

Jesús Miguel Chacón Muñoz

**PROCESADO DE GEOMETRÍA EN CAGD
MEDIANTE S-SERIES**

I.S.B.N. Ediciones de la UCLM
978-84-8427-637-1



Ediciones de la Universidad
de Castilla-La Mancha

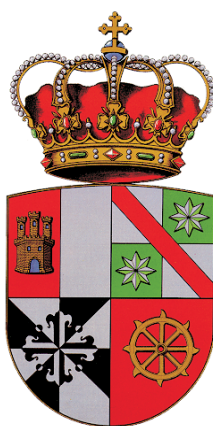
Cuenca, 2008

UNIVERSIDAD DE CASTILLA-LA MANCHA

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES

Departamento de Mecánica Aplicada e Ingeniería de Proyectos

Área de Expresión Gráfica en la Ingeniería



Procesado de Geometría en CAGD mediante s-series

Tesis Doctoral

DIRECTOR

Javier Sánchez-Reyes Fernández

AUTOR

Jesús Miguel Chacón Muñoz

Ciudad Real, 2006

*A mis Padres,
por todo su apoyo*

Agradecimientos

La realización de esta tesis doctoral no habría sido posible sin la inestimable colaboración de un gran número de personas. A todas ellas quiero dedicar estas páginas para expresarles mi agradecimiento.

Mi más sincero agradecimiento a mi director D. Javier Sánchez-Reyes Fernández, no sólo por la dedicación y apoyo prestado durante la realización de esta tesis, sino también por su excelente dirección. Javier siempre ha estado dispuesto a ayudarme para resolver cualquier problema que surgiera, aportado las correcciones y comentarios necesarios para la finalización de esta tesis doctoral.

Una mención especial merecen mis amigos del Departamento de Matemática Aplicada de la E.T.S. de Ingenieros Industriales de Ciudad Real, José Carlos Bellido, Faustino Maestre y Alberto Donoso, por su inestimable ayuda.

También quiero dar las gracias a todos mis compañeros y amigos de la Escuela Técnica Superior de Ingenieros Industriales de Ciudad Real, y en especial a Rubén Dorado, Carlos Sánchez, Pedro Roncero, Antonio González y Pedro Núñez, por la ayuda y los consejos que me han brindado a lo largo de estos años.

Por ultimo quiero agradecer a mis padres, a mis hermanos y a Candi todo el apoyo que me han dado y la confianza que han tenido en mí. Ellos han dado sentido a todo el trabajo realizado en estos años.

Ciudad Real, 2006

Índice General

Capítulo 1	Introducción	1
1.1	El CAGD (Diseño Geométrico Asistido por Ordenador)	1
1.2	El Procesado de Geometría en CAGD	2
1.3	Modelo Bernstein-Bézier de entidades resultantes del procesado de geometría	3
1.4	Deformaciones espaciales globales	4
1.5	Base polinómica alternativa a la base de Bernstein	6
1.6	Organización de la tesis: 7 capítulos y 2 anexos	9
Capítulo 2	s-series: el análogo en dos puntos de las series de Taylor	11
2.1	Definición de s-serie	12
2.2	Representación de los coeficientes de una s-serie	13
2.3	Expresión explícita de los coeficientes	14
2.4	Alternativa para el cálculo de los coeficientes	15
2.5	Interpretación de los coeficientes	16
2.6	Polinomios bivariados	20
Capítulo 3	Generación de aproximaciones spline polinómicas mediante s-series	23
3.1	Convergencia de las s-series	24
3.2	Proceso de subdivisión	26

3.3	Conversión entre las bases s-monomial y Bernstein	29
3.4	Eliminación de nodos internos	32
Capítulo 4	s-series frente a las series de Poisson para la representación de funciones analíticas	35
4.1	Series de Poisson	36
4.2	Convergencia de las series de Poisson	38
4.3	Series truncadas y polígonos de Poisson	39
4.4	Convergencia de las s-series frente a las series de Poisson	40
4.5	Ejemplos comparativos	41
Capítulo 5	Aproximación polinómica de espirales clotoides	47
5.1	Introducción	47
5.2	Cálculo de la s-serie para un segmento de clotoide	51
5.3	Ejemplos	52
Capítulo 6	Deformación de curvas y superficies	57
6.1	Introducción	58
6.2	Solución propuesta para las técnicas FFD y RFFD: Composición vía convoluciones discretas	65
6.3	Deformación de curvas y superficies	68
6.4	Ejemplos de deformación libre de objetos	73
6.5	Torsión y doblado de objetos	76
Capítulo 7	Conclusiones, aportaciones y futura línea de investigación	81
7.1	Conclusiones y aportaciones	81
7.2	Futura línea de investigación	84
Anexo A	Conversión al estándar de representación IGES	89
A.1	Archivo IGES	89

<i>Índice General</i>	III
A.2 Organización del archivo IGES	90
A.3 Ejemplos de implementación vía IGES	92
Anexo B Notación	95
Referencias	99
Índice de materias	107

Capítulo 1

Introducción

Este capítulo, un extracto del proyecto de investigación (Sánchez-Reyes, 2001a), contiene una breve introducción al CAGD (Diseño Geométrico Asistido por Ordenador), centrándose en un área muy concreta, el denominado *Procesado de Geometría*, área en la que ciertos problemas no han sido solucionados satisfactoriamente. En esta introducción se exponen cuáles de estos problemas sin solucionar se han abordado en esta tesis doctoral, sus antecedentes y estado del arte. Seguidamente se analizan las limitaciones de la representación de Bernstein-Bézier, y se introduce la solución propuesta. Finalmente se resumen los capítulos que forman esta tesis.

1.1 El CAGD (Diseño Geométrico Asistido por Ordenador)

El Diseño Geométrico Asistido por Ordenador (CAGD: *Computer Aided Geometric Design*) es la disciplina que trata de los modelos adecuados para la representación por ordenador de curvas, superficies y sólidos. El término CAGD fue acuñado por Barnhill y Riesenfeld durante la primera Conferencia celebrada en 1974 en Utah. Una historia de esta disciplina que describe las contribuciones de los pioneros se puede encontrar en Piegl (1993), o en Sabin (1990) para el tema concreto de las superficies esculpidas. Textos de referencia clásicos sobre CAGD son Hoschek y Lasser (1993), Rockwood y Chambers (1996), Farin y Hansford (2000) y Farin (2002).

El CAGD tiene gran diversidad de aplicaciones:

- Diseño de objetos diversos, como automóviles, embarcaciones y aeronaves (Fig. 1.1).
- Modelado de órganos humanos o de seres vivos en general.
- Ajuste de datos experimentales y soluciones discretizadas de ecuaciones diferenciales.



Figura 1.1: Ejemplos de aplicación del CAGD. (Fuente: Cad-Bikes)

La variedad de aplicaciones resulta tan amplia que no existe un único modelo matemático de curvas o superficies que solucione todos los problemas. Por ejemplo, la superficie utilizada para modelar la carrocería de un automóvil difícilmente podrá emplearse para visualizar datos complejos.

1.2 El Procesado de Geometría en CAGD

Un apartado especialmente interesante del CAGD lo constituye el Procesado de Geometría, cuya finalidad es el cálculo de propiedades de objetos geométricos ya construidos. En su acepción más amplia, este término incluye los algoritmos que se aplican a entidades geométricas ya existentes. Así, este tópico abarca los siguientes temas:

- Obtención de un modelo de Bernstein-Bezier de entidades resultantes del procesado.
- Aplicación de deformaciones a sólidos o superficies.
- Conversiones entre formatos de objetos geométricos.
- Obtención de curvas y superficies offset (desplazadas).
- Cálculo de intersecciones entre curvas o superficies.
- Obtención de parametrizaciones arco de curvas.
- Cálculo de longitudes de curvas y áreas de superficies.

A primera vista, muchas de estas operaciones parecen sencillas, ya que admiten expresiones algebraicas explícitas que requieren tan sólo el cómputo de integrales, derivadas y las operaciones aritméticas habituales, tales como suma, multiplicación, división, o extracción de raíces cuadradas. No obstante, el problema radica en que en la

mayoría de ocasiones no se requiere un resultado numérico. Por el contrario, cuando el resultado consiste en una nueva entidad geométrica se precisa obtener una representación compatible con los estándares utilizados. En concreto, para las curvas y superficies una representación polinómica (o racional) de Bernstein-Bézier, modelo utilizado por la mayoría de los programas de CAD/CAM.

A continuación se exponen los dos problemas abordados en esta tesis doctoral, correspondientes a los dos primeros puntos expuestos en este apartado, sus antecedentes y estado del arte.

1.3 Modelo Bernstein-Bézier de entidades resultantes del procesamiento de geometría

Los programas de CAD/CAM se basan en la representación de curvas y superficies en el estándar NURBS (*Non-uniform Rational B-splines*), compuesta por la unión de tramos de Bézier racionales. Esto permite diseñar gran variedad de curvas y superficies utilizando una única representación interna (Rogers, 2001). Por tanto se debe generar un modelo Bernstein-Bézier de aquellas entidades resultantes del procesamiento de geometría, necesario para su incorporación en los programas de CAD/CAM. La obtención de tal modelo presenta problemas en los siguientes casos (Sánchez-Reyes, 2001a):

- Entidades geométricas que inicialmente no admiten representación polinómica o racional, las denominadas entidades trascendentes (Fig. 1.2).
- Aún cuando las entidades iniciales admitan dicha representación, si la operación de procesamiento implica operaciones tales como la extracción de raíces cuadradas, que generan expresiones no polinómicas o racionales.

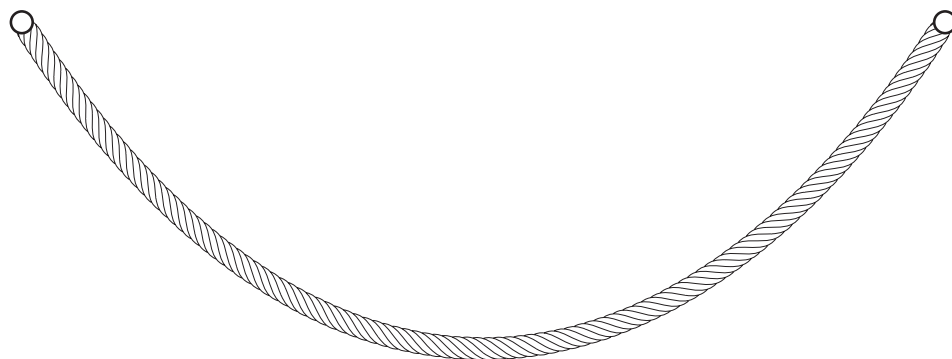


Figura 1.2: Catenaria como ejemplo de curva trascendente. (Fuente: Sánchez-Reyes, 2001a)

- Cuando, debido a que las operaciones conllevan multiplicación o composición de funciones polinómicas, el resultado, si bien polinómico o racional, presenta un grado demasiado elevado para poder ser incorporado en los programas de CAD, que suelen tener limitado el grado de las entidades que pueden manejar, por razones de eficiencia.

En estos casos la única solución viable consiste en obtener algún tipo de aproximación del resultado.

1.4 Deformaciones espaciales globales

Las técnicas de modelado 3D de superficies por ordenador se basan principalmente en la manipulación de puntos de control. No obstante, para diseñar objetos complejos es necesario combinar diversos elementos (curvas, superficies, ...) y por tanto utilizar un gran número de puntos de control. Por consiguiente, la creación y modificación de estos objetos resulta costosa para el diseñador, dado que para alterar alguna característica geométrica global de un objeto puede ser necesario modificar un número elevado de puntos de control.

Para resolver este problema se han desarrollado métodos que permiten aplicar modificaciones a todo el objeto (escalados, torsiones, *bendings* y deformaciones en general), lo que facilita notablemente el diseño de estos objetos complejos. Entre los métodos más potentes de modelado se encuentran las deformaciones espaciales (Bechmann 1994). Estos métodos consisten en tratar los puntos de control del objeto que se desea deformar de manera agrupada en lugar de individualmente, mediante herramientas capaces de transmitir una modificación a la zona del objeto que se desea modificar. Estas herramientas pueden ser estructuras espaciales que contienen al objeto (Fig. 1.3) o una curva asociada al objeto cuya forma se modifica en consonancia a la deformación deseada (Lazarus et al., 1994).

Estos problemas de diseño contrastan con la conclusión a la que puede llegar una persona no experta en el área al examinar los paquetes de software actuales, que permiten un modelado 3D sencillo con mínima interacción del usuario. Sin embargo (Vogtner 1996), la veracidad de esta conclusión depende de lo que se entienda por “Modelado 3D”. Si el objetivo consiste simplemente en la obtención de un modelo poligonal para generar imágenes la afirmación es correcta. Por el contrario, en aplicaciones de ingeniería un modelo poligonal no es suficiente, ya que la meta es la fabricación del objeto, por lo que es necesaria una descripción geométrica precisa. Por tanto el modelo del objeto y sus componentes deben mantener asociatividad, han de manipularse posteriormente, poder suministrar la geometría necesaria para el análisis por elementos finitos y probablemente ser compartido por otros modelos distintos de CAD/CAM. Además el modelo ha de

mantenerse válido durante el ciclo de vida del producto, de manera que ha de ajustarse a ciertos estándares de representación para que sea transportable a otra plataforma. Si han de verificarse todas las condiciones descritas, entonces el “modelado 3D” no resulta tan trivial.

En consecuencia, el resultado deseable sería la representación explícita del objeto deformado. Si bien esto resulta más complejo que la simple evaluación numérica necesaria para la visualización. Para obtener la representación exacta explícita es necesario calcular la composición del objeto con la estructura utilizada como herramienta de deformación. Como la base que se emplea habitualmente es la de Bernstein, necesitaremos componer funciones polinómicas en dicha base. Si bien el resultado explícito de la composición proporciona la solución polinómica exacta (el espacio de polinomios es cerrado respecto a la composición), tiene como inconveniente el coste del proceso de cálculo y el aumento de la complejidad (grado) del objeto deformado respecto al original, ya que el grado resultante es igual al producto de grados de los polinomios que intervienen en la composición. Al aplicar deformaciones sucesivas sobre un objeto, su grado aumentará geoméricamente, incrementando la complejidad del objeto y el coste de cada nueva etapa.

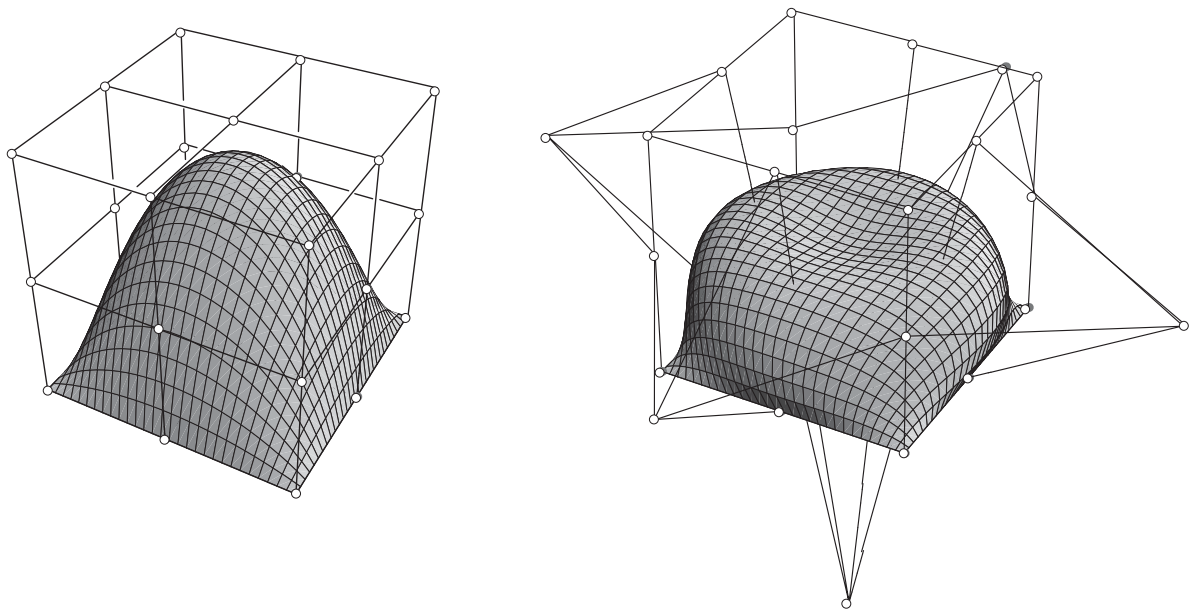


Figura 1.3: Deformación espacial de un objeto contenido en un volumen de control.

(Fuente: Sánchez-Reyes y Arranz, 1998)

1.5 Base polinómica alternativa a la base de Bernstein

En este apartado se analizan las ventajas, así como las limitaciones de la representación de Bernstein-Bézier y se describen las herramientas alternativas que proporcionan un marco unificado para la resolución de problemas en el Procesado de Geometría.

La representación Bernstein-Bézier de curvas o superficies polinómicas se ha convertido en un estándar para aplicaciones de CAGD, puesto que disfruta de numerosas ventajas (Farin, 2002; Bowyer y Woodwark, 1993):

- Elegantes propiedades geométricas, incluyendo una interpretación geométrica e intuitiva de los coeficientes como puntos de control.
- Existencia de un algoritmo tipo de Casteljau: es la única representación polinómica que admite un procedimiento *corner-cutting* (en cada paso se cortan las esquinas del polígono anterior) que simultáneamente proporciona una subdivisión, como demostraron Barry y Goldman (1988).
- Óptima estabilidad: Farouki y Goodman (1996) demostraron que no podemos encontrar otra base polinómica no negativa definida sobre el intervalo unitario con menor número de condición para evaluación.

Formalmente hablando, la base de Bernstein se puede identificar como la B-base normalizada (Carnicer y Peña, 1993) del espacio polinómico algebraico. Dado un espacio vectorial de funciones con bases normalizadas totalmente positivas, entre todas ellas existe una única base, llamada B-base, con propiedades óptimas de conservación de forma y óptima estabilidad (Peña, 1999). Además, la B-base es la única base del espacio que admite un algoritmo de tipo de Casteljau (Mainar y Peña, 1999).

Pese a estas propiedades positivas, la base de Bernstein tiene también sus puntos débiles:

- No se puede determinar el grado real (mínimo) de un polinomio en esta base de manera inmediata por inspección de coeficientes, sino que se precisa de un algoritmo no trivial.
- Los algoritmos de reducción de grado (Forrest, 1972; Lodha y Warren, 1994; Piegl y Tiller, 1995; Sunwoo y Lee, 2004; Sunwoo, 2005) tienden a ser numéricamente inestables, pues el cómputo de los nuevos puntos de control involucra combinaciones no convexas de los originales.
- Es innecesariamente complicada para representar varios polinomios con diversos grados, porque debe elevarse el grado hasta el valor máximo. Además, aunque el proceso de elevación de grado sea estable, conlleva errores numéricos.

Como consecuencia de estas desventajas, no existe una metodología unificada para obtener aproximaciones polinómicas en esta base de entidades no polinómicas arbitrarias. Podría pensarse en emplear la base de monomios, pues permite trabajar con polinomios de diversos grados y operar con ellos algebraicamente. De hecho, es sencillo obtener una aproximación polinómica (una serie de Taylor) para las funciones trascendentes más comunes. Por desgracia, si bien una expansión de Taylor resulta adecuada para aproximar una función en un intervalo simétrico alrededor de un punto $u = 0$, conduce a huecos cuando tratamos de conectar varias de estas aproximaciones (Fig. 1.4a). Esto es inaceptable en CAGD, pues las curvas y superficies se suelen definir por trozos. Además, el posterior análisis de la geometría mediante un programa de CAE (elementos finitos, dinámica de fluidos, etc) proporcionaría resultados incorrectos (Kasik et al., 2005). También la base monomial presenta inestabilidad numérica que no la hace adecuada para la representación de curvas y superficies (Farín, 2002). Asimismo, Piegl (2005) afirma que el desarrollo de estrategias para eliminar inestabilidades numéricas, entre las que se encuentra la elección de la formulación matemática correcta, constituye uno de los principales desafíos en CAD.

La solución propuesta se deriva de la utilización de la base s-monomial presentada por Sánchez-Reyes (1997). Esta base, correspondiente a una forma de Newton modificada, constituye el equivalente simétrico de la base de monomios, y de manera natural conduce a expansiones polinómicas de tipo Hermite (Fig. 1.4b) en dos puntos (Sánchez-Reyes, 2000), que además solventan los problemas de la base monomial. Estas expansiones son justo las requeridas para la representación a trozos de entidades geométricas, en las que generarían un spline con grado de continuidad máxima.

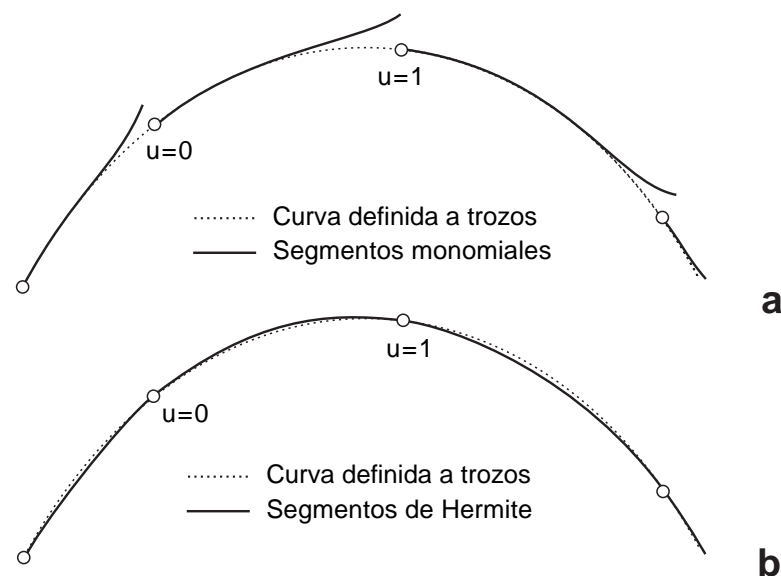


Figura 1.4: Diferentes aproximaciones polinómicas de una curva. (Fuente: Sánchez-Reyes, 2000)

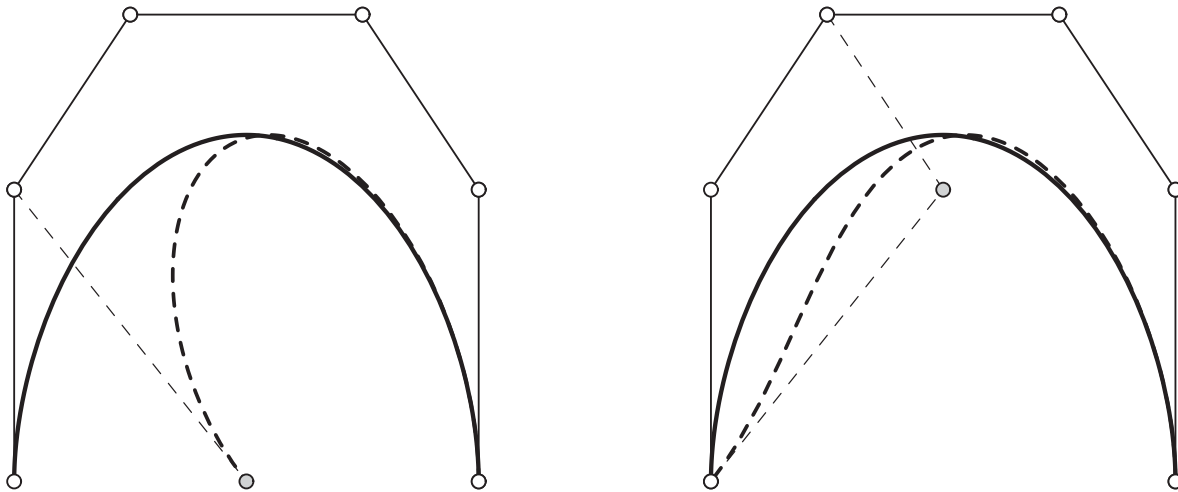


Figura 1.5: Polígono de control asociado a la base s-monomial y efecto de los puntos de control.

(Fuente: Sánchez-Reyes, 1997)

El truncamiento de estas expansiones conserva tantas derivadas en los puntos extremos como es posible, y corresponde al algoritmo de reducción de grado en la base s-monomial. Este proceso equivale a la aplicación sucesiva del algoritmo de reducción de grado de Forrest (1972). De esta forma se genera la aproximación de Hermite en los puntos extremos. Esta nueva base resulta idónea para obtener aproximaciones polinómicas de las entidades complejas que resultan del procesado de geometría. Además las matrices de cambio de base entre esta representación y la de Bernstein no presentan problemas numéricos. Por tanto, se pueden transformar con precisión los datos de entrada a la nueva base, efectuar las operaciones de procesado de geometría, reducir el grado y finalmente volver a la representación de Bézier.

Esta nueva base, como la de Bernstein, admite una representación de los coeficientes como puntos de control, definiendo un polígono de control con significado geométrico. Dicho polígono disfruta de las mismas propiedades que el polígono de Bézier (Fig. 1.5):

- Los puntos “tiran” de la curva, permitiendo un control interactivo de la misma.
- La curva es tangente al polígono en sus extremos.
- La curva se encuentra contenida en el dominio convexo del polígono.
- La representación es invariante frente a transformaciones afines.

1.6 Organización de la tesis: 7 capítulos y 2 anexos.

El Capítulo 2 introduce la herramienta matemática básica empleada en este trabajo, las *s-series*. Estas series resultan de la base *s*-monomial, basada en expansiones de Hermite en un intervalo unitario de la variable, descritas en el apartado 1.5. Este capítulo se centra en la definición de una *s-serie*, así como en la representación y cálculo de sus coeficientes.

En el Capítulo 3 se describen las estrategias para calcular de manera eficiente la aproximación de una entidad mediante *s-series*. En primer lugar se repasa la convergencia de las *s-series*. En segundo lugar se estudia el proceso de subdivisión, que permite garantizar la convergencia de la *s-serie*. A continuación se presentan los procesos de conversión entre la base *s*-monomial y la base de Bernstein. Finalmente se concatenan las curvas de Bézier resultantes de la aproximación de cada uno de los trozos en que se ha subdividido la entidad original, consiguiendo una representación B-spline más adecuada para aplicaciones CAD/CAM que la unión de varios tramos de Bézier.

En el Capítulo 4 se comparan las aproximaciones mediante *s-series* con las basadas en series de Poisson. Primero se hace un breve repaso de las series de Poisson. A continuación se compara la convergencia de las *s-series* con las de Poisson. Finalmente mediante una serie de ejemplos se comparan ambos métodos de aproximación, y se ilustra la superioridad de las *s-series*. Este capítulo corresponde a la traducción del artículo (Sánchez-Reyes y Chacón, 2005a) en revista internacional.

En el Capítulo 5 se aproxima la clotoide como ejemplo de aplicación de las estrategias de aproximación mediante *s-series* expuestas en el Capítulo 3. La clotoide, definida paramétricamente en términos de las integrales de Fresnel, es una curva trascendente. Por esto no admite una parametrización polinómica racional y no puede expresarse en el estándar NURBS. Se presenta un método para aproximar cualquier tramo de clotoide y su offset vía *s-series*, obteniendo una aproximación polinómica que puede ser incorporada en los programas de CAD/CAM. Se compara este método con el propuesto por Want et al., (2001), y se muestran una vez más las ventajas de las *s-series* frente a otros métodos de aproximación. Este capítulo contiene el material de los artículos (Sánchez-Reyes y Chacón, 2003a; 2003b), en revista internacional y nacional, respectivamente.

En el Capítulo 6 se aplican las *s-series* a las técnicas de deformación. El objetivo de este capítulo consiste en conseguir una aproximación polinómica Bernstein-Bézier de los objetos deformados. En primer lugar se presentan las técnicas de deformación libre basadas en la deformación de una estructura espacial que contiene a dicho objeto, y se describen los procesos de doblado y torsión de objetos.

En el Capítulo 7 se exponen las conclusiones y aportaciones originales de esta tesis. También se propone como línea de investigación para su desarrollo en el futuro la aproximación del offset de curvas y superficies B-spline arbitrarias.

Finalmente en el Anexo A se estudia un procedimiento para incorporar en los programas de CAD/CAM las aproximaciones generadas mediante s-series. El método elegido consiste en generar el archivo IGES de estas aproximaciones, formato de archivo neutro importable desde cualquier programa de CAD/CAM comercial. El anexo B recoge la lista de símbolos utilizados.

Capítulo 2

s-series: el análogo en dos puntos de las series de Taylor

El Diseño Geométrico Asistido por Ordenador (CAGD) se basa en la representación de entidades geométricas en el estándar NURBS (Capítulo 1), por lo que se debe obtener una aproximación polinómica o racional de aquellas funciones trascendentes que no pueden ser expresadas en la base de Bernstein. En principio se podría pensar en una aproximación mediante series de Taylor truncadas. De esta forma se obtendría una buena aproximación alrededor de un punto, pero se precisarían grados muy elevados para errores pequeños y los programas de CAD tienen limitado el grado máximo admisible. Una forma de evitar estos grados elevados sería conectar varios desarrollos de Taylor, pero en este caso aparecerían huecos en la unión de dos expansiones, como ilustra la Fig. 1.4a, algo inaceptable en una representación para CAD.

En lugar de desarrollos de Taylor en un punto, la aproximación para una representación a trozos debe basarse en el empleo de expansiones de Hermite en dos puntos, técnica sobradamente conocida en análisis numérico (Stoer y Burlirsch, 1993). Dada una función $f(t)$ definida sobre el intervalo $t \in [t_0, t_1]$, la aproximación de Hermite $H_k(f; t)$ de orden k es el único polinomio de grado $n = 2k + 1$ que tiene contacto de orden k en los extremos $\{t_0, t_1\}$ con $f(t)$, en otras palabras, el polinomio que interpola todas las derivadas $\left\{ f^{(r)}(t_0), f^{(r)}(t_1) \right\}_{r=0}^k$.

Una vez que se ha identificado la aproximación de Hermite como el método deseable, el problema es calcular de modo eficiente el desarrollo de Hermite de funciones arbitrarias. En los libros clásicos de teoría de la aproximación (Berezin y Zhidkov, 1965; Davis, 1975) se puede encontrar una expresión explícita que calcula la aproximación de Hermite de

orden k en función de las derivadas en los puntos extremos, pero los polinomios de Hermite de orden k no forman parte de los polinomios de orden mayor, es decir, no se puede hablar de una serie de Hermite. Para solucionar este problema, Sánchez-Reyes (1997) introduce la base *s*-monomial, base polinómica alternativa a la monomial basada en una expansión de Hermite en el intervalo $u \in [0,1]$. Las series resultantes se denominan *s-series*, ya que son series de potencias de un parámetro simétrico s .

2.1 Definición de *s-series*

La *s-series* (Sánchez-Reyes, 2000) de una función $a(u)$ definida sobre un dominio unitario $u \in [0,1]$ de la variable es simplemente una serie de potencias del parámetro simétrico $s = (1-u)u$, cuyos coeficientes a_k son funciones lineales de u :

$$a(u) = \sum_{k=0}^{\infty} a_k(u) s^k, \quad \begin{aligned} s &= (1-u)u \\ a_k(u) &= (1-u)a_k^0 + ua_k^1 \end{aligned} \quad (2.1)$$

Para calcular la *s-series* de una función $f(t)$ sobre un dominio general $t \in [t_0, t_1]$, basta con describirla como $a(u) = f(t(u))$ en función de la variable unitaria u mediante el siguiente cambio de variable:

$$t(u) = t_0(1-u) + t_1(u), \quad \begin{aligned} u &\in [0,1] \\ t &\in [t_0, t_1] \end{aligned} \quad (2.2)$$

Las funciones lineales $a_k(u)$ están expresadas en la base de Bernstein, es decir, los coeficientes a_k^0, a_k^1 son las ordenadas de Bézier (Fig. 2.1a). La función $s^k(u)$ es el polinomio central (k -ésimo) de Bernstein escalado $\tilde{B}_k^{2k}(u)$ de grado $n = 2k$:

$$s^k = \tilde{B}_k^{2k}(u), \quad \tilde{B}_k^{2k}(u) = (1-u)^k u^k.$$

Como muestra la Fig. 2.1b, $s^k(u)$ es una función con forma de campana, que presenta las siguientes propiedades:

- Degeneración a un impulso escalado cuando $k \rightarrow \infty$.
- Simetría respecto al punto medio $u = 1/2$, donde obtiene su máximo 2^{-2k} .
- Contacto de orden $k - 1$ con el eje horizontal en los puntos extremos $u = 0$ y $u = 1$.

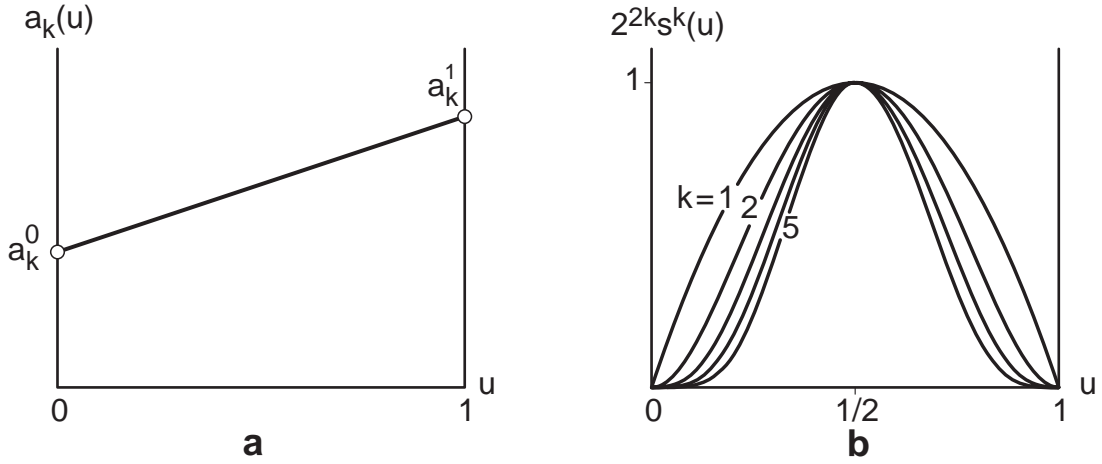


Figura 2.1: (a) Función lineal $a_k(u)$. (b) Función simétrica $s^k(u)$ del término k -ésimo de una s-serie.
(Fuente: Sánchez-Reyes, 2001b)

La principal propiedad de las s-series es que, dada una función $a(u)$ sin singularidades en una cierta región del plano complejo, puede ser expresada unívocamente como una s-serie (2.1) convergente. Además la aproximación de Hermite $H_k(a; u)$ de orden k (grado $n = 2k + 1$) resulta de truncar en el término k -ésimo la s-serie (2.1):

$$H_k(a; u) = \sum_{i=0}^k a_i(u) s^i, \quad (2.3)$$

proceso equivalente a la aplicación sucesiva del algoritmo de reducción de grado de Forrest (1972). Este es el único polinomio que reproduce las derivadas $\{a^{(r)}(u)\}_{r=0}^k$ en los puntos extremos $u = 0$ y $u = 1$.

2.2 Representación de los coeficientes de una s-serie

Antes de abordar el problema de obtener los coeficientes a_k de una función dada, se tratará cómo representar una s-serie. Cada "coeficiente" $a_k(u)$ de una s-serie $a(u)$ (2.1) es una función lineal expresada en la base de Bernstein:

$$a(u) = \sum_k a_k(u) s^k, \quad a_k(u) = (1-u)a_k^0 + ua_k^1 = \{(1-u), u\} \begin{Bmatrix} a_k^0 \\ a_k^1 \end{Bmatrix}. \quad (2.4)$$

Por tanto, una representación natural de $a_k(u)$ (Sánchez-Reyes, 2000) es la pareja de ordenadas de Bézier:

$$a_k(u) \rightarrow a_k = \begin{Bmatrix} a_k^0 \\ a_k^1 \end{Bmatrix}. \quad (2.5)$$

A partir de ahora se adoptará la notación simplificada (2.5) para representar funciones lineales. También se usará para $a_k(u)$ el término "coeficiente", teniendo en cuenta que se está tratando con funciones lineales y no con constantes. De esta forma, la representación de una *s-serie* se reduce a la representación de dos series de Taylor.

En algunos casos, como representación alternativa, también se puede descomponer $a_k(u)$ (2.4) en dos términos, uno constante y otro lineal:

$$a_k(u) = \bar{a}_k + \left(u - \frac{1}{2}\right) \Delta a_k, \quad \begin{cases} \bar{a}_k = \frac{1}{2}(a_k^0 + a_k^1) \\ \Delta a_k = a_k^1 - a_k^0 \end{cases}.$$

De este modo se puede descomponer una *s-serie* en sus componentes simétrica $\bar{a}(s)$ y antisimétrica $\Delta a(s)$:

$$a(u) = \bar{a}(s) + \left(u - \frac{1}{2}\right) \Delta a(s), \quad \begin{cases} \bar{a}(s) = \sum_k \bar{a}_k s^k \\ \Delta a(s) = \sum_k \Delta a_k s^k \end{cases}.$$

Cada coeficiente $a_k(u)$ en una *s-serie* (2.1) es una función lineal expresada en la base de Bernstein, por lo que una *s-serie* se puede interpretar como una serie de Taylor donde cada coeficiente es el par (2.5), o como la representación de dos series de Taylor.

2.3 Expresión explícita de los coeficientes

Una *s-serie* es una serie de Newton modificada (Sánchez-Reyes, 2003b) de centros $\{0,1, 0,1, 0,1, \dots\}$. Por consiguiente, los coeficientes a_k^0, a_k^1 (2.5) pueden calcularse fácilmente en términos de las derivadas de los puntos extremos, utilizando el algoritmo triangular de diferencias divididas empleado para calcular los coeficientes de una serie de Newton (Goldman, 2003). Dadas las series de Taylor de las funciones $a(u)$ y $a(1-u)$:

$$\begin{aligned} a(u) &= \sum_{i=0}^{\infty} c_i^0 u^i, & c_i^0 &= \frac{1}{i!} \left. \frac{d^i a(u)}{du} \right|_{u=0}, \\ a(1-u) &= \sum_{i=0}^{\infty} c_i^1 (1-u)^i, & c_i^1 &= \frac{1}{i!} \left. \frac{d^i a(1-u)}{du} \right|_{u=0}, \end{aligned} \quad (2.6)$$

los coeficientes a_k (2.5) quedan expresados como la combinación lineal de los coeficientes c_i^0, c_i^1 (Sánchez-Reyes, 2003b):

$$\begin{Bmatrix} a^1 \\ a^0 \end{Bmatrix}_k = \sum_{i=0}^k \binom{2k-i-1}{k-i} \begin{Bmatrix} c^0 \\ c^1 \end{Bmatrix}_k - \binom{2k-i-1}{k-i-1} \begin{Bmatrix} c^1 \\ c^0 \end{Bmatrix}_k. \quad (2.7)$$

Esta formula es una versión simplificada de la obtenida por López y Temme (2002) o Fine (1961).

2.4 Alternativa para el cálculo de los coeficientes

Un método alternativo para el cálculo del par de coeficientes a_k^0, a_k^1 (2.5) consiste en desarrollar algoritmos específicos para realizar las operaciones básicas con s-series (suma, multiplicación, división, raíz cuadrada, composición, integración, etc.) así como para calcular la s-serie de las funciones trascendentes más importantes. Este método permite obtener los coeficientes de una s-serie sin la necesidad de calcular las derivadas, mediante la combinación de s-series. Los algoritmos resultantes son parecidos a los de las series de Taylor (Knuth, 1998), por lo que las s-series pueden manipularse de igual manera. A continuación se expone un resumen de las herramientas desarrolladas por Sánchez-Reyes (2000), donde únicamente se indican las operaciones que se han utilizado en este trabajo.

Operaciones básicas con s-series

Suma. Dadas dos s-series $a(u)$ y $b(u)$, los coeficientes de la suma $c(u) = a(u) + b(u)$ se obtienen simplemente sumando los coeficientes a_k y b_k del mismo orden.

