



Representación de superficies suaves a través de superficies NURBS como herramienta de modelado geométrico en sistemas CAD

Mykholl Alexander Guamanga Chilito

Universidad del Valle
Facultad de Ciencias Naturales y Exactas
Cali, Colombia
2019

Representación de superficies suaves a través de superficies NURBS como herramienta de modelado geométrico en sistemas CAGD

Mykholl Alexander Guamanga Chilito

Trabajo de grado presentado como requisito parcial para optar al título de:
Matemático

Director:
Dr. Jhon Jairo Duque Robles

Línea de Investigación:
Análisis numérico

Universidad del Valle
Facultad de Ciencias Naturales y Exactas
Cali, Colombia
2019

*A mis padres y mi hermano, por haberme
permitido llegar hasta aquí...*

*a mi familia, amigos y compañeros, por
estar siempre a mi lado...*

*a mi sensei, por su paciencia y valiosas
enseñanzas...*

Gracias a todos.

Resumen

En este trabajo se presentan los resultados fundamentales de la teoría de interpolación, con especial énfasis en las funciones B-Splines y las superficies NURBS. El propósito de este trabajo es mostrar como el desarrollo matemático de esta teoría tiene su mejor exponente en las herramientas modernas de Diseño Geométrico Asistido por Computador (CAGD); para esto resaltamos la conexión que existe entre el resultado de convergencia de la interpolación spline y el desarrollo de herramientas para entorno de trabajo en CAGD. Finalmente, construimos una figura orgánica con un sistema CAGD, con la cual se ejemplifican las propiedades teóricas de las funciones B-Spline y las superficies NURBS.

Palabras clave: Interpolación, Splines, B-splines, NURBS, CAGD, Modelado geométrico.

Contenido

Resumen	vii
1 Introducción	2
2 Interpolación de curvas	4
2.1 Interpolación de Lagrange	5
2.2 Método de las diferencias divididas de Newton	6
2.2.1 Error en la interpolación polinómica	7
2.3 Interpolación de Hermite	13
2.3.1 Error en la interpolación de Hermite	14
2.4 Splines	15
2.4.1 Convergencia de los splines lineales	15
2.4.2 Splines cúbicos	17
2.4.3 Convergencia de los splines cúbicos	21
2.4.4 B-Splines	29
3 Modelado geométrico	38
3.1 Curvas B-Spline	42
3.2 NURBS	46
3.2.1 Curvas NURBS	47
3.2.2 Superficies NURBS	52
4 Superficies NURBS en sistemas CAGD	59
5 Conclusiones	68
Bibliografía	69

1 Introducción

En la actualidad casi que cualquier objeto que utilizamos ha sido preconcebido como un diseño geométrico, el cual debe cumplir con unos requerimientos estándar que hagan que dicho objeto tenga un funcionamiento óptimo. Desde tornillos, motores, carros, aviones, hasta ciudades enteras han sido diseñadas previamente utilizando sistemas computacionales. La forma en que funcionan estos sistemas, los algoritmos que emplean, y la matemática subyacente a los mismos, es ignorada con frecuencia por casi la mayoría de nosotros, sin conocer el proceso de diseño previo consumimos infinidad de productos que fueron generados gracias a un amplio y constante desarrollo de herramientas, el cual está sustentado en técnicas matemáticas que han venido evolucionando a grandes pasos desde los primeros años de la década de 1960.

El diseño geométrico asistido por computadora, CAGD por sus siglas en inglés, ha sido una herramienta de gran impacto en la industria del entretenimiento y la ingeniería durante los últimos 50 años, la necesidad de crear y diseñar piezas y personajes cada vez más complejos ha sido un estimulante idóneo para el desarrollo de la matemática subyacente a las herramientas de diseño, los algoritmos utilizados por las mismas a lo largo de los años se han apoyado en las diferentes técnicas de construcción de curvas y superficies, siendo las curvas y superficies de Bézier las pioneras en este campo al inicio de los años sesenta, y en especial caso las curvas y superficies NURBS y de subdivisión que vinieron luego como la evolución de estos conceptos.

Las NURBS, *Non-Uniform Rational B-Splines*, son de especial importancia porque permiten representar curvas y superficies cónicas con precisión. De esta manera los sistemas de modelado que las implementan en sus rutinas tienen la capacidad de construir digitalmente un amplia gama de objetos, desde algunos elementales como líneas rectas y planos, hasta círculos y esferas, así como también intrincadas formas esculpidas a trozos. Si bien estas cumplen a cabalidad con la tarea de representar superficies “creíbles” para el ojo humano, las mismas pierden credibilidad cuando el nivel de detalle sobre la misma aumenta, lo que hace que su implementación no sea del todo idónea para la representación de secciones específicas de la pieza a modelar porque se pierde suavidad en la misma, por ejemplo objetos orgánicos, y es aquí donde entran en juego las superficies de subdivisión, pues estas se superponen a este tipo de inconvenientes [11].

Estas dos últimas son las técnicas estándar en las diferentes industrias gracias a su facilidad de cómputo y óptimos resultados respecto al producto final requerido, pues juntas permi-

ten la representación de superficies con grados de detalle ideales para la complejidad de las formas a construir involucradas. El software de diseño utilizado por los desarrolladores más importantes de la industria de la computación gráfica, tales como AUTODESK [2], PIXAR [18], y NVIDIA, incorpora estas técnicas en sus algoritmos y cada vez demanda nuevas y mejores herramientas matemáticas que le permita innovar en el mercado y a su vez aprovechar al máximo la capacidad del hardware disponible para su implementación.

Entender con claridad y profundidad los procesos de diseño que se implementan actualmente para modelar superficies es clave para poder ir en búsqueda de mejoras en los mismos, pues los procesos actuales precisamente se han podido establecer gracias a trabajos y técnicas previas que fueron objeto de investigación constante [1, 5, 8, 9]. Por otro lado, lograr la implementación de estas rutinas en entornos computacionales mas relacionados con el cálculo científico, como por ejemplo Matlab o Wolfram Mathematica, puede constituir una herramienta útil para futuros trabajos en esta área.

La descripción matemática de este proceso de diseño fue la motivación principal para la escritura de este documento, concretamente describir la construcción de superficies y su modelamiento en un entorno CAGD a través de superficies NURBS. En este documento pretendemos explicar entonces como modelan estos sistemas de diseño las diferentes superficies, que dan vida a un sin número de objetos cotidianos, a través de una clara descripción y análisis de las superficies mencionadas, yendo desde su formulación matemática y la descripción de sus propiedades más importantes, hasta un ejemplo de la implementación de las mismas para la modelación de una pieza de muestra. Mostrar como se generan en principio todos esos objetos que luego toman vida en forma de máquinas, películas, videojuegos, etc., evidencia el alto impacto que la matemática y la computación tienen en la industria y por ende en la sociedad.

Para lograr lo anterior abordamos inicialmente en el capítulo 2 el problema de la interpolación de curvas, en él discutimos los principales métodos de resolver dicho problema y hacemos especial énfasis en las funciones *splines* y sus propiedades de convergencia, allí mostramos como tales funciones forman un espacio vectorial de funciones polinomiales a trozos y las importantes propiedades que tienen las funciones generadoras del mismo, las funciones *B-splines*, propiedades que son de vital importancia para la construcción de las curvas y superficies NURBS que son presentadas en el capítulo 3, allí mostramos como se definen estos objetos y como las propiedades de las funciones que los constituyen hacen que sean ideales para el modelado geométrico. Finalmente, en el capítulo 4, mostramos un ejemplo real de la implementación de dichas superficies dentro de uno de los programas de diseño más utilizados en la industria del diseño, Autodesk Maya, en dicho ejemplo mostramos los datos geométricos que genera dicho programa e interpretamos los mismos para mostrar la correspondencia con los objetos estudiados. Las referencias principales consultadas para la elaboración de este documento fueron [23] y [17], las mismas contienen la información principal, y discusiones mucho más profundas y completas, de los capítulos 2 y 3 respectivamente.

2 Interpolación de curvas

A menudo es importante determinar o predecir el comportamiento de determinado fenómeno que se puede observar, sin embargo la poca información que se pueda llegar a rescatar u obtener del mismo dificulta esta tarea. Considere por ejemplo que usted necesita saber con exactitud la posición con respecto al tiempo de determinada partícula que se mueve muy rápido a lo largo y ancho de un plano de referencia, para ello marca en el mismo las diferentes posiciones que dicha partícula ha ocupado con anterioridad en un determinado instante de tiempo, dado que esta se mueve con mucha rapidez es difícil seguir su pista durante mucho tiempo, en consecuencia se obtendrán un número de marcas en el plano que representan el rastro de ella durante un intervalo de tiempo dado.

Ahora bien, el proceso anterior lleva a tener un conjunto discreto de pares ordenados (t_i, x_i) de puntos en el plano de referencia; es a partir de este conjunto que se desea simular el comportamiento de la partícula en aras de reproducir el movimiento de la misma durante el intervalo de tiempo que se observó su actividad. Lo ideal en este caso específico sería conocer una función, $f(t)$, cuyos valores se ajusten perfectamente al conjunto de muestra previamente tomado, esto es, $f(t_i) = x_i, \forall i$.

Suponiendo que el conjunto de la muestra tiene $m+1$ parejas ordenadas (nodos), a la función $f(t)$ tal que:

$$f(t_i) = x_i, \quad i = 0, \dots, m$$

la denominaremos *función interpolante*.

Las funciones de mayor comodidad para usar como interpolantes, dada su facilidad de representación y operabilidad, son los polinomios. En este caso queremos hallar entonces un polinomio $P_m(x)$ de grado menor o igual a m tal que:

$$P_m(t_i) = f(t_i), \quad \forall i = 0, 1, \dots, m,$$

a este polinomio lo denominaremos *polinomio interpolador de grado m de la función f* .

En ciertos fenómenos se conocen no solo los valores que toma la función interpolante en determinado punto t_i , sino también los valores que toman sus derivadas hasta determinado orden en dicho punto. En este caso al polinomio interpolante se le exigirá no solo que sus valores sean iguales punto a punto a los de f , sino que también el de sus derivadas. Lo anterior plantea entonces el problema de interpolación general:

Sea $\Phi(x; a_1, \dots, a_n)$ una función que depende de $x \in I \subset \mathbb{R}$ y de los n parámetros $a_1, \dots, a_n$. Sean además $x_1, \dots, x_m \in I$ nodos distintos a pares con multiplicidad $r_1, \dots, r_m \in \mathbb{N}$ con $\sum_{j=1}^m r_j = n$ y sean $y_{ij} \in \mathbb{R}$, $i = 1, \dots, m$, $j = 0, \dots, r_i - 1$ valores dados.

El problema consiste en encontrar los parámetros $a_1, \dots, a_n$ tales que

$$\Phi^{(j)}(x; a_1, \dots, a_n) = y_{ij}, \quad i = 1, \dots, m, \quad j = 0, \dots, r_i - 1. \quad (2.1)$$

Existen diferentes métodos generales para encontrar el polinomio interpolante de grado m que aproxima los valores de la función f , entre ellos:

- Interpolación de Lagrange.
- Método de las diferencias divididas de Newton.
- Interpolación de Hermite.
- Interpolación por funciones spline.

A continuación discutimos los aspectos más generales de estos métodos y realizamos un breve análisis del error asociado a cada uno de ellos.

2.1. Interpolación de Lagrange

Supongamos que se quiere interpolar un conjunto de únicamente dos parejas ordenadas, lo más natural en este caso es pensar en la línea recta que une estos dos puntos, es decir, sean $P(x_0, y_0)$ y $P(x_1, y_1)$, la ecuación de la recta que los une es:

$$y = P(x) = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0). \quad (2.2)$$

Reescribiendo (2.2) se tiene que:

$$y = P(x) = \frac{y_1 - y_0}{x_1 - x_0}x + \frac{y_0x_1 - x_0y_1}{x_1 - x_0},$$

el cual es un polinomio de grado menor o igual a 1, y que satisface las condiciones $P(x_0) = y_0$ y $P(x_1) = y_1$. Lagrange propuso otro método para encontrar este polinomio; reescribiendo nuevamente (2.2) se obtiene:

$$y = P(x) = y_1 \frac{x - x_0}{x_1 - x_0} + y_0 \left(1 - \frac{x - x_0}{x_1 - x_0}\right)$$

de donde se obtiene finalmente que,

$$y = P(x) = y_1 \frac{x - x_0}{x_1 - x_0} + y_0 \frac{x - x_1}{x_0 - x_1}$$

Si definimos ahora las expresiones lineales $L_{1,0}(x)$ y $L_{1,1}(x)$ como sigue:

$$L_{1,0}(x) = \frac{x - x_1}{x_0 - x_1} \quad (2.3)$$

y

$$L_{1,1}(x) = \frac{x - x_0}{x_1 - x_0}. \quad (2.4)$$

Así podemos expresar el polinomio $P(x)$ como,

$$P(x) = y_0 L_{1,0}(x) + y_1 L_{1,1}(x) \quad (2.5)$$

donde claramente (2.5) cumple con las condiciones $P(x_0) = y_0$ y $P(x_1) = y_1$. Los términos lineales (2.3) y (2.4) se definen como *polinomios coeficientes de Lagrange* [24].

