

MINISTERIO DE CIENCIA, TECNOLOGIA Y  
MEDIO AMBIENTE

Instituto de Cibernética, Matemática y Física

# Modelación Geométrica con Splines Algebraicos

TESIS

*presentada en opción al grado de*  
Doctor en Ciencias Matemáticas

**Autor:** M.C. Victoria Hernández Mederos

**Asesores:** Dr. Pablo Barrera Sánchez, UNAM, México.  
Dr. Jorge Estrada Sarlabous, ICIMAF, Cuba.

Para Pablo, con todo  
el cariño y agradecimiento  
de una amiga,  
Vicky

La Habana, julio 2001.

*A Gabriela y Jorgito  
con todo mi amor*

## Agradecimientos

*A mi familia que me ha brindado aliento para seguir adelante en todo momento. En especial, a mi esposo Jorge Estrada, sin cuya ayuda y entusiasmo este trabajo no existiera, y quien me ha brindado su conocimiento, su cariño, amor y apoyo incondicional en toda la vida.*

*A Pablo Barrera, con quien he compartido muchos años de amistad y de mi vida profesional y a quien le debo buena parte de mi formación. A todos mis profesores de la Universidad de La Habana, a quienes recuerdo con cariño y admiración.*

*A mis amigos y mis colegas del ICIMAF, quienes me han brindado ayuda y ánimo en los momentos difíciles. En particular, a Valia Guerra mi amiga y compañera de siempre y a Lilliam Alvarez quien me inició en el camino de la investigación y de cuya amistad he disfrutado todos estos años.*

*A Richard Patterson, por su valiosa ayuda, su precioso tiempo y su cordialidad. A Marco Paluszny, quien nos invitó a trabajar en este tema. A Leonardo Traversoni, por su amistad y por su apoyo, gracias a los cuales pude conocer a los autores de los libros "sagrados" sobre splines. A Kirill Kopotun, por su respaldo generoso que me permitió participar y exponer mi trabajo en el Congreso donde los matemáticos celebraron los 60 años del Profesor L.L. Shumaker. A Daniel Filip, por su ayuda entusiasta y por todo lo que nos enseñó sobre curvas y superficies paramétricas. A Gudrun Albrech por su revisión, comentarios y sugerencias sobre cada problema que se trata en la Tesis. A Colm Mulcahy por su apoyo generoso y su amistad.*

*A todos los demás que me han ofrecido su aliento, cuando se necesita un amigo para seguir adelante...Meilan y Jorge, Sofía, Dimas, Pico, Aurorita, Minerva, Adalberto, Janko, Jesús, Lydia, Dionne, Eduardo, Bernardo, Arocha, Guilmer, Mónica ... A todos los que algún día creyeron en mi ...*

*Este trabajo fue posible gracias al apoyo financiero de varias instituciones: el ICIMAF, el CONACyT de México y la Academia de Ciencias del Tercer Mundo con el fondo para la investigación 98-195 RG/MATHS/LA y su programa Sur-Sur.*

# Contenido

<b>1</b>	<b>Introducción</b>	<b>1</b>
<b>2</b>	<b>Definiciones y resultados preliminares</b>	<b>9</b>
2.1	Curvas algebraicas planas . . . . .	9
2.1.1	Definiciones . . . . .	9
2.1.2	Continuidad geométrica . . . . .	12
2.1.3	Coordenadas baricéntricas . . . . .	15
2.1.4	Expresión en coordenadas baricéntricas . . . . .	16
2.1.5	Representación de Bézier en coordenadas baricéntricas . . . . .	18
2.1.6	Familia de curvas algebraicas definidas en el interior de un triángulo . . . . .	18
2.2	Superficies algebraicas . . . . .	20
2.2.1	Definiciones . . . . .	20
2.2.2	Continuidad geométrica . . . . .	23
<b>3</b>	<b>Ajuste con un A-spline cónico</b>	<b>25</b>
3.1	Introducción . . . . .	25
3.2	Preliminares . . . . .	26
3.3	Cálculo de los nodos del A-spline . . . . .	28
3.4	Definición de la tangente en cada nodo . . . . .	32
3.5	Ajuste de datos por medio de una cónica . . . . .	35
3.5.1	El problema general . . . . .	35
3.5.2	Ajuste de una cónica en el interior de cada triángulo . . . . .	37
3.6	Experimentación numérica . . . . .	39
<b>4</b>	<b>Diseño libre con curvas A-spline cúbicas</b>	<b>41</b>
4.1	Introducción . . . . .	41
4.2	A-spline cúbico default . . . . .	44

4.2.1	Consideraciones preliminares . . . . .	44
4.2.2	Una familia de cúbicas convexas, conexas y no singulares	45
4.2.3	Cálculo de una base para la $i$ -ésima sección . . . . .	47
4.2.4	$G^2$ continuidad del A-spline . . . . .	51
4.3	Construcción de un A-spline cúbico no default . . . . .	54
4.3.1	Máximas deformaciones . . . . .	54
4.3.2	Deformaciones graduales . . . . .	61
4.3.3	Cambio de posición del punto de interpolación. . . . .	62
4.4	Graficación de la curva A-spline . . . . .	63
4.5	Ejemplos . . . . .	64
<b>5</b>	<b>Interpolación con Superficies A-splines</b>	<b>67</b>
5.1	Introducción . . . . .	67
5.2	Preliminares . . . . .	71
5.3	Planos de la frontera . . . . .	72
5.4	Nuevo sistema de coordenadas . . . . .	73
5.5	Direcciones tangentes . . . . .	73
5.6	curvas de la frontera . . . . .	75
5.6.1	Curvas de la frontera para la configuración convexa . .	75
5.6.2	Curvas de la frontera para la configuración no convexa	76
5.7	Bloques algebraicos en coordenadas canónicas . . . . .	79
5.7.1	Bloques algebraicos en el caso convexo . . . . .	79
5.7.2	Bloques algebraicos en el caso no convexo . . . . .	81
5.8	Transformación de coordenadas . . . . .	83
5.9	Selección de los vértices de los conos . . . . .	86
5.10	Funciones de peso . . . . .	89
5.11	A-parches . . . . .	90
5.12	Graficación de la superficie A-spline . . . . .	92
5.13	Ejemplos y Experimentación numérica . . . . .	97
5.13.1	Experimentación numérica . . . . .	98
5.13.2	Ejemplos . . . . .	100
<b>6</b>	<b>Conclusiones y Recomendaciones</b>	<b>105</b>
<b>A</b>	<b>Parches triangulares de Bezier</b>	<b>109</b>
<b>B</b>	<b>Demostraciones</b>	<b>111</b>

# Capítulo 1

## Introducción

El Diseño Geométrico asistido por Computadoras, conocido en la literatura técnica por sus siglas en inglés CAGD (Computer Aided Geometric Design), es una disciplina relativamente nueva, cuyo nacimiento, como fruto de la unión de la Geometría y la Computación, puede situarse a inicios de la década del 60, con los trabajos de P. Bézier y P. de Casteljaou, ambos asociados con la industria automotriz. La tarea principal de esta disciplina es la automatización del proceso de diseño de objetos, para lo cual ha sido necesario desarrollar una gran variedad de algoritmos que permiten el uso eficiente de las curvas y superficies.

Todavía hoy en día, las curvas y superficies que más comúnmente se utilizan para resolver problemas de CAGD o modelación geométrica son las parametrizadas racionalmente. Toda curva racional paramétrica se puede representar además mediante una ecuación implícita del tipo  $f(x, y) = 0$ , donde  $f(x, y)$  es un polinomio en dos indeterminadas, es decir como una curva algebraica. Similarmente, toda superficie racional paramétrica se puede representar como una superficie algebraica, es decir mediante una ecuación implícita del tipo  $f(x, y, z) = 0$ , donde  $f(x, y, z)$  es un polinomio. Sin embargo, el recíproco de ambas afirmaciones no es cierto, por lo cual la clase de las curvas y superficies algebraicas definidas implícitamente es estrictamente mayor que la clase de las curvas y superficies parametrizadas racionalmente. En lo adelante nos referiremos a las primeras simplemente como curvas y superficies implícitas y a las segundas como paramétricas.

La gran popularidad de las curvas y superficies paramétricas y entre ellas de las curvas y superficies splines, se puede explicar debido a dos razones fundamentales. En primer lugar, las mismas se pueden representar

en términos de un polígono o malla de control [Far93], el cual nos permite controlar fácilmente su geometría. Por otro lado, son fáciles de graficar, pues para obtener un conjunto de puntos sobre ellas basta con evaluar en una cantidad finita de valores del parámetro. En la práctica, este proceso no siempre está exento de dificultades [Roy92], ya que las curvas y superficies racionales pueden poseer polos (donde se anula el denominador de la expresión racional), puntos base (donde se anulan numerador y denominador de la expresión racional) o incluso puntos con coordenadas reales que corresponden a valores complejos del parámetro.

El uso de curvas y superficies algebraicas definidas implícitamente en la solución de problemas de CAGD es mucho más reciente. Sederberg propuso por primera vez en [Sed85] la posibilidad de utilizar superficies splines, formadas por parches definidos de forma implícita, para resolver problemas de modelación geométrica. Más tarde C. Bajaj bautizó tales curvas y superficies con el nombre de *A-splines* o *splines algebraicos*. A partir de entonces comenzaron a aparecer una gran cantidad de trabajos [Baj92a)], [Baj92b)], [Baj99a)], [Baj94], [Baj95], [Baj99a)], [Baj99b)], [Pal93], [Pal94], [Pal98], [Pal99] que utilizan curvas y superficies A-splines para diseño libre, interpolación de datos, ajuste de contornos, etc. Todos estos trabajos se basan en las ventajas que tiene la representación implícita en comparación con la paramétrica. Entre ellas pueden citarse:

- Para un grado  $n$  fijo, las curvas y superficies algebraicas definidas implícitamente tienen mayor cantidad de parámetros libres que las paramétricas del mismo grado. Más precisamente [Baj99a)], las curvas implícitas tienen  $\frac{(n+1)(n+2)}{2} - 1$  grados de libertad, mientras que las paramétricas poseen sólo  $O(n)$ ; las superficies implícitas [Baj92c)] tienen  $\frac{(n+1)(n+2)(n+3)}{6} - 1$  parámetros libres y las paramétricas  $O(n^2)$ . En la práctica, los parámetros libres adicionales que se obtienen al emplear la representación implícita se pueden utilizar para imponer condiciones extra de interpolación, para lograr un mejor ajuste de los datos o para garantizar una mayor suavidad de la curva o superficie A-spline.
- Las curvas y superficies implícitas forman una familia cerrada para las operaciones más usuales que usan los sistemas de diseño geométrico, tales como el cálculo de la intersección o unión de dos curvas o superficies, la convolución, el cálculo del "offset" (los puntos que se encuentran a una distancia euclideana prefijada), etc. La subclase de las curvas y



superficies racionales paramétricas no es cerrada con respecto a ninguna de estas operaciones. Por ejemplo, el "offset" de una curva racional paramétrica puede no ser una curva racional paramétrica, pero siempre es una curva algebraica y en consecuencia tiene una representación implícita.

- Utilizando la representación implícita es muy fácil saber si un punto dado se encuentra sobre una curva o superficie y en caso de no estarlo en qué semiplano o semiespacio se encuentra. La solución del mismo problema a partir de la representación paramétrica es mucho más costosa y requiere de una fórmula de inversión. Por otra parte, el cálculo de la curva intersección entre dos superficies definidas de forma paramétrica, resulta mucho más sencillo si una de ellas se reescribe mediante su ecuación implícita. Este proceso se conoce como implicitación de la curva. Para más detalles sobre los procesos de implicitación e inversión se puede consultar [Sed84].

Entonces cabe preguntarse, si las curvas y superficies implícitas tienen todas estas ventajas ¿por qué no habían sido empleadas con anterioridad en la solución de problemas de modelación geométrica? La respuesta a esta pregunta es bien conocida. En primer lugar, debe señalarse que hasta hace relativamente poco tiempo no se disponía de algoritmos eficientes para graficar las curvas y superficies implícitas. Inicialmente, por ejemplo, para graficar curvas algebraicas cúbicas se propusieron algoritmos que introducían parametrizaciones no racionales [Pat88] y que en consecuencia eran costosos. Más adelante aparecieron otros métodos, que calculaban los puntos sobre una curva o superficie algebraica definida implícitamente por la ecuación  $f = 0$ , como las soluciones de tal ecuación polinomial. Tales algoritmos fallaban con frecuencia ante la presencia de puntos singulares, o ramas de las curvas muy próximas entre sí [Chan88]. Pero en 1994 [Tau94b)] apareció ya un algoritmo efectivo y robusto para graficar curvas y superficies implícitas, basado en un esquema de subdivisión recursiva del tipo árbol cuaternario. Otra de las desventajas de las curvas y superficies implícitas que limitaron su uso por mucho tiempo es el hecho de que estas pueden tener singularidades y más de una rama u hoja en la región de interés. En ese sentido, es importante señalar que en los últimos años han aparecido trabajos [Baj99a)], [Pal92],[Pal93] donde se demuestra que bajo ciertas condiciones, fácilmente verificables, es posible garantizar que las curvas o superficies implícitas no

tengan singularidades en la región de interés y sólo posean una rama u hoja respectivamente.

El objetivo central de este trabajo es mostrar que varios problemas de modelación geométrica pueden ser resueltos de manera exitosa utilizando curvas y superficies A-splines definidas implícitamente. Una de las motivaciones del trabajo fue el problema de reconstruir una superficie a partir de imágenes que corresponden a secciones transversales paralelas de la misma. Tal problema aparece con frecuencia en la medicina, cuando se trata de representar algún órgano a partir de imágenes digitales de sus secciones transversales, obtenidas mediante ultrasonido. Otro problema que nos sirvió de motivación fue la necesidad de definir curvas "suaves" que pudieran utilizarse para diseñar la directriz de una superficie de revolución, con el objetivo de representar una estructura ingenieril de tipo cáscara. Los problemas prácticos anteriormente mencionados nos llevaron a plantearnos los siguientes objetivos específicos:

1. Aproximar mediante una curva A-spline  $G^1$  continua y plana, un conjunto de puntos en el plano que representan el contorno de una imagen digital.
2. Interpolar un conjunto de puntos en el plano y las correspondientes direcciones tangentes por medio de una curva A-spline cúbica plana. Lograr que la curva A-spline sea  $G^2$  continua sin presuponer el conocimiento del valor de la curvatura en los puntos de interpolación.
3. Interpolar un conjunto de puntos en el espacio, vértices de una triangulación, por medio de una superficie A-spline  $G^1$  continua.

En relación con los objetivos planteados, los principales logros de este trabajo son los siguientes:

- Se propone un algoritmo afectivo para aproximar el contorno de una imagen digital por medio de una curva A-spline cónica  $G^1$  continua y plana. Este algoritmo se basa en diferentes modificaciones de métodos conocidos para resolver las distintas etapas del problema. Entre ellas vale la pena destacar las siguientes. Se encontró un método sencillo para determinar a priori los puntos de inflexión que tendrá la curva A-spline, partiendo exclusivamente de los datos que ésta aproximará. También se introdujeron pesos apropiados en el problema de ajuste

para determinar la recta que aproximará la tangente en cada nodo del A-spline. Finalmente, se estudió el problema de seleccionar una aproximación de la distancia euclideana de un punto a una cónica, que sea a la vez poco costosa de calcular y lo suficientemente precisa como para garantizar la calidad del ajuste.

- Se obtuvo una nueva familia de curvas A-spline cúbicas  $G^2$  continuas y planas muy útiles para el diseño libre. Las curvas de esta familia interpolan un conjunto de puntos y direcciones tangentes prefijadas. Además poseen una serie de parámetros libres que se pueden usar como asas geométricas para controlar con facilidad la forma de la curva. El método de interpolación asociado a la familia de curvas propuestas tiene carácter local, lo que permite introducir modificaciones en un segmento del A-spline sin que estas se “propaguen” a toda la curva. Por último, el método propuesto nos permite obtener una curva A-spline  $G^2$  continua sin exigir que el diseñador proporcione valores específicos de la curvatura en los puntos de interpolación, lo cual resulta poco natural para muchos problemas “prácticos”. A diferencia de otros esquemas de interpolación que utilizan tanto splines paramétricos como algebraicos, el método propuesto no requiere la solución de sistemas de ecuaciones para determinar los parámetros libres que garantizan la  $G^2$  continuidad de la curva A-spline.
- Se propone una familia nueva de superficies A-splines  $G^1$  continuas capaz de interpolar un conjunto de puntos en el espacio y las correspondientes direcciones normales prefijadas en esos puntos. A diferencia de otros algoritmos similares reportados en la literatura [Baj95], el método de interpolación propuesto no requiere construir previamente una envoltura de tetraedros, lo cual es una ventaja indiscutible, pues tal proceso es engorroso y costoso desde el punto de vista computacional. Además, como cada parche del A-spline se define prefijando previamente sus curvas de la frontera, el esquema de interpolación propuesto tiene un carácter local (si se modifica un punto de interpolación, sólo es necesario recalcular aquellos parches que interpolan este punto) y la condición de  $G^1$  continuidad de la superficie A-spline se logra sin tener que resolver un sistema de ecuaciones lineales.

La estructura de la Tesis es la siguiente. En el capítulo 2 se introducen los conceptos y definiciones básicas que se utilizarán a lo largo de todo el

trabajo: el concepto de curva y superficie algebraica, el de curva y superficie A-spline, el concepto de continuidad geométrica, etc. Además se obtiene la ecuación implícita de una curva algebraica en coordenadas baricéntricas y su representación de Bézier en estas coordenadas. Esta representación se utilizará más adelante, para escribir una curva A-spline cónica o cúbica. También en este capítulo se presenta un resultado muy importante de C. Bajaj [Baj99a]), que nos asegura que bajo ciertas condiciones en los coeficientes de la representación de Bézier de una curva algebraica, existe una familia de tales curvas con “buenas” propiedades en el interior del triángulo donde está definida.

Los capítulos 3,4 y 5 se dedican cada uno a los problemas que enunciamos como objetivos específicos de la Tesis. En el capítulo 3 se presenta un algoritmo para ajustar un conjunto de puntos que pertenecen al contorno de una imagen digital, por medio de una curva A-spline cónica  $G^1$  continua y plana. En el capítulo 4 se propone una familia de curvas A-spline cúbicas planas y  $G^2$  continuas, que interpola un conjunto de puntos en el plano y las correspondientes direcciones tangentes. Esta familia nos provee de una herramienta muy útil para el diseño libre de curvas. Por último, en el capítulo 5 se construye una familia de superficies A-spline  $G^1$  continuas capaces de interpolar un conjunto prefijado de puntos en el espacio.

En resumen, en la Tesis se proponen nuevas familias de curvas y superficies A-splines que permiten resolver diferentes problemas de modelación geométrica, de forma cómoda y eficiente, sin salirse del entorno de la representación implícita.

Los resultados obtenidos en este trabajo han sido divulgados en diferentes artículos y eventos científicos:

#### Artículos:

- V. Hernández, J. Estrada, P. Barrera, On the Euclidean distance from a point to a conic, *Revista Integración* (Universidad Industrial de Santander), Vol 15, No. 1, pág 45–61, 1997.
- V. Hernández, J. Estrada, P. Barrera, A new algorithm to compute the Euclidean distance from a point to a conic, aceptado para publicar, *Revista Investigación Operacional*.

- V. Hernández, S. Behar, J. Estrada, Geometric design by means of a  $G^2$  continuous A-spline, Approximation, Optimization and Mathematical economics, Physica-Verlag, Heidelberg, 2001, 133–145.
- V. Guerra, V. Hernández, Numerical aspects in locating the corner of the L-curve, Approximation, Optimization and Mathematical economics, Physica-Verlag, Heidelberg, 2001, 121–131.
- V. Hernández, D. Martínez, J. Estrada, Fitting a conic A-spline to contour image data, aceptado para publicar, Revista Investigación Operacional.

#### Eventos Científicos:

- CIMAFA'97, La Habana, Cuba, marzo 1997.  
*On the distance from a point to a conic.*
- XXXI Congreso de la Sociedad Matemática Mexicana, Hermosillo, México, octubre 1998.  
*A new algorithm to compute the euclidean distance from a point to a conic.*
- CIMAFA'99, La Habana, marzo 1999.  
*Reconstruction of 3D starlike objects from cross sections.*
- Conferencia Internacional Scattered Data Fitting, Puerto Vallarta, México, abril 1999.  
*Reconstruction of 3D starlike objects from cross sections.*
- Conferencia International Approximation and Optimization in the Caribbean, Guadeloupe, marzo, 1999.  
-A  $G^2$  continuous A-spline suitable for free form geometric design.  
-Numerical issues in locating the corner of the L-curve.
- 4th International Conference on Operation Research, La Habana, Cuba, marzo 2000  
*Fitting a conic A-spline to contour image data.*

- Conferencia Internacional Trends in Approximation Theory, Nashville, E. Unidos, mayo 2000.  
*Designing a G1 A-spline surface over a triangulation.*
- CIMAFA 2001, La Habana, marzo, 2001 y Conferencia International Approximation and Optimization in the Caribbean, Guatemala, marzo, 2001.  
*Geometric Modeling using Algebraic Splines*

## Capítulo 2

# Definiciones y resultados preliminares

Las curvas y superficies algebraicas se pueden representar de forma implícita y también, en algunas ocasiones, de forma racional paramétrica. En este capítulo vamos a introducir un conjunto de definiciones básicas y resultados relacionados con las curvas y superficies algebraicas, que se utilizarán a lo largo de todo el trabajo. Además estudiaremos la interrelación que existe entre la representación implícita y la representación paramétrica.

### 2.1 Curvas algebraicas planas

#### 2.1.1 Definiciones

**Definición 2.1** Una curva algebraica plana de grado  $n$  es el conjunto

$$\{(x, y) \in \mathbb{R}^2 \mid f(x, y) = 0\}$$

donde  $f$  es un polinomio irreducible de grado  $n$  con coeficientes reales  $c_{ij}$ ,

$$f(x, y) = \sum_{i+j \leq n} c_{ij} x^i y^j$$

Usualmente esta forma de representar las curvas algebraicas se conoce como *representación implícita*. Abusando de la notación, con frecuencia nos referiremos a una curva algebraica con el mismo nombre del polinomio que define su ecuación implícita. Las curvas algebraicas de grado 2 se llaman

*cónicas* y las de grado 3 *cúbicas*. Una subfamilia muy importante de las curvas algebraicas son las curvas que poseen una *parametrización racional*, las cuales se utilizan con frecuencia en la modelación geométrica. En lo adelante, nos referiremos a estas curvas simplemente como curvas paramétricas.

**Definición 2.2** *Una curva racional paramétrica de grado  $n$  es el conjunto*

$$\left\{ (x, y) \in \mathbb{R}^2 \mid x = \frac{f(t)}{h(t)}, y = \frac{g(t)}{h(t)} \right\}$$

donde  $f, g$  y  $h$  son polinomios de grado  $n$  con coeficientes reales, tales que  $\text{mcd}(f(t), g(t), h(t)) = 1$ .

Además del concepto de grado, con frecuencia se utiliza el concepto de *grado geométrico*, por cuanto la complejidad de muchos algoritmos depende del grado geométrico de la curva o superficie con que se trabaje.

**Definición 2.3** *El grado geométrico de una curva plana, es el máximo número de intersecciones entre la curva y una recta arbitraria, contando los puntos de intersección con coordenadas reales o complejas y las intersecciones múltiples.*

A partir del teorema de Bezout [Walk78] se puede concluir inmediatamente que el grado geométrico de una curva plana algebraica coincide con el grado del polinomio  $f$  que define su ecuación implícita. Por otro lado, para determinar el grado geométrico de una curva paramétrica de grado  $n$ , basta con sustituir las expresiones racionales de  $x$  y  $y$  en la ecuación implícita de una recta arbitraria  $ax + by + c = 0$ . De esta forma, llegamos a que el número máximo de intersecciones entre la curva y una recta coincide con el número máximo de raíces de un polinomio de grado  $n$ . Por lo tanto, las curvas paramétricas de grado  $n$ , también tienen grado geométrico  $n$ .

Utilizando la teoría de eliminación se puede demostrar que toda curva paramétrica posee una representación implícita [Sed84], pero el recíproco no es cierto. En efecto, aunque toda curva algebraica de grado 2 (cónica no degenerada) posee una parametrización racional, para  $n > 2$  el conjunto de las curvas algebraicas es estrictamente mayor que el de las curvas con parametrización racional. Por ejemplo, se puede probar que el conjunto de las curvas cúbicas con parametrización racional está formado por el subconjunto de las cúbicas que poseen un punto singular.



**Definición 2.4** Un punto  $p_0$  sobre la curva algebraica  $f(x, y) = 0$  se dice que es un **punto singular** si  $\nabla f(p_0) = (0, 0)$ . En otro caso, se dice que  $p_0$  es un punto regular. Una curva que posea al menos un punto singular se dice que es singular.

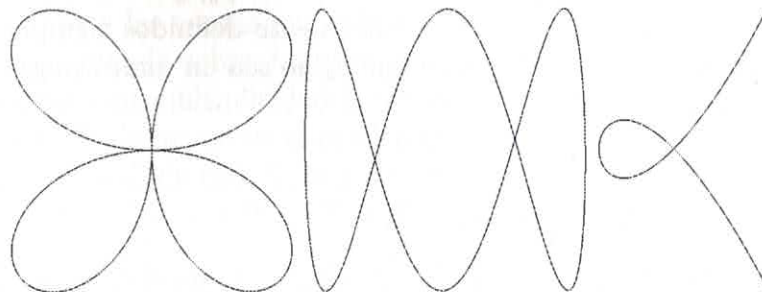


Figura 2.1: Curvas algebraicas con puntos singulares

Se puede probar que una curva algebraica de grado  $n$  posee a lo sumo  $\frac{1}{2}(n-1)(n-2)$  puntos singulares [Walk78].

Para los puntos regulares de una curva podemos definir de forma única la dirección normal a la curva en ese punto.

**Definición 2.5** Sea  $p_0 = (x_0, y_0)$  un punto sobre la curva algebraica  $f(x, y) = 0$ . La **dirección normal** o simplemente normal a la curva en el punto  $p_0$  es el vector  $\nabla f(x, y)(p_0) / \|\nabla f(x, y)(p_0)\|$ .

Nótese que si la curva algebraica está dada en forma paramétrica  $x = x(t)$ ;  $y = y(t)$  entonces la dirección normal a la curva en el punto  $p_0 = (x(t_0), y(t_0))$  es el vector  $(-\dot{y}(t_0), \dot{x}(t_0)) / \|(\dot{x}(t_0), \dot{y}(t_0))\|$ .

**Definición 2.6** Sea  $p_0$  un punto sobre la curva algebraica  $f(x, y) = 0$  y  $n_0$  la normal a la curva en  $p_0$ . La **recta tangente** a la curva en el punto  $p_0$  es la recta que pasa por  $p_0$  con dirección perpendicular a  $n_0$ .

Si  $n_0 = (n_{x0}, n_{y0})$  es la normal a la curva  $f$  en el punto  $p_0 = (x_0, y_0)$ , entonces la recta tangente a  $f$  en  $p_0$  tiene la ecuación implícita,

$$n_{x0}(x - x_0) + n_{y0}(y - y_0) = 0$$

Esta recta también se puede escribir paramétricamente como,

$$\begin{aligned}x(t) &= x_0 - tn_{y0} \\y(t) &= y_0 + tn_{x0}\end{aligned}$$

El vector normal  $n_0$  a una curva en un punto  $p_0$ , y en consecuencia la recta tangente en ese punto, están unívocamente definidos siempre que  $n_0$  no sea el vector nulo, es decir siempre que  $p_0$  no sea un punto singular de la curva.

### 2.1.2 Continuidad geométrica

En los capítulos 3 y 4 vamos a mostrar que varios problemas de modelación geométrica se pueden resolver exitosamente utilizando curvas definidas por pedazos, donde cada tramo es una curva algebraica. Esto nos conduce a la siguiente definición.

**Definición 2.7** *Una curva A-spline de grado  $n$  es una curva continua, definida por tramos, donde cada sección es una curva algebraica de grado  $n$ . Los puntos donde se unen dos tramos consecutivos del A-spline se llaman nodos o puntos de ruptura.*

En la solución de múltiples problemas de modelación es necesario utilizar curvas “suaves” y por lo tanto no basta con disponer de curvas continuas. Cuando se trabaja con curvas funcionales o paramétricas, esto se logra exigiendo que las mismas sean  $C^k$  continuas, usualmente para  $k$  pequeño. Por ejemplo, las curvas splines más populares son los splines cúbicos paramétricos, que son curvas definidas a trozos, donde cada sección es un polinomio de grado 3. Lo más usual en las aplicaciones es exigir que este tipo de curvas sean de clase  $C^2$  (que es la máxima suavidad que se puede pedir sin que todos los tramos se reduzcan al mismo polinomio), lo que es equivalente a pedir que esta condición se cumpla en los nodos.

En el caso de las curvas A-spline, el concepto de  $C^k$  continuidad se sustituye por el de  $G^k$  continuidad o continuidad geométrica de orden  $k$ . Esto se debe a que cuando trabajamos con curvas A-spline no se está presuponiendo ningún tipo de parametrización de la curva, de modo que lo interesante es exclusivamente el comportamiento geométrico de la misma.

Sean  $f : f(x, y) = 0$  y  $g : g(x, y) = 0$  dos curvas algebraicas planas que se intersecan en un punto  $p \in \mathbb{R}^2$ . El concepto  $G^k$  continuidad está estrechamente vinculado a la noción de la geometría algebraica de multiplicidad de la intersección. Intuitivamente hablando, un punto  $p$  situado sobre las curvas  $f$  y  $g$  es un punto de intersección de multiplicidad  $k$ , si al perturbar ligeramente los coeficientes de  $f$  y  $g$  resulta que  $p$  da lugar a  $k$  puntos de intersección entre las curvas perturbadas. Por ejemplo, si  $f$  es un círculo y  $p$  es un punto situado sobre  $f$ , entonces  $f$  y su recta tangente  $g$  en el punto  $p$  se intersecan con multiplicidad 2, ya que si perturbamos ligeramente los coeficientes de la recta, entonces resulta que ésta y el círculo se cortan en dos puntos. Para una definición formal de multiplicidad de la intersección entre dos curvas algebraicas vea [Hart77] y [Ful84].

Históricamente el concepto de continuidad geométrica de la intersección de dos curvas se introdujo de forma diferente a la que presentaremos a continuación. Sin embargo, gracias al trabajo [Garr91] se puede definir de una forma equivalente, más apropiada para su cálculo efectivo.

**Definición 2.8** Sean  $f : f(x, y) = 0$  y  $g : g(x, y) = 0$  dos curvas algebraicas planas y  $p \in \mathbb{R}^2$  un punto situado sobre ambas curvas. Se dice que  $f$  y  $g$  se unen con continuidad geométrica de orden  $k$  en el punto  $p$ , si la multiplicidad de la intersección entre  $f$  y  $g$  en el punto  $p$  es exactamente  $k + 1$ .

Para más detalles acerca del concepto de continuidad geométrica se puede consultar [Garr91], [Warr86], [Warr90].

Una forma práctica de hallar la multiplicidad de la intersección entre dos curvas planas es calcular la resultante entre ellas. Más precisamente, es conocido que [Walk78] si dos curvas algebraicas  $f$  y  $g$  pasan por un punto  $p = (p_x, p_y)$  y la recta  $y = p_y$  sólo corta simultáneamente a  $f$  y  $g$  en el punto  $p$ , entonces  $p_x$  es una raíz de la resultante  $R(x)$ , entre  $f$  y  $g$  con respecto a  $y$ , cuya multiplicidad coincide con la multiplicidad de la intersección entre las curvas  $f$  y  $g$  en el punto  $p$ . Si la recta  $y = p_y$  cortara a las curvas  $f$  y  $g$  en otro punto diferente de  $p$ , entonces habría que calcular la resultante  $R(y)$  entre  $f$  y  $g$  con respecto a  $x$  y determinar la multiplicidad de  $p_y$  como raíz de  $R(y)$ .

Es fácil comprobar que si las curvas algebraicas planas  $f$  y  $g$  pasan por un punto  $p$  y tienen la misma recta tangente en ese punto, entonces la coordenada correspondiente de  $p$  es una raíz de multiplicidad al menos 2 del

polinomio resultante, es decir la multiplicidad de la intersección entre  $f$  y  $g$  es al menos 2. Luego,  $f$  y  $g$  se unen con  $G^1$  continuidad en  $p$ , si comparten la misma recta tangente en ese punto. Esto conduce a la siguiente definición equivalente de  $G^1$  continuidad para una curva A-spline.

**Definición 2.9** *Sea  $S$  una curva A-spline plana. Se dice que  $S$  es  $G^1$  continua si para todo punto  $p$  sobre la curva se tiene que  $S$  y la recta tangente a  $S$  son funciones continuas en  $p$ .*

Nótese que en efecto, la definición de  $G^1$  continuidad enfatiza el hecho geométrico de que la tangente a la curva varía continuamente a lo largo de la misma. Por supuesto, toda curva spline paramétrica  $C^1$  continua, es una curva  $G^1$  continua, pero el recíproco no es cierto. En [Far93] se muestra por ejemplo, que para que una curva spline escrita en la forma de Bézier sea  $C^1$  continua no basta con que la tangente por la izquierda de cada nodo coincida con la tangente por la derecha, o sea que la curva sea  $G^1$  continua. Se requiere adicionalmente, que la razón entre las longitudes de los segmentos definidos por el nodo y sus vecinos en el polígono de control de la curva, coincida con las razones correspondientes entre los valores del parámetro asignados a los 3 nodos consecutivos. En otras palabras, la  $C^1$  continuidad depende adicionalmente de la parametrización introducida, de modo que la misma curva puede ser  $C^1$  continua con respecto a una parametrización y no serlo con respecto a otra. Sin embargo, para todo spline paramétrico  $G^1$  continuo es posible encontrar una reparametrización tal que la curva sea  $C^1$  continua (vea también [Garr91]).

En algunos problemas prácticos no basta con que la curva que se construye sea  $G^1$  continua, es necesario un mayor orden de suavidad (véase el capítulo 4). Por eso, se introduce el concepto de  $G^2$  continuidad. Supongamos que  $f$  y  $g$  son curvas algebraicas planas que pasan por un punto  $p$  y tienen la misma recta tangente en ese punto. Entonces ya vimos que  $p$  es una raíz de multiplicidad al menos 2 del polinomio resultante. Si adicionalmente  $f$  y  $g$  tienen igual curvatura en el punto  $p$  entonces se puede probar que la coordenada correspondiente  $p$  es una raíz del polinomio resultante de multiplicidad al menos 3. Cuando  $p$  no es un punto singular ni de  $f$  ni de  $g$  el recíproco también es cierto, por lo tanto la multiplicidad es al menos 3 si y sólo si las curvas planas  $f$  y  $g$  pasan por  $p$  y tienen la misma recta tangente y la misma curvatura en ese punto. Luego,  $f$  y  $g$  se unen con  $G^2$  continuidad

en  $p$  ( $p$  no es punto singular), si y sólo si comparten la misma recta tangente y la misma curvatura en ese punto.

Recordemos que para una curva plana definida implícitamente  $f(x, y) = 0$  la curvatura  $\kappa$  en un punto  $(x_0, y_0)$  situado sobre la curva está dada por:

$$\kappa = \frac{f_{yy}f_x^2 - 2f_xf_yf_{xy} + f_{xx}f_y^2}{(f_x^2 + f_y^2)^{3/2}} \quad (2.1)$$

donde todas las derivadas se evalúan en el punto  $(x_0, y_0)$ .