Producto. Dadas dos funciones lineales $a(u)$ y $b(u)$ expresadas en la base de Bernstein:

$$a(u) \rightarrow a = \begin{Bmatrix} a^0 \\ a^1 \end{Bmatrix}, \quad b(u) \rightarrow b = \begin{Bmatrix} b^0 \\ b^1 \end{Bmatrix},$$

el producto de ambas $c(u) = a(u) \cdot b(u)$ se calcula como:

$$c(u) = a(u) \cdot b(u) \rightarrow c = \begin{Bmatrix} a^0 b^0 \\ a^1 b^1 \end{Bmatrix} - \Delta a \Delta b \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} s, \quad \begin{aligned} \Delta a &= a^1 - a^0 \\ \Delta b &= b^1 - b^0 \end{aligned} \quad (2.8)$$

Por tanto, para multiplicar dos s-series basta con agrupar productos de pares de coeficientes. Esta operación se puede expresar de manera eficiente mediante convoluciones discretas (Sánchez-Reyes, 2000).

Composición: Dadas dos s-series $a(v)$ y $v(u)$, con $u \in [0,1]$ y $v \in [0,1]$, la composición $a(v(u))$ se calcula simplemente reemplazando la variable v de la serie $a(v)$ por la serie $v(u)$. Dado que una s-serie truncada admite el esquema de evaluación de Horner, esta composición implica sólo sumas y multiplicaciones, descritas anteriormente.

Integración: Dada la *s*-serie de la derivada $d(u)$ de una función $a(u)$, esta última tiene una serie cuyos coeficientes vienen dados por la fórmula recurrente:

$$\begin{Bmatrix} a^0 \\ a^1 \end{Bmatrix}_{k+1} = \frac{1}{k+1} \begin{Bmatrix} d_k^0 - (2k+1)\Delta a_k \\ -d_k^1 + (2k+1)\Delta a_k \end{Bmatrix}, \quad k = 0, 1, 2, 3, \dots \quad (2.9)$$

Obsérvese que esta fórmula necesita los coeficientes iniciales a_0 :

$$H_0(d; u) \rightarrow a_0 = \begin{Bmatrix} a_0^0 \\ a_0^1 \end{Bmatrix} = \begin{Bmatrix} a(0) \\ a(1) \end{Bmatrix}, \quad (2.10)$$

generados por la expresión (2.7) para $k = 0$.

s-serie de función sinusoidal: Para calcular los coeficientes a_k de la *s*-serie de una función sinusoidal $\sin(t)$, de argumento $t \in [t_0, t_1]$, basta con describirla en la forma (2.2) en función de la variable unitaria v :

$$a(u) = \sin(t(u)), \quad \begin{aligned} u &\in [0, 1] \\ t(u) &= t_0(1-u) + t_1 u \end{aligned} \quad (2.11)$$

Los coeficientes a_k vienen dados por (Sánchez-Reyes, 2001b):

$$a_{k+2} = \frac{1}{k+2} \left(\begin{bmatrix} 4(k+1) & -2 \\ -2 & 4(k+1) \end{bmatrix} \begin{Bmatrix} a_{k+1}^0 \\ a_{k+1}^1 \end{Bmatrix} - \frac{\Delta^2}{k+1} a_k \right), \quad k = 0, 1, 2, 3, \dots \quad (2.12)$$

donde $\Delta = t_1 - t_0$. Esta fórmula recursiva comienza para $k = 0$ con los coeficientes iniciales a_0 (2.10) y a_1 dados por la expresión (2.7) para $k = 1$:

$$\begin{Bmatrix} a_1^0 \\ a_1^1 \end{Bmatrix} = \begin{Bmatrix} a'(0) \\ a'(1) \end{Bmatrix} + \Delta a_0 \begin{Bmatrix} -1 \\ 1 \end{Bmatrix},$$

fácilmente calculables dado que la solución $a(u)$ es conocida.

2.5 Interpretación de los coeficientes

A continuación se expone un resumen de la interpretación geométrica de los coeficientes de una *s*-serie presentada por Sánchez-Reyes (1997). Asimismo, se presenta el polígono de control asociado a una *s*-serie resultante de la representación como puntos de control, introducida en el Capítulo 1, de los coeficientes de la base *s*-monomial.

Una s-serie (2.1) se puede describir en términos de sus funciones base como:

$$a(u) = \sum_{k=0}^{\infty} (a_k^0 \cdot P_k^0(u) + a_k^1 \cdot P_k^1(u)), \quad \begin{aligned} P_k^0 &= u^k (1-u)^{k+1} \\ P_k^1 &= u^{k+1} (1-u)^k \end{aligned} \quad (2.13)$$

Por consiguiente, se puede expresar en términos de los coeficientes normalizados \hat{a}_k^0, \hat{a}_k^1 :

$$\hat{a}_k^0 = \frac{a_k^0}{\binom{2k+1}{k}},$$

y de los polinomios de Bernstein como:

$$a(u) = \sum_{k=0}^{\infty} (\hat{a}_k^0 \cdot B_k^{2k+1}(u) + \hat{a}_k^1 \cdot B_{k+1}^{2k+1}(u)). \quad (2.14)$$

El interpolador de Hermite $H_k(a; u)$ de orden k de $a(u)$ (2.13):

$$H_k(a; u) = \sum_{i=0}^k (\hat{a}_i^0 \cdot B_i^{2i+1}(u) + \hat{a}_i^1 \cdot B_{i+1}^{2i+1}(u))$$

resulta de la aplicación sucesiva del algoritmo de reducción de grado de Forrest (1972) a un polinomio en la base de Bernstein. Los términos de la expresión (2.14) serán eliminados en la sucesivas reducciones de grado $(2k+1) \rightarrow (2k-1)$, de la interpolación de Hermite de orden k a la de orden $k-1$:

$$H_k(a; u) - H_{k-1}(a; u) = \hat{a}_k^0 \cdot B_k^{2k+1}(u) + \hat{a}_k^1 \cdot B_{k+1}^{2k+1}(u). \quad (2.15)$$

Si $\{b_i^k\}_{i=0}^{2k+1}$ son los coeficientes de Bézier de $H_k(a; u)$, y $\{\hat{h}_i^{k-1}\}_{i=0}^{2k+1}$ los coeficientes de $H_{k-1}(a; u)$ después de la elevación de grado $(2k-1) \rightarrow (2k+1)$, de (2.15) se puede deducir:

$$b_i^k = \hat{h}_i^{k-1} \begin{cases} i=0, \dots, k-1 \\ i=k+2, \dots, 2k+1 \end{cases}, \quad \begin{aligned} \hat{a}_k^0 &= b_k^k - \hat{h}_k^{k-1} \\ \hat{a}_k^1 &= b_{k+1}^k - \hat{h}_{k+1}^{k-1} \end{aligned}$$

Por consiguiente, los coeficientes \hat{a}_k^0, \hat{a}_k^1 pueden interpretarse como la diferencia entre los coeficientes centrales de Bézier del interpolador de Hermite de orden k y los coeficientes centrales de Bézier del interpolador de Hermite de orden $k-1$ expresados en la base de Bernstein de grado $n = 2k+1$ (Fig. 2.2). Los valores de $a(u)$ en los extremos vienen dados:

$$\begin{Bmatrix} \hat{a}_0^0 \\ \hat{a}_0^1 \end{Bmatrix} = \begin{Bmatrix} a(0) \\ a(1) \end{Bmatrix}.$$

Partiendo de los coeficientes $\hat{\mathbf{a}}^0 = \{5/2, 5/3, 1\}$ y $\hat{\mathbf{a}}^1 = \{3/2, 5/3, 1\}$ de una función $a(u)$ de grado $n = 5$, se han generado los coeficientes centrales de Bézier de las aproximaciones de Hermite de $a(u)$ mediante la aplicación sucesiva de la interpretación geométrica descrita (Fig. 2.3).

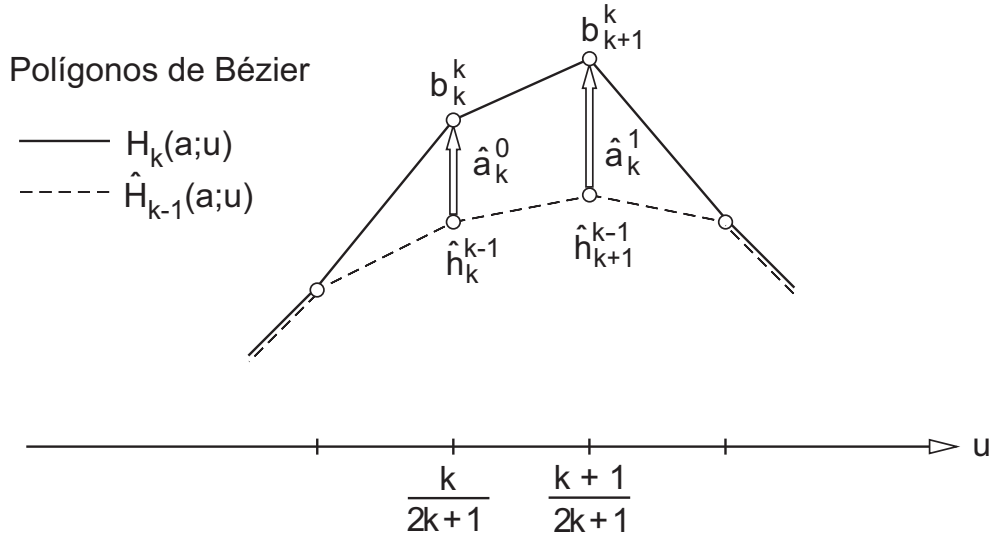


Figura 2.2: Interpretación geométrica de los coeficientes. (Fuente: Sánchez-Reyes, 1997)

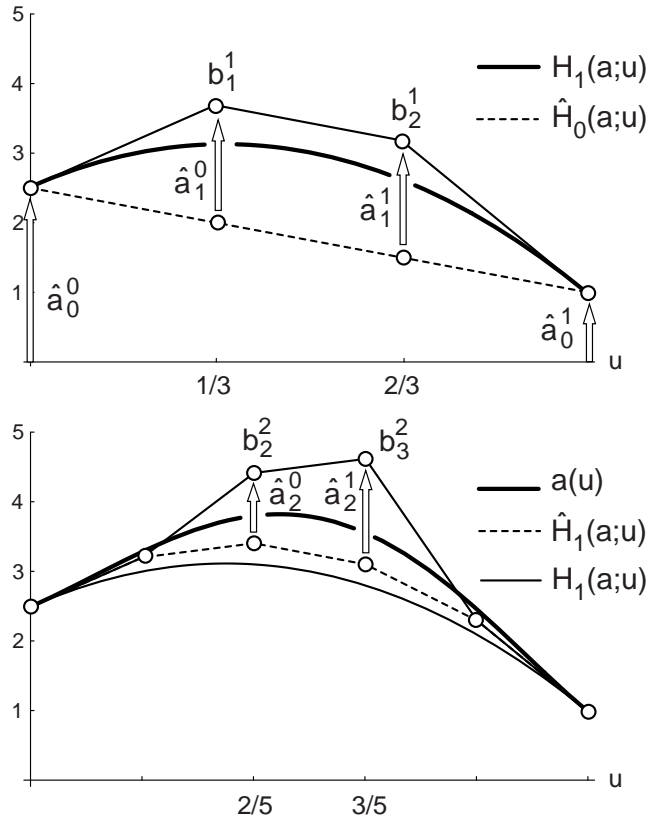


Figura 2.3: Generación de los coeficientes de Bézier de las aproximaciones de Hermite de una función de grado $n = 5$. (Fuente: Sánchez-Reyes, 1997)

Asimismo, se puede generar una representación de $H_k(a;u)$ mediante un polígono de control cuyos puntos son los pares:

$$\left(\frac{i}{2i+1}, b_i^i \right), \left(\frac{i+1}{2i+1}, b_{i+1}^i \right), \quad i = 0, \dots, k \quad (2.16)$$

correspondientes a las distintas interpolaciones de Hermite. Estos puntos son los indicados mediante flechas en las Fig. 2.2 y 2.3.

La Fig. 2.4 muestra el polígono de la función del ejemplo anterior. La principal ventaja de este polígono de control es que la reducción de grado se realiza simplemente eliminando pares de puntos. Esta operación genera el interpolador de Hermite de $a(u)$ (Fig. 2.5).

En el caso paramétrico, simplemente remplazaremos en la expresión (2.16):

$$\left(\frac{i}{2i+1}, b_i^i \right) \rightarrow \mathbf{b}_i^i, \quad \left(\frac{i+1}{2i+1}, b_{i+1}^i \right) \rightarrow \mathbf{b}_{i+1}^i.$$

Los coeficientes $\hat{\mathbf{a}}_k^0, \hat{\mathbf{a}}_k^1$ se pueden interpretar como vectores de control, ya que representan la diferencia entre puntos de control de dos curvas de Bézier.

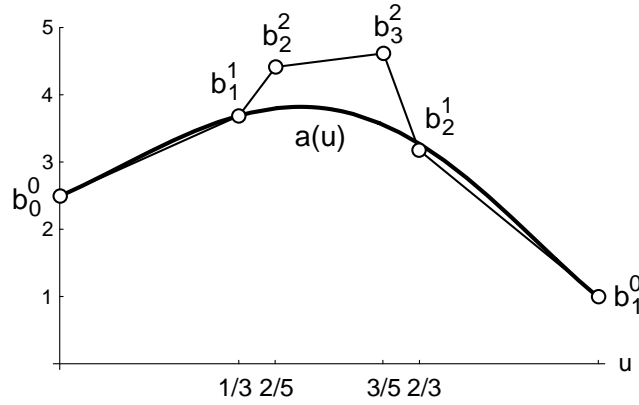


Figura 2.4: Polígono de control de una función de grado $n = 5$.

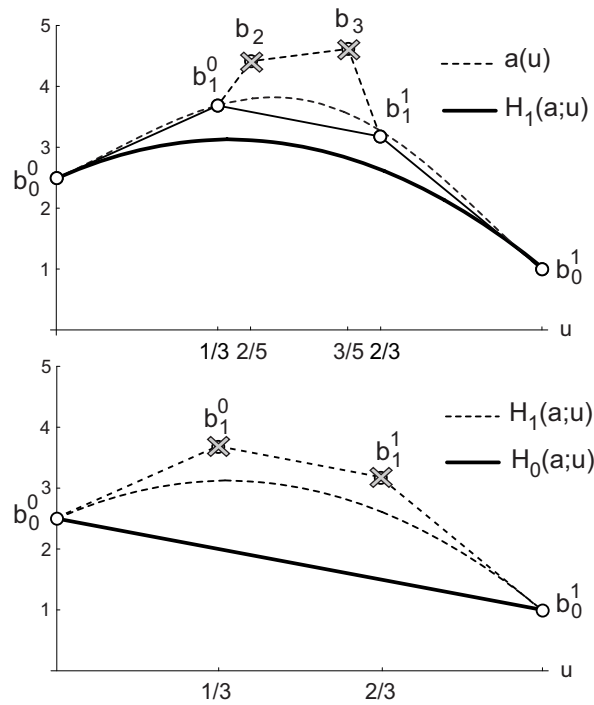


Figura 2.5: Sucesivas reducciones de grado de una función de grado $n = 5$.

(Fuente: Sánchez-Reyes, 1997)

2.6 Polinomios bivariados

La mayoría de los conceptos y definiciones del caso univariado pueden aplicarse al caso bivariado. Un polinomio bivariado $a(u,v)$ de grado $(m, n) = (2p + 1, 2q + 1)$ definido sobre el dominio $(u, v) \in [0,1] \times [0,1]$ se puede expresar en la base *s*-monomial como:

$$a(u,v) = \sum_{i=0}^p \sum_{j=0}^q a_{i,j}(u,v) s^i t^j, \quad \begin{aligned} s &= (1-u)u \\ t &= (1-v)v \end{aligned}$$

donde los coeficientes:

$$a_{i,j}(u,v) = \{(1-u), u\} \begin{bmatrix} a^{00} & a^{01} \\ a^{10} & a^{11} \end{bmatrix}_{i,j} \begin{Bmatrix} 1-v \\ v \end{Bmatrix}$$

son funciones bilineales. De forma análoga al caso univariado (2.5), los coeficientes de una *s*-serie bivariada se definen por una matriz de 2 x 2 cuyos elementos corresponden con su representación de Bézier (Fig. 2.6):

$$a_{i,j}(u,v) \rightarrow a_{i,j} = \begin{bmatrix} a^{00} & a^{01} \\ a^{10} & a^{11} \end{bmatrix}_{i,j}. \quad (2.17)$$

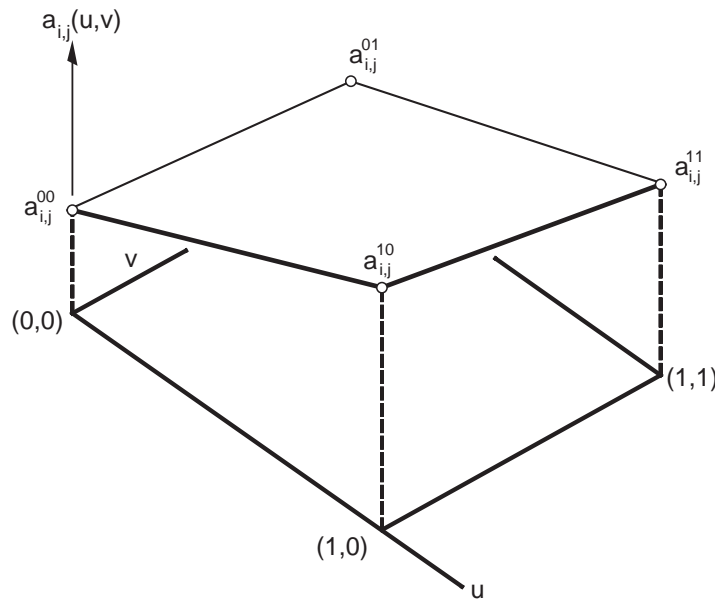


Figura 2.6: Interpretación de $a_{i,j}$ como una función bilineal en la forma de Bézier.

(Fuente: Sánchez-Reyes, 2000)

Operaciones básicas con s-series bivariadas

A continuación se exponen las operaciones con s-series bivariadas utilizadas en este trabajo.

Suma. Dadas dos s-series bivariadas $a(u)$ y $b(u)$, los coeficientes de la suma $c(u) = a(u) + b(u)$ se obtienen simplemente sumando los coeficientes del mismo orden.

$$c_{i,j} = \begin{bmatrix} a^{00} + b^{00} & a^{01} + b^{01} \\ a^{10} + b^{10} & a^{11} + b^{11} \end{bmatrix}_{i,j}.$$

Producto. Dadas dos funciones bilineales $a(u)$ y $b(u)$, el producto de ambas $c(u,v) = a(u,v) \cdot b(u,v)$ puede expresarse como:

$$ab = \begin{bmatrix} a^{00}b^{00} & a^{01}b^{01} \\ a^{10}b^{10} & a^{11}b^{11} \end{bmatrix} - s \begin{bmatrix} a_u^0 b_u^0 & a_u^1 b_u^1 \\ a_u^0 b_u^0 & a_u^1 b_u^1 \end{bmatrix} - t \begin{bmatrix} a_v^0 b_v^0 & a_v^1 b_v^1 \\ a_v^0 b_v^0 & a_v^1 b_v^1 \end{bmatrix} + st(a_u^1 - a_u^0)(b_u^1 - b_u^0) \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

$$\begin{aligned} a_u^0 &= a^{10} - a^{00} & a_v^0 &= a^{01} - a^{00} \\ a_u^1 &= a^{11} - a^{01} & a_v^1 &= a^{11} - a^{10} \end{aligned}.$$

Por tanto, para multiplicar dos s-series bivariadas, como en el caso univariado, basta con agrupar productos de coeficientes, operación muy sencilla si se utilizan convoluciones discretas de matrices (Sánchez-Reyes, 2000).

Capítulo 3

Generación de aproximaciones spline polinómicas mediante s-series

En este capítulo se han desarrollado un conjunto de herramientas necesarias para calcular de manera eficiente la aproximación de funciones univariadas y curvas mediante s-series, consiguiendo finalmente una representación expresada en el estándar NURBS adecuada para su posterior incorporación en un programa de CAD/CAM.

En primer lugar se resumen la convergencia de las s-series y el proceso de subdivisión para aquellas funciones que poseen singularidades. Este proceso persigue evitar estas singularidades, generando una aproximación convergente. También se puede aplicar a los casos en que se desean calcular aproximaciones de grado reducido para valores del error pequeños, algo necesario ya que los programas de CAD/CAM tienen limitado el grado máximo admisible.

Una vez que se ha calculado la s-serie de la función que se desea aproximar, se convierte a la base de Bernstein, uniendo cada uno de los tramos de Bézier en que se ha subdividido. El resultado final es un spline Hermítico (Grisoni et al., 1999) más adecuado para aplicaciones CAD/CAM que la representación mediante tramos de Bézier empalmados.

3.1 Convergencia de las s-series

En este apartado se resumen los resultados descritos por Sánchez-Reyes (2000) sobre la convergencia de las s-series, formulada en el plano complejo al igual que las series de Taylor. Estos resultados proceden de aplicar los teoremas clásicos de convergencia para procesos de interpolación (Davis, 1975) al caso particular de la interpolación simétrica de Hermite en dos puntos.

La serie de Taylor $a(z)$ de una función $f(z)$ de variable compleja en un punto z_0 viene dada por:

$$a(z) = \sum_{i=0}^{\infty} c_i (z - z_0)^i, \quad c_i = \frac{f^{(i)}(z_0)}{i!}. \quad (3.1)$$

Una función $f(z)$ es *analítica* en una región A si es analítica en cada punto de A . Si una función es analítica en z_0 entonces admite una representación como una serie de Taylor (3.1) alrededor de z_0 con un *radio de convergencia* $r > 0$ (Fig. 3.1). Si la función $f(z)$ no es analítica en z_0 entonces se dice que presenta una singularidad en z_0 . El radio de convergencia r es el radio del círculo más grande con centro en z_0 que se puede dibujar en A , de tal forma que no contenga ninguna singularidad en su interior. Dentro de este círculo la serie de Taylor converge uniformemente a $f(z)$ y diverge fuera de él.

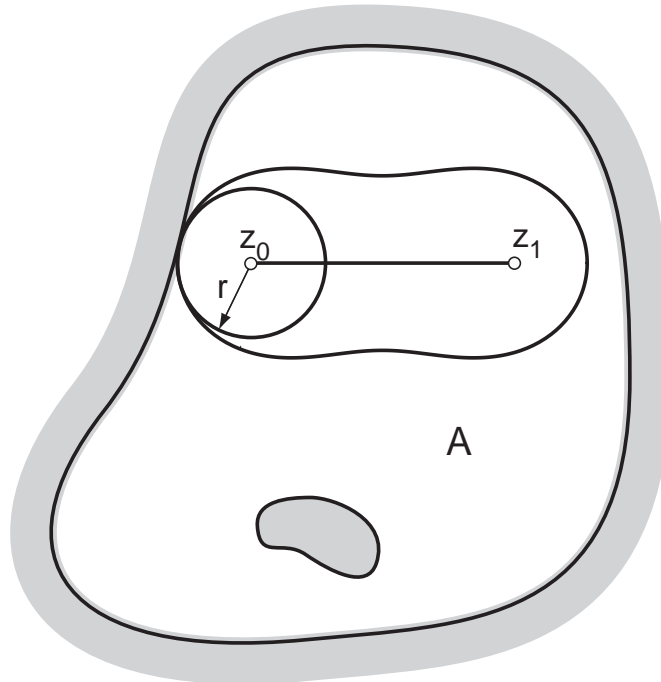


Figura 3.1: Áreas de convergencia en el plano complejo de una serie de Taylor y de Hermite.

(Fuente: Sánchez-Reyes, 2000)

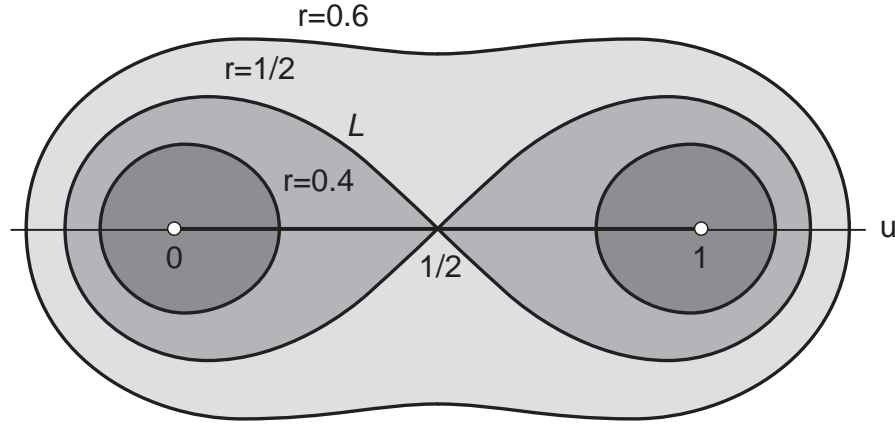


Figura 3.2: Áreas cofocales de convergencia en el plano complejo para las s-series.

(Fuente: Sánchez-Reyes, 2001b)

Dado que una s-serie es una serie de potencias en $s(z) = (1 - z)z$, simplemente se deben reemplazar los círculos con centro z_0 y radio r por curvas de la forma:

$$|s(z)| = r^2, \quad s(z) = (1 - z)z. \quad (3.2)$$

Estas curvas (3.2), conocidas como *Curvas de Cassini* (Neddham, 1997), son el lugar geométrico de puntos z tal que el producto de sus distancias a los focos $\{z = 0, z = 1\}$ es igual a la constante r^2 .

La s-serie de una función $f(z)$ converge uniformemente a $f(z)$ dentro de la curva más grande de Cassini de focos $z = 0$ y $z = 1$ que no contenga ninguna singularidad y diverge fuera. La Fig. 3.2 muestra las distintas curvas cofocales generadas al variar r , pertenecientes a 3 familias diferentes:

- $0 < r < 1/2$: Dos óvalos de Cassini, alrededor de los focos $z = 0$ y $z = 1$ respectivamente.
- $r = 1/2$: *Lemniscata de Bernoulli L*, figura con forma parecida al símbolo ∞ con un punto doble en $u = 1/2$.
- $r > 1/2$: Un contorno cerrado que contiene a ambos focos.

En la mayoría de las aplicaciones sólo interesa la convergencia sobre el intervalo real unitario $[0,1]$ de definición. La convergencia sobre el intervalo $[0,1]$ estará garantizada cuando dicho intervalo esté completamente contenido en el área compleja de convergencia. La curva de Cassini con $r = 1/2$ es la de menor área que contiene completamente al intervalo $[0,1]$, por lo que para garantizar la convergencia de la s-serie, la lemniscata de Bernoulli L no debe contener ninguna singularidad. No obstante, si L contiene alguna singularidad z^* , esta se puede evitar mediante la subdivisión del intervalo.

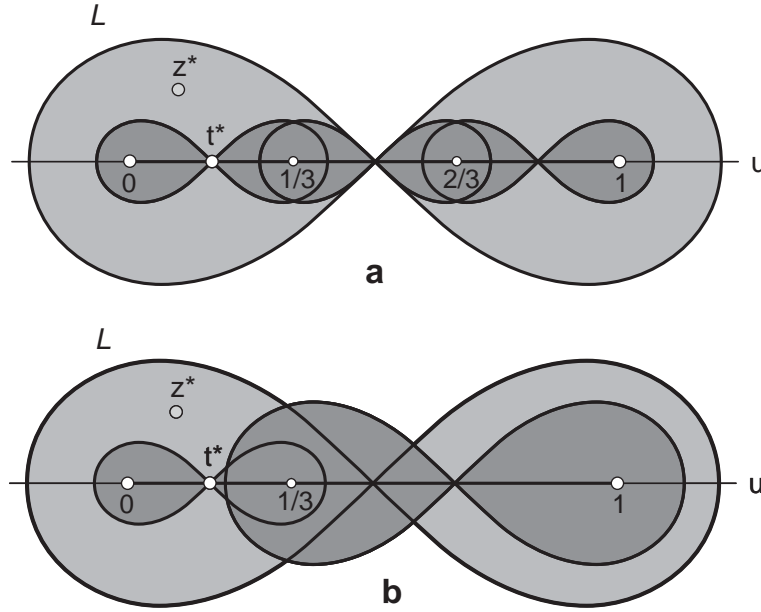


Figura 3.3: (a) Subdivisión uniforme. (b) Subdivisión no uniforme.

(Fuente: Sánchez-Reyes, 2001b)

3.2 Proceso de subdivisión

Como se ha expuesto en el apartado anterior, el área de convergencia en el plano complejo de una *s-series* es una curva de Cassini, y para garantizar la convergencia de la *s-series* de una función $f(u)$ sobre un intervalo real $u \in [0, 1]$, la lemniscata L no debe contener ninguna singularidad. La *s-series* de $f(u)$ convergerá uniformemente a $f(u)$ si L no contiene ninguna singularidad z^* , y no convergerá a la función $f(u)$ si contiene alguna singularidad.

Cuando L contiene alguna singularidad esta se puede evitar mediante la subdivisión del intervalo original, excepto cuando la singularidad es real y se encuentra dentro del intervalo $u \in [0, 1]$. La Fig. 3.3a muestra la estrategia de subdivisión uniforme para una singularidad imaginaria z^* no contenida en el eje real $u \in [0, 1]$, donde el intervalo unitario ha sido subdividido en tres segmentos de igual longitud. Si se calcula la *s-series* de cada segmento, las nuevas lemniscatas L , con focos $\{0, 1/3\}$, $\{1/3, 2/3\}$ y $\{2/3, 1\}$, son más pequeñas que la original y están contenidas dentro de la lemniscata original. De este modo, tras las subdivisiones necesarias se puede conseguir mantener todas las singularidades fuera de la lemniscata L de cada tramo de subdivisión, consiguiendo una *s-series* convergente en cada tramo, a la vez que se abarca todo el intervalo original de definición.

Si en lugar de subdividir el intervalo original en tres segmentos de igual longitud, se divide en 2 segmentos de distinta longitud, resultan dos lemniscatas L , con focos $\{0, 1/3\}$ y $\{1/3, 1\}$, no contenidas completamente en la original. Teniendo en cuenta la localización de las singularidades y escogiendo el punto de subdivisión de tal forma que las

singularidades queden fueran de las nuevas lemniscatas L , se tiene que una subdivisión no uniforme en dos trozos es más eficiente que una uniforme en 3 tramos (Fig. 3.3b). La calidad de la aproximación aumentará con la distancia entre las nuevas lemniscatas L y la singularidad.

En el caso de una singularidad real t^* situada dentro del intervalo $u \in [0,1]$ (Fig. 3.3a), la subdivisión se generará de forma que t^* esté localizada justo en el centro de uno de los intervalos en que se ha subdividido el intervalo original. Como la singularidad t^* coincide con el punto doble de la lemniscata L correspondiente, la convergencia está garantizada en todo el intervalo excepto en t^* . Un cálculo simple indica que únicamente será necesaria una subdivisión si se verifica:

$$t^* \in U, \quad U = [1/2 - \sqrt{2}/4, 1/2 + \sqrt{2}/4]. \quad (3.3)$$

Ejemplos de aproximación de funciones con singularidades

Como primer ejemplo se ha aproximado la función $f(u) = \log[(u - t^*)(u - z^*)(u - \bar{z}^*)]$ sobre $u \in [0,1]$, con una singularidad real $t^* = -0.07$ y dos complejas $z^* = 0.2 \pm 0.2i$.

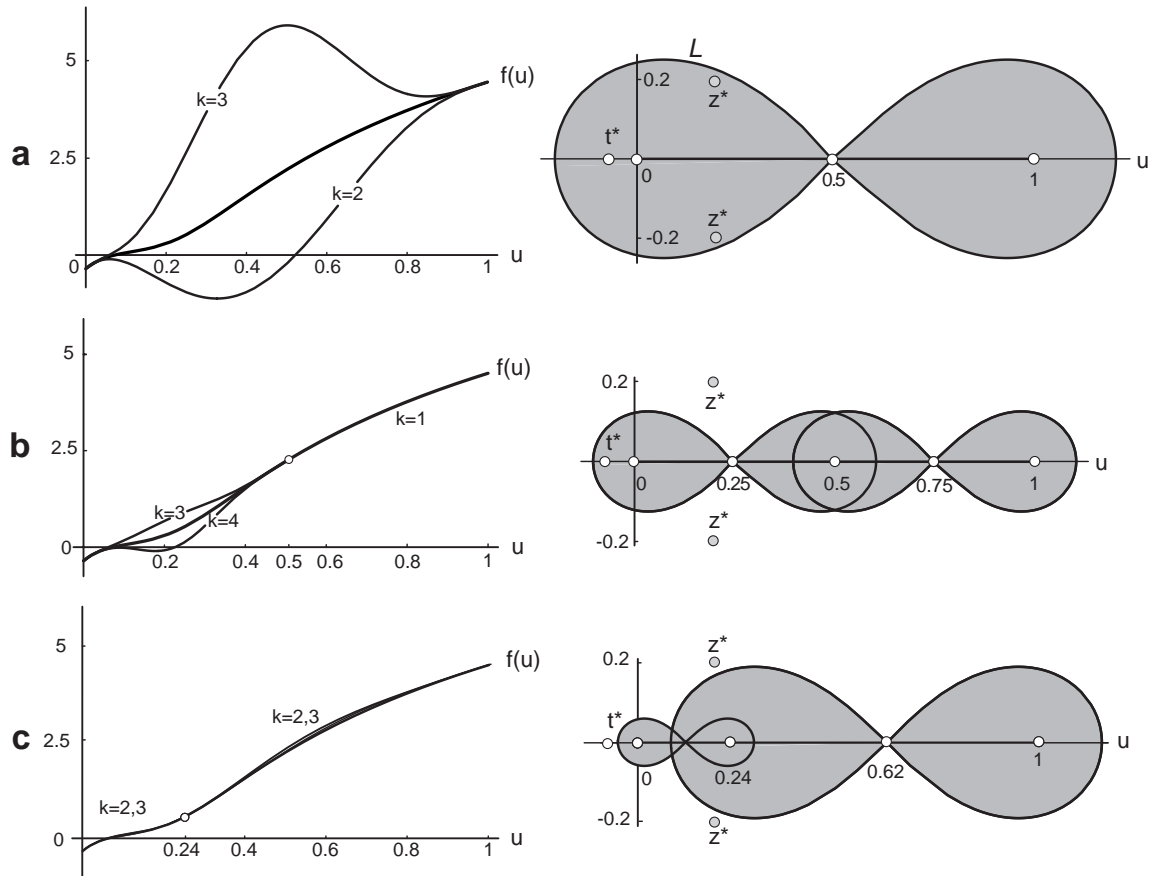


Figura 3.4: Áreas de convergencia y aproximación de Hermite de $f(u) = \log[(u - t^*)(u - z^*)(u - \bar{z}^*)]$, con $t^* = -0.07$ y $z^* = 0.2 \pm 0.2i$. (a) Sin subdivisión. (b) Subdivisión uniforme. (c) Subdivisión no uniforme.

La lemniscata L que abarca todo el intervalo $u \in [0,1]$ contiene en su interior las 3 singularidades, por lo que la *s-series* correspondiente será divergente (Fig. 3.4a). En principio se podría pensar en una subdivisión uniforme del intervalo. Si se calcula la *s-series* de cada segmento, las nuevas lemniscatas L con focos $\{0, 0.5\}$ y $\{0.5, 1\}$ son más pequeñas que la original, pero la lemniscata del tramo de la izquierda sigue conteniendo la singularidad real $t^* = -0.07$ (Fig. 3.4b). Esta singularidad hace que la *s-series* de este tramo sea divergente, mientras que la serie del tramo derecho disfruta de convergencia.

Ante estos resultados, se procederá a generar una subdivisión no uniforme, de manera que las 3 singularidades queden fuera de las lemniscatas L de cada tramo. Las nuevas lemniscatas L con focos $\{0, 0.24\}$, $\{0.24, 1\}$ son más pequeñas que la original y no contienen ninguna singularidad (Fig. 3.4c), lo que garantiza la convergencia en todos los tramos.

Como segundo ejemplo se ha aproximado la función $f(u) = \log[|u - 0.2|]$ en el intervalo $u \in [0,1]$, que presenta una singularidad real $t^* = 0.2$ situada dentro del intervalo unitario. En este caso únicamente es necesario efectuar una subdivisión ya que t^* no está contenido dentro del intervalo U (3.3), haciendo coincidir t^* con el centro de uno de los dos intervalos de subdivisión (Fig. 3.5a). De esta forma se asegura la convergencia de la aproximación en todo el intervalo excepto en la singularidad, donde la función tiende a infinito (Fig. 3.5b).

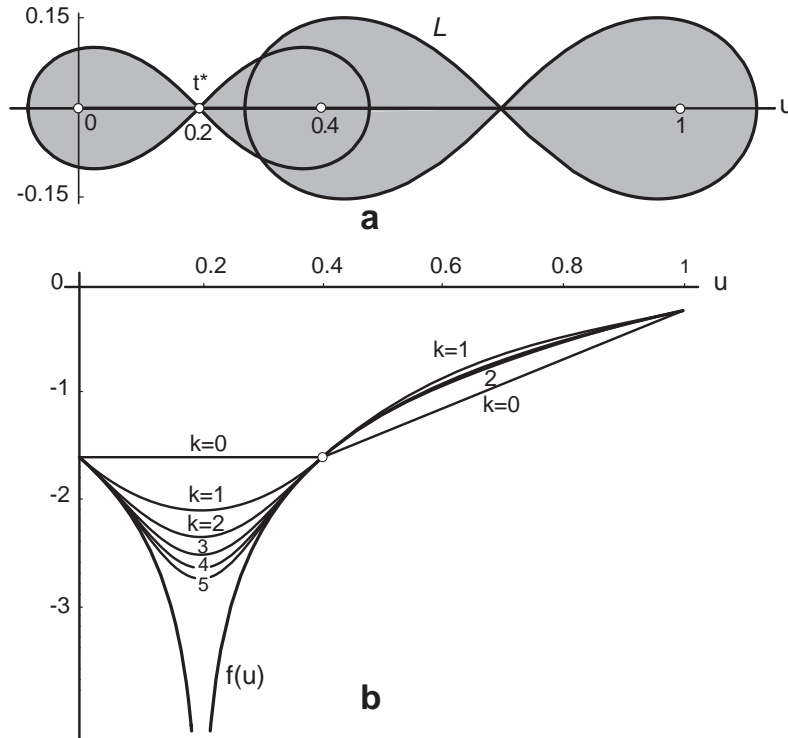


Figura 3.5: Áreas de convergencia y aproximación de Hermite de $f(u) = \log[|u - 0.2|]$.

3.3 Conversión entre las bases s-monomial y Bernstein

Una vez que se ha calculado la aproximación mediante s-series de una entidad dada, interesa obtener su representación de Bézier para su posterior incorporación en un programa de CAD/CAM. En este apartado se describen los procesos de conversión entre la base s-monomial y la de Bernstein, tanto para funciones univariadas como para bivariadas.

Las matrices de conversión entre la base de Bernstein y la s-monomial toman unas expresiones muy simples usando la funciones bases escaladas de Bernstein (Farouki y Rajan, 1988). En la practica, interesará obtener aproximaciones, que no son más que series truncadas. Como el grado n de una s-serie truncada es $n = 2k + 1$, donde k es el número de términos de la s-serie, la representación de Bézier será de grado impar.

A diferencia de la base monomial, las matrices de conversión entre la base s-monomial y la de Bernstein no están mal condicionadas, es decir, la transformación no conlleva pérdida significativa de precisión en la transformación (Sánchez-Reyes, 1997).