Observe que (2.5) se puede escribir también como,

$$P_1(x) = \sum_{k=0}^1 y_k L_{1,k}(x).$$

En general si se tienen entonces $n + 1$ puntos $x_0, x_1, \dots, x_n$ y si $f(x_k) = P(x_k)$ para $k = 0, 1, 2, 3, \dots, n$ el polinomio que pasa por estos $n + 1$ puntos es:

$$P_n(x) = \sum_{k=0}^n y_k L_{n,k}(x) \quad (2.6)$$

donde,

$$L_{n,k} = \frac{(x - x_0)(x - x_1) \dots (x - x_{k-1})(x - x_{k+1}) \dots (x - x_n)}{(x_k - x_0)(x_k - x_1) \dots (x_k - x_{k-1})(x_k - x_{k+1}) \dots (x_k - x_n)}$$

de forma más compacta se tiene entonces,

$$L_{n,k}(x) = \prod_{\substack{j=0 \\ j \neq k}}^n \frac{(x - x_j)}{(x_k - x_j)}.$$

2.2. Método de las diferencias divididas de Newton

Al observar con más cuidado el método de interpolación de Lagrange encontramos que de querer agregar un nuevo nodo sería necesario calcular totalmente el polinomio de nuevo dada la naturaleza con la que se construyó el mismo. Sea $U = \{(x_i, y_i), i = 0, \dots, n\}$ el conjunto discreto de las parejas ordenadas correspondientes a los valores a interpolar (nodos). La desventaja de la fórmula de Lagrange es la necesidad de calcular nuevamente todos los polinomios coeficientes $L_{i,j}$ cuando se agrega un nuevo nodo (x_{n+1}, y_{n+1}) al conjunto U . Considere la siguiente representación:

$$P(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \cdots + c_n(x - x_0) \cdots (x - x_{n-1}),$$

donde los coeficientes c_i son las cantidades a determinar. De la condición $P(x_i) = y_i$ se pueden calcular los valores de c_i en forma sucesiva

$$c_0 = y_0, \quad c_1 = \frac{y_1 - y_0}{x_1 - x_0}, \dots, c_n.$$

Los coeficientes c_k se pueden calcular usando el esquema recursivo de *diferencias divididas*:

$$\begin{aligned} f[x_i] &:= y_i, \\ f[x_0, x_1, \dots, x_n] &:= \frac{f[x_1, x_2, \dots, x_n] - f[x_0, x_1, \dots, x_{n-1}]}{x_n - x_0}. \end{aligned}$$

$f[x_0, x_1, \dots, x_n]$ es la diferencia dividida de orden n . Por ejemplo,

$$\begin{aligned} f[x_0, x_1, x_2] &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} \\ &= \frac{\frac{f[x_2] - f[x_1]}{x_2 - x_1} - \frac{f[x_1] - f[x_0]}{x_1 - x_0}}{x_2 - x_0} \\ &= \frac{\frac{y_2 - y_1}{x_2 - x_1} - \frac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_0} \end{aligned}$$

y así sucesivamente.

Definimos entonces el polinomio de interpolación de Newton como:

$$P_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + \cdots + f[x_0, \dots, x_n] \prod_{i=0}^{n-1} (x - x_i). \quad (2.7)$$

2.2.1. Error en la interpolación polinómica

A excepción de coincidir en los nodos de interpolación (x_i, y_i) , el polinomio interpolante $P(x)$ y la función subyacente $f(x)$ son en general diferentes. El error en la interpolación es una medida de qué tan diferentes son estas dos funciones. Esta idea general involucra entonces las distintas formas de estimar dicha medida y también las estrategias de como minimizar el mismo.

Teorema 1. Sea $f(x) \in C^{n+1}([a, b])$. Sea $P_n(x) \in \Pi_n$ tal que interpola a $f(x)$ en los $n + 1$ puntos distintos $x_0, x_1, \dots, x_n \in [a, b]$. Entonces, para todo $x \in [a, b]$ existe ξ en el intervalo cerrado $I[x_0, \dots, x_n, x]$ más pequeño que contiene a x y a los x_i tal que

$$f(x) - P_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi) w(x), \quad (2.8)$$

donde,

$$w(x) = \prod_{i=0}^n (x - x_i).$$

Demostración. El resultado es trivial cuando x coincide con alguno de los nodos de interpolación $x_i, i = 0, 1, \dots, n$. Supongamos entonces que $x \neq x_i$ y consideremos la función

$$h(z) := w(x) (f(z) - P_n(z)) - w(z) (f(x) - P_n(x)).$$

Observe que tanto x como los demás nodos de interpolación x_i son ceros de la función $h(z)$ en $I[x_0, x_1, \dots, x_n, x]$, así, por el teorema de Rolle sabemos que $h'(z)$ tiene al menos $n+1$ ceros en este mismo intervalo. Análogamente, $h''(z)$ tiene al menos n ceros, y así sucesivamente. Sabemos entonces que $h^{(n+1)}(z)$ tiene al menos un cero en $I[x_0, \dots, x_n, x]$, luego

$$h^{(n+1)}(\xi) = 0 = w(x) (f^{(n+1)}(\xi) - P_n^{(n+1)}(\xi)) - w^{(n+1)}(\xi) (f(x) - P_n(x)), \quad \xi \in I[x_0, \dots, x_n, x].$$

Dado que $P_n \in \Pi_n$ se cumple que $P_n^{(n+1)}(\xi) = 0$, por otro lado, es claro que $w^{(n+1)}(\xi) = (n+1)!$. Reemplazando estos dos términos en la ecuación anterior obtenemos que

$$0 = w(x) f^{(n+1)}(\xi) - (n+1)! (f(x) - P_n(x)), \quad \xi \in I[x_0, \dots, x_n, x]$$

esto es,

$$f(x) - P_n(x) = \frac{1}{(n+1)!} w(x) f^{(n+1)}(\xi), \quad \xi \in I[x_0, \dots, x_n, x]$$

□

Del teorema anterior y del polinomio de Newton dado por (2.7) podemos obtener un término distinto para el error. Supongamos que $x = x_{n+1}$ es un nodo de interpolación distinto a los $n+1$ nodos x_i iniciales, entonces el polinomio de Newton que interpola a f en los nodos $x_i, i = 0, 1, \dots, n+1$ está dado por

$$P_{n+1}(x_{n+1}) = f[x_0] + \dots + f[x_0, \dots, x_n] \prod_{i=0}^{n-1} (x - x_i) + f[x_0, \dots, x_{n+1}] \prod_{i=0}^n (x - x_i),$$

esto es,

$$P_{n+1}(x_{n+1}) = P_n(x_{n+1}) + f[x_0, \dots, x_{n+1}] \prod_{i=0}^n (x - x_i),$$

luego,

$$f(x_{n+1}) - P_n(x_{n+1}) = f[x_0, \dots, x_{n+1}] w(x). \quad (2.9)$$

Más aún, podemos obtener una relación entre las derivadas y las diferencias divididas de Newton. En efecto, igualando el lado derecho de las ecuaciones (2.8) y (2.9) obtenemos que

$$f[x_0, \dots, x_{n+1}] w(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi) w(x),$$

y dado que $w(x) \neq 0$ tenemos que,

$$f[x_0, \dots, x_{n+1}] = \frac{1}{(n+1)!} f^{(n+1)}(\xi), \quad \xi \in I[x_0, \dots, x_n, x_{n+1}]$$

lo cual implica que

$$f[x_0, \dots, x_n] = \frac{1}{n!} f^{(n)}(\xi), \quad \xi \in I[x_0, \dots, x_n] \quad (2.10)$$

la cual es la relación entre las derivadas y las diferencias divididas de Newton.

El error en la interpolación depende de la suavidad de f a través de su derivada de orden superior $f^{(n+1)}$, tiene un factor $\frac{1}{(n+1)!}$ que decae rápidamente a medida que $n \rightarrow \infty$, además es directamente proporcional al valor de $w(x)$, lo que indica que la interpolación podría ser mejor en algunos lugares que en otros [10].

Ahora la pregunta que surge naturalmente es la convergencia: se puede esperar siempre la convergencia de los polinomios interpolantes cuando $n \rightarrow \infty$? En otras palabras, el factor $\frac{1}{(n+1)!}$ siempre supera a los otros dos factores? Desafortunadamente la respuesta en general es NO. Hay ejemplos de funciones muy suaves para las cuales la interpolación polinómica diverge, en particular muy cerca de los extremos del intervalo de interpolación. Este comportamiento se conoce como el *fenómeno de Runge*, lo anterior se ilustra con el siguiente ejemplo:

Considere la función

$$f(x) = \frac{1}{1+x^2}$$

y la secuencia de polinomios $\{p_0, p_1, p_2, \dots\}$ donde p_k es el polinomio que interpola a f en $k+1$ puntos equidistantes en el intervalo $[-5, 5]$.

En la figura (2.1) puede verse claramente como los valores que toman los polinomios interpolantes $p_n(x)$ difieren ampliamente de los de la función f cerca de los extremos del intervalo, en la figura (2.1b) por ejemplo los valores de $p_{18}(x)$ son cercanos a 30, mientras que el valor máximo de f es 1.

El ejemplo anterior motiva la presentación de tres importantes resultados que involucran la escogencia de los nodos interpolantes utilizados para aproximar la función f .

Teorema 2 (Faber). *Suponga que para cada $n = 0, 1, 2, \dots$ se tiene un sistema prescrito de n nodos*

$$a \leq x_{n,0} < x_{n,1} < \dots < x_{n,n-1} < x_{n,n} \leq b,$$

entonces existe una función continua $f(x)$ tal que los polinomios interpolantes de f que utilizan estos nodos no convergen uniformemente a f .

Demostración. Véase [4]. □

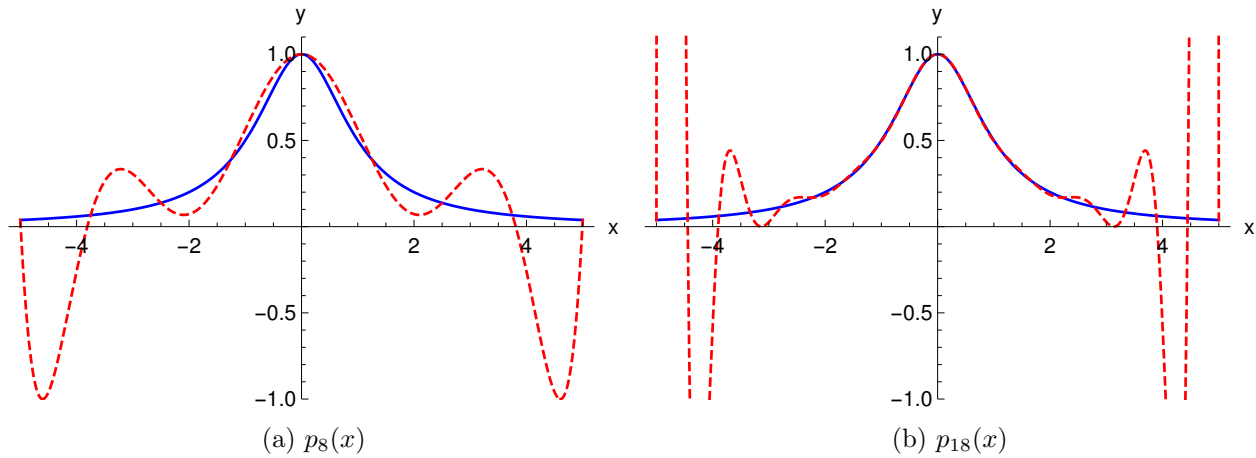


Figura 2.1: Ilustración del fenómeno de Runge. Las figuras (a) y (b) muestran la función f (en azul) junto a los polinomios interpolantes de grado 8 y 18 respectivamente (línea punteada roja), en ambos casos se utilizaron nodos equidistantes.

En otras palabras, no existe una elección adecuada de nodos que pueda acomodar todas las funciones.

Teorema 3 (Aproximación de Weierstrass). *Dado cualquier $\varepsilon > 0$ y cualquier $f \in C[a, b]$, entonces existe un polinomio p tal que*

$$|f(x) - p(x)| < \varepsilon \text{ para todo } x \in [a, b].$$

Demostración. Véase [4]. □

En otras palabras, el teorema garantiza que existe al menos una secuencia de polinomios que convergen uniformemente a f , pero no que esta secuencia sea de polinomios interpolantes de f .

Por ultimo, un resultado positivo relacionado con la convergencia de la secuencia de polinomios interpolantes de una función continua dada es el siguiente:

Teorema 4. *Si f es una función continua en $[a, b]$, entonces existe un sistema de n – nodos*

$$a \leq x_{n,0} < x_{n,1} < \cdots < x_{n,n-1} < x_{n,n} \leq b,$$

tal que la secuencia correspondiente de polinomios interpolantes converge uniformemente a f .