Entonces una definición equivalente de  $G^2$  continuidad para una curva A-spline plana es la siguiente:

**Definición 2.10** *Sea  $S$  una curva spline plana. Se dice que  $S$  es  $G^2$  continua si es  $G^1$  continua y para todo punto  $p$  sobre  $S$  la curvatura es una función continua en  $p$ .*

El concepto de  $G^2$  continuidad que acabamos de introducir se refiere exclusivamente a curvas planas, pues para curvas en el espacio es necesario exigir adicionalmente la variación continua del vector binormal [Far93].

### 2.1.3 Coordenadas baricéntricas

En los próximos dos capítulos vamos a presentar curvas A-spline, donde cada sección que compone la curva está definida en el interior de un triángulo. Esto sugiere introducir un nuevo sistema de coordenadas, llamadas *coordenadas baricéntricas*, que sólo dependen de los vértices del triángulo y en términos de las cuales la ecuación de la curva se simplifica considerablemente, así como el estudio de algunas de sus propiedades geométricas.

Consideremos un triángulo en el plano con vértices  $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{E}^2$  y otro punto cualquiera  $\mathbf{p} = (x, y)$  también en  $\mathbb{E}^2$ . Entonces  $\mathbf{p}$  siempre se puede escribir como combinación lineal baricéntrica de  $\mathbf{a}, \mathbf{b}, \mathbf{c}$ , es decir existen  $u, v, w \in \mathbb{R}$  tales que,

$$\mathbf{p} = u\mathbf{a} + v\mathbf{b} + w\mathbf{c} \quad (2.2)$$

Nótese que la expresión (2.2) solamente tiene sentido geométrico si define una combinación baricéntrica, es decir si exigimos que adicionalmente,

$$u + v + w = 1 \quad (2.3)$$

Los coeficientes  $\mathbf{u} := (u, v, w)$  se llaman *coordenadas baricéntricas* de  $\mathbf{p}$  con respecto a los puntos  $\mathbf{a}, \mathbf{b}, \mathbf{c}$ . Las ecuaciones (2.2) y (2.3) definen un sistema lineal de 3 ecuaciones para calcular las incógnitas  $u, v, w$ ,

$$\begin{pmatrix} a_x & b_x & c_x \\ a_y & b_y & c_y \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (2.4)$$

Como los puntos  $\mathbf{a} = (a_x, a_y)$ ,  $\mathbf{b} = (b_x, b_y)$ ,  $\mathbf{c} = (c_x, c_y)$  no son colineales, la matriz  $A$  de este sistema lineal es no singular. Denotemos por  $area(\mathbf{a}, \mathbf{b}, \mathbf{c})$  el área del triángulo con vértices  $\mathbf{a}, \mathbf{b}, \mathbf{c}$ . Es bien conocido que,  $area(\mathbf{a}, \mathbf{b}, \mathbf{c}) = \frac{1}{2}det(A)$ , donde  $det(A)$  denota el determinante de  $A$ .

Por lo tanto, usando la regla de Cramer tenemos que la solución del sistema (2.4), es decir las coordenadas baricéntricas del punto  $\mathbf{p}$ , se pueden escribir en términos de sus coordenadas afines  $(x, y)$  como,

$$u = \frac{area(\mathbf{p}, \mathbf{b}, \mathbf{c})}{area(\mathbf{a}, \mathbf{b}, \mathbf{c})} \quad v = \frac{area(\mathbf{a}, \mathbf{p}, \mathbf{c})}{area(\mathbf{a}, \mathbf{b}, \mathbf{c})} \quad w = \frac{area(\mathbf{a}, \mathbf{b}, \mathbf{p})}{area(\mathbf{a}, \mathbf{b}, \mathbf{c})}$$

Nótese que en particular las coordenadas baricéntricas de los vértices  $\mathbf{a}, \mathbf{b}, \mathbf{c}$  del triángulo son  $(1, 0, 0)$ ,  $(0, 1, 0)$  y  $(0, 0, 1)$  respectivamente. Por otro lado, las coordenadas baricéntricas son invariantes por transformaciones afines, pues son los coeficientes de la combinación baricéntrica de un punto en términos de los vértices de un triángulo. Más precisamente, si  $(u, v, w)$  son las coordenadas baricéntricas de  $\mathbf{p}$  con respecto a los vértices  $\mathbf{a}, \mathbf{b}, \mathbf{c}$  de un triángulo y  $\Phi$  es una transformación afín entonces  $\Phi(\mathbf{p}) = u\Phi(\mathbf{a}) + v\Phi(\mathbf{b}) + w\Phi(\mathbf{c})$ .

#### 2.1.4 Expresión en coordenadas baricéntricas

Sea  $c_i$  la  $i$ -ésima sección de una curva A-spline, definida en el interior de un triángulo  $T_i$ . Como mencionamos en la sección anterior, la introducción de las coordenadas baricéntricas nos permitirá encontrar una ecuación general simple para todas las curvas  $c_i$ , cuya dependencia de los vértices  $P_0^i, P_1^i, P_2^i$  del triángulo  $T_i$  se “esconde” en la expresión del cambio de coordenadas baricéntricas a coordenadas afines.

Sea  $\Phi$  la transformación lineal de las coordenadas baricéntricas  $(u, v, w)$  asociadas al triángulo  $T_i$  a las coordenadas afines  $(x, y)$ , dada por,

$$\begin{pmatrix} P_{0x}^i & P_{1x}^i & P_{2x}^i \\ P_{0y}^i & P_{1y}^i & P_{2y}^i \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (2.5)$$

donde  $P_j^i = (P_{jx}^i, P_{jy}^i)$  con  $j = 0, 1, 2$ .

Los vértices del triángulo  $T_i$  no son colineales, por consiguiente la matriz  $A_i$  del sistema (2.5) es no singular y la transformación  $\Phi$  tiene inversa  $\Phi^{-1}$ . Por otra parte, como  $\Phi^{-1}$  es una transformación afín, las coordenadas baricéntricas de un punto  $p$  con relación a los vértices del triángulo  $T_i$ , coinciden con las coordenadas baricéntricas de  $\Phi^{-1}(p)$  con relación a los puntos  $\Phi^{-1}(P_0^i) = (1, 0, 0)$ ,  $\Phi^{-1}(P_1^i) = (0, 1, 0)$  y  $\Phi^{-1}(P_2^i) = (0, 0, 0)$ . En lo adelante, nos referiremos al triángulo con vértices  $p_0 = (0, 0)$ ,  $p_1 = (0, 1)$  y  $p_2 = (1, 0)$  como triángulo canónico.

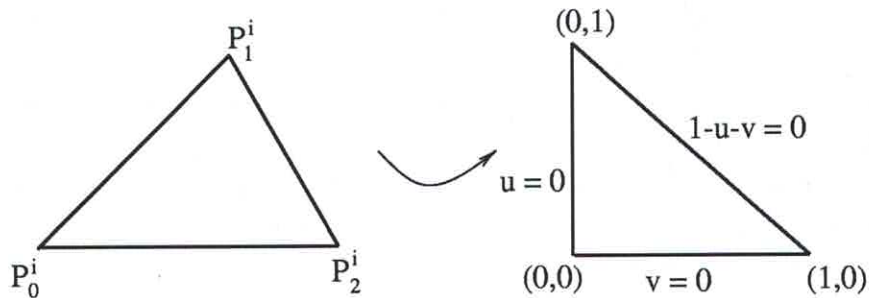


Figura 2.2: Paso de coordenadas reales a coordenadas baricéntricas

Sea  $g_i(x, y) = 0$  la ecuación implícita de la curva algebraica  $c_i$ , que define la  $i$ -ésima sección del A-spline  $S$ , donde  $g_i(x, y)$  es un polinomio de grado  $n$ . Del sistema (2.5), resulta que las coordenadas afines  $(x, y)$  se pueden expresar linealmente en términos de las coordenadas baricéntricas  $(u, v, 1 - u - v)$ , más precisamente,

$$\begin{aligned} x &= P_{0x}^i u + P_{1x}^i v + P_{2x}^i (1 - u - v) := x(u, v) \\ y &= P_{0y}^i u + P_{1y}^i v + P_{2y}^i (1 - u - v) := y(u, v) \end{aligned}$$

Por lo tanto, sustituyendo en la ecuación  $g_i(x, y) = 0$ , obtenemos la ecuación implícita de  $c_i$ , en coordenadas baricéntricas respecto a los vértices del triángulo  $T_i$ ,

$$g_i(x, y) = g_i(x(u, v), y(u, v)) := f_i(u, v) = 0 \quad (2.6)$$

Si  $p$  es un punto situado sobre la curva algebraica  $g_i(x, y) = 0$ , entonces  $\Phi^{-1}(p)$  es un punto situado sobre la curva algebraica  $f_i(u, v) = 0$  (resultante de hacer el correspondiente cambio de coordenadas) y como  $\Phi^{-1}$  es una transformación lineal, las curvas  $g_i(x, y) = 0$  y  $f_i(u, v) = 0$  son del mismo grado.

### 2.1.5 Representación de Bézier en coordenadas baricéntricas

Una forma de obtener la representación de Bézier de una curva algebraica escrita en coordenadas baricéntricas es a través de una de las superficies más populares en el área de CAGD, los parches triangulares de Bézier [Far93]. En el Anexo A de este trabajo incluimos una breve descripción de los mismos, que permite al lector completar todos los detalles.

La curva algebraica  $f_i(u, v) = 0$  se puede considerar como una curva de nivel de un parche triangular de Bézier no paramétrico, a saber, como la curva de nivel  $z = 0$  del parche triangular (A.3). Por lo tanto, haciendo uso de la expresión de los polinomios de Bernstein llegamos a la forma de Bézier de una curva algebraica dada en coordenadas baricéntricas,

$$\begin{aligned} f_i(u, v) &= \sum_{j=0}^n \sum_{k=0}^{n-j} \frac{n!}{k!j!(n-k-j)!} c_{k,j,n-k-j} u^k v^j w^{n-k-j} \\ &= \sum_{j=0}^n \sum_{k=0}^{n-j} \frac{n!}{k!j!(n-k-j)!} a_{kj} u^k v^j (1-u-v)^{n-k-j} \end{aligned} \quad (2.7)$$

donde  $a_{kj} = c_{k,j,n-k-j}$ .

### 2.1.6 Familia de curvas algebraicas definidas en el interior de un triángulo

En la literatura, es usual organizar los coeficientes  $a_{kj}^i$  de la expresión (2.7) en un arreglo triangular como el que se muestra en la figura (2.3) para el caso de las cúbicas, es decir tomando  $n = 3$ .



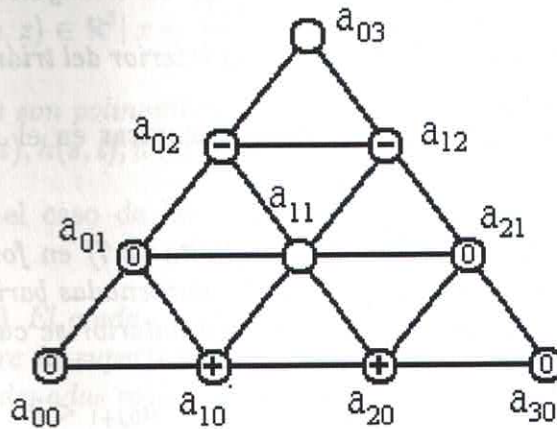


Figura 2.3: Coeficientes de una cúbica escrita en coordenadas baricéntricas

El siguiente resultado debido a [Baj99a]) es muy importante, pues nos garantiza que bajo ciertas condiciones en los coeficientes de la representación de Bézier de una curva algebraica, existe una familia de tales curvas con “buenas” propiedades en el interior del triángulo canónico.

**Lema 2.1** Sea  $f_i(u, v) = 0$  la ecuación implícita (2.7) en forma de Bézier de una curva algebraica de grado  $n$  dada en coordenadas baricéntricas. Si existe  $l \in \mathbb{Z}$  con  $0 < l < n$  tal que,

$$\begin{aligned}
 a_{kj} &\geq 0 && \text{para } k = 0, \dots, n-j \quad j = 0, \dots, l-1 \\
 a_{kj} &\leq 0 && \text{para } k = 0, \dots, n-j \quad j = l+1, \dots, n \\
 \sum_{k=0}^n a_{k0} &> 0 \\
 \sum_{k=0}^{n-j} a_{kj} &< 0 && \text{para algún } j \text{ con } k < j \leq n
 \end{aligned}$$

entonces,

- i) Para todo  $\beta$  tal que  $0 < \beta < 1$  la recta  $u = \beta(1 - v)$  que pasa por el punto  $p_1$ , interseca a la curva  $f_i(u, v) = 0$  una sola vez (contando

*multiplicidades) en el interior del triángulo canónico. Esto equivale a decir que la curva es conexa en el interior del triángulo.*

ii) *La curva  $f_i(u, v) = 0$  es no singular en el interior del triángulo canónico.*

La demostración de este lema se puede encontrar en el Anexo B y en [Baj99a)].

**Lema 2.2** *Sea  $f_i(u, v) = 0$  la ecuación implícita (2.7) en forma de Bézier de una curva algebraica de grado  $n$ , dada en coordenadas baricéntricas. Supongamos que además de las hipótesis del lema anterior se cumple que,*

$$a_{0j} = 0 \quad \text{para } j = 0, \dots, l \quad \text{y} \quad a_{0, l+1} < 0$$

*entonces la curva  $f_i(u, v) = 0$  pasa por el punto  $p_0$  y es tangente a la recta  $u = 0$  con multiplicidad  $l + 1$ . Además, para  $0 < v < 1$  no existe otra intersección entre la curva y la recta  $u = 0$ . Similarmente, si,*

$$a_{n-j, j} = 0 \quad \text{para } j = 0, \dots, l \quad \text{y} \quad a_{n-(l+1), l+1} < 0$$

*entonces la curva  $f_i(u, v) = 0$  pasa por el punto  $p_2$  y es tangente a la recta  $1 - u - v = 0$  con multiplicidad  $l + 1$ . Además, para  $u, v > 0$  no existe otra intersección entre la curva y la recta  $1 - u - v = 0$ .*

La demostración de este resultado se puede encontrar en [Baj99a)].

## 2.2 Superficies algebraicas

### 2.2.1 Definiciones

**Definición 2.11** *Una superficie algebraica de grado  $n$  en  $\mathbb{R}^3$  es el conjunto*

$$\{(x, y, z) \in \mathbb{R}^3 \mid f(x, y, z) = 0\}$$

*donde  $f$  es un polinomio irreducible de grado  $n$  con coeficientes reales  $c_{ijk}$ ,*

$$f(x, y, z) = \sum_{i+j+k \leq n} c_{ijk} x^i y^j z^k$$

**Definición 2.12** Una superficie paramétrica de grado  $n$  es el conjunto

$$\{(x, y, z) \in \mathbb{R}^3 \mid x = \frac{f(s, t)}{w(s, t)}, y = \frac{g(s, t)}{w(s, t)}, z = \frac{h(s, t)}{w(s, t)}\}$$

donde  $f, g, h$  y  $w$  son polinomios de grado  $n$  con coeficientes reales, tales que  $\text{mcd}(f(s, t), g(s, t), h(s, t), w(s, t)) = 1$ .

También en el caso de las superficies se utiliza el concepto de grado geométrico.

**Definición 2.13** El grado geométrico de una superficie es el número de intersecciones entre la superficie y una recta, contando los puntos de intersección con coordenadas reales o complejas y las intersecciones múltiples.

A diferencia de las curvas paramétricas planas donde el grado geométrico coincide con el grado, una superficie paramétrica de grado  $n$  que no tenga puntos base tiene grado geométrico  $n^2$ . En efecto, si sustituimos las expresiones racionales de  $x, y$  y  $z$  en la ecuación implícita de una recta arbitraria en el espacio  $a_1x + b_1y + c_1z + d_1 = 0$ ,  $a_2x + b_2y + c_2z + d_2 = 0$  llegamos a que el máximo número de intersecciones entre la superficie y una recta, coincide con el máximo número de intersecciones entre dos curvas algebraicas de grado  $n$ , que según el teorema de Bezout es  $n^2$ . No es difícil probar que si la superficie algebraica tuviese  $r$  puntos base, entonces su grado geométrico sería  $n^2 - r$ .

Por otro lado, una superficie algebraica implícita de grado  $n$  también tiene grado geométrico  $n$ . En efecto, si sustituimos la ecuación paramétrica de la recta en la ecuación implícita de la superficie, resulta que el número de intersecciones coincide con el número de raíces de un polinomio univariado del mismo grado que la superficie.

Por lo tanto, el grado geométrico de una superficie paramétrica de grado  $n$ , es el cuadrado del grado geométrico de una superficie implícita del mismo grado. Esto ha sugerido a muchos autores la conveniencia de utilizar superficies definidas implícitamente para resolver problemas de modelación geométrica, ya que la eficiencia computacional de cualquier operación que se realice con la superficie depende en gran medida de su grado geométrico. Por ejemplo, la intersección de dos superficies de grado geométrico  $n$  puede ser una curva de grado geométrico  $n^2$  y ésta a su vez puede poseer hasta  $n^4$  puntos singulares.

**Definición 2.14** Un punto  $p_0 = (x_0, y_0, z_0)$  sobre la superficie  $f$  se dice que es un **punto singular** si el vector  $\nabla f(p_0)$  es el vector nulo. En otro caso, se dice que  $p_0$  es un punto regular. Una superficie que posea al menos un punto singular se dice que es singular.

Para los puntos regulares de una superficie podemos definir de forma única la dirección normal a la superficie.

**Definición 2.15** Sea  $p_0 = (x_0, y_0, z_0)$  un punto sobre la superficie algebraica  $f(x, y, z) = 0$ . La **dirección normal** a la superficie en el punto  $p_0$  es el vector  $\nabla f(x, y, z)(p_0) / \|\nabla f(x, y, z)(p_0)\|$ .

Si la superficie algebraica está dada en forma paramétrica  $r(s, t) : \{x = x(s, t), y = y(s, t), z = z(s, t)\}$  entonces la dirección normal en el punto  $p_0 = (x(s_0, t_0), y(s_0, t_0), z(s_0, t_0))$  es el vector  $n_0 := r_s(s_0, t_0) \times r_t(s_0, t_0) / \|r_s(s_0, t_0) \times r_t(s_0, t_0)\|$ , donde  $r_s = (x_s, y_s, z_s)$  y  $r_t = (x_t, y_t, z_t)$ .

Ahora ya estamos en condiciones de definir el plano tangente a una superficie en un punto situado sobre la misma.

**Definición 2.16** Sea  $p_0$  un punto sobre la superficie algebraica  $f(x, y, z) = 0$  y  $n_0$  la normal a la superficie en  $p_0$ . El **plano tangente** a la superficie en el punto  $p_0$  es el plano que pasa por  $p_0$  con dirección normal igual a  $n_0$ .

Si  $n_0 = (n_{x0}, n_{y0}, n_{z0})$  es la normal a la superficie  $f$  en el punto  $p_0 = (x_0, y_0, z_0)$ , entonces la ecuación implícita del plano tangente a  $f$  en  $p_0$  está dada por,

$$n_{x0}(x - x_0) + n_{y0}(y - y_0) + n_{z0}(z - z_0) = 0$$

Nótese que el vector normal a la superficie  $f$  en un punto  $p_0$ , y en consecuencia el plano tangente en ese punto, están unívocamente definidos siempre que  $n_0$  no sea el vector nulo, es decir siempre que  $p_0$  no sea un punto singular de la superficie.

Para obtener la ecuación paramétrica del plano tangente, basta con tener en cuenta que éste es generado por los vectores  $r_s$  y  $r_t$ . Más precisamente, el plano tangente a la superficie  $r(s, t)$  en el punto  $p_0$  está dado por,

$$\begin{pmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} + r_s(p_0)u + r_t(p_0)v \quad (2.8)$$

### 2.2.2 Continuidad geométrica

En el capítulo 5 de este trabajo vamos a mostrar cómo resolver un problema de interpolación en el espacio, utilizando superficies definidas por pedazos, donde cada sección es una superficie algebraica. Esto nos lleva a la siguiente definición.

**Definición 2.17** *Una superficie A-spline de grado  $n$  es una superficie continua, definida por pedazos, donde cada pedazo es una superficie algebraica de grado  $n$ . Los pedazos que definen la superficie se conocen como parches.*

En el capítulo 5 vamos a construir una superficie A-spline “triangular”, es decir una superficie donde cada uno de los parches interpola los vértices de un triángulo en el espacio. Las curvas que definen la frontera de cada parche se conocen en la literatura como “malla de alambre” (wireframe) de la superficie o curvas de la frontera. Por otra parte, para resolver problemas de modelación geométrica se necesitan con frecuencia superficies “suaves”. Esto nos conduce también al concepto de  $G^k$  continuidad de la intersección entre dos superficies, que como veremos a continuación se reduce en última instancia a la  $G^k$  continuidad de las curvas planas que pasan por cada uno de los puntos de intersección de las dos superficies.

**Definición 2.18** *Sean  $f : f(x, y, z) = 0$  y  $g : g(x, y, z) = 0$  dos superficies algebraicas y  $p \in \mathbb{R}^3$  un punto situado sobre la curva intersección entre  $f$  y  $g$ . Se dice que  $f$  y  $g$  se unen con continuidad geométrica de orden  $k$  en el punto  $p$ , si para todo plano  $\Pi$  que interseca transversalmente en  $p$  tanto a  $f$  como a  $g$ , se cumple que la multiplicidad de la intersección en el punto  $p$  entre las curvas  $f \cap \Pi$  y  $g \cap \Pi$  es exactamente  $k$ .*

Nótese que de acuerdo con la definición previa la continuidad geométrica de la intersección de dos superficies se reduce a la continuidad geométrica de la intersección de curvas planas.

Supongamos que las superficies  $f$  y  $g$  se intersecan en la curva plana  $c$  y que comparten el mismo plano tangente para todo punto  $p$  situado sobre  $c$ . Entonces es evidente que para todo plano  $\Pi$  que pasa por  $p$  y corta transversalmente a  $f$  y  $g$  en ese punto, se cumple que las curvas  $f \cap \Pi$  y  $g \cap \Pi$  poseen la misma recta tangente en el punto  $p$ . Por lo tanto, las superficies  $f$  y  $g$  se intersecan con  $G^1$  continuidad a lo largo de la curva  $c$ .

Esto nos conduce a la siguiente definición de  $G^1$  continuidad para una superficie A-spline.

**Definición 2.19** *Una superficie A-spline  $S$  se dice que es  $G^1$  continua si para todo punto  $p$  sobre la superficie se tiene que  $S$  y el plano tangente a  $S$  son funciones continuas en  $p$ .*

De acuerdo con la definición anterior para que una superficie A-spline no singular sea  $G^1$  continua, basta con verificar que el plano tangente varía continuamente a lo largo de las curvas de la frontera de cada parche.

## Capítulo 3

# Ajuste del contorno de una imagen digital por medio de un A-spline cónico

### 3.1 Introducción

El problema de aproximar un conjunto de puntos por medio de una curva “suave” ha sido extensivamente tratado en la literatura, basta con citar la siguiente lista de artículos, que sólo representa una pequeña muestra de los existentes: [Baj94], [Baj99a], [Book79], [Gan94], [Oden93], [Pav83], [Ray94], [Sam82], [Spa96], [Shum90]. Las curvas que más comúnmente se utilizan para resolver este problema son las paramétricas [Gan94], [Oden93], [Spa96], [Lozo83] y entre ellas las curvas splines paramétricas. Esto se debe a las ventajas generales que tiene el empleo de curvas paramétricas, entre las cuales vale la pena destacar el hecho de que resulta más sencillo aislar un segmento conexo de las mismas, además de que son muy fáciles de graficar. Sin embargo, para ajustar un conjunto de datos por medio de una curva paramétrica es necesario introducir primero una parametrización de los datos, es decir asignar, de manera artificial, a cada punto de los datos un valor del parámetro que describirá la curva. Por lo tanto aquí surge el problema de cómo escoger una parametrización adecuada, o en otras palabras de cómo depende la calidad del ajuste de tal selección. Por otra parte, una vez que se ha escogido la parametrización, y una base para representar la curva spline, los coeficientes de la curva escrita como combinación lineal de la base se calculan todos a la

vez, resolviendo un sistema de ecuaciones lineales, cuya dimensión depende linealmente de la cantidad de datos. En este caso, si algún dato varía entonces tendríamos que volver a calcular algunas de las secciones del spline paramétrico.

En los últimos años han comenzado a aparecer soluciones al problema de ajuste de datos utilizando curvas splines algebraicas (A-splines) definidas implícitamente [Baj99a), [Baj99b)], [Baj94], [Tau91]. En comparación con los splines paramétricos, las curvas A-spline nos permiten un mayor control local, pues una vez que se imponen las condiciones de  $G^1$  continuidad, el resto de los parámetros libres de cada sección de la curva A-spline sólo depende de los datos contenidos en el triángulo de control de ese segmento.

En este capítulo vamos a comparar algunas de las técnicas existentes para ajustar, por medio de una curva A-spline, los puntos que describen el contorno de una imagen digital. Nuestro objetivo es proponer un algoritmo para resolver este problema, más efectivo que los reportados en la literatura, en el caso específico en que los datos provienen de una imagen digital.

Más precisamente, vamos a comparar algunos métodos conocidos para determinar los nodos de una curva spline. Adicionalmente vamos a proponer un método sencillo para localizar, a partir de un conjunto discreto de datos, la posición de un punto de inflexión de la curva con que se ajustarán los mismos. También estudiaremos cómo calcular la tangente en cada nodo del spline, un subproblema interesante cuando los datos son las coordenadas de los píxeles que describen el contorno de una imagen, y en consecuencia aparecen con frecuencia configuraciones de tipo "escalera". Por último, queremos llamar la atención acerca de la importancia de la selección de una buena aproximación de la distancia de un punto a una cónica, pues ello influye de manera importante en la calidad del ajuste que se logra en el interior de cada triángulo.

## 3.2 Preliminares

Como vimos en el Capítulo 2, un A-spline es una curva continua, definida por tramos, donde cada sección  $c_i$   $i = 1, \dots, n$  es una curva algebraica. En este capítulo vamos a construir en particular un A-spline cónico, es decir una curva A-spline donde cada segmento  $c_i$  es una curva algebraica de grado 2 (cónica), definida en el interior de un triángulo  $T_i$ . Denotemos por  $P_0^i, P_1^i, P_2^i$  los vértices del triángulo  $T_i$ . Más adelante veremos cómo se escogen los



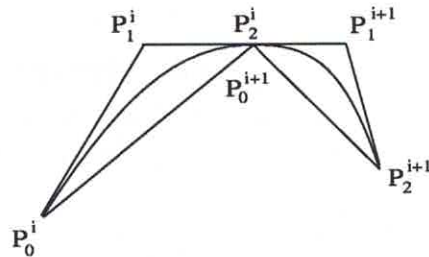


Figura 3.1: Dos secciones consecutivas de un A-spline cónico

puntos  $P_0^i$  y  $P_2^i$  que son datos del problema, es decir pertenecen al contorno de la imagen digital. Además explicaremos cómo se calculan las direcciones tangentes en esos puntos. El vértice  $P_1^i$  es el punto de intersección de las tangentes definidas en  $P_0^i$  y  $P_2^i$ .

En lo adelante vamos a imponer que cada curva  $c_i$  interpole los puntos  $P_0^i$  y  $P_2^i$  y las direcciones tangentes prefijadas en esos puntos. Por otra parte, para garantizar que la curva A-spline  $S$  sea continua, debemos exigir que  $P_2^i = P_0^{i+1}$  para  $i = 1, \dots, n-1$  y adicionalmente que  $P_2^n = P_0^1$ , si quisiéramos construir una curva cerrada.

Como ya mencionamos anteriormente, los puntos  $P_0^i$   $i = 2, \dots, n$  se llaman nodos del A-spline y los vértices de los triángulos  $T_i$   $i = 1, \dots, n$ , en cuyo interior está definida cada sección de la curva A-spline, forman el llamado polígono de control. Por otro lado, para lograr que  $S$  sea una curva  $G^1$  continua debemos exigir que los puntos  $P_1^i, P_2^i$  y  $P_1^{i+1}$ ,  $i = 1, \dots, n-1$  sean colineales.

Resumiendo, los pasos principales del algoritmo para construir el A-spline cónico que ajusta los puntos que representan el contorno de la imagen digital son los siguientes:

#### Algoritmo ASPLFIT

- 1) Determinar los nodos o puntos de ruptura del A-spline.
- 2) Calcular la dirección tangente en cada nodo.
- 3) Determinar el polígono de control del A-spline.
- 4) Ajustar los datos en el interior de cada triángulo por medio de una cónica.

- 5) Graficar la curva A-spline.

Siguiendo cada uno de los pasos del algoritmo ASPLFIT, que describimos con detalle en el resto del capítulo, se puede probar directamente que el número máximo de operaciones que requiere este método es  $O(m^2)$ , donde  $m$  representa la cantidad de puntos del contorno. Sin embargo, es importante señalar que aunque el número de operaciones es proporcional a  $m^2$ , la constante de proporcionalidad es del tipo  $1/\alpha$  donde  $\alpha$  es un número entero grande.

### 3.3 Cálculo de los nodos del A-spline

En la literatura se pueden encontrar varios métodos para calcular los nodos de una curva spline, definida paramétrica o implícitamente. En esta sección mencionaremos algunos de ellos y vamos a determinar cuál es el más apropiado, cuando se trata de construir una curva A-spline que aproxima el contorno de una imagen digital.

En [Baj94], Bajaj propone un algoritmo para seleccionar los nodos de una curva A-spline, teniendo en cuenta las normales a los segmentos de la poligonal que une los datos. Para ello considera un polígono regular de  $k$  lados, circunscrito al círculo unitario y numera consecutivamente las direcciones normales a cada lado del polígono, con los enteros entre 1 y  $k$ . La idea consiste en comparar la normal al segmento que une dos puntos consecutivos del contorno, con cada una de las  $k$  direcciones normales a los lados del polígono regular. Más precisamente, al segmento que une dos puntos del contorno se le asigna el valor  $i$ , con  $1 \leq i \leq k$ , si el producto escalar entre la normal al segmento y las normales a los lados del polígono regular, alcanza un máximo para el  $i$ -ésimo lado polígono. De esta manera, los segmentos que unen dos puntos consecutivos del contorno y que tienen asignado el mismo valor  $i$  forman un conjunto, y los puntos extremos de cada conjunto serán los nodos que estamos buscando. Nótese que la sucesión de nodos que nos proporciona este método no depende de cuál es el punto inicial del contorno que estamos analizando. Sin embargo, de acuerdo con nuestra experiencia, este método no funciona correctamente cuando los puntos del contorno son píxeles, pues en tal caso aparecen con frecuencia en el contorno, poligonales en forma de "escalera". En consecuencia, los grupos que define el algoritmo están constituidos por muy pocos puntos y por el contrario demasiados puntos resultan escogidos como nodos.

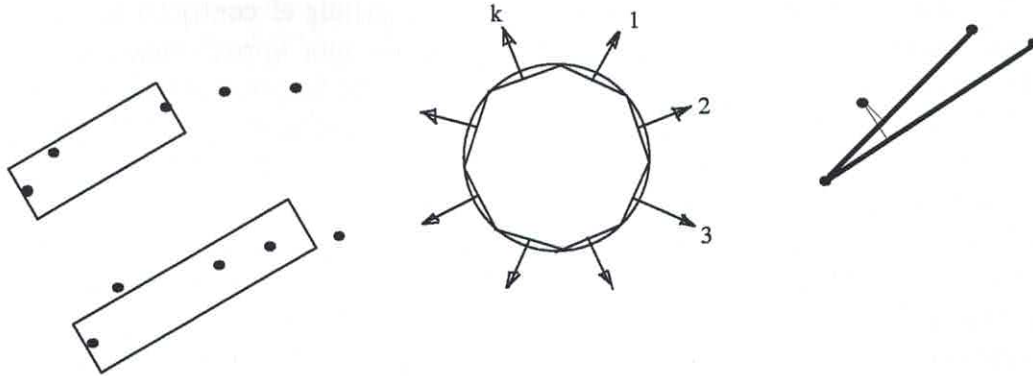


Figura 3.2: Selección de los nodos. Izquierda: usando una máscara. Centro: discretizando las direcciones normales. Derecha: construyendo una poligonal.

Otro método para seleccionar los nodos de una curva spline es el propuesto por Odesanya en [Oden93], que a su vez es una versión mejorada del algoritmo propuesto inicialmente por Lozoover y Preiss en [Lozo83]. Este método determina los nodos utilizando una ventana de tolerancia o máscara. Inicialmente se coloca la máscara en el primer punto del contorno, haciendo coincidir la recta que divide a la mitad el ancho de la máscara con la recta que pasa por el primer y el tercer punto del contorno. En cada paso, la máscara se extiende hasta el siguiente punto del contorno y el proceso se repite hasta que alguno de los puntos intermedios se salga de la máscara. Cuando esto sucede, se toma como nodo el penúltimo punto considerado del contorno. A diferencia del método propuesto por Bajaj, este algoritmo tiene la desventaja de que la sucesión de nodos que se obtiene depende de cuál sea el punto inicial donde se coloca la máscara. En [Oden93], los autores proponen utilizar cada punto del contorno como punto inicial de la máscara y a continuación construir el spline (paramétrico) que interpola los nodos obtenidos. Finalmente seleccionan el spline "óptimo", calculando una aproximación de la distancia euclídeana de los puntos del contorno a cada uno de los splines construidos y escogiendo aquel para el cual se alcanza la mínima distancia. En la práctica, este último procedimiento puede ser muy costoso cuando la cantidad de datos es grande, lo cual ocurre en el caso de imágenes digitales. En este trabajo no aparece ningún comentario acerca de cuál es el ancho "óptimo" de la máscara, aunque obviamente, en la medida que el ancho de la máscara crece, disminuye el número de nodos que nos proporciona el algoritmo.

En [Ray94] se propone un método para comprimir el contorno de una imagen digital que se basa en la construcción de una aproximación poligonal del contorno. La idea del método consiste en tratar que cada recta de la poligonal tenga la mayor longitud posible, y que a su vez el error de aproximación no crezca. Partiendo de un punto arbitrario  $P_i$  sobre el contorno de la imagen, la estrategia consiste en construir la recta  $r_{ij}$  que pasa por  $P_i$  y  $P_j$  con  $j > i + 1$  e ir incrementando  $j$  mientras la función  $F_j := L_j - E_j$  crezca, donde  $L_j$  denota la longitud del segmento  $P_i P_j$ , mientras  $E_j$  denota la suma de los cuadrados de las distancias de todos los puntos intermedios a la recta  $r_{ij}$ . Este algoritmo se puede utilizar para seleccionar los nodos de la curva spline, simplemente tomando como nodos los vértices de la poligonal que ajusta el contorno. Según los autores, respecto a otros algoritmos similares, este método tiene la ventaja de que funciona de forma totalmente automática, porque no depende de un parámetro que controla cuál es el máximo error que se desea. De acuerdo con nuestros experimentos, este método no siempre da buenos resultados, sobre todo cuando el contorno que se desea aproximar posee regiones donde los detalles son muy importantes, mientras que en otras la curva es muy suave (como sucede por ejemplo con el contorno de una cabeza humana). En tales casos, cualquier tipo de control global del error falla, pues conduce a que se pierdan los detalles o por el contrario a que se incluyan demasiados puntos en la aproximación poligonal.