Conversión de la base s-monomial a la base de Bernstein

Funciones univariadas

Dado un polinomio $a(u)$ en la base s-monomial, se definen los vectores columna \mathbf{a}^0 y \mathbf{a}^1 que contienen los coeficientes de orden k (grado $n = 2k + 1$) de $a(u)$:

$$\mathbf{a}^0 = \begin{pmatrix} a_0^0 \\ \cdot \\ \cdot \\ a_k^0 \end{pmatrix}, \quad \mathbf{a}^1 = \begin{pmatrix} a_0^1 \\ \cdot \\ \cdot \\ a_k^1 \end{pmatrix}, \quad (3.4)$$

y $\tilde{\mathbf{b}} = (\tilde{b}_0, \dots, \tilde{b}_n)^T$ el vector columna que contiene los coeficientes escalados de Bernstein de $a(u)$. Los \tilde{b}_i se calculan partiendo de los coeficientes de Bernstein estándar b_i como:

$$\tilde{b}_i = \binom{n}{i} b_i.$$

Dada la simetría de los polinomios de Bernstein, $\tilde{\mathbf{b}}$ se calcula a partir de los coeficientes (3.4) como:

$$\tilde{\mathbf{b}} = \mathbf{M}^n \cdot \mathbf{a}^0 + R(\mathbf{M}^n \cdot \mathbf{a}^1), \quad (3.5)$$

donde el operador $R(\mathbf{M})$ invierte el orden de los elementos de un vector o las filas de una matriz \mathbf{M} , y \mathbf{M}^n representa una matriz de dimensiones $(n + 1) \times (k + 1)$, cuyos coeficientes no nulos son (Sánchez-Reyes, 1997):

$$\mathbf{M}_{j,q}^n = \binom{n-2q-1}{j-q}, \quad \begin{matrix} q = 0, \dots, k \\ j = 0, \dots, n-k-1 \end{matrix}.$$

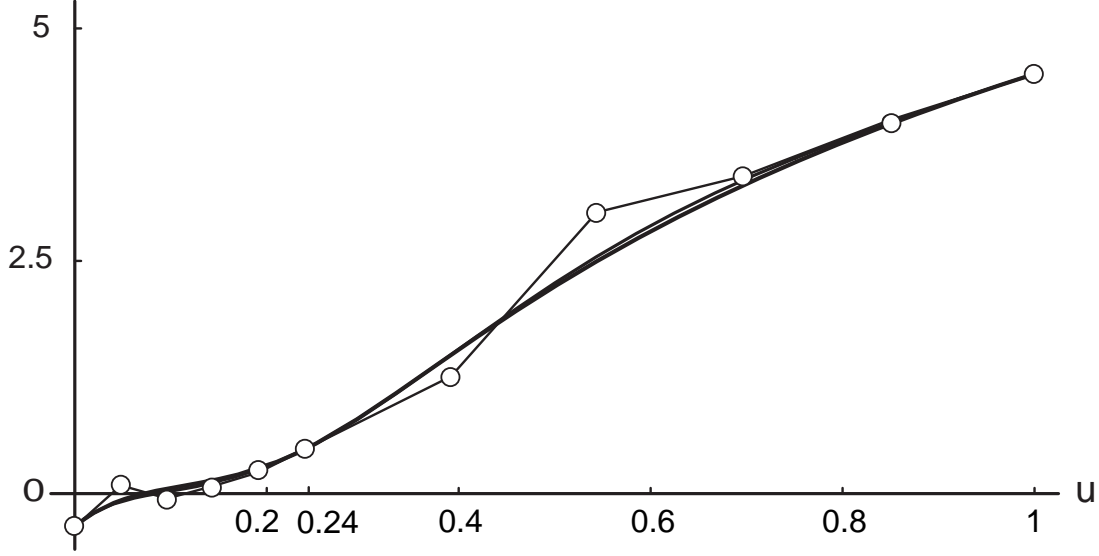


Figura 3.6: Representación de Bézier de la aproximación de Hermite de grado $n = 5$

de $f(u) = \log[(u - t^*) (u - z^*) (u - z^*)]$, con $t^* = -0.07$ y $z^* = 0.2 \pm 0.2i$.

En el ejemplo de la Fig. 3.6, partiendo de la s-serie calculada en el ejemplo del apartado anterior (Fig. 3.4c), se ha realizado el cambio de base (3.5), calculando finalmente las ordenadas de los puntos de control de la representación de Bézier no paramétrica (Farin, 2002) de cada uno de los tramos.

Funciones bivariadas

Dado un polinomio bivariado $a(u, v)$ de grado $(m, n) = (2p + 1, 2q + 1)$ definido en la base s-monomial, de forma análoga al caso univariado, se definen 4 submatrices de dimensión $(p + 1) \times (q + 1)$, correspondientes a los distintos coeficientes (2.17), \mathbf{a}^{00} , \mathbf{a}^{01} , \mathbf{a}^{10} y \mathbf{a}^{11} , de $a(u, v)$.

Se define también la matriz $\tilde{\mathbf{b}}$ de dimensión $(m + 1) \times (n + 1)$ que contiene los coeficientes escalados de Bernstein. Los $\tilde{b}_{i,j}$ se calculan partiendo de los coeficientes de Bernstein estándar $b_{i,j}$ como:

$$\tilde{b}_{i,j} = \binom{m}{i} \binom{n}{j} b_{i,j}. \quad (3.6)$$

Los coeficientes escalados de Bernstein (3.6), en función de los coeficientes s-monomiales se calculan como:

$$\tilde{\mathbf{b}} = \mathbf{M}^n \cdot \mathbf{a}^{00} \cdot \mathbf{M}^m + R(\mathbf{M}^n \cdot \mathbf{a}^{10} \cdot \mathbf{M}^m) + C(\mathbf{M}^n \cdot \mathbf{a}^{01} \cdot \mathbf{M}^m) + C(R(\mathbf{M}^n \cdot \mathbf{a}^{11} \cdot \mathbf{M}^m)) \quad (3.7)$$

donde el operador $R(\mathbf{M})$, definido en el apartado anterior, invierte el orden de las filas y el operador $C(\mathbf{M})$ invierte el orden de las columnas.

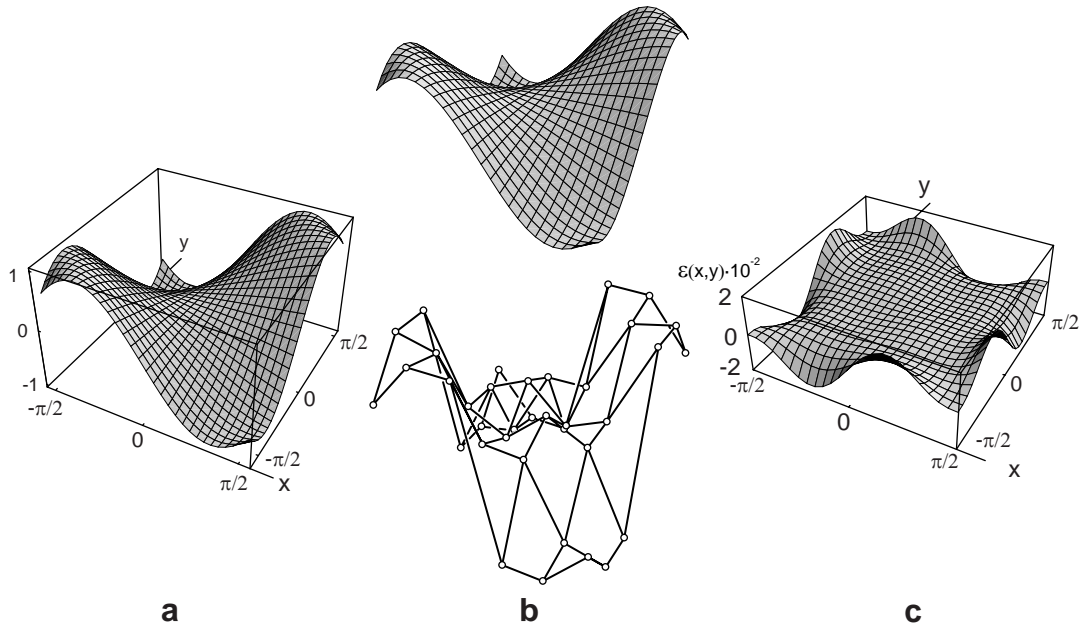


Figura 3.7: $f(x, y) = \sin(x \cdot y)$. (a) Función exacta. (b) Aproximación de Hermite de grado $(m, n) = (5, 5)$. (c) Error $\varepsilon(x, y)$.

En el ejemplo de la Fig. 3.7 se ha aproximado la función $f(x, y) = \sin(x \cdot y)$, sobre el dominio $(x, y) \in [-\pi/2, \pi/2] \times [-\pi/2, \pi/2]$. A continuación se ha realizado el cambio de base (3.7) y obtenido los puntos de Bézier de la aproximación.

Conversión de la base de Bernstein a la base s-monomial

Existen aplicaciones en las que, partiendo de la representación de Bernstein de un objeto, es necesario calcular los coeficientes de la s-serie correspondiente, como es el caso de la deformación de un objeto basada en la composición de funciones polinómicas, que se verá en el Capítulo 6.

Funciones univariadas

Los coeficientes (3.4) de un polinomio $a(u)$ en la base s-monomial de grado $n = 2k + 1$ en función de los coeficientes escalados de Bernstein $\tilde{\mathbf{b}}$ se pueden calcular como:

$$\mathbf{a}^0 = \mathbf{N}^n \cdot \tilde{\mathbf{b}}, \quad \mathbf{a}^1 = C(\mathbf{N}^n) \cdot \tilde{\mathbf{b}} \quad (3.8)$$

donde \mathbf{N}^n es una matriz de dimensión $(k + 1) \times (n + 1)$, cuyos coeficientes no nulos son (Sánchez-Reyes, 1997):

$$\mathbf{N}_{j,q}^n = (-1)^{j-q} \binom{n-j-q}{j-q}, \quad \begin{array}{l} q = 0, \dots, k \\ j = q, \dots, k \end{array}$$

Funciones bivariadas

Los coeficientes de un polinomio bivariado $a(u, v)$ de grado $(m, n) = (2p + 1, 2q + 1)$ en la base *s*-monomial (2.17), en función de los coeficientes de Bernstein, vienen dados por:

$$\begin{aligned} \mathbf{a}^{00} &= \mathbf{N}^m \cdot \tilde{\mathbf{b}} \cdot \mathbf{N}^n, & \mathbf{a}^{10} &= C(\mathbf{N}^m) \cdot \tilde{\mathbf{b}} \cdot \mathbf{N}^n \\ \mathbf{a}^{01} &= \mathbf{N}^m \cdot \tilde{\mathbf{b}} \cdot R(\mathbf{N}^n), & \mathbf{a}^{11} &= C(\mathbf{N}^m) \cdot \tilde{\mathbf{b}} \cdot R(\mathbf{N}^n) \end{aligned} \quad (3.9)$$

3.4 Eliminación de nodos internos

Partiendo de la representación de Bézier de cada uno de los trozos en que se ha subdividido la entidad que se desea aproximar, se procede a la unión de estos, consiguiendo una representación B-spline Hermítica (Grisoni et al., 1999) más adecuada que la unión de varios tramos de Bézier, pues posee menor número de puntos de control.

La unión de varios tramos de Bézier se generará aplicando el algoritmo de eliminación de nodos internos (Piegl y Tiller, 1997), que no es otra cosa que el proceso inverso al algoritmo de inserción de nodos. El objetivo del algoritmo de eliminación de nodos es, cuando ello sea posible, reducir la multiplicidad de un nodo interno de una curva B-spline, sin modificar su forma. Al reducir la multiplicidad se reduce el número de puntos de control y se obtiene una representación más compacta.

Si el spline posee continuidad C^k en un nodo, se podrá reducir su multiplicidad original r hasta un valor mínimo $r' = n - k$. Para el caso del spline Hermítico resultante de concatenar tramos de Bézier de grado $n = 2k + 1$ con continuidad C^k , la multiplicidad de los nodos internos, que inicialmente es n , podrá reducirse hasta $r' = k + 1$. En este caso concreto, el algoritmo de inserción de k nodos adicionales en un nodo de multiplicidad $k + 1$ degenera en el de Casteljau. Por consiguiente, se pueden eliminar nodos vía de Casteljau considerando que la condición de continuidad C^k entre dos curvas de Bézier es que sus $k + 1$ puntos extremos resulten de subdividir una curva de grado k .

Por lo tanto, en una curva B-spline, se sustituirán los $k + 1$ puntos extremos $\mathbf{d}_i^I, \mathbf{d}_i^{II}$, de 2 curvas de Bézier $(i, i + 1)$ unidas en el nodo u_i :

$$\begin{aligned} \mathbf{d}_i^I &= \{\mathbf{b}_{n-i-k}, \dots, \mathbf{b}_{n-i}\} \\ \mathbf{d}_i^{II} &= \{\mathbf{b}_{n-i}, \dots, \mathbf{b}_{n-i+k}\} \end{aligned}$$

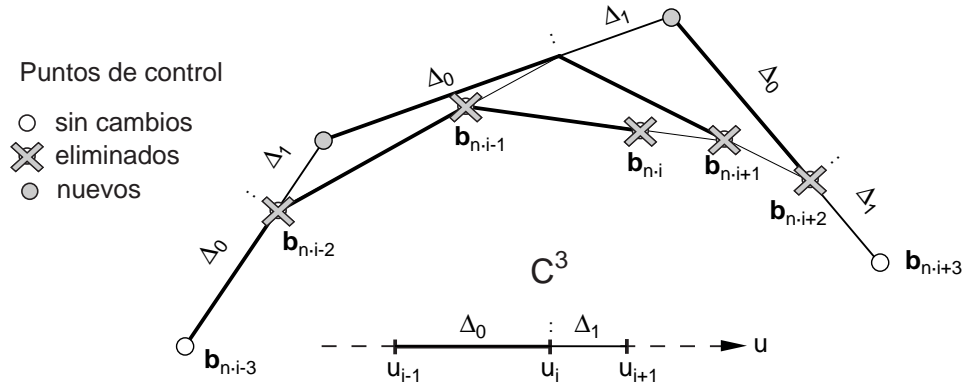


Figura 3.8: Unión C^3 de curvas de Bézier.

por los de la curva de grado k de la que resultan, cuyos puntos se calcularán aplicando el algoritmo de Casteljau mediante extrapolación (Farin, 2002) para un valor del parámetro $u = (u_{i+1} - u_{i-1}) / (u_i - u_{i-1})$ (Fig. 3.8). La extrapolación es un proceso numéricamente inestable, que debe evitarse para valores elevados del parámetro. Por consiguiente, el algoritmo de Casteljau lo aplicaremos a los puntos extremos \mathbf{d}_i^I cuando $\Delta_0 > \Delta_1$ y a los puntos \mathbf{d}_i^{II} invertidos cuando $\Delta_0 < \Delta_1$, con $\Delta_0 = (u_i - u_{i-1})$ y $\Delta_1 = (u_{i+1} - u_i)$. Este proceso se repetirá tantas veces como uniones entre curvas de Bézier existan. La Fig. 3.9 muestra un ejemplo de eliminación de nodos en un spline Hermítico de grado $n = 5$ compuesto por 3 curvas de Bézier.

Continuando con el ejemplo del apartado anterior (Fig. 3.4c) y partiendo de la representación de Bézier de la aproximación de cada uno de los tramos en que se ha subdividido la función (Fig. 3.6), se han concatenado ambas representaciones de Bézier de grado $n = 5$ (orden $k = 2$). Como resultado se tiene una curva B-spline que posee continuidad C^2 en el punto de unión. Como se ha expuesto anteriormente, la multiplicidad del nodo interno se podrá reducir hasta un valor $r' = 3$. Por consiguiente, se sustituirán los 3 puntos extremos $\{\mathbf{b}_3, \mathbf{b}_4, \mathbf{b}_5\}$, $\{\mathbf{b}_5, \mathbf{b}_6, \mathbf{b}_7\}$ de las 2 curvas de Bézier que constituyen la curva B-spline por los de la curva de grado k de la que resultan, generando la representación B-spline final (Fig. 3.10).

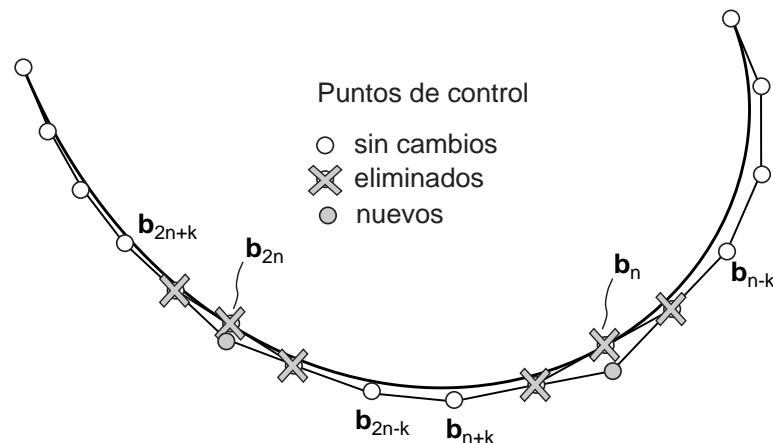


Figura 3.9: Eliminación de nodos en un spline Hermítico de grado $n = 5$.

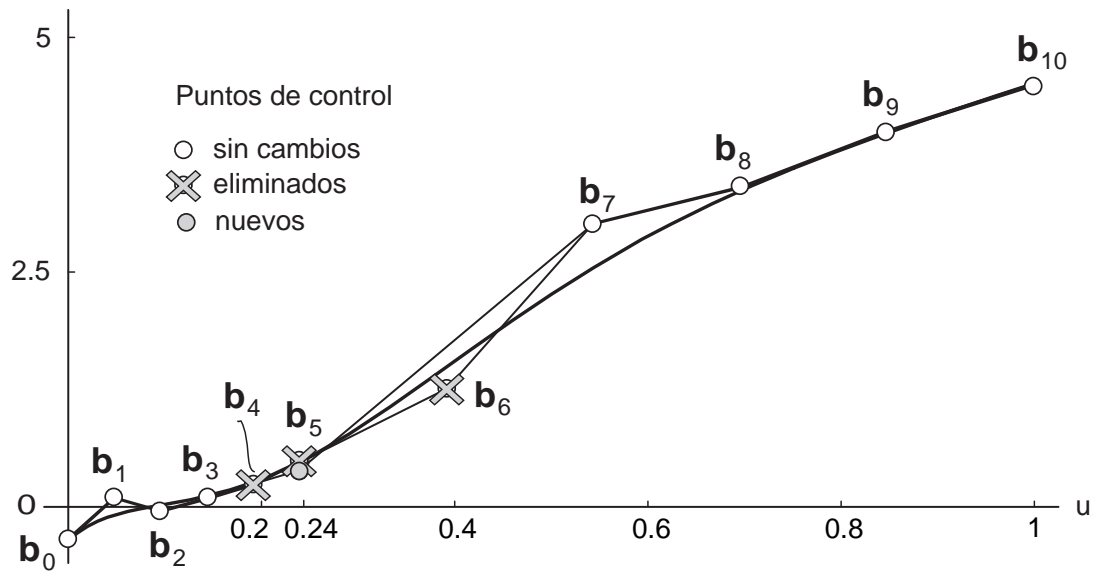


Figura 3.10: Representación B-spline de orden $k=2$ (grado $n=5$).

Capítulo 4

s-series frente a las series de Poisson para la representación de funciones analíticas

La representación paramétrica de curvas y superficies en CAD (Capítulo 1) emplea funciones polinómicas o racionales (Farin, 2002; Piegl y Tiller, 1997). Sin embargo, las funciones analíticas (Davis, 1975) proporcionan un marco necesario para la representación de funciones trascendentes no representables por los modelos polinómicos o racionales estándar, por ejemplo, curvas o superficies no algebraicas (Lawrence, 1972). Por esto, es interesante desarrollar una herramienta general para representar y manipular funciones analíticas de una manera eficiente.

Una primera opción viene proporcionada por el concepto de *B-base* (Peña, 1999) que generaliza la idea de la base de Bernstein a espacios funcionales de dimensión finita. Dado un espacio vectorial de funciones con bases normalizadas totalmente positivas, entre todas ellas existe una única base, llamada B-base, con propiedades óptimas de conservación de forma. Si un espacio tiene una B-base se puede definir una curva a través de un polígono de control de manera análoga al caso de Bézier. Estas curvas presentan todas las positivas propiedades geométricas de la representación de Bézier (convex-hull, variación disminuida, tangencia al polígono de control en los puntos finales, invariancia afín) además de la existencia de un algoritmo de subdivisión de tipo de Casteljau. Sin embargo, cada espacio vectorial de funciones tiene una B-base particular y el número de funciones base

requeridas es igual a la dimensión del espacio. Por lo tanto, no se pueden representar de manera unificada funciones analíticas arbitrarias.

También se han utilizado otras expansiones sobre intervalos finitos para aplicaciones CAD, como las series de Legendre para generar aproximaciones polinómicas del offset de una curva (Li y Hsu, 1998), de la inversa de funciones polinómicas (Farouki, 2000) o para reducciones de grado (Lee et al., 2002). En general, el cálculo de los coeficientes resulta complejo ya que involucra integrales. La resolución de estas integrales, salvo casos muy concretos como son las funciones polinómicas, precisa de métodos de integración numérica. Asimismo, la aproximación de una función mediante una serie de Legendre truncada no interpola dicha función en los puntos extremos, lo que produce huecos en la unión de varias expansiones. Por consiguiente, las series de Legendre no resultan adecuadas para aplicaciones CAD.

Morin y Goldman (2000a, 2000b, 2001a, 2001b) han presentado recientemente varios trabajos, basados en el empleo de las series de Poisson, para la aproximación de funciones analíticas en CAD. Este modelo se puede interpretar como el análogo de la representación polinómica de Bézier para la representación de funciones analíticas en intervalos semi-infinitos. Los coeficientes de estas series definen un polígono de control que presenta las positivas propiedades geométricas de la formulación de Bézier.

En este capítulo se compara la formulación de las series de Poisson con las s-series (2.1). Primero se realizará un breve repaso de las series de Poisson, pasando a continuación a una serie de ejemplos donde se compararán ambos métodos de aproximación.

4.1 Series de Poisson

El desarrollo de Poisson alrededor del punto $t = 0$ de una función $f(t)$ dada se define como:

$$f(t) = \sum_{k=0}^{\infty} p_k b_k(t), \quad b_k(t) = \frac{e^{-t} t^k}{k!}, \quad (4.1)$$

donde los coeficientes de Poisson p_k se pueden calcular fácilmente (Morin y Goldman, 2000a) mediante la expansión de Taylor de la función $g(t) = e^t f(t)$ en $t = 0$:

$$p_k = g^{(i)}(0) = \sum_{i=0}^k \binom{k}{i} f^{(i)}(0), \quad g(t) = e^t f(t). \quad (4.2)$$

Las funciones base de Poisson (4.1), representadas en la Fig. 4.1, son positivas sobre el semi-eje $t \in [0, \infty)$ y forman una partición de la unidad. Estas funciones pueden interpretarse como el caso límite de los polinomios de Bernstein de grado n cuando $n \rightarrow \infty$, definidos

sobre $t \in [0, n]$. Por lo tanto, el sumatorio (4.1) se puede interpretar como el caso limite $n \rightarrow \infty$ de una curva no paramétrica de Bézier de grado n sobre $t \in [0, n]$. Los puntos de control de esta curva no paramétrica son $\mathbf{p}_k = (k, p_k)$, es decir, con abscisas de Bézier $\{0, 1, 2, \dots\}$ uniformemente espaciadas a lo largo de eje x . Este polígono de control infinito hereda las siguientes propiedades de la representación de Bézier:

- Imita la forma de $f(t)$ en la región positiva cercana al origen $t = 0$. Los puntos \mathbf{p}_k tienen un efecto de empuje/arrastre sobre la curva, ya que $b_k(t)$ presenta unimodalidad, alcanzando su máximo en $t = k$.
- El polígono de control es tangente a $f(t)$ en \mathbf{p}_0 .
- Si la serie converge en $[0, \infty)$, las propiedades de convex-hull y variación disminuida se mantienen.
- Precisión lineal: Las ordenadas de los puntos de control de una función lineal $f(t)$ son $p_k = f(k)$.

Una *función de Poisson* es la representación (4.2) convergente en una región cercana al origen de una función. Estas definiciones se pueden extender de manera sencilla al caso paramétrico, donde la representación de Poisson presenta invariancia afín.

Si se desea tener una nueva representación de $f(t)$, con puntos de control $\mathbf{q}_k = (k\rho, q_k)$ con $k = \{0, 1, 2, \dots\}$, el resultado es equivalente a escalar el gráfico y el polígono de $f(\rho t)$ un factor de escala ρ a lo largo del eje x . Las nuevas ordenadas q_i son las de la función $f(\rho t)$ y las nuevas funciones base son $b_k(t/\rho)$. Las series de Poisson se describen como:

$$f(t) = \sum_{k=0}^{\infty} q_k b_k(t/\rho), \quad q_k = \sum_{i=0}^k \binom{k}{i} \rho^i f^{(i)}(0). \quad (4.3)$$

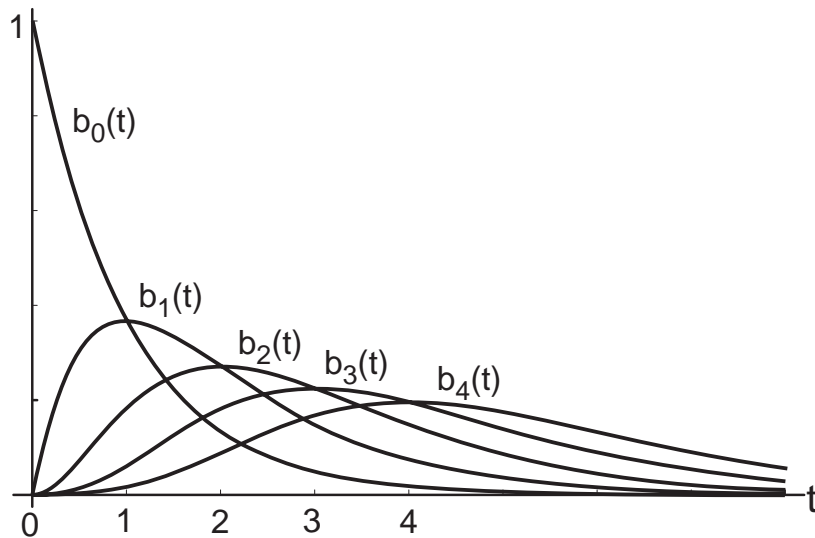


Figura 4.1: Funciones base de Poisson $b_k(t)$, $k = 0, \dots, 4$.

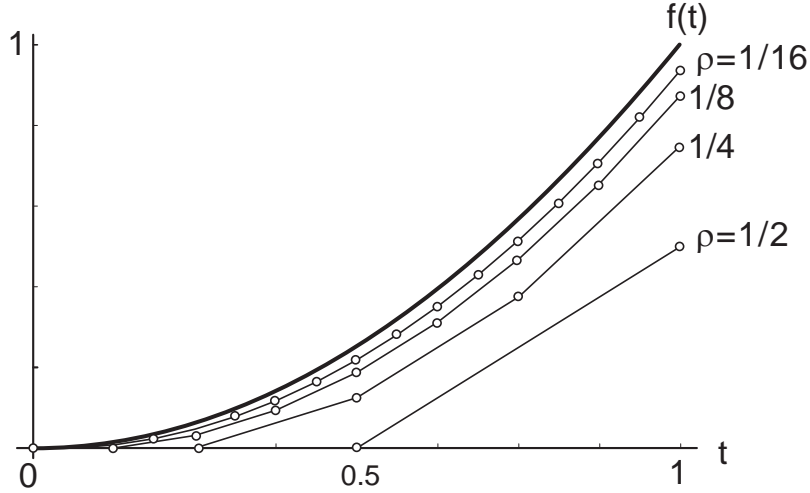


Figura 4.2: Polígonos de control de Poisson de $f(t) = t^2$, $t \in [0, 1]$.

En el marco polinómico, DeRose (1988) indica la equivalencia entre la subdivisión y la reparametrización lineal. Las nuevas ordenadas k -ésimas q_k (4.3) pueden calcularse partiendo de las originales $\{p_i\}_{i=0}^k$ (4.2) mediante el algoritmo de subdivisión de tipo de Casteljau (subdivisión del lado izquierdo) usando un sumatorio (Morin y Goodman, 2001a).

Como para las curvas de Bézier, en el caso $0 < \rho < 1$ (representación refinada), una subdivisión reiterada genera una secuencia de polígonos de control que convergen a $f(t)$ sobre el intervalo de convergencia de la serie. Estos polígonos proporcionan una aproximación lineal a trozos de $f(t)$. Esta propiedad queda ilustrada en la Fig. 4.2 para la función $f(t) = t^2$, donde como factor de escala se ha tomado $\rho = 2^{-i}$, con $i = 1, 2, 3, 4$. Hermann (2002) demostró que una determinada familia de bases que incluye la base de Bernstein y la de Poisson gozan de esta convergencia del polígono refinado.

Si se busca una representación de $f(t)$ sobre un intervalo arbitrario $[t_0, \infty)$ en lugar de $[0, \infty)$, con puntos de control sobre abscisas $\{t_0, t_0+1, t_0+2, \dots\}$, las nuevas ordenadas son las de $f(t - t_0)$. No obstante, si se intentan calcular mediante el algoritmo de subdivisión de tipo de Casteljau (subdivisión del lado derecho), el cálculo se vuelve muy complejo, ya que involucra sumas infinitas.

4.2 Convergencia de las series de Poisson

En esta sección se resumen los resultados más importantes relativos a la convergencia de las series de Poisson, básicamente igual a la convergencia de las series de Taylor (Needham, 1997) (Capítulo 3). Dado que estos resultados están formulados en el plano complejo, a partir de ahora se considerarán funciones complejas $f(z)$.

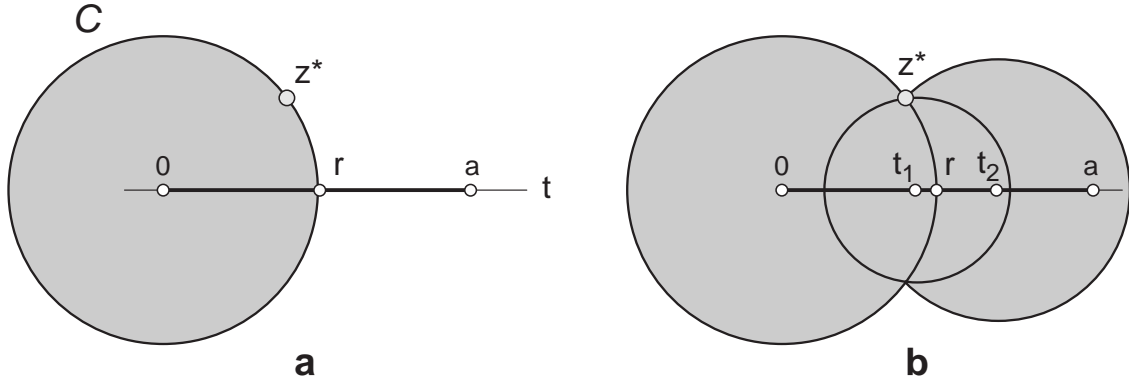


Figura 4.3: (a) Círculo C de convergencia en el plano complejo para las series de Taylor y Poisson. (b) Singularidad z^* evitada mediante extensión analítica.

El área de convergencia de la serie de Poisson de una función $f(z)$ alrededor del origen $z = 0$, al igual que para las series de Taylor, es un círculo de radio r centrado en $z = 0$ (Fig. 4.3a). Dado un intervalo real $[0, a]$, se puede garantizar la convergencia de la serie de Poisson en este intervalo si se verifica que $r > a$, es decir, si el círculo de convergencia C abarca todo el intervalo.

Para evitar una singularidad z^* situada fuera del eje real, Morin y Goldman (2001a) emplean una continuación analítica para generar una secuencia de series de Poisson convergentes, cuyos círculos de convergencia eviten z^* (Fig. 4.3b). Supongamos que se desea una aproximación convergente en un intervalo $[0, a]$. Cuando $r < a$ no se podrá emplear una serie de Poisson centrada en $t = 0$ más allá de $t = r$. Mediante la subdivisión del lado derecho se puede calcular una nueva serie centrada en $t_1 < r$ convergente más allá de $t = r$. Esta estrategia se repetirá tantas veces como sea necesario hasta abarcar todo el intervalo deseado.

Se debe resaltar una ventaja de las series de Poisson frente a las series de Taylor para el caso de funciones enteras (es decir, con $r = \infty$) y tal que $\lim_{t \rightarrow \infty} f(t) = 0$. En este caso las series de Poisson convergen uniformemente sobre el intervalo $[0, \infty)$, mientras que las series de Taylor convergen uniformemente sólo en intervalos finitos.

4.3 Series truncadas y polígonos de Poisson

Una primera limitación en las aplicaciones prácticas es que no se pueden guardar series infinitas. Por lo tanto, se debe truncar la expansión (4.3) en un cierto término k -ésimo, lo que genera la aproximación:

$$P_k(f; t) = \sum_{i=0}^k q_i b_i(\rho t). \quad (4.4)$$

Esta aproximación varía para diferentes valores de ρ . Como en el caso de las series de Taylor, la aproximación $P_k(f; t)$ tiene contacto de orden k con $f(t)$ en $t = 0$. Se debe tener en cuenta que una serie de Poisson truncada no es polinómica, por lo que no puede ser incorporada en un programa de CAD/CAM ya que no es expresable en el estándar Bernstein-Bézier. De hecho, una serie truncada no puede representar exactamente ningún polinomio, excepto la función nula.

Para eludir este inconveniente, Morin y Goldman (2000a), en lugar de una serie truncada (4.4), emplean una aproximación basada en un polígono de control de Poisson truncado de puntos \mathbf{q}_i , con $0 \leq i \leq k$. Para aproximar $f(t)$ sobre un intervalo $t \in [0, a]$ con un polígono de $k + 1$ puntos, simplemente se escoge un factor de escala $\rho = a/k$. No obstante, este polígono truncado presenta nuevos problemas:

- No cumple las propiedades de variación disminuida y convex hull. La función $f(t) = t^2$ (Fig. 4.2) proporciona un claro ejemplo.
- Proporciona una aproximación lineal a trozos, por tanto únicamente con continuidad C^0 .
- La aproximación no interpola $f(t)$ en el punto final, por lo que la extensión analítica descrita en el apartado 4.2 produce huecos en la unión de varias expansiones.

4.4 Convergencia de las s-series frente a las series de Poisson

En la Fig 4.4 se han representado conjuntamente las áreas limitadoras de convergencia, un círculo C para el caso de las series de Poisson y una lemniscata de Bernoulli L para las s-series. Estas áreas determinan la convergencia de ambas series sobre el intervalo real $u \in [0, 1]$. Si el área no contiene ninguna singularidad, entonces la correspondiente serie es convergente.

En el caso en que la singularidad está situada en la región $C - L$, la serie de Poisson será divergente, mientras que la s-serie es convergente. Únicamente si la singularidad está situada en la pequeña región $L - C$ la serie de Poisson será convergente y la s-serie divergente. Se puede concluir que, en general, las s-series disfrutan de mejor convergencia que las series de Poisson.

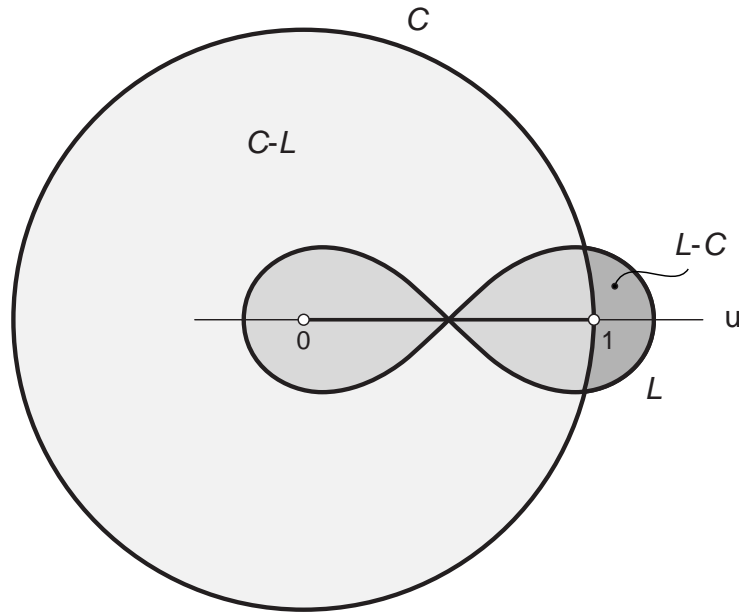


Figura 4.4: Comparación entre el área limitadora de convergencia de las series de Poisson y las s-series.

4.5 Ejemplos comparativos

En este apartado se han tomado distintos ejemplos de Morin y Goldman (2000a, 2000b, 2001a) y se han comparado las aproximaciones de Hermite con las aproximaciones de Poisson sobre un intervalo dado $t \in [0, a]$. El parámetro elegido para poder comparar ambos métodos ha sido el número de términos de ambas aproximaciones, el grado $n = 2k + 1$ en el caso de las s-series (2.1), y el índice k en el caso de Poisson (4.4).

Función logarítmica $f(t) = \log(2 - 2t + t^2)$

Como primer ejemplo se ha aproximado la función $f(t) = \log(2 - 2t + t^2)$ en $t \in [0, 3]$. Esta función presenta dos singularidades complejas $z^* = 1 \pm i$. Como muestra la Fig. 4.5a ambas singularidades están situadas fuera de la lemniscata L , por lo que las sucesivas aproximaciones mediante s-series (grado $n = 3, 5, 7$) son convergentes (Fig. 4.5b). Las aproximaciones de Poisson son divergentes para valores mayores a $t = r$, ya que el radio de convergencia de la serie de Poisson es $r = \sqrt{2}$ (Fig. 4.5c). En la figura 4.5d se ha representado el error de ambos métodos de aproximación, demostrando de esta forma la mayor precisión de las s-series ($n = 5, 7$) frente a las aproximaciones de Poisson ($k = 45$), ya que con s-series se consiguen errores muy pequeños para grados muy bajos.