Demostración. Véase [4]. □

Al leer con detenimiento el teorema de Faber se puede ver que este da cuenta de la alta probabilidad de no obtener la convergencia deseada de la secuencia de polinomios generados a partir de los nodos disponibles. Nos dice que no hay una forma universal y óptima de escoger

los nodos, más aún, nos sugiere que puede ser posible mejorar la interpolación polinomial eligiendo un conjunto distinto de nodos. En efecto, dado que los polinomios de mayor grado son extremadamente sensibles a pequeños cambios en sus coeficientes, es a menudo una mejor estrategia optimizar la elección de los puntos muestrales en vez de simplemente incrementar el número de los mismos [4].

La consecuencia más relevante entonces de los resultados anteriores es que el error en la interpolación puede minimizarse a través de una adecuada escogencia de los nodos. Si se cuenta con la flexibilidad de escoger el conjunto de muestra de la función a interpolar, sería razonable pensar en este grado de libertad para buscar la convergencia deseada en la secuencia de polinomios interpolantes de f . En (2.8) podemos ver que los puntos de interpolación x_j tienen influencia en dos términos del lado derecho de la ecuación, el más obvio es el producto

$$w(x) = \prod_{i=0}^n (x - x_i).$$

El otro término que depende de estos es $f^{(n+1)}(\xi_n)$ dado que el punto intermedio ξ_n depende de x_i . Dada la dependencia implícita de ξ_n de los nodos, minimizar el error en la interpolación no es una tarea sencilla. El enfoque principal será entonces la escogencia de los puntos de interpolación $x_0, x_1, \dots, x_n$ tales que el producto $w(x)$ sea minimizado. Cabe recalcar aquí que esta escogencia busca minimizar solamente a $w(x)$, pero esto no garantiza, en general, minimizar el error en la interpolación [23].

Una secuencia de nodos que cumple a cabalidad con esta tarea son los denominados *nodos de Chebyshev*, los cuales están definidos como sigue:

$$x_i^{(n)} = \cos\left(\frac{(2i+1)\pi}{2(n+1)}\right), \quad i = 0, 1, \dots, n.$$

Al observar la figura (2.2) se puede apreciar como a medida de que aumenta el grado del polinomio interpolante el error en la interpolación empieza a disminuir drásticamente, tanto así que en la figura (2.2d) son prácticamente indistinguibles la función f y el polinomio $p_{32}(x)$ que la interpola. Lo anterior es entonces una muestra clara de que si se tiene libertad en la escogencia de los nodos de interpolación, los de Chebyshev son una muy buena elección.

Cabe resaltar en este punto un aspecto importante respecto a los nodos de Chebyshev y al intervalo de interpolación de la función f ; inicialmente en el ejemplo los nodos deben estar en el intervalo $[-5, 5]$, pero dado que estos están definidos a partir de la función $\cos(x)$ quedan confinados al intervalo $[-1, 1]$, y de esta forma la aproximación de f queda restringida a este intervalo. Pues bien, en el caso mas general en el que el intervalo de interpolación consta de valores $x \in [a, b]$, es posible utilizar los resultados previos llevando el intervalo a la forma $[-1, 1]$, para ello basta hacer

$$x = \frac{(b-a)y + (a+b)}{2}, \quad y \in [-1, 1]$$

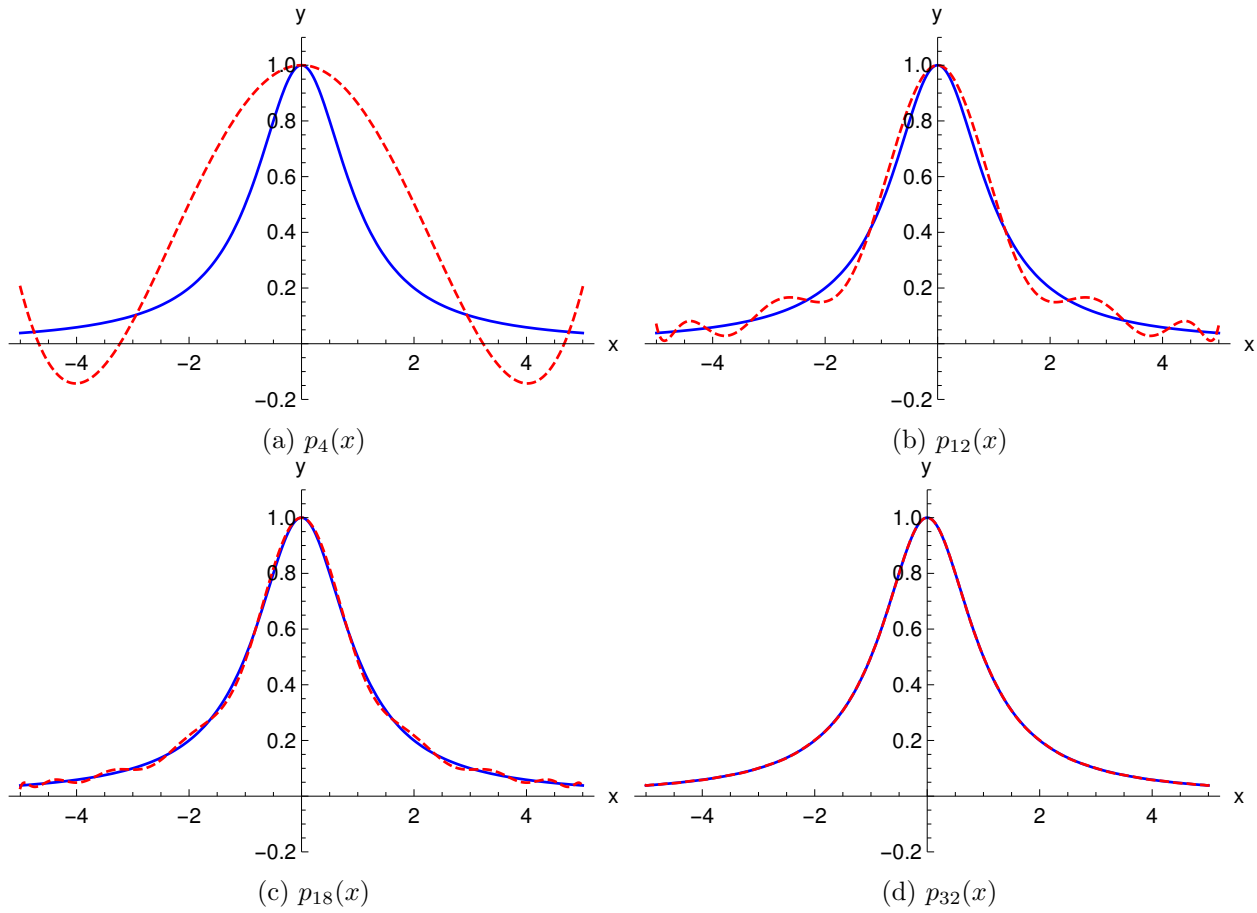


Figura 2.2: Las figuras (a)-(d) muestran la función f (en azul) junto a los polinomios interpolantes de grado 4, 12, 18, y 32 respectivamente (línea punteada roja), en todos los casos se utilizaron los nodos de Chebyshev.

esto convierte el problema de interpolación inicial de $f(x)$ sobre $[a, b]$ en el problema de interpolación $f(x) = g(x(y))$ sobre el intervalo $[-1, 1]$, así los nuevos nodos de interpolación en $[-1, 1]$ serán,

$$y_i^{(n)} = \cos\left(\frac{(2j+1)\pi}{2(n+1)}\right), \quad j = 0, 1, \dots, n$$

y los correspondientes $n+1$ nodos de interpolación en el intervalo $[a, b]$ serán

$$x_j = \frac{(b-a)y_j + (a+b)}{2}, \quad j = 0, 1, \dots, n.$$

Hasta este momento los métodos estudiados para solucionar el problema de interpolación general dado por (2.1) solo han requerido que el polinomio interpolante coincida en los nodos x_i con los de la función subyacente f , los siguientes métodos además de lo anterior consideran los valores que toman las derivadas en dichos puntos.

2.3. Interpolación de Hermite

Sean $x_0, x_1, x_2, \dots, x_n \in [a, b]$, m_i un entero no negativo asociado a cada x_i , con $i = 0, 1, 2, 3, \dots, n$, $f \in C^m[a, b]$ con $m = \max_{0 \leq i \leq n} m_i$, definimos el *Polinomio osculante* que aproxima a f como el polinomio de menor grado tal que:

$$\frac{d^{(k)}P(x_i)}{dx^k} = \frac{d^{(k)}f(x_i)}{dx^k}, \quad \forall k = 0, 1, 2, 3, \dots, m_i.$$

Ahora, si $m_i = 1$ para $i = 0, 1, 2, \dots, n$, el polinomio osculante que aproxima a f se denomina *Polinomio de Hermite*, observe que en este caso $P(x_k) = f(x_k)$ y $P'(x_k) = f'(x_k)$, $k = 0, 1, 2, \dots, n$.

Sean $f \in C([a, b])$ y $x_0, \dots, x_n \in [a, b]$ nodos distintos. El problema de interpolación consiste en determinar $P \in \Pi_{2n+1}$ tal que

$$p(x_i) = f(x_i) := f_j, \quad p'(x_j) = f'(x_j) := f'_j, \quad j = 0, \dots, n. \quad (2.11)$$

Para calcular p suponemos que este tiene la forma

$$p(x) = \sum_{j=0}^n f_j \varphi_j(x) + \sum_{j=0}^n f'_j \Psi_j(x) \quad (2.12)$$

donde $\varphi_j, \Psi_j \in \Pi_{2n+1}$ son polinomios caracterizados por

$$\begin{aligned} \varphi_j(x_i) &= \delta_{ij}, & \varphi'_j(x_i) &= 0, \\ \Psi_j(x_i) &= 0, & \Psi'_j(x_i) &= \delta_{ij}, \quad i, j = 0, \dots, n. \end{aligned}$$

Si las funciones φ_j, Ψ_j existen, entonces p en la forma (2.12) satisface las condiciones de interpolación dadas por (2.11). Si utilizamos los polinomios coeficientes de Lagrange presentes en el polinomio interpolante dado por (2.6) para definir las funciones φ_j y Ψ_j , obtenemos

$$\begin{aligned}\varphi_j(x) &:= (1 - 2L'_{n,j}(x_j)(x - x_j))L_{n,j}^2(x) \in \Pi_{2n+1} \\ \Psi_j(x) &:= (x - x_j)L_{n,j}^2(x) \in \Pi_{2n+1}\end{aligned}$$

Así, existe $P \in \Pi_{2n+1}$ que satisface (2.11).

2.3.1. Error en la interpolación de Hermite

El error en la interpolación de Hermite viene dado por el siguiente resultado.

Teorema 5. Sean $f \in C^{2n+2}([a, b])$, $x_0, x_1, \dots, x_n \in [a, b]$ con $x_i \neq x_j$ si $i \neq j$ y $P(x) \in \Pi_{2n+1}$ el polinomio interpolante que satisface (2.11). Entonces para todo $x \in [a, b]$ existe un $\zeta \in I(x, x_0, \dots, x_n)$, el intervalo más pequeño que contiene a los x_i y a x , tal que

$$f(x) - P_n(x) = \frac{w^2(x)}{(2n+2)!} f^{(2n+2)}(\zeta),$$

donde w está definido por

$$w(x) = \prod_{i=0}^n (x - x_i).$$

Demostración. La demostración de este resultado es análoga a la del teorema (1), salvo que en este caso la función $h(z)$ está definida como

$$h(z) := (f(x) - p(x))w^2(z) - (f(z) - p(z))w^2(x).$$

□

La interpolación polinómica es fácil de realizar e implementar, pero la convergencia se deteriora cuando se refina la partición $x_0, x_1, \dots, x_n$ de $[a, b]$, es decir, cuando se agregan más nodos. Refinar la partición agregando puntos nodales implica aumentar el grado del polinomio interpolante, el cual será susceptible a grandes variaciones en sus valores cuando se efectúen pequeños cambios en los coeficientes del mismo. Una estrategia efectiva para evitar este inconveniente es realizar una aproximación segmentada sobre la partición del intervalo $[a, b]$ a través de polinomios de grado no muy grande, esto es, aproximar la función f sobre cada subintervalo $[x_{i-1}, x_i]$, $i = 1, 2, \dots, n$ con polinomios p_i . Las funciones *spline* hacen uso de esta idea y tienen mejores propiedades de convergencia.

2.4. Splines

Sea $\Delta : a \leq x_0 < x_1 < \dots < x_n \leq b$ una partición del intervalo $[a, b]$. Además, representamos con $S(\Delta, p, q)$, $p, q \in \mathbb{N}_0$, $0 \leq q < p$, el conjunto de funciones $s \in C^q([a, b])$ que restringidas al intervalo $[x_{i-1}, x_i]$, $i = 1, \dots, n$, coinciden con un polinomio de grado menor o igual a p . Un elemento $s \in S(\Delta, p, q)$ se llama *polinomio spline o simplemente spline* de la partición Δ . $S(\Delta, p, q)$ es un espacio vectorial sobre $\mathbb{R}$ de dimensión finita.

