

Ministerio de Ciencia, Tecnología y Medio Ambiente
Instituto de Cibernética, Matemática y Física
Grupo de Métodos Numéricos

Ajuste de Datos por medio de un Spline Cónico

Tesis presentada en opción al grado de Maestro
en Ciencias Matemáticas

Autor: Victoria Hernández Mederos

Tutor: Dr. Pablo Barrera Sánchez, UNAM, México
Asesor: Dr. Jorge Estrada Sarlabous, ICIMAF, Cuba

Ciudad de la Habana, Cuba
1996

A todos los que algún día creyeron en mí.....

Agradecimientos

Deseo agradecer a todos los que de una u otra manera han contribuido en la realización de este trabajo. En primer lugar, a mi tutor Pablo Barrera que me propuso el tema y me ha ofrecido la oportunidad de superarme durante varios años; a mi esposo que me ha asesorado técnicamente y además me ha brindado aliento, amor y comprensión en los momentos difíciles de la realización del trabajo. A mi familia, que siempre me ha apoyado para seguir adelante, en especial a mi madre. A Jorgito, que se ha privado muchas veces de mi atención, aún sin comprender bien el por qué.

A mis profesores de la Licenciatura en Matemáticas, que me enseñaron a razonar y me iniciaron en esta profesión. A mis compañeros del Departamento de Estadística y Métodos Numéricos, donde he encontrado apoyo. A Valia Guerra y a Lilliam Alvarez por el aliento brindados, a Ramón Pico y a Ignacio Sánchez por la asesoría en la elaboración del software, a Aurora Pérez por la ayuda en la edición. A todos ellos, gracias!

Resumen

En este trabajo se presenta un método para ajustar un conjunto de datos por medio de un spline cónico. La curva de ajuste es de clase C^1 y el método propuesto es interactivo, en el sentido de que es posible escoger cuál es la máxima distancia que se desea entre los datos y la curva de aproximación. Además se deduce una fórmula de inversión para cónicas, que asigna a un punto que no pertenece a la curva, el valor del parámetro que corresponde a otro punto cercano que sí se encuentra sobre la misma. También se presenta una reparametrización de las cónicas, que aproxima la parametrización en términos de la longitud de arco y que permite generar puntos sobre la curva "uniformemente" distribuidos.

Contenido

0.1	Introducción	1
1		5
1.1	Parametrización de una cónica	5
1.2	Forma de Bezier de una sección cónica	8
1.3	Propiedades de las cónicas.	10
2		13
2.1	Planteamiento general del problema	13
2.2	Distancia de un punto a una sección cónica y del polígono al spline	17
2.3	Problema de inversión	21
2.4	Necesidad de reparametrizar las cónicas	27
3		35
3.1	Parametrización global del spline cónico	35
3.2	Algoritmo para construir el Spline Cónico de Ajuste. .	40
3.3	Programa y ejemplos de prueba	44
3.4	Conclusiones y Recomendaciones	47

0.1 Introducción

Las cónicas constituyen una de las familias de curvas más conocidas y estudiadas por la Matemática. Esto se debe a que poseen propiedades que las hacen muy útiles en la solución de variados problemas teóricos y de la industria [17] [20]. Hoy en día las cónicas forman parte de casi todos los estándares gráficos internacionales [10] [24], así como de los sistemas de CAD/CAM de reconocido prestigio mundial. De hecho, el primer sistema CAD que se conoce [14] se basó en el uso de cónicas para diseñar los fuselajes de un avión. En relación con las cónicas y otras curvas de Bezier, existe además un gran número de trabajos publicados por diferentes autores, entre los que se destacan Farin [4, 5, 6], Lee [12], Piegl [18, 19, 20], Pavlidis [15] y Pratt [21]. En estos trabajos se muestra la posibilidad de utilizar con éxito las cónicas en el ajuste de datos, en la aproximación de dibujos digitalizados (ya sea para su compactación o para el reconocimiento de patrones) y en el diseño y escalamiento de fonts [11] .

En este trabajo, se presenta un método para ajustar un conjunto de puntos del plano, por medio de una curva formada por secciones rectas y cónicas, las cuales se unen "suavemente". Esta curva es un spline, en el sentido de que está definida por secciones que se unen de modo que la curva es de clase C^1 . Cada una de las cónicas que constituyen el spline está definida por sus puntos extremos, las tangentes en dichos puntos y un parámetro adicional que permite controlar la "cercanía" de la curva al punto de intersección de las tangentes. Esta forma de describir las cónicas se conoce como representación de Bernstein-Bezier y se ha vuelto muy usual en los últimos tiempos, debido a que muchas propiedades geométricas de las cónicas, incluyendo su clasificación, se obtienen fácilmente a partir de la misma.

En la actualidad algunos autores [15] [21] opinan, que en la solución de diversos problemas de aproximación, los splines cónicos dan resultados al menos tan buenos como los cúbicos. Como se sabe, la popularidad de los splines cúbicos se debe en parte a razones históricas. Por otro lado, frecuentemente se desea construir una curva spline, donde cada sección está forzada a pasar por los dos puntos extremos,

con tangentes prefijadas. En tal caso se necesitan 4 grados de libertad y utilizando splines polinomiales, esto exige emplear secciones de al menos grado 3. Sin embargo, una cónica posee 5 grados de libertad, de modo que es posible satisfacer los requerimientos anteriores y todavía nos queda un coeficiente libre que se puede utilizar en lograr un mejor "ajuste".

Precisamente, con relación al término ajuste debemos aclarar que en nuestro trabajo, no se utiliza el mismo en el sentido usual de ajuste mínimo cuadrado. Esto se debe esencialmente a las dos razones siguientes. Supongamos que $f(x, y) = 0$ representa la ecuación implícita de una cónica y que (\tilde{x}, \tilde{y}) es un punto que no pertenece a la misma. Entonces el valor $f(\tilde{x}, \tilde{y})$ no siempre es proporcional a la "distancia" del punto a la cónica, con la distancia definida como la longitud r del vector normal a la cónica que pasa por (\tilde{x}, \tilde{y}) [15]. Más aún, se pueden construir ejemplos donde $f(\tilde{x}, \tilde{y})$ tome distintos valores, dependiendo de la posición del punto (\tilde{x}, \tilde{y}) y sin embargo la longitud r del vector normal sea la misma. Por otro lado, nuestro interés principal, es desarrollar un algoritmo para aproximar el polígono que interpola un conjunto de datos, por medio de una figura "suave" que conserve lo más posible la forma del polígono original. En ese sentido, es conocido que resulta muy difícil definir un criterio matemático de aproximación, es decir una función de costo, cuya minimización conduzca a la "mejor aproximación" de acuerdo con la percepción humana. Esto se debe a que no es fácil cuantificar criterios "estéticos" respecto a la forma que debe tener la curva que se desea [15].

Debido a todo lo anterior, el método que presentamos en este trabajo, no proporciona una solución óptima al problema de ajuste de datos, en el sentido mínimo cuadrado. Sin embargo, nuestro algoritmo posee al menos dos propiedades muy deseadas. En primer lugar, es interactivo, pues nos permite construir una curva de aproximación tan cercana a los datos como se desee, en términos de la distancia definida en la sección 2.2. Además el algoritmo diseñado es invariante por transformaciones afines, lo cual significa que el resultado de ajustar las imágenes por una transformación afín de los datos originales, es

el mismo que el de aplicar la transformación afín al spline que ajusta los datos iniciales.

La Tesis se divide en tres capítulos. El primero contiene una serie de resultados conocidos sobre las cónicas. En él se demuestra que toda cónica irreducible, se puede representar paramétricamente mediante dos funciones racionales cuadráticas. Esto permite reescribir las cónicas en términos de los polinomios de Bernstein, dando lugar a la llamada representación de Bezier. Al final del capítulo se ofrecen, sin demostración, una serie de propiedades de las cónicas, las cuales se deducen de la representación de Bezier. Estas propiedades se utilizan más adelante a lo largo de todo el trabajo.

El capítulo dos es el núcleo de la Tesis. En éste se define una forma de medir la distancia de un punto a un segmento de cónica. Esta definición es muy útil en dos sentidos. En primer lugar, permite cuantificar la calidad de la aproximación que se va a construir. Además se utiliza para deducir una fórmula de inversión, con un especial significado geométrico. En el capítulo se demuestra además, la necesidad de reparametrizar las cónicas que constituyen el spline, cuando uno desea obtener puntos de la curva uniformemente distribuidos en términos de la longitud de arco. Para resolver este problema, se propone una reparametrización local de las cónicas, que es elegante y poco costosa.

Finalmente, en el capítulo tres se ofrece una parametrización global del spline cónico, que tiene en cuenta la geometría de cada sección y garantiza que la curva obtenida sea de clase C^1 . En el mismo se incluye una descripción del algoritmo propuesto, así como algunos ejemplos ilustrativos de los resultados del ajuste.

Los aspectos más novedosos de esta Tesis se pueden resumir de la siguiente forma. En la Tesis se deduce una fórmula de inversión, que asigna a un punto (\tilde{x}, \tilde{y}) que no pertenece a la curva, el valor t del parámetro que corresponde a un punto (x, y) sobre la cónica $c(t)$, donde (x, y) es el punto que define la distancia entre (\tilde{x}, \tilde{y}) y $c(t)$. La solución de este problema es muy importante, ya que debido a la imposibilidad de utilizar una aritmética infinita, con frecuencia un

punto (\tilde{x}, \tilde{y}) que debía pertenecer a la cónica, no satisface la ecuación implícita de la misma. Por eso, es necesario garantizar que la fórmula de inversión que tengamos, asigne a un punto que no esté sobre la curva, el valor del parámetro que corresponde a otro punto "cercano" que sí pertenece a la misma.

El segundo resultado novedoso está vinculado con la reparametrización de una cónica. Es conocido que la mayor ventaja de la representación paramétrica es lo fácil que resulta generar un conjunto de puntos sobre la curva. Sin embargo, en nuestro trabajo nos enfrentamos muy pronto al problema de que los puntos obtenidos para valores equidistantes del parámetro, se distribuyen irregularmente, concentrándose en algunos segmentos de la curva. Por eso, es necesario reparametrizar cada sección cónica en términos de un parámetro que aproxime, en cierto sentido, la longitud de arco. Esto nos permitió obtener puntos regularmente distribuidos, para valores equidistantes del nuevo parámetro.

Capítulo 1

1.1 Parametrización de una cónica

Clásicamente las cónicas se conocen por su ecuación implícita en coordenadas cartesianas:

$$F(x, y) = 0 \quad (1.1)$$

donde F es un polinomio cuadrático en x y y que escribiremos como:

$$F(x, y) = a_{20}x^2 + a_{11}xy + a_{02}y^2 + a_{10}x + a_{01}y + a_{00} \quad (1.2)$$

La ecuación implícita es muy útil cuando se trata de determinar si un punto dado pertenece o no a la curva y su posición con respecto a la misma, en caso de que no pertenezca. Por otro lado, si la curva se puede parametrizar, entonces a través de la ecuación paramétrica se pueden obtener fácilmente un conjunto de puntos sobre un segmento determinado de la misma. Por eso, en esta sección vamos a probar que toda cónica irreducible, se puede parametrizar en términos de dos funciones racionales cuadráticas.

Teorema 1 *Sea $F(x, y) = 0$ la ecuación implícita de una cónica irreducible. Entonces existen dos funciones racionales cuadráticas $q_x(t)$ y $q_y(t)$, tales que el conjunto de puntos $(x(t), y(t))$ con*

$$\begin{aligned} x(t) &= q_x(t) \\ y(t) &= q_y(t) \end{aligned}$$

describe la misma cónica.