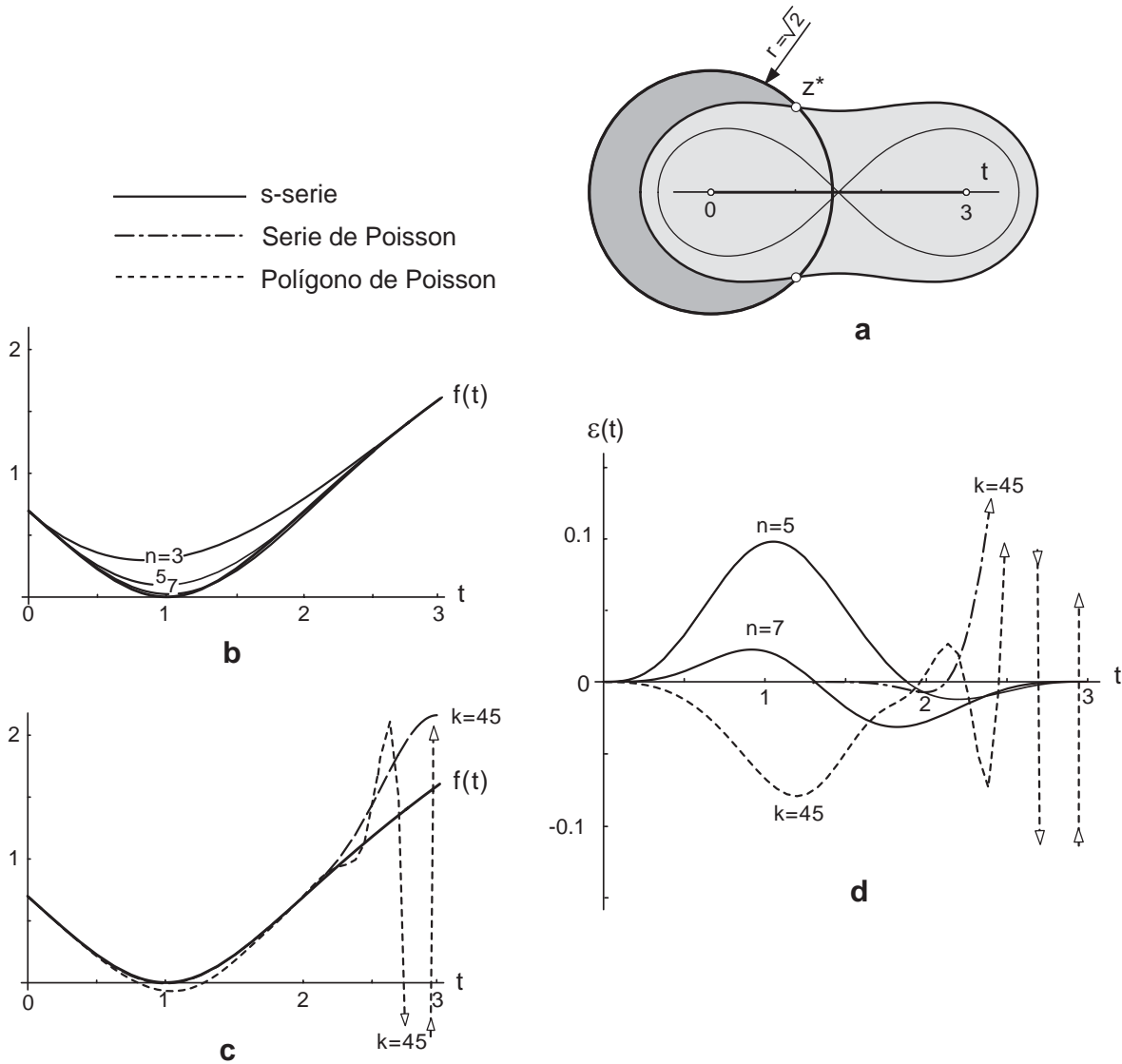


Figura 4.5: $f(t) = \log(2 - 2t + t^2)$. (a) Áreas de convergencia para las series de Poisson y las s -series. (b) Aproximación mediante s -serie (grado $n = 3, 5, 7$). (c) Polígono truncado y serie de Poisson ($k = 45$). (d) Error $\varepsilon(t)$.

Función racional $f(t) = \frac{1}{1+t}$

Otro ejemplo es el proporcionado por la función racional $f(t) = (1+t)^{-1}$ en $t \in [0, 2]$. Esta función tiene una singularidad real $t^* = -1$. Esta singularidad está fuera de la lemniscata L (Fig. 4.6a), por lo que la aproximación de Hermite es convergente, como ilustra la Fig. 4.6b. Las aproximaciones de Poisson son divergentes para valores de $t > 1$ (Fig. 4.6c), ya que el radio de convergencia de la serie de Poisson es $r = 1$. La Fig. 4.6d muestra el error $\varepsilon(t)$ de ambos métodos de aproximación, donde queda reflejada la divergencia de las aproximaciones de Poisson. Obsérvese que para conseguir una precisión similar con ambos métodos, los polígonos de Poisson necesitan muchos más términos que las s -series.

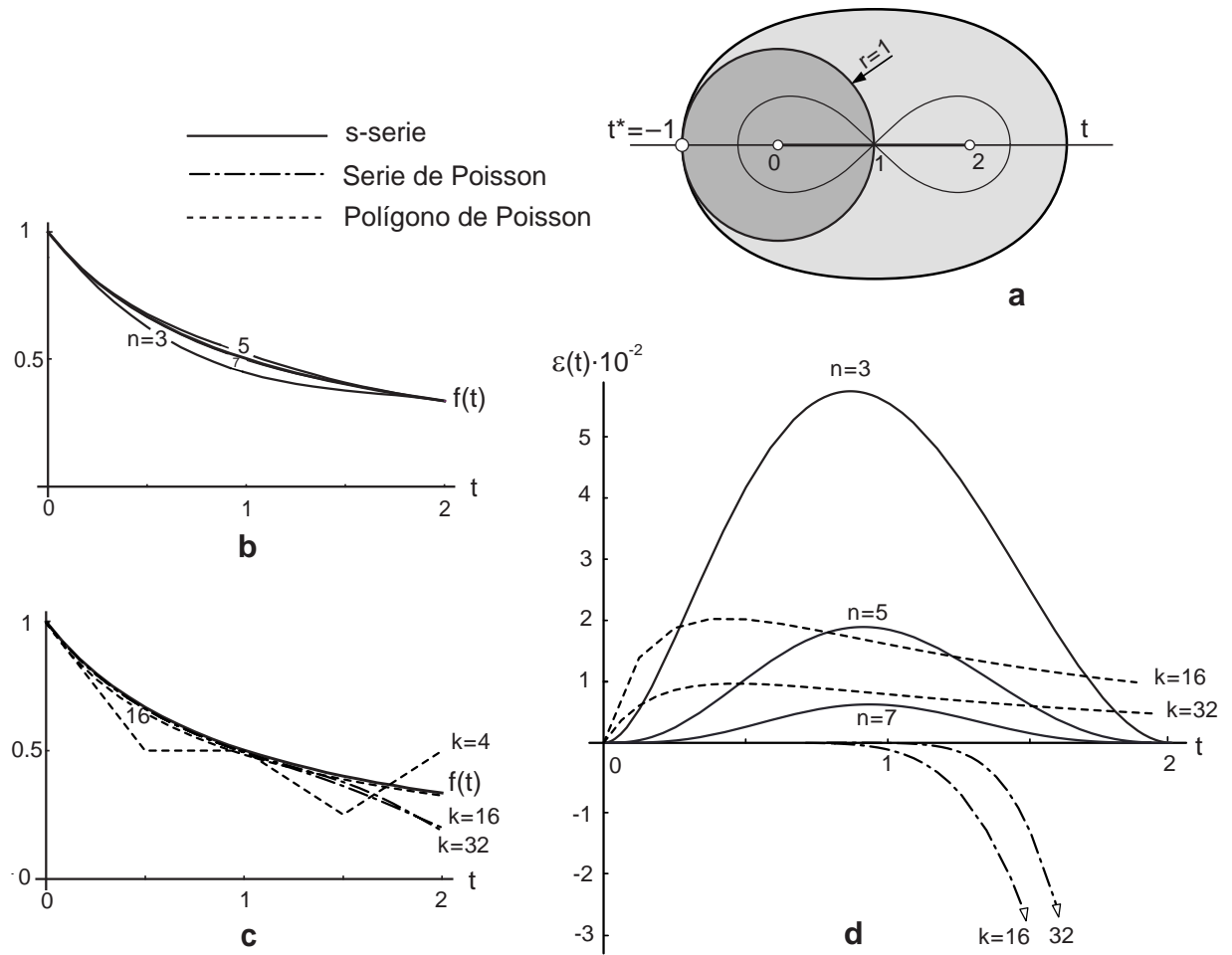


Figura 4.6: $f(t) = (1+t)^{-1}$. (a) Singularidad t^* de $f(t)$ y áreas de convergencia. (b) s-serie ($n = 3, 5, 7$). (c) Series de Poisson truncada ($k = 16, 32$) y polígono de Poisson ($k = 4, 16, 32$). (d) Error $\varepsilon(t)$.

Espiral de Arquímedes

La espiral de Arquímedes $\mathbf{c}(t)$ está definida paramétricamente como $\mathbf{c}(t) = t \cdot (\cos t, \sin t)$. Se han calculado distintas aproximaciones de esta curva sobre $t \in [0, 4\pi]$. Las funciones componentes no presentan ninguna singularidad por lo que la convergencia está garantizada tanto para las s-series como para las aproximaciones de Poisson. La Fig. 4.7a muestra la s-serie ($n = 7, 9$) junto al polígono de control de Poisson ($k = 400, 800$) y la serie de Poisson ($k = 400, 800$). La calidad de estas aproximaciones se ha medido usando el error Pitagórico $\varepsilon(t)$, es decir la distancia para un valor t entre la curva exacta $\mathbf{c}(t)$ y la aproximación. En el caso de la s-serie se ha subdividido el dominio inicial por el punto medio $t = 2\pi$ con el objetivo de disminuir el error máximo. Como muestra la Fig. 4.7b, el error $\varepsilon(t)$ es nulo en los puntos extremos y tiene forma de campana. Nótese que, para conseguir un error máximo aceptable, en el caso de la s-serie basta con un grado $n = 9$, mientras que, tanto para los polígonos de Poisson como la serie de Poisson, las aproximaciones ($k = 800$) son distinguibles visualmente de la curva exacta.

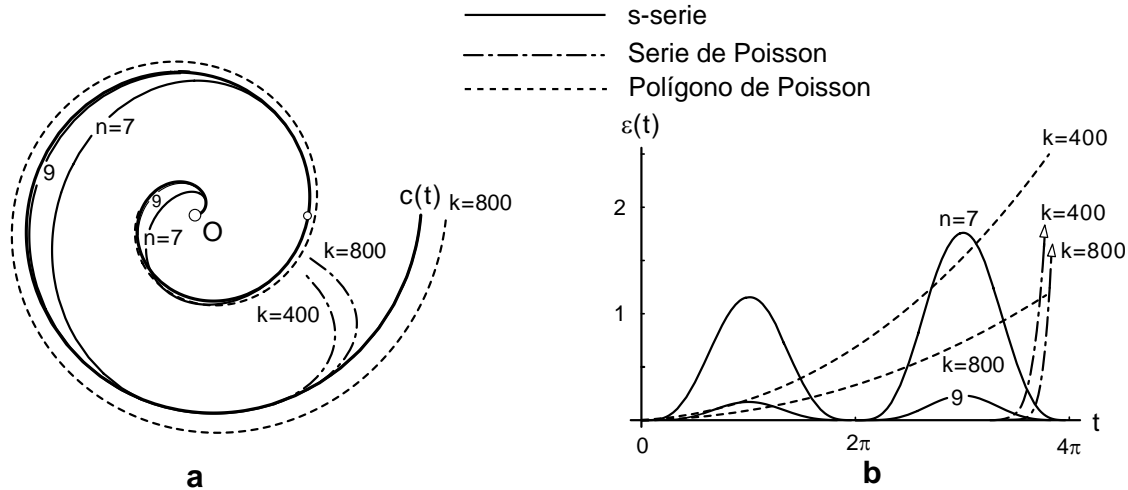


Figura 4.7: Espiral de Arquímedes. (a) Aproximaciones: *s*-serie (grado $n = 7, 9$) y Poisson ($k = 400, 800$). (b) Error $\varepsilon(t)$.

Semicírculo unitario

En este ejemplo se ha considerado una parametrización trigonométrica del semicírculo $c(t) = (\cos t, \sin t)$, $t \in [0, \pi]$. Como en el caso de la espiral, las funciones componentes no presentan singularidades, por lo que la convergencia está garantizada en ambas aproximaciones. La Figura 4.8a muestra que las aproximaciones mediante *s*-series ($n = 3, 5$) presentan un resultado de mejor calidad que las aproximaciones de Poisson ($k = 25, 50$).

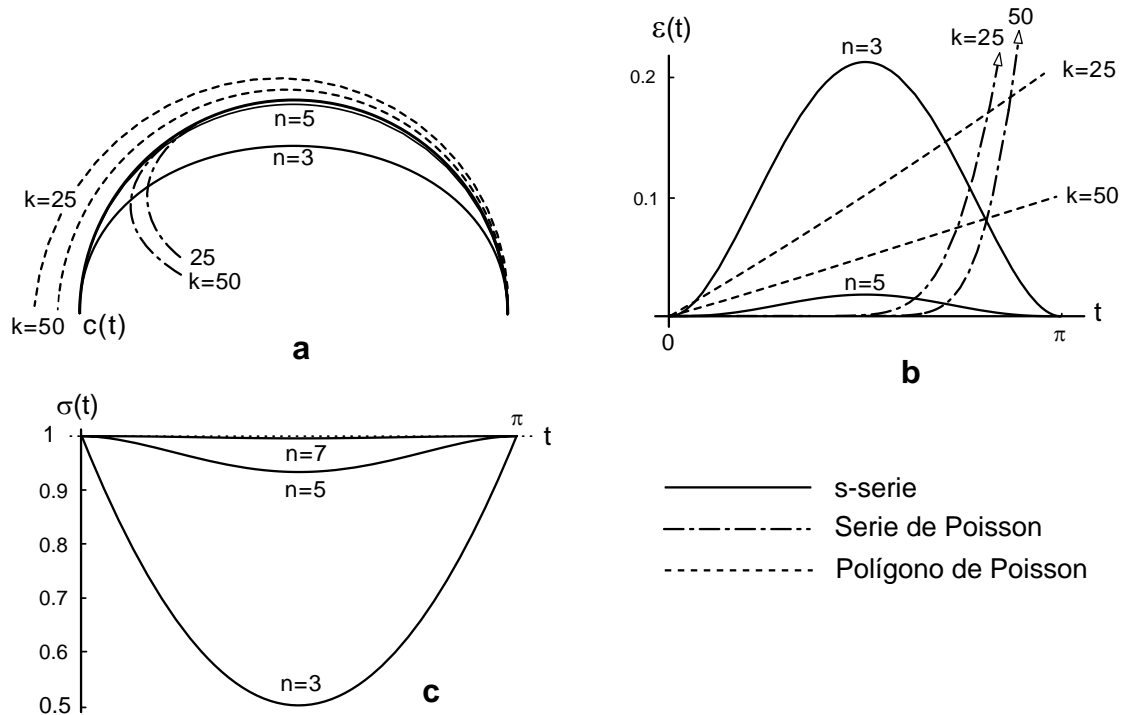


Figura 4.8: Semicírculo. (a) Aproximaciones: *s*-serie (grado $n = 3, 5$) y Poisson ($k = 25, 50$). (b) Error $\varepsilon(t)$. (c) Velocidad paramétrica $\sigma(t)$ de la aproximación vía *s*-series.

La Fig. 4.8b muestra el error de ambas aproximaciones, reflejando la superioridad de las s-series frente a las aproximaciones de Poisson. Este ejemplo ilustra otra ventaja de la aproximación de Hermite (Sánchez-Reyes y Chacón, 2003a), la tendencia a mantener la parametrización original de la longitud del arco, como muestra la figura 4.8c que representa la velocidad paramétrica $\sigma(t) = |dc(t)/dt|$.

Superficie bivariada $f(x, y) = \sin x \sin y$

En este ejemplo se ha aproximado la función bivariada $f(x, y) = \sin x \sin y$, sobre el dominio $(x, y) \in [0, 2\pi] \times [0, 2\pi]$. Tanto las aproximaciones mediante s-series como las de Poisson son convergentes, ya que las funciones componentes no presentan ninguna singularidad, si bien la s-serie posee mejor precisión que la serie de Poisson, como muestran las gráficas del error (Fig. 4.9).

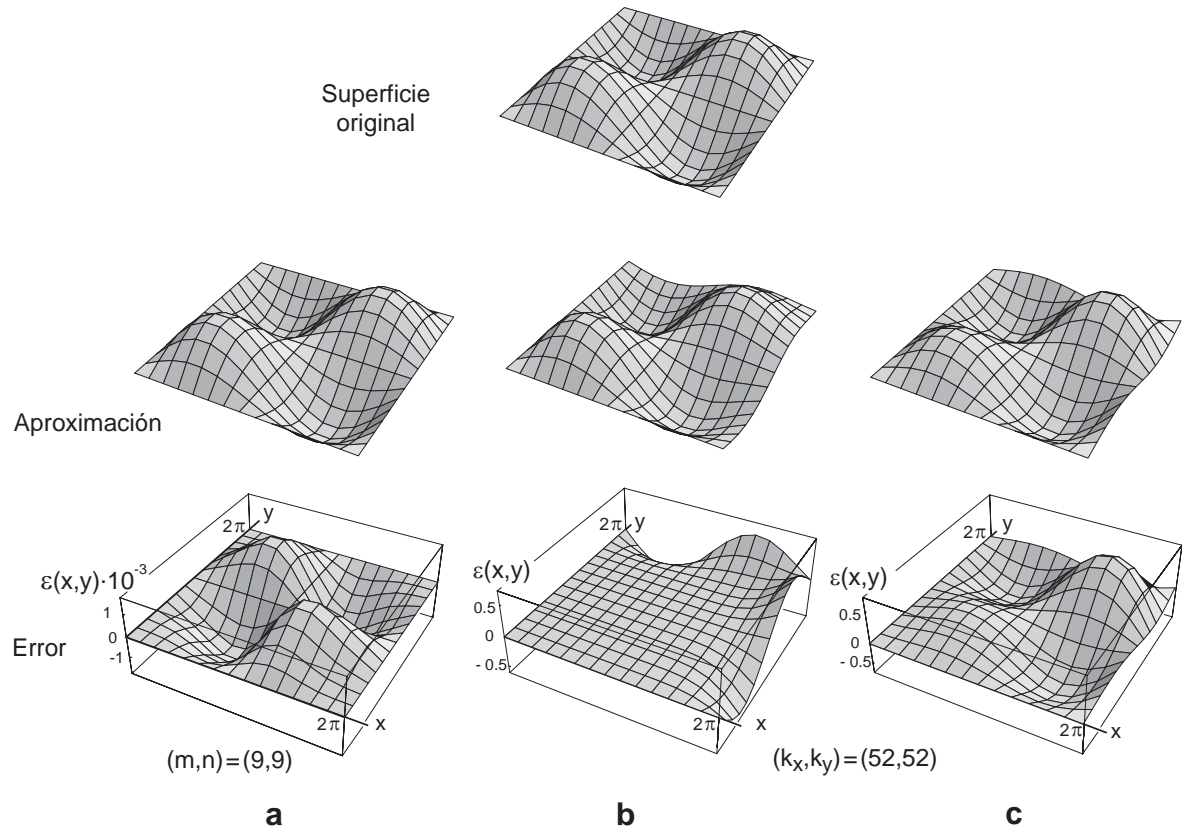


Figura 4.9: $f(x, y) = \sin x \sin y$. Aproximaciones y sus correspondientes errores $\varepsilon(x, y)$. (a) s-serie $(m, n) = (9, 9)$. (b) Serie de Poisson $(k_x, k_y) = (52, 52)$. (c) Polígono de control de Poisson $(k_x, k_y) = (52, 52)$.

Catenoide

Finalmente se ha calculado la aproximación del semi-catenode $f(z, \theta) = [r(z) \cdot \cos \theta, r(z) \cdot \sin \theta, z]$, superficie de revolución generada por el giro de una catenaria $r(z) = \cosh z$ alrededor del eje vertical z , con $(z, \theta) \in [-2, 2] \times [0, \pi]$. Al igual que en el ejemplo anterior, las aproximaciones mediante s-series y series de Poisson son convergentes ya que las funciones componentes no presentan ninguna singularidad. En la Fig. 4.10 se han representado las distintas aproximaciones con sus correspondientes errores. Una vez más se puede observar que las aproximaciones de Poisson presentan menor precisión que la aproximaciones con s-series.

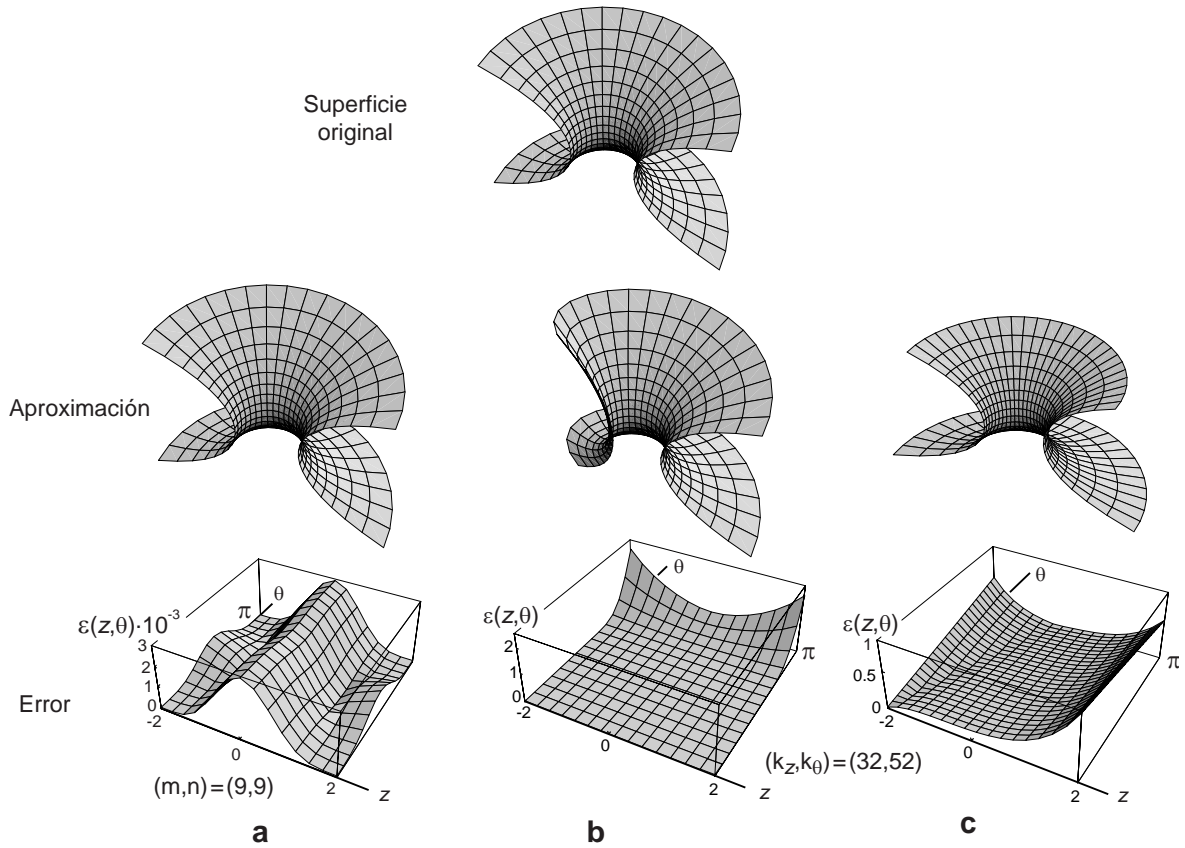


Figura 4.10: Semi-catenode. Aproximaciones y sus correspondientes errores $\varepsilon(z, \theta)$. (a) s-serie $(m, n) = (9, 9)$. (b) Serie de Poisson $(k_z, k_\theta) = (32, 52)$. (c) Polígono de control de Poisson $(k_z, k_\theta) = (32, 52)$.

Capítulo 5

Aproximación polinómica de espirales clotoides

En este capítulo, como ejemplo de aplicación de las técnicas expuestas en los capítulos 2 y 3, se propone un método para generar una aproximación polinómica, basada en el empleo de s-series, de un segmento arbitrario de clotoide sobre un intervalo finito. Al truncar una s-serie (Capítulo 2) en el término k -ésimo se genera el interpolador de Hermite de orden k , es decir, el polinomio de grado $2k + 1$ que reproduce en los extremos hasta la derivada k -ésima de la curva original. Empalmando estas aproximaciones se obtiene un spline Hermítico que exhibe continuidad C^k en las uniones y que tiende a mantener la parametrización original de la longitud arco. Esta aproximación resulta mucho más adecuada para aplicaciones CAD que un desarrollo de Taylor truncado o la combinación de expansiones de Taylor presentado por Wang et al. (2001).

5.1 Introducción

La espiral clotoide

La clotoide (Gomes, 1909), cuya forma se asemeja al muelle de un reloj mecánico, también conocida como *espiral de Cornu* o *espiral de Euler* (Lawrence, 1972), es la curva plana cuya curvatura varía linealmente con la longitud del arco. Esta propiedad hace que los segmentos de clotoide se usen en el trazado de carreteras y vías de ferrocarril o trayectorias de vehículos robotizados, como curva de transición entre rectas y/o arcos de círculo (Meek y Thomas, 1991; Walton y Meek, 1990; Kanayama y Miyake, 1985; Scheuer y Fraichard, 1997). Las clotoides también constituyen los tramos componentes de

las curvas de interpolación denominadas *splines geométricos* o *intrínsecos* (Hoschek y Lasser, 1993).

La clotoide $\mathbf{c}(t)$ se define paramétricamente en términos de las integrales de Fresnel, $C(t)$, $S(t)$, como:

$$\mathbf{c}(t) = \begin{pmatrix} C(t) \\ S(t) \end{pmatrix}, \quad \begin{cases} C(t) = \int_0^t \cos\left(\frac{\pi}{2} \xi^2\right) d\xi \\ S(t) = \int_0^t \sin\left(\frac{\pi}{2} \xi^2\right) d\xi \end{cases}. \quad (5.1)$$

Escogiendo un sistema de coordenadas adecuado, para $t > 0$ la clotoide está definida en el primer cuadrante, parte del origen \mathbf{O} para $t = 0$ y tiende al punto $\mathbf{c}(\infty) = (1/2, 1/2)$ para $t \rightarrow \infty$ (Fig. 5.1).

La clotoide (5.1) presenta las siguientes propiedades remarcables:

- Longitud de arco $l(t) = t$, equivalente a parametrización con velocidad unitaria:

$$\sigma(t) = \left\| \frac{d}{dt} \mathbf{c}(t) \right\| = \frac{d}{dt} l(t) = 1. \quad (5.2)$$

- Curvatura: $k(t) = \pi t$.
- Angulo $\alpha(t)$ entre la tangente y el eje x : $\alpha(t) = (\pi/2)t^2$.

Desafortunadamente la clotoide y su offset, al estar definidas en términos de las integrales de Fresnel (5.1), son curvas trascendentes. Por tanto, no admiten una parametrización racional y no pueden expresarse en el estándar NURBS.

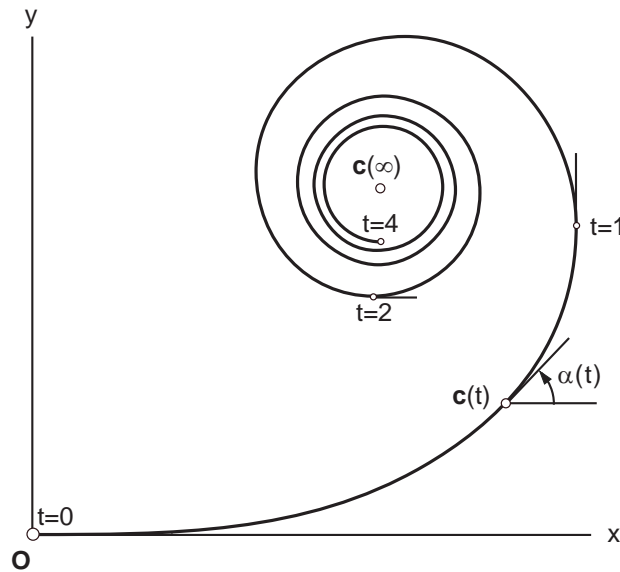


Figura 5.1: Espiral clotoide $\mathbf{c}(t)$, $t \in [0, 4]$.

Aproximaciones polinómicas de la clotoide

Todas las aproximaciones que involucran funciones no polinómicas, como raíces cuadradas (Boersman, 1960) o funciones trigonométricas (Heald, 1985; Cody, 1968) no son admisibles, ya que no pueden ser expresadas en el estándar NURBS. En cuanto a aproximaciones polinómicas, se podría pensar en series de Taylor truncadas, solución propuesta por Press et al. (1992) para valores pequeños de t , y Wang et al. (2001). Dado que las integrales de Fresnel no presentan ninguna singularidad, queda garantizada la convergencia del desarrollo. Con este método se obtendría una buena aproximación de la curva alrededor de un punto, pero se precisaría un grado muy elevado para conseguir errores pequeños con una única aproximación, y los programas de CAD tienen limitado el grado máximo admisible. Además el resultado en la base monomial debe convertirse a la base de Bernstein, conversión mal condicionada para grados elevados como demostraron Daniel y Daubisse (1989) y Farouki (1991). Esta deficiencia, no mencionada por Wang et al. (2001), puede implicar una importante pérdida de precisión en la transformación.

La división del dominio de interés en varios trozos con el objeto de evitar grados elevados (Capítulo 2), calculando el desarrollo de Taylor de grado reducido de cada trozo, no es una solución válida, ya que aparecen huecos cuando intentamos unir dos expansiones de Taylor (Fig. 5.2a). Para aplicaciones en vehículos robotizados, estas discontinuidades tendrían efectos catastróficos ya que producirían velocidades y aceleraciones infinitas.

Para solucionar este problema Wang et al. (2001) presentan un método alternativo. La aproximación sobre cada intervalo (t_i, t_{i+1}) resulta de combinar los desarrollos de Taylor (grado k) $a^k(u)$ y $a^k(1-u)$ (2.6) en dos nodos consecutivos con dos polinomios de Hermite $H^{2k+1}(u)$ de grado $2k+1$:

$$a^k(u) \cdot H^{2k+1}(u) + a^k(1-u) \cdot H^{2k+1}(1-u).$$

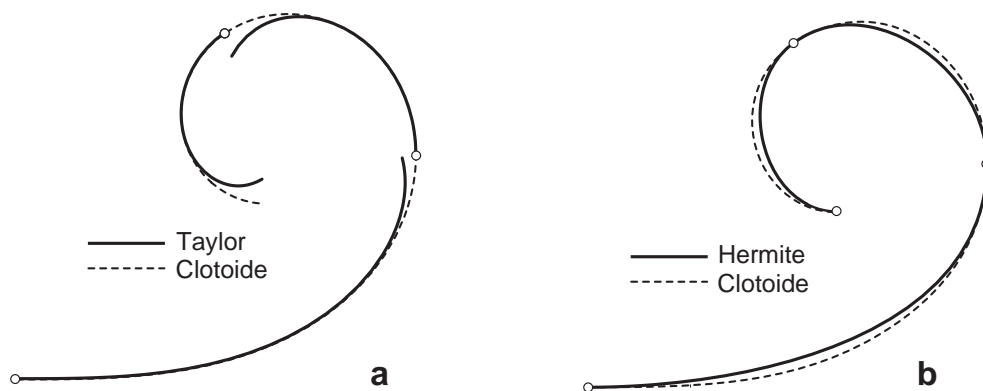


Figura 5.2: Aproximaciones polinómicas de la clotoide. (a) Taylor. (b) Hermite.

Finalmente, se unen las aproximaciones sobre cada intervalo, generando una representación B-spline. Estos polinomios de Hermite presentan las siguientes propiedades:

$$H^{2k+1}(0) = 1, \quad \left\{ \frac{d^i}{du^i} (H^{2k+1}(0)) = 0 \right\}_{i=1}^k, \quad H^{2k+1}(1) = \left\{ \frac{d^i}{du^i} (H^{2k+1}(1)) = 0 \right\}_{i=1}^k = 0.$$

Este método, aparte de emplear una formulación muy compleja para calcular las expansiones de Taylor, necesita grados elevados. Para alcanzar continuidad C^k se emplea un grado $n = 3k + 1$.

Splines Hermíticos y s-series

Frente a los desarrollos de Taylor propuestos, la solución obvia a la cuestión es el empleo de s-series (2.1), expansiones de Hermite en 2 puntos. La convergencia de las s-series (Capítulo 3) queda garantizada cuando $f(z)$ no presenta singularidades en el plano complejo, condición satisfecha por las integrales de Fresnel (5.1).

La aproximación de Hermite de orden k resulta muy conveniente para una representación a trozos (Capítulo 2). Esta aproximación genera un spline Hermítico de grado $n = 2k + 1$ que exhibe continuidad C^k en las uniones (Fig. 5.2b), menor que el utilizado ($n = 3k + 1$) por Wang et al. (2001). En particular, para obtener una continuidad C^2 , únicamente necesitamos $n = 5$. Con el método propuesto por Wang et al. (2001) bastaría $n = 7$, aunque con el fin de reducir el error utilizan $n = 8$.

Otra ventaja de la interpolación de Hermite es que tiende a preservar la parametrización arco original (5.2), ya que la aproximación de Hermite $H_k(\mathbf{c}; t)$ de orden k de la clotoide tiene una velocidad paramétrica:

$$\sigma_k(t) = \left\| \frac{d}{dt} H_k(\mathbf{c}, t) \right\|, \quad (5.3)$$

cuyas derivadas $0, 1, \dots, k-1$, coinciden en los puntos extremos con las de la clotoide original (5.2):

$$\begin{aligned} \sigma_k(t_0) &= \sigma_k(t_1) = 1 \\ \sigma_k^{(r)}(t_0) &= \sigma_k^{(r)}(t_1) = 0, \quad r = 1, \dots, k-1 \end{aligned}$$

lo cual resulta ser beneficioso en aplicaciones tales como el control de movimientos de vehículos robotizados. La convergencia de $H_k(\mathbf{c}; t)$ a $\mathbf{c}(t)$, cuando $k \rightarrow \infty$, conlleva la convergencia de las derivadas $d/dt(H_k(\mathbf{c}; t))$ a $d/dt(\mathbf{c}(t))$ ya que ambas funciones, $H_k(\mathbf{c}; t)$ y $\mathbf{c}(t)$, son analíticas. Esto garantiza la convergencia de la velocidad paramétrica $\sigma_k(t)$ (5.3) a la función unidad. La interpolación *geométrica de Hermite*, desarrollada por de Boor et al. (1987), no es recomendable para aproximar la clotoide, incluso cuando pudiéramos

conseguir una alta continuidad geométrica para un grado dado. La ventaja de esta técnica es la libertad en la elección de la parametrización, pero esto podría destruir la tendencia a mantener la parametrización de la longitud arco.

5.2 Cálculo de la s-serie para un segmento de clotoide

Para calcular la aproximación de Hermite $H_k(\mathbf{c}; t)$ de orden k de la clotoide $\mathbf{c}(t)$ sobre un intervalo dado $t \in [t_0, t_1]$, basta con aproximar sus dos componentes, esto es, las integrales de Fresnel (5.1):

$$H_k(\mathbf{c}; t) = \begin{pmatrix} H_k(C; t) \\ H_k(S; t) \end{pmatrix}. \quad (5.4)$$

A continuación se describe, paso a paso, cómo aproximar $S(t) = \int_0^t \sin\left(\frac{\pi}{2} \xi^2\right) d\xi$ sobre un intervalo arbitrario $t \in [t_0, t_1]$ mediante el método propuesto en el apartado 2.4, aunque también se podría haber utilizado la expresión (2.7) en función de las derivadas. La s-serie para $C(t)$ (5.4) se calcularía de forma análoga.

1. Rescribir ξ como función $\xi(u)$ sobre un dominio unitario $u \in [0, 1]$.

En principio las s-series (2.1) deben estar definidas sobre un dominio unitario $u \in [0, 1]$ de la variable. Para conseguirlo, expresaremos la variable de integración ξ como una función de u mediante la siguiente transformación lineal:

$$\xi(u) = t_0(1-u) + t_1(u) \rightarrow \xi = \begin{Bmatrix} t_0 \\ t_1 \end{Bmatrix}, \quad \begin{matrix} u \in [0, 1] \\ \xi(u) \in [t_0, t_1] \end{matrix}.$$

2. Expresión de la función $\alpha(u) = \frac{\pi}{2} \xi^2(u)$ como una s-serie.

El polinomio cuadrático $\alpha(u)$ se define como producto de dos funciones lineales. En consecuencia, adoptando la notación (2.5) basta aplicar la regla del producto de dos s-series (2.8):

$$\alpha(u) \rightarrow \alpha = \begin{Bmatrix} \alpha_0^0 \\ \alpha_1^0 \end{Bmatrix} + \begin{Bmatrix} \alpha_0^1 \\ \alpha_1^1 \end{Bmatrix} s = \frac{\pi}{2} \begin{Bmatrix} (t_0)^2 \\ (t_1)^2 \end{Bmatrix} - \frac{\pi}{2} (\Delta\xi)^2 \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} s, \quad \Delta\xi = t_1 - t_0.$$

3. Composición $\sin(\alpha(u))$

Primero, rescribimos $\alpha(u)$ en la forma (2.11):

$$\alpha(u) = \alpha_0^0(1-v) + \alpha_1^0 v, \quad v(u) = \frac{\alpha(u) - \alpha_0^0}{\alpha_1^0 - \alpha_0^0}, \quad v \in [0, 1], \quad (5.5)$$

en términos de una función $v(u)$ sobre un dominio unitario. Ahora podemos reemplazar $v(u)$ (5.5) en la serie de coeficientes (2.12) de la función sinusoidal.

4. Integración

Finalmente se calcula la s-serie de $S(t(u))$, teniendo en cuenta que en dicha expresión se emplea una variable u unitaria, en vez de $t \in [t_0, t_1]$ (2.2). Para ello se integra la derivada:

$$\frac{d}{du} S(t(u)) = d(u) = (\Delta \xi) \sin(\alpha(u)), \quad \Delta \xi = t_1 - t_0,$$

empleando la fórmula recurrente (2.9), donde el coeficiente inicial a_0 (2.10) es conocido:

$$a_0 = \begin{Bmatrix} a(0) \\ a(1) \end{Bmatrix} = \begin{Bmatrix} S(t_0) \\ S(t_1) \end{Bmatrix}.$$

Hasta ahora se han supuesto series infinitas. En la práctica interesará obtener aproximaciones, que no son más que series truncadas (2.3). Por tanto, para calcular las aproximaciones de Hermite $H_k(S; t)$ y $H_k(C; t)$ de orden k de las integrales de Fresnel, en el paso 3 basta trabajar con la aproximación de orden k de la función sinusoidal, y análogamente al integrar en el paso 4.

5.3 Ejemplos

En esta sección se han obtenido distintas aproximaciones de la clotoide, con y sin subdivisión en el intervalo $t \in [0, 1]$, correspondiente a un ángulo $\alpha \in [0, \pi/2]$. Este intervalo, escogido por Wang et al. (2001), corresponde al máximo rango empleado en el diseño de carreteras.

Aproximación sin subdivisión

Siguiendo el procedimiento descrito en la sección 5.2, en la Fig. 5.3 se han calculado las sucesivas aproximaciones de orden k de las integrales de Fresnel definidas en el intervalo $t \in [0, 1]$. Obsérvese que para un grado $n = 5$ ($k = 2$) las aproximaciones de Hermite $H_2(S; t)$ y $H_2(C; t)$ de las integrales de Fresnel, representadas con línea fina, son indistinguibles visualmente de las integrales exactas, $S(t)$ y $C(t)$, representadas con línea gruesa.

Partiendo de las aproximaciones de Hermite de las integrales de Fresnel, en la Fig. 5.4a se ha representado la aproximación de Hermite $H_k(\mathbf{c}; t)$ de la clotoide. Obsérvese que para grado $n = 7$ ($k = 3$) la aproximación de Hermite coincide prácticamente con la clotoide. La aproximación $H_k(\mathbf{c}; t)$ tiene contacto de orden k con la clotoide $\mathbf{c}(t)$ en los extremos, por lo

que la unión de varios tramos también tendrá continuidad C^k , continuidad que no se habría conseguido mediante expansiones de Taylor.

Empleando la técnica descrita en el apartado 3.3 se han calculado los puntos de control de la representación de Bézier de la aproximación de Hermite de la clotoide (Fig. 5.4b).

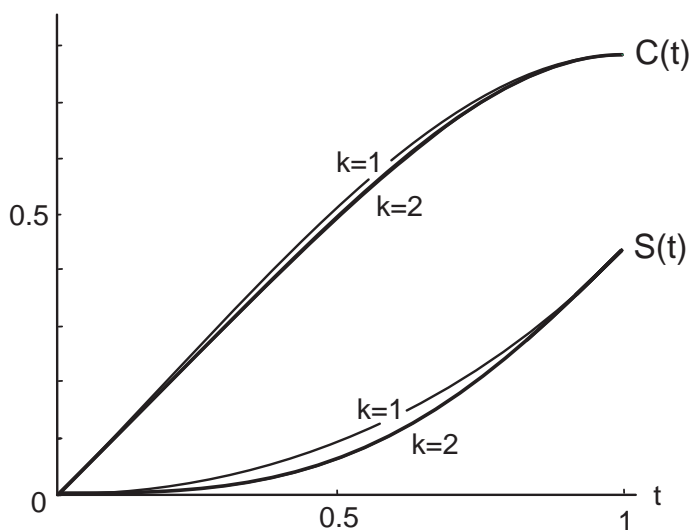


Figura 5.3: Aproximaciones de Hermite $H_k(S; t)$ y $H_k(C; t)$ de orden $k = 1, 2$ de las integrales de Fresnel.