En la práctica los espacios más usados, sobre todo en problemas de valores de frontera, son $S(\Delta, 3, 2)$ (*splines cúbicos*) y $S(\Delta, 3, 1)$ (*splines cúbicos de Hermite*.) En aplicaciones donde se presentan polos o rápido crecimiento de la función se utilizan con frecuencia splines racionales o exponenciales.

En las líneas siguientes discutiremos el caso,

$$S_k := S(\Delta, k, k - 1)$$

y por simplicidad empezamos con *splines lineales*:

$$S_1(\Delta) = \{f \in C([a, b]) : f|_{[x_{i-1}, x_i]} \in \Pi_1, \quad i = 1, \dots, n\}.$$

Para toda función $f \in C([a, b])$ existe un único $s \in S_1(\Delta)$ tal que $f(x_i) = s(x_i)$, para $i = 0, \dots, n$; s se puede calcular interpolando f con una función lineal en el intervalo $[x_{i-1}, x_i]$,

$$s(x) = \frac{1}{x_i - x_{i-1}} (f(x_i)(x - x_{i-1}) + f(x_{i-1})(x_i - x)), \quad x \in [x_{i-1}, x_i]$$

2.4.1. Convergencia de los splines lineales

Consideremos una sucesión de particiones Δ^k de $[a, b]$:

$$\Delta^k : a = x_0^k < x_1^k < \dots < x_k^k = b$$

tal que

$$\bar{\Delta}^k := \max_{i=0, \dots, k-1} x_{i+1}^k - x_i^k$$

tienda a cero cuando $k \rightarrow \infty$. Sean $f \in C([a, b])$ una función y Δ^k una partición fija, que en lo sucesivo se denotará simplemente Δ .

El spline lineal de interpolación s satisface en la posición $x \in [x_{i-1}, x_i]$:

$$s(x) = f(x_{i-1}) \underbrace{\frac{x_i - x}{x_i - x_{i-1}}}_{=\alpha} + f(x_i) \underbrace{\frac{x - x_{i-1}}{x_i - x_{i-1}}}_{=1-\alpha}$$

$$f(x) = f(x)\alpha + f(x)(1 - \alpha)$$

$$\Rightarrow |f(x) - s(x)| \leq |f(x) - f(x_{i-1})|\alpha + |f(x) - f(x_i)|(1 - \alpha).$$

f es uniformemente continua; por lo tanto existe para todo $\varepsilon > 0$ un $\delta > 0$ tal que

$$|x - y| \leq \delta \Rightarrow |f(x) - f(y)| \leq \varepsilon$$

Entonces se tiene de $\bar{\Delta} \leq \delta$ que

$$|f(x) - s(x)| \leq \varepsilon(\alpha + (1 - \alpha)) = \varepsilon$$

y con esto

$$\|f - s\|_{\infty} \leq \varepsilon$$

y así se ha demostrado que:

Para todo $f \in C([a, b])$ y $\varepsilon > 0$ existe $\delta > 0$ tal que

$$\bar{\Delta} \leq \delta \Rightarrow \|f - s\|_{\infty} \leq \varepsilon.$$

Finalmente concluimos que

$$s = s^{(k)} \rightarrow f \text{ respecto de la norma } \|\cdot\|_{\infty} \text{ cuando } k \rightarrow \infty.$$

De forma análoga a la interpolación de Lagrange se puede representar el spline s como

$$s(x) = \sum_{i=0}^n f(x_i) \varphi_i(x), \quad x \in [a, b]$$

donde las funciones base $\varphi_i(x)$ están dadas por

$$\begin{aligned} \varphi_0(x) &:= \begin{cases} \frac{x_1 - x}{x_1 - x_0}, & x \in [x_0, x_1] \\ 0, & x \notin [x_0, x_1] \end{cases} \\ \varphi_i(x) &:= \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}}, & x \in [x_{i-1}, x_i] \\ \frac{x_{i+1} - x}{x_{i+1} - x_i}, & x \in [x_i, x_{i+1}] \\ 0, & x \notin [x_{i-1}, x_{i+1}] \end{cases} \quad i = 1, \dots, n-1 \\ \varphi_n(x) &:= \begin{cases} \frac{x - x_{n-1}}{x_n - x_{n-1}}, & x \in [x_{n-1}, x_n] \\ 0, & x \notin [x_{n-1}, x_n] \end{cases} \end{aligned}$$

Las funciones φ_i se denominan *funciones tienda* y tienen soporte $[x_{i-1}, x_{i+1}]$. Para calcular s en el punto $x \in (x_{i-1}, x_{i+1})$ solo es necesario evaluar $\varphi_{i-1}(x)$ y $\varphi_i(x)$, dado que $\varphi_j(x) = 0$ para todo $j \neq i, i-1$.

2.4.2. Splines cúbicos

La descripción de splines cúbicos no es tan sencilla, s es un polinomio de tercer grado en cada intervalo $[x_{i-1}, x_i]$; se necesitan $4n$ parámetros para describir s . La regularidad de $s \in C^2([a, b])$ se usa para prescribir tres condiciones en los nodos internos $x_i, i = 1, \dots, n-1$:

$$s^{(j)}(x_i^-) = s^{(j)}(x_i^+), \quad j = 0, 1, 2.$$

Por lo tanto, un spline cúbico posee por lo menos $4n - (3n - 3) = n + 3$ grados de libertad y no se puede determinar mediante las $n + 1$ condiciones de interpolación

$$s(x_i) = f(x_i), \quad i = 0, \dots, n$$

Entonces es necesario prescribir 2 *condiciones de frontera*. Existen varias formas de hacer esto:

1. Si con s se quiere aproximar una función periódica, entonces tiene sentido exigir

$$s'(x_0) = s'(x_n), \quad s''(x_0) = s''(x_n)$$

2. Si f es diferenciable en x_0 y x_n entonces se imponen las condiciones

$$s'(x_0) = f'(x_0), \quad s'(x_n) = f'(x_n)$$

3. Si se quiere extender s a $\{x : x < x_0\}$ o $\{x : x > x_n\}$ como una función lineal dos veces diferenciable, entonces

$$s''(x_0) = s''(x_n) = 0$$

Splines contruidos de esta forma se denominan *splines naturales*

En los tres casos el spline s está determinado de forma única, mediante interpolación y condiciones de frontera. Para demostrar este hecho utilizaremos algunos resultados preliminares.

Dada una función $f \in C^2([a, b])$, se define la seminorma

$$\|f\|^2 := \int_a^b |f''(x)|^2 dx, \quad (2.13)$$

la cual cumple todas las propiedades de una norma excepto $\|f\| = 0$ cuando $f = 0$, pues $\|f\| = 0$ puede darse para funciones que no sean idénticamente cero, por ejemplo para todas las funciones lineales $f(x) = cx + d$.

Teorema 6. Sean $f \in C^2([a, b])$, y $s \in S_3(\Delta)$ arbitrario. Entonces

$$\|f - s\|^2 = \|f\|^2 - \|s\|^2 - 2 \left[(f'(x) - s'(x)) s''(x) \Big|_a^b - \sum_{i=1}^n (f(x) - s(x)) s'''(x) \Big|_{x_{i-1}^+}^{x_i^-} \right] \quad (2.14)$$

Demostración. Por la definición (2.13) tenemos que

$$\begin{aligned}\|f - s\|^2 &= \int_a^b |f''(x) - s''|^2 dx \\ &= \|f\|^2 - 2 \int_a^b f''(x) s''(x) dx + \|s\|^2 \\ &= \|f\|^2 - 2 \int_a^b (f''(x) - s''(x)) s'' dx - \|s\|^2.\end{aligned}$$

Integrando por partes para $i = 1, 2, \dots, n$ se tiene que

$$\begin{aligned}\int_{x_{i-1}}^{x_i} (f''(x) - s''(x)) s''(x) dx &= (f'(x) - s'(x)) s''(x) \Big|_{x_{i-1}}^{x_i} - \int_{x_{i-1}}^{x_i} (f'(x) - s'(x)) s'''(x) dx \\ &= (f'(x) - s'(x)) s''(x) \Big|_{x_{i-1}}^{x_i} - (f(x) - s(x)) s'''(x) \Big|_{x_{i-1}^+}^{x_i^-} \\ &\quad + \int_{x_{i-1}}^{x_i} (f(x) - s(x)) s^{(4)}(x) dx.\end{aligned}$$

Observe que $s^{(4)} \equiv 0$ en cada subintervalo (x_{i-1}, x_i) , y f', s', s'' son continuas en $[a, b]$, así, sumando las fórmulas anteriores para $i = 1, 2, \dots, n$ y usando el hecho de que

$$\sum_{i=1}^n (f'(x) - s'(x)) s''(x) \Big|_{x_{i-1}}^{x_i} = (f'(x) - s'(x)) s''(x) \Big|_a^b,$$

concluimos que

$$\|f - s\|^2 = \|f\|^2 - \|s\|^2 - 2 \left[(f'(x) - s'(x)) s''(x) \Big|_a^b - \sum_{i=1}^n (f(x) - s(x)) s'''(x) \Big|_{x_{i-1}^+}^{x_i^-} \right].$$

□

Con la ayuda del teorema anterior se puede demostrar la *propiedad de norma mínima* de las funciones spline cúbicas.

Teorema 7. Sea $f \in C^2([a, b])$. Entonces

$$\|f - s\|^2 = \|f\|^2 - \|s\|^2 \geq 0$$

para todo $s \in S_3(\Delta)$ que satisface las condiciones de interpolación $s(x_i) = f(x_i)$ para $i = 0, 1, \dots, n$ y que además satisface una de las siguientes condiciones de frontera:

1) f periódica: $s^{(k)}(a) = s^{(k)}(b)$, para $k = 0, 1, 2$.

2) f diferenciable en $x = a$ y $x = b$: $s'(a) = f'(a)$, $s'(b) = f'(b)$.

$$3) \quad s''(a) = s''(b) = 0.$$

En todos los casos el spline s es único.

Demostración. La existencia se demostrará más adelante de forma constructiva. En cada una de las condiciones de frontera anteriores el término

$$(f'(x) - s'(x)) s''(x) \Big|_a^b - \sum_{i=1}^n (f(x) - s(x)) s'''(x) \Big|_{x_{i-1}^+}^{x_i^-}$$

de (2.14) se anula, esto prueba la propiedad minimal de la función spline s . La unicidad del mismo puede verse de la siguiente manera; si existiera otro $\bar{s} \in S_3(\Delta)$ con las propiedades enumeradas, entonces $\bar{s}$ sería una función f como la del teorema, entonces

$$\|\bar{s} - s\|^2 = \|\bar{s}\|^2 - \|s\|^2 \geq 0,$$

y como el papel de s y $\bar{s}$ se puede intercambiar, obtenemos que

$$0 \leq \|s\|^2 - \|\bar{s}\|^2 = \|s - \bar{s}\|^2 = \int_a^b |s''(x) - \bar{s}''(x)|^2 dx = 0.$$

Puesto que s'' y $\bar{s}''$ son funciones continuas, $s''(x) = \bar{s}''(x)$ para todo $x \in [a, b]$. Integrando obtenemos que

$$\bar{s}(x) = s(x) + cx + d,$$

y dado que $\bar{s}(a) = s(a)$, $\bar{s}(b) = s(b)$ se tiene que $c = d = 0$, es decir, $s = \bar{s}$. $\square$

Este teorema afirma entonces que el spline natural (condición de frontera 3) minimiza la integral $\|f\|^2$. Como $|f''(x)|^2$ es una medida de la curvatura de f en el punto x , entonces s es la función dos veces diferenciable más suave que interpola los datos $f(x_i)$, en el sentido de que proporciona el menor valor de $\|f\|^2$.