Demostración

Sea (x_0, y_0) un punto sobre la cónica. Consideremos la familia de rectas que pasan por (x_0, y_0) con pendiente t :

$$y = y_0 + t(x - x_0) \quad (1.3)$$

Del teorema de Bezout [8] es conocido que una recta y una cónica irreducible se intersectan a lo sumo en dos puntos. Por lo tanto, todo punto sobre la cónica puede ser obtenido como el punto de intersección con la cónica de alguna recta que pasa por (x_0, y_0) .

Esto significa que la cónica se puede parametrizar, en términos de la pendiente t de las rectas que pasan por (x_0, y_0) . Calculemos entonces el otro punto de intersección de la cónica con la recta por (x_0, y_0) con pendiente t .

Sustituyendo (1.3) en (1.1) resulta que:

$$\begin{aligned} F(x, y) &= F(x, y_0 + t(x - x_0)) \\ &= a_{20}x^2 + a_{11}x(y_0 + t(x - x_0)) + a_{02}(y_0 + t(x - x_0))^2 \\ &\quad + a_{10}x + a_{01}(y_0 + t(x - x_0)) + a_{00} = 0 \end{aligned}$$

Desarrollando esta expresión y agrupando en potencias de x se obtiene que:

$$F(x, y) = c_2(t)x^2 + c_1(t)x + c_0(t) = 0 \quad (1.4)$$

donde

$$\begin{aligned} c_0(t) &= a_{02}x_0^2t^2 + (-2a_{02}y_0x_0 - a_{01}x_0)t + a_{02}y_0^2 + a_{01}y_0 + a_{00} \\ c_1(t) &= -2a_{02}x_0t^2 + (-a_{11}x_0 + 2a_{02}y_0 + a_{01})t + a_{11}y_0 + a_{10} \\ c_2(t) &= a_{02}t^2 + a_{11}t + a_{20} \end{aligned}$$

Como $F(x, y)$ es una cónica, $c_2(t)$ no es el polinomio nulo . Por tanto, de (1.4) resulta que :

$$x^2 + \frac{c_1(t)}{c_2(t)}x + \frac{c_0(t)}{c_2(t)} = 0 \quad (1.5)$$

Por otro lado, como (x_0, y_0) es uno de los puntos de intersección de la recta con la cónica, x_0 es una de las raíces de la ecuación (1.5). Para cada recta con pendiente t , sea $x_1(t)$ la abscisa del otro punto de intersección. Entonces:

$$x_1(t) = -x_0 - \frac{c_1(t)}{c_2(t)} \quad (1.6)$$

Para obtener la ordenada $y_1(t)$ del punto de intersección basta con sustituir $x_1(t)$ en (1.3):

$$y_1(t) = y_0 + t(-2x_0 - c_1(t)/c_2(t)) \quad (1.7)$$

Finalmente, si sustituimos en (1.6) y (1.7) las expresiones para $c_1(t)$ y $c_2(t)$ y agrupamos en potencias de t , llegamos a que:

$$x_1(t) = \frac{d_2^x t^2 + d_1^x t + d_0^x}{a_{02} t^2 + a_{11} t + a_{20}} \quad (1.8)$$

$$y_1(t) = \frac{d_2^y t^2 + d_1^y t + d_0^y}{a_{02} t^2 + a_{11} t + a_{20}} \quad (1.9)$$

donde los d_i^x y d_i^y $i = 0, 1, 2$ dependen del punto (x_0, y_0) y de los coeficientes de la cónica en (1.2).

Específicamente:

$$\begin{aligned} d_2^x &= a_{02} x_0 \\ d_1^x &= -2a_{02} y_0 - a_{01} \\ d_0^x &= -x_0 a_{20} - a_{11} y_0 - a_{10} \end{aligned}$$

mientras que

$$\begin{aligned} d_2^y &= -a_{20} y_0 - a_{01} - a_{11} x_0 \\ d_1^y &= -2x_0 a_{20} - a_{10} \\ d_0^y &= y_0 a_{20} \end{aligned}$$

□

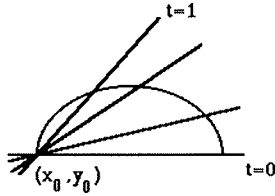


Fig.1 Parametrización de una cónica.

Nótese que si en particular t se restringe al intervalo $[0, 1]$, entonces $(x_1(t), y_1(t))$ describe una sección de la cónica. Esta sección está definida por los puntos en que cortan a la cónica, las rectas que pasan por (x_0, y_0) con pendiente $t = 0$ y $t = 1$ respectivamente (vea la figura 1). En general, cualquier sección de la cónica se puede describir en términos de un parámetro u que tome valores en $[0, 1]$, pues si la sección deseada se obtiene en (1.8) y (1.9) para t en $[a, b]$, basta con hacer el cambio de parámetro $u = (t - a)/(b - a)$, con u en $[0, 1]$. Mediante esta transformación (1.8) y (1.9) se pueden reescribir en términos de u aunque con otros coeficientes. Por eso, con frecuencia vamos a suponer que t toma valores en $[0, 1]$.

1.2 Forma de Bezier de una sección cónica

Cada uno de los polinomios cuadráticos que aparecen en el numerador y denominador de $x_1(t)$ y $y_1(t)$ en (1.8) y (1.9) se puede reescribir en términos de los polinomios de Bernstein de grado 2,

$$B_i^2(t) = \binom{2}{i} t^i (1-t)^{2-i} \quad i = 0, 1, 2$$

teniendo en cuenta que :

$$\begin{aligned} t^2 &= B_2^2(t) \\ t &= 1/2 B_1^2(t) + B_2^2(t) \\ 1 &= B_0^2(t) + B_1^2(t) + B_2^2(t) \end{aligned} \quad (1.10)$$

Nótese que el denominador de $x_1(t)$ y $y_1(t)$ en (1.8) y (1.9) es $c_2(t)$, que en términos de (1.10) se expresa como:

$$c_2(t) = w_0 B_0^2(t) + w_1 B_1^2(t) + w_2 B_2^2(t) \quad \text{donde}$$

$$\begin{aligned}
 w_0 &= a_{20} \\
 w_1 &= a_{20} + 1/2a_{11} \\
 w_2 &= a_{20} + a_{11} + a_{02}
 \end{aligned}$$

De manera similar si sustituimos (1.10) en el numerador de $x_1(t)$ y $y_1(t)$ y denotamos por $b_0, b_1, b_2 \in E^2$ los puntos :

$$\begin{aligned}
 b_0 &= (d_0^x, d_0^y)^t / w_0 \\
 b_1 &= (1/2d_1^x + d_0^x, 1/2d_1^y + d_0^y)^t / w_1 \\
 b_2 &= (d_2^x + d_1^x + d_0^x, d_2^y + d_1^y + d_0^y)^t / w_2
 \end{aligned}$$

entonces la cónica $c(t) = (x_1(t), y_1(t))^t$ se puede escribir como:

$$c(t) = \begin{bmatrix} x_1(t) \\ y_1(t) \end{bmatrix} = \frac{w_0 b_0 B_0^2(t) + w_1 b_1 B_1^2(t) + w_2 b_2 B_2^2(t)}{w_0 B_0^2(t) + w_1 B_1^2(t) + w_2 B_2^2(t)} \quad (1.11)$$

Esta expresión de una cónica mediante los polinomios de Bernstein de grado 2, se conoce como la forma de Bernstein-Bezier de la cónica. Los puntos b_0, b_1 y b_2 forman el llamado polígono de control de la cónica, mientras que w_0, w_1 y w_2 son pesos asociados a los vértices del polígono de control.

La representación de Bernstein-Bezier de una cónica, también se obtiene al proyectar sobre un plano, una curva de Bezier polinomial de grado 2. Por eso en la actualidad, algunos autores [5] [6], incluso definen las cónicas como una proyección de las curvas de Bezier cuadráticas. En tal caso, es fácil ver que los puntos $b_i \in E^2$ $i = 0, 1, 2$ del polígono de control de la cónica son la proyección de los puntos $[w_i b_i, w_i]$ $i = 0, 1, 2$ del polígono de control de la curva de Bezier de grado 2, de la cual proviene la cónica.

1.3 Propiedades de las cónicas.

En esta sección presentamos, a manera de resumen, algunas de las propiedades más importantes de las cónicas. Estas propiedades son la base de muchos de los conceptos y criterios que se desarrollan a lo largo de este trabajo. Además, se utilizan ampliamente en el diseño de un algoritmo para ajustar datos mediante secciones cónicas.

Propiedades

- Una cónica $c(t)$ pasa por los puntos extremos de su polígono de control, es decir:

$$c(0) = b_0 \qquad c(1) = b_2$$

Además es tangente en los extremos a los vectores $\Delta b_0 = b_1 - b_0$ y $\Delta b_1 = b_2 - b_1$ definidos por el polígono de control, de modo que:

$$\dot{c}(0) = \frac{2w_1}{w_0} \Delta b_0 \qquad \dot{c}(1) = \frac{2w_1}{w_2} \Delta b_1$$

Por lo tanto, la "forma" de una cónica está casi determinada por su polígono de control.

- Los pesos w_i influyen de manera importante en la "forma" de una cónica. El conjunto de pesos w_i que describen una cónica no es único. Por ejemplo, si todos los pesos w_i se multiplican por factor no nulo, entonces ese factor aparece en el numerador y en el denominador de (1.11) y se cancela sin producir ningún efecto. Por tanto, los pesos w_i y \hat{w}_i $i=0,1,2$ con $\hat{w}_i = \alpha w_i$ $\alpha \neq 0$, describen la misma cónica.
- Se puede demostrar [5] que también los pesos $\hat{w}_i = \rho^{2-i} w_i$ $i=0,1,2$ con $\rho \neq 0$ describen la misma cónica. Si en particular se toma $\rho = \sqrt{w_2/w_0}$ entonces resulta que $\hat{w}_0 = \hat{w}_2 = w_2$. Por lo tanto, reescalando los pesos con el factor $\alpha = 1/w_2$, tenemos que toda cónica con $w_0, w_2 \neq 0$ puede reescribirse de modo que $\hat{w}_0 = \hat{w}_2 = 1$. Una cónica que satisface esta condición se dice que está en forma estándar.

- Si $c(t)$ es una cónica escrita en la forma de Bezier estándar y denotamos por S el punto $c(1/2)$, conocido como punto de apoyo de la cónica, se puede probar [6], que el punto medio m del segmento $\overline{b_0b_2}$, S y el vértice b_1 son colineales y la razón entre ellos es w_1 (vea la figura 2), o sea:

$$\frac{\|S - m\|^{1/2}}{\|S - b_1\|^{1/2}} = w_1$$

Por lo tanto, cada polígono b_0, b_1, b_2 define una familia de cónicas que dependen del parámetro w_1 . En la medida en que w_1 crece, la cónica asociada a ese parámetro se aproxima al vértice b_1 del polígono de control.

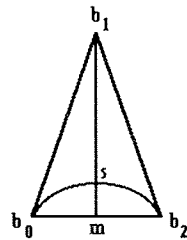


Fig.2 Relación entre m, s, b_1, w_1 .