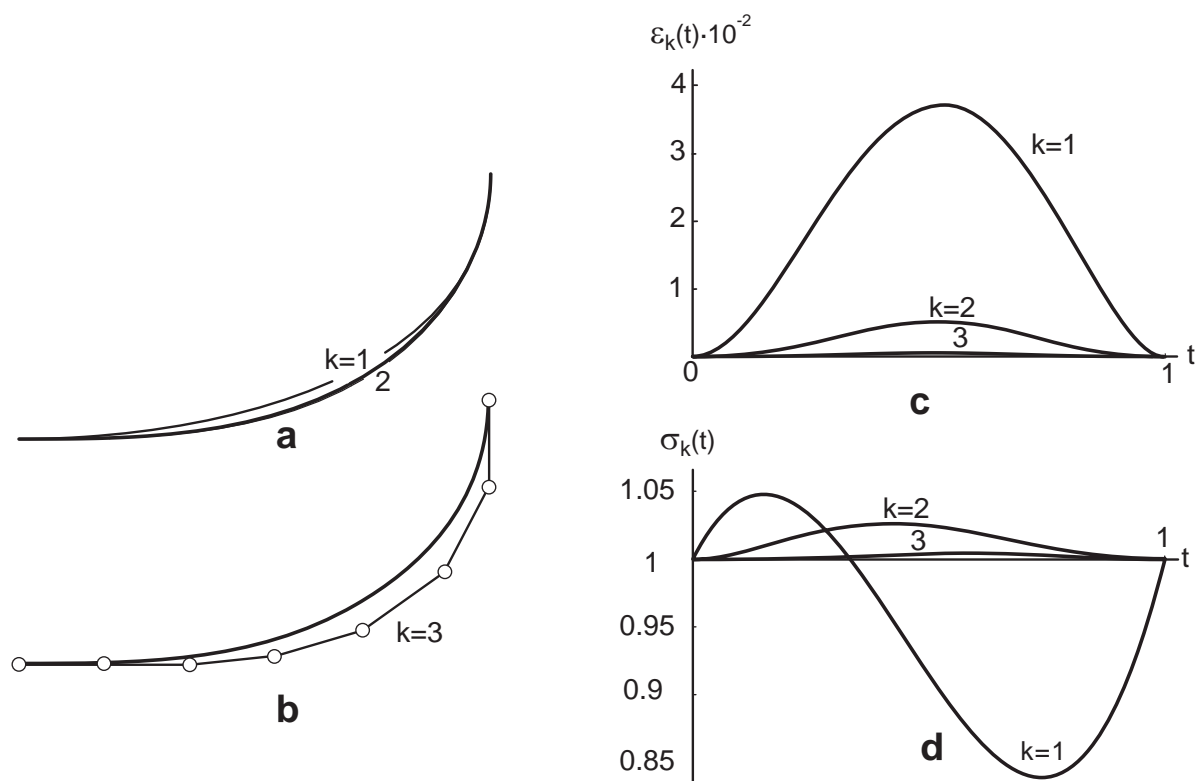


Figura 5.4: (a) Aproximaciones de Hermite $H_k(c; t)$ de orden $k = 1, 2, 3$ de la clotoide para $t \in [0, 1]$.

(b) Representación de Bézier. (c) Error Pitagórico $\varepsilon_k(t)$. (d) Velocidad paramétrica $\sigma_k(t)$.

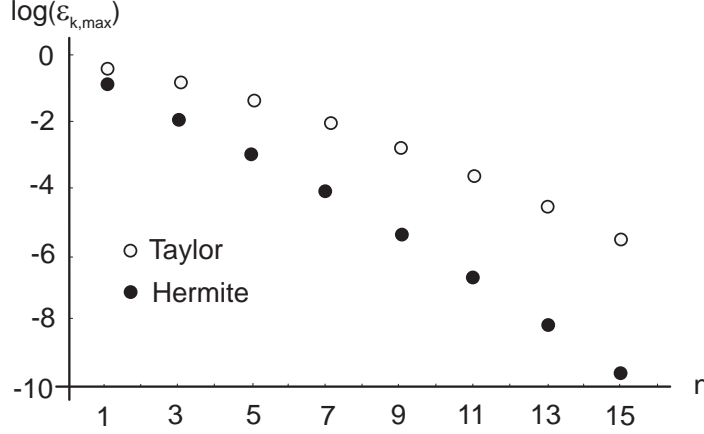


Figura 5.5: Error máximo $\varepsilon_{k,max}(t)$ de la aproximación de Taylor y Hermite de la clotoide en $t \in [0,1]$.

La calidad de las aproximaciones se ha evaluado mediante el error $\varepsilon_k(t)$ Pitagórico en el plano de Cornú:

$$\varepsilon_k(t) = \|\mathbf{c}(t) - H_k(\mathbf{c}; t)\|. \quad (5.6)$$

La Fig. 5.4c muestra cómo la gráfica del error $\varepsilon_k(t)$, con forma de campana, disminuye a medida que aumenta el grado de la serie, es mayor en la parte central y se anula en ambos extremos. La velocidad paramétrica $\sigma_k(t)$ de la aproximación se muestra en la Fig. 5.4d, donde se puede observar que $\sigma_k(t) \rightarrow 1$ a medida que aumenta el grado de la aproximación $H_k(\mathbf{c}; t)$. La superioridad de esta aproximación respecto a la expansión de Taylor queda reflejada en la Fig. 5.5, donde se compara el error máximo, en $t \in [0,1]$, para la interpolación de Taylor y la de Hermite (en función del grado $n = 2k + 1$).

Aproximación mediante spline Hermítico

Para conseguir aproximaciones de grado reducido con un cierto error, simplemente basta con subdividir la curva en distintos segmentos, calculando para cada segmento la correspondiente aproximación de Hermite de orden k . De esta forma se podría aproximar cualquier tramo de clotoide, sea cual sea su longitud, con grado reducido y un error tan pequeño como se desee, mediante el proceso repetido de subdivisión, estimando el error mediante muestreo discreto.

En la Fig. 5.6a se ha subdividido el dominio inicial $t \in [0,1]$ por el punto medio $t = 1/2$, calculando las aproximaciones de Hermite $H_k(\mathbf{c}; t)$ de grado $n = 3, 5$ ($k = 1, 2$) de la clotoide. A continuación se ha convertido esta aproximación a la representación de Bézier. El error $\varepsilon_k(t)$ (5.6) disminuye considerablemente después de la subdivisión (Fig. 5.6b), como se muestra en la tabla 5.1, que muestra la relación entre los errores máximos con y sin subdivisión. Se puede observar que la velocidad paramétrica de la aproximación $\sigma_k(t)$, representada en la Fig. 5.6c, presenta continuidad C^{k-1} en el nodo interno de subdivisión.

Finalmente se han concatenado los dos segmentos de Bézier de grado $n = 5$ ($k = 2$) en una curva B-spline (Fig. 5.7) mediante el algoritmo de eliminación de nodos (Capítulo 3). Este spline tiene un nodo interno de grado de multiplicidad $k + 1$, por lo que presenta continuidad C^k .

Tabla 5.1: Reducción del error por subdivisión.

Orden k	1	2	3
$\frac{\varepsilon_{k,max}(\text{subdivisión})}{\varepsilon_{k,max}(\text{sin subdivisión})}$	0.109	0.031	0.009

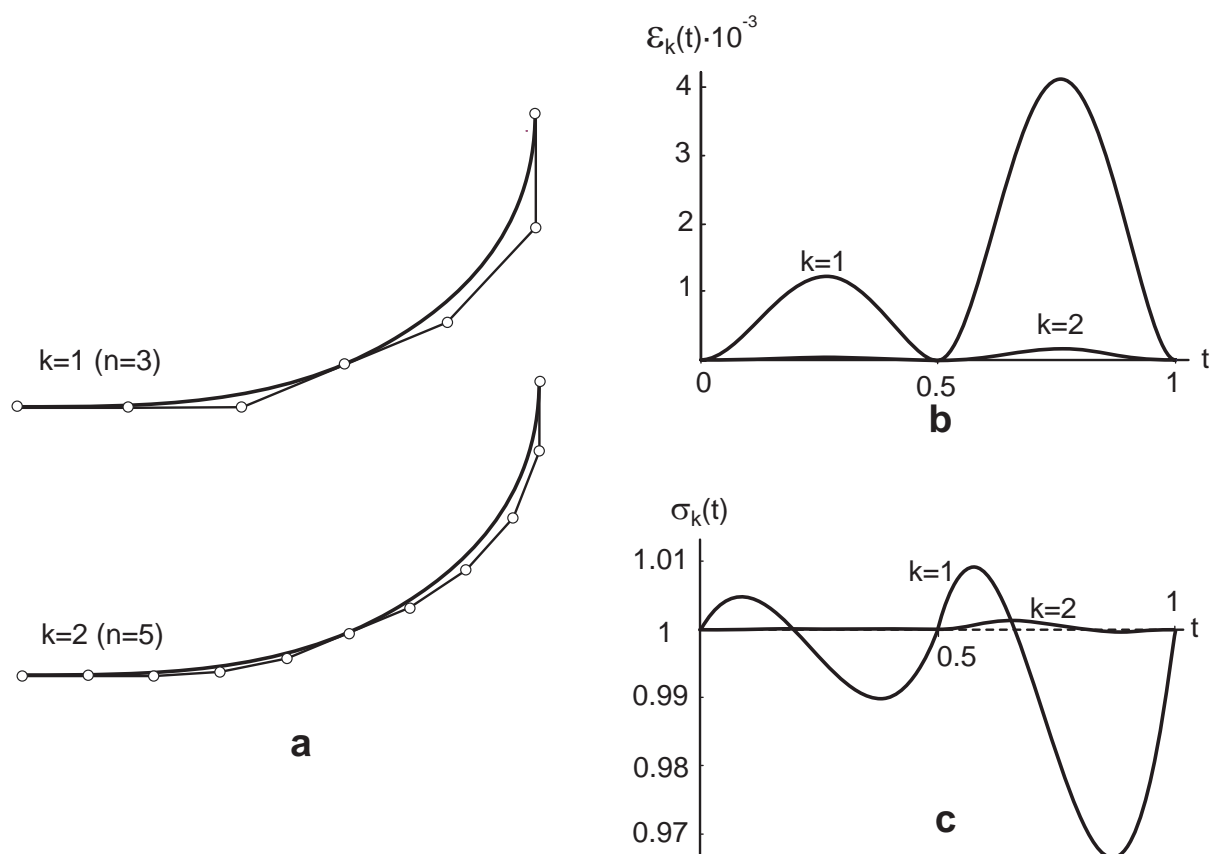


Figura 5.6: (a) Aproximaciones de Hermite $H_k(\mathbf{c}; t)$ de orden $k = 1, 2$ de la clotoide para $t \in [0, 1]$ con subdivisión en $t = 0.5$. (b) Representación de Bézier. (c) Error Pitagórico $\varepsilon_k(t)$.
(d) Velocidad paramétrica $\sigma_k(t)$.

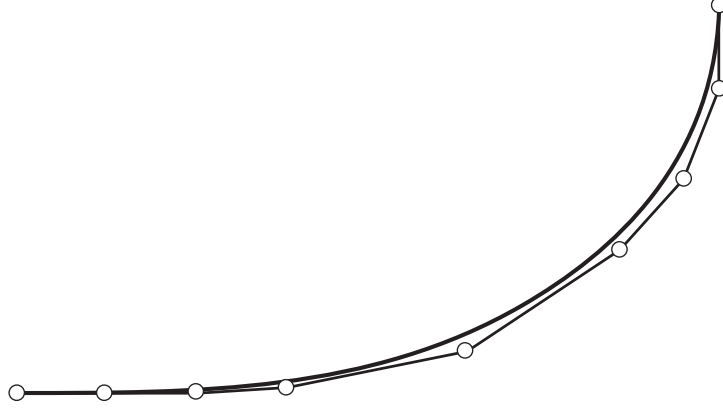


Figura 5.7: Curva B-spline y polígono de control de la aproximación de la clotoide de grado $n = 5$ ($k = 2$).

Aproximación del offset de la clotoide

El offset $\mathbf{c}_d(t)$ de la clotoide para un desplazamiento d viene dado por:

$$\mathbf{c}_d(t) = \mathbf{c}(t) \pm d\mathbf{n}(t), \quad \mathbf{n}(t) = \begin{pmatrix} -\sin\left(\frac{\pi}{2}t^2\right) \\ \cos\left(\frac{\pi}{2}t^2\right) \end{pmatrix}. \quad (5.7)$$

Para aproximar $\mathbf{c}_d(t)$ mediante s-series basta con calcular las aproximaciones de orden k de la clotoide $\mathbf{c}(t)$ y de las funciones sinusoidales que aparecen en la normal $\mathbf{n}(t)$ (5.7). La Fig. 5.8a muestra las aproximaciones de Hermite $H_k(\mathbf{c}_d; t)$ de grado $n = 3, 5$ ($k = 1, 2$) del offset de la clotoide tras subdividir el dominio inicial $t \in [0, 1]$ por el punto medio $t = 1/2$. Como ilustra la Fig. 5.8b, al igual que en la aproximación de la clotoide, el error $\varepsilon_k(t)$ (5.6) disminuye considerablemente después de la subdivisión. Obsérvese que la aproximación del offset para un grado $n = 3$ ($k = 1$) coincide prácticamente con el offset exacto.

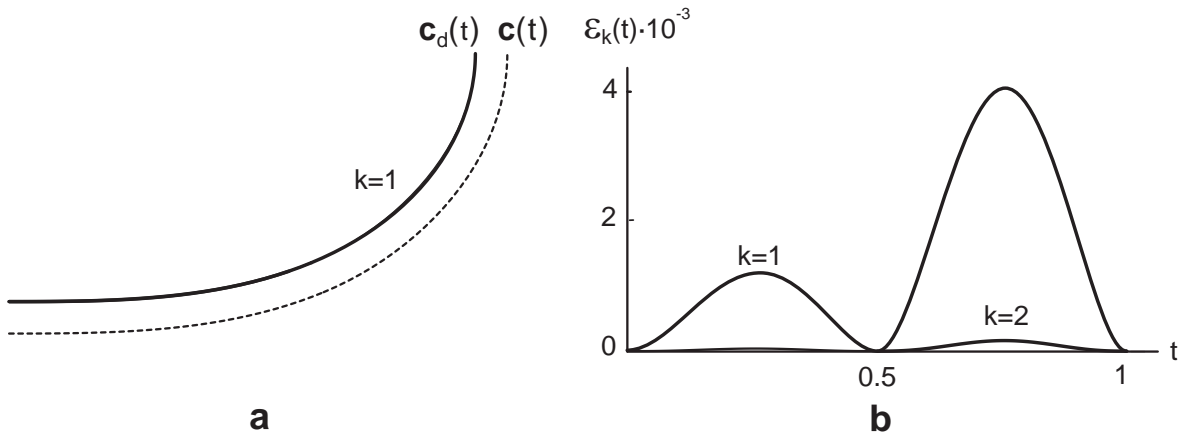


Figura 5.8: (a) Aproximaciones de Hermite $H_k(\mathbf{c}_d; t)$ de orden $k = 1, 2$ del offset de la clotoide para $t \in [0, 1]$ con subdivisión en $t = 0.5$. (b) Error Pitagórico $\varepsilon_k(t)$.

Capítulo 6

Deformación de curvas y superficies

Las técnicas de deformación libre FFD (Free-Form Deformation) modifican la forma de un objeto mediante la deformación de una estructura espacial en la que está contenido. Estas técnicas son muy utilizadas para la modificación de gráficos y generación de animaciones, en las que únicamente se necesita la visualización del objeto generada mediante su aproximación poligonal. Sin embargo, cuando va a ser incorporado en un programa de CAD/CAM para su fabricación, se necesita su expresión explícita. En este caso se debe calcular la composición polinómica del objeto que se desea deformar en la base de Bernstein con la estructura espacial de deformación. Este método presenta el inconveniente de que el resultado posee un grado muy elevado. Se propone un método para generar esta composición en la base s-monomial. Este método permite calcular de manera sencilla la expresión explícita del objeto deformado a la vez que permite reducir su grado, manteniendo controlado en todo momento el grado final, así como el coste de cálculo de la deformación. Finalmente se expresa el objeto deformado en la base de Bernstein, lo que permite su incorporación en un programa de CAD/CAM.

Además se estudian los procesos de doblado (*bending*) y torsión (*twist*) de superficies, operaciones muy útiles para la representación de entidades complejas. Como resultado del uso de estas herramientas de deformación se obtienen objetos no polinómicos, por lo que no pueden expresarse en el estándar NURBS. Se presenta un método para aproximar estas técnicas de deformación, obteniendo una representación polinómica del objeto deformado que puede ser incorporada en los programas de CAD/CAM.

6.1 Introducción

Deformación de objetos

El Diseño Geométrico Asistido por Ordenador (CAGD) se basa principalmente en la representación polinómica en la base de Bernstein (Capítulo 1), lo que permite diseñar una amplia variedad de objetos de una forma muy flexible. Cuando se diseñan objetos muy complejos su número de puntos de control aumenta, por lo que alterar estos objetos requiere modificar un gran número de puntos de control (Farin, 2002).

Una mejora consiste en tratar los puntos de control de manera agrupada en lugar de individualmente, mediante herramientas capaces de transmitir una modificación a la zona del objeto que se desea modificar. Estas herramientas pueden ser estructuras espaciales que contienen al objeto. Al alterar la forma de estas estructuras se transmite la deformación al objeto. La ventaja radica en que resulta más sencillo definir la deformación sobre estas estructuras que directamente sobre el objeto (Cavendish, 1995).

La gran importancia que tiene la deformación libre de objetos queda avalada por la amplia variedad de técnicas publicadas. Cada una de ellas determina una función de deformación que asocia una nueva posición a cada punto del objeto. A continuación se han clasificado y resumido las técnicas más relevantes.

Deformaciones basadas en estructuras espaciales

Los primeros trabajos en deformación libre (FFD) (Sederberg y Parry, 1986) están basados en la deformación de objetos contenidos en una estructura espacial (*lattice*) regularmente dividida y definida por una matriz tridimensional de puntos de control en la base de Bernstein. Posteriormente Griessmair y Purgathofer (1989) presenta una técnica basada en estructuras trivariadas B-spline (B-spline FFD). A continuación Coquillart (1990) extiende la técnica FFD ampliando las estructuras utilizadas a formas no paralelepípedas (EFFD), más apropiadas para definir deformaciones cilíndricas, esféricas, etc. Otra técnica consiste en la utilización de estructuras espaciales racionales (RFFD) (Lamousin y Waggenspack, 1994) donde los pesos asociados a los puntos de la estructura también intervienen en la deformación.

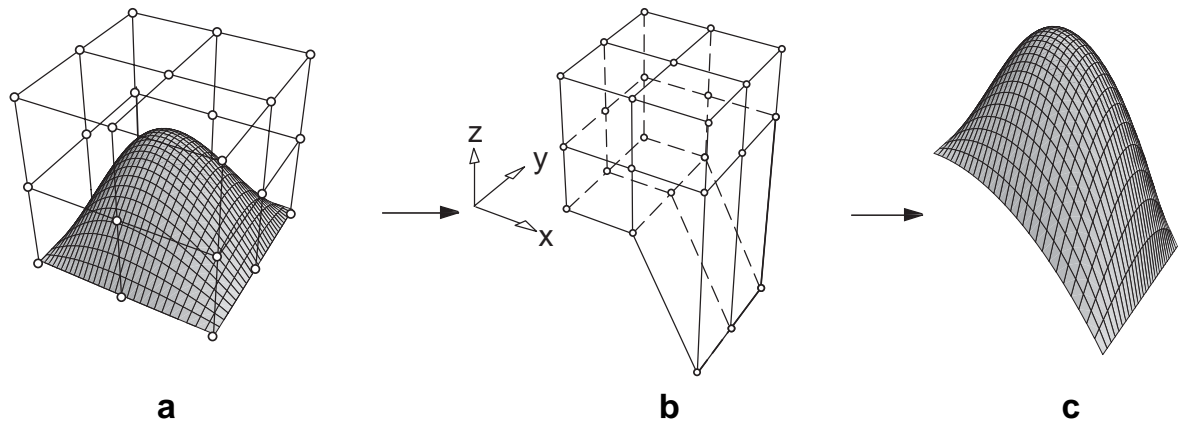


Figura 6.1: Proceso de deformación libre. (a) Objeto dentro de estructura de deformación. (b) Estructura deformada. (c) Objeto deformado.

Cavendish (1995) integra la técnica FFD en el proceso de diseño de superficies funcionales univaluadas. Este tipo de superficies aparecen en piezas fabricadas por estampación o moldeo. La técnica FFD permite aumentar la variedad de las piezas diseñadas.

Hui (2003) propone un método basado en la técnica FFD para deformar modelos representados en CSR (*Constructive Shell Representation*). La representación CSR, presentada por Menon (1994), constituye una alternativa a la representación paramétrica de las superficies que constituyen las fronteras de sólidos. El problema es que está basada en superficies implícitas no admitidas por los principales programas comerciales CAD/CAM.

En todos estos casos la técnica de deformación, representada en la Fig. 6.1, consiste en asociar el objeto a la estructura de deformación (Bechmann, 1994). A continuación se modifica la estructura en función del cambio que se desea realizar en el objeto. Finalmente se expresa el objeto deformado en su referencia inicial.

Deformaciones axiales

Lazarus et al. (1994) proponen como herramienta de deformación una curva (AxDf). En primer lugar se calculan las coordenadas curvilíneas del objeto a deformar y a continuación se modifica la forma de la curva. El objeto deformado se genera manteniendo las coordenadas curvilíneas, anteriormente calculadas, en el sistema de coordenadas de la curva deformada. Los autores utilizan el sistema de coordenadas propuesto por Klok (1986). Este sistema de coordenadas constituye una alternativa al triedro de Frenet, y su principal ventaja es que la rotación del triedro a lo largo de la curva es mínima. Peng et al. (1997) presentan una extensión del método propuesto por Lazarus et al. (1994) en el que se mantiene constante la longitud del objeto deformado.

Una limitación de Lazarus et al. (1994) es que el objeto no puede torsionarse manipulando la curva. Hui (2002) resuelve este problema utilizando un par de curvas B-spline formado por una curva primaria y la aproximación de su offset, llamada curva de orientación. Estas curvas definen la posición y orientación del sistema de coordenadas curvilíneo. Modificando la curva primaria se generan estrechamientos y doblados. La torsión del objeto se consigue girando la curva de orientación alrededor de la curva primaria.

Chang y Rockwood (1994) proponen una técnica de deformación, basada en una generalización del algoritmo de Casteljau, en la que el objeto se deforma a lo largo de una curva de Bézier. El usuario define en cada segmento del polígono de control de la curva de Bézier un sistema de coordenadas local. La deformación del objeto viene determinada por el polígono de control y la dirección de los sistemas de coordenadas locales.

Deformaciones directas

Otras técnicas de deformación libre (Borrel y Bechmann, 1991) se basan en la deformación directa del objeto (DFFD). En este caso no se utiliza una estructura espacial como herramienta de deformación sino que se modifica directamente el espacio que contiene el objeto utilizando una función general de deformación, definida a través de restricciones que determinan el camino a lo largo del cual se pueden mover los puntos del objeto que se desea deformar (Bechmann y Gerber, 2003).

Hu et al. (2001) presentan un método basado en la técnica FFD pero que, en lugar de modificar la estructura de deformación directamente, calcula la posición de los puntos del *lattice* en función de los desplazamientos de varios puntos del objeto.

Otros métodos de deformación

Jin y Li (2000) proponen un método de deformación global basado en la utilización de dos objetos de control *star-shaped*, es decir, que cada uno posee un punto central \mathbf{O} tal que cualquier rayo lanzado desde dicho punto corta a la superficie del objeto una única vez. La función de deformación consiste en un escalado no uniforme, de centro \mathbf{O} que depende de la dirección radial. El factor de escala $e = r_B / r_A$ es al igual cociente de radios, definidos como las distancias de los puntos de los objetos de control A, B a sus respectivos puntos centrales $\mathbf{O}_A, \mathbf{O}_B$.

Ninguno de estos métodos asocia restricciones físicas a la deformación de objetos. Hirota et al. (1999) calculan una deformación de la estructura espacial de tal forma que se preserva el volumen del objeto. Esta técnica resulta muy útil para aplicaciones como el diseño de contenedores, donde el volumen está fijado a priori.

Implementación de las técnicas de deformación libre

La deformación libre de objetos ha sido un tema de investigación muy estudiado en la industria automovilística durante la última década (Field, 2005). Un ejemplo es el trabajo que recientemente ha presentado Sarraga (2003). El autor aplica la técnica FFD para corregir el *springback* y la distorsión en el ensamblaje de paneles de automóviles, disminuyendo el tiempo empleado en el rediseño de dichas piezas. En el caso del *springback*, el panel no adopta la forma impuesta por el útil de estampación ya que tiende a retroceder. Por consiguiente, se debe rediseñar dicho útil para que el panel tome la forma deseada. La distorsión en el ensamblaje aparece cuando un panel se deforma a causa de las cargas impuestas por las piezas adyacentes. En este caso la mejor solución consiste en cambiar la forma de uno o varios paneles.

Las modificaciones para compensar estos defectos están formuladas en función de los desplazamientos de una lista de puntos, proporcionados por el análisis por elementos finitos del modelo CAD/CAM que se desea rediseñar. Asimismo, modificar modelos industriales CAD/CAM es una tarea muy laboriosa, ya que suelen estar formados por un gran número de superficies B-spline y curvas de recorte. Sarraga, partiendo de una lista de puntos y sus desplazamientos, genera una función de deformación volumen B-spline mediante mínimos cuadrados. A continuación, el modelo CAD/CAM deformado exacto se genera mediante la composición del objeto con el volumen deformado. Finalmente el objeto deformado exacto se aproxima mediante superficies B-spline bicúbicas.

Otro ejemplo que ilustra la importancia de las técnicas de deformación libre es su incipiente implementación en los programas de CAD/CAM, como es el caso de UDT (Universal Deformation Technology) que aparecerá en la futura versión 4.0 del programa Rhino (McNeel, 2004).

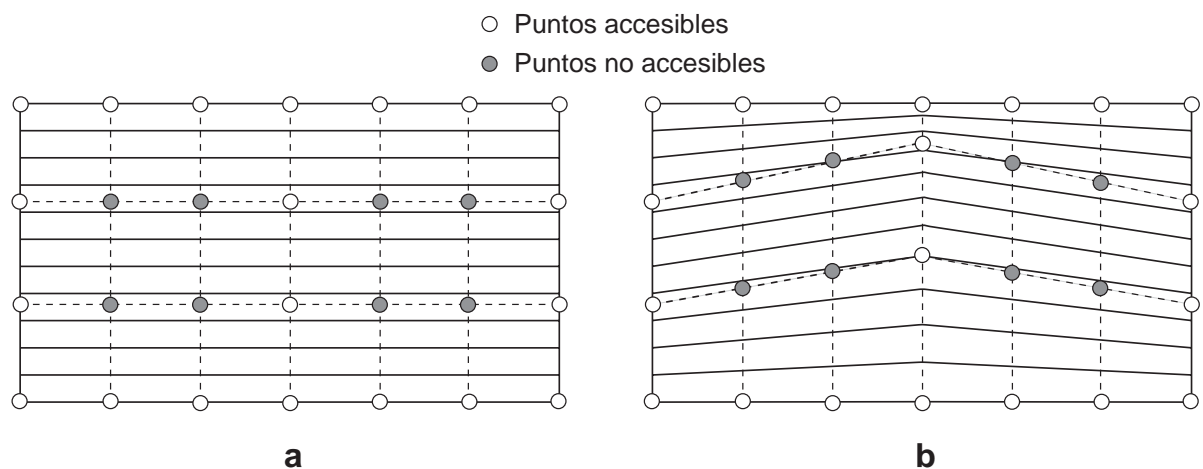


Figura 6.2: Deformación con Adobe Illustrator®. (a) Objeto sin deformar y estructura de deformación. (b) Problemas de continuidad.

El programa Adobe Illustrator[®] posee una herramienta para deformar objetos 2D que emplea la técnica FFD. El objeto deformado se aproxima mediante curvas B-spline cúbicas. No obstante, el programa no muestra su polígono de control. Dicha herramienta utiliza como estructura de deformación una malla de superficies de Coons, cuya ventaja es que permite definir la función de deformación de manera sencilla. Cada superficie de Coons se genera a partir de 4 curvas cúbicas de Bézier. Por consiguiente, la estructura de deformación está compuesta por la unión de mallas de puntos de control de grado (3, 3). Esta herramienta presenta dos problemas. En primer lugar no permite acceder a los puntos interiores de la estructura de deformación, perdiendo flexibilidad en el diseño. En segundo lugar, al utilizar superficies de Coons, aparecen problemas de continuidad en la unión de dos tramos de un mismo objeto deformados con distintas superficies de Coons. Como ejemplo se ha deformado una malla regular ortogonal plana (Fig. 6.2a) utilizando como estructura de deformación 2 superficies de Coons cuyo perímetro coincide con el de la malla. Los problemas de continuidad descritos quedan reflejados en la Fig. 6.2b, donde las líneas de la malla, en las fronteras de los dos tramos deformados, únicamente mantienen continuidad posicional.

Representación del objeto deformado

Cada una de las técnicas de deformación introducidas definen una función de deformación $F: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ que asocia a cada punto \mathbf{p} su nueva posición $F(\mathbf{p})$. Determinada matemáticamente F nos planteamos el problema de representar el objeto final deformado. Dependiendo de la aplicación que se le vaya a dar al objeto deformado se podrán tomar dos caminos bien distintos.

El primer método consiste en generar una aproximación discreta del objeto deformado. Se muestrea una serie de puntos \mathbf{p}_i del objeto original, a continuación se calcula su posición final $F(\mathbf{p}_i)$ y se visualiza como resultado una aproximación poligonal del objeto. El proceso de cálculo es muy sencillo, pero sólo se genera una aproximación, no la representación explícita polinómica (o racional) requerida por los programas de CAD/CAM.

El segundo método consiste en calcular la composición del objeto con la función de deformación. De esta forma se obtiene la expresión explícita exacta del objeto deformado, por lo que el resultado final es transferible a un programa de CAD/CAM. Este método tiene como inconveniente el gran coste del proceso de cálculo y el aumento de la complejidad del objeto deformado respecto al original. La composición es factible con aquellas técnicas cuya función de deformación admita una expresión explícita. A continuación se presentan los diferentes casos que pueden aparecer dependiendo de la naturaleza de la función de deformación.

Función de deformación explícita polinómica

Cuando la función de deformación admite una expresión explícita polinómica, el objeto deformado presenta un grado elevado que impide que el resultado pueda ser incorporado en los programas de CAD/CAM, que suelen tener limitado el grado admisible. En concreto, la composición de una superficie de grado (m_u, m_v) con un volumen de grado (n_x, n_y, n_z) genera una superficie deformada de grado $((n_x+n_y+n_z) \cdot m_u, (n_x+n_y+n_z) \cdot m_v)$.

Un ejemplo son las técnicas de deformación FFD que utilizan estructuras espaciales de deformación prismáticas (excepto la técnica EFFD), representadas de manera estándar utilizando polinomios de Bernstein, las técnicas de deformación directa DFFD (Borrel y Bechmann, 1991; Bechmann y Gerber, 2003), y la técnica de deformación axial propuesta por Chang y Rockwood (1994) (Fig. 6.3).

Función de deformación explícita no polinómica

Cuando la función de deformación admite una expresión explícita no polinómica, la composición de un objeto polinómico con dicha función genera objetos deformados no polinómicos, por lo que no se podrían incorporar en los programas de CAD/CAM. La solución propuesta consiste en generar una aproximación polinómica de la función de deformación, pasando de esta forma al caso anterior. El método de torsión y doblado propuesto por Barr (1984) constituye un ejemplo, ya que la función de deformación involucra funciones trigonométricas.

Función de deformación sin representación explícita

Finalmente, tenemos casos en los que no se puede aplicar la composición del objeto con la función de deformación, debido a que dicha función no admite una representación explícita con funciones elementales. Por ejemplo la técnica EFFD, los métodos de deformación axial (Lazarus et al., 1994; Peng et al. 1997; Hui, 2002) y el método de deformación global propuesto por Jin y Li (2000).

En la técnica EFFD hay que definir una malla de referencia y otra deformada (Fig. 6.4). Ambas mallas determinan las funciones F y G respectivamente, representadas cada una como un producto tensorial de Bezier. La función de deformación, definida como la composición de F^{-1} con G , no tiene una representación explícita, ya que no se puede hallar F^{-1} .

En los métodos de deformación axial ocurre lo mismo que en la técnica EFFD, pero en este caso se tiene una curva de referencia y otra deformada (Fig. 6.5). Por lo tanto, en lugar de asociar el objeto a una malla de referencia, se asocia a una curva de referencia.

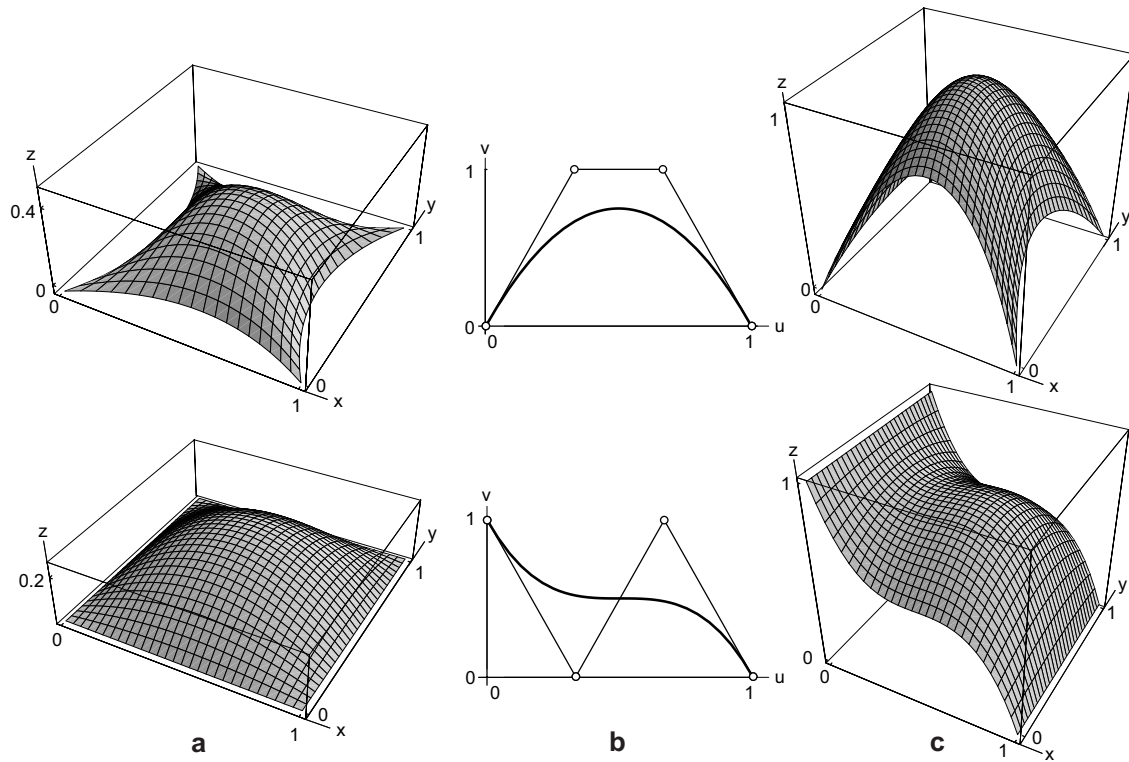


Figura 6.3: Proceso de deformación propuesto por Chang y Rockwood (1994). (a) Superficie sin deformar. (b) Curva de Bézier. (c) Superficie deformada.

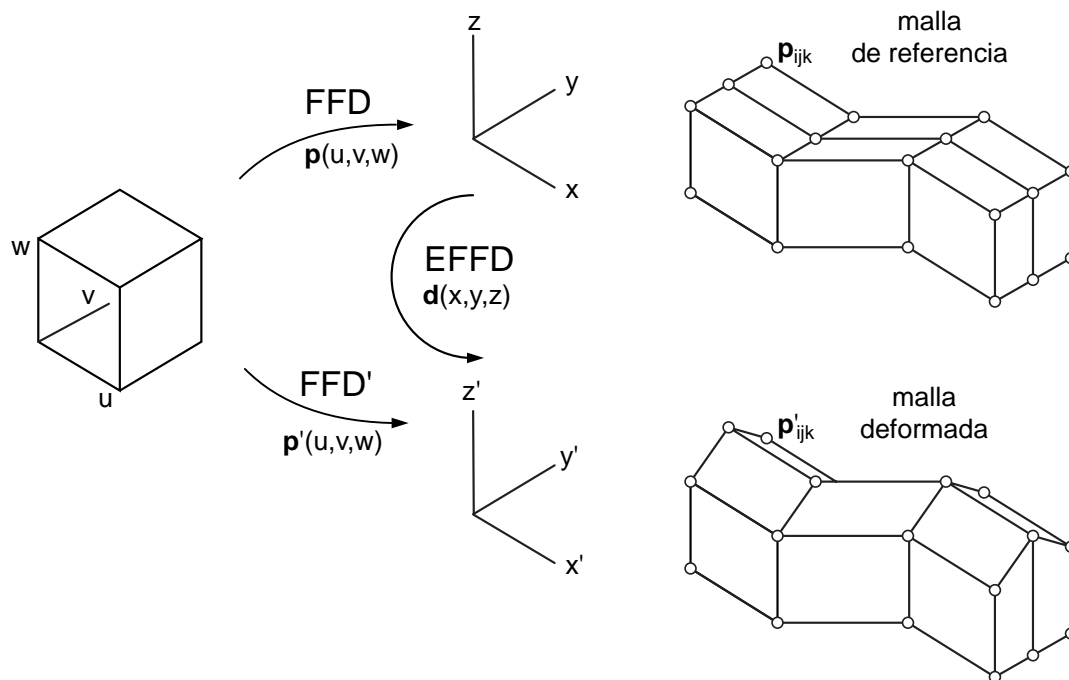


Figura 6.4: Proceso de deformación mediante técnica EFFT.

(Fuente: Sánchez-Reyes y Arranz, 1998)

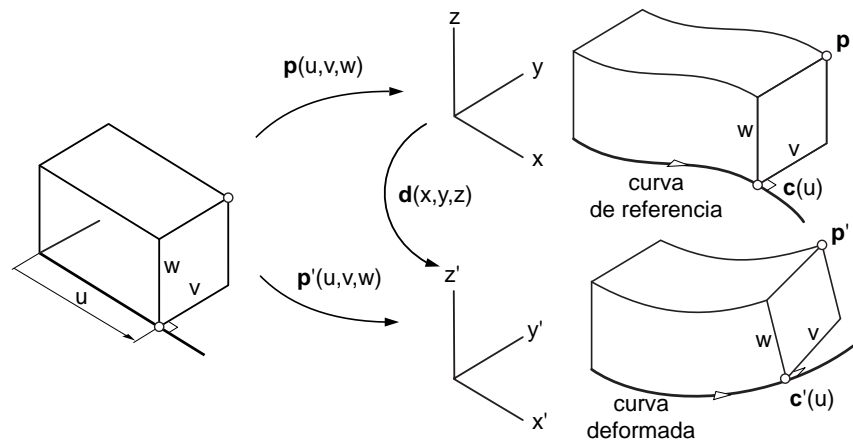


Figura 6.5: Proceso de deformación mediante técnica axial.