Mostraremos ahora a través de un enfoque constructivista la existencia de los splines que satisfacen las condiciones de frontera 1,2, o 3. Sea Δ una partición fija de $[a, b]$ y sean $y_0, y_1, \dots, y_n \in \mathbb{R}$. Definamos ahora las cantidades $h_{j+1} := x_{j+1} - x_j$, $j = 0, 1, \dots, n-1$, y $M_j := s''(x_j)$, $j = 0, 1, \dots, n$. Queremos mostrar que M_j es solución de un sistema de ecuaciones lineales y que a partir de allí se puede calcular s con facilidad. Observe que s restringido al intervalo $[x_j, x_{j+1}]$ es un polinomio cúbico, entonces s'' es lineal a trozos y se puede escribir en términos de M_j de la siguiente forma

$$s''(x) = \frac{1}{h_{j+1}} (M_j(x_{j+1} - x) + M_{j+1}(x - x_j)), \quad x \in [x_j, x_{j+1}],$$

integrando esta función dos veces sobre el intervalo $[x_j, x_{j+1}]$, $j = 0, 1, \dots, n-1$ obtenemos que

$$s'(x) = -M_j \frac{(x_{j+1} - x)^2}{2h_{j+1}} + M_{j+1} \frac{(x - x_j)^2}{2h_{j+1}} + A_j \tag{2.15}$$

$$s(x) = M_j \frac{(x_{j+1} - x)^3}{6h_{j+1}} + M_{j+1} \frac{(x - x_j)^3}{6h_{j+1}} + A_j(x - x_j) + B_j, \tag{2.16}$$

donde A_j y B_j son constantes de integración. Dado que $s(x_j) = y_j$ y $s(x_{j+1}) = y_{j+1}$, las constantes de integración satisfacen

$$\begin{aligned} y_j &= M_j \frac{h_{j+1}^2}{6} + B_j \\ y_{j+1} &= M_{j+1} \frac{h_{j+1}^2}{6} + A_j h_{j+1} + B_j, \end{aligned}$$

de donde se obtiene que,

$$\begin{aligned} B_j &= y_j - M_j \frac{h_{j+1}^2}{6} \\ A_j &= \frac{y_{j+1} - y_j}{h_{j+1}} - \frac{h_{j+1}}{6} (M_{j+1} - M_j). \end{aligned}$$

Sustituyendo estos valores en (2.16) obtenemos la representación deseada para s en términos de los M_j , esto es

$$s(x) = \frac{M_j}{6} \left[\frac{(x_{j+1} - x)^3}{h_{j+1}} - h_{j+1}^2 \right] + M_{j+1} \frac{(x - x_j)^3}{6h_{j+1}} + \left[\frac{y_{j+1} - y_j}{h_{j+1}} - \frac{h_{j+1}}{6} (M_{j+1} - M_j) \right] (x - x_j) + y_j. \quad (2.17)$$

Ahora, utilizando la continuidad de s' podemos obtener $n - 1$ ecuaciones para determinar M_j , reemplazando A_j en (2.15) obtenemos que

$$\begin{aligned} s'(x) &= -M_{j-1} \frac{(x_j - x)^2}{2h_j} + M_j \frac{(x - x_{j-1})^2}{2h_j} + \frac{y_j - y_{j-1}}{h_j} - \frac{h_j}{6} (M_j - M_{j-1}), \\ s'(x) &= -M_j \frac{(x_{j+1} - x)^2}{2h_{j+1}} + M_{j+1} \frac{(x - x_j)^2}{2h_{j+1}} + \frac{y_{j+1} - y_j}{h_{j+1}} - \frac{h_{j+1}}{6} (M_{j+1} - M_j), \end{aligned}$$

para $x \in [x_{j-1}, x_j]$, $[x_j, x_{j+1}]$, respectivamente. Puesto que

$$\begin{aligned} s'(x_j^-) &= M_j \frac{h_j}{2} + \frac{y_j - y_{j-1}}{h_j} - \frac{h_j}{6} (M_j - M_{j-1}) \\ &= M_j \frac{h_j}{3} + M_{j-1} \frac{h_j}{6} + \frac{y_j - y_{j-1}}{h_j} \\ s'(x_j^+) &= -M_j \frac{h_{j+1}}{2} + \frac{y_{j+1} - y_j}{h_{j+1}} - \frac{h_{j+1}}{6} (M_{j+1} - M_j) \\ &= -M_j \frac{h_{j+1}}{3} - M_{j+1} \frac{h_{j+1}}{6} + \frac{y_{j+1} - y_j}{h_{j+1}}, \end{aligned}$$

la condición $s'(x_j^-) = s'(x_j^+)$ produce $n - 1$ ecuaciones, de las $n + 1$ necesarias, para calcular $M_j, j = 1, 2, \dots, n - 1$

$$M_j \frac{h_{j+1} + h_j}{3} + M_{j-1} \frac{h_j}{6} + M_{j+1} \frac{h_{j+1}}{6} = \frac{y_{j+1} - y_j}{h_{j+1}} - \frac{y_j - y_{j-1}}{h_j} \quad (2.18)$$

las condiciones necesarias restantes para obtener M_j se obtienen de las condiciones de frontera 1, 2, o 3. Para los splines naturales se reemplaza en (2.18) los términos

$$M_0 = s''(a) = 0, \quad M_n = s''(b) = 0.$$

Reescribiendo (2.18) como

$$M_j + \mu_j M_{j-1} + \lambda_j M_{j+1} = d_j$$

donde,

$$\mu_j := \frac{h_j}{2(h_{j+1} + h_j)}, \quad \lambda_j := \frac{h_{j+1}}{2(h_{j+1} + h_j)}, \quad d_j := \frac{3}{h_{j+1} + h_j} \left(\frac{y_{j+1} - y_j}{h_{j+1}} - \frac{y_j - y_{j-1}}{h_j} \right),$$

podemos plantear el sistema de ecuaciones de la forma $Ax = b$, esto es

$$\begin{bmatrix} 1 & \lambda_1 & & & \\ \mu_2 & 1 & \lambda_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \lambda_{n-2} \\ & & & \mu_{n-1} & 1 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ \vdots \\ M_{n-1} \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{n-1} \end{bmatrix}$$

Dado que $\mu_j + \lambda_j = \frac{1}{2}, j = 1, \dots, n-1$, la matriz de coeficientes del sistema es diagonal dominante, luego, podemos aplicar el sistema de eliminación de Gauss sin pivote y por tanto la matriz es invertible, esto significa que el sistema tiene solución única para b arbitrario y en consecuencia el problema de interpolación a través de splines cúbicos (naturales) tiene solución única [23].

2.4.3. Convergencia de los splines cúbicos

Las propiedades de convergencia de las funciones spline cúbicas se establecen con el siguiente resultado [21], el cual requiere la siguiente definición previa:

Definición 1 (Módulo de continuidad). Sea $f : X \rightarrow Y$. Se define el módulo de continuidad $\omega(f, \cdot) : (0, \infty) \rightarrow [0, \infty]$ mediante

$$\omega(f, \Delta) := \sup \{ |f(x) - f(y)| : x, y \in X, |x - y| \leq \Delta \}.$$

Teorema 8. Sea $f \in C^3([a, b])$ y sea $s \in S_3(\Delta)$ el spline interpolante que satisface una de las condiciones de frontera 1, 2 o $s''(a) = f''(a), s''(b) = f''(b)$. Si f''' es Lipschitz continua, entonces

$$\|s^{(k)} - f^{(k)}\| \leq C_k \bar{\Delta}^{3-k} \omega(f''', \bar{\Delta}), \quad k = 0, 1, 2, \quad (2.19)$$

donde $C_0 = \frac{9}{8}, C_1 = C_2 = \frac{9}{4}$, y $\omega(f''', \bar{\Delta})$ denota el módulo de continuidad de f''' en $\bar{\Delta}$.

Demostración. Consideraremos aquí el caso en el cual se cumple la condición de frontera 2, los demás casos se demuestran de forma análoga. Si s satisface la condición 2 entonces las $n - 1$ ecuaciones dadas por (2.18) junto a las ecuaciones

$$\frac{h_1}{3}M_0 + \frac{h_1}{6}M_1 = \frac{y_1 - y_0}{h_1} - y'_0, \quad (2.20)$$

$$\frac{h_n}{6}M_{n-1} + \frac{h_n}{3}M_n = y'_n - \frac{y_n - y_{n-1}}{h_n}, \quad (2.21)$$

forman el sistema de n ecuaciones que determina a s de forma única en términos de los valores de sus segundas derivadas M_j . Definamos $z_j := M_j - f''(x_j)$ y $R := \max\{|z_j| : j = 0, \dots, n\}$. Supondremos que se alcanzará dicho máximo para $j \in 0, \dots, n$, así, $R = |z_j|$ para algún j . Si $j \in 1, \dots, n - 1$, entonces por (2.18) se tiene que

$$\frac{h_j}{6}z_{j-1} + \frac{h_j + h_{j+1}}{3}z_j + \frac{h_{j+1}}{6}z_{j+1} = p_j, \quad (2.22)$$

donde

$$p_j := \frac{y_{j+1} - y_j}{h_{j+1}} + \frac{y_j - y_{j-1}}{h_j} - \frac{h_j}{6}f''(x_{j-1}) - \frac{h_j + h_{j+1}}{3}f''(x_j) - \frac{h_{j+1}}{6}f''(x_{j+1}).$$

Aplicando el teorema de Taylor en la ecuación anterior sobre las funciones f y f'' alrededor de $x = x_j$ con adecuados $\xi_j, \varrho_j \in [x_{j-1}, x_j]$ y $\eta_j, \sigma_j \in [x_j, x_{j+1}]$ obtenemos que

$$p_j = -\frac{h_j^2}{6}(f'''(\xi_j) - f'''(\varrho_j)) + \frac{h_{j+1}^2}{6}(f'''(\eta_j) - f'''(\sigma_j)),$$

por lo tanto,

$$|p_j| \leq \frac{h_j^2 + h_{j+1}^2}{6}\omega(f''', \bar{\Delta}).$$

De otro lado, de la ecuación (2.22) se obtiene que

$$|p_j| \geq -\frac{h_j}{6}|z_{j-1}| + \frac{h_j + h_{j+1}}{3}|z_j| - \frac{h_{j+1}}{6}|z_{j+1}| \geq \frac{h_j + h_{j+1}}{6}R,$$

así,

$$\frac{h_j + h_{j+1}}{6}R \leq \frac{h_j^2 + h_{j+1}^2}{6}\omega(f''', \bar{\Delta}),$$

y dado que $h_j^2 + h_{j+1}^2 \leq (h_j + h_{j+1})^2$, concluimos que

$$R \leq 2\bar{\Delta}\omega(f''', \bar{\Delta}). \quad (2.23)$$

Supongamos ahora que el máximo se alcanza cuando $j = 0$, esto es, $R = |z_0|$. Por la ecuación (2.20) tenemos que

$$\frac{h_1}{3}z_0 + \frac{h_1}{6}z_1 = p_0, \quad (2.24)$$

donde

$$p_0 := \frac{y_1 - y_0}{h_1} - y'_0 - \frac{h_1}{3} f''(x_0) - \frac{h_1}{6} f''(x_1).$$

Aplicando el teorema de Taylor sobre f y f'' alrededor de $x = x_0$ con adecuados $\eta_0, \sigma_0 \in [x_0, x_1]$ obtenemos que

$$p_0 = \frac{h_1^2}{6} (f'''(\eta_0) - f'''(\sigma_0)),$$

por lo tanto

$$|p_0| \leq \frac{h_1^2}{6} \omega(f''', \bar{\Delta}).$$

De otro lado, de la ecuación (2.24) se tiene que

$$|p_0| = \left| \frac{z_0}{3} h_1 + \frac{z_1}{6} h_1 \right| \geq \frac{|z_0|}{3} h_1 - \frac{|z_1|}{6} h_1 = \frac{R}{3} h_1 - \frac{|z_1|}{6} h_1 \geq \frac{R}{3} h_1 - \frac{R}{6} h_1,$$

así,

$$|p_0| \geq \frac{R}{6} h_1,$$

y podemos concluir que

$$R \leq h_1 \omega(f''', \bar{\Delta}) \leq \bar{\Delta} \omega(f''', \bar{\Delta}) \leq 2\bar{\Delta} \omega(f''', \bar{\Delta}). \quad (2.25)$$

Finalmente supongamos que el máximo se alcanza cuando $i = n$, esto es, $R = |z_n|$. Por la ecuación (2.21) tenemos que

$$\frac{h_n}{6} z_{n-1} + \frac{h_n}{3} z_n = p_n, \quad (2.26)$$

donde

$$p_n := y'_n - \frac{y_n - y_{n-1}}{h_n} - \frac{h_n}{6} f''(x_{n-1}) - \frac{h_n}{3} f''(x_n).$$

Aplicando el teorema de Taylor sobre f y f'' alrededor de $x = x_n$ con adecuados $\xi_n, \varrho_n \in [x_{n-1}, x_n]$ obtenemos que

$$p_n = \frac{h_n^2}{6} (f'''(\xi_n) - f'''(\varrho_n)),$$

por lo tanto

$$|p_n| \leq \frac{h_n^2}{6} \omega(f''', \bar{\Delta}).$$

De otro lado, de la ecuación (2.26) se tiene que

$$|p_n| = \left| \frac{h_n}{6} z_{n-1} + \frac{h_n}{3} z_n \right| \geq \frac{|z_n|}{3} h_n - \frac{|z_{n-1}|}{6} h_n = \frac{R}{3} h_n - \frac{|z_{n-1}|}{6} h_n \geq \frac{R}{3} h_n - \frac{R}{6} h_n,$$

así,

$$|p_n| \geq \frac{R}{6} h_n,$$

y podemos concluir que

$$R \leq h_n \omega(f''', \bar{\Delta}) \leq \bar{\Delta} \omega(f''', \bar{\Delta}) \leq 2\bar{\Delta} \omega(f''', \bar{\Delta}). \quad (2.27)$$

De (2.23), (2.25), y (2.27) se tiene entonces que

$$|M_j - f''(x_j)| = |z_j| \leq 2\bar{\Delta} \omega(f''', \bar{\Delta}), \quad j = 0, 1, \dots, n. \quad (2.28)$$

Consideremos ahora $x \in [x_j, x_{j+1}]$. Derivando dos veces (2.17) y restando $f''(x)$ se obtiene que

$$s''(x) - f''(x) = \frac{(x_{j+1} - x)}{h_{j+1}}(z_j + f''(x_j) - f''(x)) + \frac{(x - x_j)}{h_{j+1}}(z_{j+1} + f''(x_{j+1}) - f''(x)),$$

aplicando el teorema de Taylor en esta ecuación sobre $f''(x)$ alrededor de $x = x_j$ y $x = x_{j+1}$ con adecuados $\xi_j, \eta_j \in [x_j, x_{j+1}]$ obtenemos que

$$s''(x) - f''(x) = \frac{(x_{j+1} - x)}{h_{j+1}}z_j + \frac{(x - x_j)}{h_{j+1}}z_{j+1} + \frac{(x_{j+1} - x)(x - x_j)}{h_{j+1}}(f'''(\xi_j) - f'''(\eta_j)),$$

luego, por (2.28) se tiene entonces que

$$|s''(x) - f''(x)| \leq 2\bar{\Delta} \omega(f''', \bar{\Delta}) + \frac{\bar{\Delta}}{4} \omega(f''', \bar{\Delta}) = \frac{9}{4} \bar{\Delta} \omega(f''', \bar{\Delta}),$$

la cual corresponde a la afirmación (2.19) para $k = 2$.

