UN ALGORITMO DE MARCHA PARA LA INTERSECCION DE DOS SUPERFICIES*

OMAIRA RODRIGUEZ
ESCUELA DE COMPUTACION
FACULTAD DE CIENCIAS
UNIVERSIDAD CENTRAL DE VENEZUELA

APARTADO 47002
LOS CHAGUARAMOS 1041 -A
CARACAS - VENEZUELA

TELEFONO: + 58-2-662-7233
FAX: + 58-2-662-7121
E-mail: orodrig@dino.conicit.ve

*Trabajo parcialmente financiado por CONICIT

ABSTRACT. Se propone un algoritmo heurístico para encontrar la curva intersección de dos superficies dadas en forma implicita. Los puntos sobre la intersección se obtienen paso a paso haciendo marchar un cubo sobre la curva.

KEY WORDS. Geometría, marcha, superficies, curva, intersección.

1. INTRODUCCION

Cualquier sistema diseñado para manejar superficies, necesita producir representaciones de curvas que yacen sobre esas superficies, tales como el contorno de datos experimentales, intersección de superficies y cálculos de curvas siluetas para la producción de dibujos. Por su interés computacional, matemático, así como su amplia aplicación este tópico ha recibido mucha atención en los últimos años.

Se propone un algoritmo heurístico para obtener la curva intersección de dos superficies conocidas, dadas en forma implícita, en una región del espacio.

Nuestro algoritmo usa una técnica simple de marcha para 3D que rastrea la curva intersección de dos superficies definidas en forma implícita

$$f(x,y,z) = 0, \quad g(x,y,z) = 0$$

Se imponen ciertas condiciones de regularidad sobre la curva intersección: que la intersección sea transversal, que la curva sea simple y conexa.

La selección de los puntos se realiza paso a paso, en cada uno de ellos se determina una dirección de movimiento y se selecciona aquel punto que se encuentre mas cercano a, o sobre, la curva intersección, de esta manera se obtiene una secuencia de puntos que describen la curva, o segmento de ella.

2. DESCRIPCION DEL ALGORITMO

Sea P_0 un punto inicial, el cual puede ser conocido o derivado por otro método, perteneciente a la curva intersección de las dos superficies f y g dadas en forma implícita

$$f(x,y,z) = 0$$
$$g(x,y,z) = 0 .$$

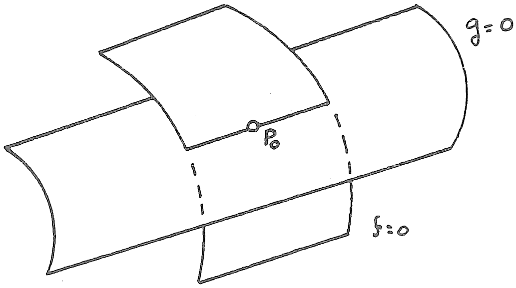

Fig. 1

Consideremos un cubo cuyo centro es el punto inicial $P_0(x_0,y_0,z_0)$ Fig. 2, es decir, la curva intersección pasa por el centro del cubo. Nos interesa entonces determinar el próximo punto de la curva, o lo que es lo mismo, encontrar el punto mas cercano a la curva intersección y además determinar la dirección de movimiento.

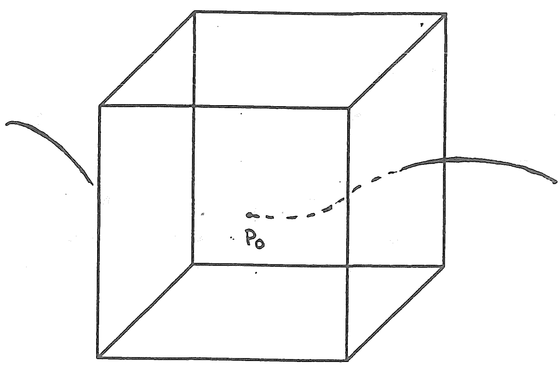

Fig. 2

Cada uno de los vértices, centros de cara y puntos medios de aristas son posibles candidatos a ser próximos puntos de nuestra curva Fig. 3.