De acuerdo con el análisis previo, podemos concluir que los nodos de la curva spline se pueden tomar como los vértices de una poligonal que aproxime el contorno de la imagen. El método para determinar esta poligonal debe permitirnos controlar el máximo error permitido. Según nuestra experiencia, el método propuesto en [Oden93] es el que da mejores resultados, si se selecciona adecuadamente un ancho de máscara que ignore el ruido de los datos, pero preserve los detalles importantes del contorno. Por ejemplo, en la reconstrucción de los contornos de la cabeza humana que se muestra en la sección de ejemplos, los nodos se seleccionaron utilizando una máscara de ancho 5 píxeles para todas las secciones transversales, con excepción de las que representan las orejas, en las cuales el ancho de máscara adecuado resultó ser de 3 píxeles. Por otra parte, nuestros experimentos nos permiten afirmar que, desde el punto de vista de la calidad del ajuste obtenido, resulta innecesario y además extremadamente costoso, calcular para todas las sucesiones de nodos posibles el correspondiente spline cónico de ajuste y seleccionar entre todos ellos el "óptimo". Esto se puede evitar si para resolver el problema de ajuste (3.4) se utiliza una buena aproximación de la distancia Euclideana

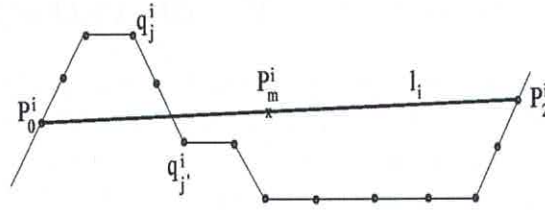


Figura 3.3: El punto  $q_j^i$  es el nuevo nodo y punto de inflexión de la curva A-spline.

de un punto a un cónica.

Como las cónicas no poseen puntos de inflexión, cualquier punto de inflexión del A-spline tiene que ser un nodo. En consecuencia, adicionalmente al método escogido para seleccionar los nodos del spline, es necesario disponer de un algoritmo, que a partir de los puntos que describen el contorno de la imagen, sea capaz de escoger aquellos que serán puntos de inflexión de la curva A-spline. Este algoritmo debe ser muy económico desde el punto de vista computacional, pues la cantidad de puntos que describen el contorno y que en consecuencia se deben analizar es muy grande.

Supongamos que  $P_0^i$  y  $P_2^i$  son vértices consecutivos de la poligonal que aproxima el contorno de la imagen digital. Entonces vamos a introducir un nuevo vértice entre ellos, si la cantidad de puntos que se encuentran del mismo lado de la recta  $l_i : l_i(x, y) = 0$ , que pasa por los puntos  $P_0^i$  y  $P_2^i$ , es menor o igual que el 80% de los datos contenidos entre ambos puntos. Más precisamente, sean  $q_j^i, j = 1, \dots, n_i$  los puntos del contorno situados entre  $P_0^i$  y  $P_2^i$  y denotemos por

$$\Gamma_i = \{q_j^i \mid l_i(q_j^i)l_i(q_{j+1}^i) < 0 \text{ ó } l_i(q_j^i)l_i(q_{j-1}^i) < 0\}$$

el conjunto de los puntos donde cambia el signo de la evaluación en la ecuación de la recta  $l_i$ . Entonces, tomaremos como punto de inflexión de la curva A-spline (y en consecuencia como nodo) el punto  $q_j^i$ , tal que,

$$\|q_j^i - P_m^i\|_2 = \min\{\|q_j^i - P_m^i\|_2, \quad q_j^i \in \Gamma_i\}$$

donde  $P_m^i$  es el punto medio del segmento  $P_0^i P_2^i$  (vea la figura 3.3).

En resumen, vamos a tomar como nodos de la curva A-spline  $S$  los puntos obtenidos por el algoritmo descrito en [Oden93] más los puntos seleccionados como puntos de inflexión de la curva.

### 3.4 Definición de la tangente en cada nodo

Una vez que hemos determinado los puntos de ruptura o nodos del A-spline, necesitamos calcular la recta tangente en cada nodo, pues los vértices del  $i$ -ésimo triángulo del polígono de control son dos nodos consecutivos  $P_0^i$  y  $P_2^i$  y el punto de intersección  $P_1^i$ , de las tangentes definidas en cada nodo.

Para determinar la tangente que vamos a prefijar en cada nodo, Bajaj *et al* proponen en [Baj94] calcular la parábola que interpola cada nodo y sus vecinos más próximos por la izquierda y por la derecha. En tal caso se asigna como tangente en el nodo, la tangente a la parábola en ese punto. Adicionalmente, nosotros probamos también la posibilidad de tomar como tangente, la recta tangente al polinomio de grado 4 que interpola el nodo y sus dos vecinos más próximos a cada lado. Ninguno de estos dos métodos da buenos resultados cuando los puntos provienen de una imagen digital. Esto se debe a que tales datos poseen mucho ruido y en consecuencia las aproximaciones de la tangente que se obtienen a partir de considerar unos pocos vecinos del nodo en cuestión, resultan muy imprecisas, como se muestra en la figura 3.4. Por otro lado, es bien conocido que la diferenciación numérica basada en el uso de los polinomios de interpolación, es un proceso inestable [Con80], de modo que aún cuando los datos iniciales fuesen precisos, los valores aproximados de las derivadas no suelen ser igualmente precisos.

Por todo lo anterior, vamos a proponer a continuación otro método para determinar la tangente en cada nodo. Recordemos que  $q_j^i, j = 1, \dots, n_i$  son los puntos del contorno situados en el  $i$ -ésimo triángulo del polígono de control del A-spline. Sea  $\Lambda_i = \{q_{\lfloor n_i/2 \rfloor + 1}^{i-1}, \dots, q_{n_i-1}^{i-1} = q_1^i, \dots, q_{\lfloor n_i/2 \rfloor}^i\}$  el conjunto de los "vecinos" de  $P_0^i$ . Para simplificar la notación, vamos a denotar por  $p_j^i = (x_j^i, y_j^i)$  con  $j = 1, \dots, m_i$ , los puntos del conjunto  $\Lambda_i$ . Supongamos ahora que cada punto  $p_j^i$  está asociado al parámetro  $t_j^i$  que representa la longitud acumulativa de la cuerda,

$$t_1^i = 0$$

$$t_{j+1}^i = t_j^i + \|p_{j+1}^i - p_j^i\|_2, \quad j = 1, \dots, m_i - 1$$

Nótese que  $P_0^i = p_{j^*}^i = (x_{j^*}^i, y_{j^*}^i)$  con  $j^* = \lfloor \frac{n_i-1}{2} \rfloor$ . Entonces nos interesa calcular la recta paramétrica  $r(t) = (x(t), y(t))$ ,

$$x(t) = a_1(t - t_{j^*}^i) + x_{j^*}^i$$

$$y(t) = a_2(t - t_{j^*}^i) + y_{j^*}^i$$

que pasa por el punto  $P_0^i$  y que mejor ajusta, en el sentido mínimo cuadrado, los puntos del conjunto  $\Lambda_i$ . Más precisamente, los parámetros libres  $a_1$  y  $a_2$  de la recta  $r(t)$  se calculan resolviendo los siguientes problemas de mínimos cuadrados lineales,

$$\min_{a_1} \sum_{j=1}^{m_i} w_j^2 (x_{j^*}^i - x(t_j^i))^2 \quad (3.1)$$

$$\min_{a_2} \sum_{j=1}^{m_i} w_j^2 (y_{j^*}^i - y(t_j^i))^2 \quad (3.2)$$

donde los pesos  $w_j$  están dados por  $w_j^2 = 1/\|p_j^i - p_{j^*}^i\|_2$   $j = 1, \dots, m_i$ ,  $j \neq j^*$ ,  $w_{j^*} = 1$ .

Obsérvese que los pesos seleccionados son mayores para los puntos más cercanos al nodo, de modo que la recta de ajuste no sólo está obligada a pasar por el nodo, sino que también ajusta mejor los puntos más próximos a éste. Es fácil verificar que la solución de los problemas (3.1) y (3.2) es,

$$a_1^* = \frac{\sum_{j=1}^{m_i} w_j^2 (x_j^i - x_{j^*}^i)(t_j^i - t_{j^*}^i)}{\sum_{j=1}^{m_i} w_j^2 (t_j^i - t_{j^*}^i)^2} \quad (3.3)$$

$$a_2^* = \frac{\sum_{j=1}^{m_i} w_j^2 (y_j^i - y_{j^*}^i)(t_j^i - t_{j^*}^i)}{\sum_{j=1}^{m_i} w_j^2 (t_j^i - t_{j^*}^i)^2}$$

respectivamente. Por lo tanto, vamos a tomar  $[a_1^*, a_2^*]$  como vector tangente en el nodo  $P_0^i$ .

En la figura 3.4 se muestra la recta tangente en un nodo, obtenida utilizando los diferentes métodos que hemos mencionado. Como se puede apreciar, los datos (marcados con  $\cdot$ ) tienen forma de "escalera" y las aproximaciones de la tangente que se basan en el uso de algún polinomio de interpolación, no reflejan el comportamiento de los datos en una vecindad del nodo. Lo contrario sucede con la recta que pasa por el nodo y ajusta a sus vecinos en el sentido mínimo cuadrado.

Finalmente, es importante destacar que si  $P_0^i$  es un punto de inflexión, entonces en ocasiones es necesario corregir ligeramente la dirección de la tangente que hemos propuesto, con el objetivo de garantizar que la mayor cantidad posible de los puntos del contorno estén incluidos en algún triángulo. En ese caso vamos a tomar como tangente en el nodo, la recta que pasa por

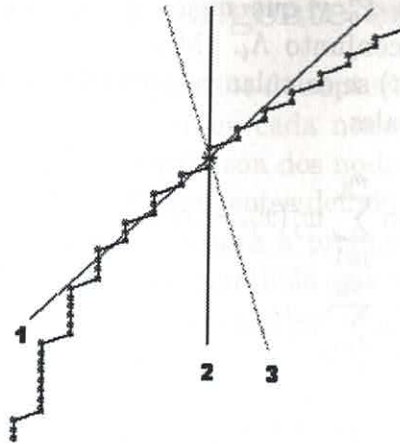


Figura 3.4: Tangente obtenida usando: 1. Recta de ajuste, 2. Parábola de interpolación, 3. Polinomio de interpolación de grado 4

el punto  $P_0^i$  y por un punto  $q_j$  con  $\hat{j} \neq j^*$ , que se encuentra entre  $P_0^{i-1}$  y  $P_0^{i+1}$ . A su vez (vea la figura 3.5), el punto  $q_j$  se selecciona maximizando el ángulo entre los segmentos,

- $\overline{P_0^i q_j^{i-1}}$  y  $\overline{P_0^{i-1} P_0^i}$ , para  $q_j^{i-1}$  a la "izquierda" de  $P_0^i$ .
- $\overline{P_0^i q_j^i}$  y  $\overline{P_0^{i+1} P_0^i}$ , para  $q_j^i$  a la "derecha" de  $P_0^i$ .

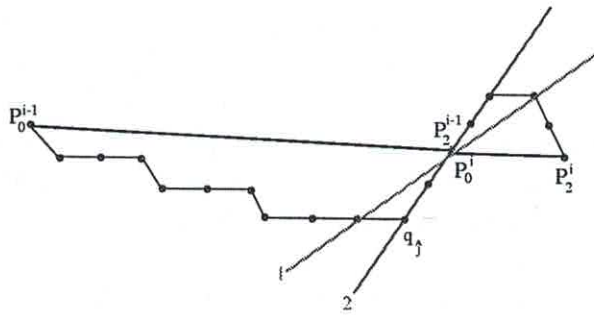


Figura 3.5: Corrección de la tangente en un punto de inflexión: 1. Recta de ajuste, 2. Recta tangente corregida



### 3.5 Ajuste de datos por medio de una cónica

Una vez que se ha construido el polígono de control, nos resta por determinar cada segmento de curva  $c_i$  que definirá el A-spline en el interior del triángulo  $T_i$  para  $i = 1, \dots, n$ . Como estamos interesados en aproximar el contorno de una imagen, la curva  $c_i$  se determinará exigiendo que ajuste los puntos del contorno que están contenidos en el interior del triángulo  $T_i$ .

#### 3.5.1 El problema general

En la literatura se puede encontrar una amplia variedad de métodos para ajustar datos por medio de una cónica. Entre ellos podemos citar [Book79], [Baj99a)], [Sam82], [Gan94], [Pav83]. Dado un conjunto de  $n$  puntos  $q_j$  en el plano, la técnica clásica de ajuste consiste en determinar una cónica  $C$  que minimice la distancia media cuadrática, es decir una solución del problema,

$$\min\left\{\frac{1}{n} \sum_{j=1}^n d^2(q_j, C)\right\} \quad (3.4)$$

donde  $d(q_j, C)$  es la distancia Euclideana de  $q_j$  a  $C$ .

Desafortunadamente, no es posible dar una expresión explícita para la distancia Euclideana de un punto a una cónica. Por eso, para resolver el problema (3.4) es necesario sustituir  $d(q_j, C)$  por una aproximación de la distancia Euclideana del punto  $q_j$  a la cónica  $C$ .

Sea  $f(x, y)$  un polinomio de grado 2 y supongamos que,

$$f(x, y) = 0 \quad (3.5)$$

es la ecuación implícita de la cónica  $C$ .

Entonces la aproximación más común para la distancia Euclideana de  $q_j$  a  $C$  es la llamada *distancia algebraica* dada por  $|f(q_j)|$ . En términos de la misma, el problema de hallar la cónica que mejor ajusta un conjunto de datos, se convierte en,

$$\min\left\{\frac{1}{n} \sum_{j=1}^n f^2(q_j)\right\} \quad (3.6)$$

Esta última formulación del problema es muy popular, pues la solución de (3.6) se traduce en resolver un sistema de ecuaciones lineales homogéneo en los coeficientes de la cónica. Sin embargo, para evitar que la solución de

este sistema sea la trivial, es necesario introducir una restricción adicional al problema de mínimos cuadrados. Algunos autores han propuesto restricciones lineales o cuadráticas en los coeficientes. Por otra parte, dependiendo del tipo de restricciones que se impongan, el método de ajuste puede tener propiedades especiales. Por ejemplo, los algoritmos propuestos por Taubin en [Tau91] y Bookstein en [Book79] son invariantes por transformaciones afines. Pero es necesario tener en cuenta que cualquier restricción adicional que se imponga introduce un sesgo en la selección de la cónica de ajuste, eliminando posibles candidatos a la solución óptima. Como veremos en la próxima sección, este tipo de problema no se presenta cuando se trata de ajustar una cónica en el interior de cada triángulo, pues en tal caso se puede reescribir la cónica, de modo que la solución trivial no sea una solución factible del problema.

En la práctica, la distancia algebraica sólo se puede considerar como una buena aproximación de la distancia Euclideana de un punto a una cónica, si el punto en cuestión se encuentra muy próximo a la curva. Esto nos explica por qué algunos autores han propuesto mejores aproximaciones, las cuales se pueden calcular mediante una fórmula cerrada [Tau94a]) o por medio de un proceso iterativo [Pon92],[Her97], aunque sea al precio de perder la linealidad del correspondiente problema de mínimos cuadrados. En general, está claro que mientras más precisas son las aproximaciones a la distancia Euclideana, resulta más caro computacionalmente calcular la mejor curva de ajuste. Pero utilizar aproximaciones muy baratas, puede conducir a que el ajuste obtenido sea pobre. De modo que se trata de buscar un buen compromiso entre ambos aspectos. En ese sentido, nosotros proponemos utilizar como aproximación de la distancia Euclideana la fórmula dada por Taubin en [Tau94a)],

$$d(q_j, C) = \frac{|f(q_j)|}{\|\nabla f(q_j)\|} \quad (3.7)$$

Finalmente, para concluir esta sección vamos a mencionar la variante paramétrica de solución del problema de ajuste de cónicas [Spa96], [Gan94]. Como toda cónica se puede representar en forma paramétrica, el problema de mínimos cuadrados (3.4) se puede resolver asociando a cada punto con un valor del parámetro. Sin embargo, en tal caso el problema es de tipo no lineal y su dimensión crece con la cantidad de puntos.

### 3.5.2 Ajuste de una cónica en el interior de cada triángulo

Recordemos que como cada una de las curvas que componen un A-spline está definida en el interior de un triángulo, resulta conveniente introducir el sistema de coordenadas baricéntricas que sólo depende de los vértices del triángulo. En las nuevas coordenadas, la expresión implícita de una sección  $c_i$  del A-spline es del mismo tipo que la de cualquier otro segmento  $c_j$  y sólo se diferencian en los coeficientes. En particular, tomando  $n = 2$  en (2.7) resulta que cualquier cónica  $c_i$  se puede expresar en coordenadas baricéntricas y en términos de los polinomios de Bernstein como,

$$\sum_{j=0}^2 \sum_{k=0}^{2-j} a_{kj}^i u^k v^j (1-u-v)^{2-k-j} = 0 \quad (3.8)$$

donde hemos incluido en los coeficientes  $a_{kj}^i$  el factor  $\frac{2!}{k!j!(2-k-j)!}$  de (2.7).

**Lema 3.1** *Toda cónica  $c_i$  de la familia,*

$$c_i : g_i(u, v) = a_{02}^i v^2 + a_{10}^i u(1-u-v) = 0 \quad (3.9)$$

que se obtiene tomando  $a_{00}^i = a_{01}^i = a_{20}^i = a_{11}^i = 0$  en (3.8) pasa por los puntos  $P_0^i$  y  $P_2^i$  y es tangente en esos puntos a los lados correspondientes del triángulo  $T_i$ . Si adicionalmente  $a_{02}^i a_{10}^i < 0$ , entonces existe un segmento conexo de curva contenido en el triángulo canónico que satisface las condiciones de interpolación previas.

#### Demostración

La demostración de este resultado es una consecuencia inmediata de los Lemas 2.1 y 2.2 del Capítulo 2. ■

Como  $a_{02}^i \neq 0$  entonces podemos multiplicar la ecuación (3.9) por  $-1/a_{02}^i$ , de modo que la familia de cónicas (3.9) se puede reescribir como,

$$c_i : g_i(u, v) = -v^2 + w_1^i u(1-u-v) = 0 \quad (3.10)$$

donde hemos denotado por  $w_1^i$  el parámetro libre  $-a_{10}^i/a_{02}^i$ .

Sean  $(u_j^i, v_j^i)$  las coordenadas baricéntricas del punto  $q_j^i$  contenido en el triángulo  $T_i$  para  $i = 1, \dots, n_i$ . Si aproximamos la distancia Euclideana de  $q_j^i$  a una cónica por la distancia algebraica, entonces el problema de mínimos cuadrados correspondiente está dado por,

$$\min_{w_1^i} \left\{ G(w_1^i) := \frac{1}{n_i} \sum_{j=1}^{n_i} g_i^2(u_j^i, v_j^i) \right\} \quad (3.11)$$

La solución de este problema conduce a una ecuación lineal en  $(w_1^i)^2$  de donde resulta que,

$$(w_1^i)^2 = \frac{\sum_{j=1}^{n_i} (v_j^i)^2 u_j^i (1 - u_j^i - v_j^i)}{4 \sum_{j=1}^{n_i} (u_j^i)^2 (1 - u_j^i - v_j^i)^2} \quad (3.12)$$

En la práctica, si los puntos  $q_j^i$  no están muy próximos a una cónica, o si están muy irregularmente distribuidos, entonces el parámetro óptimo obtenido mediante (3.12) da lugar a una cónica que está lejos de la cónica que mejor ajusta los datos en la distancia Euclideana. Este problema se presenta con relativa frecuencia y por eso resulta muy importante seleccionar una buena aproximación de la distancia euclideana, que por otra parte sea poco costosa de calcular. Una aplicación interesante de este tipo puede encontrarse en [Gue99], donde se utiliza una cónica para ajustar un conjunto de puntos pertenecientes a una curva en forma de "L". Tal curva se usa para seleccionar el parámetro de regularización, en la solución de problemas discretos mal planteados.

En el caso de datos provenientes de una imagen digital, que con frecuencia tienen ruido y gran variabilidad, nosotros recomendamos utilizar la aproximación de la distancia Euclideana dada por (3.7). Si  $f_i(x, y) = 0$  es la ecuación implícita en coordenadas reales de la curva  $c_i$ , entonces en términos de la aproximación (3.7) de la distancia Euclideana, el problema de mínimos cuadrados a resolver sería,

$$\min_{w_1^i} \left\{ G(w_1^i) := \frac{1}{n_i} \sum_{j=1}^{n_i} \frac{f_i^2(x_j^i, y_j^i)}{\|\nabla f_i(x_j^i, y_j^i)\|^2} \right\} \quad (3.13)$$

Nótese que como

$$f_i(x_j^i, y_j^i) = g_i(u_j^i, v_j^i) \text{ y } \nabla f_i(x_j, y_j) = \left( \frac{\partial g_i}{\partial u} \frac{\partial u}{\partial x} + \frac{\partial g_i}{\partial v} \frac{\partial v}{\partial x}, \frac{\partial g_i}{\partial u} \frac{\partial u}{\partial y} + \frac{\partial g_i}{\partial v} \frac{\partial v}{\partial y} \right) (u_j^i, v_j^i)$$

entonces la función  $G(w_1^i)$  se puede escribir en términos del parámetro libre  $w_1^i$  y de las coordenadas baricéntricas  $(u_j^i, v_j^i)$  de los puntos  $q_j^i$ . En este caso, el problema de mínimos cuadrados es no lineal en  $w_1^i$ , con la restricción  $w_1^i > 0$ . Pero como (3.7) es una mejor aproximación de la distancia Euclideana de un punto a una cónica que la distancia algebraica, entonces la

cónica de ajuste correspondiente aproxima mejor el contorno de la imagen digital.

### 3.6 Experimentación numérica

El algoritmo descrito en este capítulo se probó aproximando los contornos de imágenes de resonancia magnética nuclear, obtenidas a partir de la base de datos de la Universidad de North Carolina, ftp.cs.unc.edu. Las imágenes corresponden a las secciones transversales de una cabeza humana. La figura 3.6 muestra el gráfico de los A-splines cónicos que aproximan el contorno de 25 imágenes y es la salida de un programa MATLAB (versión 5.3) que calcula y grafica cada una de las curvas A-spline. Para graficar cada sección cónica se utilizó la parametrización de Bernstein-Bézier de las curvas racionales [Far93]. Como se señaló previamente, para determinar los nodos del A-spline se utilizó para todos los contornos, con excepción de los que describen las orejas, una máscara de ancho 5 píxeles. En el caso de las orejas el ancho de máscara utilizado fue de 3 píxeles. Cada una de las cónicas que ajusta los datos en el interior de un triángulo se calculó resolviendo el problema de mínimos cuadrados no lineal (3.13), aproximando la distancia Euclídeana mediante (3.7).

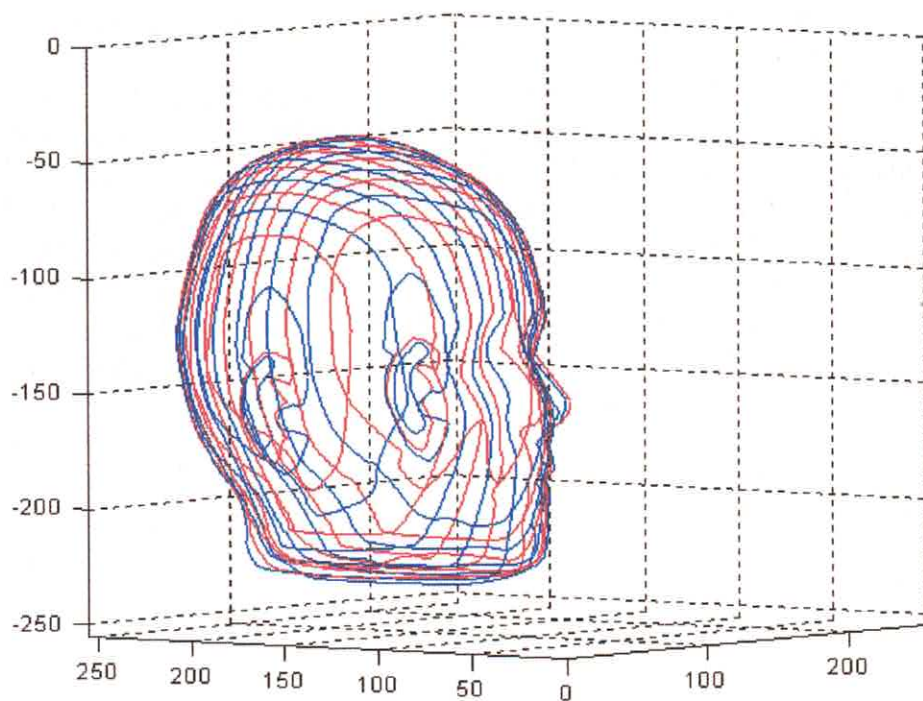


Figura 3.6: Aproximación de las secciones transversales de una cabeza humana por medio de A-splines cónicos

## Capítulo 4

# Diseño libre con curvas A-spline cúbicas

### 4.1 Introducción

Las curvas A-spline no sólo se pueden utilizar para ajustar datos, también suelen ser un instrumento muy efectivo cuando se trata de diseñar libremente el contorno de algún objeto plano, o la curva directriz de una superficie de revolución. En este tipo de diseño, frecuentemente se tiene un conjunto de puntos “claves” para dibujar el contorno y se desea construir una curva “suave” que pase por los esos puntos, o sea que satisfaga ciertas condiciones de interpolación. En otros casos, los puntos fijos, no son exactamente datos del problema, pero le sirven al diseñador como guía o “esqueleto” de la curva que se desea obtener.

En una buena parte de los problemas, basta con exigir que la tangente a la curva spline varíe continuamente a lo largo de ella, es decir que la curva sea  $G^1$  continua. Sin embargo, en algunos problemas prácticos es necesario exigir adicionalmente que la curva obtenida sea  $G^2$  continua. Esto sucede por ejemplo, cuando se trata de diseñar la curva generatriz de las superficies de revolución que modelan las llamadas estructuras de cáscara [Her73]. Estas son estructuras tridimensionales cuyo espesor se puede considerar despreciable, tales como carrocerías, cimentaciones laminares de revolución o depósitos laminares elevados. Para evitar la ruptura de tales estructuras debido a tensiones internas que se producen por desbalance de ciertas fuerzas, es imprescindible exigir que la curva generatriz posea adicionalmente cur-

vatura continua en todos los puntos, es decir sea  $G^2$  continua. Por otra parte, muchas veces es necesario que la curva a modelar posea puntos de inflexión y en consecuencia puntos donde la curvatura sea cero. Esto no puede lograrse si se emplea un spline cónico, pues ninguna cónica no degenerada posee puntos con curvatura nula. Por todo lo anterior, en este capítulo vamos a utilizar una familia de A-splines cúbicos  $G^2$  continuos para el diseño libre.

En el primer capítulo mencionamos que el conjunto de las curvas algebraicas de grado mayor o igual que 3 es estrictamente mayor que el conjunto de las curvas racionales paramétricas del mismo grado. En particular, las curvas algebraicas cúbicas poseen 1 grado de libertad más, el cual se puede utilizar para un mejor control de su geometría.

Históricamente, las primeras soluciones al problema de interpolar un conjunto de puntos en el plano, con direcciones tangentes prefijadas, se basaron en el uso de splines racionales paramétricos. En efecto, Farin demostró que es posible construir un spline cúbico de Bézier  $G^2$  continuo, que satisfaga todas las condiciones de interpolación impuestas. Sin embargo, ningún nodo de este tipo de curva puede ser un punto de inflexión, de modo que el diseñador no tiene control de la posición de los puntos de inflexión. Por otra parte, una de las estrategias usuales para controlar la forma de la curva, es permitir que el diseñador escoja libremente la posición de un punto adicional de interpolación, en el interior de cada sección de la curva spline. Si esta condición se le impone al spline cúbico racional de Bézier ( $G^2$  continuo), entonces todos los parámetros libres quedan determinados de forma única, por lo cual resulta imposible tener un mayor control de la geometría de la curva.

Otra solución al problema planteado, que se basa en el uso de splines paramétricos, es el llamado spline cúbico directo [Far93]. Este tipo de curvas splines  $G^2$  continuas, interpola los puntos y tangentes prefijadas, pero no posee grados de libertad adicionales que puedan utilizarse como asas geométricas. Adicionalmente de Boor et al [Boor87] y Farin [Far93] han construido un spline cúbico de Bézier  $G^2$  continuo que interpola un conjunto de puntos y tangentes dadas y adicionalmente valores prefijados de la curvatura en esos puntos. En este caso no es posible interpolar además un punto en el interior de cada segmento, ni tampoco se dispone de grados de libertad adicionales que nos permitan modificar la forma de algún segmento de la curva. Por otro lado, para un problema de diseño libre, resulta poco natural suponer que el diseñador puede dar valores para la curvatura en los nodos.

El uso de las curvas A-splines para resolver problemas de interpolación es mucho más reciente, vea [Baj99a)] y las referencias de este trabajo. Como



mencionamos anteriormente, esto se puede explicar debido a dos factores fundamentales. En primer lugar, porque las curvas algebraicas pueden poseer varias ramas y singularidades, lo cual es poco deseable cuando se trata de utilizarlas para el diseño. Por otra parte, durante mucho tiempo no se disponía de algoritmos eficientes para graficarlas. Afortunadamente, hoy en día muchas de estas dificultades han sido superadas y esto ha contribuido a que muchos autores comiencen a valorar las ventajas de las curvas algebraicas y la posibilidad de utilizarlas para resolver problemas que usualmente sólo se atacaban con curvas paramétricas. Por ejemplo, en varios artículos pioneros Bajaj [Baj94], Patterson y Paluszny [Pal93],[Pal94] mostraron cómo construir una curva spline “pegando adecuadamente” segmentos de curvas de una familia de curvas algebraicas cúbicas y no singulares, que además sólo poseen una rama en la región de interés. Además, en esos trabajos se proponen algoritmos robustos y eficientes para graficar curvas cúbicas implícitas.

Un antecedente importante de las curvas que presentamos en este capítulo, es sin lugar a dudas la familia de A-splines cúbicos  $G^2$  continua introducida por Patterson y Paluszny en [Pal98]. En este trabajo, sin embargo, los datos a interpolar no son totalmente arbitrarios, ya que cada pareja de puntos y direcciones tangentes impone ciertas restricciones a la posición de los datos siguientes. En un artículo posterior y muy reciente [Pal99] los mismos autores muestran cómo contruir una familia más poderosa de curvas A-spline cúbicas  $G^2$  continuas, que no tiene las limitaciones anteriormente citadas. En este caso, los parámetros libres se utilizan directamente como asas geométricas para controlar la forma de la curva. Adicionalmente, las curvas de esta familia son capaces de interpolar valores prefijados de la curvatura en los nodos, un punto en el interior de cada segmento y la tangente a la curva en ese punto.

La familia de curvas A-spline cúbicas  $G^2$  continuas que se presenta en este capítulo, no es tan general como la introducida en [Pal99], que permite adicionalmente prefijar los valores de la curvatura en los nodos y la tangente en el punto de interpolación situado en el interior de cada sección. Sin embargo, nuestra familia dispone de un conjunto de asas geométricas que permiten controlar con mucha efectividad la simetría de cada segmento de la curva, además que hacen factible introducir deformaciones en una sección, afectando exclusivamente a los segmentos vecinos, de modo que el resto de la curva conserva su forma. Por otra parte, para muchos problemas prácticos lo más importante es que la curva resultante sea  $G^2$  continua y no cuál es valor específico de la curvatura en los puntos de interpolación, más aún si se tiene en cuenta que variaciones relativamente grandes en los valores prefijados de

la curvatura en los nodos, no se traducen en modificaciones apreciables en la forma de la sección de curva correspondiente.

## 4.2 A-spline cúbico default

### 4.2.1 Consideraciones preliminares

El problema que queremos resolver en este capítulo es el siguiente. Dado un conjunto de  $n$  puntos en cada uno de los cuales se ha prescrito una dirección tangente, queremos construir una curva A-spline cúbica que pase por los puntos con las tangentes prefijadas. En lo adelante nos referiremos a estas condiciones diciendo que el A-spline interpola los puntos y tangentes.

Para construir el A-spline  $S$  vamos a comenzar definiendo un polígono asociado al mismo, al cual llamaremos **polígono de control**, tomando prestada la terminología que se utiliza para este fin en el caso de las curvas de Bézier [Far93]. El polígono está formado por dos de los lados de una serie de triángulos consecutivos, donde cada triángulo posee un vértice en común con cada uno de sus dos vecinos. Más precisamente, denotemos por  $P_0^i, P_1^i, P_2^i$  los vértices del  $i$ -ésimo triángulo  $T_i$ . Los puntos  $P_0^i$  y  $P_2^i$  son datos consecutivos del problema, mientras que  $P_1^i$  es el punto de intersección de las tangentes prefijadas en  $P_0^i$  y  $P_2^i$ .

Denotemos por  $c_i$  la  $i$ -ésima sección cúbica de  $S$ , que se definirá en el interior del triángulo  $T_i$ , exigiendo que interpole los puntos  $P_0^i$  y  $P_2^i$  y las direcciones tangentes prefijadas en esos puntos. Para garantizar que  $S$  sea una curva continua debemos exigir entonces que  $P_2^i = P_0^{i+1}$  para  $i = 1, \dots, n-1$ . Si quisiéramos construir una curva cerrada debíamos exigir adicionalmente que  $P_2^n = P_0^1$ .

Por otro lado, está claro que para que la tangente varíe continuamente a lo largo de toda la curva, es decir para que  $S$  sea  $G^1$  continua, es necesario que los puntos  $P_1^i, P_2^i, P_1^{i+1}$  sean colineales para  $i = 1, \dots, n$ , (asumiendo que  $P_1^{n+1} = P_1^1$  en el caso de curvas cerradas).

A continuación damos una breve descripción del algoritmo ASPLINT2D, que desarrollamos en el resto del capítulo, y que nos permite construir una curva A-spline  $G^2$  continua que interpola un conjunto de puntos en el plano y las correspondientes direcciones tangentes.

**Algoritmo ASPLINT2D**

- 1) Construir el polígono de control de la curva A-spline a partir de los puntos de interpolación y las direcciones tangentes prefijadas en esos puntos.
- 2) Calcular y graficar el A-spline cúbico default que interpola el baricentro de cada triángulo definido por el polígono de control.
- 3) Mientras el diseñador desee:
  - 3.1) Seleccionar un triángulo  $i$ .
  - 3.2) Introducir la modificación deseada en la  $i$ -ésima sección del A-spline.
  - 3.3) Recalcular y graficar los segmentos  $i-1$ ,  $i$  e  $i+1$  del A-spline.

Analizando cada uno de los pasos del algoritmo ASPLFIT2D, es fácil probar que el número máximo de operaciones que requiere este método es  $O(n) + O(m)$ , donde  $n$  representa la cantidad de puntos de interpolación y  $m$  el número de modificaciones que realiza el diseñador a partir del A-spline default que interpola los baricentros de cada triángulo.

**4.2.2 Una familia de cúbicas convexas, conexas y no singulares**

En el capítulo 2 mencionamos que como cada curva que compone un A-spline está definida en el interior de un triángulo, resulta conveniente introducir un nuevo sistema de coordenadas que sólo depende de los vértices del triángulo: las coordenadas baricéntricas. Esto tiene la ventaja de que la expresión implícita de una sección  $c_i$  del A-spline es lo más sencilla posible y es del mismo tipo que la de cualquier otro segmento  $c_j$  y sólo se diferencian en los coeficientes. En particular, tomando  $n = 3$  en (2.7) obtenemos que toda cúbica  $c_i$  se puede expresar en coordenadas baricéntricas y en términos de los polinomios de Bernstein como,

$$\sum_{j=0}^3 \sum_{k=0}^{3-j} a_{kj}^i u^k v^j (1-u-v)^{3-k-j} = 0 \quad (4.1)$$

donde hemos incluido en los coeficientes  $a_{kj}^i$  el factor  $\frac{3!}{k!j!(3-k-j)!}$  de (2.7).

En los trabajos de M. Paluszny y R. Patterson [Pal92], [Pal93] y en el trabajo posterior de C. Bajaj [Baj99a)], se muestra que es posible construir una familia de curvas A-spline cúbicas que satisfacen las condiciones que se enumeran en el siguiente lema.