Ahora bien, dado que $s(x_j) = f(x_j)$, para $j = 0, \dots, n$, entonces por el teorema de Rolle existen $\xi_j \in [x_j, x_{j+1}]$ tales que $s'(\xi_j) - f'(\xi_j) = 0$, y así, para cada $x \in [a, b]$ existe entonces ξ_j tal que $|x - \xi_j| \leq \bar{\Delta}$, luego

$$s'(x) - f'(x) = \int_{\xi_j}^x s''(t) - f''(t) dt \leq \int_{\xi_j}^x |s''(t) - f''(t)| dt \leq \int_{\xi_j}^x \frac{9}{4} \bar{\Delta} \omega(f''', \bar{\Delta}) dt = \frac{9}{4} \bar{\Delta} \omega(f''', \bar{\Delta})(x - \xi_j),$$

es decir,

$$|s'(x) - f'(x)| \leq \frac{9}{4} \bar{\Delta}^2 \omega(f''', \bar{\Delta}),$$

obteniendo así la afirmación (2.19) para $k = 1$.

Finalmente, como para cada $x \in [a, b]$ existe $x_j, j = 0, \dots, n$ tal que $|x - x_j| \leq \frac{\bar{\Delta}}{2}$, entonces

$$s(x) - f(x) = \int_{x_j}^x s'(t) - f'(t) dt \leq \int_{x_j}^x |s'(t) - f'(t)| dt \leq \int_{x_j}^x \frac{9}{4} \bar{\Delta}^2 \omega(f''', \bar{\Delta}) dt = \frac{9}{4} \bar{\Delta}^2 \omega(f''', \bar{\Delta})(x - x_j),$$

es decir,

$$|s(x) - f(x)| \leq \frac{9}{8} \bar{\Delta}^3 \omega(f''', \bar{\Delta}),$$

obteniendo de esta forma la afirmación (2.19) para $k = 0$. □

Las constantes C_k pueden tener incluso valores más pequeños que mejoran el estimativo del error: $C_0 = \frac{5}{384}$, $C_1 = \frac{1}{24}$, y $C_2 = \frac{3}{8}$. Más aún, en dicho caso C_0 y C_1 son óptimos [15].

Las tablas (2.1) y (2.2) resumen los aspectos más importantes de los métodos de interpolación presentados hasta ahora. Acto seguido, presentamos un ejemplo que permite contrastar los resultados obtenidos al interpolar una muestra de datos con dos de los métodos presentados anteriormente.

Método	Fórmula asociada
Lagrange	$P_n(x) = \sum_{k=0}^n y_k L_{n,k}(x), \text{ con } L_{n,k}(x) = \prod_{\substack{j=0 \\ j \neq k}}^n \frac{(x - x_j)}{(x_k - x_j)}$
Newton	$P_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + \cdots + f[x_0, \dots, x_n] \prod_{i=0}^{n-1} (x - x_i)$
Hermite	$p(x) = \sum_{j=0}^n f_j \varphi_j(x) + \sum_{j=0}^n f'_j \Psi_j(x),$ $\varphi_j(x) := (1 - 2L'_{n,j}(x_j)(x - x_j))L_{n,j}^2(x),$ $\Psi_j(x) := (x - x_j)L_{n,j}^2(x)$
Splines	$s(x) = \frac{M_j}{6} \left[\frac{(x_{j+1} - x)^3}{h_{j+1}} - h_{j+1}^2 \right] + M_{j+1} \frac{(x - x_j)^3}{6h_{j+1}}$ $+ \left[\frac{y_{j+1} - y_j}{h_{j+1}} - \frac{h_{j+1}}{6} (M_{j+1} - M_j) \right] (x - x_j) + y_j,$ $M_j := s''(x_j), j = 0, \dots, n$

Tabla 2.1: Métodos de interpolación y fórmulas asociadas.

Método	Error asociado
Lagrange	$f(x) - P_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi) w(x)$
Newton	$f(x) - P_n(x) = f[x_0, \dots, x_{n+1}] w(x)$
Hermite	$f(x) - P_n(x) = \frac{w^2(x)}{(2n+2)!} f^{(2n+2)}(\zeta)$
Splines	$ f(x) - s(x) \leq \frac{9}{8} \bar{\Delta}^3 \omega(f''', \bar{\Delta})$

Tabla 2.2: Métodos de interpolación y errores asociados.

Ejemplo comparativo

Para poder visibilizar de manera más clara la discusión analítica realizada previamente, es necesario mostrar los resultados que se obtienen al aproximar curvas con los métodos de interpolación presentados. La figura (2.3) muestra a **Anger**, personaje de la película animada *Inside Out*, en este ejemplo se aproximó el contorno superior de su silueta, para ello se ubicaron puntos (x_i, y_i) iniciales sobre el mismo y progresivamente se introdujeron nodos intermedios para refinar la partición.

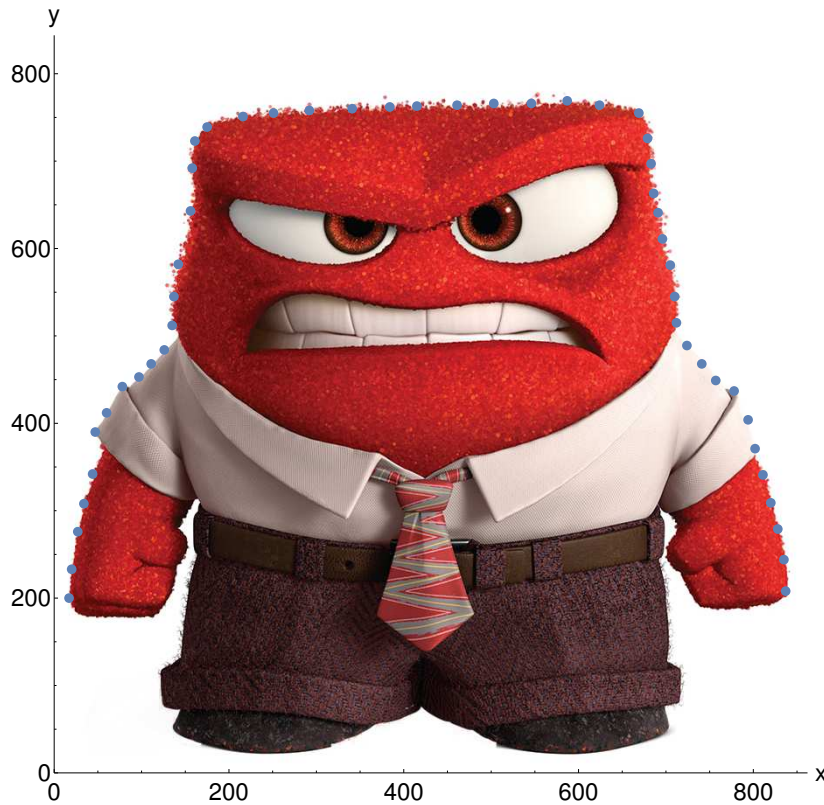


Figura 2.3: Anger ©Pixar Animation Studios.

En primera instancia se realizó la interpolación a través de splines; se situaron un total de 49 nodos iniciales sobre el contorno superior para la primera aproximación del mismo, posterior a esto se refinó la partición agregando los nodos promedio $(\bar{x}, \bar{y})$ entre cada par de nodos consecutivos (x_i, y_i) y (x_{i+1}, y_{i+1}) , lo anterior se realizó tres veces, obteniendo un total de 97, 193, y 385 nodos a interpolar en el primero, segundo, y tercer refinamiento, respectivamente. La figura (2.4) muestra al personaje junto a las aproximaciones de su silueta obtenidas en cada caso.

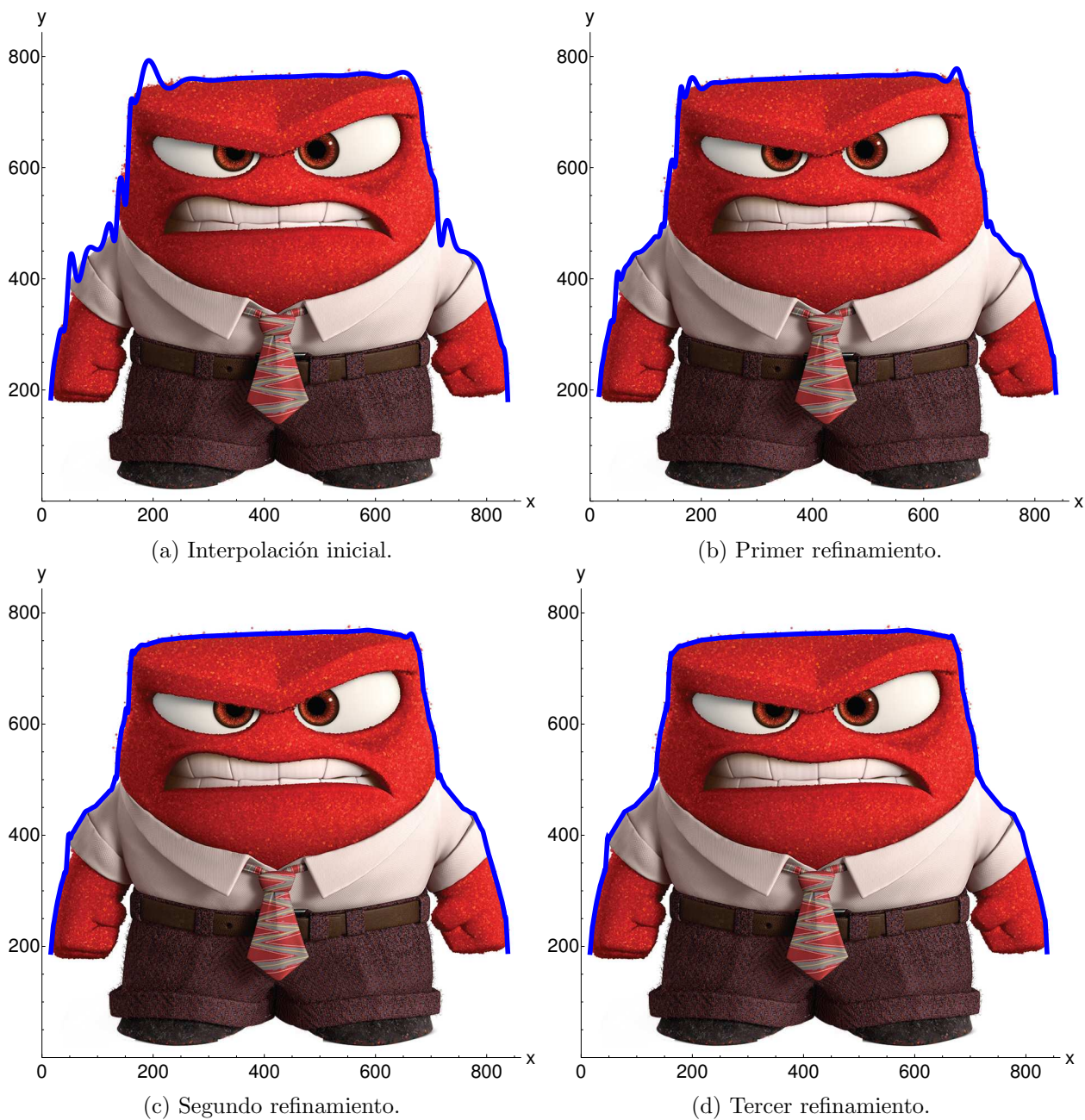


Figura 2.4: La curva interpolada es el contorno superior de **Anger**.