(Fuente: Sánchez-Reyes y Arranz, 1998)

Para componer funciones polinómicas en la base de Bernstein, DeRose (1988) propone los métodos de Blossom y el algoritmo del producto, destacando su utilidad en los procesos de deformación. Estos métodos se aplican a estructuras de Bézier tipo Simplexes, representación de Bézier en coordenadas baricéntricas, y se generaliza a otras representaciones de Bézier (DeRose et al., 1993; Liu y Mann, 1997). Finalmente, Patterson (2005) ha propuesto recientemente el cálculo de dicha composición empleando *mixed polynomials*, polinomios definidos en dos conjuntos de variables, tal que son homogéneos sobre cada uno de estos conjuntos. No obstante, estos métodos no tratan el control de la complejidad del objeto resultante ni de las representaciones intermedias en el proceso de composición.

Feng et al. (2002) proponen la composición polinómica en la base de Bézier mediante un método de interpolación. Este método se aplica para deformar objetos poligonales, formados por triángulos, mediante la técnica B-Spline FFD. En primer lugar se divide el objeto por los planos correspondientes a los nodos de la herramienta de deformación. A continuación se aplica la técnica FFD a los tramos de Bézier que constituyen dicha herramienta. El objeto resultante se representa mediante superficies de Bézier recortadas, cuyas curvas de recorte son los bordes exteriores de dichos polígonos coplanarios.

6.2 Solución propuesta para las técnicas FFD y RFFD: Composición vía convoluciones discretas

La solución propuesta por Sánchez-Reyes y Arranz (1998) (Fig. 6.6) para la técnica FFD consiste en componer vía convoluciones discretas (Capítulo 2) el objeto que se desea deformar en la base s-monomial con la herramienta de deformación definida en la base de Bernstein. Como resultado es trivial obtener una representación de grado reducido del objeto deformado en la base s-monomial.

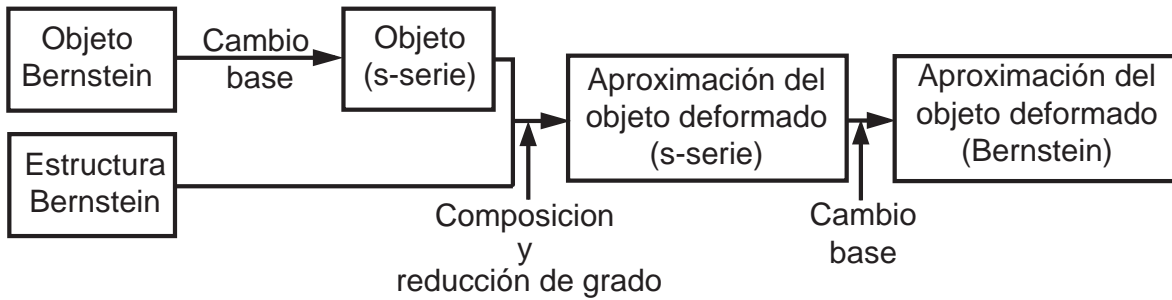


Figura 6.6: Proceso de deformación libre propuesto.

La composición vía convoluciones discretas nos permite calcular de manera sencilla los coeficientes del objeto deformado expresado en la base s -monomial. Finalmente el objeto deformado de grado reducido se expresa en la base de Bernstein, operación que utiliza matrices de cambio de base de dimensiones reducidas.

La ventaja de este método radica en la capacidad de reducir el grado de manera sencilla, controlando la complejidad de la representación al eliminar los términos de grado elevado, los cuales tienen una influencia muy pequeña en la forma del objeto. Además de reducir el grado del objeto deformado, también se reduce el coste del cálculo de la deformación. Este método se ha aplicado a la técnica de deformación libre FFD, RFFD y B-spline FFD. Asimismo no se puede generalizar al método EFFD pues genera objetos no polinómicos ya que involucra la inversa del producto tensorial de Bézier de la malla de referencia.

Una alternativa consiste en componer en la base de Bernstein vía convoluciones discretas el objeto con la estructura espacial (Sánchez-Reyes, 2003a). De esta forma se consigue la representación exacta del objeto deformado en la base de Bernstein, pasando posteriormente a la base s -monomial para reducir el grado. Este cambio de base tiene el inconveniente de utilizar matrices de dimensiones muy elevadas. Finalmente el objeto deformado de grado reducido se convierte a la base de Bernstein. Como muestra la Fig. 6.7 este método no parece adecuado ya que, según resultados preliminares, la composición en la base de Bernstein introduce errores numéricos.

Se ha generado la composición en la base de Bernstein, calculando a continuación la reducción de grado tanto en la base s -monomial como en la de Bernstein (Fig. 6.8), derivada de la transformación entre la base de Bernstein escalada y la s -monomial expuesta por Sánchez-Reyes (1997). El resultado obtenido, el mismo con ambos métodos, presenta errores numéricos, por consiguiente no pueden provenir de la reducción de grado, sino de la composición. En cambio, cuando la composición se genera en la base s -monomial no aparecen errores.

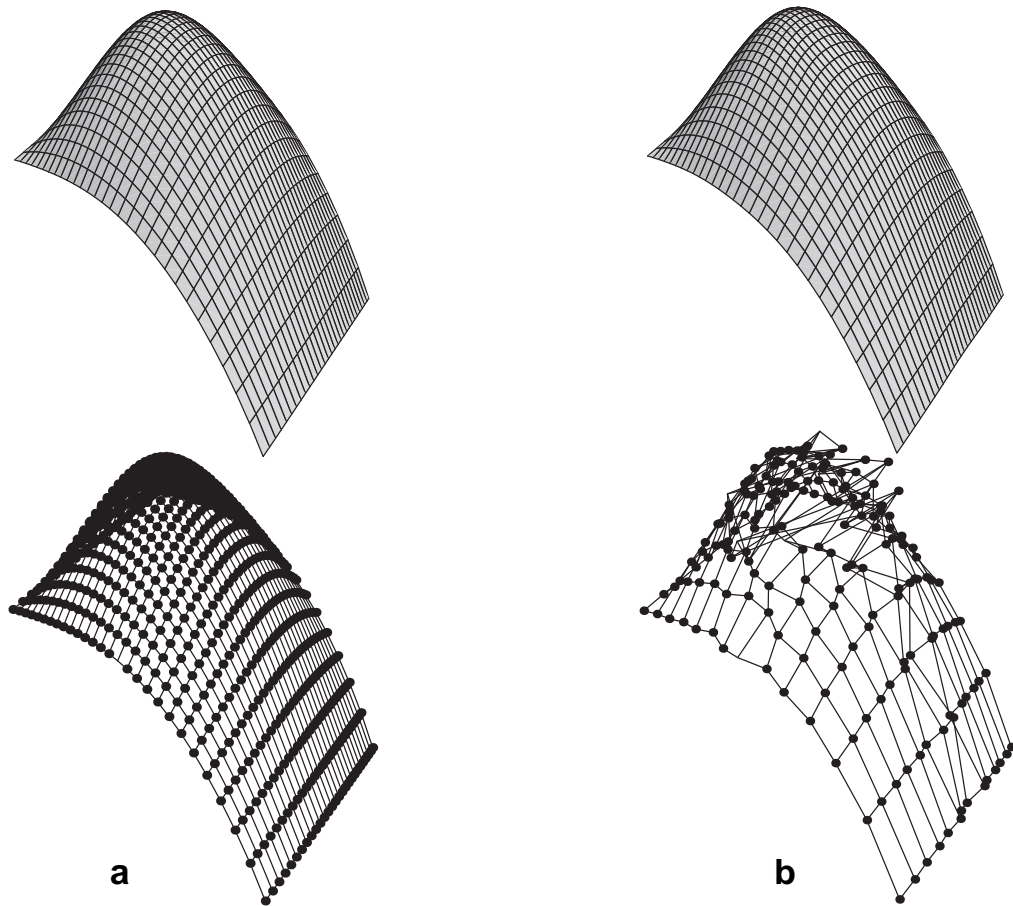


Figura 6.7: Errores numéricos. Composición superficie de Bézier de grado $(3, 3)$ con volumen de Bézier de grado $(n_x, n_y, n_z) = (2, 4, 3)$. (a) Superficie deformada de grado $(27, 27)$. (b) Superficie aproximada de grado $(13, 13)$.

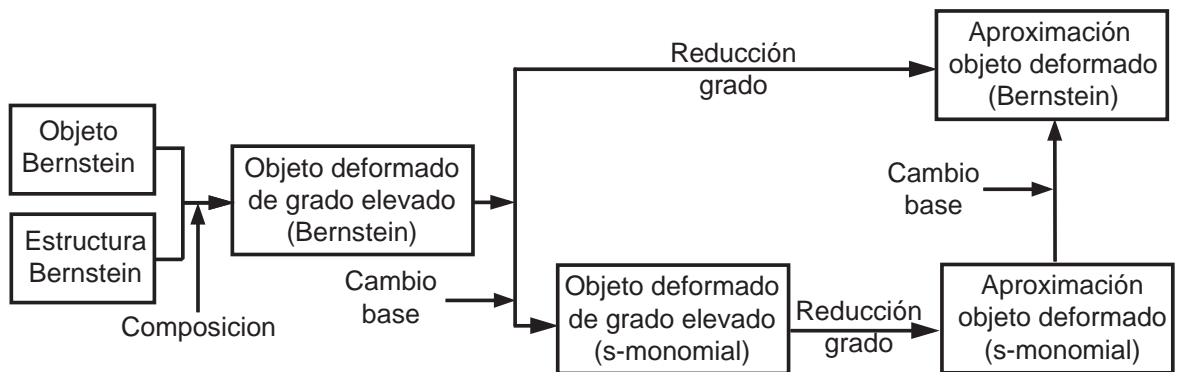


Figura 6.8: Composición en la base de Bernstein. Reducción de grado en la base s-monomial y base de Bernstein.

6.3 Deformación de curvas y superficies

A continuación se expone el proceso de deformación de curvas y superficies propuesto, basado en la composición del objeto a deformar definido en la base s-monomial con la herramienta de deformación definida en la base de Bernstein.

Deformación de una curva en R^2

En el caso de la deformación de una curva en R^2 (Fig. 6.9), la herramienta de deformación utilizada es una superficie plana. Partimos de una curva $\mathbf{a}(t) = (u(t), v(t))$ de grado n definida en la base s-monomial y una superficie $\mathbf{b}(u, v) = (x(u, v), y(u, v))$ de grado (m_u, m_v) definida en la base de Bernstein:

$$\mathbf{b}(u, v) = \sum_{i=0}^{m_u} \sum_{j=0}^{m_v} \mathbf{b}_{i,j} B_{i,j}^{m_u, m_v}(u, v),$$

con puntos de control $\mathbf{b}_{i,j} = (x, y)_{i,j}$. La curva deformada $\mathbf{a}_{\text{FFD}}(t)$ se genera calculando la composición $\mathbf{b}(u(t), v(t))$, reemplazando:

$$\begin{cases} u \rightarrow u(t) \\ v \rightarrow v(t) \end{cases} \text{ en } \mathbf{b}(u, v).$$

De esta manera obtenemos la curva deformada de grado $(m_u + m_v) \cdot n$ en la base s-monomial:

$$\mathbf{a}_{\text{FFD}}(t) = \sum_{i=0}^{m_u} \sum_{j=0}^{m_v} \mathbf{b}_{i,j} B_{i,j}^{m_u, m_v}(u(t), v(t)).$$

Esta operación no presenta gran complejidad ya que sólo involucra sumas y multiplicaciones de s-series (Capítulo 2).

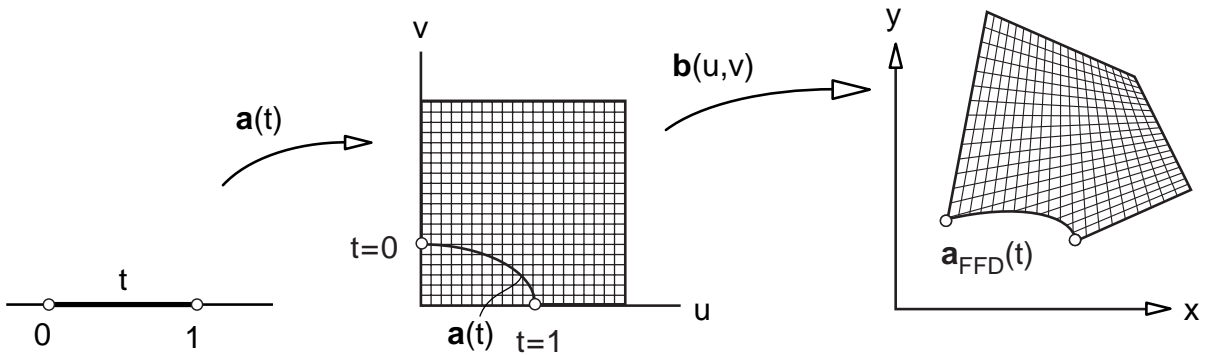


Figura 6.9: Proceso de deformación de una curva en R^2 .

Curvas sobre superficies (*trimming curve*)

En el caso de la deformación de una curva en R^3 (Fig. 6.10), el resultado obtenido es una *trimming curve* que genera una superficie recortada. El proceso de composición es el mismo que el descrito anteriormente, salvo que en este caso la herramienta de deformación es una superficie $\mathbf{b}(u, v)$ de grado (m_u, m_v) en R^3 definida en la base de Bernstein:

$$\mathbf{b}(u, v) = \sum_{i=0}^{m_u} \sum_{j=0}^{m_v} \mathbf{b}_{i,j} B_{i,j}^{m_u, m_v}(u, v),$$

con puntos de control $\mathbf{b}_{i,j} = (x, y, z)_{i,j}$. La curva deformada $\mathbf{a}_{\text{FFD}}(t)$ se genera calculando la composición $\mathbf{b}(u(t), v(t))$, para lo cual simplemente remplazaremos:

$$\begin{cases} u \rightarrow u(t) \\ v \rightarrow v(t) \end{cases} \text{ en } \mathbf{b}(u, v),$$

obteniendo finalmente la curva deformada de grado $(m_u + m_v) \cdot n$ en la base s-monomial:

$$\mathbf{a}_{\text{FFD}}(t) = \sum_{i=0}^{m_u} \sum_{j=0}^{m_v} \mathbf{b}_{i,j} B_{i,j}^{m_u, m_v}(u(t), v(t)).$$

La composición de una curva B-spline con una superficie de Bézier no entraña dificultad, ya que simplemente consiste en aplicar el método propuesto en este apartado tantas veces como tramos constituyan la curva original. En el caso de la composición con una superficie B-spline, se precisan divisiones adicionales de la curva al atravesar líneas isoparamétricas correspondientes a los nodos de la superficie. Por consiguiente, el número de tramos de la curva aumenta. Renner y Weiß (2004) han presentado recientemente dos métodos para aproximar la composición de curvas y superficies B-spline. No obstante, ambos métodos utilizan una formulación muy compleja ya que están basados en la representación exacta de la curva mediante la composición vía Blossom.

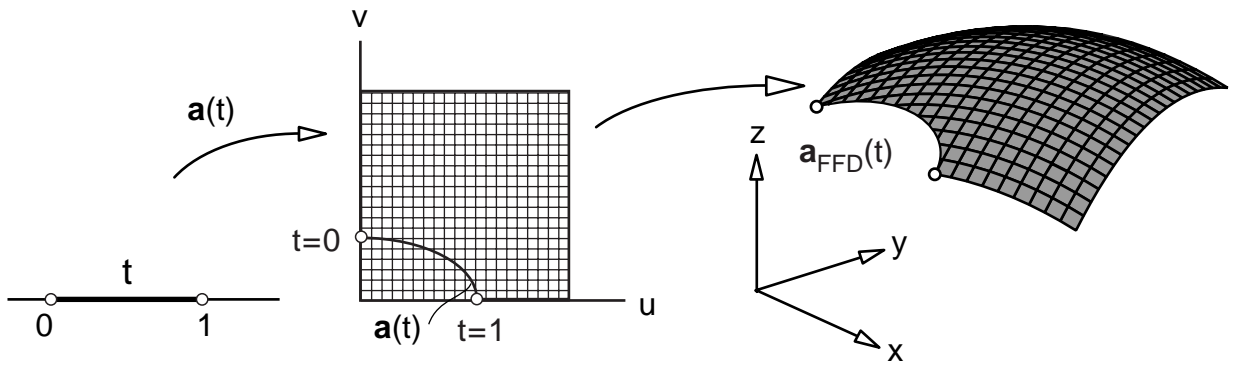


Figura 6.10: Proceso de deformación de una curva en R^3 . (Fuente: Sánchez-Reyes, 2003a)

Deformación de una superficie

En el caso de la deformación de una superficie en R^3 , la herramienta de deformación es un volumen (Fig. 6.11). El proceso de composición es el mismo que el descrito anterior. Partimos de una superficie $\mathbf{b}(u, v) = (x(u, v), y(u, v), z(u, v))$ de grado (m_u, m_v) definido en la base s-monomial y un volumen $\mathbf{d}(x, y, z)$ de grado (n_x, n_y, n_z) definido en la base de Bernstein:

$$\mathbf{d}(x, y, z) = \sum_{i=0}^{n_x} \sum_{j=0}^{n_y} \sum_{k=0}^{n_z} \mathbf{d}_{i,j,k} B_{i,j,k}^{n_x, n_y, n_z}(x, y, z),$$

con puntos de control $\mathbf{d}_{i,j,k} = (x, y, z)_{i,j,k}$. La superficie deformada $\mathbf{b}_{\text{FFD}}(t)$ se genera calculando la composición $\mathbf{d}(x(u, v), y(u, v), z(u, v))$, para lo cual simplemente remplazaremos:

$$\begin{cases} x \rightarrow x(u, v) \\ y \rightarrow y(u, v) \\ z \rightarrow z(u, v) \end{cases} \text{ en } \mathbf{d}(x, y, z),$$

obteniendo finalmente la superficie deformada de grado $((n_x + n_y + n_z) \cdot m_u, (n_x + n_y + n_z) \cdot m_v)$ en la base s-monomial:

$$\mathbf{b}_{\text{FFD}}(x, y, z) = \sum_{i=0}^{n_x} \sum_{j=0}^{n_y} \sum_{k=0}^{n_z} \mathbf{d}_{i,j,k} B_{i,j,k}^{n_x, n_y, n_z}(x(u, v), y(u, v), z(u, v)).$$

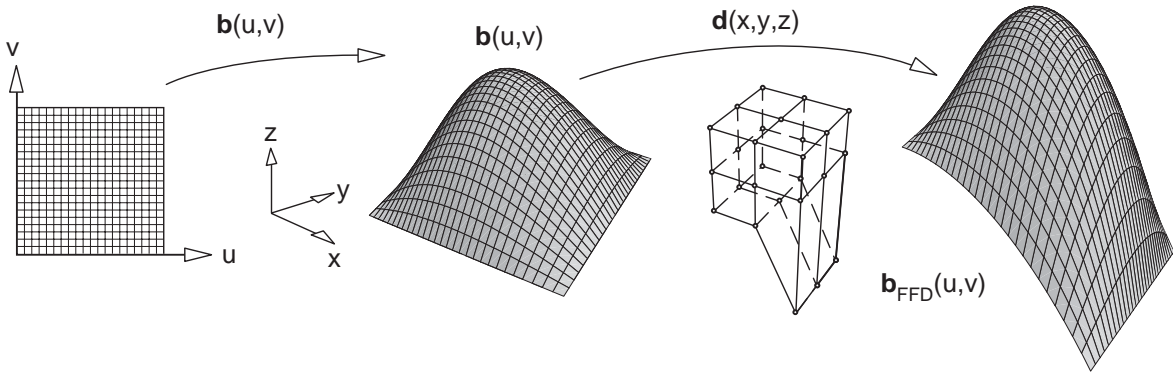


Figura 6.11: Proceso de deformación de una superficie.

Deformación racional

La solución propuesta también se puede aplicar a la técnica RFFD simplemente calculando las coordenadas homogéneas del objeto deformado exacto. Para reducir de grado el objeto resultante se pueden utilizar dos métodos:

- 1- Aproximar cada una de las coordenadas homogéneas, generando una representación racional.
- 2- Aproximar las coordenadas afines del objeto deformado, generando en este caso una representación polinómica.

La Fig. 6.12 muestra la deformación de una curva de Bézier cúbica, utilizando una superficie racional de Bézier plana de grado $(m_u, m_v) = (4, 3)$ como herramienta de deformación, con los pesos asociados a los puntos igual a la unidad, excepto el peso w de uno de ellos. En primer lugar se ha pasado la curva de la base de Bernstein a la base s-monomial aplicando la ecuación (3.8) para funciones univariadas. Como resultado de la composición entre la curva y la superficie racional se obtiene una curva racional de grado $n = 21$ (Fig. 6.12a). La aproximación de las coordenadas homogéneas genera resultados convergentes (Fig. 6.12b). Sin embargo, la aproximación de las coordenadas afines genera expansiones no convergentes (Fig. 6.12c), debido a la aparición de singularidades, imaginarias y/o reales, procedentes de los ceros del denominador.

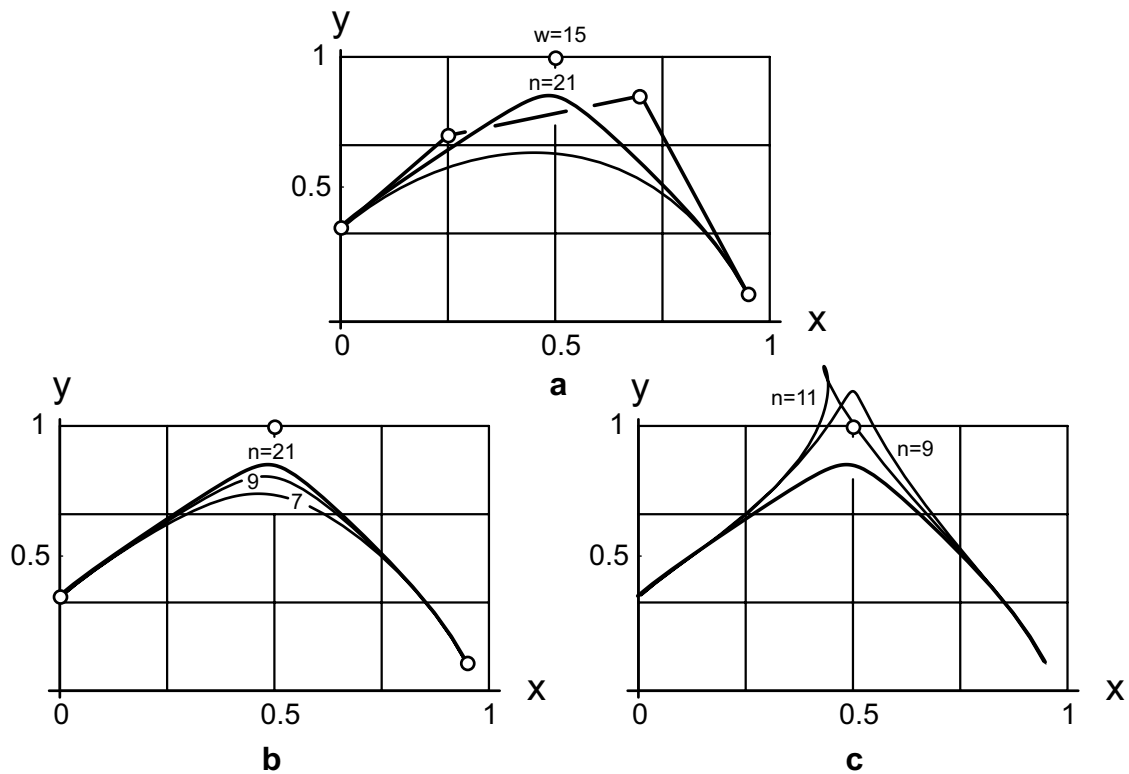


Figura 6.12: Deformación libre racional RFFD. (a) Curva de Bézier original (grado $n = 3$) y curva deformada exacta (grado $n = 21$). (b) Aproximación de coordenadas homogéneas (grado $n = 7, 9$). (c) Aproximación de coordenadas afines (grado $n = 9, 11$).

Deformación local de objetos

Un tema de gran interés es la deformación local de objetos. Se propone utilizar la técnica B-spline FFD (Griessmair y Purgathofer, 1989) que emplea estructuras trivariadas B-spline. En primer lugar se divide el objeto de Bézier por las intersecciones con los planos isoparamétricos correspondientes a los nodos de la herramienta de deformación. Cada tramo se representa mediante una superficie de Bézier recortada (Feng et al. 2002), cuya curvas de recorte son las intersecciones del objeto con los planos de subdivisión. Seguidamente se deforma la malla B-spline y se divide en sus volúmenes de Bézier componentes. A continuación, utilizando el método presentado en el Capítulo 6, se calcula la composición de cada tramo del objeto con el volumen de Bezier en el que está contenido. De esta forma, manteniendo las curvas de recorte, se genera la representación exacta, de grado elevado, del objeto deformado.

El problema es que cuando reducimos de grado el objeto exacto deformado no se asegura la continuidad en la unión de dos tramos deformados, ya que la aproximación de Hermite sólo garantiza continuidad en las fronteras de las superficies de Bézier originales antes del recorte.

Estos problemas se pueden evitar parametrizando una superficie recortada como una superficie de Bézier no recortada y aplicando a continuación la propuesta presentada anteriormente (Fig. 6.13). La parametrización citada se realizará componiendo la superficie plana de Coons, definida por las respectivas curvas de recorte, con la superficie original (Vries-Baayens, 1991). El inconveniente de este método es que genera un gran número de superficies, además el grado de las superficies resultantes es muy elevado. Sin embargo, esta parametrización nos permite calcular una aproximación del objeto deformado que garantiza la continuidad en la unión, a lo largo de las curvas de recorte, de los diferentes tramos que lo constituyen.

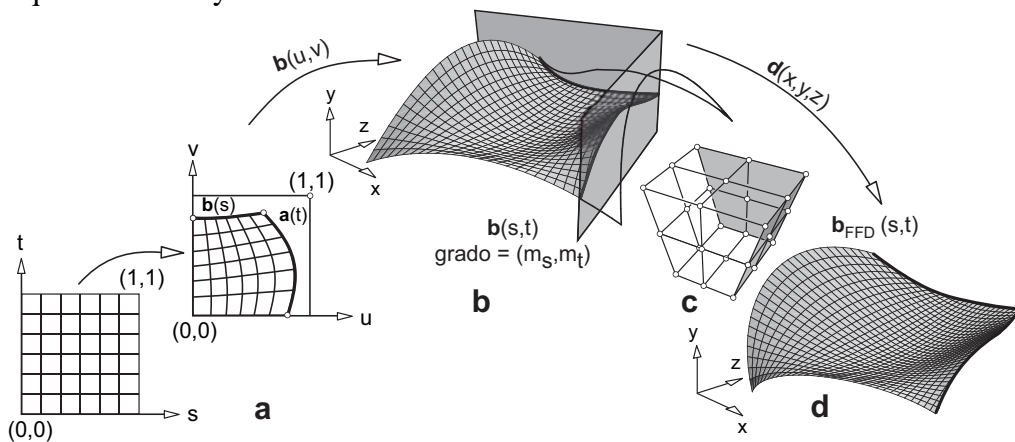


Figura 6.13: Deformación B-spline FFD. (a) Parametrización del dominio de una superficie de Bézier recortada con 2 curvas $a(t)$ y $b(s)$. (b) Superficie de Bézier en R^3 . (c) Volumen de Bézier. (d) Superficie deformada.

En general la conversión de una superficie recortada, definida por una superficie de Bézier de grado (m_u, m_v) y 4 curvas de recorte de grado n , genera una superficie de grado $(m_s, m_t) = (n \cdot (m_u + m_v), n \cdot (m_u + m_v))$. Normalmente tendremos 4 curvas de recorte, aunque por simplicidad en la Fig. 6.13 se han representado dos. La composición de esta superficie con un volumen de grado (n_x, n_y, n_z) origina una superficie deformada de grado muy elevado $((n_x + n_y + n_z) \cdot m_s, (n_x + n_y + n_z) \cdot m_t)$. Por ejemplo, el resultado de deformar una superficie bicúbica y curvas de recorte cúbicas con un volumen de grado $(1, 1, 1)$ es una superficie de grado $(54, 54)$.

6.4 Ejemplos de deformación libre de objetos

En este apartado se muestran una serie de ejemplos que ilustran el método propuesto de deformación.

Deformación de una curva en R^2

Como primer ejemplo se ha deformado una curva de Bézier de grado $n = 3$. En este caso se ha utilizado una superficie de Bézier en R^2 de grado $(m_u, m_v) = (4, 3)$ como herramienta de deformación (Fig. 6.14a). En primer lugar se ha pasado la curva de la base de Bernstein a la base s-monomial aplicando la ecuación (3.8) para funciones univariadas. A continuación se ha deformado la superficie, transmitiéndose dicha deformación a la curva original. Como resultado de la composición entre la curva y la superficie se obtiene una curva de grado $n = 21$ (Fig. 6.14b). La Fig. 6.14c muestra la aproximación de grado $n = 5$ de la curva deformada exacta. El error de dicha aproximación se muestra en la Fig. 6.14d.

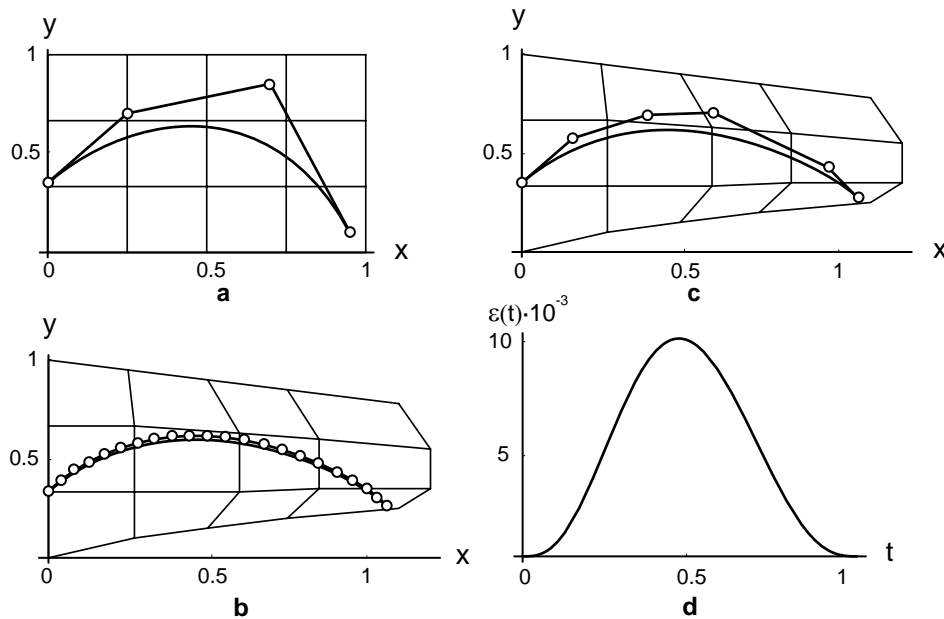


Figura 6.14: Deformación de una curva de Bézier. (a) Curva de Bézier original (grado $n = 3$).

(b) Curva deformada exacta (grado $n = 21$). (c) Reducción de grado (grado $n = 5$).

(d) Error $\varepsilon(t)$.

Curva sobre superficie (trimming curve)

En segundo lugar se han compuesto dos curvas de Bézier de grado $n = 5$ (Fig. 6.15a), que aproximan un círculo, con una superficie de Bézier en R^3 de grado $(5, 5)$ (Fig. 6.15b). Como resultado se han obtenido dos *trimming curve* en R^3 de grado $n = 50$ (Fig. 6.15c). Con el objeto de reducir el grado de dichas curvas se ha calculado la aproximación de grado $n = 7$ y $n = 9$, respectivamente (Fig. 6.16a).

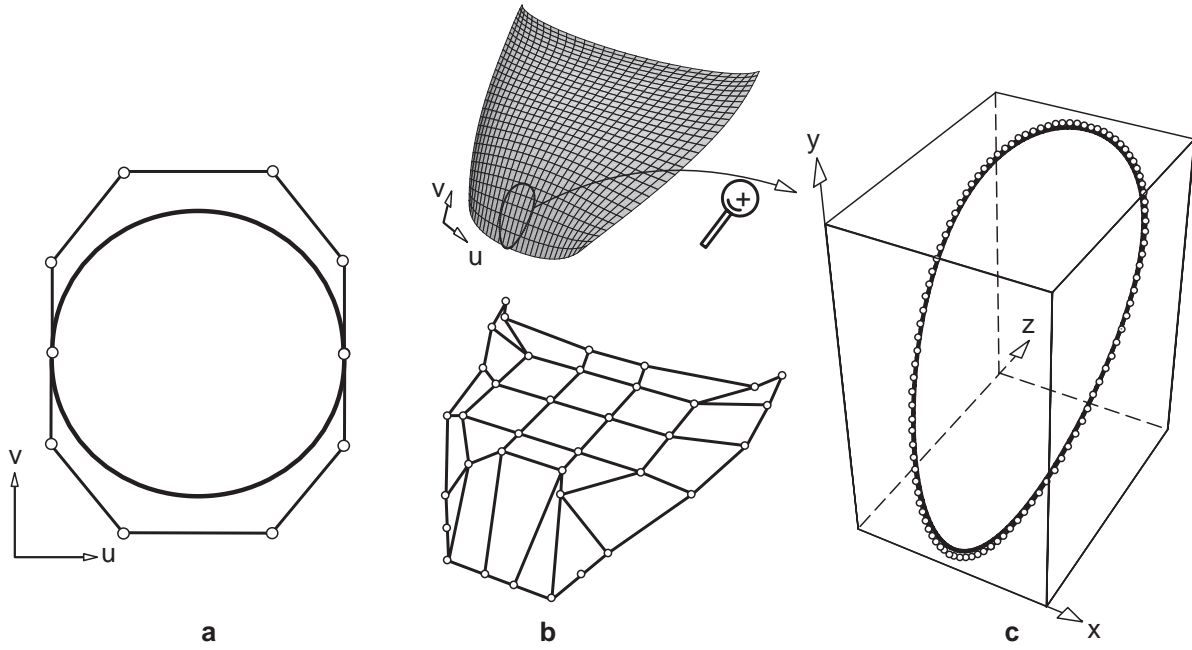


Figura 6.15: Composición de dos curvas de Bézier con superficie en R^3 . (a) Curvas sin deformar de Bézier (grado $n = 5$). (b) Superficie de grado $(5, 5)$. (c) Curvas de Bézier deformadas ($n = 50$).

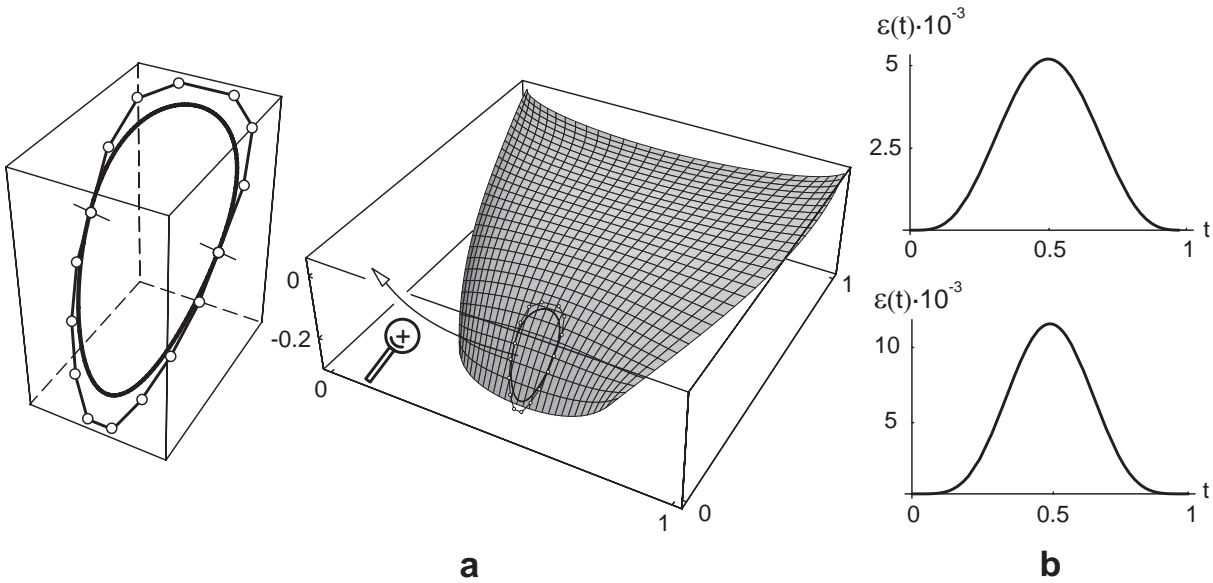


Figura 6.16: Composición de dos curvas de Bézier con superficie en R^3 . (a) Aproximaciones (grado $n = 7, 9$). (b) Error $\varepsilon(t)$.

El error de estas aproximaciones se ha representado en la Fig. 6.16b, donde puede observarse cómo, al ser el error nulo en los extremos, la unión de ambas aproximaciones no presenta huecos.

Deformación de una superficie

Finalmente se ha deformado la superficie del ejemplo anterior (Fig. 6.17a), para lo que se ha utilizado un volumen de Bézier de grado $(1,1,1)$ como herramienta. En este caso se ha modificado la forma rotando los puntos de una de las caras del volumen (Fig. 6.17b). En primer lugar se ha pasado la superficie de la base de Bernstein a la s -monomial aplicando la ecuación (3.9) para funciones bivariadas. A continuación se ha calculado la composición entre la superficie y el volumen deformado, transmitiéndose así la deformación del volumen a la superficie. Como resultado se obtiene una superficie deformada de grado $(15, 15)$ representada en la Fig. 6.17c.

La Fig. 6.18 muestra la aproximación de grado $(7, 7)$ de la superficie deformada exacta, así como el error de dicha aproximación.

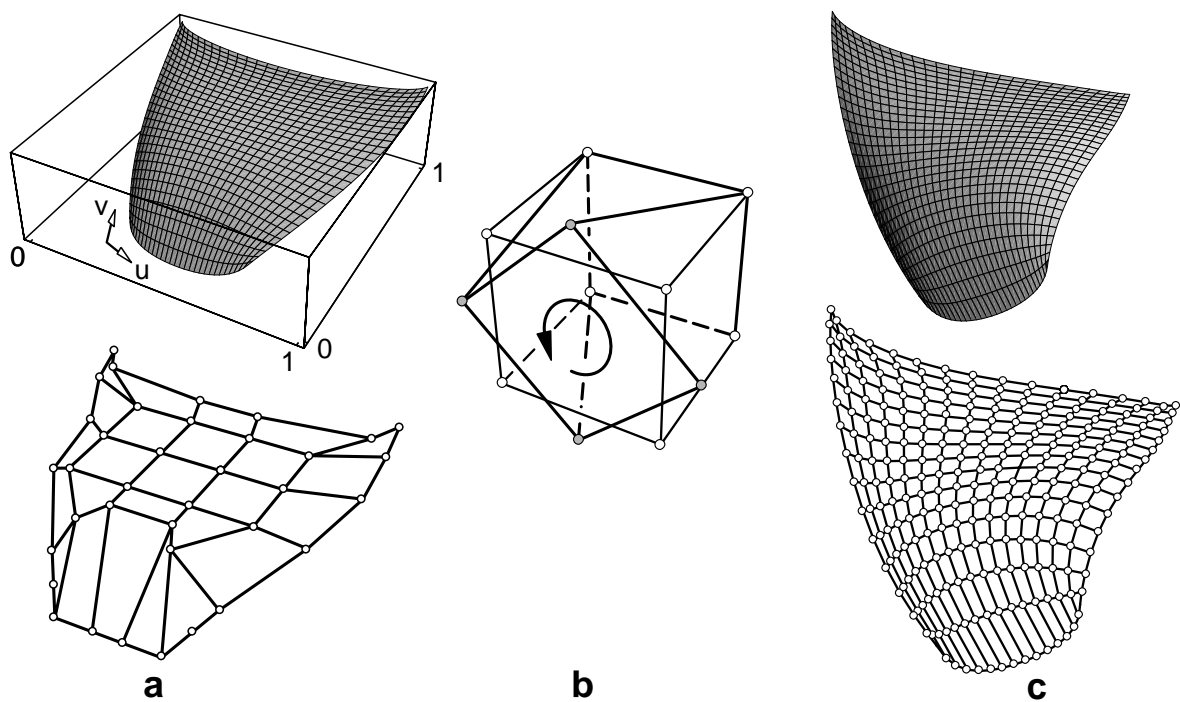


Figura 6.17: Deformación de una superficie de Bézier.(a) Superficie original de grado $(5, 5)$.
(b) Volumen de Bézier de grado $(1,1,1)$. (c) Superficie deformada exacta de grado $(15, 15)$.