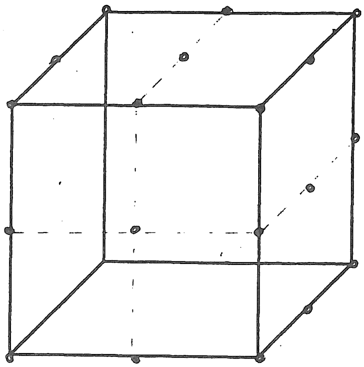

Fig. 3

Bajo nuestra suposición tenemos en total 26 candidatos a ser próximos puntos. Las ternas (i, j, k) i = -1, 0 , 1; j = -1 , 0 , 1 ; k = -1 , 0, 1 nos permiten identificar cada uno de los vértices, centros de cara y puntos medios de las aristas. Cada uno de los 26 candidatos pertenecen a una y solo una de estas ternas de puntos. A su vez sirven para identificar la dirección de movimiento a lo largo de la curva, dada por (dirX, dirY, dirZ).

Así pues (dirX, dirY, dirZ) = (1, -1, 1) nos indica que el próximo punto mas cercano a la curva será

$$(X_{old} + \text{delta } X,\ Y_{old} - \text{deltaY},\ Z_{old} + \text{deltaZ})$$

donde deltaX, deltaY, deltaZ son los respectivos incrementos a las correspondientes variables en R^3 y X_{old} , Y_{old}, Z_{old} es la posición actual del centro del cubo.

Para la selección del primer punto se evaluan las funciones f(x,y,z) y g(x,y,z) en cada uno de nuestros candidatos a próximo punto, para el primer paso 26 puntos. Nuestro próximo punto P_{next} es aquel que minimice

$$|f(x,y,z)| + |g(x,y,z)| \qquad (1)$$

Efectivamente, si (x^* , y^*, z^*) es un cero de la curva intersección, entonces $f(x^*,y^*,z^*) = 0$ y $g(x^*,y^*,z^*) = 0$ por lo tanto es razonable suponer que si el punto (x,y,z) minimiza $|f|+|g|$ éste será usualmente el más cercano a la curva buscada.La selección de este punto nos determina la dirección inicial de curva En caso de que más de un punto satisfaga la condición, arbitrariamente seleccionamos uno de ellos. Esta selección puede ser mejorada como se explicará mas adelante.

Teniendo estos dos puntos, el punto inicial dado (centro del cubo P_{old}) y el primer punto escogido P_{new} , podemos obtener un próximo punto cercano a la curva moviéndonos (marchando) al próximo cubo, como se explica a continuación.

Consideremos ahora que el punto P_{new} es el centro de nuestro cubo, luego dependiendo de su procedencia podemos establecer los siguientes casos, en cada uno de ellos hemos limitado el número de puntos a considerar como candidatos a próximos puntos.

i) P_{new} proviene de un centro de cara, del cubo cuyo centro es P_{old}.

ii) P_{new} proviene de un vértice, del cubo cuyo centro es P_{old}.

iii) P_{new} proviene de un punto medio de arista, del cubo cuyo centro es P_{old}.

Para el caso i) los candidatos a ser próximos puntos más cercanos a nuestra curva, se reducen a aquellos que se encuentran en la cara de enfrente (del próximo cubo) a la dirección de movimiento Fig. 4.

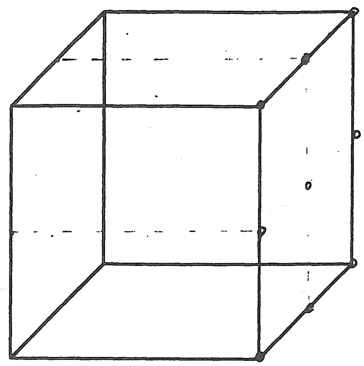

Fig. 4

Para seleccionar el nuevo punto evaluamos las funciones f y g en cada uno de los nueve puntos candidatos, el próximo punto será aquel que satisfaga la condición de minimización (1).La dirección de movimiento (dirX, dirY, dirZ) viene dada por la posición , en el cubo, del punto seleccionado.

Si el centro del cubo proviene de un vértice (caso ii) solo nos interesan aquellos puntos que estan situados en la esquina diagonal opuesta al vértice de procedencia.