**Lema 4.1** *Toda curva cúbica  $c_i$  de la familia,*

$$c_i : f_i(u, v) = -v^3 + a_{10}^i u(1-u-v)^2 + a_{20}^i u^2(1-u-v) + a_{02}^i v^2(1-u-v) + a_{11}^i uv(1-u-v) + a_{12}^i uv^2 = 0 \quad (4.2)$$

que se obtiene tomando  $a_{00}^i = a_{30}^i = a_{01}^i = a_{21}^i = 0$  y  $a_{03}^i = -1$  en (4.1), pasa por los puntos  $P_0^i$  y  $P_2^i$  y es tangente en esos puntos a los lados correspondientes del triángulo  $T_i$ . Adicionalmente, si  $a_{10}^i, a_{20}^i \geq 0$  y ambos no son simultáneamente cero y  $a_{02}^i, a_{12}^i \leq 0$  y tampoco se anulan ambos simultáneamente (vea figura 2.3), entonces  $c_i$  es una curva convexa, conexa y no singular en el interior del triángulo  $T_i$ .

La demostración de este lema, que presentamos en el anexo B, se basa en la demostración del lema 2.1 para curvas algebraicas de grado  $n$ , con  $n \geq 2$ , que se puede encontrar en [Baj99a)] y también en el Anexo B.

Supongamos que los coeficientes del A-spline se escogen de manera que se satisfacen las hipótesis del Lema 4.1. Entonces para obtener una curva  $G^2$  continua basta con exigir que la curvatura del A-spline sea continua en los puntos donde se unen dos secciones consecutivas, es decir en los vértices  $P_0^i$  y  $P_2^i$  de cada triángulo  $T_i$ ,  $i = 1, \dots, n$ .

**Lema 4.2** [Pal92], [Baj99a)] *Sea  $k_0^i$  la curvatura de  $c_i$  en el punto  $P_0^i$  y  $k_2^i$  la curvatura en  $P_2^i$ . Entonces,*

$$k_0^i = \frac{|a_{12}^i|}{|a_{20}^i|} \frac{\Delta_i}{(g_0^i)^3} \quad k_2^i = \frac{|a_{02}^i|}{|a_{10}^i|} \frac{\Delta_i}{(g_2^i)^3} \quad (4.3)$$

donde

$$g_0^i = \|P_0^i - P_1^i\|, \quad g_2^i = \|P_2^i - P_1^i\| \quad y \quad \Delta_i = \left| \det \begin{pmatrix} P_0^i & P_1^i & P_2^i \\ 1 & 1 & 1 \end{pmatrix} \right|$$

es el área de  $T_i$ .

La demostración completa de este resultado se puede encontrar [Pal92]. En el Anexo B se pueden encontrar también las ideas fundamentales de la demostración, sin incluir todos los detalles.

A partir de la expresión (4.3) obtenida en el Lema 4.2 podemos concluir que la condición de  $G^2$  continuidad del A-spline,

$$k_0^{i+1} = k_2^i \quad i = 1, \dots, n-1$$

es equivalente a exigir que los coeficientes  $a_{kj}^i$  satisfagan el sistema de ecuaciones no lineales,

$$\frac{|a_{12}^{i+1}|}{|a_{20}^{i+1}|} \frac{\Delta_{i+1}}{(g_0^{i+1})^3} = \frac{|a_{02}^i|}{|a_{10}^i|} \frac{\Delta_i}{(g_2^i)^3} \quad i = 1, \dots, n-1 \quad (4.4)$$

En la siguiente sección veremos cómo es posible evitar la solución de este sistema no lineal, si se introduce una "base" conveniente para representar cada sección del A-spline.

### 4.2.3 Cálculo de una base para la $i$ -ésima sección

El objetivo de este epígrafe es buscar una nueva representación de cada sección  $c_i$  del A-spline, de modo que la condición de  $G^2$  continuidad (4.4) se transforme en una condición lineal en los coeficientes de la curva. Para obtener la representación deseada es necesario tener en cuenta que, desde el punto de vista geométrico, cada sección  $c_i$  del A-spline, definida en el triángulo  $T_i$  con vértices  $P_0^i P_1^i$  y  $P_2^i$ , puede tener uno de los siguientes comportamientos extremos:

- Pegarse al lado  $P_0^i P_1^i$ .
- Pegarse al lado  $P_1^i P_2^i$ .
- Pegarse simultáneamente a  $P_0^i P_1^i$  y  $P_1^i P_2^i$ .

La idea consiste entonces en generar 3 curvas cúbicas de la familia (4.2), escritas en coordenadas baricéntricas, cada una de las cuales corresponda a uno de los posibles comportamientos anteriores en el triángulo canónico. Una vez obtenidas estas curvas "extremas", cualquier otra curva cúbica de la misma familia, pudiera pensarse como una combinación convexa de las curvas "extremas".

Denotemos por  $C_0^i$  la cúbica de la familia (4.2) que se pegará al lado  $P_0^i P_1^i$ . Similarmente, sea  $C_2^i$  la cúbica que se aproximará al lado  $P_2^i P_1^i$  tanto como se desee y finalmente, denotemos por  $C_1^i$  la cúbica que se pega simultáneamente a  $P_0^i P_1^i$  y  $P_1^i P_2^i$ . Con el propósito de garantizar que en la curva  $c_i$ , no se anulen los coeficientes  $a_{10}^i$  y  $a_{20}^i$  que aparecen en el denominador de las expresiones (4.3) para la curvatura en los puntos extremos, vamos a incluir en las 3 curvas  $C_0^i$ ,  $C_1^i$  y  $C_2^i$ , los términos  $a_{10}^i u(1-u-v)^2$  y  $a_{20}^i u^2(1-u-v)$  y adicionalmente el término  $-v^3$ . De este modo garantizamos que cualquier curva  $c_i$  que se obtenga como combinación convexa de  $C_0^i$ ,  $C_1^i$  y  $C_2^i$  contenga los 3 términos mencionados.

Por otra parte, si queremos que la curva  $C_0^i$  se pegue  $P_0^i P_1^i$ , debemos exigir que su curvatura en el punto  $P_0^i$  se anule, lo cual equivale según (4.3) a pedir que  $a_{12}^i = 0$ . Luego, la curva  $C_0^i$  pertenece a la familia,

$$C_0^i : -v^3 + a_{10}^i u(1-u-v)^2 + a_{20}^i u^2(1-u-v) + a_{02}^i v^2(1-u-v) + b_{11}^i uv(1-u-v) = 0 \quad (4.5)$$

De manera similar, si queremos que la curva  $C_2^i$  se pegue al segmento  $P_2^i P_1^i$ , debemos exigir que su curvatura en el punto  $P_2^i$  sea cero, lo cual se logra tomando  $a_{02}^i = 0$ . Por lo tanto, la curva  $C_2^i$  es un miembro de la familia,

$$C_2^i : -v^3 + a_{10}^i u(1-u-v)^2 + a_{20}^i u^2(1-u-v) + a_{12}^i uv^2 + c_{11}^i uv(1-u-v) = 0 \quad (4.6)$$

Finalmente, para garantizar que la curva  $C_1^i$  se aproxime simultáneamente a  $P_0^i P_1^i$  y  $P_1^i P_2^i$ , basta con exigir que su curvatura en los dos puntos  $P_0^i$  y  $P_2^i$  se anule. Esto se logra si tomamos  $a_{02}^i = a_{12}^i = 0$ , de modo que la curva  $C_1^i$  pertenece a la familia,

$$C_1^i : -v^3 + a_{10}^i u(1-u-v)^2 + a_{20}^i u^2(1-u-v) + a_{11}^i uv(1-u-v) = 0 \quad (4.7)$$

Nótese que como el punto  $p_2$  es la imagen de  $P_0^i$  por el cambio de coordenadas reales a coordenadas baricéntricas, en estas últimas coordenadas la curva  $C_0^i$  se aproxima al segmento  $p_1 p_2$  y tiene curvatura cero en  $p_2$ . Recíprocamente, como  $p_0$  es la imagen de  $P_2^i$  al realizar el cambio de coordenadas reales a coordenadas baricéntricas, la curva  $C_2^i$ , escrita en coordenadas baricéntricas, se aproxima al segmento  $p_0 p_1$  y tiene curvatura cero en  $p_0$ . Esto se puede observar claramente en la figura 4.1

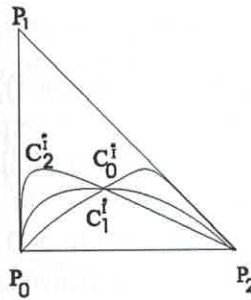


Figura 4.1: Cúbicas en el triángulo canónico

**Lema 4.3** Sean  $(u_i, v_i)$  las coordenadas baricéntricas de un punto  $Q_i$  situado en el interior del triángulo  $T_i$  y

$$a_{10}^i = \frac{v_i^3}{u_i(1-u_i-v_i)^2}, \quad a_{20}^i = \frac{v_i^3}{u_i^2(1-u_i-v_i)}$$

Entonces para todo  $K > 0$ ,

i) Una cúbica  $C_0^i$  de la familia (4.5) con,

$$a_{02}^i = -\frac{Kv_i^3}{u_i(1-u_i-v_i)^2}, \quad b_{11}^i = \frac{v_i^2(Kv_i^2 + u_iv_i + u_i^2 - u_i)}{u_i^2(1-u_i-v_i)^2}$$

pasa por el punto  $Q_i$  y es conexa, convexa y no singular en el interior del triángulo  $T_i$ .

ii) Una cúbica  $C_1^i$  de la familia (4.7) con,

$$a_{11}^i = -\frac{v_i^2}{u_i(1-u_i-v_i)}$$

interpola el punto  $Q_i$  y es conexa, convexa y no singular en el interior del triángulo  $T_i$ .

iii) Una cúbica  $C_2^i$  de la familia (4.6) con,

$$a_{12}^i = -\frac{Kv_i^3}{u_i^2}, \quad c_{11}^i = \frac{v_i^2(Kv_i^2 - u_i)}{u_i^2(1-u_i-v_i)}$$

pasa por el punto  $Q_i$  y es conexa, convexa y no singular en el interior del triángulo  $T_i$ .

### Demostración

Vamos a indicar solamente cómo se prueba i), pues los otros dos casos son muy similares. Sustituyendo en (4.5) las expresiones de  $a_{10}^i, a_{20}^i, a_{02}^i$  y  $b_{11}^i$  y las coordenadas  $(u_i, v_i)$  del punto  $Q_i$ , es fácil verificar que  $C_0^i(Q_i) = 0$ . Por otro lado, nótese que si  $Q_i$  está en el interior del triángulo  $T_i$ , entonces  $a_{10}^i, a_{20}^i > 0$ , mientras que  $a_{02}^i < 0$  para todo  $K > 0$  y  $a_{12}^i = 0$ . Luego del lema 4.1, podemos concluir que  $C_0^i$  es una curva convexa, conexa y no singular en el interior del triángulo  $T_i$ . ■

Es fácil probar que en la medida en que  $K$  crece la curva  $C_0^i$  tiende a la curva  $(1 - u - v)(u_i v - v_i u)v = 0$ , que es el producto de 3 rectas, la base y el lado  $p_1 p_2$  del triángulo canónico y la recta que pasa por  $p_0$  y por el punto de interpolación. En otras palabras, entre todas las curvas que satisfacen las condiciones de interpolación impuestas, la curva  $C_0^i$  se aproxima lo más posible al lado  $p_1 p_2$  del triángulo canónico, que corresponde al segmento  $P_0^i P_1^i$  del triángulo en las coordenadas reales. Similarmente, se puede verificar de forma inmediata que cuando  $K \rightarrow \infty$  la curva  $C_2^i$  converge a la curva  $uv(u_i v + v_i - v_i u - v) = 0$ , que es el producto de 3 rectas, la base y el lado  $p_0 p_1$  del triángulo canónico y la recta que pasa por  $p_2$  y por el punto de interpolación, es decir  $C_2^i$  se aproxima a la curva que más se pega al lado  $P_1^i P_2^i$ , entre todas las que satisfacen las condiciones de interpolación.

Sin embargo, para  $K$  suficientemente grande, las variaciones en su valor no se aprecian desde el punto de vista geométrico en la forma de la curva correspondiente. Por eso en la práctica se toma  $K = 100$  y abusando de la notación vamos a denotar otra vez las curvas correspondientes a este valor por  $C_0^i$  y  $C_2^i$ .

Una vez que hemos construido las curvas cúbicas  $C_0^i, C_1^i$  y  $C_2^i$  que describen los posibles comportamientos "extremos" en el interior del triángulo  $T_i$ , en lo adelante consideraremos sólo curvas de la familia (4.2) que se escriban como una combinación lineal convexa de las curvas  $C_0^i, C_1^i$  y  $C_2^i$  y que en consecuencia satisfagan todas las condiciones de interpolación impuestas,

$$c_i(u, v) = r_0^i C_0^i(u, v) + r_1^i C_1^i(u, v) + r_2^i C_2^i(u, v) \quad i = 1, \dots, n \quad (4.8)$$

donde

$$\begin{aligned} r_0^i + r_1^i + r_2^i &= 1 \\ r_0^i \geq 0, \quad r_1^i \geq 0, \quad r_2^i &\geq 0. \end{aligned} \quad (4.9)$$



Sustituyendo en  $c_i(u, v)$ , las expresiones (4.5),(4.7),(4.6) de las curvas  $C_0^i, C_1^i$  y  $C_2^i$  respectivamente, obtenemos,

$$c_i(u, v) = -v^3 + a_{10}^i u(1-u-v)^2 + a_{20}^i u^2(1-u-v) + r_0^i a_{02}^i v^2(1-u-v) + r_2^i a_{12}^i uv^2 + (r_0^i b_{11}^i + r_1^i a_{11}^i + r_2^i c_{11}^i)uv(1-u-v) = 0 \quad (4.10)$$

Nótese que como  $C_0^i, C_1^i$  y  $C_2^i$  son curvas conexas, convexas y no singulares en el interior del triángulo  $T_i$ , entonces cualquier combinación convexa de ellas también satisface estas propiedades. Por otro lado, las curvas  $C_0^i, C_1^i$  y  $C_2^i$  interpolan el punto  $Q_i$  prefijado en el interior de  $T_i$ . En consecuencia, cualquier curva  $c_i$ , obtenida como combinación convexa de ellas también pasa por el punto  $Q_i$ .

En lo adelante, los parámetros libres  $r_0^i, r_1^i$  y  $r_2^i$  se utilizarán con dos objetivos principales. En primer lugar, para garantizar la  $G^2$  continuidad del A-spline. Además estos parámetros se pueden usar como asas geométricas que nos permitirán controlar la forma de cada segmento  $c_i$  del A-spline. En ese sentido, observe que como la curva  $C_1^i$  tiene curvatura cero en los puntos  $P_0^i$  y  $P_2^i$ , entonces cualquier curva de la familia (4.8) con coeficiente  $r_1^i$  creciente tiende a pegarse simultáneamente a los lados  $P_0^i P_1^i$  y  $P_2^i P_1^i$  del triángulo  $T_i$ . Por el contrario, si el coeficiente de  $C_1^i$  decrece y los coeficientes de  $C_0^i$  y  $C_2^i$  se igualan, entonces la curva resultante se aleja simultáneamente de ambos lados del triángulo  $T_i$  en forma de pico. Finalmente, si se incrementa el coeficiente de  $C_0^i$  (respectivamente el de  $C_2^i$ ), entonces se obtiene una curva menos simétrica, que se aproxima al lado  $P_0^i P_1^i$  (respectivamente a  $P_1^i P_2^i$ ). Por lo tanto, a través de los coeficientes de la combinación convexa (4.8) se puede controlar la forma de cada segmento del A-spline de manera similar a como hacen Paluszny *et al* en [Pal98] usando los llamados parámetros de tensión.

#### 4.2.4 $G^2$ continuidad del A-spline

En esta sección vamos a mostrar que efectivamente, en términos de las nuevas incógnitas  $r_0^i, r_1^i$  y  $r_2^i$ , la condición de  $G^2$  continuidad del A-spline se convierte en un sistema de ecuaciones lineales. En efecto, de acuerdo con (4.4) para lograr la continuidad de la curvatura del A-spline, en el nodo  $P_2^i = P_0^{i+1}$ , donde se unen dos secciones consecutivas de la familia (4.10) debemos exigir que,

$$r_0^i \left| \frac{a_{02}^i}{a_{10}^i} \right| \frac{\Delta_i}{(g_2^i)^3} = r_2^{i+1} \left| \frac{a_{12}^{i+1}}{a_{20}^{i+1}} \right| \frac{\Delta_{i+1}}{(g_0^{i+1})^3} \quad (4.11)$$

Sustituyendo en (4.11) las expresiones de  $a_{02}^i, a_{10}^i, a_{12}^{i+1}$  y  $a_{20}^{i+1}$  en términos de las coordenadas  $u_i, v_i$  del punto de interpolación, podemos reescribir la condición de  $G^2$  continuidad como,

$$r_0^i \frac{\Delta_i}{(g_2^i)^3} = r_2^{i+1} (1 - u_{i+1} - v_{i+1}) \frac{\Delta_{i+1}}{(g_0^{i+1})^3} \quad (4.12)$$

Nótese que ahora las condiciones de  $G^2$  continuidad del A-spline dan lugar a un sistema de ecuaciones lineales en términos de los coeficientes  $r_j^k$  de la combinación lineal convexa. Si además consideramos las ecuaciones (4.9), entonces tenemos que para construir un A-spline  $G^2$  continuo es necesario resolver el siguiente sistema subdeterminado de ecuaciones lineales ( $2n - 1$  ecuaciones,  $3n$  incógnitas) con restricciones de signo en las incógnitas  $r_j^k$ ,

$$\begin{cases} \alpha_i r_0^i - \beta_i r_2^{i+1} = 0 & i = 1, \dots, n-1 \\ r_0^i + r_1^i + r_2^i = 1 & i = 1, \dots, n \end{cases} \quad (4.13)$$

$$r_0^i \geq 0, \quad r_1^i \geq 0, \quad r_2^i \geq 0.$$

donde

$$\alpha_i = \frac{\Delta_i}{(g_2^i)^3} \quad y \quad \beta_i = (1 - u_{i+1} - v_{i+1}) \frac{\Delta_{i+1}}{(g_0^{i+1})^3} \quad (4.14)$$

Puesto que el sistema lineal (4.13) tiene la solución no trivial,

$$r_0^i = r_2^i = 0, \quad r_1^i = 1, \quad i = 1, \dots, n \quad (4.15)$$

el problema posee en efecto infinitas soluciones.

### Observaciones

1. Si los vértices  $P_1^i$  y  $P_1^{i+1}$  de dos triángulos consecutivos se encuentran en lados diferentes de la recta que pasa por  $P_0^i$  y  $P_2^{i+1}$ , como muestra la figura 4.2 entonces el A-spline está forzado a poseer un punto de inflexión en el segmento formado por las curvas  $c_i$  y  $c_{i+1}$ . Pero ambas secciones son curvas convexas. Por lo tanto, el único posible punto de inflexión es el nodo  $P_2^i = P_0^{i+1}$ . Para lograr que la curvatura de

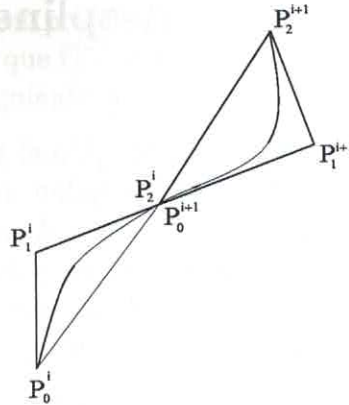


Figura 4.2: Necesidad de introducir un punto de inflexión

$c_i$  sea cero en el punto  $P_2^i$  basta con tomar  $r_0^i = 0$ , pues  $C_1^i$  y  $C_2^i$  tienen curvatura cero en ese punto. Similarmente, si tomamos  $r_2^{i+1} = 0$  entonces resulta que la curva  $c_{i+1}$  tiene curvatura cero en el punto  $P_0^{i+1}$ , ya que lo mismo sucede para las curvas  $C_0^{i+1}$  y  $C_1^{i+1}$ . Nótese que la condición de  $G^2$  continuidad en un punto de inflexión se puede expresar también mediante la ecuación (4.13), si en lugar de calcular  $\alpha_i$  y  $\beta_i$  como en (4.14), asignamos  $\alpha_i = 1$  y  $\beta_i = -1$ .

2. Si el A-spline es una curva cerrada, entonces debemos considerar adicionalmente la siguiente ecuación, que corresponde a la condición de  $G^2$  continuidad en el punto  $P_2^n = P_0^1$ ,

$$r_0^n \frac{\Delta_n}{(g_2^n)^3} = r_2^1 (1 - u_1 - v_1) \frac{\Delta_1}{(g_0^1)^3}$$

3. En lo adelante nos referiremos al A-spline cuyos coeficientes  $r_j^k$  están dados por (4.15) como el A-spline *default* para el polígono de control dado. Esta curva tiene curvatura cero en todos los nodos y en ese sentido es una curva simétrica.
4. El hecho de que el A-spline asociado a un polígono de control no sea único puede considerarse como una ventaja, ya que nos permite utilizar los grados de libertad disponibles, como asas geométricas para controlar de manera natural la forma de cada sección de la curva.

### 4.3 Construcción de un A-spline cúbico no default

En la práctica, una vez que el diseñador ha definido el polígono de control de la curva A-spline, podemos pensar en el A-spline *default*  $S$  como una primera aproximación a la curva deseada por éste. A partir del mismo, se puede entonces seleccionar un triángulo cualquiera y modificar la sección correspondiente del A-spline. Por supuesto, este proceso se puede repetir e ir cambiando en cada paso una nueva sección hasta llegar a la curva que se desea.

#### 4.3.1 Máximas deformaciones

La curva A-spline *default* es una curva simétrica, por cuanto posee curvatura cero en todos los nodos. Supongamos ahora que el diseñador desea modificar la  $i$ -ésima sección del A-spline para obtener una curva más asimétrica, variando lo menos posible el resto de las secciones. Como señalamos anteriormente, esto se puede lograr maximizando el coeficiente  $r_0^i$  o  $r_2^i$  en (4.8) para obtener la (única) curva que más se aproxima al lado  $P_0^i P_1^i$  o  $P_1^i P_2^i$  respectivamente. También es posible obtener una curva simétrica variando lo menos posible el resto de los segmentos del A-spline. Por ejemplo, para obtener la (única) curva que más se despega simultáneamente de  $P_0^i P_1^i$  y  $P_1^i P_2^i$ , basta con minimizar el coeficiente  $r_1^i$ , manteniendo  $r_0^i = r_2^i$ .

Desafortunadamente, si cambiamos la curvatura en el punto  $P_0^i$  o en  $P_2^i$ , entonces es necesario modificar también las curvaturas en  $P_2^{i-1}$  y  $P_0^{i+1}$  para preservar la  $G^2$  continuidad del A-spline, de modo que además de la  $i$ -ésima sección, se afectarán las secciones  $i-1$  e  $i+1$ , mientras que el resto del A-spline no sufre cambios. Por lo tanto, sólo es necesario recalcular los coeficientes  $r_0^j, r_1^j$  y  $r_2^j$  para  $j = i-1, i, i+1$ .

#### Empujando la $i$ -ésima sección hacia $P_0^i P_1^i$

Sea  $\tilde{S}_i$  un A-spline cúbico que coincide con el A-spline default  $S$  excepto en los triángulos  $T_{i-1}, T_i$  y  $T_{i+1}$ , donde se desea obtener una curva más asimétrica, empujando  $c_i$  hacia el lado  $P_0^i P_1^i$ . Denotemos por  $\tilde{r}_0^j, \tilde{r}_1^j, \tilde{r}_2^j$  los coeficientes de la  $j$ -ésima sección del A-spline  $\tilde{S}_i$ . Como la  $j$ -ésima sección de  $\tilde{S}_i$  y  $S$  coinciden para  $j \neq i-1, i, i+1$  entonces,  $\tilde{r}_0^j = r_0^j$ ,  $\tilde{r}_1^j = r_1^j$  y  $\tilde{r}_2^j = r_2^j$  para  $j \neq i-1, i, i+1$ .

Por otro lado, para preservar la  $G^2$  continuidad en los vértices  $P_0^{i-1}$  y  $P_2^{i+1}$ , debemos exigir que  $\tilde{r}_2^{i-1} = r_2^{i-1}$  y  $\tilde{r}_0^{i+1} = r_0^{i+1}$ . Los restantes coeficientes deben satisfacer el siguiente sistema de ecuaciones lineales,

$$\left\{ \begin{array}{l} \tilde{r}_0^{i-1} + \tilde{r}_1^{i-1} = 1 - r_2^{i-1} \\ \tilde{r}_0^i + \tilde{r}_1^i + \tilde{r}_2^i = 1 \\ \tilde{r}_1^{i+1} + \tilde{r}_2^{i+1} = 1 - r_0^{i+1} \\ \alpha_{i-1}\tilde{r}_0^{i-1} - \beta_{i-1}\tilde{r}_2^i = 0 \\ \alpha_i\tilde{r}_0^i - \beta_i\tilde{r}_2^{i+1} = 0 \\ \tilde{r}_0^j \geq 0, \quad \tilde{r}_1^j \geq 0, \quad \tilde{r}_2^j \geq 0, \quad j = i-1, i, i+1 \end{array} \right. \quad (4.16)$$

Supongamos ahora que ni  $P_0^i$  ni  $P_2^i$  son puntos de inflexión del A-spline, es decir que  $\frac{\alpha_i}{\beta_i} > 0$  y  $\frac{\alpha_{i-1}}{\beta_{i-1}} > 0$ .

Para obtener una curva que se pegue al lado  $P_0^i P_1^i$  es necesario maximizar el coeficiente de  $C_0^i$  y minimizar el coeficiente de  $C_2^i$  en (4.8). Por lo tanto, vamos a asignar  $\tilde{r}_2^i = 0$ , de donde resulta que  $\tilde{r}_0^{i-1} = 0$  y  $\tilde{r}_1^{i-1} = 1 - r_2^{i-1}$ , teniendo en cuenta que  $\alpha_{i-1} \neq 0$  y  $\beta_{i-1} \neq 0$ . En consecuencia el sistema de ecuaciones lineales (4.16) se reduce a,

$$\left\{ \begin{array}{l} \tilde{r}_0^i + \tilde{r}_1^i = 1 \\ \alpha_i\tilde{r}_0^i - \beta_i\tilde{r}_2^{i+1} = 0 \\ \tilde{r}_1^{i+1} + \tilde{r}_2^{i+1} = 1 - r_0^{i+1} \\ \tilde{r}_0^j \geq 0, \quad \tilde{r}_1^j \geq 0, \quad \tilde{r}_2^j \geq 0 \quad j = i-1, i, i+1 \end{array} \right.$$

Las infinitas soluciones de este sistema se pueden reescribir en términos del coeficiente  $\tilde{r}_0^i$  que queremos maximizar. En efecto,

$$\tilde{r}_1^i = 1 - \tilde{r}_0^i \quad \tilde{r}_2^{i+1} = \frac{\alpha_i}{\beta_i}\tilde{r}_0^i \quad \tilde{r}_1^{i+1} = 1 - r_0^{i+1} - \frac{\alpha_i}{\beta_i}\tilde{r}_0^i$$

Entre todas las soluciones vamos a escoger el mayor  $\tilde{r}_0^i$  tal que  $\tilde{r}_1^i, \tilde{r}_2^{i+1}$  y  $\tilde{r}_1^{i+1}$  se mantengan en el intervalo  $[0, 1]$ . Nótese que para  $r_0^{i+1}$  dado, la ecuación  $\tilde{r}_1^{i+1} = 1 - r_0^{i+1} - \frac{\alpha_i}{\beta_i}\tilde{r}_0^i$  describe un haz de rectas (en las variables  $\tilde{r}_0^i$  y  $\tilde{r}_1^{i+1}$ ), que pasan por el punto  $(\tilde{r}_0^i, \tilde{r}_1^{i+1}) = (0, 1 - r_0^{i+1})$ . Las rectas de este haz dependen del valor que tenga la pendiente  $-\frac{\alpha_i}{\beta_i}$ , que en todos los casos es negativa. En particular, la recta que pasa por los puntos  $(0, 1 - r_0^{i+1})$  y  $(1, 0)$  tiene pendiente  $-(1 - r_0^{i+1})$ . Por lo tanto, si la pendiente  $-\frac{\alpha_i}{\beta_i}$  de la recta que estamos considerando es mayor que  $-(1 - r_0^{i+1})$  entonces  $\tilde{r}_0^i$  puede alcanzar el máximo valor factible 1, sin que  $\tilde{r}_1^{i+1}$  se salga del intervalo  $[0, 1]$ .

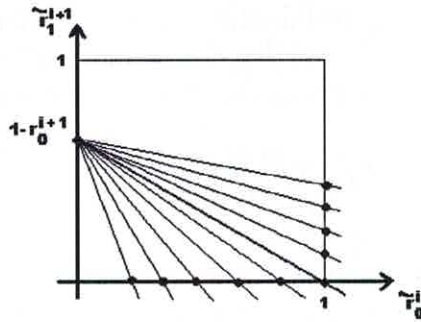


Figura 4.3: Haz de rectas para determinar los parámetros libres del A-spline del caso Izquierdo

De lo contrario, el máximo valor que puede tomar  $\tilde{r}_0^i$  es el que corresponde al punto de intersección de las rectas  $\tilde{r}_1^{i+1} = 1 - r_0^{i+1} - \frac{\alpha_i}{\beta_i} \tilde{r}_0^i$  y  $\tilde{r}_1^{i+1} = 0$  (vea la Figura 4.3).

En resumen, debemos considerar dos posibilidades dentro de este caso, al cual nos referiremos como caso izquierdo,

#### Caso izquierdo

1. Si  $\frac{\alpha_i}{\beta_i} \leq 1 - r_0^{i+1}$ , entonces asignamos  $\tilde{r}_0^i = 1$ . En consecuencia,  $\tilde{r}_1^i = 0$ ,  $\tilde{r}_2^{i+1} = \frac{\alpha_i}{\beta_i}$  y  $\tilde{r}_1^{i+1} = 1 - r_0^{i+1} - \frac{\alpha_i}{\beta_i} \geq 0$ .
2. Si  $\frac{\alpha_i}{\beta_i} > 1 - r_0^{i+1}$ , entonces asignamos  $\tilde{r}_0^i = \frac{\beta_i}{\alpha_i} (1 - r_0^{i+1}) < 1$  y se cumple que  $\tilde{r}_1^i = 1 - \frac{\beta_i}{\alpha_i} (1 - r_0^{i+1}) > 0$ ,  $\tilde{r}_2^{i+1} = 1 - r_0^{i+1}$  y  $\tilde{r}_1^{i+1} = 0$ .

#### Empujando la $i$ -ésima sección hacia $P_1^i P_2^i$

Sea  $\tilde{S}r_i$  un A-spline cúbico que coincide con el A-spline default  $S$  excepto para los triángulos  $T_{i-1}$ ,  $T_i$  and  $T_{i+1}$ . Supongamos ahora que en el  $i$ -ésimo triángulo se desea modificar el segmento  $c_i$  para obtener una curva menos simétrica, que se pegue lo más posible al lado  $P_2^i P_1^i$ . Como señalamos anteriormente, esto se puede lograr maximizando el coeficiente de  $C_2^i$  y minimizando el coeficiente de  $C_0^i$ . Por eso vamos a tomar  $\tilde{r}_0^i = 0$ , de donde resulta que el sistema (4.16) que nos garantiza la  $G^2$  continuidad del A-spline, se reduce en este caso a,

$$\left\{ \begin{array}{l} \tilde{r}_0^{i-1} + \tilde{r}_1^{i-1} = 1 - r_2^{i-1} \\ \tilde{r}_1^i + \tilde{r}_2^i = 1 \\ \tilde{r}_1^{i+1} = 1 - r_0^{i+1} \\ \alpha_{i-1} \tilde{r}_0^{i-1} - \beta_{i-1} \tilde{r}_2^i = 0 \\ \tilde{r}_2^{i+1} = 0 \\ \tilde{r}_0^j \geq 0, \quad \tilde{r}_1^j \geq 0, \quad \tilde{r}_2^j \geq 0, \quad j = i-1, i, i+1 \end{array} \right. \quad (4.17)$$

Tomando como variable libre de este sistema el coeficiente  $\tilde{r}_2^i$  que queremos maximizar llegamos a que,

$$\tilde{r}_1^i = 1 - \tilde{r}_2^i \quad \tilde{r}_0^{i-1} = \frac{\beta_{i-1} \tilde{r}_2^i}{\alpha_{i-1}} \quad \tilde{r}_1^{i-1} = 1 - r_2^{i-1} - \frac{\beta_{i-1} \tilde{r}_2^i}{\alpha_{i-1}}$$

Al igual que el *Caso Izquierdo* vamos a tomar entre las posibles soluciones aquella que nos proporciona el mayor  $\tilde{r}_2^i$  para el cual  $\tilde{r}_1^i, \tilde{r}_0^{i-1}$  y  $\tilde{r}_1^{i-1}$  están en el intervalo  $[0, 1]$ . Para  $r_2^{i-1}$  dado la ecuación  $\tilde{r}_1^{i-1} = 1 - r_2^{i-1} - \frac{\beta_{i-1} \tilde{r}_2^i}{\alpha_{i-1}}$  describe un haz de rectas que pasan por el punto  $(\tilde{r}_2^i, \tilde{r}_1^{i-1}) = (0, 1 - r_2^{i-1})$ . En este haz, las rectas dependen del valor de la pendiente  $-\frac{\beta_{i-1}}{\alpha_{i-1}}$  (negativa) de modo que en particular, la recta que pasa por los puntos  $(0, 1 - r_2^{i-1})$  y  $(1, 0)$  tiene pendiente  $-(1 - r_2^{i-1})$ . Luego si la pendiente de una recta del haz es mayor que  $-(1 - r_2^{i-1})$  entonces  $\tilde{r}_2^i$  puede alcanzar el máximo valor 1, manteniendo  $\tilde{r}_1^{i-1}$  en el intervalo  $[0, 1]$ . De lo contrario, el máximo valor que puede tomar  $\tilde{r}_2^i$  es la abcisa del punto de intersección de las rectas  $\tilde{r}_1^{i-1} = 1 - r_2^{i-1} - \frac{\beta_{i-1} \tilde{r}_2^i}{\alpha_{i-1}}$  y  $\tilde{r}_1^{i-1} = 0$  (vea la Figura 4.4).

En resumen, en este caso, debemos tener en cuenta dos alternativas,

#### *Caso Derecho*

1. Si  $\frac{\beta_{i-1}}{\alpha_{i-1}} \leq 1 - r_2^{i-1}$ , entonces podemos tomar  $\tilde{r}_2^i = 1$ , y de aquí resulta  $\tilde{r}_1^i = 0, \tilde{r}_0^{i-1} = \frac{\beta_{i-1}}{\alpha_{i-1}}$  y  $\tilde{r}_1^{i-1} = 1 - r_2^{i-1} - \frac{\beta_{i-1}}{\alpha_{i-1}} > 0$
2. Si  $\frac{\beta_{i-1}}{\alpha_{i-1}} > 1 - r_2^{i-1}$ , entonces se puede tomar  $\tilde{r}_2^i = \frac{\alpha_{i-1}}{\beta_{i-1}}(1 - r_2^{i-1}) < 1$ , y obtenemos  $\tilde{r}_1^{i-1} = 0, \tilde{r}_0^{i-1} = 1 - r_2^{i-1}$  y  $\tilde{r}_1^i = 1 - \frac{\alpha_{i-1}}{\beta_{i-1}}(1 - r_2^{i-1}) > 0$ .