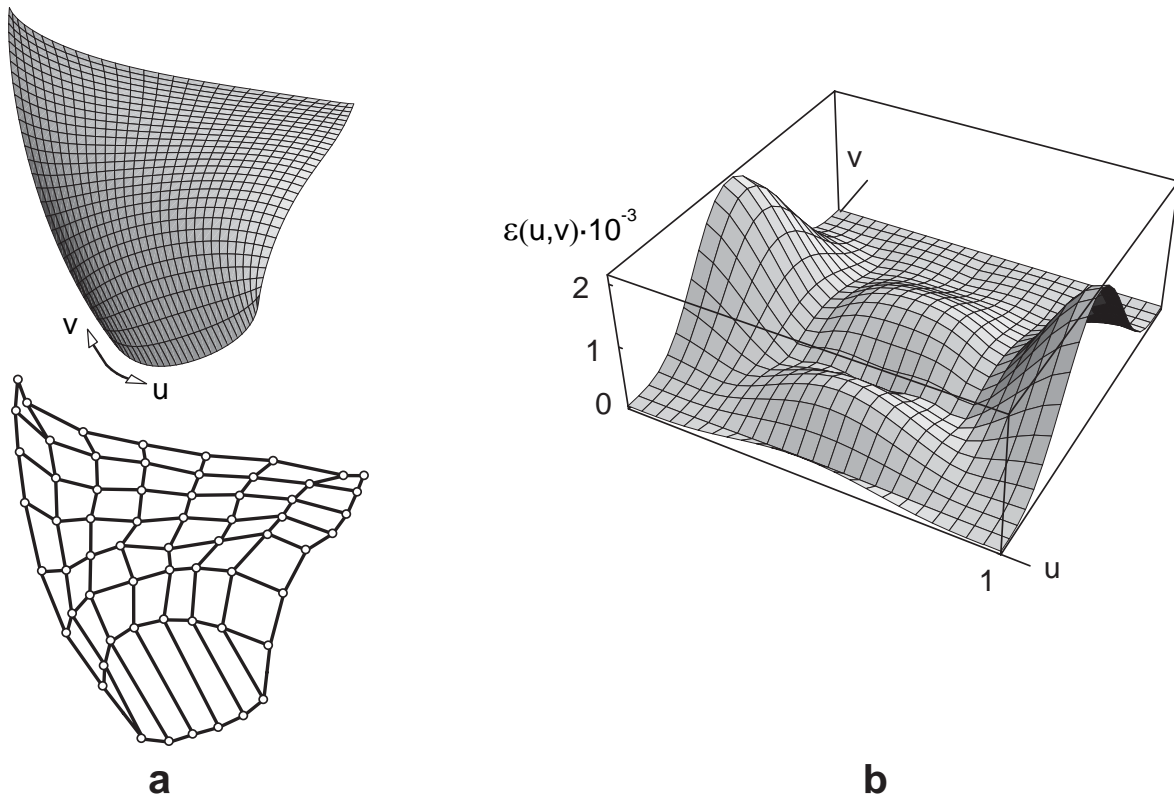


Figura 6.18: Deformación de una superficie de Bézier. (a) Superficie de Bézier aproximada de grado (7, 7). (b) Error $\varepsilon(u, v)$.

6.5 Torsión y doblado de objetos

En este apartado se estudia la torsión (twist) y doblado (bending) de objetos, operaciones muy útiles para la representación de objetos complejos. Estas técnicas de deformación resultan muy intuitivas, ya que permiten simular procesos de fabricación de objetos, proporcionando herramientas de diseño muy sencillas de utilizar.

El problema que presentan estas herramientas de deformación es que aún cuando el objeto inicial sea polinómico, el deformado no lo será. Como solución se propone calcular la aproximación de Hermite de la función de deformación, generando a continuación su representación en la base de Bernstein.

Los principales programas comerciales de diseño no tienen implementadas estas técnicas. El programa Adobe Illustrator[®] constituye una excepción, pues posee una herramienta para doblar objetos 2D basada en la técnica FFD. Como herramienta de deformación bastaría con la malla de la superficie reglada que conecta las aproximaciones de dos arcos de circunferencia concéntricos. Adobe Illustrator[®], por simplicidad en su implementación, utiliza una herramienta compuesta por 3 mallas de Coons bicúbicas.

Torsión de objetos

La torsión alrededor del eje z de un objeto definido en R^3 por sus coordenadas (x, y, z) viene dada por las siguientes ecuaciones (Barr, 1984):

$$\begin{cases} X(u, v) = xC_\theta - yS_\theta & \theta = k \cdot z \\ Y(u, v) = xS_\theta + yC_\theta, & C_\theta = \cos \theta \\ Z(u, v) = z & S_\theta = \sin \theta \end{cases} \quad (6.1)$$

donde k es el ángulo de torsión por unidad de longitud.

El objeto deformado se genera calculando la composición del objeto con la función de deformación. Como muestra la ecuación (6.1), la función de deformación no es polinómica, por lo que el resultado de esta composición no puede incorporarse en un programa de CAD/CAM. Por consiguiente, se debe generar una aproximación polinómica de la función de deformación (6.1), calculando a continuación la composición del objeto con la aproximación de la función de deformación.

El método propuesto consiste en generar la aproximación de Hermite de grado n_z de las funciones C_θ , S_θ en los extremos del intervalo $z \in [z_{min}, z_{max}]$ que contiene completamente al objeto. Esta aproximación se calculará, en términos de las derivadas de los puntos extremos (2.6), aplicando la ecuación (2.7) que genera los coeficientes de una s-serie univariada. Finalmente, tras convertir el resultado a la base de Bernstein, se generan los puntos de control de la aproximación de la función de deformación. En concreto, la estructura de deformación estará formada por una malla lineal en las direcciones x e y , y de grado n_z en la dirección z . Como ejemplo ilustrativo se ha simulado la torsión del capó de un coche representado mediante una superficie de Bézier de grado (5, 5) (Fig. 6.19).

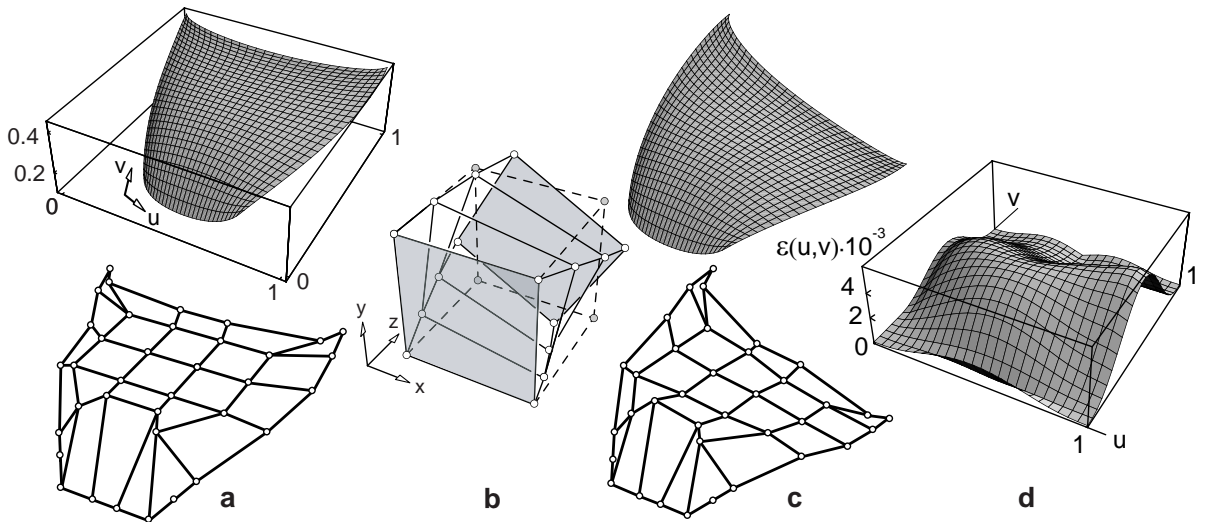


Figura 6.19: Torsión de superficie de Bézier. (a) Superficie original de grado (5, 5).

(b) Aproximación de grado $(n_x, n_y, n_z) = (1, 1, 3)$ de la función de deformación. (c) Aproximación de grado (5, 5) de la superficie deformada. (d) Error $\varepsilon(u, v)$.

Doblado de objetos

El doblado de un objeto a lo largo del eje x , con centro en el origen, viene expresado por las ecuaciones (Barr, 1984):

$$\begin{aligned} X &= \begin{cases} -S_\theta (y - R) & \text{si } x_{\min} \leq x \leq x_{\max} \\ -S_\theta (y - R) + C_\theta (x - x_{\min}) & \text{si } x \leq x_{\min} \\ -S_\theta (y - R) + C_\theta (x - x_{\max}) & \text{si } x \geq x_{\max} \end{cases} \\ Y &= \begin{cases} C_\theta (y - R) + R, & \text{si } x_{\min} \leq x \leq x_{\max} \\ C_\theta (y - R) + R + S_\theta (x - x_{\min}) & \text{si } x \leq x_{\min} \\ C_\theta (y - R) + R + S_\theta (x - x_{\max}) & \text{si } x \geq x_{\max} \end{cases}, \quad (6.2) \\ Z &= z \end{aligned}$$

con:

$$R = 1/k, \quad \theta = k \cdot \hat{x}, \quad C_\theta = \cos \theta, \quad S_\theta = \sin \theta,$$

donde k , expresada en radianes por unidad de longitud, representa la curvatura, R es el radio de curvatura, y:

$$\hat{x} = \begin{cases} x_{\min}, & \text{si } x \leq x_{\min} \\ x, & \text{si } x_{\min} \leq x \leq x_{\max} \\ x_{\max}, & \text{si } x \geq x_{\max} \end{cases}.$$

Como muestran las ecuaciones (6.2), en el proceso de doblado aparecen 2 regiones, los extremos y la parte central (Fig. 6.20a). En los extremos el ángulo de doblado es constante. En esta región la deformación consiste en una rotación y una traslación del objeto. En la parte central, llamada también región de doblado, el ángulo de doblado varía linealmente. Las zonas de los extremos vienen definidas por x_{\min} y x_{\max} , siendo la zona de doblado la definida por los valores de x tal que $x_{\min} \leq x \leq x_{\max}$. Al igual que en el caso anterior, la función de deformación debe aproximarse ya que no es polinómica. En concreto, los puntos de control de la estructura de deformación, correspondiente a la región de doblado, coinciden con los de la superficie reglada que conecta las aproximaciones de dos arcos de circunferencia concéntricos (Fig. 6.20b).

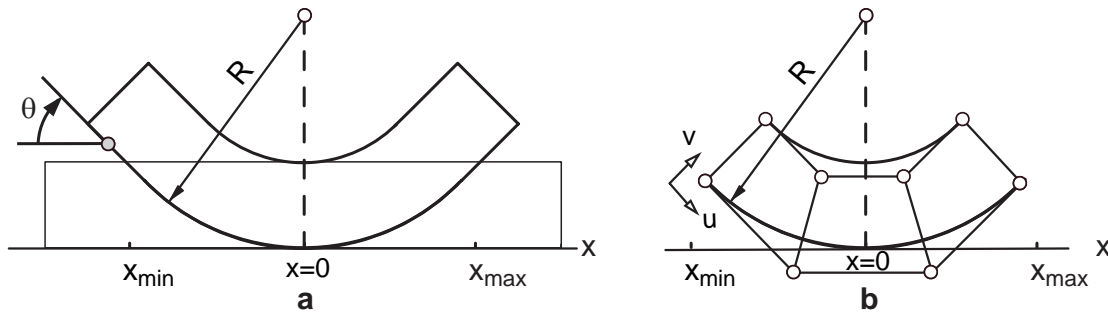


Figura 6.20: (a) Regiones de doblado de un objeto. (b) Aproximación de grado $(m_u, m_v) = (3, 1)$ de la región doblado.

La Fig. 6.21 muestra el doblado de un texto formado por curvas de Bézier cúbicas y completamente contenido dentro de la región de doblado. El resultado se ha aproximado mediante curvas de Bézier cúbicas. Como ejemplo final se han simulado la torsión y la flexión del ala de un avión definida por 2 superficies de Bézier de grado $(m_u, m_v) = (1, 3)$ (Fig. 6.22).

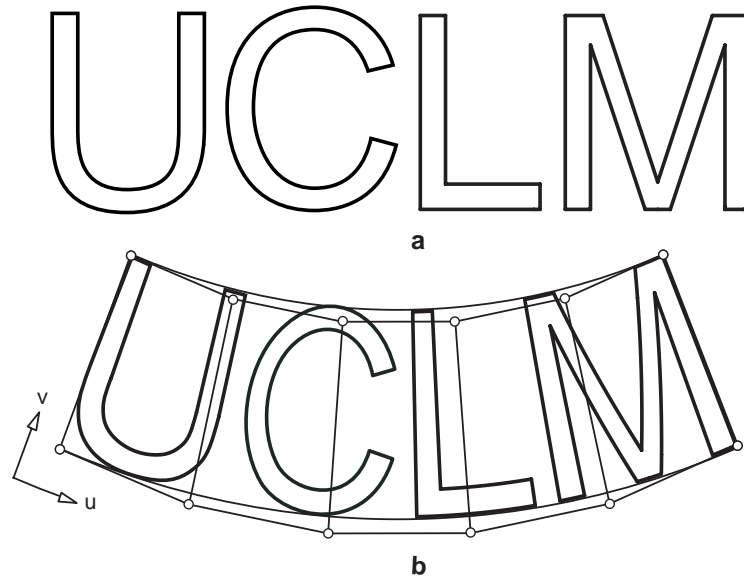


Figura 6.21: Doblado de texto en 2D. (a) Texto original formado por curvas cúbicas de Bézier. (b) Aproximación de grado $(m_u, m_v) = (5, 1)$ de la función de deformación y del texto deformado (grado $n = 3$).

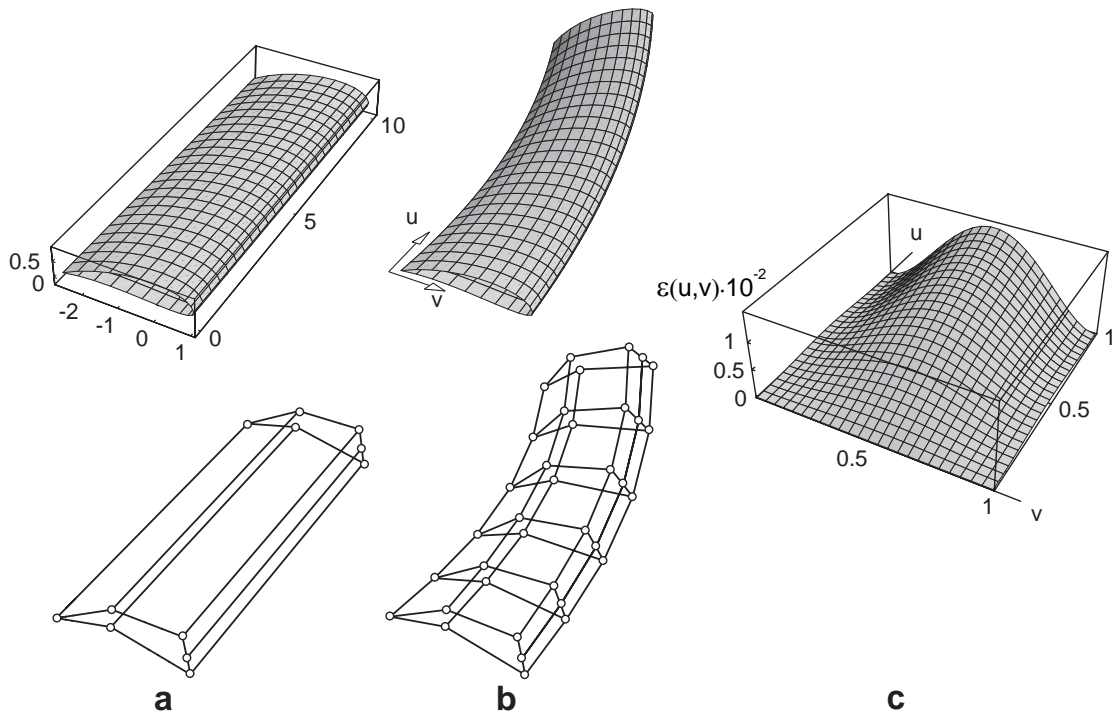


Figura 6.22: Torsión y doblado de dos superficies de Bézier. (a) Superficies originales de grado $(m_u, m_v) = (1, 3)$. (b) Aproximaciones de grado $(m_u, m_v) = (5, 3)$ de la superficie deformada. (c) Error $\epsilon(u, v)$.

Capítulo 7

Conclusiones, aportaciones y futura línea de investigación

En este Capítulo se destacan las conclusiones obtenidas en esta tesis y sus aportaciones más relevantes. Asimismo, se enumeran los artículos en que se ha basado este trabajo y las ponencias que ha originado. Finalmente se propone una línea de investigación como continuación del trabajo realizado.

7.1 Conclusiones y aportaciones

Obtención de un modelo de Bernstein-Bézier

En el caso de entidades geométricas de grado demasiado elevado o que no admiten una representación polinómica o racional, la única solución viable para poder incorporarlas en los programas de CAD/CAM consiste en obtener algún tipo de aproximación del resultado.

Después de estudiar las series de Taylor y de Poisson como estrategias de aproximación se ha llegado a las siguientes conclusiones:

- Una única expansión truncada de Taylor necesita un grado muy elevado para garantizar un error pequeño. Un grado elevado no es admisible en los programas de CAD/CAM y además implica una severa pérdida de precisión en la transformación a la base de Bernstein-Bézier.
- Si utilizamos una serie de Poisson truncada como técnica de aproximación, el resultado obtenido no es polinómico, por lo que no se puede incorporar en los programas de CAD/CAM, basados en el estándar de Bernstein-Bézier.

- La división del intervalo de definición de la entidad que se desea aproximar no resulta admisible. Al empalmar varias expansiones de Taylor o de Poisson de grado reducido, con el objeto de reducir el grado de estas aproximaciones o evitar una singularidad, aparecen huecos en las uniones, inadmisibles en una representación a trozos.

En lugar de series de Taylor o series de Poisson, la solución natural para obtener una aproximación polinómica consiste en el empleo de s-series, una modificación de las series de Newton que conduce a expansiones de Hermite en 2 puntos.

- Para el mismo grado, las aproximaciones mediante s-series son más exactas que las de Taylor y Poisson.
- El error de las aproximaciones mediante s-series, con forma de campana y nulo en los extremos, está distribuido de manera más uniforme sobre el intervalo de interés que en el caso de las series de Taylor y Poisson.
- Si se empalman aproximaciones de orden k (grado $2k + 1$) se obtiene un spline Hermítico que disfruta de continuidad C^k .
- La transformación a la base de Bernstein se puede realizar sin errores numéricos.

Para abordar con éxito la aproximación mediante s-series de entidades geométricas que no admiten una representación polinómica o racional se ha estudiado el proceso de subdivisión para aquellas funciones que poseen singularidades. Este proceso persigue evitar estas singularidades, generando aproximaciones convergentes. También se ha aplicado a los casos en que se desean calcular aproximaciones de grado reducido para valores del error pequeños, algo necesario ya que los programas de CAD/CAM tienen limitado el grado máximo admisible.

Calculada la aproximación mediante s-series de una entidad dada, se convierte a la base de Bernstein para su posterior incorporación en un programa de CAD/CAM. Finalmente se concatenan en una representación B-spline los tramos de Bézier en que se ha subdividido el objeto original. El resultado final es un spline Hermítico más adecuado que la representación de varios tramos de Bézier empalmados, ya que posee un menor número de puntos de control.

Los resultados corresponden al material contenido en:

- 2 Artículos en revistas internacionales (Sánchez-Reyes y Chacón 2003; 2005a).
- 1 Artículo en revista nacional (Sánchez-Reyes y Chacón 2003b).
- 3 Ponencias en congresos internacionales (Sánchez-Reyes y Chacón 2002; 2003c; Sánchez-Reyes, Chacón y Dorado 2003).

Deformación de curvas y superficies

Las técnicas de deformación libre se basan en la modificación de la forma de un objeto mediante la deformación de una estructura espacial en la que está contenido. Cuando el objeto deformado va a ser incorporado en un programa de CAD/CAM para su fabricación se necesita su expresión explícita. En este caso se debe calcular la composición polinómica del objeto que se desea deformar con la estructura espacial de deformación, por lo que el objeto resultante tras la deformación presenta un grado muy elevado. Se ha generado la composición vía convoluciones discretas del objeto con la estructura espacial en la base de Bernstein, pasando posteriormente a la base s-monomial para reducir el grado. Este método, aparte de utilizar matrices de cambio de base de dimensiones muy elevadas, introduce errores numéricos.

El método propuesto consiste en componer vía convoluciones discretas el objeto a deformar en la base s-monomial con la herramienta de deformación definida en la base de Bernstein, para así calcular fácilmente los coeficientes del objeto deformado en la base s-monomial. La ventaja de este método radica en la capacidad de reducir el grado de manera sencilla, controlando la complejidad de la representación así como el coste de cálculo de la deformación. Finalmente el objeto deformado de grado reducido se expresa en la base de Bernstein, operación que utiliza matrices de cambio de base de dimensiones reducidas. De esta forma el objeto deformado puede incorporarse en un programa de CAD/CAM.

Este método es aplicable a las técnicas FFD que utilizan estructuras espaciales de deformación prismáticas, representadas con polinomios de Bernstein. En principio no se puede aplicar a la técnica EFFD, pues genera objetos no polinómicos, ya que la función de deformación, como se ha expuesto en el Capítulo 6, involucra la inversa de una función polinómica de 3 variables. No obstante, la solución planteada puede emplearse generando previamente la aproximación de dicha función inversa vía s-series, si se lograra generalizar para funciones de 3 variables la técnica propuesta por Sánchez-Reyes (2001b).

El método presentado se ha aplicado a la técnica de deformación libre FFD. Sin embargo, como se ha descrito en el Capítulo 6, puede extenderse de manera muy sencilla a la técnica RFFD, generando una aproximación racional del objeto deformado.

La técnica expuesta en el Capítulo 6 se ha utilizado para modificar globalmente objetos formados por superficies de Bézier, pero se extiende de manera inmediata a objetos B-spline, simplemente dividiendo el objeto en sus tramos de Bézier componentes, y aplicando el método a cada tramo.

Finalmente se ha presentado un método para aproximar las técnicas de torsión (*twist*) y doblado (*bending*) de superficies. Estas operaciones son muy útiles para representar

objetos complejos, pero el resultado no es polinómico, por lo que no pueden expresarse en el estándar NURBS. Como resultado final se ha obtenido una representación polinómica del objeto deformado incorporable en los programas de CAD/CAM.

Los resultados obtenidos han dado lugar a 2 ponencias en congresos internacionales (Sánchez-Reyes y Chacón, 2004; 2005b).

7.2 Futura línea de investigación: Curvas y superficies offset (desplazadas)

Como futura línea de investigación se propone la aproximación del offset de curvas y superficies B-spline arbitrarias, problema de procesado de geometría que no ha sido resuelto satisfactoriamente. Este problema debe solucionarse mediante algoritmos apropiados para aplicaciones CAD. Por consiguiente, deben ser robustos, sencillos de implementar, y generadores de herramientas fáciles de utilizar e intuitivas para el diseñador (Piegl, 2005).

En general el offset de una curva polinómica o racional no admite una representación estándar, por lo que la utilización de técnicas de aproximación parece ser el único procedimiento factible para abordar la cuestión (Elber et al., 1997; Maekawa, 1999; Piegl y Tiller, 1999).

Lee et al. (1996) y Elber et al. (1997) presentan 5 métodos de aproximación del offset de curvas planas polinómicas y racionales, y los comparan con las técnicas más relevantes. La eficiencia de cada aproximación se mide a través del número de puntos de control necesarios para un valor del error máximo prefijado. Lee et al. (1996) y Elber et al. (1997) utilizan la representación de Bézier o NURBS (dependiendo de la naturaleza de la curva generadora y de la aproximación) del error funcional. Esta medida del error, propuesta por Elber y Cohen (1991), se define como el cuadrado de la distancia entre la aproximación del offset (desplazado una distancia d) y la curva generadora, menos d^2 .

Este método permite determinar un límite superior del error funcional, simplemente tomando el valor del coeficiente mayor de la representación de Bézier o NURBS del error. Asimismo, el problema de localización paramétrica del error máximo se reduce a calcular el máximo de una curva funcional de Bézier o NURBS. Este valor del parámetro indica por donde subdividir un cierto tramo cuya aproximación presenta un error mayor que el máximo prefijado

La representación del error expuesta se puede generar cuando se comparan dos entidades expresadas en el estándar NURBS, en este caso concreto la aproximación del offset y la curva generadora. Sin embargo, no se podrá hallar la representación de Bézier o

NURBS del error cuando la curva generadora sea trascendente. En estos casos se debe muestrear la función error para obtener su valor máximo.

Li y Hsu (1998) presentan una técnica basada en las series de Legendre para aproximaciones polinómicas del offset de una curva, pero como ya se ha explicado en el Capítulo 4, las series de Legendre no resultan adecuadas para aplicaciones CAD.

Piegl y Tiller (1999) proponen un método para aproximar el offset de curvas y superficies NURBS basado en una interpolación de puntos muestreados del offset exacto mediante una curva B-spline de grado n que exhibe continuidad C^{n-1} . Piegl y Tiller (1999) aplican el método propuesto a dos de los ejemplos utilizados por Lee et al. (1996), consiguiendo aproximaciones con un número de puntos de control menor.

Se propone generar un algoritmo para aproximar el offset de una curva B-spline arbitraria, basado en el empleo de s-series, que optimice el número de puntos de control y tramos en función del error máximo predefinido (Fig. 7.2). Como resultado se obtiene una curva B-spline de grado $n = 2k+1$ que presenta continuidad C^k . Obsérvese que, para conseguir la misma continuidad, este método emplea un grado mayor que el de Piegl y Tiller (1999). No obstante, cálculos preliminares (Chacón, 2005) muestran que aproximaciones basadas en el empleo de s-series, Fig. 7.3 y Fig. 7.4, precisan un número de puntos de control mucho menor que el obtenido por Piegl y Tiller (1999). Asimismo, la diferencia entre ambos métodos aumenta a medida que disminuye la cota del error máximo, como muestran las tablas 7.1 y 7.2. Este algoritmo también se podría aplicar para la aproximación de offset funcionales, de gran interés en el mecanizado de superficies con fresas cilíndricas o toroidales (Maekawa, 1999), donde la distancia d se reemplaza por una función arbitraria.

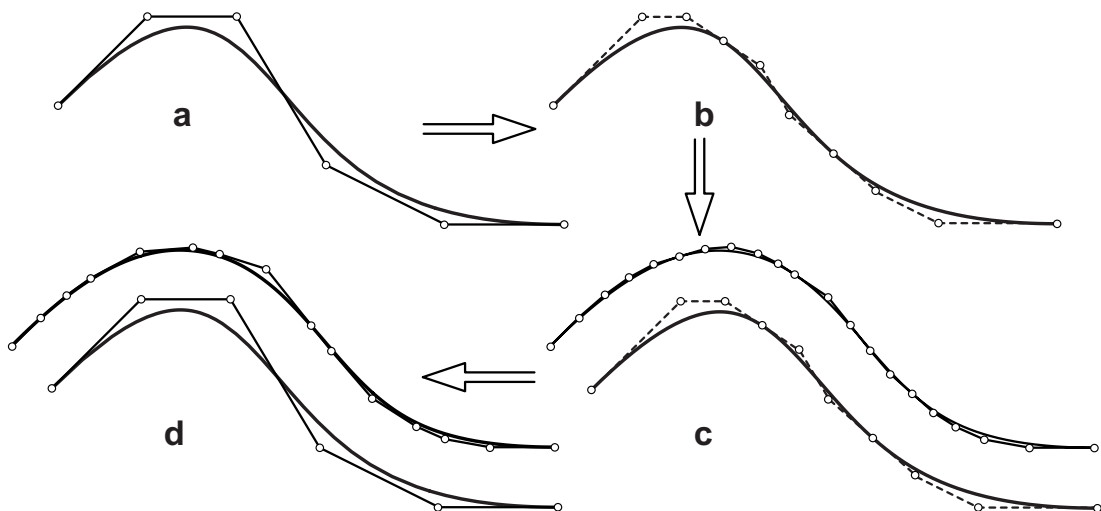


Figura 7.2: Aproximación (grado $n = 5$) del offset de una curva B-spline cúbica ($d = 0.2$, $\varepsilon = 10^{-3}$).

- (a) Curva B-spline generadora. (b) Subdivisión curva generadora en tramos de Bézier.
- (c) Representación de Bézier de la aproximación. (d) Representación B-spline.

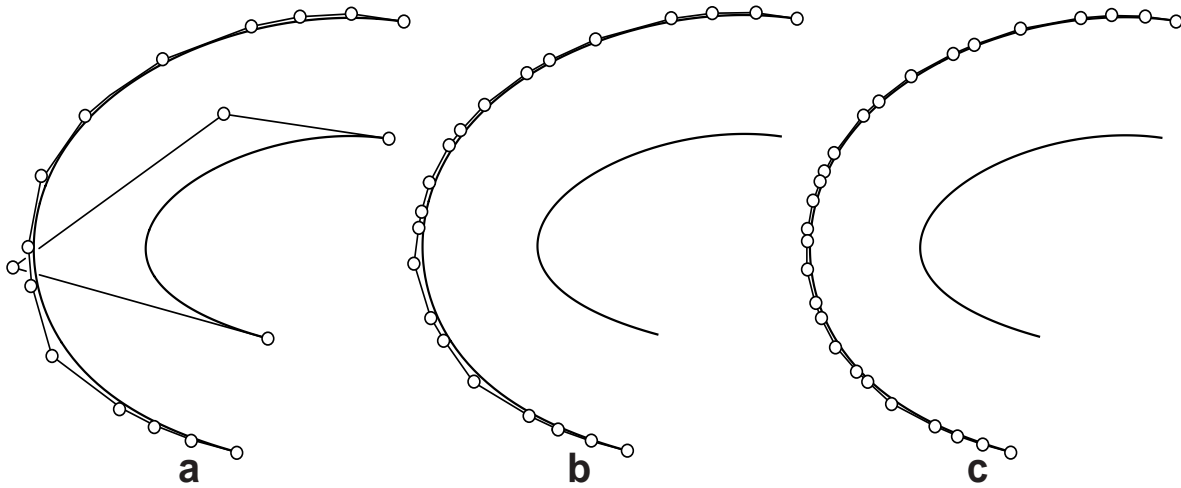


Figura 7.3: Aproximación de grado $n=5$ del offset de una curva de Bézier cúbica ($d=1$).

(a) Error = 10^{-3} . (b) Error = 10^{-4} . (c) Error = 10^{-5} .

Tabla 7.1: Comparación aproximación ejemplo Fig. 7.3.

ε	Lee et al. (1996) grado = 7	Piegl y Tiller (1999) grado = 3	s-series grado = 5
10^{-2}	29	9	9
10^{-3}	43	15	15
10^{-4}	71	25	18
10^{-5}	127	43	30

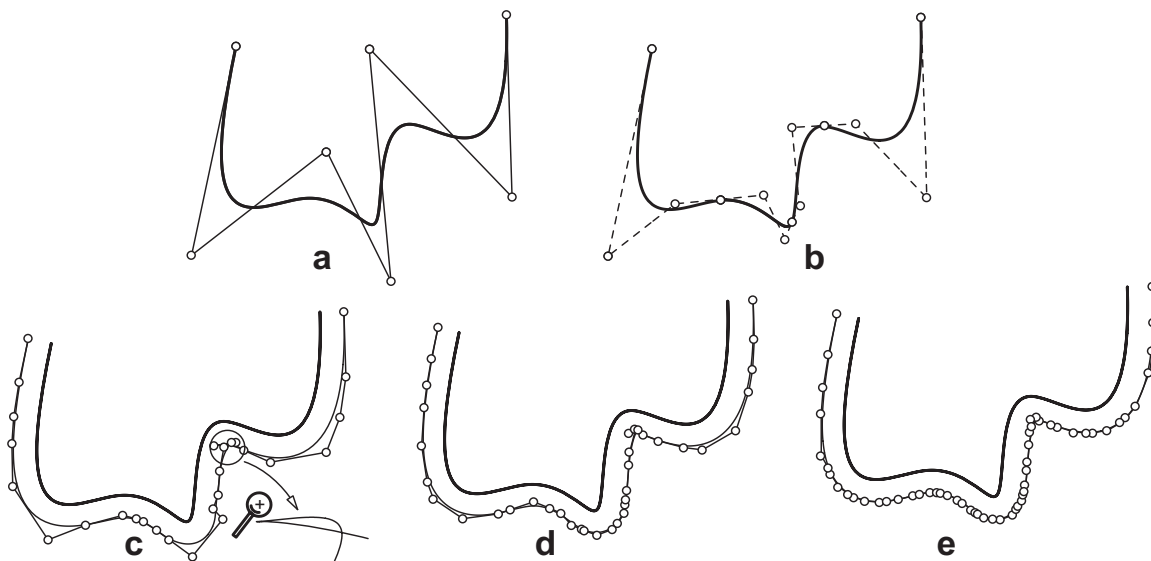


Figura 7.4: Aproximación del offset de curva B-spline cúbica ($d=0.5$). (a) Curva B-spline generadora. (b) División en tramos de Bézier. (c) Error = 10^{-3} . (d) Error = 10^{-4} . (e) Error = 10^{-5} .

Tabla 7.2: Comparación aproximación ejemplo Fig. 7.4.

ε	Lee et al. (1996) grado = 7	Piegl y Tiller (1999) grado = 3	s-series grado = 5
10^{-2}	92	33	27
10^{-3}	120	56	39
10^{-4}	176	101	60
10^{-5}	302	179	84

La mayoría de las técnicas publicadas para la aproximación de offset de una curva no se han aplicado a superficies (Piegl y Tiller, 1999). No obstante, Kulczycka y Nachman (2002) reproducen 5 métodos para aproximar el offset de superficies B-spline, dos de los cuales, basados en mínimos cuadrados y utilizados en curvas planas, han sido generalizados por Kulczycka y Nachman (2002) para el caso de superficies. La mayoría de los métodos aproximan el offset de cada uno de los tramos de Bezier que constituyen la superficie B-spline. Kulczycka y Nachman (2002) clasifican los trabajos reproducidos según el grado de continuidad que consiguen en la unión de estas aproximaciones. Únicamente 2 de ellos alcanzan continuidad C^2 . El primero, basado en mínimos cuadrados, aproxima la superficie B-spline entera en lugar de aproximar los tramos de Bézier que la componen. En el segundo caso la aproximación es la superficie cuya malla de control es el offset de la malla refinada de la superficie original. Esta técnica emplea un número de puntos de control elevado, ya que, con el fin de mejorar la calidad de la aproximación, precisa refinar la malla de la superficie original.

Se pretende generalizar el algoritmo propuesto anteriormente para el offset de curvas, basado en el empleo de s-series, para la aproximación del offset de superficies B-spline. Los resultados obtenidos se contrastarán con los obtenidos por Piegl y Tiller (1999) y Kulczycka y Nachman (2002).

Anexo A

Conversión al estándar de representación IGES

En los capítulos anteriores se ha estudiado un método, basado en el empleo de las series, para generar aproximaciones polinómicas de aquellas entidades geométricas que no admiten una representación polinómica, denominadas entidades trascendentes, o bien de objetos que presentan un grado demasiado elevado para poder ser incorporados en los programas de CAD/CAM. En ambos casos, las aproximaciones polinómicas resultantes pueden expresarse en el estándar NURBS.

Este Anexo describe cómo incorporar las aproximaciones en programas comerciales CAD/CAM. El método elegido consiste en generar el archivo IGES de estas aproximaciones, formato de archivo neutro importable por cualquier programa de CAD/CAM comercial.

A1 Archivo IGES

Intercambio de archivos IGES

Initial Graphics Exchange Specification, IGES, es un formato de archivo neutro de dominio público, estándar internacional ANSI para el intercambio de datos entre diferentes sistemas de CAD/CAM, soportado por la IGES/PDES Organization (IPO).

IGES permite representar muchos tipos de entidades, con una gama de complejidad que oscila desde algo tan sencillo como una línea hasta superficies complejas.

Estructura de un archivo IGES

La unidad de datos fundamental de un archivo IGES es la entidad. Hay dos categorías principales de entidades:

- Entidades Geométricas: definen formas poligonales, puntos, curvas, superficies, sólidos y relaciones.
- Entidades No geométricas: proporcionan atributos o características específicas para entidades geométricas. Las entidades no geométricas son vista, dibujo, nota general, cota, propiedad y entidades de asociatividad.

Cada entidad está representada por una entrada en la sección de entradas de datos y la sección de datos de parámetro del archivo IGES (Farin, 1997).

La Tabla A.1 contiene los tipos de entidades geométricas más importantes que pueden ser descritos mediante un archivo IGES.

Tabla A.1: Tipo de entidades de un archivo IGES.

<i>Entidades Geométricas</i>	
100. Arco circular	134. Nodo
108. Plano (limitado)	140. Superficie Offset
110. Línea(s)	142. Curva sobre superficie
112. Curva B-Spline paramétrica	154. Cilindro circular recto
114. Superficie B-spline paramétrica	156. Cono circular recto trucado
116. Punto	158. Esfera
120. Superficie de revolución	160. Toroide
126. Curva B-Spline racional	162. Sólido de revolución
128. Superficie B-Spline racional	164. Sólido de Extrusión
130. Curva offset	

A2 Organización del archivo IGES

Cada archivo IGES, dividido en 80 columnas, tiene cinco secciones identificadas por la letra de la columna 73 de cada línea (S, G, D, P o T) (Farin, 1997), secciones que se detallan a continuación:

Sección inicial

Comentario para lectura humana, generado manualmente por la persona que crea el archivo IGES, contiene información necesaria para el usuario receptor.

Sección de datos globales

Esta sección contiene información (la aplicación emisora, autor, organización, estándar de dibujo, sistema de medición, versión IGES y otros datos) necesarios para la aplicación receptora.

Sección de entrada de datos

La sección de entrada de datos sirve como índice para el archivo IGES. Posee una entrada de directorio para cada entidad. Una entrada de directorio tiene 20 campos justificados a la derecha de 8 caracteres cada uno, en dos líneas consecutivas que almacenan información aplicable a cada tipo de entidad. La Tabla A.2 muestra los campos más importantes de esta sección.

Tabla A.2: Campos de la Sección de Entrada de directorio de un archivo IGES.

1. Tipo de entidad
2. Puntero hacia los datos de parámetros para dicha entidad
4. Patrón de fuente de línea
5. Nivel
10. Número de líneas desde el comienzo de la sección de entrada del directorio
11. Tipo de entidad (igual que en el campo 1)
12. Grosor de línea
13. Color
14. Número de campos de la entrada de datos de parámetros
15. Número de formato
18. Etiqueta de entidad
20. Número de líneas desde el comienzo de la sección de entrada del directorio (igual que el campo 10)

Sección de datos de parámetro

Contiene información geométrica específica para cada tipo de entidad, como los puntos finales de una entidad de línea, o el centro y radio de un círculo. Las entradas de datos del parámetro tienen una longitud diferente dependiendo del tipo de elemento.