Se puede ver claramente en este caso como cada refinamiento consecutivo hace que el polinomio spline se “pegue” cada vez más a la silueta del personaje; de hecho después del tercer refinamiento la diferencia entre la misma y su aproximación se reduce considerablemente.

Por otro lado, al realizar la interpolación utilizando el método de las diferencias divididas de Newton, y considerando los mismos 49 nodos iniciales sobre la silueta se obtiene el polinomio interpolante de grado 48 de la figura (2.5a). Este polinomio en efecto pasa por cada uno de los nodos situados originalmente sobre el contorno del personaje, pero la oscilación es tan

alta en la región central que no es posible visualizar el personaje. Observe que los valores del polinomio en dicha región son del orden de 10^9 , mientras que los de los nodos alcanzan poco menos del orden de 10^2 , así que para tener una visualización más clara de lo que ocurre es necesario restringir el rango del polinomio a este orden, lo anterior se puede apreciar en la figura¹ (2.5b).

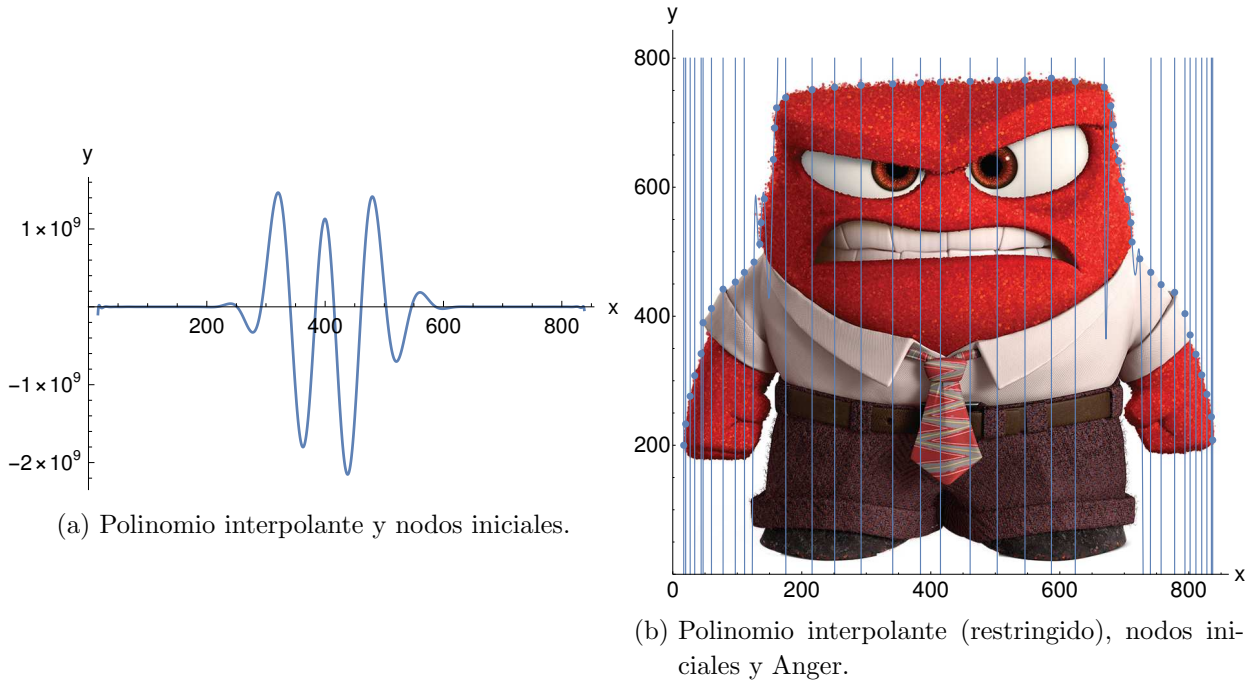


Figura 2.5: Interpolación polinómica del contorno superior de **Anger**.

La aproximación anterior no converge de ninguna forma hacia la silueta del personaje, de hecho si se refina la partición inicial y se recalcula el polinomio se obtendrá uno de oscilaciones incluso mucho mayores, así, se muestra que este método no es conveniente para la aproximación de esta muestra de datos. Este ejemplo ilustra entonces de manera clara una de las propiedades más importantes de los splines, la convergencia con base en el refinamiento. Es precisamente esta propiedad la que hace que estas funciones sean predilectas a la hora de aproximar funciones que interpolan muestras de datos en un sin número de experimentos, lo anterior junto con la fácil implementación computacional y la estabilidad de los procesos computacionales asociados, han hecho que estas funciones tomen un marcado protagonismo en los procesos de diseño geométrico, pues en ellos estas características son de vital importancia para crear las herramientas que permitan concebir productos que satisfagan todo tipo de necesidades presentes en las diferentes industrias.

Como lo mencionamos al inicio de esta sección las funciones splines forman un espacio vecto-

¹Las gráficas de este capítulo fueron generadas utilizando las funciones *InterpolatingPolynomial* e *Interpolation* que vienen implementadas en el software *Wolfram Mathematica 12*.

rial de dimensión finita, las funciones *B-spline* que presentamos a continuación forman una base para dicho espacio, además de esto, ellas poseen importantes propiedades que las hacen ideales para la construcción de curvas y superficies paramétricas que son empleadas a fondo por los sistemas de diseño geométrico asistido por computadora.

2.4.4. B-Splines

Las funciones spline son casos particulares de funciones polinomiales a trozos asociadas a una partición

$$\Delta = \{a = x_0 < x_1 < \cdots < x_n = b\}$$

de un intervalo $[a, b]$ por nodos x_i . En general, una función real $f : [a, b] \rightarrow \mathbb{R}$ es llamada *función polinomial a trozos* de orden r o grado $r-1$ si, para cada $i = 0, \dots, n-1$, la restricción de f al subintervalo (x_i, x_{i+1}) coincide con un polinomio $p_i(x)$ de grado $\leq r-1$, $p_i \in \prod_{r-1}$. Para lograr una correspondencia 1-1 entre f y la secuencia $(p_0(x), p_1(x), \dots, p_{n-1}(x))$, definimos f en los nodos $x_i, i = 0, \dots, n-1$, de tal forma que sea continua por la derecha, $f(x_i) := f(x_i^+), 0 \leq i \leq n-1$, y $f(x_n) = f(b) := f(x_n^-)$.

Luego, al generalizar los splines cúbicos, se definen funciones spline S_Δ de grado k como funciones polinomiales a trozos de grado k que son $k-1$ veces diferenciables al interior de los nodos $x_i, 1 \leq i \leq n-1$, de Δ . Por $S_{\Delta,k}$ se denota el conjunto de todas las funciones spline S_Δ de grado k , el cual es un espacio vectorial real de dimensión $n+k$. En efecto, el polinomio $S_{\Delta,k}|_{[x_0, x_1]}$ está determinado únicamente por sus $k+1$ coeficientes, lo cual a su vez fija las $k-1$ derivadas ($= k$ condiciones) del siguiente polinomio $S_\Delta|_{[x_1, x_2]}$ en x_1 , así que solo un grado de libertad queda disponible para elegir $S_\Delta|_{[x_1, x_2]}$. Pasa exactamente lo mismo en todos los subsecuentes polinomios $S_\Delta|_{[x_i, x_{i+1}]}, i = 2, \dots, n-1$, se encuentra entonces que $\dim S_{\Delta,k} = k+1 + (n-1) \cdot 1 = k+n$.

Las B-splines son funciones polinomiales a trozos con notables propiedades: son no negativas y se desvanecen en todas partes, excepto en unos pocos intervalos contiguos de la forma $[x_i, x_{i+1}]$. Más aún, el espacio de funciones $S_{\Delta,k}$ tiene una base que consiste de B-splines. Como consecuencia, las B-splines proveen un marco para el cálculo eficiente y numéricamente estable de splines.

Para definir las B-splines, se introduce la función $f : \mathbb{R} \rightarrow \mathbb{R}$

$$f_x(t) := (t-x)_+ := \max(t-x, 0) = \begin{cases} t-x, & \text{para } t > x \\ 0, & \text{para } t \leq x \end{cases}$$

y sus potencias $f_x^r, r \geq 0$, en particular para $r = 0$,

$$f_x^0 := \begin{cases} 1, & \text{para } t > x \\ 0, & \text{para } t \leq x. \end{cases}$$

La función $f_x^r(\cdot)$ está compuesta por dos polinomios de grado $\leq r$: el polinomio $P_0(t) := 0$ para $t \leq x$ y el polinomio $P_1(t) := (t - x)^r$ para $t > x$. Note que $f_x^r(\cdot)$ depende de un parámetro real x , y la definición de $f_x^r(t)$ es la de una función de x la cual es continua por la derecha para cada valor de t fijado. Claramente, para $r \geq 1$, $f_x^r(t)$ es continuamente diferenciable $(r - 1)$ veces tanto respecto a t como a x .

Bajo ciertas condiciones la diferencia dividida generalizada $f[t_i, t_{i+1}, \dots, t_{i+r}]$ de una función real $f(t)$, está bien definida para cualquier segmento de números reales $t_i \leq t_{i+1} \leq \dots \leq t_{i+r}$, incluso si los t_j no son mutuamente distintos. La única condición es que f sea $(s_j - 1)$ veces diferenciable en $t = t_j$, $j = i, i+1, \dots, i+r$, si el valor de t_j ocurre s_j veces en el subsegmento $t_i \leq t_{i+1} \leq \dots \leq t_j$ terminado en t_j . En este caso, la diferencia dividida de una función está definida como

$$f[t_i, t_{i+1}, \dots, t_{i+r}] := \frac{f^{(r)}(t_i)}{r!}, \quad \text{si } t_i = t_{i+r} \quad (2.29)$$

$$f[t_i, t_{i+1}, \dots, t_{i+r}] := \frac{f[t_{i+1}, \dots, t_{i+r}] - f[t_i, \dots, t_{i+r-1}]}{t_{i+r} - t_i}, \quad \text{en otro caso.} \quad (2.30)$$

La definición anterior permite además ver que las diferencias divididas de la función f son combinaciones lineales de sus derivadas en los puntos t_j

$$f[t_i, t_{i+1}, \dots, t_{i+r}] = \sum_{j=i}^{i+r} \alpha_j f^{(s_j-1)}(t_j), \quad (2.31)$$

donde, en particular, $\alpha_{i+r} \neq 0$. La justificación de la ecuación (2.31) recae en un proceso inductivo sobre r en el cual el paso base, $r = 0$, se obtiene de forma inmediata por la primera línea de la definición, (2.29), mientras que al considerar el paso inductivo, $r = k \Rightarrow r = k+1$, se deben tener en cuenta los casos en los cuales los valores t_j aparezcan s_j veces en el subsegmento $t_{i+1} \leq t_{i+2} \leq \dots \leq t_{i+k+1}$.

Sea $r \geq 1$ un entero y $\mathbf{t} = \{t_i\}_{i \in \mathbb{Z}}$ cualquier sucesión infinita de números reales t_i , no decreciente y no acotada, tal que

$$\inf t_i = -\infty, \quad \sup t_i = +\infty, \quad \text{y} \quad t_i < t_{i+r} \text{ para todo } i \in \mathbb{Z}$$

Se define la i -ésima B-spline de orden r asociada con la sucesión $\mathbf{t}$ como la función de x dada por

$$B_{i,r,\mathbf{t}}(x) := (t_{i+r} - t_i) f_x^{r-1}[t_i, t_{i+1}, \dots, t_{i+r}]. \quad (2.32)$$

Claramente $B_{i,r,\mathbf{t}}(x)$ está bien definida para todo $x \neq t_i, t_{i+1}, \dots, t_{i+r}$ y por (2.31), se tiene que (2.32) es una combinación lineal de las funciones

$$(t_j - x)_+^{r-s_j} \Big|_{t=t_j}, \quad i \leq j \leq i+r, \quad (2.33)$$

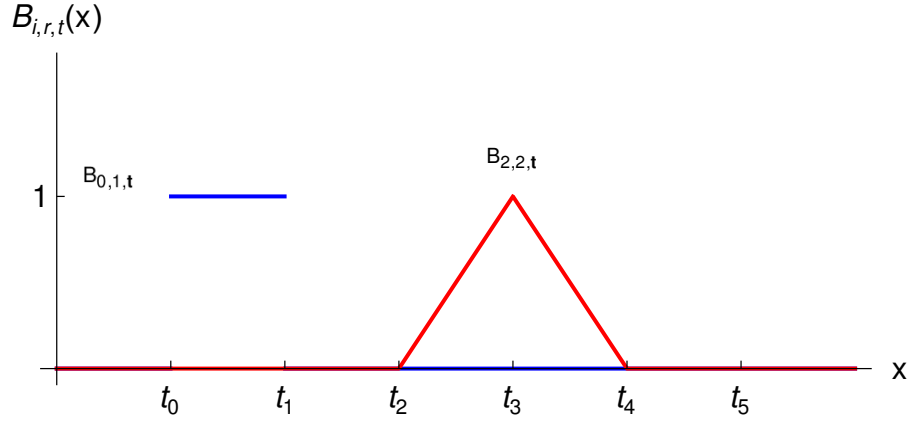


Figura 2.6: Ejemplos B-spline.

si el valor de t_j ocurre s_j veces dentro del subsegmento $t_i \leq t_{i+1} \leq \dots \leq t_j$. Si además el valor de t_j ocurre n_j veces dentro del segmento completo $t_i \leq t_{i+1} \leq \dots \leq t_{i+r}$, entonces la definición de s_j muestra que para cada índice σ con $1 \leq \sigma \leq n_j$ existe exactamente un entero $l = l(\sigma)$ que satisface

$$t_l = t_j, \quad s_l = \sigma, \quad i \leq l \leq i + r.$$

Así, $B_{i,r,t}$ es también una combinación lineal de

$$(t_j - x)_+^s, \quad \text{donde } r - n_j \leq s \leq r - 1, i \leq j \leq i + r.$$

Lo anterior muestra entonces que la función $B_{i,r,t}(x)$ coincide con un polinomio de grado a lo más $r - 1$ sobre cada uno de los intervalos abiertos del conjunto

$$(-\infty, t_i) \cup \{(t_j, t_{j+1}), i \leq j < i + r; t_j < t_{j+1}\} \cup (t_{i+r}, +\infty),$$

es decir, $B_{i,r,t}(x)$ es una función polinomial a trozos de orden r con respecto a la partición del eje real dada por los puntos t_k , con $k \in T_{i,r}$, donde

$$T_{i,r} := \{j : i \leq j < i + r; t_j < t_{j+1}\} \cup \{i + r\}.$$

El siguiente teorema da cuenta de las propiedades de mayor importancia de las funciones B-spline:

Teorema 9.