#### **Puntos de inflexión**

Consideremos ahora el caso en que  $P_0^i$  o  $P_2^i$  son puntos de inflexión del A-spline. Si  $P_0^i$  es un punto de inflexión, pero  $P_2^i$  no lo es entonces podemos suponer que  $\frac{\alpha_{i-1}}{\beta_{i-1}} = -1$  y  $\frac{\alpha_i}{\beta_i} > 0$  (vea 1. en las Observaciones). En este

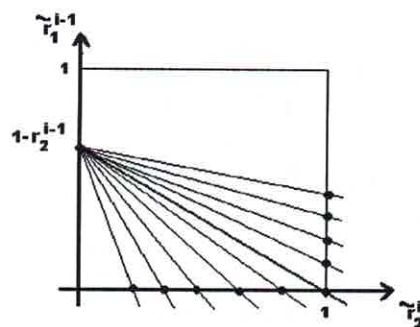


Figura 4.4: Haz de rectas para determinar los parámetros libres del A-spline del caso Derecho

caso es posible empujar la curva hacia el lado  $P_0^i P_1^i$  aunque no hacia  $P_1^i P_2^i$ . Procediendo de manera similar a los casos anteriores, es fácil probar que los valores que se deben asignar a los coeficientes libres son los mismos que en el caso *Izquierdo 2*. Por otro lado, si  $P_2^i$  es un punto de inflexión y  $P_0^i$  no lo es, entonces  $\frac{\alpha_i}{\beta_i} = -1$  y  $\frac{\alpha_{i-1}}{\beta_{i-1}} > 0$ , de manera que se puede empujar la curva hacia  $P_1^i P_2^i$ , asignándole a los parámetros libres los mismos valores que en el caso *Derecho 2*, pero no es posible empujar la curva hacia el lado  $P_0^i P_1^i$ . Obsérvese que si existe un punto de inflexión, las deformaciones admisibles acentúan la presencia del mismo, como es de esperar. Si tanto  $P_0^i$  como  $P_2^i$  son puntos de inflexión del A-spline, entonces no es posible modificar el A-spline *default* original, lo cual es bastante natural ya que el A-spline *default* tiene curvatura cero en ambos puntos!

Supongamos ahora que el diseñador quisiera modificar la  $i$ -ésima sección del A-spline *default*, para obtener una curva simétrica que se pegue simultáneamente lo más posible a los lados  $P_0^i P_1^i$  y  $P_1^i P_2^i$  del triángulo, o por el contrario que se aleje simultáneamente de ambos lados. A continuación veremos que también es posible lograr este tipo de efecto geométrico, si se escogen convenientemente los coeficientes de las curvas  $C_0^i, C_1^i$  y  $C_2^i$ .

#### Empujando la $i$ -ésima sección hacia ambos lados

Sea  $\widetilde{Sd}_i$  un A-spline cúbico que coincide con el spline *default*  $S$  excepto en los triángulos  $T_{i-1}, T_i$  y  $T_{i+1}$ , pues sabemos que al modificar la  $i$ -ésima sección del A-spline, se afectan inevitablemente las secciones  $i-1$  e  $i+1$ .



Para lograr que la sección  $c_i$  se pegue simultáneamente a los lados  $P_0^i P_1^i$  y  $P_1^i P_2^i$  del triángulo  $T_i$ , debemos anular los coeficientes  $\tilde{r}_0^i$  y  $\tilde{r}_2^i$  de las curvas  $C_0^i$  y  $C_2^i$  y en consecuencia hacer  $\tilde{r}_1^i = 1$ . Por otro lado, para preservar la  $G^2$  continuidad del A-spline en los vértices  $P_0^{i-1}$  y  $P_2^{i+1}$  vamos a exigir que  $\tilde{r}_2^{i-1} = r_2^{i-1}$  y  $\tilde{r}_0^{i+1} = r_0^{i+1}$ . En consecuencia, los restantes coeficientes deben satisfacer el siguiente sistema de ecuaciones lineales,

$$\begin{cases} \tilde{r}_0^{i-1} + \tilde{r}_1^{i-1} = 1 - r_2^{i-1} \\ \tilde{r}_1^{i+1} + \tilde{r}_2^{i+1} = 1 - r_0^{i+1} \\ \alpha_{i-1} \tilde{r}_0^{i-1} - \beta_{i-1} \tilde{r}_2^i = 0 \\ \alpha_i \tilde{r}_0^i - \beta_i \tilde{r}_2^{i+1} = 0 \end{cases}$$

Ahora, como  $\alpha_{i-1}, \beta_{i-1}, \alpha_i$  y  $\beta_i$ , son diferentes de 0, de la condición  $\tilde{r}_0^i = \tilde{r}_2^i = 0$  resulta que  $\tilde{r}_0^{i-1} = \tilde{r}_2^{i+1} = 0$  y por lo tanto  $\tilde{r}_1^{i-1} = 1 - r_2^{i-1}$  y  $\tilde{r}_1^{i+1} = 1 - r_0^{i+1}$ .

Resumiendo, para lograr el efecto deseado en este caso debemos tomar,

$$\begin{aligned} \tilde{r}_0^{i-1} = 0, \quad \tilde{r}_1^{i-1} = 1 - r_2^{i-1}, \quad \tilde{r}_2^{i-1} = r_2^{i-1}, \quad \tilde{r}_0^i = \tilde{r}_2^i = 0 \\ \tilde{r}_1^i = 1, \quad \tilde{r}_0^{i+1} = r_0^{i+1}, \quad \tilde{r}_1^{i+1} = 1 - r_0^{i+1}, \quad \tilde{r}_2^{i+1} = 0 \end{aligned}$$

### Alejando la $i$ -ésima sección de ambos lados

Sea  $\tilde{S}_i$  un A-spline cúbico que coincide con el A-spline *default*  $S$  en todas sus secciones, excepto en los triángulos  $T_{i-1}, T_i$  y  $T_{i+1}$ . Si ahora deseamos que el segmento  $c_i$  del A-spline se aleje lo más posible de los lados  $P_0^i P_1^i$  y  $P_1^i P_2^i$  del triángulo  $T_i$ , entonces debemos igualar los coeficientes de  $C_0^i$  y  $C_2^i$  y minimizar el coeficiente de  $C_1^i$ , es decir  $\tilde{r}_1^i$ . De nuevo para preservar la  $G^2$  continuidad en los vértices  $P_0^{i-1}$  y  $P_2^{i+1}$  vamos a imponer que  $\tilde{r}_2^{i-1} = r_2^{i-1}$  y  $\tilde{r}_0^{i+1} = r_0^{i+1}$ , de modo que los restantes coeficientes deben satisfacer el siguiente sistema de ecuaciones,

$$\begin{cases} \tilde{r}_0^{i-1} + \tilde{r}_1^{i-1} = 1 - r_2^{i-1} \\ 2\tilde{r}_2^i = 1 - \tilde{r}_1^i \\ \tilde{r}_1^{i+1} + \tilde{r}_2^{i+1} = 1 - r_0^{i+1} \\ \alpha_{i-1} \tilde{r}_0^{i-1} - \beta_{i-1} \tilde{r}_2^i = 0 \\ \alpha_i \tilde{r}_0^i - \beta_i \tilde{r}_2^{i+1} = 0 \end{cases}$$

Teniendo en cuenta que  $\tilde{r}_0^i = \tilde{r}_2^i$ , de las ecuaciones anteriores resulta que  $\tilde{r}_0^{i-1}$  y  $\tilde{r}_2^{i+1}$  se pueden expresar en términos del coeficiente  $\tilde{r}_1^i$  que queremos

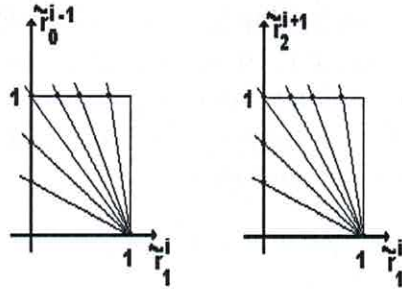


Figura 4.5: Haz de rectas para determinar los parámetros libres del A-spline que se aleja de los lados del triángulo.

minimizar,

$$\tilde{r}_0^{i-1} = \frac{\beta_{i-1}}{2\alpha_{i-1}}(1 - \tilde{r}_1^i), \quad \tilde{r}_2^{i+1} = \frac{\alpha_i}{2\beta_i}(1 - \tilde{r}_1^i)$$

La primera de estas ecuaciones corresponde a un haz de rectas que pasan por el punto  $(\tilde{r}_1^i, \tilde{r}_0^{i-1}) = (1, 0)$  y tienen pendientes negativas  $-\frac{\beta_{i-1}}{2\alpha_{i-1}}$ . La segunda, representa un haz de rectas que pasan por  $(\tilde{r}_1^i, \tilde{r}_2^{i+1}) = (1, 0)$  con pendientes  $-\frac{\alpha_i}{2\beta_i}$ . Por lo tanto, es necesario determinar el menor valor de  $\tilde{r}_1^i$  tal que  $\tilde{r}_0^{i-1}$  y  $\tilde{r}_2^{i+1}$  estén simultáneamente en el intervalo  $[0, 1]$ . Nótese que si ambas pendientes son mayores que  $-1$ , entonces  $\tilde{r}_1^i$  puede alcanzar el mínimo valor factible, es decir 0, ya que  $\tilde{r}_2^{i+1}$  y  $\tilde{r}_0^{i-1}$  se mantienen en  $[0, 1]$ . De lo contrario, el mínimo valor para  $\tilde{r}_1^i$  está dado por la máxima abscisa entre dos puntos: el punto de intersección entre las rectas  $\tilde{r}_0^{i-1} = \frac{\beta_{i-1}}{2\alpha_{i-1}}(1 - \tilde{r}_1^i)$  y  $\tilde{r}_0^{i-1} = 1$  y el punto de intersección entre las rectas  $\tilde{r}_2^{i+1} = \frac{\alpha_i}{2\beta_i}(1 - \tilde{r}_1^i)$  y  $\tilde{r}_2^{i+1} = 1$  (vea la figura 4.5).

En resumen, los coeficientes del A-spline que se aleja lo más posible de ambos lados del triángulo  $T_i$  deben tener los siguientes valores,

$$\begin{aligned} \tilde{r}_2^{i-1} &= r_2^{i-1}, & \tilde{r}_0^{i+1} &= r_0^{i+1} \\ \tilde{r}_1^i &= \begin{cases} 0 & \text{si } \frac{\alpha_i}{2\beta_i} < 1 \text{ y } \frac{\beta_{i-1}}{2\alpha_{i-1}} < 1 \\ \max \left\{ 1 - 2\frac{\beta_i}{\alpha_i}, 1 - 2\frac{\alpha_{i-1}}{\beta_{i-1}} \right\}, & \text{en otro caso} \end{cases} \\ \tilde{r}_0^{i-1} &= \frac{\beta_{i-1}}{2\alpha_{i-1}}(1 - \tilde{r}_1^i), & \tilde{r}_0^i = \tilde{r}_2^i &= (1 - \tilde{r}_1^i)/2, & \tilde{r}_2^{i+1} &= \frac{\alpha_i}{2\beta_i}(1 - \tilde{r}_1^i) \end{aligned}$$

$$\tilde{r}_1^{i+1} = 1 - r_0^{i+1} - \frac{\alpha_i}{2\beta_i}(1 - \tilde{r}_1^i), \quad \tilde{r}_1^{i-1} = 1 - r_2^{i-1} - \frac{\beta_{i-1}}{2\alpha_{i-1}}$$

Como  $\tilde{r}_0^i = \tilde{r}_2^i$ , si alguno de los vértices  $P_0^i$  o  $P_2^i$  es un punto de inflexión, entonces  $\tilde{r}_0^i = \tilde{r}_2^i = 0$ , y regresamos por lo tanto al A-spline *default*, es decir en este caso no es posible alejarse de los lados del triángulo  $T_i$ .

### 4.3.2 Deformaciones graduales

A continuación mostraremos que también es posible introducir deformaciones graduales en la  $i$ -ésima sección del A-spline *default*, simplemente haciendo combinaciones lineales convexas de las curvas que describen las máximas deformaciones y que vimos en la sección anterior. Teniendo en cuenta el comportamiento geométrico de las curvas previas, en lo adelante nos referiremos a  $\tilde{S}l_i$  como la curva *izquierda* y a  $\tilde{S}r_i$  como la curva *derecha*. De manera similar nos referiremos a la curva  $\tilde{S}s_i$  como *aguda*, mientras que  $\tilde{S}d_i$  se identificará como curva del tipo *hombros*.

Para modelar todas las posibles deformaciones graduales en el interior de un triángulo, basta con hacer combinaciones convexas de las curvas  $\tilde{S}l_i$ ,  $\tilde{S}r_i$  y  $\tilde{S}d_i$ , pues una curva aguda se puede obtener asignándole un valor pequeño al coeficiente de  $\tilde{S}d_i$  y valores "grandes" e iguales para los coeficientes de las curvas  $\tilde{S}l_i$  y  $\tilde{S}r_i$ . Como las curvas  $\tilde{S}l_i$ ,  $\tilde{S}r_i$  y  $\tilde{S}d_i$  con  $G^2$  continuas y coinciden con el A-spline *default* en todos los triángulos, excepto en  $T_{i-1}$ ,  $T_i$  y  $T_{i+1}$ , entonces cualquier combinación convexa de ellas cumple la mismas propiedades. Por lo tanto, si se desea cambiar la  $i$ -ésima sección del A-spline *default*, introduciendo deformaciones graduales en el triángulo  $T_i$ , entonces basta con considerar las curvas del tipo  $R_0\tilde{S}l_i + R_1\tilde{S}d_i + R_2\tilde{S}r_i$ , donde el diseñador puede cambiar los coeficientes  $R_0$ ,  $R_1$ ,  $R_2$  de la combinación lineal convexa, hasta obtener una curva con la forma deseada. Obsérvese que en la medida en que se incrementa el coeficiente  $R_0$  logramos el efecto de empujar la  $i$ -ésima sección del A-spline hacia la "izquierda" acercándonos por tanto a una curva del tipo *izquierdo*. Si por el contrario, se incrementa el coeficiente  $R_2$ , entonces la  $i$ -ésima sección del A-spline se pegará hacia la "derecha", obteniendo una curva que se aproxima a la del tipo *derecho*. Finalmente, si aumentamos el valor del coeficiente  $R_1$ , entonces la  $i$ -ésima sección del A-spline tiende a pegarse a ambos lados del triángulo  $T_i$ , como las curvas del tipo *hombros*, mientras que si incrementamos simultáneamente los coeficientes  $R_0 = R_2$ , entonces la sección  $i$ -ésima de la curva A-spline tiende a separarse de los lados del triángulo, al igual que las curvas del tipo *agudo*.

### 4.3.3 Cambio de posición del punto de interpolación.

En este epígrafe mostramos que también es posible modificar la forma de una sección cualquiera del A-spline, si cambiamos la posición del punto de interpolación que se selecciona en el interior de un triángulo escogido. Sea  $S$  un A-spline cúbico dado y supongamos que se desea cambiar la posición del punto de interpolación  $Q_i$  prefijado en el  $i$ -ésimo triángulo. Al introducir una modificación en el triángulo  $i$ , se afectarán también los triángulos  $i-1$  e  $i+1$ , de modo que vamos a denotar por  $\widetilde{SI}_i$ , el A-spline cúbico cuyos coeficientes  $\widetilde{r}_j^i$  coinciden con los de  $S$  en todos los triángulos, excepto en los triángulos  $T_{i-1}$ ,  $T_i$  y  $T_{i+1}$ .

Nótese que si cambiamos la posición de  $Q_i$  entonces varían las coordenadas baricéntricas  $(u_i, v_i)$  de este punto, de modo que debemos actualizar el parámetro  $\beta_{i-1}$ . Una vez introducido el nuevo punto de interpolación, vamos a recalcular el segmento  $c_i$  que corresponde a la solución *default* para el  $i$ -ésimo triángulo (la cual puede modificarse posteriormente, de acuerdo al interés del diseñador). Por lo tanto, asignamos como cero los coeficientes  $\widetilde{r}_0^i$  y  $\widetilde{r}_2^i$  de las curvas  $C_0^i$  y  $C_2^i$  y en consecuencia  $\widetilde{r}_1^i = 1$ . Por otra parte, para preservar la  $G^2$  continuidad del A-spline en los puntos  $P_0^{i-1}$  y  $P_2^{i+1}$ , vamos a conservar los valores de los parámetros  $r_2^{i-1}$  y  $r_0^{i+1}$ , asignándole el mismo valor a los coeficientes correspondientes del A-spline  $\widetilde{SI}_i$ , es decir tomamos  $\widetilde{r}_2^{i-1} = r_2^{i-1}$  y  $\widetilde{r}_0^{i+1} = r_0^{i+1}$ . El resto de los coeficientes  $\widetilde{r}_j^l$  con  $l = i-1, i+1$  deben satisfacer el siguiente sistema de ecuaciones lineales,

$$\begin{cases} \widetilde{r}_0^{i-1} + \widetilde{r}_1^{i-1} = 1 - r_2^{i-1} \\ \widetilde{r}_1^{i+1} + \widetilde{r}_2^{i+1} = 1 - r_0^{i+1} \\ \alpha_{i-1}\widetilde{r}_0^{i-1} - \beta_{i-1}\widetilde{r}_2^i = 0 \\ \alpha_i\widetilde{r}_0^i - \beta_i\widetilde{r}_2^{i+1} = 0 \end{cases}$$

Como  $\alpha_{i-1}, \beta_{i-1}, \alpha_i$  y  $\beta_i$  son diferentes de cero, de la condición  $\widetilde{r}_0^i = \widetilde{r}_2^i = 0$  resulta que  $\widetilde{r}_0^{i-1} = \widetilde{r}_2^{i+1} = 0$  y por lo tanto  $\widetilde{r}_1^{i-1} = 1 - r_2^{i-1}$  y  $\widetilde{r}_1^{i+1} = 1 - r_0^{i+1}$ .

En resumen, los parámetros del A-spline  $\widetilde{SI}_i$  están dados por,

$$\begin{aligned} \widetilde{r}_1^{i-1} &= 0, & \widetilde{r}_1^{i-1} &= 1 - r_2^{i-1}, & \widetilde{r}_2^{i-1} &= r_2^{i-1}, & \widetilde{r}_0^i &= \widetilde{r}_2^i = 0, \\ \widetilde{r}_1^i &= 1, & \widetilde{r}_0^{i+1} &= r_0^{i+1}, & \widetilde{r}_1^{i+1} &= 1 - r_0^{i+1}, & \widetilde{r}_2^{i+1} &= 0 \end{aligned}$$

## 4.4 Graficación de la curva A-spline

Las curvas cúbicas no singulares no se pueden parametrizar racionalmente. Por eso, inicialmente algunos autores propusieron graficar estas curvas introduciendo parametrizaciones no racionales [Pat88]. Actualmente estos métodos son poco utilizados, porque resultan costosos desde el punto de vista computacional. En su lugar, han aparecido algoritmos más eficientes para graficar curvas algebraicas de grado arbitrario, entre los cuales se destacan [Tau94b] y [Chan88].

En [Pal94] se propone en particular un algoritmo muy eficiente para graficar curvas cúbicas, definidas en el interior de un triángulo y que satisfacen las condiciones de interpolación que hemos mencionado. Las curvas que se muestran en los ejemplos de la siguiente sección fueron graficadas utilizando este algoritmo, que parte de la expresión (4.2) de la curva  $f_i(u, v)$  en coordenadas baricéntricas. El algoritmo calcula las coordenadas baricéntricas de los puntos de intersección entre la curva y una malla triangular. Partiendo de un punto  $P$  situado sobre la curva (vea la figura 4.6 tomada de [Pal94]), el algoritmo se mueve sobre la malla (con un recorrido específico) hasta encontrar dos vértices  $A = (u_0, v_0)$  y  $B = (u_1, v_1)$  de la misma, tales que  $f_i(u_0, v_0)f_i(u_1, v_1) \leq 0$ .

Todos los puntos situados sobre la recta que pasa por  $A$  y  $B$ , tienen una coordenada baricéntrica en común. La otra coordenada baricéntrica del punto  $R$  de intersección entre la recta y la curva, se obtiene resolviendo una ecuación de tercer grado (la que resulta de sustituir la ecuación de la recta en la ecuación de la cúbica). Este método es especialmente útil cuando todos los segmentos del A-spline tienen la misma ecuación en coordenadas baricéntricas, como sucede por ejemplo con el A-spline default, que interpola el baricentro de cada triángulo. En ese caso, sólo es necesario calcular una vez las coordenadas baricéntricas de los puntos sobre la curva. En efecto, si  $(u, v, w)$  son las coordenadas baricéntricas de un punto sobre la curva, entonces para obtener las coordenadas afines del punto correspondiente sobre la curva definida en el  $i$ -ésimo triángulo, basta con aplicar la transformación lineal de coordenadas (2.5).

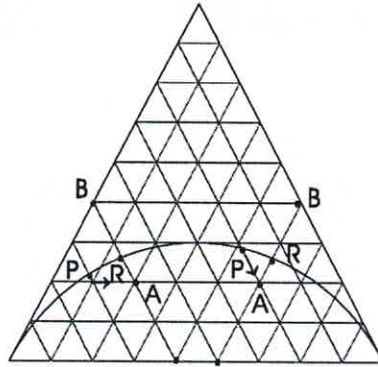


Figura 4.6: Algoritmo para graficar un segmento de cúbica

## 4.5 Ejemplos

En esta sección mostramos algunos ejemplos que nos permiten comprobar cómo el algoritmo propuesto en este capítulo nos ofrece una amplia gama de posibilidades para el diseño de curvas suaves. Los ejemplos mostrados son el resultado de las corridas de un programa en Lenguaje C++, elaborado Dionne León y Eduardo Mesa, estudiantes de 4to año de la Licenciatura en Ciencias de la Computación. El programa tiene carácter interactivo y ofrece facilidades al diseñador para dibujar los puntos y tangentes iniciales y modificar la curva A-spline obtenida, hasta llegar a la solución deseada.

### Ejemplo 1

El objetivo de este ejemplo es probar el efecto que producen las máximas deformaciones que se describen en la sección 4.3.1, sobre la forma de la curva que define un segmento del A-spline. El polígono de control del A-spline está formado por sólo 3 triángulos, y en este ejemplo se modifica el segmento de curva que corresponde al triángulo intermedio. Por supuesto, como se ha explicado previamente, esto introduce también cambios en el primer triángulo y en el tercero.

En la Figura 4.7 (arriba) se exhiben los gráficos del A-spline *default*  $\mathcal{S}_d$  (izquierda) y del spline  $\mathcal{S}_s$  (derecha) obtenido al alejar, de ambos lados del triángulo de control, la segunda sección del A-spline *default*. Esta misma Figura (abajo) muestra el gráfico de la curva  $\mathcal{S}_l$  (izquierda) que se obtiene al empujar el segundo segmento del A-spline *default* hacia el lado  $P_0^2 P_1^2$  y también el gráfico de la curva A-spline  $\mathcal{S}_r$  (derecha) que se obtiene al empujar el segundo segmento del A-spline *default* hacia el lado  $P_1^2 P_2^2$ .

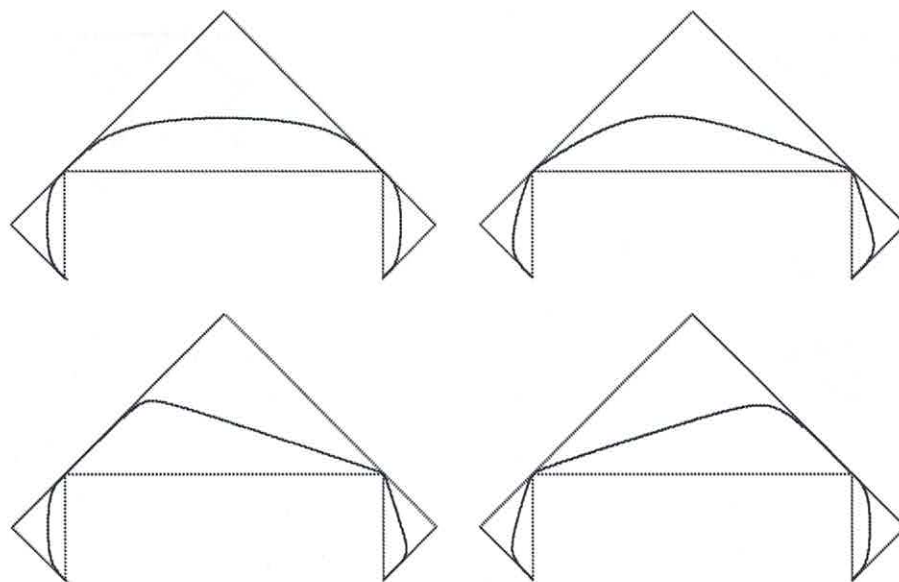


Figura 4.7: Máximas deformaciones de una sección del A-spline

### Ejemplo 2

La figura 4.8 muestra 3 curvas obtenidas para el mismo polígono de control, variando exclusivamente la posición de varios puntos de interpolación del A-spline cúbico *default*. Como se puede apreciar, la selección de un punto arbitrario de interpolación en el interior de cada triángulo es una poderosa herramienta para el diseño libre.

### Ejemplo 3

En este ejemplo se muestran 3 diseños diferentes realizados utilizando A-splines cúbicos. En todos los casos la curva que se muestra es el A-spline *default*, donde el diseñador ha escogido libremente la posición del punto de interpolación.

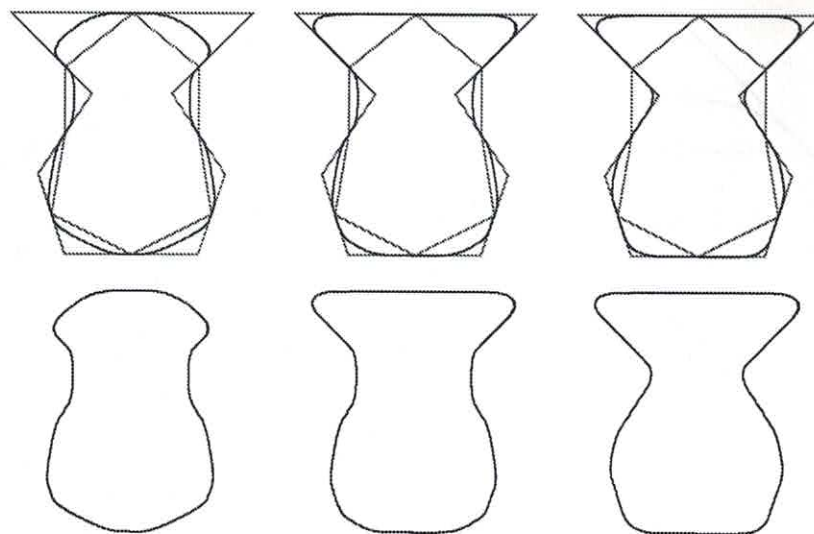


Figura 4.8: Efecto de cambiar la posición del punto de interpolación

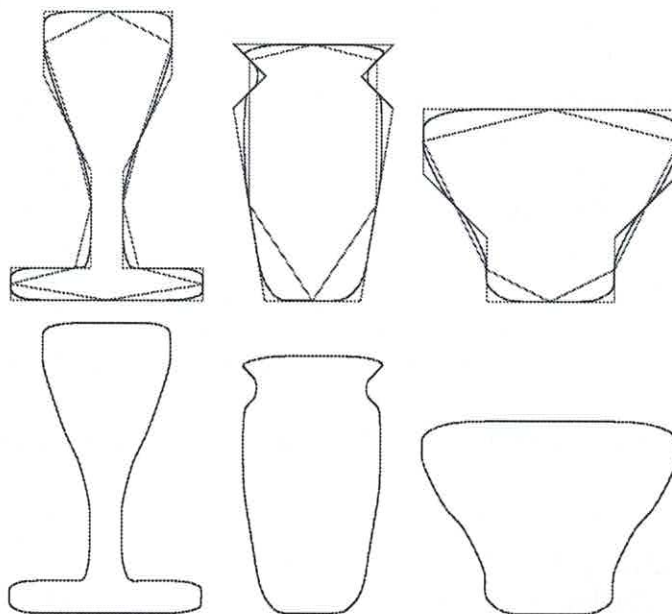


Figura 4.9: Objetos diseñados con el A-spline default para diferentes polígonos de control



## Capítulo 5

# Interpolación con Superficies A-splines

### 5.1 Introducción

Un problema que se presenta con frecuencia hoy en día vinculado al diseño geométrico computarizado es el siguiente. Dado un conjunto de puntos en el espacio tridimensional Euclideo (3D) y una dirección normal asociada a cada punto, construir una superficie "suave" que interpole estos datos, es decir que pase por los puntos y cuya normal en cada uno de ellos sea proporcional a la normal prefijada. Para especificar la topología de la superficie que se quiere diseñar, muchos métodos actuales presuponen que previamente se ha construido una triangulación de los datos, tal que la intersección de cualquier pareja de triángulos es vacía, un punto de interpolación, o un lado común a dos triángulos exclusivamente. De este modo, la unión de los triángulos define una superficie que puede considerarse como una aproximación lineal a la solución del problema. Esta aproximación es continua, pero para la mayoría de los problemas prácticos esto resulta insuficiente y se requiere por lo menos de una superficie con plano tangente continuo en todos los puntos ( $G^1$  continua).

En la actualidad, la mayor parte de los problemas de interpolación de datos en el espacio se resuelven utilizando superficies spline definidas paramétricamente (sobre todo de tipo racional) como las superficies de Bézier, los parches de Coons y las superficies B-spline [Boehm84], [Boor78], [Far86]. En comparación, la representación implícita ha sido poco utilizada, aún

cuando toda superficie racional paramétrica se puede representar mediante una ecuación algebraica de tipo implícito. Si bien la clase de las superficies definidas de forma implícita es mucho más amplia que la de las superficies de tipo paramétrico, su poco uso puede explicarse debido a que estas últimas poseen algunas características que son indeseables, como por ejemplo la presencia de hojas múltiples, autointersecciones y singularidades. Por otra parte, la graficación de tales superficies resulta en general más difícil y costosa, desde el punto de vista computacional, que la graficación de las superficies definidas paraméricamente.

Sin embargo, en los últimos años la representación implícita ha comenzado a utilizarse cada vez con mayor frecuencia y un número creciente de autores ha reconocido la importancia y ventajas que puede tener la misma en la modelación de objetos o en la reconstrucción de imágenes a partir de datos dispersos. En ese sentido, se puede decir que el pionero de los trabajos en esta temática fue Sederberg [Sed85] y que adicionalmente resultan importantes el trabajo Dahmen [Dahm89] que usa parches cuadráticos y los trabajos de Dahmen *et al* [Dahm93] y Guo [Guo91] que usan parches cúbicos. También debemos destacar el trabajo de Bajaj y Ihm [Baj92a] que utiliza parches de grado 5 para triangulaciones convexas y parches de grado 7 para triangulaciones arbitrarias. Además han aparecido un gran número de trabajos de Bajaj y sus colaboradores entre los cuales podemos citar [Baj92b],[Baj94],[Baj95],[Baj97].

Este creciente interés puede explicarse debido a varias razones, como explicamos en la introducción de la Tesis. En primer lugar, porque a diferencia de las superficies paramétricas, el conjunto de las superficies algebraicas es cerrado bajo ciertas operaciones que con frecuencia se realizan en los sistemas de modelación geométrica. Por otra parte, porque para el mismo grado, las superficies definidas implícitamente poseen más grados de libertad, lo cual las hace más flexibles para aproximar formas complicadas, permitiendo obtener superficies splines que o bien poseen órdenes superiores de suavidad o un menor número de parches. Finalmente, la representación implícita es muy útil en la modelación de sólidos, ya que si la frontera del sólido es una superficie definida implícitamente, entonces resulta muy fácil determinar si un punto dado está en el interior o en el exterior del sólido.

En principio, el algoritmo "perfecto" para diseñar una superficie A-spline de interpolación debería reunir los siguientes requisitos [Dahm93]:

- i) La superficie resultante debe poseer por lo menos plano tangente

continuo en todos los puntos, o sea debe ser  $G^1$  continua.

- ii) Los parches algebraicos que componen la superficie deben tener el menor grado posible.
- iii) La superficie no sólo debe interpolar los datos, sino también las direcciones normales prefijadas en cada uno de los puntos de interpolación.
- iv) El esquema de construcción de la superficie debe tener un carácter local.

Obviamente, la  $G^1$  continuidad de la superficie es crucial desde el punto de vista del diseño. Por otro lado, utilizar parches de grado pequeño (2 o 3) nos da la opción de parametrizar la superficie, además de que facilita el proceso de selección de la hoja "correcta" del parche, cuando el mismo posee múltiples hojas en la región de interés. La condición iii) es importante como un instrumento para controlar la forma de la superficie. Finalmente, el requerimiento iv) tiene una trascendencia práctica muy grande, ya que si los parámetros de cada parche sólo dependen de los datos y las direcciones normales en una vecindad pequeña del parche, entonces el cálculo de los coeficientes de los parches se puede realizar en paralelo, lo cual reduce los requerimientos de almacenamiento y tiempo de cálculo. Además, el carácter local de un algoritmo permite modificar un parche o algún dato, sin tener que volver a calcular toda la superficie A-spline.

Aunque en los últimos años se ha producido un indudable avance en los métodos para diseñar superficies A-spline de interpolación, hasta el momento ningún algoritmo puede considerarse "perfecto" atendiendo a las condiciones i)-iv). En general los algoritmos existentes pueden dividirse en dos grandes grupos.

El primero de ellos incluye aquellos algoritmos que comienzan construyendo las curvas de la frontera de cada uno de los parches triangulares que definen el spline. Estas curvas se conocen en la literatura como "malla de alambre" de la superficie. A lo largo de este capítulo nosotros nos vamos a referir a ellas como curvas de la frontera de cada parche. Los algoritmos de este grupo requieren que cada parche contenga no sólo los puntos de interpolación, o sea los vértices de la triangulación introducida, sino también las 3 curvas prefijadas. Esta exigencia da lugar a que el grado de los parches sea elevado. En particular, Bajaj *et al* prueban en [Baj92a)] que si se desea generar una

superficie A-spline  $G^1$  continua que contenga tres curvas perfijadas, entonces debemos utilizar parches algebraicos de grado por lo menos 5 para triangulaciones convexas y de grado al menos 7 para triangulaciones arbitrarias. Pero una superficie algebraica **general** de grado 7 suele poseer frecuentemente múltiples hojas y singularidades y tiene 120 coeficientes, que pueden hacer su cálculo extremadamente costoso y lento. La gran ventaja de los algoritmos de este tipo radica en el hecho de que el cálculo de los A-parches tiene carácter local, o sea los coeficientes de un parche no dependen de sus vecinos, lo cual garantiza que se puedan introducir modificaciones en un parche sin que sea necesario recalcular toda la superficie. El algoritmo que proponemos en este capítulo para interpolar datos 3D por medio de un spline algebraico pertenece a este grupo.

En el segundo grupo se encuentran los algoritmos que calculan directamente los parches, sin prefijar inicialmente las curvas de la frontera. Esto permite utilizar superficies de grado más bajo, pero a cambio de ello es necesario exigir que los parches vecinos se unan “suavemente” a lo largo de la curva común entre ellos. Algunos algoritmos dentro de este grupo [Dahm93], [Baj95], requieren construir previamente una envoltura de tetraedros que tienen como base la triangulación de los puntos a interpolar. Cada parche del A-spline se define entonces en el interior de un tetraedro. En [Baj95] se presenta por primera vez una condición suficiente que garantiza que los parches algebraicos, definidos en el interior de cada tetraedro, sólo tienen una hoja y son conexos. Pero el precio que hay que pagar por tal ventaja es que el algoritmo para construir la envoltura de tetraedros es complicado y costoso y puede dar lugar a que crezca considerablemente el número de parches requeridos para definir la superficie. Por ejemplo, en el peor de los casos, asociado a una pareja de triángulos vecinos resulta necesario definir 8 tetraedros y en consecuencia 8 parches.

En muchos algoritmos de ambos grupos aparece con frecuencia el problema de cómo seleccionar los parámetros libres que garanticen que el parche no posea varias hojas, huecos u otras singularidades. Algunos autores [Dahm93], [Baj92b)] resuelven este problema introduciendo criterios heurísticos para la selección de los parámetros.