- Una de las grandes ventajas que tiene escribir una sección cónica en la forma de Bezier estándar es que su clasificación depende únicamente del peso w_1 , de modo que con $w_1 > 0$ se cumple que [12]:
 - si $w_1 < 1$ la curva es una sección de elipse
 - si $w_1 = 1$ es una sección de parábola y
 - si $w_1 > 1$ es una sección de hipérbola.
- La expresión (1.11) de una cónica se puede reescribir como:

$$c(t) = \sum_{i=0}^2 \alpha_i b_i \tag{1.12}$$

donde

$$\alpha_i = \frac{w_i B_i^2(t)}{\sum_{j=0}^2 w_j B_j^2(t)}$$

Como $\alpha_0 + \alpha_1 + \alpha_2 = 1$, de (1.12) resulta que una cónica es una combinación baricéntrica de los puntos de su polígono de control. Por lo tanto, las cónicas son invariantes por transformaciones afines, es decir, si A es afín entonces :

$$A(c(t)) = \sum_{i=0}^2 \alpha_i A(b_i)$$

- Si todos los pesos w_i $i=0,1,2$ son positivos, entonces en (1.11) $c(t)$ es una combinación convexa de los puntos del polígono de control. Por tanto, (1.11) define el segmento de la cónica se encuentra en el triángulo $b_0 b_1 b_2$. El resto de la curva se denomina "segmento" complementario, aunque estrictamente hablando puede ocurrir que no sea un segmento.
- Sea $\tilde{c}(t)$ la curva definida por

$$\tilde{c}(t) = \frac{w_0 b_0 B_0^2(t) - w_1 b_1 B_1^2(t) + w_2 b_2 B_2^2(t)}{w_0 B_0^2(t) - w_1 B_1^2(t) + w_2 B_2^2(t)} \quad (1.13)$$

donde todos los pesos w_i $i=0,1,2$ son positivos. Entonces se puede probar [12] que $\tilde{c}(t)$ y $c(t)$ describen la misma cónica. Más aún, en [12] se demuestra que para cada $t \in [0, 1]$ los puntos $b_1, c(t)$ y $\tilde{c}(t)$ son colineales. Por lo tanto, la ecuación (1.13) con $t \in [0, 1]$ describe el segmento complementario de $c(t)$.

- Las cónicas también son invariantes por transformaciones proyectivas [12], es decir la imagen proyectiva de una cónica sigue siendo una cónica (aunque puede suceder, por ejemplo, que la proyección de una elipse sea una parábola).

Capítulo 2

2.1 Planteamiento general del problema

En términos generales, el problema que vamos a resolver consiste en aproximar por medio de una curva, los vértices del polígono que interpola un conjunto de datos en el plano. La curva de aproximación debe ser tal, que el polígono "suavizado" se parezca lo más posible al original. Esta curva es la frontera de una región plana, sobre la cual se construirá una malla. Por eso, una vez generada la curva de aproximación, es necesario escoger cuatro puntos sobre la misma, que subdividirán la frontera en igual número de subfronteras. Finalmente, se debe seleccionar un número fijo de puntos en cada subfrontera. La distribución de los puntos sobre la frontera debe ser controlada, ya que estos son los extremos de la malla a construir.

Este problema surgió vinculado al de generación de mallas óptimas, que investiga el grupo dirigido por el Dr. Pablo Barrera, de la Universidad Autónoma de México. Como es conocido, la generación de una malla sobre una región del plano, es el primer paso para la solución numérica de una ecuación diferencial en derivadas parciales, utilizando el método de los elementos finitos.

Por otro lado, en la construcción de una malla con propiedades óptimas (celdas convexas y con fronteras suaves) es importante que la frontera de la región sea una curva regular y suave.

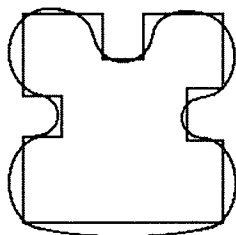


Fig.3 Spline cúbico que suaviza un polígono.

Cuando la frontera es muy irregular se puede aproximar por medio de un spline cúbico paramétrico. Sin embargo, cuando la frontera de la región es un polígono no convexo, la aproximación mediante un spline de suavizamiento [2] no da buenos resultados, pues si bien la nueva frontera es suave, muchas veces no se parece al polígono original. Vea la figura 3.

Por eso, la curva mediante la cual vamos a aproximar el polígono definido por los datos, debe conservar, siempre que sea posible, las secciones rectas del mismo, suavizando únicamente sus vértices. Ya con anterioridad, Liao [13] y Pavlidis [15] consideraron en sus trabajos, la conveniencia de utilizar cónicas para ajustar un conjunto de datos. Esta técnica también se emplea en el diseño automatizado de formas geométricas, por ejemplo los caracteres alfanuméricos que utilizan los procesadores de texto.

Veamos a continuación las ideas generales del método de ajuste que vamos a emplear. Denotemos por $I_i = (x_i, y_i)$ $i=0, \dots, m-1$ los puntos del plano que definen el polígono a aproximar, es decir los vértices del polígono. La esencia del método consiste en sustituir cada vértice I_i por una sección cónica $c_i(t)$, cuando el ángulo θ_i que se forma en I_i no es "demasiado" llano (específicamente si $\theta_i < 172^\circ$). En caso contrario, no tiene sentido considerar ese vértice y el punto I_i se elimina, pues es casi colineal con sus vecinos.

Suponiendo que $c_i(t)$ se escribe en la forma estándar, tenemos que calcular entonces los puntos b_0^i, b_1^i y b_2^i de su polígono de control y

el peso w_1^i . Un requisito importante que debe cumplir la curva a construir, es que las secciones cónicas se unan "suavemente" con las secciones rectas. Por eso, vamos a tomar como tangentes a cada cónica $c_i(t)$ en los extremos, los lados que definen el vértice I_i . De acuerdo con las propiedades de las cónicas que mencionamos en el Capítulo anterior, esto significa que b_1^i se tomará como el vértice I_i que queremos suavizar.

De este modo, el problema se reduce a determinar, sobre un lado dado, en qué posición se escoge el punto inicial b_0^i y el punto final b_2^i de la sección cónica $c_i(t)$, como se muestra en la figura 4.

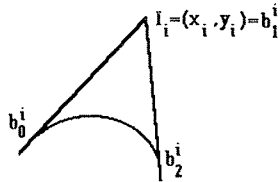


Fig.4 ¿Dónde van los puntos extremos de la cónica que suaviza un vértice?

Para automatizar el proceso de selección, de los vértices extremos del polígono de control de cada cónica, hay que tener en cuenta las longitudes de los lados del polígono definido por los datos. En ese sentido, vamos a considerar por separado el caso de un polígono regular, o sea aquel cuyos lados tienen todos la misma longitud, del de un polígono irregular.

Si el polígono a suavizar tiene todos sus lados de tamaño l , entonces resulta natural que todas las cónicas comiencen y terminen a la misma distancia del vértice que van a ajustar. Por eso, en nuestro algoritmo se toma el punto inicial y final de cada cónicas a una distancia $l/3$ del vértice correspondiente. Como consecuencia de esto, dos cónicas consecutivas se unirán en este caso por medio de una recta, como se aprecia en la figura 5 y en el cuadrado de la figura 12.

Si por el contrario, los lados del polígono tienen diferentes longitudes, entonces las cónicas no deben empezar y terminar a la misma distancia del vértice correspondiente. En este caso, es lógico que se conserven

secciones rectas, en aquellos lados de mayor tamaño. Por eso, en nuestro algoritmo, se calcula la longitud mínima l_{min} de todos los lados y se va comparando en cada paso con la longitud l_i de un lado. La idea es que la curva de ajuste incluya una sección recta sobre un lado, si su longitud l_i es mayor que $2l_{min}$. Esto se logra tomando los puntos extremos de las cónicas asociadas a los vértices I_{i-1} e I_i a una distancia $l_{min}/2$ de sus respectivos vértices, como muestra la figura 5.

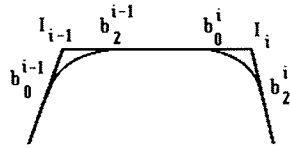


Fig.5 El lado l_i incluye una sección recta.

Si la longitud l_i de un lado es menor o igual que $2l_{min}$, entonces la curva puede aproximar ese lado del polígono, sin necesidad de incluir una sección recta sobre el mismo. Por eso en el algoritmo que proponemos, el punto final del polígono de control de la cónica $c_{i-1}(t)$ coincide, en este caso, con el inicial de $c_i(t)$ como se muestra en la figura 6, y ambos se toman como el punto medio de ese lado.

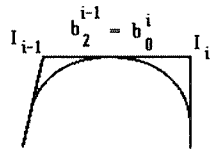


Fig.6 El lado l_i no incluye ninguna sección recta.

Finalmente, para que quede bien definida cada una de las cónicas que constituyen la curva, debemos decidir qué peso w_1^i se le asocia a cada $c_i(t)$. En otras palabras, debemos escoger una cónica de toda la familia que depende de ese parámetro. Como mencionamos en el Capítulo 1, el valor de w_1^i está relacionado con la distancia que existe entre el punto de apoyo de la cónica y su vértice b_1^i . En la siguiente sección veremos cómo calcular el peso intermedio de una cónica, si deseamos que su polígono se encuentre a una "distancia" prefijada de la misma.

2.2 Distancia de un punto a una sección cónica y del polígono al spline

En esta sección vamos definir la distancia entre un punto situado en el triángulo de control de una sección cónica y la misma. Esto nos permitirá dar una medida de la calidad del ajuste construido, al poder cuantificar la distancia entre el polígono definido por los datos y el spline cónico de ajuste.

Definición 1 Sea $c(t)$ una sección cónica con polígono de control b_0, b_1, b_2 y sea O un punto situado en el triángulo $\Delta b_0 b_1 b_2$. Si denotamos por S el punto de apoyo de la cónica y por A su complementario, entonces la distancia entre O y $c(t)$, $d(O, c(t))$, se define como la longitud del segmento \overline{OP} , donde P es el punto de intersección de la sección cónica, con la recta que pasa por O y A .

A continuación probaremos que entre los puntos situados en los segmentos $\overline{b_0 b_1}$ y $\overline{b_1 b_2}$, el más alejado de la sección cónica es precisamente el vértice b_1 .

Teorema 2 Sea $\Delta b_0 b_1 b_2$ el triángulo de control de la sección cónica $c(t)$ y sea O un punto del segmento $\overline{b_0 b_1}$ o del segmento $\overline{b_1 b_2}$. Entonces

$$\max_{0 \leq t \leq 1, O \in \overline{b_0 b_1} \cup \overline{b_1 b_2}} d(O, c(t)) = \|S - b_1\|^{1/2}$$

donde S es el punto de apoyo de la cónica y $\|\cdot\|$ es la norma euclídeana cuadrática.

Demostración

Sea r la recta que pasa por b_0 y el punto de apoyo S . Por el teorema de Bezout, es conocido que una recta y una cónica se intersectan a lo sumo en dos puntos [8]. Por tanto, la gráfica de la sección cónica contenida entre b_0 y S tiene que estar, por encima o por debajo de r .

Por otro lado, como la cónica es tangente en b_0 , a la recta que pasa por b_0 y b_1 , entonces el segmento de cónica entre b_0 y S se halla por encima de la recta r .

Sea O un punto del segmento $\overline{b_0b_1}$ y A el complementario del punto de apoyo S , osea $A = \tilde{c}(1/2)$ con $\tilde{c}(t)$ dada por (1.13). Sean P y Q los puntos de intersección de la recta que pasa por O y A con la cónica y la recta r respectivamente. Entonces como se evidencia en la figura 7 tenemos que:

$$\|P - O\| \leq \|Q - O\|$$

Por definición,

$$\|P - O\|^{1/2} = d(O, c(t))$$

Por otro lado, como el punto de apoyo S , el vértice b_1 y A son colineales, se cumple que:

$$d(b_1, c(t)) = \|b_1 - S\|^{1/2}$$

y además resulta evidente que:

$$\|Q - O\| \leq \|b_1 - S\|$$

para todo $O \in \overline{b_0b_1}$. Luego:

$$d(O, c(t)) = \|P - O\|^{1/2} \leq \|Q - O\|^{1/2} \leq \|b_1 - S\|^{1/2} = d(b_1, c(t))$$

para todo $O \in \overline{b_0b_1}$.