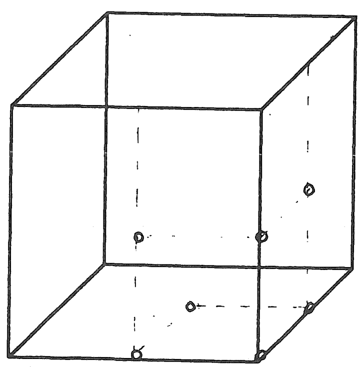

Fig. 5

P_{next} se obtiene al evaluar las funciones f y g en cada uno de estos 7 puntos, el punto seleccionado es el que satisface la condición de minimización (1). De nuevo, la dirección de movimiento (dirX,dirY,dirZ) esta dada por la posición, en el cubo, que ocupa el punto seleccionado.

El último caso es cuando el centro del cubo proviene del punto medio de una arista (caso iii), aquí los candidatos a próximos puntos son aquellos situados en la esquina contigua, opuesta en diagonal, a la arista de procedencia. Al igual que en los otros casos la dirección de movimiento (dirX,dirY,dirZ) esta dada por la posición que en el cubo ocupa el punto seleccionado.

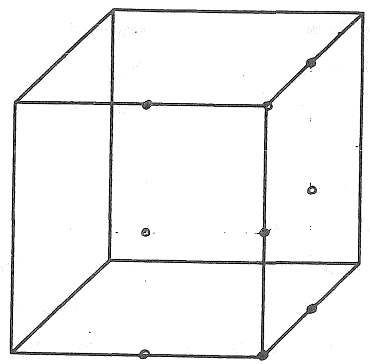

Fig. 6

En general, si tenemos dos puntos consecutivos de la curva: P_{old} y P_{new}, se escoge P_{next}, evaluando f y g en los puntos determinados por la dirección de movimiento dada por estos y de acuerdo a cada uno de los casos ya estudiados. El proceso continua actualizando las posiciones

$$P_{old} = P_{new}$$
$$P_{new} = P_{next}$$

3. ALGUNAS CONSIDERACIONES

La primera vez que se genera un próximo punto se necesitan 26*2 evaluaciones de la función, en los siguientes puntos el máximo número de evaluaciones llega a ser 9*2 en el peor de los casos lo cual implica que se disminuye considerablemente el número de evaluaciones de la función por cada punto generado.

La dirección inicial va a permitir orientar el trazado de la curva, luego la selección del punto inicial y la dirección inicial juegan un papel importante en el éxito del algoritmo, se podría utilizar un procedimiento más sofisticado para esta selección, sin embargo se debe tomar en cuenta los costos computacionales que esto acarrearía.

Es razonable esperar que si (x^*, y^*, z^*) pertenece a la curva intersección desearíamos que al menos dos de nuestros puntos seleccionados como candidatos esten sobre la curva, sin embargo esto no siempre es cierto, ya que la curva puede pasar entre estos dos puntos. También puede ocurrir que existan más de dos puntos donde f y g se anulan, esto nos indica un comportamiento irregular de la curva, estos casos los excluimos de esta discusión.

Una de las posibles mejoras de este algoritmo consiste en la modificación del criterio de selección cuando más de un punto satisface la condición de minimización por una que tome en cuenta a las funciones envueltas bien sea su derivada o cualquier otra propiedad inherente a la función.

Hemos pensado que es posible automatizar el paso de recorrido, para ello se podría utilizar la noción de continuidad geométrica que implicaría la evaluación de la función en tan solo tres puntos.

El algoritmo propuesto genera una secuencia de puntos en coordenadas de mundo que luego son desplegadas usando un mecanismo de visualización en R^3. El usuario puede decidir el recorrido que hará sobre la curva. Así mismo puede modificar el paso del recorrido.

REFERENCIAS

1 Asteasu, C . 'Intersection of arbitrary surfaces'. Computer Aided Design. Vol 20 No 9 (November 1988) pp 533-538.

2 Chandler, R.E. 'A tracking algorithm for implicitly defined curves'. IEEE Computer Graphics & Applications Vol 9 (March 1988) pp 83-89.