## 5.2 Preliminares

Dada una triangulación de los datos en el espacio, en este capítulo proponemos un nuevo algoritmo, al que llamaremos **ASPLINT3D**, para construir una superficie A-spline que interpole los puntos prefijados. El método propuesto sustituye cada triángulo por una superficie algebraica definida implícitamente (A-parche), de modo que los parches se unen de forma suave, dando lugar a un spline cuyo plano tangente varía continuamente en todos los puntos ( $G^1$  continua). Inspirados en el trabajo de Hamann et al [Ham91], el algoritmo propone una solución más simple al problema, que se basa en subdividir el mismo en subproblemas más sencillos. Una vez prefijadas las curvas de la frontera de un parche, cada subproblema consiste entonces en definir una superficie muy sencilla que llamaremos bloque y que contiene únicamente una de las curvas e interpola adicionalmente las direcciones normales prefijadas en los puntos extremos de la misma. Utilizando ciertas funciones de peso, se define finalmente la ecuación implícita de cada A-parche como una combinación lineal convexa de las ecuaciones implícitas de los 3 bloques.

A continuación presentamos una descripción muy breve del algoritmo **ASPLINT3D** para construir una superficie A-spline de interpolación, partiendo de una triangulación  $\Delta$  de un conjunto finito de puntos en el espacio. En las secciones posteriores profundizaremos en cada uno de los pasos del algoritmo.

**Algoritmo ASPLINT3D:** Para cada triángulo  $T \in \Delta$ :

- 1) Seleccionar 3 planos  $\Pi_0, \Pi_1$  y  $\Pi_2$ , cada uno conteniendo una arista de  $T$ .
- 2) Dado el vector normal en cada vértice de  $T$  calcular las direcciones tangentes inducidas en cada uno de los dos planos de la frontera que inciden en ese vértice.
- 3) Definir para cada plano de la frontera  $\Pi_i$  una curva  $c_i$  que interpole los puntos extremos de la arista de  $T$  contenidos en  $\Pi_i$ ,  $i = 0, 1, 2$ , así como las direcciones tangentes prefijadas en 2).
- 4) Para cada plano de la frontera calcular la ecuación implícita de una superficie algebraica que contiene la curva  $c_i$ ,  $S_i : S_i(X, Y, Z) = 0$ ,  $i = 0, 1, 2$ .

- 5) Calcular las funciones de peso  $w_i(X, Y, Z)$ ,  $i = 0, 1, 2$ .
- 6) Construir el parche algebraico  $S$  como,

$$S = \{(X, Y, Z) : \sum_{i=0}^2 w_i(X, Y, Z) S_i(X, Y, Z) = 0\}$$

- 7) Graficar la superficie A-spline.

Si  $m$  representa el número de triángulos en  $\Delta$  no es difícil verificar que la cantidad de operaciones que requiere el algoritmo anterior es proporcional a  $m$ , es decir es del orden  $O(m)$ .

### 5.3 Planos de la frontera

Denotemos por  $P_0, P_1, P_2$  los vértices de un triángulo  $T$  dado y por  $N_0, N_1$  y  $N_2$  las correspondientes direcciones normales prefijadas en los puntos  $P_0, P_1$  y  $P_2$  respectivamente. La normal  $N_i$  en cada vértice  $P_i$ ,  $i = 0, 1, 2$  se puede calcular como el promedio de las direcciones normales a cada una de las caras de la triangulación que inciden en ese vértice.

Para cada arista del triángulo  $T$  vamos a construir un plano que la contiene y donde debe estar situada una de las curvas de la frontera del correspondiente parche. En lo adelante, nos referiremos a tales planos como los **planos de la frontera** y los denotaremos por  $\Pi_0, \Pi_1$  y  $\Pi_2$ . Además nos vamos a referir al plano que pasa por los vértices de  $T$  como el **plano base**.

El plano  $\Pi_0$  es un plano que contiene los puntos  $P_0$  y  $P_1$ . Si la arista  $P_0P_1$  pertenece solamente a un triángulo  $\Delta$ , entonces  $\Pi_0$  es el plano que pasa por  $P_0$  y  $P_1$  y cuya normal es perpendicular al plano base. Por otra parte, si la arista  $P_0P_1$  pertenece a dos triángulos vecinos, entonces se calcula el plano más "cercano" ( en el sentido mínimo cuadrado) a los dos triángulos vecinos e imponemos que el plano  $\Pi_0$  sea perpendicular a éste. El plano de la frontera  $\Pi_1$  contiene los vértices  $P_1$  y  $P_2$ , mientras que el plano  $\Pi_2$  pasa por los vértices  $P_2$  y  $P_0$ . Ambos se calculan de manera similar a como definimos  $\Pi_0$ . Denotaremos por  $M_0, M_1$  y  $M_2$  las direcciones normales a los planos de la frontera  $\Pi_0, \Pi_1$  y  $\Pi_2$  respectivamente.

## 5.4 Introducción de un nuevo sistema de coordenadas

Para simplificar la solución de nuestro problema, en esta sección vamos a introducir un nuevo sistema de coordenadas  $(x, y, z)$  **asociado con cada plano de la frontera**. El objetivo que se persigue con este cambio de coordenadas es lograr que la ecuación implícita de cada plano de la frontera sea muy sencilla. Esto nos permitirá trabajar con familias de curvas en el espacio, contenidas en cada plano de la frontera y cuya expresión analítica en las nuevas coordenadas es también muy simple, a la vez que nos facilitará la construcción de los bloques, cada uno de los cuales contiene una de las curvas de la frontera.

En la sección 5.8 mostraremos en detalle que si los datos tienen una configuración general, siempre es posible introducir un nuevo sistema de coordenadas asociado a cada plano de la frontera, tal que la ecuación implícita del plano sea  $z = 0$ . Denotemos por  $(x, y, z)$  las coordenadas asociadas al plano de la frontera que contiene los vértices  $P_i$  y  $P_j$ ,  $i, j \in \{0, 1, 2\}$ . En las nuevas coordenadas los vértices  $P_i$  y  $P_j$  se transforman en los puntos  $p_0 = (0, 0, 0)$  y  $p_1 = (1, 0, 0)$  respectivamente. Adicionalmente, vamos a exigir que las direcciones normales  $N_i$  y  $N_j$  en los puntos  $P_i$  y  $P_j$  se transformen en vectores proporcionales a las direcciones  $n_0 = (1, 0, 0)$  y  $n_1 = (1, 1, 0)$  en los puntos  $p_0$  y  $p_1$ .

En lo adelante, nos vamos a referir a las nuevas coordenadas  $(x, y, z)$ , como las coordenadas **canónicas**, mientras que las coordenadas originales  $(X, Y, Z)$  serán identificadas como las coordenadas **reales**.

## 5.5 Direcciones tangentes

En la próxima sección vamos a construir las curvas de la frontera  $c_0, c_1$  y  $c_2$  del parche definido sobre un triángulo  $T$ , las cuales se encuentran situadas en los planos  $\Pi_0, \Pi_1$  y  $\Pi_2$  respectivamente. Para ello necesitamos definir previamente las direcciones tangentes a cada curva, en los vértices del triángulo.

Denotaremos por  $T_{00}$  y  $T_{10}$  las direcciones tangentes en los puntos  $P_0$  y  $P_1$  a la curva  $c_0$ . Como  $T_{00}$  y  $T_{10}$  deben ser ortogonales a  $N_0$  y  $N_1$  respectivamente, y deben estar contenidas en plano  $\Pi_0$  definimos,

$$T_{00} = M_0 \times N_0$$

$$T_{10} = M_0 \times N_1$$

El cálculo de las direcciones tangentes  $T_{11}, T_{21}$  a la curva  $c_1$  en los puntos  $P_1$  y  $P_2$  y de las direcciones tangentes  $T_{22}, T_{02}$  a la curva  $c_2$  en los puntos  $P_2$  y  $P_0$  respectivamente se realiza de forma similar. Teniendo en cuenta las tangentes prefijadas en los vértices de un triángulo vamos a considerar dos configuraciones: convexa y no convexa.

**Definición 5.1** [Baj99b)]. Sean  $P_i$  y  $P_j$  dos vértices del triángulo  $T$  contenidos en uno de los tres planos de la frontera. Denotemos por  $\theta_i$  el ángulo entre la recta que pasa por  $P_i$  y  $P_j$  y la tangente definida en  $P_i$ . Similarmente, denotemos por  $\theta_j$  el ángulo entre la recta que pasa por  $P_i$  y  $P_j$  y la tangente definida en  $P_j$ . Si  $\theta_i$  y  $\theta_j$  son ambos ángulos agudos u obtusos, decimos que la configuración de los vértices  $P_i$  y  $P_j$  es **no convexa**. En caso contrario, decimos que la configuración es **convexa**.

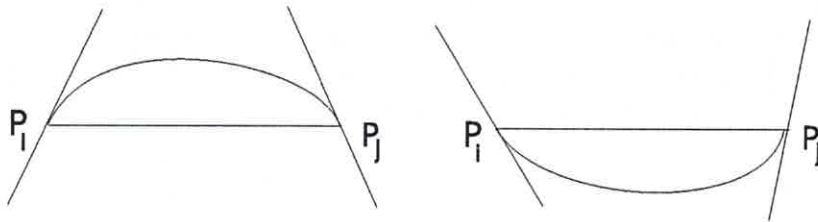


Figura 5.1: Configuraciones convexas

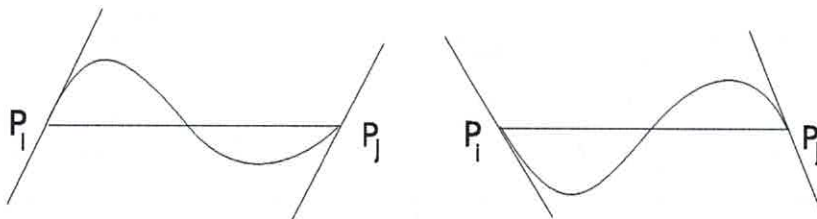


Figura 5.2: Configuraciones no convexas

A continuación veremos cómo construir las curvas de la frontera de cada parche, dependiendo del tipo de configuración de cada pareja de vértices del triángulo base.



## 5.6 Curvas de la frontera en coordenadas canónicas

Una vez que hemos introducido formalmente el nuevo sistema de coordenadas vamos a describir la familia de curvas de la frontera.

### 5.6.1 Curvas de la frontera para la configuración convexa

Supongamos que la configuración de los vértices  $P_i$  y  $P_j$  contenidos en uno de los planos de la frontera es convexa, de acuerdo con la definición 5.1. Entonces es posible construir una familia de curvas cónicas que pasen por los puntos  $P_i$  y  $P_j$  con las tangentes prefijadas anteriormente.

**Lema 5.1** *Toda cónica de la familia*

$$c : c(x, y) = y^2 - 4w_1^2x(1 - x - y) = 0 \quad (5.1)$$

*que depende del parámetro libre  $w_1 \geq 0$ , pasa por los puntos  $p_0$  y  $p_1$  y tiene normales en esos puntos proporcionales a  $n_0$  y  $n_1$  respectivamente.*

**Demostración** Sustituyendo las coordenadas de  $p_0$  y  $p_1$  en (5.1) se obtiene inmediatamente que  $c(p_0) = c(p_1) = 0$ . Por otro lado,

$$\nabla c = (-4w_1^2(1 - x - y) + 4w_1^2x, 2y + 4w_1^2x, 0)$$

Por tanto,

$$\begin{aligned} \nabla c|_{p_0} &= (-4w_1^2, 0, 0) = -4w_1^2n_0 \\ \nabla c|_{p_1} &= (4w_1^2, 4w_1^2, 0) = 4w_1^2n_1 \end{aligned}$$

El parámetro libre  $w_1 \geq 0$  se puede utilizar para controlar la forma de la cónica, pues es conocido [Far93] que en la medida que  $w_1$  crece la cónica que pasa por los puntos  $p_0$  y  $p_1$  se aproxima al punto de intersección de las tangentes prefijadas en esos puntos, mientras que cuando  $w_1$  tiende a cero se aproxima al segmento  $P_0P_1$ . En [Ham91] se propone adicionalmente una forma de asignar automáticamente un valor a  $w_1$ . ■

### 5.6.2 Curvas de la frontera para la configuración no convexa

Supongamos ahora que la configuración de los vértices  $P_i$  y  $P_j$  que están contenidos en uno de los planos de la frontera del parche es no convexa, según la definición 5.1. Entonces la curva de la frontera contenida en el plano dado debe poseer un punto de inflexión situado entre  $P_i$  y  $P_j$ . En consecuencia, en el caso no convexo debemos utilizar curvas de grado por lo menos 3.

**Lema 5.2** *Toda curva cúbica de la familia*

$$c : c(x, y) = -xy^2w_2 - y^3 - w_1x^2(1 - x - y) + w_1(2 + \tau)x(1 - x - y)y/\tau + w_2y^2(1 - x - y) + w_1x(1 - x - y)^2 \quad (5.2)$$

que depende de los parámetros libres  $w_1$ ,  $w_2$  y  $\tau$  pasa por los puntos  $p_0$  y  $p_1$ , tiene normales en esos puntos proporcionales a  $n_0$  y  $n_1$  respectivamente y posee un punto de inflexión en  $p_{1/2} = (\frac{1}{2}, 0)$ . Además, si  $w_1$  y  $w_2$  tienen signos diferentes, entonces la cúbica correspondiente posee un arco conexo que une los puntos  $p_0$  y  $p_1$ .

#### Demostración

Sustituyendo las coordenadas de  $p_0$ ,  $p_1$  y  $p_{1/2}$  en (5.2) es fácil verificar que  $c(p_0) = c(p_1) = c(p_{1/2}) = 0$ .

Por otro lado, las derivadas parciales de cualquier cúbica de la familia (5.2) están dadas por,

$$\frac{\partial c}{\partial x} = -2w_2y^2 - 4w_1x(1 - x - y) + w_1x^2 + w_1(2 + \tau)y(1 - x - y)/\tau - w_1(2 + \tau)xy/\tau + w_1(1 - x - y)^2$$

$$\frac{\partial c}{\partial y} = -2w_2yx - 3y^2 + w_1x^2 + w_1(2 + \tau)x(1 - x - y)/\tau - w_1(2 + \tau)xy/\tau + 2w_2y(1 - x - y) - w_2y^2 - 2w_1x(1 - x - y)$$

$$\frac{\partial c}{\partial z} = 0$$

Luego, si sustituimos las coordenadas de  $p_0$  y  $p_1$  obtenemos,

$$\begin{aligned}\nabla c|_{p_0} &= (w_1, 0, 0) = w_1 n_0 \\ \nabla c|_{p_1} &= (w_1, w_1, 0) = w_1 n_1\end{aligned}$$

es decir el vector normal a cualquier cúbica de la familia (5.2) en los puntos  $p_0$  y  $p_1$  es proporcional a  $n_0$  y  $n_1$  respectivamente.

Por otra parte, haciendo uso de las derivadas parciales en el punto  $p_{1/2}$  podemos escribir la ecuación implícita de la recta tangente en ese punto a una cúbica de la familia (5.2) como,

$$r_t: -w_1(x - 1/2) - 1/2(w_1 - w_1(2 + \tau)/\tau)y = 0 \quad (5.3)$$

Usando MAPLEV calculamos entonces la resultante con respecto a  $x$  entre la cúbica (5.2) y la recta tangente (5.3) obteniendo el polinomio cúbico en  $y$ ,

$$p(y) = -w_1^3(\tau + w_2\tau + 2w_2)y^3/\tau$$

Nótese que  $y = 0$  es una raíz de orden 3 del polinomio  $p(y)$ , por lo cual podemos afirmar que  $p_{1/2}$  es un punto de inflexión de la cúbica.

Finalmente, usando los resultados del lema 2.1 se puede probar directamente que si  $w_1$  y  $w_2$  tienen signos diferentes, entonces la cúbica correspondiente de la familia (5.2) posee un arco convexo y conexo que une los puntos  $p_0$  y  $p_{1/2}$  y otro arco también conexo y convexo que une los puntos  $p_{1/2}$  y  $p_1$ . ■

El parámetro libre  $\tau$  nos permite controlar la pendiente de la recta tangente a una cúbica de la familia en el punto de inflexión.

**Lema 5.3** *Supongamos que  $p_0$  y  $p_1$  son las imágenes de  $P_i$  y  $P_j$  por la transformación lineal  $\Upsilon$ , de las coordenadas reales a las coordenadas canónicas, asociadas al plano de la frontera que contiene los puntos  $P_i$  y  $P_j$ . Entonces es posible escoger el parámetro libre  $\tau$ , exigiendo que la pendiente de la recta tangente en el punto de inflexión a la curva (5.2) escrita en coordenadas reales, tenga un valor prefijado  $\tau_r$ .*

### Demostración

La recta tangente a una curva de la familia (5.2) en el punto de inflexión  $p_{1/2}$  se puede escribir paramétricamente como,

$$\left. \begin{aligned} x &= x \\ y &= \frac{1}{2}(2x - 1)\tau \\ z &= 0 \end{aligned} \right\} \quad (5.4)$$

teniendo en cuenta que en coordenadas canónicas, una curva de esta familia está contenida en el plano  $z = 0$ .

Sea  $r_{ij}$  la recta que pasa por los puntos  $P_i$  y  $P_j$ . Denotemos por  $R = (R_X, R_Y, R_Z)$  un punto en coordenadas reales, tal que el ángulo que se forma entre  $r_{ij}$  y la recta que pasa por el punto medio de  $P_iP_j$  y  $R$ , sea el ángulo con tangente prefijada  $\tau$ . Entonces el correspondiente valor de  $\tau$  en el modelo canónico puede calcularse exigiendo que la imagen de  $R$  por la transformación lineal de coordenadas  $\Upsilon$ , sea un punto en coordenadas canónicas que esté sobre la recta tangente a la curva (en esas coordenadas) en el punto  $p_{1/2}$ . En efecto, si

$$\begin{aligned} x &= \Upsilon_x(X, Y, Z) \\ y &= \Upsilon_y(X, Y, Z) \\ z &= \Upsilon_z(X, Y, Z) \end{aligned}$$

entonces de (5.4) basta con exigir que,

$$\Upsilon_y(R) = \frac{1}{2}(2\Upsilon_x(R) - 1)\tau \quad (5.5)$$

Por tanto, de (5.5) resulta que podemos tomar,

$$\tau = \frac{2\Upsilon_y(R)}{2\Upsilon_x(R) - 1}$$

en el buen entendido de que si  $2\Upsilon_x(R) = 1$  entonces  $\tau = \infty$  ■

En la práctica, para obtener curvas cúbicas que no tengan un cambio de curvatura muy abrupto, ni forma de "S", vamos a exigir que la pendiente de la recta tangente a la curva ( en coordenadas reales) en el punto de inflexión sea  $-\frac{1}{2}$  cuando los ángulos  $\theta_i$  y  $\theta_j$  son agudos y  $\frac{1}{2}$  cuando ambos ángulos son obtusos.

## 5.7 Bloques algebraicos en coordenadas canónicas

En las secciones anteriores construimos las curvas de la frontera definidas en coordenadas canónicas sobre el plano  $z = 0$ . En esta sección vamos a construir, para cada curva de la frontera, una superficie algebraica definida implícitamente  $s : s(x, y, z) = 0$  a la que llamaremos bloque y cuya restricción al plano  $z = 0$  es la curva en cuestión. Más precisamente, cada bloque es un cono cuya curva directriz es la curva de la frontera prefijada.

### 5.7.1 Bloques algebraicos en el caso convexo

Como hemos visto en secciones previas, si la configuración (en coordenadas reales) de una pareja de vértices  $P_i$  y  $P_j$  del triángulo que estamos analizando es convexa, entonces la curva de la frontera que pasa por ambos vértices es una cónica de la familia (5.1) escrita en coordenadas canónicas. En este caso, el bloque correspondiente será una superficie de la familia de cuádricas siguiente:

$$s : s(x, y, z) = (p_x x + p_y y + p_z z + p_c)z - y^2 + 4w_1^2 x(1 - x - y) = 0 \quad (5.6)$$

donde  $p_x, p_y, p_z$  y  $p_c$  son parámetros libres que vamos a determinar exigiendo que  $s$  sea un cono (con vértice  $r_0$ ) que satisfaga las condiciones de interpolación.

**Lema 5.4** *Toda superficie cuadrática  $s$  de la familia (5.6) satisface las siguientes condiciones:*

- i) *La restricción de  $s$  al plano  $z = 0$  es una cónica de la familia (5.1).*
- ii) *Si  $p_x = p_c = 0$  entonces la normal a  $s$  en los puntos  $p_0$  y  $p_1$  es proporcional a los vectores  $n_0$  y  $n_1$  respectivamente.*

### Demostración

- i) Sustituyendo  $z = 0$  en la ecuación (5.6) se obtiene inmediatamente que  $s(x, y, 0) = -y^2 + 4w_1^2 x(1 - x - y)$ .

ii) Derivando en (5.6) y teniendo en cuenta que  $p_x = p_c = 0$  llegamos a que,

$$\begin{aligned}\nabla s(0, 0, 0) &= [4w_1^2, 0, 0] = 4w_1^2 n_0 \\ \nabla s(1, 0, 0) &= [-4w_1^2, -4w_1^2, 0] = -4w_1^2 n_1\end{aligned}$$

Por lo tanto, los vectores normales a  $s$  en  $p_0$  y  $p_1$  son proporcionales  $n_0$  y  $n_1$  respectivamente. ■

**Lema 5.5** *Sea  $F(x, y, z)$  un polinomio de grado  $n$  en las variables  $x, y, z$ . Si el desarrollo en serie de Taylor de  $F$  alrededor del punto  $r_0$  es una forma de grado  $n$ , entonces la superficie  $F(x, y, z) = 0$  es un cono con vértice  $r_0$ .*

### Demostración

Desarrollando  $F$  en serie de Taylor alrededor del punto  $r_0 = (x_0, y_0, z_0)$  se obtiene,

$$F(x, y, z) = F_0 + F_1(x - x_0, y - y_0, z - z_0) + \dots + F_n(x - x_0, y - y_0, z - z_0)$$

donde cada  $F_i$  es un polinomio homogéneo de grado  $i$ . Pero por hipótesis,  $F$  es una forma de grado  $n$ , es decir  $F_0 = F_1 = \dots = F_{n-1} = 0$  y en consecuencia,

$$F(x, y, z) = F_n(x - x_0, y - y_0, z - z_0)$$

Sea  $r_1 = (x_1, y_1, z_1)$  un punto sobre la superficie  $F$  y consideremos la recta  $l$  que pasa por los puntos  $r_0$  y  $r_1$ ,

$$l : l(t) = r_0 + t(r_1 - r_0)$$

Si  $(x(t), y(t), z(t))$  es un punto que está en  $l$  entonces,

$$F(x(t), y(t), z(t)) = F_n(t(x_1 - x_0), t(y_1 - y_0), t(z_1 - z_0))$$

Pero  $F_n$  es un polinomio homogéneo de grado  $n$ . Por lo tanto, de la última igualdad obtenemos,

$$F(x(t), y(t), z(t)) = t^n F_n(x_1 - x_0, y_1 - y_0, z_1 - z_0) = t^n F(x_1, y_1, z_1) = 0$$

Luego, todo punto sobre la recta  $l$  pertenece a la superficie, es decir  $F(x, y, z) = 0$  es un cono con vértice  $r_0$ . ■

De acuerdo con el Lema 5.5 sabemos entonces que una superficie cuadrática de la familia (5.6) es un cono con vértice  $r_0 = (x_0, y_0, z_0)$  si,

$$\begin{aligned} s(r_0) &= 0 \\ \nabla s(r_0) &= (0, 0, 0) \end{aligned}$$

Estas ecuaciones definen un sistema de 4 ecuaciones no lineales que se pueden resolver tomando  $p_y$  como parámetro libre. Una de las soluciones de tal sistema, que obtuvimos usando MAPLEV, es  $p_z = p_y^2/4$ ,  $x_0 = 0$ ,  $y_0 = 1$  y  $z_0 = 2/p_y$ . Definiendo  $p_y = 2/\varepsilon$ , donde  $\varepsilon$  es a su vez un parámetro libre, llegamos a que la superficie,

$$s : s(x, y, z) = (2\varepsilon y - \varepsilon^2 z)z - y^2 + 4w_1^2 x(1 - x - y) = 0 \quad (5.7)$$

es un cono cuadrático con vértice  $r_0 = (0, 1, 1/\varepsilon)$  cuya restricción al plano  $z = 0$  es la curva (5.1). Nótese además que la normal a  $s$  en los puntos  $p_0$  y  $p_1$  es proporcional a  $n_0$  and  $n_1$  respectivamente.

### 5.7.2 Bloques algebraicos en el caso no convexo

Supongamos ahora que la configuración (en coordenadas reales) de la pareja de vértices  $P_i$  y  $P_j$  del triángulo que estamos analizando no es convexa. Entonces la curva de la frontera que pasa por ambos vértices es una cúbica de la familia (5.2) escrita en coordenadas canónicas. En este caso, vamos a extraer el bloque que contiene a la curva de la frontera, de la siguiente familia de superficies cúbicas dadas también en coordenadas canónicas,

$$\begin{aligned} s : s(x, y, z) = & ((p_x x + p_y y + p_z z + p_c) y + (q_x x + q_z z + q_c) z + \\ & (o_c - o_c x) x) z + c(x, y) = 0 \end{aligned} \quad (5.8)$$

donde  $c(x, y)$  es una cúbica de la familia (5.2) y  $p_x, p_y, p_z, p_c, q_x, q_z, q_c$  y  $o_c$  son parámetros libres que vamos a determinar exigiendo que  $s$  sea un cono (con vértice  $r_0$ ), que satisfice las condiciones de interpolación.

**Lema 5.6** *Toda superficie cúbica  $s$  de la familia (5.8) satisfice las siguientes condiciones:*

- i) La restricción de  $s$  al plano  $z = 0$  es una cúbica de la familia (5.2).
- ii) La normal a  $s$  en los puntos  $p_0$  y  $p_1$  es proporcional a los vectores  $n_0$  y  $n_1$  respectivamente.

### Demostración

- i) Sustituyendo  $z = 0$  en (5.8) resulta que  $s(x, y, 0) = c(x, y)$  donde  $c(x, y)$  es una cúbica de la familia (5.2).
- ii) Por otro lado, derivando en (5.8) obtenemos,

$$\begin{aligned}\nabla s(0, 0, 0) &= [w_1, 0, 0] = w_1 n_0 \\ \nabla s(1, 0, 0) &= [w_1, w_1, 0] = w_1 n_1\end{aligned}$$

Luego, los vectores normales a  $s$  en  $p_0$  y  $p_1$  son proporcionales a  $n_0$  y a  $n_1$ , respectivamente.

Del Lema 5.5 sabemos que para que  $s$  sea un cono con vértice  $r_0 = (x_0, y_0, z_0)$  debemos exigir que  $s(r_0) = 0$  y que adicionalmente todas las derivadas parciales de orden menor que 3 se anulen en  $r_0$ . Estas condiciones dan lugar a un sistema de 10 ecuaciones no lineales para determinar los 8 parámetros libres más las 3 coordenadas del vértice del cono. Una solución de este sistema se calculó utilizando MAPLEV, conservando inicialmente a  $p_x$  como parámetro libre. El valor de  $p_x$  se determinó más tarde imponiendo que  $z_0 = 1/\varepsilon$ , donde  $\varepsilon$  también es una variable libre, obteniéndose,

$$\begin{aligned}s : s(x, y, z) = & \\ z \left( y \left( \varepsilon x (w_1(2 + \tau) + 4\tau w_2) / \tau + (3 + 2w_2)\varepsilon y - \varepsilon^2(3 + w_2)z - 2w_2\varepsilon \right) + \right. & \\ \left. (-2w_2\varepsilon^2 x + \varepsilon^3 z + w_2\varepsilon^2)z + \varepsilon w_1(2 + \tau)x(x - 1) / \tau \right) + c(x, y) = 0 & \\ & (5.9)\end{aligned}$$

Esta superficie es un cono cúbico con vértice  $r_0 = (0, 1, 1/\varepsilon)$ , cuya restricción al plano  $z = 0$  es una curva  $c(x, y)$  de la familia (5.2). Nótese que adicionalmente, las normales a  $s$  en  $p_0$  y  $p_1$  son proporcionales a  $n_0$  y  $n_1$  respectivamente.



## 5.8 Transformación de coordenadas reales a coordenadas canónicas

Una vez que hemos descrito cómo construir las curvas de la frontera de cada parche y los bloques algebraicos, utilizando las coordenadas canónicas, vamos a mostrar que efectivamente, para cada plano de la frontera  $\Pi_i$ ,  $i = 0, 1, 2$  es posible introducir una transformación lineal (no degenerada)  $\Upsilon_i$ , de las coordenadas reales  $(X, Y, Z)$  a las canónicas  $(x, y, z)$ , tal que el plano  $\Pi_i$  se transforme en el plano  $z = 0$ . La construcción de las tres transformaciones lineales asociadas a cada parche se realiza de forma similar, por lo cual sólo mostraremos explícitamente cómo calcular  $\Upsilon_0$ .

Sea  $\Upsilon_0(X, Y, Z) = (x, y, z)$  la transformación lineal

$$(x, y, z)^t = A_0(X - X_0, Y - Y_0, Z - Z_0)^t \quad (5.10)$$

donde  $(X_0, Y_0, Z_0)$  son las coordenadas reales del vértice  $P_0$  y  $A_0$  es la matriz (incógnita) asociada a  $\Upsilon_0$ , cuya primera fila vamos a denotar por  $a$ , la segunda por  $c$  y la última por  $d$ .

Como mencionamos previamente,  $\Upsilon_0$  debe transformar el plano  $\Pi_0$  dado en coordenadas reales, en el plano  $z = 0$  en coordenadas canónicas. Esto es equivalente a exigir que las imágenes por  $\Upsilon_0$  de 3 puntos situados en el plano  $\Pi_0$  sean 3 puntos en el plano  $z = 0$ . Más precisamente, vamos a exigir que,

$$\begin{aligned} i) \Upsilon_0(P_0) &= p_0 \\ ii) \Upsilon_0(P_1) &= p_1 \\ iii) \Upsilon_0(P_0 + T_{00}) &= (*, *, 0) \end{aligned}$$

donde recordemos que  $p_0 = (0, 0, 0)$ ,  $p_1 = (1, 0, 0)$  y  $T_{00}$  es la tangente a la curva  $c_0$  en el punto  $P_0$ .

De la propia definición (5.10) está claro que la condición i) se cumple inmediatamente. Sea  $P_1 = (X_1, Y_1, Z_1)$ , entonces las condiciones ii) y iii) dan lugar al siguiente sistema de 4 ecuaciones lineales (en los coeficientes incógnitas de la matriz  $A_0$ )

$$A_0(X_1 - X_0, Y_1 - Y_0, Z_1 - Z_0)^t = (1, 0, 0)^t \quad (5.11)$$

$$T_{00} \cdot d^t = 0 \quad (5.12)$$

Denotemos por  $S_0(X, Y, Z) = 0$  la ecuación implícita en coordenadas reales del bloque que contiene la curva de la frontera  $c_0$  situada en el plano

$\Pi_0$ . Si utilizamos la transformación lineal de coordenadas  $\Upsilon_0$ , resulta obvio que,

$$S_0(X, Y, Z) = s_0(\Upsilon_0(X, Y, Z))$$

donde  $s_0(x, y, z) = 0$  denota la ecuación implícita en coordenadas canónicas del bloque, dada por (5.7) si la configuración de los vértices  $P_0$  y  $P_1$  contenidos en el plano  $\Pi_0$  es convexa y por (5.9) si la configuración es no convexa. Entonces, para  $i = 0, 1$  tenemos,

$$S_0(P_i) = s_0(\Upsilon_0(P_i)) = s_0(p_i) = 0$$

o sea el bloque  $S_0$  contiene los puntos  $P_0$  y  $P_1$ . Por otro lado, como  $\Upsilon_0$  es una transformación lineal, los planos tangentes a  $S_0$  en los puntos  $P_0$  y  $P_1$  se transforman por  $\Upsilon_0$ , en los planos tangentes a  $s_0$  en  $p_0$  y  $p_1$  respectivamente. Usando la regla de la cadena es fácil ver que,

$$\nabla S_0(X, Y, Z)^t = A_0^t \nabla s_0(x, y, z)^t$$

Pero recordemos que independientemente de que  $s_0$  sea un bloque cuadrático o cúbico,  $\nabla s_0(p_0)$  es proporcional a  $n_0 = (1, 0, 0)$  y  $\nabla s_0(p_1)$  es proporcional a  $n_1 = (1, 1, 0)$ . Por lo tanto,

$$\begin{aligned} \nabla S_0(P_0)^t &= \alpha_0 A_0^t n_0^t = \alpha_0 a^t \\ \nabla S_0(P_1)^t &= \alpha_1 A_0^t n_1^t = \alpha_1 (a + c)^t \end{aligned}$$

donde  $\alpha_0$  y  $\alpha_1$  son constantes no nulas. Por lo tanto, las ecuaciones implícitas de los planos tangentes a  $S_0$  en los puntos  $P_0$  y  $P_1$  están dadas respectivamente por,

$$\begin{aligned} \Pi_{t0}: (X - X_0, Y - Y_0, Z - Z_0) \cdot a^t &= 0 \\ \Pi_{t1}: (X - X_1, Y - Y_1, Z - Z_1) \cdot (a + c)^t &= 0 \end{aligned}$$

Finalmente, vamos a exigir que las normales a  $S_0$  en los puntos  $P_0$  y  $P_1$  sean proporcionales a  $N_0$  y  $N_1$  respectivamente. Esto es equivalente a imponer que dos vectores linealmente independientes y ortogonales a  $N_j$  estén contenidos en el plano  $\Pi_{tj}$ ,  $j = 0, 1$ .

Recordemos que  $T_{j0}$ ,  $j = 0, 1$  es el vector tangente en el punto  $P_j$  a la curva  $c_0$  (intersección entre  $S_0$  y  $\Pi_0$ ). Por tanto, basta con exigir que los

vectores  $T_{j0}$  y  $V_{j0} = T_{j0} \times N_j$  estén ambos situados en el plano  $\Pi_{tj}$ ,  $j = 0, 1$ , o sea que,

$$T_{00} \cdot a^t = 0 \quad (5.13)$$

$$V_{00} \cdot a^t = 0 \quad (5.14)$$

$$T_{10} \cdot (a + c)^t = 0 \quad (5.15)$$

$$V_{10} \cdot (a + c)^t = 0 \quad (5.16)$$

Las condiciones (5.13), (5.14), (5.15), (5.16) definen 4 ecuaciones lineales más para calcular los coeficientes de la matriz  $A_0$  que representa la transformación lineal de coordenadas  $\Upsilon_0$ .

Resumiendo, para calcular los 9 elementos  $x = (a, c, d)$  de la matriz  $A_0$  asociada a la transformación de coordenadas  $\Upsilon_0$  es necesario resolver el sistema de 8 ecuaciones lineales (5.11), (5.12), (5.13), (5.14), (5.15), (5.16), que se puede escribir matricialmente como,

$$Cx^t = e \quad (5.17)$$

donde la matriz  $C$  está dada por,

$$C = \begin{pmatrix} P_{10} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & P_{10} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & P_{10} \\ V_{00} & \mathbf{0} & \mathbf{0} \\ T_{00} & \mathbf{0} & \mathbf{0} \\ T_{01} & T_{01} & \mathbf{0} \\ V_{01} & V_{01} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & T_{00} \end{pmatrix}$$

con  $P_{10} = P_1 - P_0$ , mientras que el lado derecho es  $e = (1, 0, 0, 0, 0, 0, 0, 0)^t$ . Nótese que todos los elementos que aparecen en la matriz  $C$  son vectores filas de 3 componentes, en particular  $\mathbf{0} = (0, 0, 0)$ .

