. A distância entre I e A é de 0.2514 unidades. A circunferência que passa por P e Q definida neste artigo, fornece o ponto A = (-0.2476, 2.0258, 4.1351) como aproximação para o próximo ponto da interseção. Com apenas 2 iterações do algoritmo clássico, obtemos convergência para o ponto da interseção I = (-0.2427, 1.9852, 4.0000). Neste caso, a distância entre I e A fica reduzida a 0.1411 unidades. Neste exemplo a distância entre I e Q é de 0.9552 unidades no primeiro método e é de 0.9707 unidades no segundo.

Exemplo 4: Consideremos o plano $F(u,v) = (u, v, u-v+1)$ e o elipsóide $G(u,v) = (\cos(v)\cos(u), 2\sin(v)\cos(u), 3\sin(u))$. Usando um vetor tangente de comprimento constante igual a 0.001, o algoritmo tradicional da caminhada sobre a interseção de F e G encontra um total de 100 pontos da interseção, executando um total de 200 iterações no trecho do algoritmo que trata da convergência dos pontos aproximados para os pontos da interseção. A distância total percorrida nestes 100 pontos é de 0.0016 unidades (Fig. 4). Usando o passo circular de comprimento constante também igual a 0.001, obtemos 100 pontos, num total de 300 iterações na convergência para pontos da interseção. Neste caso, a distância total percorrida é de 0.1484 unidades (Fig. 5).

Exemplo 5: Consideremos agora os cilindros circulares

$$\begin{aligned} F(u,v) &= (u, \frac{1-v^2}{1+v^2}, \frac{2v}{1+v^2}) \text{ e} \\ G(u,v) &= (\frac{1-v^2}{1+v^2}, \frac{2v}{1+v^2}, u). \end{aligned}$$

Usando um vetor tangente de comprimento constante igual a 0.01, o algoritmo tradicional encontra um total de 100 pontos da interseção, executando um total de 200 iterações na convergência para os pontos da interseção. A distância total percorrida nestes 100 pontos é de 0.3646 unidades (Fig. 6). Usando o passo circular de comprimento constante também igual a 0.01, obtemos 100 pontos num total de 296 iterações na convergência para pontos da interseção. Neste caso, a distância total percorrida é de 1.0099 unidades (Fig. 7). Se com estes mesmos cilindros, reduzirmos o tamanho do passo da caminhada para 0.001 unidades, o passo circular tem um melhor desempenho do que o passo obtido usando-se apenas o vetor tangente. Neste caso, a caminhada segundo o método clássico encontra 100 pontos num total de 197 iterações, percorrendo uma distância total de 0.0475 unidades sobre a curva interseção, enquanto que o passo circular encontra 100 pontos da interseção executando 188 iterações e percorrendo um total de 0.1010 unidades sobre a curva interseção.

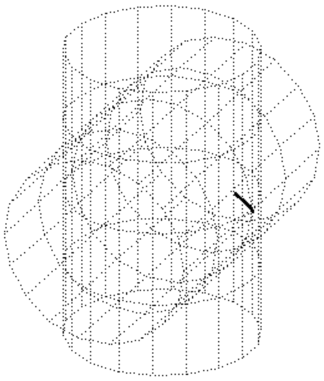

Figure 6: Exemplo 5 - Resultado obtido pelo método tradicional

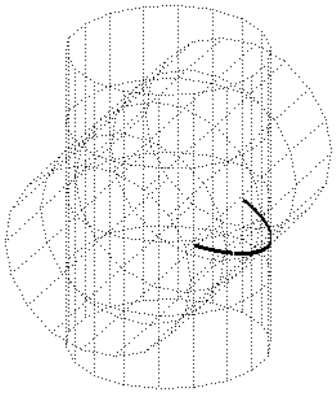

Figure 7: Exemplo 5 - Resultado obtido pelo método proposto

6 Considerações finais

Em muitos exemplos testados, o esforço computacional do método proposto não é muito maior do que o usado no método da caminhada simples clássica (vetor tangente de comprimento constante). Se for usado uma caminhada orientada pelo vetor tangente com comprimento variando ponto a ponto, de acordo com a curvatura da curva interseção φ (como em [Barnhill–Kersey (1990)] ou [Stoyanov (1992)]) certamente o passo circular aqui proposto será menos custoso sob o ponto de vista computacional. O cálculo da curvatura ponto a ponto equivale ao cálculo de uma expressão da forma

$$\kappa = \frac{||\varphi'(t) \times \varphi''(t)||}{||\varphi'(t)||^3}$$

Em quase todos os exemplos testados, o passo circular forneceu uma melhor aproximação inicial para os pontos da interseção, bem como uma maior distância percorrida na curva interseção. Pode ser mostrado que a maior distância percorrida é consequência da melhor aproximação inicial.

7 Agradecimentos

O primeiro autor, Lenimar N. Andrade, é o professor da UFPb e durante a elaboração deste trabalho esteve afastado da mesma, realizando doutorado na UNICAMP, com apoio financeiro do PICD/CAPES.

8 Referências

R. E. Barnhill, G. Farin, M. Jordan, B. R. Piper, Surface/surface intersection, *Computer Aided Geometric Design* **4** (1987), 3–16.

R. E. Barnhill, S. N. Kersey, A marching method for parametric surface/surface intersection, *Computer Aided Geometric Design* bf 7 (1990), 257–280.

M. P. Carmo, *Elementos de Geometria Diferencial,* Ao Livro Técnico S.A. e Editora Universidade de Brasília, 1971.

E. G. Houghton, R. F. Emnett, Implementation of a divide-and-conquer method for intersection of parametric surfaces, *Computer Aided Geometric Design* **2** (1985), 173–183.

P. Koparkar, Surface intersection by switching from recursive subdivision to iterative refinement, *The Visual Computer* **8** (1991), 47–63.

G. A. Kriezis, P. V. Prakash, N. M. Patrikalakis, Method for intersecting algebraic surfaces with rational polynomial patches, *Computer-Aided Design* **22** (1990), 645–654.

D. Lasser, Intersection of parametric surfaces in the Bernstein-Bézier representation, *Computer-Aided Design* **18** (1986), 186–192.

G. Müllenheim, Convergence of a surface/surface intersection algorithm, *Computer Aided Geometric Design* **7** (1990), 415–423.

G. Müllenheim, On determining start points for a surface/surface intersection algorithm, *Computer Aided Geometric Design* **8** (1991), 401–408.

N. M. Patrikalakis, Surface-to-surface intersections, *IEEE Computer Graphics & Applications* **13** (1993), 89–95.

Tz. E. Stoyanov, Marching along surface/surface intersection curves with an adaptive step length, *Computer Aided Geometric Design* **9** (1992), 485–489.