Análogamente se demuestra para $O \in \overline{b_1 b_2}$. □

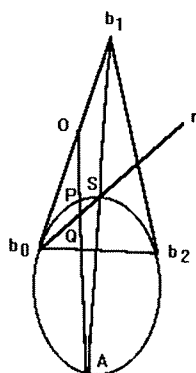


Fig. 7 Demostración del Teorema 2

A lo largo del trabajo hemos mencionado varias veces que la curva que se utilizará para ajustar nuestros datos es un spline cónico. Por eso, a continuación se define precisamente una curva de ese tipo.

Definición 2 Sea $u_0 < u_1 < \dots < u_l$ con $u_i \in \mathcal{R}$ $i = 0, \dots, l$. Un spline cónico es una curva continua $s(u)$, que en cada segmento $[u_i, u_{i+1}]$ se reduce a una sección cónica $c_i(u)$. Los puntos u_i se llaman nodos del spline.

Nótese que como una sección cónica puede ser una recta (cuando los vértices del polígono de control son colineales), entonces un spline cónico puede incluir segmentos de cónicas y segmentos de rectas.

Una vez que ha definido la distancia entre un punto y una sección cónica, necesitamos extender la definición a la distancia entre el polígono constituido por los datos originales y el spline cónico de ajuste. Esto nos permitirá dar una medida de la calidad del ajuste.

Definición 3 Sea P el polígono definido por los datos originales $I_i = (x_i, y_i)$ $i = 0, \dots, m - 1$ y denotemos por $s(u)$ el spline cónico

constituido por l secciones cónicas y rectas, $l \geq m$, donde cada cónica $c_i(u)$ aproxima el i -ésimo vértice de P , $i = 0, \dots, m-1$. Entonces la distancia entre P y $s(u)$, $d(P, s(u))$, se define como:

$$d(P, s(u)) = \sum_{i=0}^{m-1} d(P_i, c_i(u)) \quad (2.1)$$

donde $P_i = (b_0^i, b_1^i, b_2^i)$ es polígono de control de $c_i(u)$ y, de acuerdo con el Teorema 2, $d(P_i, c_i(u)) := d_i = \|S^i - b_1^i\|^{1/2}$ con S^i el punto de apoyo de $c_i(u)$.

Según vimos en el Capítulo 1, cada triángulo de control está asociado con una familia de cónicas estándares, que depende del peso intermedio. Si denotamos por w_1^i el peso intermedio de cualquiera de las cónicas, cuyo polígono de control es el triángulo P_i , resulta claro que el valor máximo de d_i se alcanza cuando $c_i(u)$ es la recta que pasa por b_0^i y b_2^i , es decir para $w_1^i = 0$. En tal caso,

$$d(P_i, c_i(u)) = \|b_m^i - b_1^i\|^{1/2} := e_i$$

donde b_m^i es el punto medio del segmento $\overline{b_0^i b_2^i}$.

Similarmente se tiene que la distancia $d_i \rightarrow 0$ cuando $w_1^i \rightarrow \infty$. Por lo tanto, la distancia (2.1) entre el polígono P definido por los datos y el spline $s(u)$ se puede normalizar a $(0,1]$ tomando

$$d(P, s(u)) = \sum_{i=0}^{m-1} d_i / \sum_{i=0}^{m-1} e_i \quad (2.2)$$

donde $d_i = \|S^i - b_1^i\|^{1/2}$ y $e_i = \|b_m^i - b_1^i\|^{1/2}$.

Con esta nueva definición, la distancia entre el polígono que forman los datos y el spline de ajuste sería 0 (en el límite) si no suavizáramos ningún vértice y sería 1 si "truncáramos" cada vértice I_i , sustituyéndolo por el segmento de recta $\overline{b_0^i b_2^i}$.

Ahora ya estamos en condiciones de resolver el siguiente problema. Dado un polígono P y $0 < \varepsilon < 1$, contruir un spline cónico $s(u)$, tal

que $d(P, s(u)) = \varepsilon$.

Una posible solución del problema anterior consiste en "distribuir uniformemente" la distancia ε , entre todas las cónicas que constituyen el spline, es decir tomar en (2.2) $d_i = \varepsilon e_i$ para todo i . Pero una de las propiedades de las cónicas nos asegura que

$$w_1^i = \frac{\|S^i - b_m^i\|^{1/2}}{\|S^i - b_1^i\|^{1/2}}$$

Por lo tanto, para lograr que $d(P, s(u)) = \varepsilon$ basta con aproximar el i -ésimo vértice del polígono P , con una cónica estándar con peso w_1^i dado por:

$$w_1^i = \frac{e_i - d_i}{d_i} = \frac{e_i - \varepsilon e_i}{\varepsilon e_i} = \frac{1 - \varepsilon}{\varepsilon} \quad (2.3)$$

Nótese que cuando $\varepsilon \rightarrow 0$, $w_1^i \rightarrow \infty$ y cuando $\varepsilon \rightarrow 1$, $w_1^i \rightarrow 0$. Por otro lado, de (2.3) resulta que en esta solución del problema planteado, todas las cónicas que suavizan los vértices del polígono P tienen el mismo peso intermedio.

2.3 Problema de inversión

Uno de los problemas que aparecen con más frecuencia cuando se trabaja con curvas planas racionales, y en particular con cónicas, es el problema de inversión. Dado un punto P situado sobre la curva racional paramétrica $s(u)$, el problema de inversión [9], [23] consiste en determinar el valor de u tal que $s(u) = P$.

Como veremos a continuación, para generar los puntos de la frontera que se mantendrán fijos como extremos de la malla, debemos resolver también el problema de inversión. Supongamos que ya se ha construido el spline cónico $s(u)$ que aproxima la frontera de la región, entonces el siguiente paso consiste en subdividir la misma en cuatro subfronteras, de modo que las líneas "horizontales" de la malla a construir irán de la subfrontera 1 a la 3, mientras que las "verticales"

unirán puntos de la subfrontera 2 con puntos de la 4.

Sean R_1, R_2, R_3 y R_4 los cuatro puntos que subdividen la frontera y denotemos por $\tilde{u}_1, \tilde{u}_2, \tilde{u}_3, \tilde{u}_4$ los valores del parámetro u tales que $s(\tilde{u}_i) = R_i$ $i = 1, \dots, 4$, es decir las soluciones del problema de inversión para los extremos de las subfronteras. Si se desea construir una malla de orden n_i x n_j , entonces se deben seleccionar n_i puntos fijos en cada una de las subfronteras 1 y 3 y n_j en cada una de las dos restantes. Los puntos de la subfrontera 1, por ejemplo, se obtienen evaluando el spline cónico en n_i valores del parámetro u , todos contenidos en el intervalo $[\tilde{u}_1, \tilde{u}_2]$. De manera similar hay que proceder con el resto de las subfronteras.

Por eso, en esta sección vamos a dar algunas ideas de cómo se resuelve el problema de inversión para una recta y para una cónica, puesto que el spline $s(u)$ está constituido por segmentos de estas curvas.

Comencemos por el caso de una recta. Sea $r(t)$ el segmento de recta cuyos puntos extremos son $P = (P_x, P_y)^t$ y $Q = (Q_x, Q_y)^t$ y sea $O = (O_x, O_y)^t$ un punto del segmento PQ . Entonces existe t $0 \leq t \leq 1$ tal que

$$r(t) = (1-t) \begin{bmatrix} P_x \\ P_y \end{bmatrix} + t \begin{bmatrix} Q_x \\ Q_y \end{bmatrix} = \begin{bmatrix} O_x \\ O_y \end{bmatrix} \quad (2.4)$$

De (2.4) resulta inmediatamente que si $P_x - Q_x \neq 0$ entonces

$$t = \frac{P_x - O_x}{P_x - Q_x} \quad (2.5)$$

En otro caso,

$$t = \frac{P_y - O_y}{P_y - Q_y} \quad (2.6)$$

En la práctica, debido a la necesidad de utilizar una aritmética finita, el punto O no pertenece generalmente a $r(t)$. Si en tal situación utilizamos la fórmula de inversión (2.5), entonces el valor \hat{t} obtenido es tal que $r(\hat{t})$ es la proyección de O sobre $r(t)$ paralela al eje y . Pero la proyección paralela a uno de los ejes coordenados puede estar muy alejada del punto en cuestión. Por eso si O no está en $r(t)$ es más

conveniente calcular su proyección ortogonal sobre la recta y utilizar la fórmulas (2.5) y (2.6) para el punto proyectado.

En el caso de una cónica, la solución del problema de inversión es mucho más complicada y requiere del uso de una herramienta algebraica muy importante, que forma parte de la teoría de eliminación y se conoce como resultante [25], [26]. Tanto el problema de inversión, como el de implicitación (obtener la ecuación implícita de una curva plana paramétrica y racional) fueron resueltos en la segunda mitad del siglo XIX [22], pero sólo recientemente [9],[23],[27], se han extendido las técnicas clásicas a los polinomios de Bernstein y las curvas de Bezier, que como se sabe juegan un rol central en los problemas de CAGD.

Las resultantes nos permiten determinar si dos polinomios $f(t)$ y $g(t)$ de grado N y linealmente independientes tienen o no una raíz común, sin necesidad de calcular todas las raíces de cada uno de ellos. En general, la idea consiste en obtener N polinomios de grado $N - 1$, cuyo anulamiento simultáneo sea equivalente al anulamiento de los dos polinomios originales. La resultante de Bezout, que denotaremos por $R(f, g)$, se define entonces como una matriz $N \times N$ cuyos elementos R_{ij} $i, j = 0, \dots, N - 1$ dependen linealmente de los coeficientes de f y g y se tiene el siguiente resultado central [9].

Teorema 3 *Los polinomios $f(t)$ y $g(t)$ tienen una raíz común si y sólo si $\text{Det } [R(f, g)] = 0$.*

En [9] también se demuestra que cuando f y g tienen una raíz común t , la misma se calcula resolviendo un sistema de ecuaciones lineales, cuya matriz es el menor principal de orden $N - 1$ de $R(f, g)$, mientras que el lado derecho es la última columna de la propia matriz y las incógnitas son t^{N-1}, \dots, t^2, t . Por lo tanto, se cumple el siguiente teorema [9].

Teorema 4 *Si los polinomios $f(t)$ y $g(t)$ tienen una raíz común t_0 entonces:*

$$t_0 = \frac{\text{Cofactor}[R_{N-1,N-2}(f, g)]}{\text{Cofactor}[R_{N-1,N-1}(f, g)]}$$

donde $R_{N-1,N-1}$ es el menor principal de orden $N - 1$ de R y $R_{N-1,N-2}$ es la matriz que resulta de eliminar en R la fila $N - 1$ y la columna $N - 2$.