Para una distribución general de los puntos de interpolación, el sistema (5.17) tiene por lo menos una solución y en consecuencia tiene realmente infinitas soluciones. Entre ellas, es posible seleccionar una solución tal que la matriz  $A_0$  sea de rango 3 y por lo tanto  $\Upsilon_0$  sea una transformación lineal no degenerada.

Para ello, calculamos la solución mínimo cuadrada de norma mínima del sistema (5.17), que denotaremos por  $x_{min}$  y una base del subespacio nulo de

*C.* Supongamos, por ejemplo, que el subespacio nulo de  $C$  tiene dimensión 2 y que los vectores  $z_1$  y  $z_2$  forman una base del mismo. Entonces vamos a tomar,

$$x = x_{min} + \alpha z_1 + (1 - \alpha) z_2$$

donde el parámetro libre  $\alpha \in (0, 1)$  se escoge de modo que la matriz

$$A_\alpha = \begin{pmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{pmatrix}$$

asociada a la transformación lineal  $\Upsilon_0$  sea no singular. Más precisamente, si  $s_3(\alpha)$  denota el menor valor singular de la matriz  $A_\alpha$ , entonces  $\alpha$  es la solución del problema no lineal,

$$\begin{aligned} \max s_3(\alpha) \\ \alpha \in (0, 1) \end{aligned}$$

## 5.9 Selección de los vértices de los conos

La selección de los vértices de cada uno de los conos que se usarán para definir un A-parche, influye de manera importante en la forma final del mismo. En esta sección, describimos brevemente las ideas principales del algoritmo que utilizamos para determinar una posición adecuada para los vértices. Para más detalles se puede ver [Her00c].

En general, resulta natural exigir que el vértice de cada cono se encuentre alejado del plano de la frontera que contiene su curva directriz, ya que esto hará más fácil la selección de la hoja “apropiada” del cono, es decir de aquella hoja del cono que contiene el segmento de curva directriz que une dos vértices del triángulo base. Además esto evita que el vértice del cono quede en el interior del volumen limitado por los planos de la frontera, es decir en la región donde estará definido el parche. En lo que sigue, vamos a ver que el algoritmo propuesto depende esencialmente de si la curva directriz del cono, y en consecuencia el cono, están compartidos entre dos parches vecinos o no.

Según mostramos anteriormente, el vértice del cono cuadrático que se utiliza cuando la configuración de una arista es convexa, así como el vértice del cono cúbico que se usa cuando la configuración es no convexa, tienen

coordenadas canónicas  $(0, 1, 1/\varepsilon)$ , donde  $\varepsilon$  es un parámetro libre. En otras palabras, esto significa que, en coordenadas canónicas, el vértice de cada cono pertenece a una recta en el espacio que se puede escribir paramétricamente como,

$$r : r(t) = (x(t), y(t), z(t)) = (0, 1, t)$$

Pero la transformación de coordenadas reales a canónicas y su inversa son lineales. Por lo tanto, en coordenadas reales, también el vértice  $v$  del cono debe estar situado sobre una recta que denotaremos por  $R$ . Para fijar ideas, vamos a suponer que  $v$  es el vértice del cono que contiene la curva de la frontera que pasa por los puntos  $P_0$  y  $P_1$ . Sea  $A_0$  la matriz asociada a la transformación lineal de las coordenadas reales, a las coordenadas canónicas asociadas al plano de la frontera  $\Pi_0$ , que contiene la curva directriz del cono. Entonces las coordenadas reales  $(v_X, v_Y, v_Z)$  del vértice del cono deben satisfacer que,

$$A_0(v_X - X_0, v_Y - Y_0, v_Z - Z_0)^t = (0, 1, 1/\varepsilon)^t \quad (5.18)$$

El algoritmo para determinar la posición exacta del vértice  $v$  consta de 3 pasos fundamentales:

- 1) Calcular una posición inicial  $v^0$  para el vértice  $v$ .
- 2) Determinar si la posición  $v^0$  es "adecuada".
- 3) Corregir la posición de  $v^0$ , en caso necesario.

Hagamos algunos comentarios sobre cada uno de los pasos del algoritmo anterior.

- 1) La posición inicial  $v^0 = (v_X^0, v_Y^0, v_Z^0)$  del vértice del cono es el punto donde la recta  $R$  (sobre la cual debe estar situado el vértice) corta al plano que pasa por  $P_0$  y es perpendicular al plano de la frontera que contiene la curva directriz del cono. Si denotamos por  $N = (N_X, N_Y, N_Z)$  la normal al plano que pasa por  $P_0$  y es perpendicular a  $\Pi_0$ , entonces podemos escribir su ecuación implícita como,

$$N_X(X - X_0) + N_Y(Y - Y_0) + N_Z(Z - Z_0) = 0 \quad (5.19)$$

donde  $(X_0, Y_0, Z_0)$  son las coordenadas reales del punto  $P_0$ . Por lo tanto, basta con exigir que  $v^0$  satisfaga la ecuación (5.19), lo cual se puede escribir de forma compacta como,

$$N.(v - P_0)^t = 0 \quad (5.20)$$

Pero de (5.18) tenemos que,

$$(v - P_0)^t = A_0^{-1}(0, 1, 1/\epsilon)^t$$

ya que  $A_0$  es una matriz inversible. Por lo tanto, sustituyendo en (5.20) llegamos a que,

$$N.A_0^{-1}(0, 1, 1/\epsilon)^t = 0 \quad (5.21)$$

La ecuación (5.21) es lineal en  $1/\epsilon$ , de modo que resolviéndola es fácil determinar el valor del parámetro libre  $\epsilon$ , que nos garantiza que el vértice inicial  $v^0$  se encuentre situado en el plano perpendicular al plano de la frontera  $\Pi_0$ .

- 2) ¿Cuándo se considera “adecuada” la posición del vértice  $v$ ? Esto depende de si la curva directriz del cono pertenece a un sólo parche o es compartida entre dos parches vecinos. Supongamos otra vez que la curva directriz del cono que estamos analizando, pasa por los puntos  $P_0$  y  $P_1$  del triángulo base.

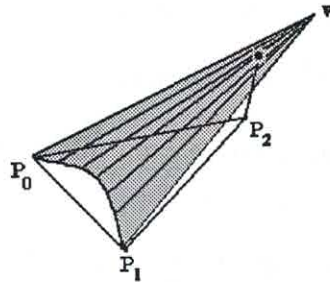


Figura 5.3: Posición “adecuada”: vértice de un cono no compartido

Si esta curva sólo pertenece a un parche, entonces la posición del vértice  $v$  se considera “adecuada” cuando la recta que parte de  $P_2$  y es perpendicular al plano base, corta al plano que contiene a los puntos  $P_0, P_1$  y  $v$  en un punto interior al triángulo  $P_0P_1v$  (vea la figura 5.3).

Similarmente, cuando la curva directriz del cono pertenece a dos parches vecinos definidos sobre los triángulos  $i$  e  $i + 1$  entonces la posición del vértice  $v$  se considera "adecuada", si las rectas que parten de  $P_2^i$  y  $P_2^{i+1}$  y que son perpendiculares a los planos que contienen cada triángulo base, cortan al plano que pasa por  $P_0$ ,  $P_1$  y el vértice  $v$  del cono, en el sector triangular contenido entre las rectas que pasan por  $P_0$  y  $v$  y por  $P_1$  y  $v$  respectivamente (vea la figura 5.4).

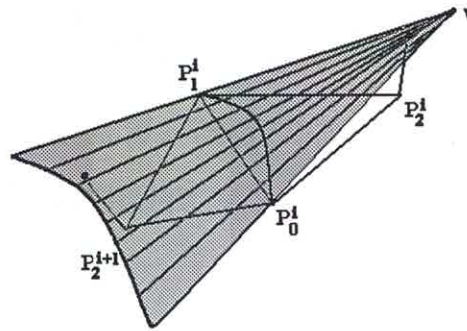


Figura 5.4: Posición "adecuada": vértice de un cono compartido

Al situar el vértice de cada cono en una posición "adecuada" garantizamos que los mismos cubran cada uno de los triángulos, de modo que las rectas perpendiculares a cada plano base los corten. Como veremos en la sección 5.12 esto resulta importante para graficar el A-parche.

- 3) Si la curva directriz del cono sólo pertenece a un parche, entonces el proceso de correcciones en el paso 3) sólo se realiza una vez, es decir es posible proponer apriori una posición "adecuada" para el vértice del cono. Por el contrario, cuando el cono es compartido entre dos parches vecinos, el proceso de correcciones es de tipo iterativo y en cada paso se corrige la posición del vértice del paso anterior, hasta llegar a una posición "adecuada" según la definición dada.

## 5.10 Funciones de peso

En esta sección vamos a definir 3 funciones de peso, cada una de las cuales estará asociada a un plano de la frontera. Estas funciones se usarán para

construir el parche algebraico y tienen como objetivo garantizar que la intersección del parche con cada plano de la frontera sea exactamente la curva de la frontera prefijada en ese plano.

Denotemos por  $\Pi_i(X, Y, Z) = 0$  la ecuación implícita del plano  $\Pi_i$ ,  $i = 0, 1, 2$ . Entonces vamos a definir la función de peso  $w_i$ ,  $i = 0, 1, 2$  como,

$$w_i(X, Y, Z) = \Pi_j(X, Y, Z)^2 \Pi_k(X, Y, Z)^2 \quad j, k \neq i \quad (5.22)$$

**Lema 5.7** *Las funciones de peso definidas por (5.22) poseen las siguientes propiedades:*

- i) Si  $(X, Y, Z) \in \Pi_j$ , entonces  $w_i(X, Y, Z) = 0$ , para  $i \neq j$ .
- ii) Si  $(X, Y, Z) \in \Pi_j$ , entonces  $\nabla w_i(X, Y, Z) = 0$ , para  $i \neq j$ .

#### **Demostración**

- i) Sea  $(X, Y, Z) \in \Pi_j$ , entonces  $\Pi_j(X, Y, Z) = 0$ . Por lo tanto, de (5.22)  $w_i(X, Y, Z) = 0$  para  $i \neq j$ .
- ii)  $\nabla w_i(X, Y, Z) = 2(\Pi_j(X, Y, Z)\nabla\Pi_j(X, Y, Z)\Pi_k(X, Y, Z)^2 + \Pi_j(X, Y, Z)^2\Pi_k(X, Y, Z)\nabla\Pi_k(X, Y, Z))$

Luego, si  $(X, Y, Z) \in \Pi_j$  para  $j \neq i$  entonces  $\nabla w_i(X, Y, Z) = 0$ . ■

## **5.11 A-parches**

Como habíamos mencionado previamente, cada parche algebraico correspondiente a un triángulo  $T \in \Delta$  se definirá en coordenadas reales  $(X, Y, Z)$  y en términos de los bloques implícitos  $S_i : S_i(X, Y, Z) = 0$  y de las funciones de peso  $w_i(X, Y, Z)$ ,  $i = 0, 1, 2$ . Para ello, previamente debemos seleccionar el signo "correcto" para la ecuación implícita de los conos  $S_i$ . En efecto, la transformación lineal de coordenadas reales a coordenadas canónicas nos garantiza que la normal a cada cono en los puntos extremos de su curva directriz es proporcional al vector normal prefijado en ese punto. Por lo tanto, para garantizar que ambos vectores tengan el mismo sentido, debemos escoger el signo de la ecuación implícita del cono que nos asegure que el factor



de proporcionalidad es positivo. Supongamos entonces que ya se ha asignado a la ecuación de cada cono  $S_i$  el signo "correcto". Entonces el A-parche correspondiente a un triángulo  $T \in \Delta$  se define como,

$$S = \{(X, Y, Z) : \sum_{i=0}^2 w_i(X, Y, Z) S_i(X, Y, Z) = 0\} \quad (5.23)$$

**Teorema 5.1** *El A-parche dado por (5.23) posee las siguientes propiedades*

- i)  $S \cap \Pi_j$  contiene la curva de la frontera  $c_j$ , para  $j = 0, 1, 2$ .
- ii) Si  $(X, Y, Z) \in c_j$ , entonces para  $j = 0, 1, 2$ ,  $\nabla S(X, Y, Z)$  es proporcional a  $\nabla S_j(X, Y, Z)$ .

### Demostración

- i) Sea  $(X, Y, Z) \in \Pi_j$ , para un  $j$  fijo,  $j = 0, 1, 2$ . Entonces,  $S(X, Y, Z) = w_j(X, Y, Z) S_j(X, Y, Z)$ . Pero por construcción,  $S_j \cap \Pi_j = c_j$ . Por lo tanto, si  $(X, Y, Z) \in c_j$  entonces  $S_j(X, Y, Z) = S(X, Y, Z) = 0$ , es decir la propiedad i) se cumple.
- ii) Si  $(X, Y, Z) \in c_j$ , entonces  $S_j(X, Y, Z) = 0$ . En consecuencia usando las propiedades de las funciones de peso dadas en el Lema 5.7 llegamos a que,  $\nabla S(X, Y, Z) = w_j(X, Y, Z) \nabla S_j(X, Y, Z)$

■

Finalmente, vamos a probar que haciendo uso de los A-parches contruidos, podemos definir una superficie A-spline  $G^1$  continua que interpola un conjunto finito de puntos en el espacio 3D.

**Teorema 5.2** *Sea  $\Delta$  una triangulación de un conjunto finito de puntos en el espacio 3D y sea  $A_s$  la superficie A-spline, que en cada triángulo  $T \in \Delta$  se define por medio del A-parche  $S$  dado por (5.23). Entonces  $A_s$  es una superficie  $G^1$  continua que interpola los vértices de la triangulación  $\Delta$ .*

**Demostración**

Puesto que  $A_s$  se reduce a un A-parche dado por (5.23) para cada triángulo  $T \in \Delta$ , sabemos que todas las curvas de la frontera de cada A-parche están contenidas en  $A_s$ . Pero por construcción, cada una de estas curvas pasa por dos de los puntos de interpolación. Por lo tanto,  $A_s$  interpola los puntos prefijados.

Por otro lado, cada curva de la frontera es común a dos parches vecinos, por lo cual  $A_s$  es una superficie continua. Más aún, según probamos en el Teorema 5.1, la normal a cada parche en un punto situado sobre una de sus curvas de la frontera, se reduce a la normal al bloque que contiene esta curva. Pero el bloque que contiene a una curva de la frontera es común para los dos parches vecinos que contienen la misma curva. Por lo tanto, dos parches vecinos comparten el mismo vector normal a lo largo de la curva de la frontera que tienen en común, en otras palabras la superficie A-spline  $A_s$  es  $G^1$  continua. ■

Nótese que como todos los algoritmos de su tipo, ASPLINT3D utiliza parches de grado elevado (6 ó 7 en nuestro caso), aunque de un tipo muy particular, de modo que un buen número de los coeficientes de la ecuación implícita de cada parche se sabe a priori que son nulos y en consecuencia no es necesario calcularlos. Además, a diferencia de otros algoritmos de su tipo, para calcular los coeficientes de cada parche, ASPLINT3D no requiere de la solución de sistemas de ecuaciones lineales muy grandes o con estructura muy complicada, basta con determinar la transformación lineal de coordenadas asociada a cada plano de la frontera.

**5.12 Graficación de la superficie A-spline**

Es bien conocido que una de las dificultades más serias asociadas con el uso de las superficies algebraicas definidas implícitamente está relacionada con los algoritmos para su graficación eficiente. Sin embargo, en los últimos años han aparecido varios algoritmos que proponen diferentes alternativas para la graficación de este tipo de superficies. La mayor parte de ellos se puede incluir en uno de los siguientes tipos:

- i) **Intersección con rayos** : Este tipo de algoritmos calcula los puntos de intersección entre los rayos que pasan por un punto fijo y la superficie

definida implícitamente. Para determinar los puntos sobre la superficie hay que calcular, para cada rayo, las raíces de un polinomio en una indeterminada del mismo grado que la superficie algebraica, y escoger entre ellas la raíz "apropiada". En [Sed89], Sederberg considera un algoritmo de este tipo e introduce una nueva base de polinomios, (la base piramidal de Bernstein) para calcular las raíces de los polinomios de forma más estable numéricamente.

- ii) **Poligonalización:** Los algoritmos de este tipo construyen una aproximación poligonal de la superficie implícita que se desea graficar, donde los vértices de los polígonos son puntos situados en la superficie. En los algoritmos más recientes, la malla de puntos se refina adaptativamente con el propósito de generar más puntos en las regiones donde la superficie es más compleja [Baj97].
- iii) **Subdivisión recursiva:** Esta clase de algoritmos se basa en realizar sucesivas subdivisiones de un cubo inicial en cubos o tetraedros contenidos dentro de este. Determinan entonces cuáles de ellos contienen una sección de la superficie y avanzan de un cubo o tetraedro "ocupado" al siguiente. En el interior de cada cubo o tetraedro, la superficie se aproxima por medio de un poliedro. Un ejemplo de este tipo de algoritmo es el presentado en [Tau94b]) que también puede utilizarse para graficar curvas algebraicas. El mismo parte de un cuadrado para graficar las curvas y de un cubo para graficar las superficies, los cuales se consideran como un pixel de baja resolución para el gráfico inicial. En cada paso posterior, se eliminan aquellos pixeles que no son cortados por la curva o superficie y se subdividen los que si pudieran ser cortados. De este modo, el algoritmo reduce el problema de graficar una curva o superficie algebraica, al problema de determinar si la curva o superficie en cuestión pasa por un cuadrado o cubo respectivamente.

Para graficar una superficie A-spline como la que hemos construido a lo largo de este capítulo, nosotros utilizamos un algoritmo que se basa en el cálculo de las intersecciones de la superficie con un haz de rayos, o sea un algoritmo que pertenece a la primera clase. En el mismo se introducen adicionalmente, algunos criterios para aislar la hoja conexas de cada A-parche que contiene las curvas de la frontera prefijadas.

A continuación vamos a describir los principales pasos del algoritmo para graficar cada uno de los A-parches que definen el A-spline. En primer lugar,

se selecciona un valor entero  $n$ , que está vinculado con la cantidad de rayos que se utilizarán para intersecar cada A-parche. Supongamos entonces que se desea graficar el A-parche correspondiente al triángulo con vértices  $P_0, P_1, P_2$ . Para el valor de  $n$  dado, se construye una malla triangular de

$$m = (n + 1)(n + 2)/2$$

puntos  $V_i$ ,  $i = 1, \dots, m$  situados en el triángulo y se selecciona además un punto  $Q$  en  $\mathbb{R}^3$ .

Para  $i = 1, \dots, m$ , sea  $r_i$  la recta que pasa por los puntos  $Q$  y  $V_i$ , que escriberemos en forma paramétrica como,

$$r_i : Q + td_i \quad (5.24)$$

donde  $d_i$  es el vector  $d_i = (V_i - Q) / \|V_i - Q\|$ .

Sustituyendo la ecuación paramétrica (5.24) de la recta  $r_i$  en las ecuaciones implícitas de cada uno de los bloques  $S_0, S_1$  y  $S_2$  y de los planos de la frontera  $\Pi_0, \Pi_1$  y  $\Pi_2$ , es fácil calcular los coeficientes del polinomio en una indeterminada,

$$p_i(t) := S(r_i(t)) = 0 \quad (5.25)$$

al que nos referiremos, en lo adelante, como **polinomio de disparo**. Este polinomio es de grado 6 cuando todos los bloques son conos cuadráticos, o sea cuando las configuraciones de todas las aristas del triángulo base sean convexas y de grado 7 en otro caso.

Para cada rayo o "disparo"  $r_i$ , si el punto  $Q$  se escoge convenientemente, entonces existe un único punto sobre la hoja conexa del A-parche que contiene las curvas de la frontera prefijadas. Las coordenadas  $R_i$  de este punto se pueden calcular sustituyendo en (5.24),  $t = t_i^*$  donde  $t_i^*$  es la raíz "correcta" del polinomio de disparo (5.25). Para seleccionar esta raíz se procede de la siguiente forma:

- i) Para cada cono  $S_0, S_1$  y  $S_2$  se determina el punto "correcto" de intersección del rayo  $r_i$  con el cono (vea la figura 5.5). Aquí el término "correcto" se refiere a un punto sobre el cono, que está situado sobre alguna recta que parte de su vértice y termina en el segmento de la curva directriz del cono que pertenece a la frontera del A-parche.

Como los conos son superficies regladas, en nuestro algoritmo de grado 2 ó 3, es fácil determinar cuál es la raíz del polinomio  $S_j(r_i(t))$ ,  $j =$

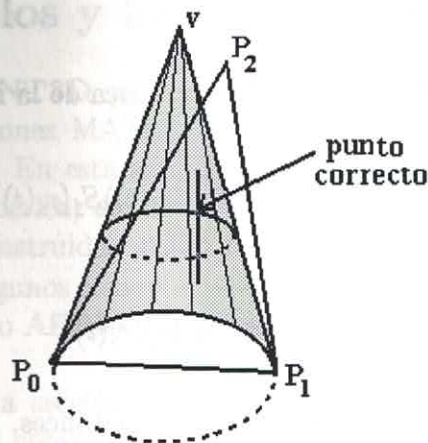


Figura 5.5: Selección de la raíz "correcta" del polinomio de disparo

$0, 1, 2, i = 1, \dots, m$  asociada al punto "correcto" sobre el cono  $S_j$ . Denotémosla por  $t_j^i$  y sean,

$$t_{min}^i = \min\{t_0^i, t_1^i, t_2^i\}$$

$$t_{max}^i = \max\{t_0^i, t_1^i, t_2^i\}$$

- ii) Si  $S(t_{min}^i)S(t_{max}^i) \leq 0$ , entonces existe al menos una raíz real del polinomio de disparo  $p_i(t)$  en  $[t_{min}^i, t_{max}^i]$ . De lo contrario, es necesario calcular todas sus raíces reales. En ambos casos, la raíz "correcta"  $t_i^*$ , es decir la raíz correspondiente al punto  $R_i$  situado en la hoja del A-parche que contiene las curvas de la frontera, se determina escogiendo entre todos los posibles candidatos (o sea los puntos asociados a cada raíz real) aquel que se encuentra a distancia mínima, de otros puntos vecinos sobre el A-parche que ya han sido previamente calculados.

**Teorema 5.3** Sea  $Q$  el punto de intersección de los planos de la frontera  $\Pi_0, \Pi_1$  y  $\Pi_2$ . Entonces el polinomio de disparo (5.25) se puede factorizar como  $p_i(t) = t^4 q_i(t)$ , donde  $q_i(t)$  es un polinomio de grado 2 ó 3.

**Demostración**

Sustituyendo en (5.23) la ecuación paramétrica de la recta (5.24) obtenemos,

$$p_i(t) := S(r_i(t)) = \sum_{j=0}^2 w_j(r_i(t)) S_j(r_i(t))$$

Pero de (5.22) tenemos que

$$w_j(r_i(t)) = \Pi_l(r_i(t))^2 \Pi_k(r_i(t))^2 \quad (5.26)$$

para  $l, k \neq j$  con  $j, l, k \in \{0, 1, 2\}$

Sea  $Q = (Q_x, Q_y, Q_z)$  y  $d_i = (d_{ix}, d_{iy}, d_{iz})$ . Entonces, teniendo en cuenta que  $M_k$  es la normal al plano  $\Pi_k$  con  $k = 0, 1, 2$  y que en consecuencia podemos escribir la ecuación implícita de  $\Pi_k$  como,

$$M_k(X - X_k, Y - Y_k, Z - Z_k)^t = 0$$

llegamos a que,

$$\begin{aligned} \Pi_k(r_i(t)) &= M_k(Q_x + td_{ix} - X_k, Q_y + td_{iy} - Y_k, Q_z + td_{iz} - Z_k)^t \\ &= M_k(Q_x - X_k, Q_y - Y_k, Q_z - Z_k) + t M_k(d_{ix}, d_{iy}, d_{iz})^t \end{aligned} \quad (5.27)$$

Pero  $Q$  es el punto de intersección de los planos de la frontera. Por lo tanto,  $\Pi_k(Q) = 0$ , es decir el primer sumando en (5.27) se anula y obtenemos,

$$\Pi_k(r_i(t)) = \alpha_{ik} t \quad (5.28)$$

donde  $\alpha_{ik} = M_k d_i^t$ . Sustituyendo (5.28) en (5.26) se tiene entonces que,

$$w_j(r_i(t)) = (\alpha_{il} t)^2 (\alpha_{ik} t)^2 = q_{ij} t^4$$

donde  $q_{ij} = \alpha_{il}^2 \alpha_{ik}^2$  con  $l, k \neq j$  y  $j, k, l \in \{0, 1, 2\}$ . En consecuencia,

$$S(r_i(t)) = t^4 \sum_{j=0}^2 q_{ij} S_j(r_i(t)) = t^4 q_i(t)$$

donde  $q_i(t)$  es un polinomio de grado 2 ó 3. ■

Nótese que de acuerdo con el Teorema anterior, si el vértice  $Q$  del haz de rayos se escoge como el punto de intersección de los planos de la frontera  $\Pi_0, \Pi_1$  y  $\Pi_2$ , entonces para calcular un punto  $R_i$  sobre la hoja del A-parche que contiene las curvas de la frontera, basta con calcular las raíces reales del polinomio  $q_i(t)$  (que son a los sumo 3) y escoger entre ellas la "correcta".

## 5.13 Ejemplos y Experimentación numérica

El algoritmo **ASPLINT3D** que se presenta en este capítulo fue programado a través de 37 funciones MATLAB (versión 5.3) y probado con un buen número de ejemplos. En esta sección vamos a mostrar los resultados de la experimentación numérica, así como los gráficos de algunas superficies A-splines que fueron construidas utilizando el algoritmo.

A continuación algunos comentarios de interés acerca de la instrumentación práctica del algoritmo **ASPLINT3D**.

- i) Al final de la sección anterior vimos que desde el punto de vista teórico, resulta conveniente tomar como vértice  $Q$  del haz de rayos para graficar cada parche, el punto de intersección de los planos de la frontera, pues de esta manera logramos reducir el problema al cálculo de las raíces de un polinomio de grado 2 ó 3, en vez del original **polinomio de disparo** cuyo grado es 6 ó 7.
- ii) En la práctica, sin embargo, pudimos comprobar que la calidad del gráfico depende más de la posición del punto  $Q$  en relación con el A-parche, que del grado del polinomio cuyas raíces debemos calcular. En efecto, cuando los planos de la frontera se intersecan en un punto que se encuentra distante de la recta perpendicular al triángulo base y que pasa por baricentro de éste, entonces algunos rayos que parten de  $Q$  pueden cortar a la superficie en más de un punto situado sobre la hoja "correcta" del A-parche (o sea aquella que contiene las curvas de la frontera prefijadas). En ese caso, resulta muy difícil determinar cuál es la raíz "correcta" del polinomio de disparo, ya que en realidad más de una raíz pertenecen a la hoja "correcta" del parche.
- iii) Para resolver este problema, en nuestro programa tomamos  $Q$  como el punto al infinito en la dirección perpendicular al plano base. En otras palabras, cortamos cada parche con rectas que parten de los puntos  $V_i$  del triángulo base (vea sección 5.12) y que son perpendiculares al mismo. Cada una de estas rectas suele cortar al A-parche en puntos que pertenecen a diferentes hojas, por lo cual resulta más fácil determinar cuál es la raíz "correcta". Esto se evidencia en el hecho de que en la inmensa mayoría de las veces, el polinomio de disparo  $p_i(t)$ , asociado al rayo o disparo  $r_i$ , cambia de signo en el intervalo  $[t_{min}^i, t_{max}^i]$  y más

aún, sólo posee una raíz real en ese intervalo, que es por lo tanto la raíz "correcta".

En resumen, el método que se propone en este capítulo para graficar la superficie A-spline, se basa en calcular sus intersecciones con rayos que parten del triángulo base y que son perpendiculares al mismo. Esto requiere calcular, para cada rayo o disparo, las raíces reales de un polinomio de a lo sumo grado 7. En la práctica, sin embargo, casi nunca es necesario calcular todas las raíces, ya que de acuerdo con nuestra experiencia, si el vértice de cada cono se escoge de forma adecuada y los rayos con que se corta cada parche son perpendiculares al triángulo base, entonces la raíz deseada se encuentra en un intervalo cerrado, que se determina a partir del punto donde cada rayo corta a cada uno de los 3 conos.

### 5.13.1 Experimentación numérica

Como ya hemos mencionado con anterioridad, no disponemos de una condición suficiente que nos garantice que el polinomio de disparo  $p_i(t)$ , asociado al rayo  $r_i$ , cambia de signo en el intervalo  $[t_{min}^i, t_{max}^i]$ . Sin embargo, en esta sección vamos mostrar experimentalmente, que si el vértice del haz de rayos con los cuales vamos a intersectar el A-parche, es el punto al infinito situado en la dirección perpendicular al plano base, entonces con mucha frecuencia ocurre que el polinomio de disparo no sólo cambia de signo en el intervalo mencionado, sino que además posee una única raíz real en el mismo.

Los resultados obtenidos con nuestro programa MATLAB para construir y graficar el A-spline de interpolación, se muestran en la tabla 1. La primera columna de la tabla se refiere al fichero de datos que se utilizó en el experimento. En la segunda, se muestra el valor de  $n$  y en la tercera el número de rayos "interiores" con los cuales se intersectó el A-parche. En este caso, el término "interior" se refiere a un rayo perpendicular al plano base, que pasa por un punto situado en el interior del triángulo correspondiente. La cantidad de cambios de signo y el número de raíces reales que se cuantifican en las dos siguientes columnas, se deben comparar con el número de rayos interiores, pues la frontera del A-parche se graficó a partir de la curvas que la definen.

En aras de brevedad, en la tabla 1 sólo incluimos los resultados de 7 de los 16 parches del elipsoide cuyos datos se encuentran en el fichero elisoipde41 (vea también el ejemplo 4 de la próxima sección). Como se puede apreciar,



para los parches del 3 al 7 de este A-spline, hay un polinomio de disparo que no cambia de signo en el intervalo  $[t_{min}^i, t_{max}^i]$  correspondiente a algún rayo  $r_i$  con  $1 \leq i \leq 6$ . Sin embargo, en todos estos casos resultó que  $t_{min}^i$  es una raíz aproximada del polinomio  $p_i(t)$  aunque  $p_i(t_{min}^i)p_i(t_{max}^i) > 0$ .

Fichero de datos	n	Rayos interiores	A-parche	Cambios de signo en $[t_{min}^i, t_{max}^i]$	Raíces en $[t_{min}^i, t_{max}^i]$
datQ2	5	6	1	6	6
datQ4	6	10	1	10	10
datQ5	5	6	1	6	6
superev2	5	6	1	6	6
datC1	7	15	1	15	15
datC3	5	6	1	6	6
datC5	5	6	1	6	6
superev2	5	6	1	6	6
datQ3	5	6	1	6	6
	5	6	2	6	6
superev21	8	21	1	21	21
	8	21	2	21	21
superev6	5	6	1	6	6
	5	6	2	6	6
	5	6	3	6	6
	5	6	4	6	6
elipsoide41	5	6	1	6	6
	5	6	2	6	6
	5	6	3	5	6
	5	6	4	5	6
	5	6	5	5	6
	5	6	6	5	6
	5	6	7	5	6

Tabla 1 Cambios de signos y raíces del polinomio de disparo

## 5.13.2 Ejemplos

**Ejemplo 1** En este ejemplo mostramos algunos de los resultados de la corrida del programa para construir y graficar el A-spline, que interpola los puntos y normales siguientes,

$$\begin{aligned} P_0 &= (0, 0, 2) & N_0 &= (0, 0, 1) \\ P_1 &= (0, 1, 0) & N_1 &= (0, 2, 0) \\ P_2 &= (3, 0, 0) & N_2 &= (0.6667, 0, 0) \end{aligned}$$

Cada pareja de vértices del triángulo base define una configuración convexa. Por lo tanto, todas las curvas de la frontera son cónicas y los bloques correspondientes son conos cuadráticos. La figura 5.6 muestra los 3 conos cuadráticos y el A-parche obtenido mediante el algoritmo **ASPLINT3D** tomando  $w_1 = 0.7071$  para todas las cónicas de la frontera y  $n = 5$  para la graficación del A-parche.

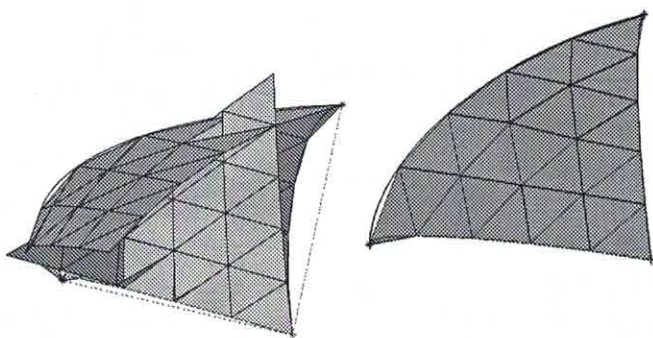


Figura 5.6: Tres conos cuadráticos y el A-parche correspondiente

**Ejemplo 2** Para este ejemplo escogimos 4 puntos situados sobre un elipsoide y calculamos las correspondientes direcciones normales. Puesto que los datos provienen de un elipsoide, resulta natural esperar que todas las curvas de la frontera de cada uno de los dos parches sea convexa, es decir

una cónica, como en efecto sucede. La figura 5.7 muestra los dos triángulos, las normales y el esqueleto o malla alambre de cada parche. Además incluye el gráfico del A-spline obtenido tomando  $n = 8$  y el parámetro libre  $w_1 = 0.7071$  para todas las cónicas de la frontera.

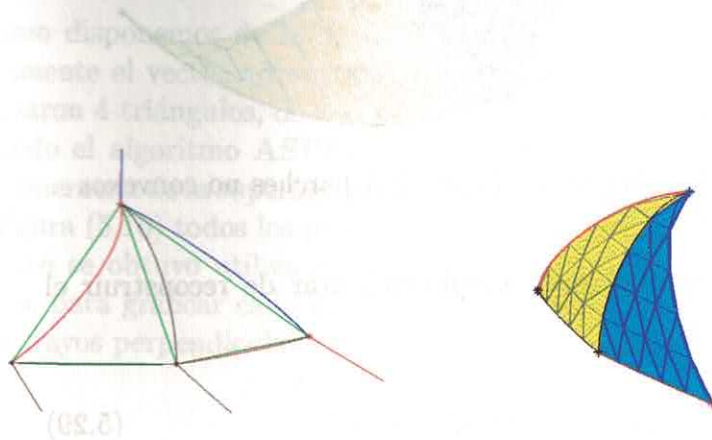


Figura 5.7: Esqueleto y A-spline formado por 2 parches convexos

**Ejemplo 3** Los 4 puntos de este ejemplo y las correspondientes direcciones normales son los siguientes,

$$\begin{array}{ll}
 P_0 = (2, 0, 0) & N_0 = (2, 0, -1) \\
 P_1 = (.01, .01, .028) & N_1 = (1.355, 1.355, -1) \\
 P_2 = (0, 2, 0) & N_2 = (0, 2, -1) \\
 P_3 = (-2, 0, 0) & N_3 = (-2, 0, -1)
 \end{array}$$

Como se puede apreciar en la figura 5.8 (izquierda) en ambos parches de este ejemplo aparecen aristas cuya configuración es no convexa. En ese caso, las curvas de la frontera son cúbicas y tomamos  $w_1$  y  $w_2$  de signos contrarios y con valor absoluto 0.05. En el resto de las aristas, las curvas de la frontera son cónicas. La parte derecha de la figura muestra el gráfico del A-spline, obtenido utilizando  $n = 8$ , es decir intersectando cada parche con  $m = 45$  rayos perpendiculares a cada plano base.