Sección final

Indica el final del archivo de datos IGES y sirve para chequeo del archivo. Cada línea tiene en la columna 73 y 74 una letra identificadora de la sección y el número de líneas desde el comienzo de la sección de entrada del directorio, usado por IGES para el puntero de datos en la sección de entrada de datos.

A3 Ejemplos de implementación vía IGES

Como ejemplo se ha generado el archivo IGES de la aproximación de la clotoide con subdivisión (Fig. 5.6). En dicho archivo (Fig. A.1) se indican las distintas secciones que lo constituyen, así como las partes más importantes que forman cada sección.

La Fig. A.2 muestra el archivo IGES de la representación B-spline procedente de la unión de las dos curvas de Bézier, después de la eliminación de nodos internos (Fig. 5.7). Como puede observarse, este archivo es mucho más compacto que el de dos curvas de Bézier empalmadas.

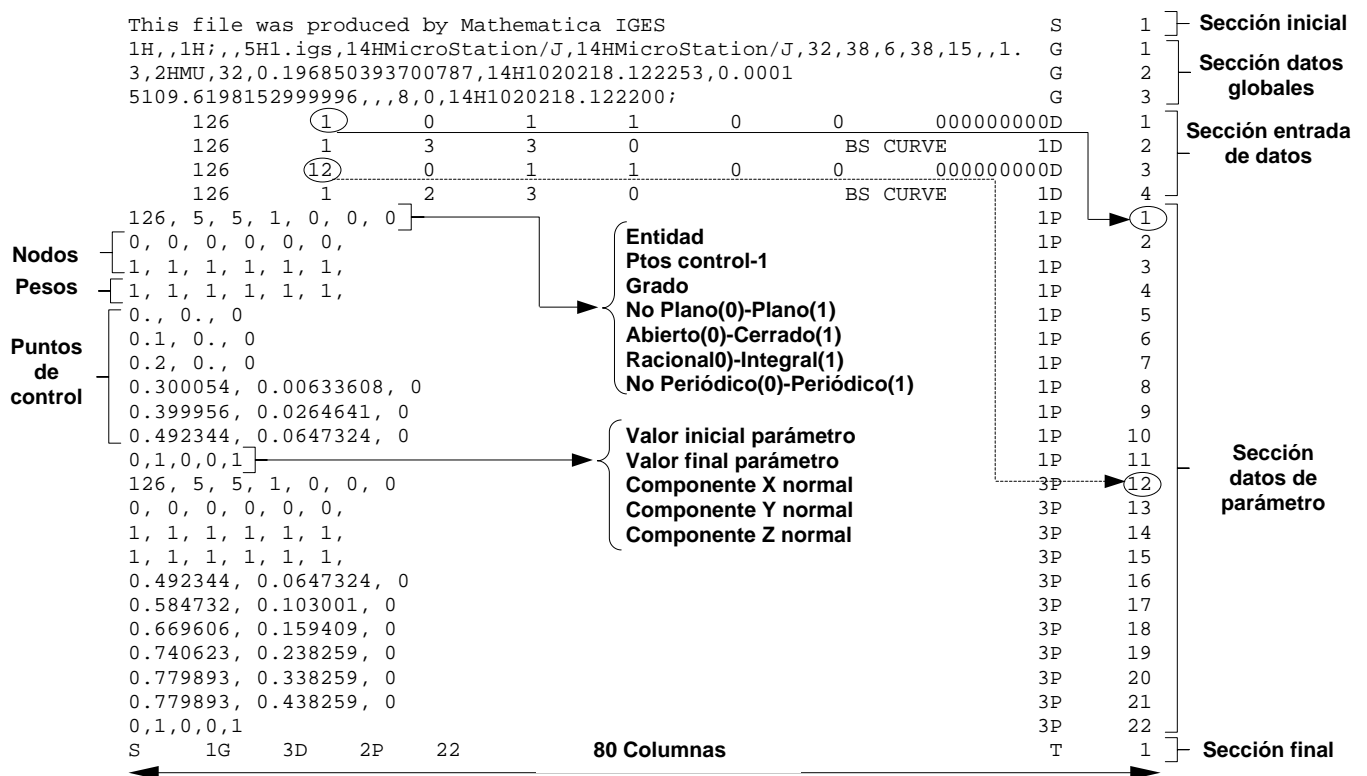


Figura A.1: Archivo IGES de 2 curvas de Bézier.

Este archivo se ha importado en el programa de CAD MicroStation® (Fig. A.3).

```

This file was produced by Mathematica IGES
1H,,1H;,,5H1.igs,14HMicroStation/J,14HMicroStation/J,32,38,6,38,15,,1.
3,2HMU,32,0.196850393700787,14H1020218.122253,0.0001
5109.6198152999996,,,8,0,14H1020218.122200;
126      1      0      1      1      0      0      00000000D
126      1      8      3      0      BS CURVE      1D
126, 8, 5, 1, 0, 0, 0
0, 0, 0, 0, 0, 0,
0.5, 0.5, 0.5,
1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1,
0., 0., 0
0.1, 0., 0
0.2, 0., 0
0.300054, 0.00633608, 0
0.499858, 0.0465921, 0
0.669606, 0.159409, 0
0.740623, 0.238259, 0
0.779893, 0.338259, 0
0.779893, 0.438259, 0
0,1,0,0,1
S      1G      3D      2P      15
S      1      1P      1
1      1P      2
0.5      1P      3
1      1P      4
1      1P      5
0.      1P      6
0.1      1P      7
0.2      1P      8
0.300054      1P      9
0.499858      1P      10
0.669606      1P      11
0.740623      1P      12
0.779893      1P      13
0.779893      1P      14
0,1,0,0,1      1P      15
S      1G      3D      2P      15      T      1

```

Figura A.2: Archivo IGES de una curva B-spline.

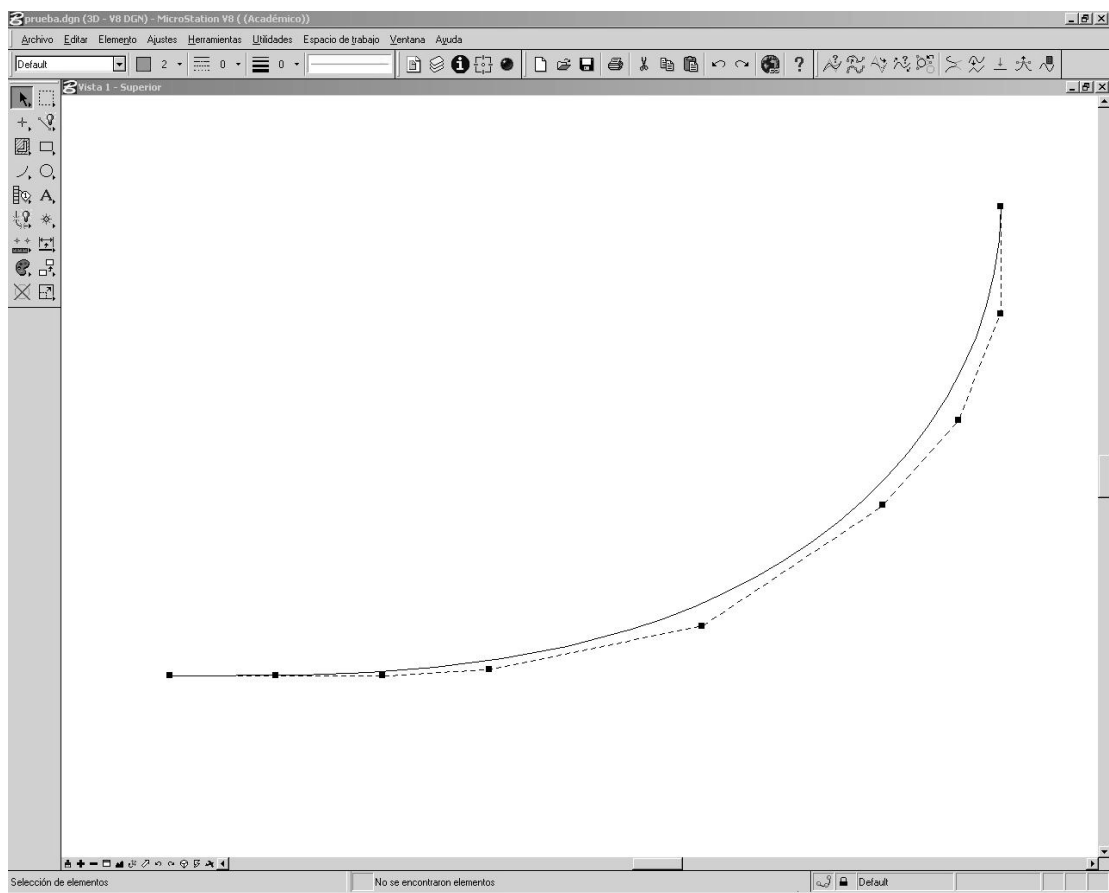


Figura A.3: Incorporación de curva B-spline en MicroStation®.

Anexo B

Notación

A	región
$\mathbf{a}(t)$	curva sin deformar
$\mathbf{a}_{\text{FFD}}(t)$	curva deformada
$a(u)$	polinomio univariado
$a(u, v)$	polinomio bivariado
$\mathbf{a}^0, \mathbf{a}^1$	vectores columna de coeficientes de una s-serie univariada
$\mathbf{a}^{00}, \mathbf{a}^{01}, \mathbf{a}^{10}, \mathbf{a}^{11}$	matrices de coeficientes de una s-serie bivariada
$a_k(u)$	coeficiente de una s-serie univariada, función lineal de u
$a_{i,j}(u, v)$	coeficiente de una s-serie bivariada, función lineal de u y v .
a_k^0, a_k^1	coeficientes de una s-serie univariada
$a_{i,j}^{0,0}, a_{i,j}^{1,0}, a_{i,j}^{0,1}, a_{i,j}^{1,1}$	coeficientes de una s-serie bivariada
\bar{a}_k	término constante de una s-serie univariada
Δa_k	término lineal de una s-serie univariada
$\bar{a}(s)$	componente simétrica de una s-serie univariada
\hat{a}_k^0, \hat{a}_k^1	coeficientes normalizados de una s-serie univariada
$\Delta a(s)$	componente antisimétrica de una s-serie univariada
$B_k^{2k+1}, B_{k+1}^{2k+1}$	polinomios de Bernstein de grado $2k+1$
$\tilde{\mathbf{b}}$	vector columna o matriz de coeficientes escalados de Bernstein
\tilde{b}_i	coeficientes escalados de Bernstein de una función univariada

$\tilde{b}_{i,j}$	coeficientes escalados de Bernstein de una función bivariada
\mathbf{b}	vector columna o matriz de coeficientes de Bernstein
b_i	coeficientes de Bernstein de una función univariada
$b_{i,j}$	coeficientes de Bernstein de una función bivariada
b_i^k	coeficientes de Bézier de la interpolación de Hermite de orden k
$\mathbf{b}(u, v)$	superficie sin deformar
$\mathbf{b}_{\text{FFD}}(u, v)$	superficie deformada
$b_k(t)$	funciones base de Poisson
$C(\mathbf{M})$	operador que invierte el orden de las columnas de una matriz \mathbf{M}
C	círculo de convergencia de una serie de Taylor
$C(t)$	integral de Fresnel
$\mathbf{c}(t)$	clotoide
$\mathbf{c}_d(t)$	offset clotoide
c_i^0	coeficientes serie de Taylor en 0
c_i^1	coeficientes serie de Taylor en 1
d	desplazamiento offset
$H_k(f, t)$	aproximación de Hermite de orden k de $f(t)$
\hat{h}_i^{k-1}	coeficientes de la interpolación de Hermite de orden $k - 1$ después de la elevación de grado $(2k - 1) \rightarrow (2k + 1)$
k	orden de una s-serie curvatura
L	lemniscata de Bernoulli
$l(t)$	longitud arco
\mathbf{M}^n	matriz de conversión de la base s-monomial a la base de Bernstein, dimensión $(n + 1) \times (k + 1)$
m	grado de una s-serie
\mathbf{N}^n	matriz de conversión de la base de Bernstein a la base s-monomial, dimensión $(k + 1) \times (n + 1)$
$\mathbf{n}(t)$	normal
n	grado de una s-serie

$P_k^0(u), P_k^1(u)$	funciones base de una s-serie univariada
p	orden de una s-serie bivariada
p_k	coeficientes de una serie de Poisson
q	orden de una s-serie bivariada
$R(\mathbf{M})$	operador que invierte el orden de los elementos de un vector o las filas de una matriz \mathbf{M}
r	radio de convergencia de una serie de Taylor multiplicidad original nodo interno
r'	multiplicidad mínima nodo interno
$S(t)$	integral de Fresnel
s	parámetro simétrico de una s-serie, polinomio función de u
t	parámetro simétrico de una s-serie, polinomio función de v variable
t^*	singularidad real
U	intervalo de determinación del número subdivisiones
u	variable unitaria nodo curva B-spline
v	variable unitaria
w	peso asociado a un punto de control
z	variable compleja
z^*	singularidad imaginaria
ε	error
$\alpha(t)$	ángulo entre la tangente a la clotoide y el eje x
ρ	factor de escala de una serie de Poisson
$\sigma(t)$	velocidad paramétrica

Referencias

- Barr, A. (1984). Global and local deformations of solid primitives. *Computer Graphics* 18 (3), 21-30.
- Barry, P.J., Goldman R.N. (1988). De Casteljau-type subdivision is peculiar to Bézier curves. *Computer-Aided Design* 20 (3), 114-116.
- Bechmann, D., Gerber, D. (2003). Arbitrary shaped deformations with DOGME. *The Visual Computer* 19 (2-3), 175-186.
- Bechmann, D. (1994). Space Deformation Models Survey. *Computers & Graphics* 18 (4), 571-586.
- Berezin, I.S., Zhidkov, N.P. (1965). *Computing Methods*. Pergamon.
- Boersma, J. (1960). Computation of Fresnel Integrals. *Math. Comp.* 14, 380.
- Borrel, P., Bechmann, D. (1991). Deformation of n-dimensional objects. *Journal Computational Geometry Apply* 1 (4), 427-453.
- Bowyer, A., Woodwark, J. (1993). *Introduction to Computing with Geometry*. Information Geometers.
- Carnicer, J.M., Peña, J.M. (1993). Shape preserving representations and optimality of the Bernstein basis. *Advances in Computational Mathematics* 1, 173-196.
- Cavendish, J. (1995). Integrating feature-based surface design with freeform deformation. *Computer-Aided Design* 27 (9), 703-711.
- Chacón, J.M. (2005). Polynomial approximation to offsets of NURBS curves and surfaces via s-power series. *SIAM Conference of Geometric Design & Computing*. Abstracts, pág. 32. Phoenix (EE.UU) 30 Octubre-3 Noviembre 2005.

- Chang, Y.K., Rockwood, A.P. (1994). A generalized de Casteljau Approach to 3D Free-form Deformation. *Computer Graphics Proceedings, Annual Conference Series*, 1994, 257-260. ACM.
- Cody, W.J. (1968). Chebyshev Approximations for the Fresnel Integrals. *Math. Comp.* 22, 450-453.
- Coquillart, S. (1990). Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling. *Computer Graphics* 24 (4), 187-196.
- Daniel, M., Daubisse, J.C. (1989). The numerical problem of using Bézier curves and surfaces in the power basis. *Computer Aided Geometric Design* 6 (2), 121-128.
- Davis, P.J. (1975). *Interpolation and Approximation*. Dover, New York.
- de Boor, C., Höllig, K., Sabin, M. (1987). High accuracy geometric Hermite interpolation. *Computer Aided Geometric Design* 4, 269-278.
- DeRose, T.D. (1988). Composing Bézier simplexes. *ACM Transactions on Graphics* 7 (3), 198-221.
- DeRose, T.D., Goldman, R.N., Hagen, H., Mann, S. (1993). Functional Composition Algorithms via Blossoming. *ACM Transactions on Graphics* 12 (2), 113-135.
- Elber, G., Cohen, E. (1991). Error bounded variable distance offset operator for free form curves and surfaces. *Journal Computational Geometry Apply* 1 (1), 67-78.
- Elber, G., Lee, I., Kim, M.-S. (1997). Comparing Offset Curve Approximation Methods. *IEEE Computer Graphics and Applications* 17 (3), 62-71.
- Farin, G. (1997). *NURBS Curves and surfaces from projective geometry to practical use* (2nd Ed.). A K Peters.
- Farin, G. (2002). *Curves and surfaces for Computer Aided Geometric Design* (5th Ed.). Morgan Kaufmann.
- Farin, G., Hansford, D. (2000). *The Essentials of CAGD*. AK Peters.
- Farouki, R.T. (1991). On the stability of transformations between power and Bernstein polynomial forms. *Computer Aided Geometric Design* 8 (1), 29-36.
- Farouki, R.T. (2000). Convergent inversion approximations for polynomials in Bernstein form. *Computer Aided Geometric Design* 17 (2), 179-196.

- Farouki, R.T., Rajan, V.T. (1988). On the numerical condition of polynomials in Bernstein form. *Computer Aided Geometric Design* 4 (3), 191-216.
- Farouki, R.T., Goodman T.N.T. (1996). On the optimal stability of the Bernstein basis. *Mathematics of Computation* 65, 1553-1566.
- Feng, J., Nishita, T., Jin, X., Peng, Q. (2002). B-spline free-form deformation of polygonal object as trimmed Bézier surfaces. *The Visual Computer* 18, 493-510.
- Field, D.A. (2005). Changes in CAGD development supporting automotive industrial design (styling) during the past 25 years (Industrial perspective). SIAM Conference of Geometric Design & Computing. Phoenix (EE.UU) 30 octubre-3 noviembre 2005.
- Fine, M. (1961). A Note on Two-point Taylor's series for solving ordinary differential Equations. *Journal of the Aerospace Sciences* 28, 671-672.
- Forrest, A.R. (1972). Interactive Interpolation and Approximation by Bézier curves. *Computer Journal* 15, 71-79.
- Goldman, R.N. (2003). *Pyramid Algorithms, A dynamic Approach to Curves and Surfaces for Geometric Modelling*. Morgan Kaufmann.
- Gomes Teixeira, F. (1909). *Traité des Courbes Spéciales, Remarquables Planes et Gauches, Tome II*. Reprinted by Éditions Jacques Gabay.
- Griessmair J., Purgathofer W. (1989). Deformation of solids with trivariate B-spline. *Proceedings of Eurographics'89*, 137-148.
- Grisoni, L., Blanc, C., Schlick, C. (1999). Hermitian B-splines. *Computer Graphics Forum* 18 (4), 237-248.
- Heald, M.A. (1985). Rational approximations for the Fresnel Integrals. *Mathematics of Computation* 44 (170), 459-461.
- Hermann, T. (2002). Degree elevation for generalized Poisson functions. *Computer Aided Geometric Design* 19 (1), 67-70.
- Hirota, G., Maheshwari, R., Lin, M. (1999). Fast Volume-Preserving Free Form Deformation Using Multi-Level Optimization. *Fifth Symposium on Solid Modeling 1999*, 234-245. ACM
- Hoschek, J., Lasser, D. (1993). *Fundamentals of Computer Aided Geometric Design*. AK Peters, Wellesley, MA, USA.

- Hu, S.-M., Zhang, H., Tai, C.-L., Sun, J.-G. (2001). Direct manipulation of FFD: efficient explicit solutions and decomposable multiple point constraints. *The Visual Computer* 17, 370-379.
- Hui, K. (2002). Free-form design using axial curve-pairs. *Computer-Aided Design* 34, 583-595.
- Hui, K. (2003). Free-form deformation of constructive shell models. *Computer-Aided Design* 35, 1221-1234.
- Jin, X., Li, Y.F. (2000). Three-Dimensional Deformation Using Directional Polar Coordinates. *Journal of Graphics Tools* 5 (2), 15-24.
- Kanayama, Y., Miyake, N. (1985). Trajectory Generation for Mobile Robots. *Proceedings of 3rd International Symposium of Robotic Research* 16-23, Gouvieux (FR).
- Kasik, D.J., Buxton, W., Ferguson, D.R. (2005). Ten CAD challenges. *IEEE Computer Graphics and Applications*, 81-92.
- Klok, F. (1986). Two moving coordinate frames for sweeping along a 3D trajectory. *Computer Aided Geometric Design* 3, 217-229.
- Knuth, D.E. (1998). *The Art of Computer Programming, Volume 2: Seminumerical Algorithms* (3rd Ed.). Addison-Wesley.
- Kulczycka, M.A., Nachman, L.J. (2002). Qualitative and quantitative comparisons of B-spline offset surface approximation methods. *Computer-Aided Design* 34, 19-26.
- Lamousin, H.J., Waggenspack, W.N. (1994). NURBS-Based Free-Form Deformations. *IEEE Computer Graphics and Applications*, 59-65.
- Lawrence, J.D. (1972). *A catalog of special plane curves*. Dover, New York.
- Lazarus, F., Coquillart, S., Jancène, P. (1994). Axial deformations: an intuitive deformation technique. *Computer-Aided Design* 26 (8), 607-613.
- Lee, I., Kim, M.-S., Elber, G. (1996). Planar curve offset based on circle approximation. *Computer-Aided Design* 28 (8), 617-630.
- Lee, B.-G., Park, Y., Yoo, J. (2002). Application of Legendre-Bernstein basis transformations to degree elevation and degree reduction. *Computer Aided Geometric Design* 19 (9), 709-718.

- Li, Y.M., Hsu, V.Y. (1998). Curve offsetting based on Legendre series. *Computer Aided Geometric Design* 15 (7), 711-720.
- Liu, W., Mann, S. (1997). An optimal algorithm for expanding the composition of polynomials. *ACM Transactions on Graphics*, 16 (2), 155-178.
- Lodha, S., Warren, J. (1994). Degree reduction of Bézier simplexes. *Computer-Aided Design* 26 (10), 735-746.
- López, J., Temme, N.M. (2002). Two-Point Taylor Expansions of Analytic Functions. *Studies in Applied Mathematics* 109 (4), 297-311.
- Maekawa, T. (1999). An overview of offset curves and surfaces. *Computer-Aided Design* 31, 165-173.
- Mainar, E., Peña, J.M. (1999). Corner cutting algorithms associated with optimal shape preserving representations. *Computer Aided Geometric Design* 16 (9), 883-906.
- McNeel, R. (2004). Universal Deformation™ technology announced. <http://www.rhino3d.com/brazil/udt-pr1.pdf>.
- Meek, D.S., Thomas, R.S.D. (1991). A guided clothoid spline. *Computer Aided Geometric Design* 8 (2), 163-174.
- Menon J.P. (1994). Constructive shell representations for freeform surfaces and solids. *IEEE Computer Graphics and Applications* 14 (2), 24-36.
- Morin, G., Goldman, R. (2000a). A subdivision scheme for Poisson curves and surfaces. *Computer Aided Geometric Design* 17 (9), 813-833.
- Morin, G., Goldman, R. (2000b). Poisson Approximation. *Geometric Modeling and Processing 2000*, April 10-12, Hong Kong, IEEE Computer Society, 141-149.
- Morin, G., Goldman, R. (2001a). Trimming analytic functions using right sided Poisson subdivision. *Computer-Aided Design* 33 (11), 813-824.
- Morin, G., Goldman, R. (2001b). The Analytic Blossom. En Lyche, T., Schumaker LL., (eds), *Mathematical Methods for Curves and Surfaces*, Oslo, Vanderbilt University Press, Nashville, TN, 325-346.
- Needham, T. (1997). *Visual Complex Analysis*. Clarendon Press, Oxford.
- Patterson, R.R. (2005). Composition of parametrizations, using the paired algebras of forms and sites. Aceptado para su publicación en *Computer Aided Geometric Design*.

- Peng, Q., Jin, X., Feng, J. (1997). Arc-Length-Based Axial Deformation and Length Preserved Animation. *Proceedings of the IEEE Computer Animation*, 86-92.
- Peña, J.M. (1999). Stability and error analysis of shape preserving representations. En: Peña J.M., (eds), *Shape preserving representations in Computer-Aided Geometric Design*, Nova Science, NewYork, 85-97.
- Piegl, L. (1993). *Fundamental Developments of Computer-Aided Geometric Modeling*. Academic Press, San Diego, CA.
- Piegl, L. (2005). Ten challenges in computer-aided design. *Computer-Aided Design* 37, 461-470.
- Piegl, L., Tiller, W. (1995). Algorithm for degree reduction of B-spline curves. *Computer-Aided Design* 27 (2), 101-110.
- Piegl, L., Tiller, W. (1997). *The NURBS Book* (2nd Ed.), Springer.
- Piegl, L., Tiller, W. (1999). Computing offsets of NURB curves and surfaces. *Computer-Aided Design* 31, 147-156.
- Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery B.P. (1992). *Numerical recipes in C. The art of scientific computing* (2nd Ed.), Cambridge University Press.
- Renner, G., Weiß, V. (2004). Exact and approximate computation of B-spline curves on surfaces. *Computer-Aided Design* 36, 351-362.
- Rockwood, A., Chambers, P. (1996). *Interactive Curves and Surfaces: A multimedia tutorial on CAGD*. AK Peters, San Francisco, USA.
- Rogers, D.F. (2001). *An Introduction to NURBS: With Historical perspective*. Morgan Kaufmann.
- Sabin, M. (1990). Sculptured Surface Definitions—A Historical Survey. En Rogers, F., Earnshaw, R., (eds), *Computer Graphics Techniques*, Springer, New York, 285-337.
- Sánchez-Reyes, J. (1997). The Symmetric Analogue of the Polynomial Power Basis. *ACM Transactions on Graphics* 16 (3) (July), 319-357.
- Sánchez-Reyes, J. (2000). Applications of the s-Power Basis in Geometry Processing. *ACM Transactions on Graphics* 9 (1), 27-55.
- Sánchez-Reyes, J. (2001a). *Proyecto de Investigación*. Plaza CU-1/00. Universidad de Castilla-La Mancha.

- Sánchez-Reyes, J. (2001b). Inversion approximations for functions via s-power series. *Computer Aided Geometric Design* 18 (6), 587-608.
- Sánchez-Reyes, J. (2003a). Algebraic manipulation in the Bernstein form made simple via convolutions. *Computer-Aided Design* 35, 959-967.
- Sánchez-Reyes, J. (2003b). Two-point Taylor expansions are Newton series. Submitted to *Studies in Applied Mathematics*.
- Sánchez-Reyes, J., Arranz, J.C. (1998). Bases de potencias simétricas para deformación libre de curvas y superficies polinómicas. *Proceedings del CEIG '98, 7º Congreso Español de Informática Gráfica*, 281-294. Ourense 17-19 junio 1998.
- Sánchez-Reyes, J., Chacón, J.M. (2002). Aproximación polinómica de espirales clotoideas mediante s-series. *Proceedings del XIII Congreso Internacional de Expresión Gráfica en la Ingeniería*, 74-83. Santander 5-7 junio 2002.
- Sánchez-Reyes, J., Chacón, J.M. (2003a). Polynomial approximation to clothoids via s-power series. *Computer-Aided Design* 35, 1305-1313.
- Sánchez-Reyes, J., Chacón, J.M. (2003b). Aproximación polinómica de espirales clotoideas mediante s-series. *Anales de Ingeniería Gráfica* 16, 52-59.
- Sánchez-Reyes, J., Chacón, J.M. (2003c). s-power series: an alternative to Poisson series for geometry processing in CAD. *Proceedings del XIII ADM-XV INGEGRAF Congreso Internacional sobre herramientas y métodos en diseño de ingeniería*, CD-ROM. Nápoles (Italia) 4-6 junio 2003.
- Sánchez-Reyes, J., Chacón, J.M. (2004). Hermite Approximation of Deformed curves and Surfaces. 6th International Conference on Mathematical Methods for Curves and Surfaces. Abstracts, pág. 29. Tromsø (Noruega) 1-6 julio 2004.
- Sánchez-Reyes, J., Chacón, J.M. (2005a). s-power series: an alternative to Poisson series for manipulating analytic functions. *Computer Aided Geometric Design* 22 (2), 103-119.
- Sánchez-Reyes, J., Chacón, J.M. (2005b). Aproximación de Hermite para la deformación libre de curvas y superficies. *Proceedings del Congreso Internacional Conjunto XVII Ingegraf*, CD-ROM. Sevilla 1-3 junio 2005.
- Sánchez-Reyes, J., Chacón, J.M., Dorado, R. (2003). s-power series: an alternative to Poisson series for describing analytic functions. *SIAM Conference of Geometric Design & Computing*. Abstracts, pág. 58. Seattle (EE.UU) 10-13 septiembre 2003.

- Sarraga, R. (2003). Modifying CAD/CAM surfaces according to displacements prescribed at a finite set of points. *Computer-Aided Design* 36, 343-349.
- Scheuer, A., Fraichard, T. (1997). Collision-free and continuous-curvature path planning for car-like robots. *Proceedings of the IEEE International Conference on Robotics and Automation* 1, 867-873, Albuquerque NM (US).
- Sederberg, T.W., Parry, S.R. (1986). Free-form deformation of solid geometric models. *Siggraph'86. Computer Graphics* 20 (4), 151-196.
- Stoer, J., Burlirsch, R. (1993). *Introduction to Numerical Analysis*. Springer, New York, USA.
- Sunwoo, H. (2005). Matrix Representation for multi-degree reduction of Bézier curves. *Computer Aided Geometric Design* 22, 261-273.
- Sunwoo, H., Lee, N. (2004). A unified matrix representation for degree reduction of Bézier curves. *Computer Aided Geometric Design* 21 (2), 151-164.
- Vogtner, G.W. (1996). 3D modeling: Harder than it looks. *Computer Graphics World* 19 (8), 7.
- Vries-Baayens, A.E. (1991). Conversión of a composite trimmed Bézier surface into composite Bézier surfaces. En Laurent, P.J., Le Méhauté, A. y Schumaker, L.L., (eds), *Curves and Surfaces*, Academic Press, Boston, 485-489.
- Walton, D.J., Meek, D.S. (1990). Clothoidal splines. *Computers&Graphics* 14 (1), 95-100.
- Wang, L.Z., Miura, K., Nakamae, E., Yamamoto, T., Wang, T.J. (2001). An approximation approach of the clothoid curve defined in the interval $[0, \pi/2]$ and its offset by free-form curves. *Computer-Aided Design* 33 (14), 1049-1058.

Índice de materias

Adobe illustrator, 61,62,76

afín

coordenadas, 71

transformación, 6, 8

algoritmo

eliminación de nodos, 32, 55

inserción de nodos, 32

reducción de grado, 6, 7, 13, 17

subdivisión, 35, 38

aproximación lineal, 38, 40

área de convergencia, 26, 39

Arranz, J.C., 5, 64, 65, 104

AxDf, 59

Barr, A., 63, 77, 78, 98

base s-monomial, 7, 8, 12, 16, 20, 29, 30,
31, 32, 57, 65

b-Base, 6, 35

Bechmann, D., 66, 67, 68, 69, 70, 71, 73,
75, 82, 95

bending, 4, 57, 76, 84

Berezin, IS., 11, 99

Bernstein

base, 6, 7, 8, 11, 12, 13, 14, 15, 17, 23,

29, 31, 35, 38, 49, 57

coeficientes, 29, 30, 32, 96

coeficientes escalados, 29, 30, 31, 95,
96

modelo, 2, 81

polinomio, 17, 29, 36, 63, 83, 95

representación, 1, 6, 31

Bézier

curva, 60, 64, 71, 73, 86

ordenadas, 12, 13

superficie, 67, 69, 72, 73, 74, 75, 76,
77

volumen, 67, 75

Blanc, C., 101

Blossom, 65, 69

Boersma, J., 49, 99

Borrel, P., 60, 63, 99

Bowyer, A., 6, 99

Buxton, W., 102

B-spline

curva, 32, 33, 55, 56, 69, 85, 87, 90,
93, 97

estructuras trivariadas, 58, 72

FFD, 58, 65, 66, 72

representación, 8, 32, 33, 34, 50, 82,
85, 92

- superficie, 69, 87, 90
- Burlirsch, R., 11, 105
- CAE, 7
- CAGD, 1, 2, 3, 6, 7, 11, 58
- Carnicer, 6, 99
- Cassini, 25, 26
- Casteljau, 32, 33, 35, 38, 60, 99
- Catenaria, 3, 46
- Catenoide, 46
- Cavendish, J., 58, 59, 99
- círculo, 24, 25, 39, 40, 44, 47, 74, 91, 96
- Clotoide, 9, 47, 48, 49, 50, 51
- Cody, WJ., 49, 100
- Cohen, E., 83, 100
- combinación convexa, 6
- continuidad
 - geométrica, 51
 - grado, 7, 87
 - posicional, 62
- convex hull, 35, 37, 40
- convoluciones, 15, 21, 65, 66, 67, 83
- Coons, 62, 72, 76
- Coquillart, S., 58, 100, 102
- CSR, 59
- curva,
 - deformada, 59, 68, 69, 71, 73, 95
 - plana, 47
 - polinómica, 84
 - racional, 71
 - sobre superficie, 74, 90
 - transición, 47
- trascendente, 3, 9
- curvatura, 47, 48, 78, 96
- Chacón, J.M., 9, 45, 82, 84, 85, 99, 105
- Chambers, P., 1, 103
- Chang, Y.K., 60, 63, 64, 82, 88, 98, 100
- Daniel, M., 49, 100
- Daubisse, JC., 49, 100
- Davis, PJ., 11, 24, 35, 100
- de Boor, C., 50, 100
- deformación
 - curvas, 68
 - directa, 60, 63
 - estructura, 59, 60, 61, 62, 77, 78
 - libre, 9, 57, 58, 59, 66, 71, 75, 83, 105
 - superficies, 68
- DeRose, T.D., 38, 65, 100
- DFFD, 60, 63
- diferencias divididas, 14
- distorsión, 61
- Dorado, R., 81, 104
- EFFD, 58, 63, 64, 66, 83
- Elber, G., 84, 100, 102
- error Pitagórico, 43, 53, 54, 55, 56
- escalado, 4, 12, 29, 30, 31, 60, 95, 96
- espiral
 - Arquímedes, 43, 44
 - Clotoide, 47, 48
 - Cornut, 47
 - Euler, 47

- factor de escala, 37, 38, 40, 60, 97
- Farin, G., 1, 6, 30, 35, 58, 90, 100
- Farouki, R.T., 6, 29, 36, 49, 100
- Feng, J., 65, 72, 103
- Ferguson, D.R., 102
- Field, D. A., 61, 101
- Fine, M., 15, 101
- Forrest, A. R., 6, 7, 13, 17
- Fraichard, T., 47, 105
- Frenet, 59
- Fresnel, 9, 48, 49, 50, 51, 52, 53, 96, 97, 99, 100
- función
- analítica, 24, 35, 36, 39, 40, 50
 - trascendente, 3, 6, 9, 11, 15, 35, 48, 85, 89
- Gerber, D., 60, 63, 99
- Goodman, T.N.T., 6, 101
- Goldman, R., 14, 36, 39, 40, 41, 103
- Gomes Teixeira, F., 47, 101
- Griessmair J., 58, 72, 101
- Grisoni, L., 23, 32, 101
- Hagen, H., 100
- Hansford, D., 1, 100
- Heald, M.A., 49, 101
- Hermann, T., 38, 101
- Hirota, G., 60, 101
- Horner, 15
- Hoschek, J., 1, 48, 101
- Hsu, V.Y., 36, 85, 102
- Hu, S., 60, 100
- Hui, K., 50, 60, 63, 102
- IGES, 9, 89, 90, 91, 92, 93
- interpolación, 17, 24, 48, 50, 54, 65, 85, 96
- intervalo unitario, 8, 26, 58
- Jancène, P., 102
- Jin, X., 60, 63, 101, 102, 103
- Kanayama, Y., 47, 102
- Kasik, F.J., 7, 102
- Kim, M.-S., 100, 102
- Klok, F., 59, 102
- Knuth, D.E., 15, 102
- Kulczycka, M.A.,
- Lamousin, H.J., 58, 102
- Lasser, D., 1, 48, 101
- Lawrence, J.D., 35, 47, 102
- Lazarus, F., 4, 59, 60, 63, 102
- Legendre, 36, 85, 102
- Lee, B.-G., 36, 102, 103
- Lee, N., 6, 36, 84, 85, 86, 100, 102
- lemniscata, 25, 26, 27, 28, 40, 41, 42, 96
- Li, Y.F., 60, 63, 102
- Li, Y.M., 36, 85, 102

- Liu, W., 65, 102
 Lodha, S., 6, 103
 longitud arco, 47, 51, 96
 López, J., 15, 103

 Maekawa, T., 84, 86, 103
 Mainar, E., 84, 86, 103
 Mann, S., 65, 102
 McNeel, R., 61, 103
 Meek, D.S., 47, 103, 106
 Menon J.P., 59, 103
 MicroStation, 93
 Miura, K., 106
 Miyake, N., 47, 102
 modelado, 1, 4, 5
 Morin, G., 36, 38, 39, 40, 41, 103

 Nachman, L.J., 87, 88, 102
 Nakamae, E., 106
 Needham, T., 38, 103

 offset
 curva, 36, 84, 85, 86, 87, 90
 superficie, 87, 88, 90
 operador, 29, 30, 96, 97
 óvalo, 25

 panel, 61
 parametrización, 9, 38, 44, 45, 47, 48, 50, 51, 72

 Park, Y., 102
 Parry, S.R., 58, 106
 Patterson, R.R., 65, 103
 Peng, Q., 59, 63, 103
 Peña, J.M., 6, 35, 99, 103
 Piegl, L., 1, 6, 7, 32, 35, 84, 85, 86, 87, 88, 103, 104
 Poisson
 coeficientes, 36
 serie, 39, 40, 41, 42, 43, 45, 46, 81, 97
 Press, W.H., 49, 104
 Purgathofer W., 58, 72, 101

 Rajan, V.T., 6, 29, 100
 Rango, 52
 Renner, G., 69, 104
 RFFD, 58, 65, 66, 71, 83
 Rhino, 61, 103
 Rockwood, A.P., 1, 60, 63, 64, 83, 99, 104
 Rogers, D.F., 3, 104
 Rotación, 59, 78

 Sabin, M., 1, 100, 104
 Sánchez-Reyes, J., 3, 5, 6, 7, 8, 9, 12, 13, 14, 15, 16, 21, 24, 29, 31, 45, 65, 66, 82, 83, 84, 104, 105
 Sarraga, R., 61, 105
 Scheuer, A., 47, 105
 Sederberg, T.W., 58, 106
 semicírculo, 44

- series de Taylor, 6, 14, 24, 26, 96, 97
 - coeficientes, 14
- simplexes, 65, 100, 103
- spline Hermítico, 23, 32, 33, 47, 50, 54, 82
- springback, 61
- Stoer, J., 11, 106
- Sunwoo, H., 6, 106
- superficie
 - bivariada, 45
 - esculpida, 1
 - implícitas, 59
 - plana, 68, 72
 - racional, 71
 - recortada, 69, 72, 73
 - revolución, 46, 90
- Temme, N.M., 15, 103
- Teukolsky, S.A., 104
- Thomas, R.S.D., 47, 103
- Tiller, W., 6, 32, 35, 84, 85, 86, 87, 88, 104
- torsion, 9, 57, 60, 61, 63, 76, 77, 79, 84
- trimming curve, 69, 74
- twist, 57, 76, 84
- UDT, 61
- variación disminuida, 35, 37, 40
- velocidad paramétrica, 45, 50, 54, 97
- Vetterling, W.T., 104
- Vogtner, G.W., 4, 106
- Vries-Baayens, 72, 106
- Waggenpack, W.N., 58, 102
- Walton, D.J., 47, 106
- Wang, L.Z., 47, 49, 50, 52, 106
- Wang, T.J., 106
- Warren, J., 6, 103
- Weiß, V., 69, 104
- Woodwark, J., 6, 99
- Yamamoto, T., 106
- Yoo, J., 102
- Zhidkov, NP., 11, 99