Estos resultados se aplican directamente a la solución del problema de inversión para una curva plana paramétrica y racional. En efecto, sea

$$x = \frac{f(t)}{w(t)} \quad y = \frac{g(t)}{w(t)}$$

con $f(t)$, $g(t)$ y $w(t)$ polinomios de grado N , la ecuación paramétrica de la curva $c(t)$. Definamos los polinomios auxiliares:

$$\begin{aligned} p(x, t) &= w(t)x - f(t) \\ q(y, t) &= w(t)y - g(t) \end{aligned}$$

Entonces un punto $O = (x, y)$ pertenece a $c(t)$ si y sólo si existe un t_0 tal que:

$$\begin{aligned} p(x, t_0) &= 0 \\ q(y, t_0) &= 0 \end{aligned}$$

es decir si $p(x, t)$ y $q(y, t)$ tienen una raíz común t_0 .

Pero según el teorema 3 esto ocurre cuando $\text{Det } R[p, q] = 0$. Nótese que como polinomios en t , los coeficientes de p y q dependen linealmente de x y y respectivamente. Por lo tanto, los elementos de la matriz resultante dependen linealmente de x y y y $\text{Det } R[p, q]$ es un polinomio de grado N en x y y . Luego, del teorema 4 se concluye que t_0 es el cociente de dos polinomios de grado $N - 1$ en x y y .

En particular, cuando $f(t)$, $g(t)$ y $w(t)$ están escritas en términos de los polinomios de Bernstein, como ocurre con las curvas de Bezier, se pueden deducir expresiones explícitas para los elementos $R_{ij}(p, q)$ de

la matriz resultante, las cuales dependen de los puntos y pesos del polígono de control de la curva [9].

De lo anterior se concluye, que en el caso de una cónica escrita en la forma de Bezier, la fórmula de inversión es del tipo:

$$t = \frac{N(x, y)}{M(x, y)} \quad (2.7)$$

donde $N(x, y)$ y $M(x, y)$ son polinomios lineales en x y y que dependen de los puntos b_0 , b_1 y b_2 del polígono de control y de los pesos w_0 , w_1 y w_2 asociados a los mismos.

La ecuación (2.7) define un haz de rectas que dependen del parámetro t . Por eso, si el punto $O = (x, y)$ no pertenece a la cónica, el valor \hat{t} obtenido de (2.7) es tal que $\hat{O} := c(\hat{t})$ es la proyección de O sobre la cónica. El centro de proyección es la intersección de las rectas $N(x, y) = 0$ y $M(x, y) = 0$ [27]. Mas aún, en [27] se prueba que para cada punto \hat{A} situado sobre la cónica existe una fórmula de inversión $t = \tilde{N}(x, y)/\tilde{M}(x, y)$ que geoméricamente corresponde a proyectar, con centro en \hat{A} , sobre la cónica.

Tal como vimos en el problema de inversión para una recta, la selección del centro de proyección es importante desde el punto de vista geométrico, pues una selección inadecuada puede provocar que la proyección de un punto O que no esté sobre la curva, sea otro punto $\hat{O} = c(\hat{t})$ relativamente alejado de O . Por eso, nos interesa deducir una fórmula de inversión, que a cada O que no esté sobre la cónica $c(t)$, le asocie el punto \hat{O} de $c(t)$ que define la distancia entre O y $c(t)$ (ver definición 1).

Como la distancia a una cónica se mide en términos del haz de rectas que pasan por el complementario A del punto de apoyo S , entonces resulta conveniente tomar a A como centro de proyección. Denotemos por $b_0 = (b_{0x}, b_{0y})$, $b_1 = (b_{1x}, b_{1y})$, $b_2 = (b_{2x}, b_{2y})$ los puntos del polígono de control. Entonces las coordenadas del punto de apoyo

$S = (S_x, S_y)$ se obtienen de

$$S = c(1/2) = \frac{b_0 + 2w_1b_1 + b_2}{2(1 + w_1)}$$

mientras que las de su complementario $A = (A_x, A_y)$ se obtienen de

$$A = \tilde{c}(1/2)$$

con $\tilde{c}(t)$ dada por (1.13). De aquí se llega a que

$$A = \frac{b_0 - 2w_1b_1 + b_2}{2(1 - w_1)}$$

Por otro lado, sabemos que $A = (A_x, A_y)$ es la intersección de las rectas

$$\begin{aligned} N(x, y) &:= n_{00} + n_{10}x + n_{01}y = 0 \\ M(x, y) &:= m_{00} + m_{10}x + m_{01}y = 0 \end{aligned}$$

que definen la fórmula de inversión (2.7). Por lo tanto, se debe cumplir que:

$$N(A_x, A_y) = 0 \tag{2.8}$$

$$M(A_x, A_y) = 0 \tag{2.9}$$

Vamos a exigir además que en la fórmula de inversión $t = 0$ para el vértice b_0 , $t = 1$ para el vértice b_2 y $t = 1/2$ para el punto de apoyo S . Esto equivale a exigir que se satisfagan las siguientes ecuaciones:

$$N(b_{0x}, b_{0y}) = 0 \tag{2.10}$$

$$N(b_{2x}, b_{2y}) - M(b_{2x}, b_{2y}) = 0 \tag{2.11}$$

$$2N(S_x, S_y) - M(S_x, S_y) = 0 \tag{2.12}$$

Sustituyendo en las ecuaciones (2.8), (2.9) y (2.12) las coordenadas de A y S y resolviendo el sistema de ecuaciones lineales (2.8)-(2.12) con el sistema de computación simbólica MAPLE V, se obtuvieron expresiones para los coeficientes de las rectas $N(x, y) = 0$ y $M(x, y) = 0$, los cuales dependen de los puntos del polígono de control y de w_1 .

Denotemos por $\det(P_0, P_1, P_2)$ al determinante asociado a los puntos $P_0 = (x_0, y_0)$, $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$

$$\det(P_0, P_1, P_2) = \begin{vmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ 1 & 1 & 1 \end{vmatrix}$$

Entonces las rectas $N(x, y) = 0$ y $M(x, y) = 0$, que se obtienen al resolver el sistema de ecuaciones lineales, se pueden escribir de manera compacta como:

$$\begin{aligned} N(x, y) &= 2w_1 \det(O, b_0, b_1) + \det(O, b_2, b_0) \\ M(x, y) &= 2[(1 - w_1) \det(O, b_2, b_0) + w_1 \det(b_0, b_1, b_2)] \end{aligned}$$

Por tanto, la fórmula de inversión para O con centro en el complementario A del punto de apoyo es:

$$t = \frac{2w_1 \det(O, b_0, b_1) + \det(O, b_2, b_0)}{2[(1 - w_1) \det(O, b_2, b_0) + w_1 \det(b_0, b_1, b_2)]} \quad (2.13)$$

2.4 Necesidad de reparametrizar las cónicas

Como mencionamos en la sección anterior, una vez que se ha diseñado el spline cónico que aproxima la frontera de la región, es necesario seleccionar un conjunto de puntos sobre la misma, cuya cantidad depende de las dimensiones de la malla que se va a construir. Estos puntos se mantendrán fijos como extremos de la malla, por lo que su distribución es muy importante, ya que influye en las propiedades de la misma.

En esta sección vamos a ver cómo se pueden generar puntos sobre una cónica, cuya distribución sea aproximadamente uniforme, en términos de la longitud de arco. Este se puede considerar como el primer paso en el control de la posición de los puntos de la frontera, pues cualquier otro tipo de distribución (exponencial, concentrada alrededor de un

punto fijo, etc) se puede obtener, de forma relativamente simple, a partir la distribución uniforme.

Supongamos entonces que la i -ésima sección del spline es la cónica $c(u) = (x(u), y(u))$ $u \in [u_i, u_{i+1}]$ sobre cuya traza se necesitan $n + 1$ puntos. La forma más natural de calcular estos puntos consiste en pasar al parámetro local

$$t = v(u) = \frac{u - u_i}{u_{i+1} - u_i} \quad t \in [0, 1]$$

y evaluar en $n + 1$ valores equidistantes de t en $[0, 1]$:

$$t_j = j/n \quad j = 0, \dots, n$$

Sin embargo, si se procede de esta manera, los puntos que se obtienen sobre la gráfica de la cónica aparecen concentrados hacia el centro o hacia los extremos de la curva, en dependencia de si w_1^i es mayor o menor que 1. En [7] Farouki y Sakkalis hacen referencia a este "defecto" de las curvas racionales.

Esto explica por qué es necesario reparametrizar las cónicas, si se quiere generar puntos con distribución uniforme. De manera exacta, el resultado deseado sólo se puede obtener si parametrizamos la cónica en términos de su longitud de arco, lo cual resulta una tarea extremadamente costosa. Sin embargo, existen varios métodos para lograrlo aproximadamente. En [3] se considera uno de ellos, basado en el algoritmo de Taubin. A continuación se presenta otra solución del problema, que se basa en la aproximación de la parametrización en términos de la longitud de arco.

Es conocido que si una curva $c(t)$ se reescribe en términos de la longitud de arco

$$s = s(t) = \int_{t_0}^t \|\dot{c}(t)\| dt$$

entonces el vector tangente es unitario en todos los puntos, es decir $\|\dot{c}(s)\| = 1$ para todo s .

Por eso, una forma de aproximar s consiste en tomar $s = h(t)$, donde $h(t)$ es una función que vamos a construir de modo que la norma del vector tangente $\|\dot{c}(s = h(t))\|$ sea 1 en los puntos extremos y en el punto de apoyo, es decir para $s = 0, 1/2, 1$. Teniendo en cuenta la regla de la cadena resulta que

$$\begin{aligned} \|\dot{c}(u)\|^2 &= \left(\frac{dx}{du}\right)^2 + \left(\frac{dy}{du}\right)^2 = \left(\frac{dx}{ds} \frac{ds}{du}\right)^2 + \left(\frac{dy}{ds} \frac{ds}{du}\right)^2 \\ &= \left(\frac{ds}{du}\right)^2 \left[\left(\frac{dx}{ds}\right)^2 + \left(\frac{dy}{ds}\right)^2 \right] \end{aligned} \quad (2.14)$$

Aproximemos las derivadas respecto a s en (2.14) por los cocientes incrementales

$$\begin{aligned} \frac{dx}{ds} &\approx \frac{x(s + \Delta s) - x(s)}{\Delta s} \\ \frac{dy}{ds} &\approx \frac{y(s + \Delta s) - y(s)}{\Delta s} \end{aligned}$$

Entonces

$$\|\dot{c}(u)\|^2 \approx \left(\frac{ds}{du}\right)^2 \left[\left(\frac{x(s + \Delta s) - x(s)}{\Delta s}\right)^2 + \left(\frac{y(s + \Delta s) - y(s)}{\Delta s}\right)^2 \right] \quad (2.15)$$