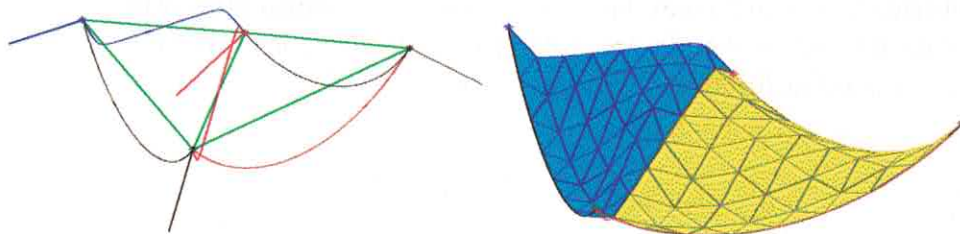


Figura 5.8: Esqueleto y A-spline formado por 2 A-parches no convexos

**Ejemplo 4** El objetivo de este ejemplo es tratar de reconstruir el elipsoide,

$$(x/3)^2 + (y/3)^2 + (z/2.5)^2 - 1 = 0 \quad (5.29)$$

a partir de su “polo norte” y su “polo sur”, más 8 puntos situados en la intersección del elipsoide con el plano  $z = 0$ . Utilizando estos 10 puntos definimos una triangulación de la superficie, que sólo consta de 16 triángulos y a partir de la ecuación implícita del elipsoide calculamos la normal en cada uno de los puntos escogidos. La figura 5.9 muestra los 5 primeros A-parches del A-spline calculado, tomando  $n = 8$ . Todas las curvas de la frontera son cónicas ( con  $w_1 = 0.7071$ ), lo cual corresponde al hecho de que los datos fueron extraídos de un elipsoide.

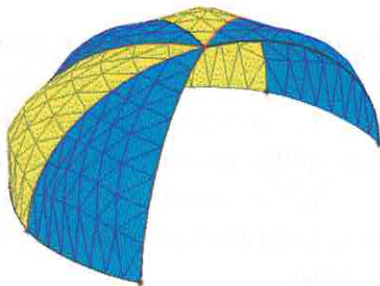


Figura 5.9: A-spline definido por 5 A-parches convexos

**Ejemplo 5** Los 6 puntos que se utilizaron como datos de este ejemplo pertenecen a la superficie de revolución

$$z = 1 + \text{sen}(-1.2 + \sqrt{x^2 + y^2}) \quad (5.30)$$

Como disponemos de la ecuación implícita de la superficie, calculamos directamente el vector normal en cada punto escogido. Con estos 6 puntos se formaron 4 triángulos, de modo que la superficie A-spline que calculamos utilizando el algoritmo **ASPLINT3D** está formada por 4 A-parches. La curva generatriz de la superficie (5.30) es una cúbica y como se puede apreciar en la figura (5.10) todos los parches que definen el A-spline son no convexos. El gráfico se obtuvo utilizando  $n = 8$  en el algoritmo **ASPLINT3D**. Por lo tanto, para graficar cada parche se calculó la intersección del mismo con  $m = 45$  rayos perpendiculares al plano base correspondiente.

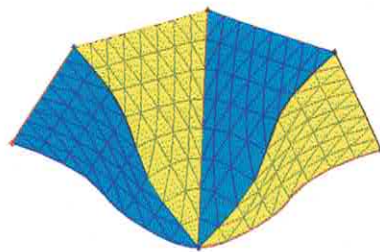


Figura 5.10: A-spline formado por 4 A-parches no convexos

## Capítulo 6

# Conclusiones y Recomendaciones

A través del desarrollo de este trabajo hemos mostrado que las curvas y superficies A-splines también puede utilizarse con éxito para resolver diferentes problemas de modelación geométrica. Desde nuestro punto de vista, esto significa que un sistema ideal de diseño geométrico computarizado debería incluir tanto las curvas y superficies splines parametrizadas racionalmente, como las curvas y superficies A-splines, pues cada una de ellas es óptima para resolver ciertas tareas específicas.

En particular, en nuestro trabajo probamos cómo utilizar las curvas y superficies A-spline para resolver tres problemas diferentes de modelación geométrica.

En primer lugar, se propuso un algoritmo para aproximar el contorno de una imagen digital, por medio de una curva A-spline cónica. Aunque algoritmos similares han sido reportados previamente en la literatura, fue necesario hacer un estudio detallado de cada uno de los pasos fundamentales e introducir ciertas adaptaciones que nos permitieran resolver exitosamente el problema, teniendo en cuenta que los puntos que pertenecen al contorno de la imagen digital tienen coordenadas enteras y por lo tanto poseen con frecuencia distribuciones especiales, tipo "escalera".

En segundo lugar, obtuvimos una nueva familia de curvas A-spline  $G^2$  continuas con una serie de propiedades que las hacen óptimas para el diseño libre. Entre ellas vale la pena mencionar las siguientes. La familia propuesta posee varios parámetros libres que se pueden utilizar de manera intuitiva como asas geométricas para controlar la forma de cada uno de los segmentos

de la curva A-spline. Por otro lado, la condición de  $G^2$  continuidad de la curva A-spline se logra sin conocer los valores de la curvatura en los puntos de interpolación y sin necesidad de resolver un sistema de ecuaciones lineales o no lineales. Otras de las ventajas indiscutibles del esquema de interpolación propuesto es su carácter local que le permite al diseñador modificar una sección de la curva A-spline, sin que sea necesario reclacular toda la curva, sino exclusivamente las dos secciones vecinas.

Por último, en la Tesis se obtuvo una familia de superficies A-spline,  $G^1$  continuas, que interpolan un conjunto de puntos en el espacio, vértices de una triangulación 3D. Usando esta familia de curvas, se propuso un algoritmo de interpolación que construye cada parche de la superficie A-spline definiendo previamente sus curvas de la frontera. Gracias a esta estrategia, y a la introducción de un sistema adecuado de coordenadas, asociado con cada uno de los planos de la frontera que contiene a estas curvas, es posible calcular los coeficientes de cada parche sin necesidad de resolver un sistema global de ecuaciones lineales. En otros algoritmos similares, el cálculo de los coeficientes de la superficie de interpolación requiere la solución de un sistema de ecuaciones lineales, cuyo tamaño puede ser grande y cuya estructura puede ser complicada. Por otro lado, el hecho de que cada parche de la superficie A-spline dependa esencialmente de sus curvas de la frontera nos garantiza que el esquema de interpolación propuesto tiene un carácter local, es decir que si varía la posición de un punto de interpolación entonces se afectan solamente los parches del A-spline que pasan por ese punto.

Sin lugar a dudas una conclusión importante de esta Tesis es que debido a las ventajas que tiene para ciertos problemas específicos el uso de las curvas y superficies algebraicas definidas implícitamente, es necesario continuar buscando soluciones a diferentes subproblemas que todavía obstaculizan su empleo en la modelación geométrica. Entre ellos podemos mencionar los siguientes:

- El problema de generar un conjunto de puntos sobre una curva o superficie algebraica definida implícitamente que sea “representativo” de la misma, o más aún, cuya distribución sea conveniente, por ejemplo distribución uniforme según la longitud de arco, distribución proporcional a la curvatura, etc.
- El problema de encontrar un método eficiente para “aislar” la rama u hoja de la curva o superficie implícita que está en la región de interés

y satisface las condiciones de interpolación impuestas por la tarea de modelación. En otras palabras, la necesidad de encontrar algoritmos eficientes para generar un número suficientemente grande de puntos sobre la componente conexa de la curva o superficie que va a ser utilizada.

La solución de los problemas anteriores nos permitiría definir un algoritmo que, utilizando únicamente curvas y superficies A-splines, nos permita reconstruir la superficie de un objeto, del cual se conocen sus intersecciones con planos paralelos. En efecto, utilizando el algoritmo propuesto en el capítulo 3 podemos definir los contornos de cada una de las secciones transversales del objeto. Más adelante tendríamos que escoger un conjunto de puntos "representativos" de las curvas A-spline que describen cada uno de los contornos. El siguiente paso sería construir una triangulación 3D, de modo que cada triángulo posea vértices en dos contornos consecutivos. Finalmente, utilizando el método propuesto en el capítulo 5, podríamos definir una superficie A-spline  $G^1$  continua que interpole los vértices de la triangulación y que será nuestra aproximación a la superficie que se desea reconstruir.



## Anexo A

### Parches triangulares de Bezier

Un parche triangular de Bezier de grado  $n$  es una *superficie paramétrica*  $b_n(\mathbf{u})$  definida para  $\mathbf{p} \in \mathbb{E}^2$  en el interior de un triángulo dominio y escrita en términos de las coordenadas baricéntricas  $\mathbf{u}$  de  $\mathbf{p}$ , con respecto a los vértices del triángulo dominio. La geometría de un parche triangular de Bezier depende de su *mallado de control* que debe ser previamente definida y que está formada por  $\frac{1}{2}(n+1)(n+2)$  puntos  $b_{ijk} \in \mathbb{E}^3$  con  $i+j+k=n$ . Usualmente estos puntos se organizan en un arreglo triangular, como el que se muestra a continuación para  $n=3$ ,

$$\begin{array}{c} b_{030} \\ b_{021} \ b_{120} \\ b_{012} \ b_{111} \ b_{210} \\ b_{003} \ b_{102} \ b_{201} \ b_{300} \end{array}$$

Sea  $\mathbf{i} = (i, j, k)$  y  $|\mathbf{i}| = i + j + k$ . Denotemos por  $\mathbf{b}_i$  el punto  $b_{ijk}$  de la mallado de control. Un parche triangular de grado  $n$  se escribe en términos de los polinomios de Bernstein bivariados de igual grado  $B_i^n(\mathbf{u})$  dados por,

$$B_i^n(\mathbf{u}) = \binom{n}{\mathbf{i}} u^i v^j w^k = \frac{n!}{i!j!k!} u^i v^j w^k$$

con  $|\mathbf{i}| = n$ . Aquí se está suponiendo que  $B_i^n(\mathbf{u}) = 0$  si alguno de los índices  $(i, j, k)$  es negativo. Por ejemplo, para  $n=3$  los polinomios cúbicos de Bernstein, que se organizan en correspondencia con los puntos de la mallado de control, están dados por,

$$\begin{array}{c}
 v^3 \\
 3v^2w \quad 3uv^2 \\
 3vw^2 \quad 6uvw \quad 3u^2v \\
 w^3 \quad 3uw^2 \quad 3u^2w \quad u^3
 \end{array}$$

Finalmente, para  $\mathbf{p} \in \mathbb{E}^2$  con coordenadas baricéntricas  $\mathbf{u}$ , y una malla de control formada por los puntos  $\mathbf{b}_i$  con  $|\mathbf{i}| = n$ , el punto correspondiente sobre el parche triangular de Bezier se obtiene mediante,

$$\begin{aligned}
 b_n(\mathbf{u}) &= \sum_{|\mathbf{j}|=n} \mathbf{b}_j B_j^n(\mathbf{u}) \\
 &= \sum_{j=0}^n \sum_{i=0}^{n-j} b_{i,j,n-i-j} B_{i,j,n-i-j}^n(\mathbf{u})
 \end{aligned} \tag{A.1}$$

En particular, un *parche no paramétrico* definido mediante la función,

$$z = \sum_{|\mathbf{i}|=n} c_i B_i^n(\mathbf{u}) \tag{A.2}$$

donde  $c_i \in \mathbb{E}^1$  se puede escribir como un parche triangular de Bezier. Para eso basta con tener en cuenta que los polinomios de Bernstein satisfacen la siguiente propiedad [Far93],

$$u = \sum_{|\mathbf{i}|=n} \frac{i}{n} B_i^n(\mathbf{u}), \quad v = \sum_{|\mathbf{i}|=n} \frac{j}{n} B_i^n(\mathbf{u}), \quad w = \sum_{|\mathbf{i}|=n} \frac{k}{n} B_i^n(\mathbf{u})$$

Por lo tanto, la función (A.2) se puede expresar explícitamente como,

$$\begin{pmatrix} u \\ v \\ w \\ z \end{pmatrix} = \sum_{|\mathbf{i}|=n} \begin{pmatrix} i/n \\ j/n \\ k/n \\ c_i \end{pmatrix} B_i^n(\mathbf{u}) \tag{A.3}$$

En otras palabras, la malla de control de un parche no paramétrico está formada por puntos cuyas coordenadas  $(u, v, w) = \frac{1}{n}(i, j, k)$ , definen una  $n$ -partición del triángulo dominio.

## Anexo B

### Demostraciones

En este anexo incluimos demostraciones que pertenecen a otros autores, con el objetivo de hacer nuestro trabajo autocontenido.

#### Demostración del lema 2.1

i) A continuación vamos a probar que bajo las hipótesis del lema 2.1  $c_i$  es una curva conexa, lo que es equivalente a mostrar que si  $0 < \beta < 1$ , entonces la recta  $u = \beta(1 - v)$  corta a la curva  $f_i(u, v)$  en un sólo punto en el interior del triángulo canónico. Nótese que al variar  $\beta$  en el intervalo  $(0, 1)$  tenemos un haz de rectas que pasan por el punto  $p_1 = (0, 1)$  y que cortan el segmento  $p_0p_2$  del triángulo canónico. Para determinar la intersección entre la curva y una recta del haz, sustituimos la ecuación de la recta en  $f_i(u, v)$ , obteniendo el polinomio en una indeterminada,

$$\begin{aligned} B_\beta(v) &= f_i(\beta(1 - v), v) \\ &= \sum_{j=0}^n \sum_{k=0}^{n-j} \frac{n!}{k!j!(n-k-j)!} a_{kj} (\beta(1 - v))^k v^j (1 - \beta(1 - v) - v)^{n-k-j} \\ &= \sum_{j=0}^n \left( \sum_{k=0}^{n-j} a_{kj} \frac{(n-j)!}{k!(n-k-j)!} \beta^k (1 - \beta)^{n-k-j} \right) \frac{n!}{j!(n-j)!} v^j (1 - v)^{n-j} \\ &= \sum_{j=0}^n b_j(\beta) B_j^n(v) \end{aligned}$$

donde

$$b_j(\beta) = \sum_{k=0}^{n-j} a_{kj} B_k^{n-j}(\beta)$$

y  $B_j^m$  son los polinomios de Bernstein de grado  $m$  dados por

$$B_j^m(v) = \frac{m!}{j!(m-j)!} v^j (1-v)^{m-j}$$

De las hipótesis tenemos que:

$$\begin{aligned} b_0(\beta) &> 0 \\ b_j(\beta) &\geq 0 \quad \text{para } j = 1, \dots, l-1 \\ b_j(\beta) &\leq 0 \quad \text{para } j = l+1, \dots, n \end{aligned}$$

Supongamos que  $b_n(\beta) = \dots = b_{n-(p+1)}(\beta) = 0$ , pero  $b_{n-p}(\beta) < 0$  para algún  $p$  tal que  $0 \leq p \leq n-l-1$ . Entonces  $B_\beta(v)$  se puede escribir como,

$$\begin{aligned} B_\beta(v) &= \sum_{j=0}^{n-p} b_j(\beta) \frac{n!}{j!(n-j)!} v^j (1-v)^{n-j} \\ &= (1-v)^p \sum_{j=0}^{n-p} \frac{b_j(\beta) n!}{(n-p)!(n-j)! j!(n-p-j)!} v^j (1-v)^{n-j-p} \\ &= (1-v)^p \sum_{j=0}^{n-p} c_j(\beta) B_j^{n-p}(v) \end{aligned}$$

donde  $c_j = \frac{b_j(\beta) n!}{(n-p)!(n-j)!}$  para  $j = 0, \dots, n-p$ . Por lo tanto,  $c_0(\beta) > 0$  y  $c_{n-p}(\beta) < 0$ , mientras que para cada  $\beta$  fijo, la sucesión  $c_0(\beta), c_1(\beta), \dots, c_{n-p}(\beta)$  solamente cambia de signo una vez. Como la función  $(1-v)^p$  no se anula en el interior del triángulo canónico, un punto de intersección entre la recta  $u = \beta(1-v)$  y la curva  $f_i(u, v) = 0$  tiene que ser un cero de la función de Bezier  $C_\beta(v) := \sum_{j=0}^{n-p} c_j(\beta) B_j^{n-p}(v)$ . Para cada  $\beta$  en  $(0, 1)$ , ésta es una función de Bezier cuyo polígono de control está formado por los puntos  $(j/(n-p), c_j(\beta))$  con  $j = 0, \dots, n-p$ . En consecuencia, teniendo en cuenta la propiedad de variación reducida de las curvas de Bezier [Far93], podemos afirmar que la función de Bezier  $C_\beta(v)$  tiene a lo sumo una raíz para  $v \in (0, 1)$ . Por otro

lado, como  $C_\beta(v)$  es una función continua en  $(0,1)$  y  $C_\beta(0) = c_0(\beta) > 0$ , mientras que  $C_\beta(1) = c_{n-p}(\beta) < 0$ , entonces la función  $B_\beta(v)$  tiene al menos un cero en  $(0,1)$ . Luego,  $B_\beta(v)$  tiene exactamente una raíz en  $(0,1)$ , es decir la curva  $c_i$  y la recta  $u = \beta(1-v)$  se intersecan exactamente en un punto en el interior del triángulo canónico. No es difícil probar que las coordenadas del punto de intersección entre la curva  $c_i$  y la recta  $u = \beta(1-v)$ , dependen continuamente del parámetro  $\beta$ . Por lo tanto, el segmento de la curva  $c_i$  contenido en el interior del triángulo canónico es conexo.

ii) A continuación vamos a probar que  $c_i$  no tiene puntos singulares en el interior del triángulo canónico. Supongamos lo contrario, o sea que  $(u^*, v^*)$  es un punto singular de  $c_i$  situado en el interior del triángulo canónico. Entonces,

$$\frac{\partial f_i}{\partial u}(u^*, v^*) = \frac{\partial f_i}{\partial v}(u^*, v^*) = 0$$

Sea  $\beta^* = \frac{u^*}{(1-v^*)}$ ,  $0 < \beta^* < 1$ . Consideremos la intersección entre la curva  $c_i$  y la recta del haz  $u = \beta(1-v)$  con  $\beta = \beta^*$ . Sea  $B_{\beta^*}(v)$  el polinomio en  $v$  indeterminada que se obtiene sustituyendo  $u = \beta^*(1-v)$  en la ecuación de  $c_i$ . Entonces,

$$B_{\beta^*}(v^*) = f_i(\beta^*(1-v^*), v^*) = f_i(u^*, v^*) = 0$$

Por otro lado,

$$\frac{dB_{\beta^*}}{dv}(v^*) = \left[ -\frac{\partial f_i}{\partial u}\beta^* + \frac{\partial f_i}{\partial v} \right] (u^*, v^*) = 0$$

es decir,  $v^*$  es un cero doble del polinomio  $B_{\beta^*}(v)$ . Esto significa que podemos encontrar un valor de  $\beta$  suficientemente cercano a  $\beta^*$ , tal que el correspondiente polinomio univariado  $B_\beta(v)$  posee dos raíces diferentes muy próximas a  $v = v^*$ , lo cual contradice el hecho de que toda recta del haz  $u = \beta(1-v)$ , corta a la curva  $c_i$  en un único punto en el interior del triángulo canónico. ■

### Demostración del lema 4.1

Teniendo en cuenta la relación que existe entre las coordenadas afines  $(x, y)$  y las coordenadas baricéntricas  $(u, v)$  dada por (2.5), para mostrar que la curva  $c_i$  pasa por los puntos  $P_0^i$  y  $P_2^i$  basta con verificar que  $f_i(1, 0) = 0$  y  $f_i(0, 0) = 0$  respectivamente. Esto resulta inmediato de (4.2). Veamos ahora que la curva  $c_i$  es tangente a los lados  $p_0p_1$  y  $p_1p_2$  del triángulo canónico. En efecto, como

$$\frac{\partial f_i}{\partial u}(0, 0) \neq 0 \quad \frac{\partial f_i}{\partial v}(0, 0) = 0$$

la recta tangente a  $c_i$  en el punto  $p_0 = (0, 0)$  es  $u = 0$ , que corresponde al lado  $p_0p_1$  del triángulo canónico, el cual a su vez se transforma en  $P_1^iP_2^i$  mediante el cambio de coordenadas baricéntricas a coordenadas reales. Análogamente se puede probar que la recta tangente a  $c_i$  en el punto  $p_2 = (1, 0)$  es la recta  $1 - u - v = 0$  que corresponde al lado  $p_1p_2$  del triángulo canónico. A su vez esta recta se transforma en  $P_0^iP_1^i$  mediante el cambio de coordenadas baricéntricas a coordenadas reales.

Las hipótesis de este lema son un caso particular de las hipótesis del lema 2.1 ( $n = 3, l = 1$ ) para el caso de una curva cúbica. Por lo tanto, podemos afirmar que  $c_i$  es una curva conexa en el interior de  $T_i$  y que no tiene puntos singulares en ese triángulo.

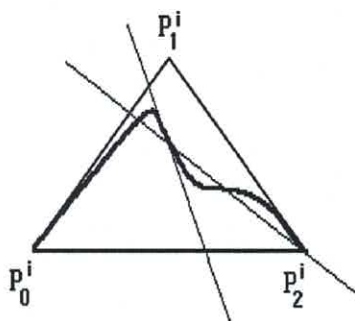


Figura B.1: La cúbica  $c_i$  no puede tener ningún punto de inflexión en el interior del triángulo

Finalmente, para probar que la cúbica  $c_i$  es convexa basta con verificar que no tiene ningún punto de inflexión en el interior de  $T_i$ . Para ello, se

supone lo contrario, es decir que existe al menos un punto de inflexión (vea la figura B.1 tomada de [Baj99a]) y se prueba que existe una recta que pasa por el punto de inflexión y que corta a la cúbica en 4 puntos, lo cual contradice el teorema de Bezout [Walk78].

### Demostración del lema 4.2

La demostración completa de este resultado se puede encontrar [Pal92]. Aquí sólo damos las ideas fundamentales, sin incluir todos los detalles.

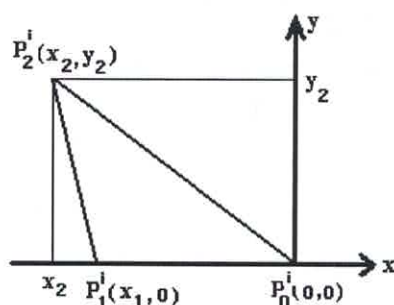


Figura B.2: Traslación del triángulo  $T_i$  para calcular la curvatura

Como la curvatura de una curva es invariante por rotaciones y traslaciones, sin perder generalidad podemos suponer que los vértices  $P_0^i, P_1^i, P_2^i$  del triángulo  $T_i$  tienen coordenadas afines  $(0, 0)$ ,  $(x_1, 0)$  y  $(x_2, y_2)$  respectivamente (vea la figura B.2 tomada de [Pal93], con  $x_1$  y  $x_2$  diferentes de 0). De acuerdo con (2.1) para hallar la curvatura de  $c_i$  en  $(x, y) = (0, 0)$  necesitamos calcular las derivadas parciales de primer y segundo orden de  $f_i(u, v)$  con respecto a  $x$  y a  $y$ , teniendo en cuenta la relación que existe entre las coordenadas afines y las coordenadas baricéntricas. En particular, de (2.5) sabemos que,

$$\begin{pmatrix} 0 & x_1 & x_2 \\ 0 & 0 & y_2 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (\text{B.1})$$

Por tanto,

$$u = -\frac{1}{x_1}x + \left(\frac{x_2 - x_1}{y_2 x_1}\right)y + 1 \quad (\text{B.2})$$

$$v = \frac{1}{x_1}x + \left(\frac{x_2}{y_2x_1}\right)y \quad (\text{B.3})$$

$$w = \frac{y}{y_2} \quad (\text{B.4})$$

Por otro lado,

$$\frac{\partial f_i}{\partial x} = \frac{\partial f_i}{\partial u} \frac{\partial u}{\partial x} + \frac{\partial f_i}{\partial v} \frac{\partial v}{\partial x} + \frac{\partial f_i}{\partial w} \frac{\partial w}{\partial x} \quad (\text{B.5})$$

y de (B.2), (B.3) y (B.4) tenemos que  $\frac{\partial u}{\partial x} = -\frac{1}{x_1}$ ,  $\frac{\partial v}{\partial x} = \frac{1}{x_1}$  y  $\frac{\partial w}{\partial x} = 0$ . Las derivadas parciales de  $f_i(u, v)$  con respecto a  $u$  y  $v$  se pueden calcular directamente de (4.2). Evaluado todas las derivadas en  $(0, 0)$  se obtiene que,

$$\frac{\partial f_i}{\partial x}(0, 0) = 0$$

Similarmente, es fácil ver que,

$$\frac{\partial f_i}{\partial y}(0, 0) = \frac{a_{20}^i}{y_2}$$

Por lo tanto, sustituyendo en (2.1) tenemos que,

$$k_0 = \frac{\left| \frac{\partial^2 f_i}{\partial x^2} \right|}{\left| \frac{\partial f_i}{\partial y} \right|} = \frac{|y_2|}{|a_{20}^i|} \left| \frac{\partial^2 f_i}{\partial x^2} \right| \quad (\text{B.6})$$

donde todas las derivadas parciales están evaluadas en  $(0, 0)$ . A partir de la expresión que se obtiene para  $\frac{\partial f_i}{\partial x}$  se puede calcular inmediatamente  $\frac{\partial^2 f_i}{\partial x^2}$  empleando la regla de la cadena. Después de varios pasos se llega a que,

$$\frac{\partial^2 f_i}{\partial x^2}(0, 0) = 2 \frac{a_{12}^i}{x_1^2}$$

Por tanto,

$$k_0 = 2 \frac{|a_{12}^i| y_2}{|a_{20}^i| x_1^2} = \frac{|a_{12}^i|}{|a_{20}^i|} \frac{\Delta_i}{(g_0^i)^3}$$



## Bibliografía

- [Baj92a)] Bajaj, C. L. y I. Ihm, Smoothing polyhedra using implicit algebraic splines, *Computer Graphics* **26** (1992), 79–88.
- [Baj92b)] Bajaj, C. L. y I. Ihm, Algebraic surface design with Hermite interpolation, *ACM Trans. on Graphics* **11** (1992), 61–91.
- [Baj92c)] Bajaj, C. L., The emergence of algebraic curves and surfaces in geometric design, Tech. Report Purdue University, CSD-TR-92-056 (1992).
- [Baj94] Bajaj C. L. y G. Xu, Data fitting with cubic A-splines. *Computer Graphics International, CGI94*, Melbourne, Australia, 1994.
- [Baj95] Bajaj, C. L., Chen J. y G. Xu, Modeling with cubic A-patches, *ACM Transactions on Graphics* **14** (1995), 103–133.
- [Baj97] Bajaj C. L., J. Blinn, *et al*, *Introduction to Implicit Surfaces*, Morgan Kaufmann Publishers, 1997.
- [Baj99a)] Bajaj, C. L. y G. Xu, A-Splines: Local interpolation and approximation using  $G^k$ -continuous piecewise real algebraic Curves, *Computer Aided Geometric Design* **16** (1999), 557–578.
- [Baj99b)] Bajaj, C. L y G. Xu, Error bounded regular algebraic spline curves, in *Proc. of the Fifteenth Annual Symposium on Computational Geometry*, Florida, 1999, 333–340.
- [Boehm84] Boehm, W., Farin G. y J. Kahmann, A survey of curve and surface methods in CAGD, *Computer Aided Geometric Design* **1** (1984), 1–60.
- [Book79] Bookstein, F. L., Fitting conic sections to scattered data, *Comput. Vision, Graphics and Image Processing* **9** (1979), 56–71.

- [Boor78] de Boor, C., *A Practical Guide to Splines*, Springer Verlag, New York, 1978.
- [Boor87] de Boor, C., Höllig K. y M. Sabin, High accuracy geometric Hermite interpolation, *Computer Aided Geometric Design* **4** (1987), 269–278.
- [Chan88] Chandler, R. E., A tracking algorithm for implicitly defined curves, *IEEE Computer Graphics and Applications* **8** (1988), 83–89.
- [Con80] Conte, S.D. y C. de Boor, *Elementary Numerical Analysis*, Mc Graw-Hill, 1980.
- [Dahm89] Dahmen, W., Smooth piecewise quadric surfaces in *Mathematical Methods in Computer Aided Geometric Design*, T. Lyche and L.L. Shumaker (eds.), Academic Press, New York, 1989, 181–193.
- [Dahm93] Dahmen, W. y T. M. Thamm-Schaar, Cubicoids: modeling and visualization, *Computer Aided Geometric Design* **10** (1993), 89–108.
- [Dou92] Dougherty, E., *An Introduction to Morphological Image Processing*, SPI-The International Society for Optical Engineering, Washington, 1992.
- [Far86] Farin, G., Triangular Bernstein-Bézier patches, *Computer Aided Geometric Design* **3** (1986), 83–127.
- [Far93] Farin, G., *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, 1993.
- [Ful84] Fulton, W., *Intersection Theory*, Springer Verlag, Berlin, 1984.
- [Gan94] Gander, W., Golub, G.H. y R. Strebler, Least squares fitting of circles and ellipses, *BIT* **34** (1994), 558–578.
- [Garr91] Garrity, T. y J. Warren, Geometric continuity, *Computer Aided Geometric Design* **8** (1991), 51–65.
- [Gonz92] González, R. C. y R. E. Woods, *Digital Image Processing*, Addison-Wesley, Massachusetts, 1992.
- [Gue99] Guerra-Ones, V., Hernández-Mederos, V., Numerical aspects in locating the corner of the L-curve, *Approximation, Optimization and Mathematical economics*, Physica-Verlag, Heidelberg, 2001, 121–131.

- [Guo91] Guo, B., Surface generation using implicit cubics, in *Scientific Visualization of Physical Phenomena*, N.M Patrikalakis (ed.), Springer-Verlag, Tokyo, 1991, 485–530.
- [Ham91] Hamann, B., G. Farin y M. Nielson, A parametric triangular patch based on generalized conics, in *NURBS for Curve and Surface Design*, G. Farin (ed.), SIAM, Philadelphia, 1991, 75–85.
- [Hans94] Hansford, D., R. E. Barnhill y G. Farin, Curves with quadric boundary precision, *Computer Aided Geometric Design* **11** (1994), 519–531.
- [Hart77] Hartshorne, R., *Algebraic Geometry*, Springer Verlag, Berlin, 1977.
- [Her73] Hernández, P., Unified Approach to the General Theory of Shells, Proc. IASS, Kielce Symposium, 1973.
- [Her97] Hernández-Mederos, Estrada-Sarlabous J. y P. Barrera-Sánchez, P., On the Euclidean distance from a point to a conic, *Revista Integración* **1** (1997), 45–61.
- [Her99] Hernández-Mederos, V., Behar Jequín, S. y J. Estrada-Sarlabous, Geometric design by means of a  $G^2$  continuous A-spline, *Approximation, Optimization and Mathematical economics*, Physica-Verlag, Heidelberg, 2001, 133–145.
- [Her00a)] Hernández-Mederos, Martínez-Morera D., Estrada-Sarlabous J., Fitting a conic A-spline to contour image data, *Revista Investigación de Operaciones* (para publicar en 2001).
- [Her00c)] Hernández-Mederos y J. Estrada-Sarlabous, Designing a  $G^1$  A-spline surface over a triangulation, *Reporte de Investigación del ICIMAF*.
- [Lozo83] Lozover, O., Preiss, K., Automatic construction of a cubic B-spline representation for a general curve, *Comput. and Graphics* **2** (1983), 149–153.
- [Moo91] Moore, D. y J. Warren, Approximation of dense scattered data using algebraic surfaces, Proc. of 24th Hawaii Int. Conference on System Sciences, Kauai, Hawaii, 1991, 681–690.

- [Oden93] Odesanya, O. S., Waggenspack, W. N. y D. E. Thompson, Construction of biological surface models from cross sections, *IEEE Transactions on Biomedical Engineering* **4** (1993), 329–334.
- [Pal92] Paluszny M. y R. Patterson, Curvas algebraicas de grado tres en modelación geométrica, Quinta Escuela Venezolana de Matemáticas, Univ. de los Andes, Mérida, Venezuela, 1992.
- [Pal93] Paluszny, M. y R. Patterson, A family of tangent continuous cubic algebraic splines, *ACM Transc. on Graphics* **12** (1993), 209–232.
- [Pal94] Paluszny, M. y R. Patterson,  $G^2$ -continuous cubic algebraic splines and their efficient display, *Proc. of Curves and Surfaces in Geometric Design*, P. J. Laurent, A. Le Méhauté, L. L. Schumacker (eds.), A. K. Peters, Wellesley, MA, 1994, 353–359.
- [Pal98] Paluszny, M. y R. Patterson, Geometric control of  $G^2$ -cubic A-splines, *Computer Aided Geometric Design* **15** (1998), 261–287.
- [Pal99] Paluszny, M., Tovar F. y R. Patterson,  $G^2$  composite cubic Bézier curves, *Journal of Computational and Applied Mathematics*, **102** (1999), 49–71.
- [Pat88] Patterson, R., Parametrizing and Graphing nonsingular cubic curves, *Computer Aided Design* **20** (1988), 615–623.
- [Pav83] Pavlidis, T., Curve fitting with conic splines, *ACM Trans. on Graphics* **1** (1983), 1–31.
- [Pon92] Ponce, A., Hoogs, J., Kriegman, D., On using CAD models to compute the pose of curved 3D objects, *Comput. Vision, Graphics and Image Processing* **2** (1992), 184–197.
- [Ray94] Ray, B. Kr. y K. S. Ray, A non-parametric sequential method for polygonal approximation of digital curves, *Pattern Recognition Letters* **15** (1994), 161–167.
- [Roy92] Royappa, A.V., *Symbolic Methods in Computer Graphics and Geometric Modeling*, Ph.D. Thesis, Purdue University, 1992.

- [Sam82] Sampson, P.D., Fitting conic sections to very scattered data: an iterative refinement of Bookstein algorithm, *Comp. Vision, Graphics, and Image Processing* **18** (1982), 97-108.
- [Sed84] Sederberg, T.W., *Implicit and Parametric Curves and Surfaces for Computer Aided Geometric Design*, Ph.D. Thesis, Purdue University, 1984.
- [Sed85] Sederberg, T.W., Piecewise algebraic surface patches, *Computer Aided Geometric Design* **2** (1985), 53-59.
- [Sed89] Sederberg, T.W., Scan line display of algebraic surfaces, *Computer Graphics* **3** (1989), 147-156.
- [Shum90] Shumaker, L. L., *Reconstructing 3D objects from cross sections, Computation of Curves and Surfaces*, Kluwer Academic Publishers, Dordrecht, 275-309, 1990.
- [Spa96] Spaeth, H., Orthogonal squared distance fitting with parabolas, *Numerical Methods and Error Bounds, Mathematical Research Vol. 89*, Akademie Verlag (1996), 261-269.
- [Tau91] Taubin, G., Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation, *IEEE Trans. on Pattern Anal. Machine Intell.* **11** (1991), 1115-1138.
- [Tau94a) Taubin, G., Distance approximations for rasterizing implicit curves, *ACM Trans. on Graphics* **13** (1994), 3-42.
- [Tau94b) Taubin, G., Rasterizing algebraic curves and surfaces, *IEEE Computer Graphics and Applications* **2** (1994), 14-23.
- [Walk78] Walker, R., *Algebraic Curves*, Springer Verlag, New York, 1978.
- [Warr86] Warren, J., *On algebraic surfaces meeting with geometric continuity*, Ph.D. Thesis, Department of Computer Science, Cornell University, 1986.
- [Warr90] Warren, J., *Several notions of geometric continuity for implicit plane curves*, *Monografías de la Academia de Ciencias de Zaragoza* **2**, 1990.

- [Xu01] Xu, G., Bajaj, C. y H. Huang,  $C^1$  Modeling with A-patches from Rational Trivariate Functions Computer Aided Geometric Design **18** (2001), 221-243.