Utilizando que $c(s)$ está escrita en la forma estándar de Bezier

$$c(s) = \begin{bmatrix} x(s) \\ y(s) \end{bmatrix} = \frac{b_0(1-s)^2 + 2w_1b_1s(1-s) + b_2s^2}{(1-s)^2 + 2w_1s(1-s) + s^2}$$

calculamos mediante el sistema de computación algebraica MAPLEV el lado derecho de (2.15) y obtuvimos

$$\|\dot{c}(u)\|^2 \approx \left(\frac{ds}{du}\right)^2 \psi(s, \Delta s, w_1, B) \quad (2.16)$$

donde $\psi(s, \Delta s, w_1, B)$ es una función que depende de $s, \Delta s, w_1$ y el polígono de control $B = (b_0, b_1, b_2)$ de la cónica. En particular para $s = 0, 1/2, 1$ se obtuvo

$$\psi(0, \Delta s, w_1, B) = 4w_1^2 \|b_0 - b_1\|^2 \quad (2.17)$$

$$\psi(1/2, \Delta s, w_1, B) = \frac{4\|b_0 - b_2\|^2}{(1 + w_1)^2} \quad (2.18)$$

$$\psi(1, \Delta s, w_1, B) = 4w_1^2 \|b_1 - b_2\|^2 \quad (2.19)$$

Para garantizar que los valores de s se mantengan en el intervalo $[0, 1]$, vamos a exigir que la función $h(t)$ satisfaga las siguientes condiciones

$$h(0) = 0, \quad h(1/2) = 1/2, \quad h(1) = 1$$

Denotemos por

$$m_0 = \dot{h}(0), \quad m_{1/2} = \dot{h}(1/2), \quad m_1 = \dot{h}(1)$$

Entonces, para $u = u_i$, $t = v(u_i) = 0$ y $s = h(0) = 0$. Por lo tanto

$$\left. \frac{ds}{du} \right|_{u=u_i} = \left. \frac{ds}{dt} \right|_{t=0} \left. \frac{dt}{du} \right|_{u=u_i} = \frac{\dot{h}(0)}{\Delta_i} = \frac{m_0}{\Delta_i} \quad (2.20)$$

donde $\Delta_i = u_{i+1} - u_i$. Sustituyendo (2.20) en (2.16) y teniendo en cuenta (2.17) obtenemos

$$\|\dot{c}(u_i)\|^2 \approx \left(\frac{m_0}{\Delta_i} \right)^2 4w_1^2 \|b_0 - b_1\|^2$$

Por lo tanto, exigir que la norma del vector tangente sea 1 en el extremo inicial de la cónica, es equivalente a pedir que se cumpla

$$\left(\frac{m_0}{\Delta_i} \right)^2 4w_1^2 \|b_0 - b_1\|^2 = 1$$

de donde se obtiene que la pendiente m_0 de $h(t)$ en $t = 0$ debe ser

$$m_0 = \frac{\Delta_i}{2w_1 \|b_0 - b_1\|} \quad (2.21)$$

De manera similar se puede ver que de las ecuaciones (2.16) y (2.19) resulta que la pendiente m_1 de $h(t)$ en $t = 1$ debe ser

$$m_1 = \frac{\Delta_i}{2w_1 \|b_1 - b_2\|} \quad (2.22)$$

mientras que de (2.16) y (2.18) se obtiene que la pendiente m_{12} de $h(t)$ en $t = 1/2$ debe satisfacer

$$m_{12} = \frac{\Delta_i(1 + w_1)}{2\|b_0 - b_2\|} \quad (2.23)$$

Vamos a construir $h(t)$ como un spline formado por dos secciones cónicas

$$s = h(t) = \begin{cases} h_1(t) & \text{para } 0 \leq t \leq 1/2 \\ h_2(t) & \text{para } 1/2 \leq t \leq 1 \end{cases} \quad (2.24)$$

donde $h_1(t) = (x_1(t), y_1(t))$ y $h_2(t) = (x_2(t), y_2(t))$. Si tomamos $x_i(t) = t$, $i = 1, 2$, entonces las ecuaciones implícitas para las cónicas $h_i(t)$ son del tipo $f_i(t, y_i) = 0$ $i = 1, 2$, con $f_i(t, y_i)$ un polinomio de segundo grado en t y y_i .

La curva $f_1(t, y_1)$ tiene que cumplir los requisitos que le exigimos a $h(t)$ para $t = 0$ y $t = 1/2$, es decir debe pasar por los puntos $(0, 0)$ y $(1/2, 1/2)$ y la pendiente de la recta tangente a f_1 en esos puntos, debe ser m_0 y m_{12} respectivamente. Esto equivale a exigir que $f_1(t, y_1)$ satisfaga las siguientes condiciones:

$$f_1(0, 0) = 0 \quad f_1(1/2, 1/2) = 0$$

$$\left(\frac{\partial f_1}{\partial t} / \frac{\partial f_1}{\partial y_1} \right) \Big|_{(0,0)} = -m_0$$

$$\left(\frac{\partial f_1}{\partial t} / \frac{\partial f_1}{\partial y_1} \right) \Big|_{(1/2, 1/2)} = -m_{12}$$

De aquí se obtiene un sistema de ecuaciones lineales en los 6 coeficientes de f_1 . Si además exigimos que el coeficiente de y_1^2 sea cero, entonces se llega a una expresión lineal en y_1 y cuadrática en t . Despejando y_1 de la misma, resulta la siguiente fórmula para $h_1(t)$

$$h_1(t) = \begin{cases} \frac{2(m_0 m_{12} - 1)t^2 + (m_0 - m_0 m_{12})t}{2(m_{12} + m_0 - 2)t + 1 - m_{12}} & \text{si } \delta_0 > 0 \text{ y } \gamma_0 \neq 2 \\ t & \text{si } \delta_0 \leq 0 \text{ y } \gamma_0 \neq 2 \\ (2 - m_0)t^2 + m_0 t & \text{si } \gamma_0 = 2 \end{cases} \quad (2.25)$$

donde $\delta_0 = (1 - m_0)(m_{12} - 1)$ y $\gamma_0 = m_0 + m_{12}$.

Análogamente se procede con $f_2(t, y_2)$ obteniéndose la siguiente fórmula para $h_2(t)$

$$h_2(t) = \begin{cases} 0.5 + \frac{2(m_{12}m_1 - 1)(t - 0.5)^2 + (m_{12} - m_{12}m_1)(t - 0.5)}{2(m_1 + m_{12} - 2)(t - 0.5) + 1 - m_1} & \text{si } \delta_1 > 0 \text{ y } \gamma_1 \neq 2 \\ t & \text{si } \delta_1 \leq 0 \text{ y } \gamma_1 \neq 2 \\ 0.5 + (2 - m_{12})(t - 0.5)^2 + m_{12}(t - 0.5) & \text{si } \gamma_1 = 2 \end{cases} \quad (2.26)$$

donde $\delta_1 = (1 - m_{12})(m_1 - 1)$ y $\gamma_1 = m_{12} + m_1$.

Ejemplo

Consideremos la cónica estándar con peso $w_1 = 2.5$ y polígono de control $b_0 = (1, 0)$, $b_1 = (0, 0)$ y $b_2 = (1, 0.5)$. Vamos a generar $n = 20$ puntos sobre la misma utilizando dos métodos diferentes:

1. Evaluando directamente $c(t)$ en 20 valores equidistantes de t en $[0, 1]$ $t_i = (i - 1)/19$ $i = 1, \dots, 20$.
2. Reparametrizando $c(s) = c(h(t))$ y evaluando en los valores de s que se obtienen a partir de los t_i equidistantes, es decir en $s_i = h(t_i)$ $i = 1, \dots, 20$ con $h(t)$ dada por (2.24), (2.25) y (2.26).

Para comparar numéricamente los resultados de este experimento, calculamos la razón *min/max* entre las longitudes mínima y máxima, respectivamente, de la cuerda definida por dos puntos consecutivos. Considerando la longitud de la cuerda entre dos puntos, como una aproximación de la longitud de arco, está claro, que si la distribución de los puntos fuera exactamente uniforme, entonces *min/max* sería 1. En la tabla 1 se observan los valores de la razón *min/max*, para cada uno de los métodos empleados. Gráficamente, la distribución de los puntos se aprecia en las figuras 8 y 9.

Método	min/max
1	0.06
2	0.62

Tabla 1

De todo lo anterior, resulta evidente que si se utiliza la reparametrización que proponemos en esta sección, la distribución de los puntos sobre la cónica, es aproximadamente uniforme respecto a la longitud de arco, en la medida en que aumenta el número de puntos generados y por consiguiente, la longitud de arco se aproxima mejor por las secantes correspondientes.

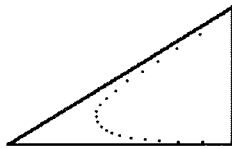


Fig. 8 Distribución de puntos para valores equidistantes del parámetro.

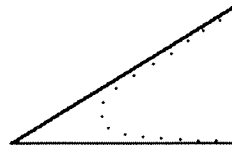


Fig.9 Distribución de puntos después de reparametrizar la cónica.

Capítulo 3

3.1 Parametrización global del spline cónico

Como vimos en la definición 2 del Capítulo 2, para describir un spline cónico necesitamos un parámetro global u , que indica dónde comienza y dónde termina cada una de las secciones que constituyen la curva. En nuestro caso, esto resulta además importante, pues la curva se debe subdividir en 4 segmentos, para generar un conjunto de puntos fijos sobre cada uno de ellos.

Sea u el parámetro global que describe al spline $s(u)$ formado por l secciones. Entonces el problema consiste en ver cómo escogemos $l + 1$ valores u_i , tales que:

$$s(u) = s_i(u) \quad \text{para } u_i \leq u \leq u_{i+1} \quad i = 0, \dots, l - 1$$

donde $s_i(u)$ es una sección cónica o una recta.

La forma más sencilla de proceder, sería tomar $u_i = i$ para todo i . Esta parametrización se conoce como equidistante o uniforme y en la práctica no da buenos resultados. Ello se debe a que la parametrización uniforme, ignora la geometría de la curva en el siguiente sentido. Si pensamos en el parámetro u , como el tiempo en que se recorre la curva, entonces $\Delta_i := u_{i+1} - u_i$ constante para todo i , significa que se utiliza el mismo tiempo en recorrer cada sección del spline, con independencia de la "forma" de esta.

Supongamos por ejemplo, que la i -ésima sección del spline es una cónica y la siguiente una recta, y que necesitamos generar 10 puntos sobre el segmento formado por estas dos secciones. Si tomamos 10 valores equidistantes de u en el intervalo $[u_i, u_{i+2}]$, entonces está claro que al ser $\Delta_i = \Delta_{i+1}$, los primeros 5 valores corresponderán a la cónica y los siguientes a la recta. Sin embargo, con 5 puntos no se describe tan bien una cónica como una recta.

Lo anterior indica que la parametrización más adecuada, será aquella en que la longitud Δ_i de cada intervalo $[u_i, u_{i+1}]$, dependa de la forma de la i -ésima sección del spline. En ese sentido, lo ideal es tomar u como la longitud de arco de la curva, ya que en tal caso Δ_i sería exactamente la longitud de arco de s_i . En la práctica, resulta muy difícil y costoso, calcular exactamente la longitud de arco de un segmento de cónica. Por eso se suele tomar:

$$\begin{aligned} u_0 &= 0 \\ u_{i+1} &= u_i + d_i \quad i = 0, \dots, l-1 \end{aligned} \quad (3.1)$$

donde d_i es una aproximación de la longitud de arco de la i -ésima sección del spline. Si s_i es un segmento de recta, d_i se puede calcular exactamente, mientras que si es una cónica, se puede evaluar ésta en un número finito de puntos y aproximar la longitud de arco, por la suma de las longitudes de los segmentos, definidos por cada pareja de puntos consecutivos.

Otro problema interesante en relación con la parametrización del spline, es el estudio de la clase de continuidad del mismo, la cual depende no sólo de la parametrización global, sino también de la parametrización local de cada sección. Como las rectas y cónicas que constituyen el spline, se unen con tangente continua, se sabe que es posible construir una parametrización, tal que la curva sea de clase C^1 [6]. A continuación vamos a probar que nuestra parametrización garantiza que esto se cumpla.

Teorema 5 Sea $s(u) = s_i(u)$ $u \in [u_i, u_{i+1}]$ $i = 0, \dots, l-1$ un spline cónico, donde cada cónica $s_i(u)$ está parametrizada como se indica en la sección 2.4. Entonces $s(u)$ es una curva de clase C^1 .

Demostración

Debemos probar que:

$$\|\dot{s}_{i-1}(u_i)\|^2 = \|\dot{s}_i(u_i)\|^2 \quad i = 1, \dots, l-1 \quad (3.2)$$

De acuerdo con la reparametrización de la curva que realizamos en la sección 2.4, tenemos que:

$$\dot{s}_i(u) = \frac{ds_i}{ds} \frac{ds}{dt} \frac{dt}{du} \quad (3.3)$$

donde

$$t = v(u) = \frac{u - u_i}{u_{i+1} - u_i}, \quad s = h(t)$$

y $h(t)$ es la función definida por (2.24). Esta función satisface las condiciones

$$\begin{aligned} h(0) &= 0 & h(1) &= 1 \\ \dot{h}(0) &= m_0^i & \dot{h}(1) &= m_1^i \end{aligned}$$

donde m_0^i y m_1^i están dadas como en (2.21) y (2.22) respectivamente, sustituyendo b_0 , b_1 , b_2 y w_1 por el polígono de control b_0^i , b_1^i , b_2^i y el peso intermedio w_1^i de la i -ésima cónica.

Por lo tanto, para $u = u_i$, $t = 0$ y $s = 0$ obtenemos de (3.3)

$$\dot{s}_i(u_i) = \left. \frac{ds_i}{ds} \right|_{s=0} \dot{h}(0) \dot{v}(u_i) = \frac{m_0^i}{\Delta_i} \left. \frac{ds_i}{ds} \right|_{s=0} \quad (3.4)$$

Ahora, según una propiedad de las cónicas que mencionamos en el Capítulo 1

$$\left. \frac{ds_i}{ds} \right|_{s=0} = 2w_1^i(b_1^i - b_0^i) \quad (3.5)$$

Luego, sustituyendo (3.5) en (3.4) llegamos a que

$$\|\dot{s}_i(u_i)\|^2 = 4(w_1^i)^2 \|b_1^i - b_0^i\|^2 \left(\frac{m_0^i}{\Delta_i} \right)^2 \quad (3.6)$$

Pero en la sección 2.4 escogimos m_0^i de modo que el lado derecho de (3.6) fuese 1 (vea (2.21)), para garantizar que la norma del vector tangente, en el extremo inicial, fuese 1. Por tanto,

$$\|\dot{s}_i(u_i)\|^2 = 1 \quad (3.7)$$

Procediendo de manera similar es fácil ver que

$$\|\dot{s}_{i-1}(u_i)\|^2 = 4(w_1^{i-1})^2 \|b_2^{i-1} - b_1^{i-1}\|^2 \left(\frac{m_1^{i-1}}{\Delta_{i-1}} \right)^2 \quad (3.8)$$

donde, de acuerdo con lo demostrado en la sección 2.4

$$m_1^{i-1} = \frac{\Delta_{i-1}}{2w_1^{i-1} \|b_1^{i-1} - b_2^{i-1}\|} \quad (3.9)$$

Luego, sustituyendo (3.9) en (3.8) obtenemos que

$$\|\dot{s}_{i-1}(u_i)\|^2 = 1 \quad (3.10)$$

Finalmente, de las ecuaciones (3.7) y (3.10) resulta que la condición (3.2) se cumple. Por lo tanto, las parametrizaciones local y global escogidas garantizan que el spline $s(u)$ es de clase C^1 .

□

Nótese que si $s_i(u)$ es una recta, entonces se puede escribir como cónica, tomando los puntos de su polígono de control colineales y asignándole a w_1^i un valor arbitrario. En particular, podemos considerar que w_1^i se escoge de modo que el lado derecho de (3.6) sea 1. Por lo tanto, el spline cónico es de clase C^1 , aún cuando alguno de sus segmentos sea una recta.



Fig. 10 Spline cónico cuyo parámetro global no tiene en cuenta la longitud de arco de cada segmento.

Para concluir esta sección mostramos dos gráficos que ilustran el efecto de la parametrización global, sobre la distribución de los puntos que pertenecen al spline. En el de la figura 10, se puede apreciar la distribución irregular de los puntos, que se obtiene al ajustar 200 datos (malla de 50×50) de la frontera del mapa de México, por medio de un spline cónico de clase C^1 , cuya parametrización global no tiene en cuenta la longitud de arco de cada segmento del spline.

La figura 11 corresponde a los resultados obtenidos, cuando se ajustan los mismos datos, por medio un spline cónico parametrizado globalmente, según (3.1), es decir teniendo en cuenta la longitud de arco de cada segmento.

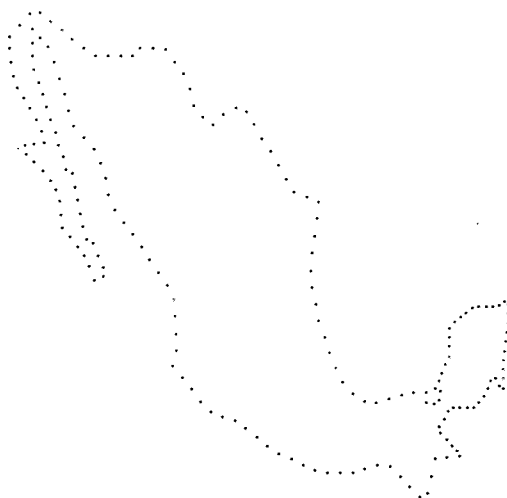


Fig. 11 Spline cónico cuyo parámetro global aproxima a la longitud de arco.

3.2 Algoritmo para construir el Spline Cónico de Ajuste.

En esta sección vamos a describir de manera general, el algoritmo que hemos diseñado para ajustar un conjunto de puntos del plano, por medio de un spline cónico. Los detalles relacionados con cada paso del mismo, han sido tratados a lo largo del trabajo, por lo que en algunos pasos se remitirá al lector a la sección correspondiente.

ALGORITMO

Sea $I_i = (x_i, y_i)$ $i = 0, \dots, m - 1$ el conjunto de puntos del plano que se desea ajustar. Los pasos a seguir para construir el spline cónico de ajuste $s(u)$ son los siguientes:

1. Construir el polígono que interpola los datos y eliminar los puntos que se consideren colineales. Específicamente, calcular el ángulo θ_i que se forma en cada vértice I_i $i=0, \dots, m-1$ y eliminar el punto I_i si $\cos\theta_i < -0.99$. Actualizar el número m de vértices
2. Seleccionar un valor para la distancia ε , $0 < \varepsilon < 1$ entre el polígono y el spline de ajuste (vea la sección 2.2).
3. Definir la estructura del spline. Calcular el polígono de control b_0^i, b_1^i, b_2^i y el peso w_1^i $i=0, \dots, m-1$ de cada una de las cónicas que constituyen el spline.
 - 3.1 Calcular la longitud l_i $i=0, \dots, m-1$ de cada lado del polígono que interpola los datos. Sea l_{min} la longitud del menor lado y l_{max} la del mayor.
 - 3.2 Si $l_{min} \neq l_{max}$ ir al paso 3.3. De lo contrario, calcular b_2^i y b_0^{i+1} $i=0, \dots, m-1$ (con $b_0^0 = b_0^m$) a partir de las ecuaciones:

$$\|b_2^i - I_i\| = l_{min}/3 \quad \|b_0^{i+1} - I_{i+1}\| = l_{min}/3$$
 Ir al paso 3.4
 - 3.3 Para $i=0, \dots, m-1$
 Si $l_i > 2l_{min}$ calcular b_2^i y b_0^{i+1} $i=0, \dots, m-1$ (con $b_0^0 = b_0^m$) a partir de las ecuaciones:

$$\|b_2^i - I_i\| = l_{min}/2 \quad \|b_0^{i+1} - I_{i+1}\| = l_{min}/2$$
 De lo contrario, $b_2^i = b_0^{i+1}$ y ambos resultan de la relación:

$$\|I_i - b_2^i\| = \|I_{i+1} - b_0^{i+1}\| = l_i/2$$
 - 3.4 Para $i=0, \dots, m-1$, tomar $b_1^i = I_i$
 - 3.5 Para $i=0, \dots, m-1$, calcular el peso intermedio de cada cónica (vea la sección 2.2) $w_1^i = (1 - \varepsilon)/\varepsilon$

4. Calcular aproximadamente la longitud de arco d_j $j=0, \dots, l-1$ de cada una de las secciones cónicas y rectas que constituyen el spline (el número l de secciones del spline se obtiene del paso anterior).
5. Parametrizar globalmente el spline, o sea calcular los nodos u_j $j=0, \dots, l$ del mismo según la fórmula (vea la sección 3.1)

$$u_0 = 0$$

$$u_{j+1} = u_j + d_j \quad j = 1, \dots, l - 1$$
6. Generar un número prefijado de puntos sobre el spline cónico.
 - 6.1 Leer las coordenadas de los 4 puntos R_1, R_2, R_3 y R_4 que subdividirán la frontera de la región ajustada por medio del spline.
 - 6.2 Identificar a qué sección del spline pertenece cada punto R_i $i = 1, \dots, 4$.
 - 6.3 Resolver el problema de inversión para cada punto R_i $i=1, \dots, 4$ (vea la sección 2.3) o sea obtener los valores u_{R_i} del parámetro u tales que:

$$s(u_{R_i}) = R_i \quad i = 1, \dots, 4$$
 - 6.4 Leer el número de líneas "horizontales" y "verticales" que tendrá la malla a construir. En otras palabras definir la cantidad de puntos que se desea en cada una de las 4 subfronteras (sección 2.3).
 - 6.5 Sea n_i la cantidad de puntos que se necesita en la i -ésima subfrontera. Entonces para $i=1, \dots, 4$ repetir los siguientes pasos:

- 6.5.1 Generar n_i puntos equidistantes uu_k $k = 0, \dots, n_i - 1$ en el intervalo $[u_{R_i}, u_{R(i+1)}]$
 - 6.5.2 Si $uu_k \in [u_j, u_{j+1}]$ $k = 0, \dots, n_i - 1$, $j = 0, \dots, l - 1$, calcular $t_k = (uu_k - u_j)/(u_{j+1} - u_j)$
 - 6.5.3 Si la j -ésima sección del spline es una recta, evaluarla en t_k .
 - 6.5.4 Si la j -ésima sección del spline es una cónica, evaluar en t_k $k = 0, \dots, n_i - 1$ la función $h(t)$ dada por (2.24) en la sección (2.4), o sea calcular $s_k = h(t_k)$
Evaluar entonces la cónica en s_k .
7. Graficar los puntos obtenidos sobre la frontera ajustada por medio del spline cónico.

FIN del algoritmo

Algunos comentarios finales acerca del algoritmo. Es fácil verificar que si se aplica una transformación afín a los datos a ajustar, entonces el punto inicial y final de cada cónica es la imagen, por la propia transformación, del que corresponde según el algoritmo previo. Por otro lado, la selección del peso intermedio de cada cónica sólo depende de el parámetro ε escogido. Por lo tanto, teniendo en cuenta además que las cónicas son invariantes por transformaciones afines (sección 1.3), resulta que el algoritmo propuesto también lo es. Esto significa que los siguientes procedimientos son equivalentes. Primero aplicar una transformación afín a los datos y después construir el spline cónico de ajuste, o primero ajustar los datos y después aplicar la transformación afín al spline. Claro que la primera forma de proceder es más económica, puesto que la cantidad de datos es siempre mucho menor que el número de puntos que se necesita para graficar el spline cónico.

Por otra parte, respecto a otros algoritmos para ajustar datos mediante cónicas [15], nuestro algoritmo tiene la ventaja de que es interactivo, pues es posible escoger la distancia entre el polígono que forman los datos y el spline. Más precisamente, se puede construir un spline tan "cercano" a los datos como se desee. Esto depende únicamente de la selección del parámetro ε , de modo que cuando $\varepsilon \rightarrow 0$ el spline de ajuste tiende al polígono original (cuando $\varepsilon \rightarrow 1$

el spline cónico se aproxima al polígono que se forma "truncando" las esquinas del original). Sin embargo, nuestro algoritmo presupone que los datos han sido previamente ajustados por un polígono [16], de modo que cada vértice del mismo se "suavizará" por medio de una cónica.

3.3 Programa y ejemplos de prueba

Para probar el algoritmo propuesto en este trabajo, se elaboró un programa en Lenguaje *C*, que a partir de los datos originales, construye el spline cónico que aproxima la frontera de la región. El programa grafica el polígono que forman los datos y el spline cónico de ajuste, de modo que se pueden apreciar gráficamente los resultados del ajuste. Además, al final de cada corrida, se ofrece un gráfico con los puntos sobre el spline, que se mantendrán fijos como extremos de la malla a construir.

Entre otras facilidades, el programa elaborado ofrece al usuario, la alternativa de escoger los cuatro puntos que subdividen la curva, o de que estos se calculen de forma automática, de acuerdo con las dimensiones de la malla a construir. Además, se ofrece una opción default ($\epsilon = 0.5$), para la distancia entre el polígono que forman los datos y el spline cónico de ajuste.

Los resultados del programa se validaron construyendo mallas, sobre las regiones cuya frontera es el spline cónico de ajuste. Para ello se utilizó el sistema UNAMALLA [1]. En todos los ejemplos probados, se pudo comprobar que las mallas construidas sobre regiones cuya frontera fue ajustada por medio de un spline cónico, tienen mejores propiedades que las construidas sobre la misma región, cuando la frontera se forma mediante interpolación lineal de los datos, o incluso cuando se ajusta por medio de un spline cúbico.

La figura 12 muestra algunos polígonos sencillos, ajustados por medio del spline cónico que se propone en este trabajo. Como se puede

apreciar en la misma, las rectas y secciones cónicas que constituyen la curva, se unen con suavidad, de modo que geoméricamente se hace evidente, que la curva tiene tangente continua en todos los puntos.

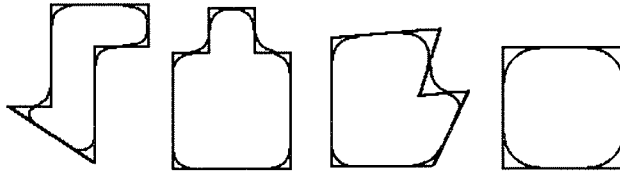


Fig.12 Algunos polígonos suavizados.

En la tabla 2 se ofrece una comparación de los resultados obtenidos, para tres juegos de datos, cuando se consideran las siguientes opciones:

Opción 1: Las cónicas que constituyen el spline se reparametrizan de siguiendo el procedimiento que se explica en la sección 2.4.

Opción 2: Las cónicas no se reparametrizan y se evalúan directamente en la representación de Bezier.

El primer juego de datos corresponde a los vértices de un cuadrado de lado 1. El segundo, son 77 puntos que describen la frontera de la bahía de la Habana, considerada como una curva cerrada. El tercer fichero está formado por los siguientes puntos: (2,0),(6,0),(8,4),(4,8) y (0,4), que representan un polígono de 5 lados.

La comparación se basa en el cálculo de la razón min/max entre la mínima y la máxima distancia entre dos puntos consecutivos, situados sobre el spline cónico. Aproximando la longitud de arco entre dos puntos, por la longitud de la cuerda, está claro que la distribución de los puntos sería exactamente uniforme, si $min/max = 1$.

Como se puede apreciar en la tabla 2, la reparametrización introducida mejora notablemente la uniformidad de la distribución de los puntos. Esto es muy importante, pues constituye el primer paso para generar puntos con una distribución prefijada, lo cual es vital para controlar las propiedades que tendrá la malla.

Datos	Extremos de las subfron.	ε	Orden de la Malla	Opción 1 min/max	Opción 2 min/max
Cuadrado de lado 1	(0.5,1) (0,0.5) (0.5,0) (1,0.5)	0.4	10x10	0.911	0.560
Bahía Habana	(0.208,0.219) (0.640,0.130) (0.936,0.689) (0.329,0.897)	0.2	40x40	0.443	0.199
Polígono de 5 lados	(4,0) (0.55,3.32) (4,7.2) (7.44,3.32)	0.4	20x20	0.747	0.393

tabla 2

3.4 Conclusiones y Recomendaciones

En la Tesis se presenta un método para ajustar un conjunto de puntos del plano por medio de un spline cónico, es decir mediante una curva formada por secciones rectas y cónicas, que se unen de modo que el spline es de clase C^1 . El método propuesto, supone que los datos del problema han sido previamente ajustados por un polígono, de modo que el spline a construir "suavizará" todos los vértices del mismo. El algoritmo diseñado es interactivo, en el sentido de que permite escoger cuál es la máxima distancia que se desea, entre los datos y la curva de ajuste. Además, es invariante por transformaciones afines, lo cual tiene gran importancia desde el punto de vista práctico.

Una vez que se ha construido el spline cónico que ajusta los datos del problema y que aproxima la frontera de la región plana, se seleccionan cuatro puntos que subdividen la misma. Estos puntos definirán los segmentos de la frontera que se deben unir, mediante las líneas "horizontales" y "verticales" de la malla. Los puntos extremos de esas líneas se mantienen fijos en el proceso de construcción de la malla óptima, por lo que resulta muy importante controlar la posición de los mismos.

Por todo lo anterior, durante el desarrollo del trabajo fue necesario resolver una serie de problemas adicionales, con vistas a generar puntos pertenecientes al spline cónico y cuya distribución no fuese arbitraria. En ese sentido, se presenta en la Tesis una reparametrización de las secciones cónicas, muy económica y elegante, en tanto se basa en el uso de un spline cónico de dos secciones. Esta reparametrización aproxima a la parametrización en términos de la longitud de arco y garantiza que se puedan obtener puntos sobre la curva, con distribución aproximadamente uniforme.

Además para controlar la calidad de la aproximación construida, en el trabajo se introdujo una medida de la "distancia" entre el polígono definido por los datos y la curva de ajuste. Este concepto, nos permitió deducir una fórmula de inversión para las secciones cónicas, que

asigna a un punto que no pertenece a la cónica, el valor del pámetro que corresponde a otro "cercano" y que sí se halla sobre la misma.

En el futuro próximo, el software elaborado siguiendo el algoritmo propuesto, debe ser incorporado al sistema UNAMALLA [1], con vistas a mejorar la calidad de la aproximación de la frontera, que se utiliza actualmente en este sistema. Por otro lado, sería conveniente incluir en el programa diseñado un procedimiento como el que se presenta en [16], para ajustar un polígono a los datos originales. Esto garantizaría, que solamente se ajustaran los vértices "significativos", lo cual reduce el costo computacional.

También es recomendable, que el algoritmo incluya la posibilidad de que el usuario escoja, en algunos casos, los vértices a ajustar y la proximidad de las cónicas a cada uno de ellos. De esta forma, el método de ajuste propuesto, se puede utilizar con éxito en el diseño libre de figuras geométricas planas.

Bibliografía

- [1] Barrera, P., Castellanos, L., Pérez, A. UNAMALLA: Un sistema para la generación de mallas en regiones planas irregulares. Reporte Técnico, Facultad de Ciencias, UNAM, 1994.
- [2] de Boor, C. A Practical Guide to Splines, Springer-Verlag, 1978.
- [3] Estrada, J., Hernández, V. Generation of points on a curve with prescribed distribution of their arc length. (aceptado para publicar en la revista Investigación de Operaciones), 1995.
- [4] Farin, G. Algorithms for rational Bezier curves. Computer- Aided Design, Vol. 15, No. 2, feb 1983, pp 73-77.
- [5] Farin, G. From conics to NURBS: A Tutorial and Survey. IEEE Computer Graphics and Applications, sept 1992, pp 78-86.
- [6] Farin, G. Curves and Surfaces for Computer-Aided Geometric Design. Academic Press, 1992.
- [7] Farouki, R., Sakkalis, T. Real Rational Curves Are Not Unit Speed, Computer-Aided Geometric Design, Vol. 8, No. 2, Mar. 1991, pp 151-158.
- [8] Fulton, W. Algebraic Curves. An Introduction to Algebraic Geometry, W. A. Benjamin Inc., New York, 1969.
- [9] Goldman, R.N., Sederberg, T.W., Anderson, D.C. Vector Elimination: A Tecnique for the implicitation, inversion, and intersection of planar parametric rational polynomial curves. Computer Aided Geometric Design 1, 327-356, 1984.

- [10] Initial Graphics Exchange Specification Doc. No. NB-SIR 86-3359, NIST, Gaithersburg, Md., 1986.
- [11] Knuts, D.E. T X and METAFONT, New Directions in Typesetting. Digital Press and American Mathematical Society, Bedford, Mass. 1979.
- [12] Lee, E. The rational Bezier representation for conics. Geometric Modeling: Algorithms and New Trends, G. Farin, ed, SIAM, Philadelphia, 1987, pp 3-19.
- [13] Liao, Y.Z. A two-stage method of fitting conic arcs and straight-line segments to digitized contours. Proc. IEEE Pattern Recognition and Image Processing Conference, Dallas, Texas, 1981, pp. 224-229.
- [14] Liming, R. Practical Analytical Geometry with Applications to Aircraft, Macmillan, New York, 1944.
- [15] Pavlidis, T. Curve Fitting with Conic Splines, ACM Trans. on Graphics, 2, 1, 1-31, Jan 1983.
- [16] Pavlidis, T. Curve Fitting as a Pattern Recognition Problem. Proc. 6th Int. Conf. Pattern Recognition (Munich, Oct 1982), IEEE Computer Society Press, Silver Spring, Md., pp 853-859, 1982.
- [17] Penna, M.A. and Patterson, R.R. Projective Geometry and its Applications to Computer Graphics. Prentice Hall, 1986.
- [18] Piegler, L. Defining curves containing conic segments. Computer and Graphics, 8, 1984, pp. 177-182.
- [19] Piegler, L. A Technique for Smoothing Scattered Data with Conic Sections. Computers in Industry, Vol. 9, No. 3, Nov. 1987, pp. 223-237.
- [20] Piegler, L. On NURBS: A Survey. IEEE Computer Graphics and Applications, Vol. 11, No. 1, Jan 1991, pp 55-71.

- [21] Pratt, V. Techniques for Conic Splines. Proceedings of SIGGRAPH 1985, ACM, New York, 1985, pp. 151-159.
- [22] Salmon, G. Modern Higher Algebra, Hodges, Smith, and Co., Dublin, 1866.
- [23] Sederberg, T. W. Algebraic Geometry for Computer-Aided Geometric Design, IEEE Computer Graphics and Applications, pp 52-59, 1986.
- [24] Van Dam, A. PHIGS+ Functional Description Revision 3.0. Computer Graphics, Vol. 22., No. 3, July 1988, pp. 125-218.
- [25] Van Der Waerden, B.L. Modern Algebra, Volumes I,II, Frederick Ungar Publishing Co., New York, 1950.
- [26] , Walker R.J. Algebraic Curves, Springer-Verlag, New York, 1950.
- [27] Wang W., Joe B. The Geometric Interpretation of Inversion Formulae for Rational Plane Curves. Computer Aided Geometric Design 12, 469-489, 1995.