- i.* $B_{i,r,t}(x) = 0$ para $x \notin [t_i, t_{i+r}]$
- ii.* $B_{i,r,t}(x) > 0$ para $x \in (t_i, t_{i+r})$
- iii.* Para todo $x \in \mathbb{R}$ se cumple que

$$\sum_i B_{i,r,t}(x) = 1,$$

y la suma contiene solo finitos términos no nulos.

Demostración.

- i.* Para $x < t_i \leq t \leq t_{i+r}$, se tiene que $f_x^{r-1}(t) = (t-x)^{r-1}$ es un polinomio de grado $r-1$ en t , el cual tiene una r -ésima diferencia dividida que se anula

$$f_x^{r-1}[t_i, t_{i+1}, \dots, t_{i+r}] = 0 \implies B_{i,r}(x) = 0$$

Aclaremos lo anterior con un ejemplo, consideremos la sucesión $\mathbf{t} = \{t_1, t_2, t_3, t_4\}$ tales que

$$x < t_1 < t_2 < t_3 < t_4$$

y la segunda B-spline de orden 2 asociada a la misma:

$$\begin{aligned} B_{2,2,\mathbf{t}}(x) &= (t_4 - t_2) f_x[t_2, t_3, t_4] \\ &= (t_4 - t_2) \frac{f_x[t_3, t_4] - f_x[t_2, t_3]}{t_4 - t_2} \\ &= \frac{f_x[t_4] - f_x[t_3]}{t_4 - t_3} - \frac{f_x[t_3] - f_x[t_2]}{t_3 - t_2} \\ &= \frac{f_x(t_4) - f_x(t_3)}{t_4 - t_3} - \frac{f_x(t_3) - f_x(t_2)}{t_3 - t_2} \\ &= \frac{\begin{cases} t_4 - x, & t_4 > x \\ 0, & t_4 \leq x \end{cases} - \begin{cases} t_3 - x, & t_3 > x \\ 0, & t_3 \leq x \end{cases}}{t_4 - t_3} - \frac{\begin{cases} t_3 - x, & t_3 > x \\ 0, & t_3 \leq x \end{cases} - \begin{cases} t_2 - x, & t_2 > x \\ 0, & t_2 \leq x \end{cases}}{t_3 - t_2} \\ &= \frac{(t_4 - x) - (t_3 - x)}{t_4 - t_3} - \frac{(t_3 - x) - (t_2 - x)}{t_3 - t_2} \\ &= \frac{t_4 - t_3}{t_4 - t_3} - \frac{t_3 - t_2}{t_3 - t_2} \\ &= 0 \end{aligned}$$

Si por otro lado consideramos x tal que

$$t_i \leq t \leq t_{i+r} < x$$

trivialmente se tendría que $f_x^{r-1} = 0$; todas las funciones de potencia truncadas se anularían, pues $x > t_j; j \in \{t_i, t_{i+1}, \dots, t_{i+r}\}$.

ii. Para $r = 1$ y $x \in (t_i, t_{i+r})$ se tiene por definición que

$$\begin{aligned}
 B_{i,1,t} &= (t_{i+1} - t_i) f_x^{1-1} [t_i, t_{i+1}] \\
 &= (t_{i+1} - t_i) f_x^0 [t_i, t_{i+1}] \\
 &= (t_{i+1} - t_i) \frac{f_x^0 [t_{i+1}] - f_x^0 [t_i]}{t_{i+1} - t_i} \\
 &= f_x^0 (t_{i+1}) - f_x^0 (t_i) \\
 &= \begin{cases} (t_{i+1} - x)^0, & t_{i+1} > x \\ 0, & t_{i+1} \leq x \end{cases} - \begin{cases} (t_i - x)^0, & t_i > x \\ 0, & t_i \leq x \end{cases} \\
 &= 1 - 0 \\
 &= 1.
 \end{aligned}$$

Si $r > 1$ definamos

$$N_{i,r}(x) = \frac{B_{i,r}(x)}{t_{i+r} - t_i} \equiv f_x^{r-1} [t_i, \dots, t_{i+r}]. \quad (2.34)$$

Para $r \geq 2$, con $x \in (t_i, t_{i+r})$ se cumple la siguiente recursión

$$N_{i,r}(x) = \frac{x - t_i}{t_{i+r} - t_i} N_{i,r-1}(x) + \frac{t_{i+r} - x}{t_{i+r} - t_i} N_{i+1,r-1} \quad (2.35)$$

Dado que $x \in (t_i, t_{i+r})$, se cumple que $x - t_i > 0$ y $t_{i+r} > t_i$, luego, $\frac{x-t_i}{t_{i+r}-t_i} > 0$; por otro lado, $t_{i+r} > x$, así, $t_{i+r} - x > 0$, luego, $\frac{t_{i+r}-x}{t_{i+r}-t_i} > 0$. Además, $N_{i,r}(x)$ tiene el mismo signo que $B_{i,r}(x)$, y dado que $B_{i,1}(x) > 0$ para $x \in (t_i, t_{i+r})$, inducción sobre r utilizando la recursión definida anteriormente muestra que $B_{i,r} > 0$ para $x \in (t_i, t_{i+r})$.

iii. Supongamos primero que $t_j < x < t_{j+1}$. Entonces, por **i.**, $B_{i,r}(x) = 0$ para todo i, r tal que $i + r < j$ y $i \geq j + 1$, es decir, todos aquellos intervalos que están antes y después de (t_j, t_{j+1}) . Luego,

$$\sum_i B_{i,r}(x) = \sum_{i=j-r+1}^j B_{i,r}(x)$$

Por la definición de diferencias divididas y de función B-spline, se tiene que

$$B_{i,r}(x) = f_x^{r-1} [t_{i+1}, \dots, t_{i+r}] - f_x^{r-1} [t_i, \dots, t_{i+r-1}]$$

Luego,

$$\begin{aligned}
\sum_j B_{i,r}(x) &= \sum_{i=j-r+1}^j f_x^{r-1}[t_{i+1}, \dots, t_{i+r}] - f_x^{r-1}[t_i, \dots, t_{i+r-1}] \\
&= f_x^{r-1}[t_{j-r+2}, \dots, t_{j+1}] - f_x^{r-1}[t_{j-r+1}, \dots, t_j] \\
&\quad + f_x^{r-1}[t_{j-r+3}, \dots, t_{j+2}] - f_x^{r-1}[t_{j-r+2}, \dots, t_{j+1}] \\
&\quad \vdots \\
&\quad + f_x^{r-1}[t_j, \dots, t_{j-1+r}] - f_x^{r-1}[t_{j-1}, \dots, t_{j-1+r-1}] \\
&\quad + f_x^{r-1}[t_{j+1}, \dots, t_{j+r}] - f_x^{r-1}[t_j, \dots, t_{j+r-1}] \\
\sum_j B_{i,r}(x) &= f_x^{r-1}[t_{j+1}, \dots, t_{j+r}] - f_x^{r-1}[t_{j-r+1}, \dots, t_j] \\
&= 1 - 0
\end{aligned}$$

La última igualdad se tiene porque la función $f_x^{r-1}(t) = (t-x)^{r-1}$ es un polinomio de grado en t para $t_j < x < t_{j+1} \leq t \leq t_{j+r}$, por lo cual $f_x^{r-1}[t_{j+1}, \dots, t_{j+r}] = 1$ (por (2.10)); y la función $f_x^{r-1}(t) = (t-x)^{r-1} = 0$ se anula para $t_{j-r+1} \leq t \leq t_j < x < t_{j+1}$. Para $t = t_j$, la afirmación se sigue por la continuidad de $B_{i,r}$ por la derecha,

$$B_{i,r}(t_j) = \lim_{y \rightarrow t_j} B_{i,r}(y).$$

□

Ahora, una de las características más importantes de las funciones B-spline es que forman una base para el espacio $S_{\Delta,k}$. Para mostrar esto es necesario construir una sucesión $\mathbf{t} = (t_j)$ de tal forma que las B-splines $B_{i,k+1,\mathbf{t}}(x)$ de orden $k+1$ formen una base para dicho espacio. Con esto en mente, asociaremos la partición

$$\Delta = \{a = x_0 < x_1 < x_2 < \dots < x_n = b\}$$

con cualquier sucesión, no decreciente y no acotada, $\mathbf{t} = (t_j)_{j \in \mathbb{Z}}$, que satisfaga

$$\begin{aligned}
t_j &< t_{j+k+1} \text{ para todo } j, \\
t_{-k} &\leq \dots \leq t_0 := x_0 < t_1 := x_1 < \dots < t_n := x_n,
\end{aligned}$$

sucesión de la cual solo importará la subsucesión finita

$$t_{-k}, t_{-k+1}, \dots, t_{n+k}.$$

Además de lo anterior, necesitaremos hacer uso del siguiente teorema:

Teorema 10. *Para toda sucesión finita, no decreciente*

$$t_0 \leq t_1 \leq \cdots \leq t_n$$

de números reales, la matriz de orden $n + 1$

$$V_n(t_0, t_1, \dots, t_n) := \begin{bmatrix} \Pi^{(s_0-1)}(t_0) \\ \Pi^{(s_1-1)}(t_1) \\ \vdots \\ \Pi^{(s_n-1)}(t_n) \end{bmatrix}$$

es no singular. Donde $\Pi(t)$ es el vector fila dado por

$$\Pi(t) := \left[1, t, \dots, \frac{t^n}{n!} \right],$$

y $\Pi^{(k)}(t_j)$ denota la k -ésima derivada de la función evaluada en el punto t_j .

Demostración. Véase [23]. □

Teorema 11. *Las restricciones $B_{i,k+1,t}|[a,b]$ de las $n + k$ B-splines $B_i = B_{i,k+1,t}, i = -k, -k + 1, \dots, n - 1$, sobre $[a, b]$ forman una base para el espacio $S_{\Delta,k}$.*

Demostración. Cada una de las $B_i, -k \leq i \leq n - 1$, coincide con un polinomio de grado menor o igual a k sobre cada uno de los subintervalos $[x_j, x_{j+1}]$ de $[a, b]$ y tienen derivada hasta el orden $k - 1$ en los nodos interiores $t_j = x_j, j = 1, 2, \dots, n - 1$, de Δ . Lo anterior muestra que $B_i|[a, b] \in S_{\Delta,k}$ para todo $i = -k, \dots, n - 1$. Dado que previamente mostramos que la dimensión del espacio $S_{\Delta,k}$ es $n + k$, solo resta mostrar que las funciones $B_i, -k, \dots, n - 1$, son linealmente independientes sobre $[a, b]$.

Supongamos que $\sum_{i=-k}^{n-1} a_i B_i(x) = 0$ para todo $x \in [a, b]$. Consideremos el primer subintervalo $[x_0, x_1]$ de la partición Δ . Entonces se tiene que

$$\sum_{i=-k}^0 a_i B_i(x) = 0 \quad \text{para } x \in [x_0, x_1],$$

pues $B_i|[x_0, x_1] = 0$ para $i \geq 1$ por la propiedad **i.** del teorema anterior. Probaremos ahora que las funciones $B_{-k}, \dots, B_0$ son linealmente independientes sobre $[x_0, x_1]$. Observe que por (2.31) y (2.33) cada una de las funciones $B_i(x), -k \leq i \leq 0$ puede escribirse de la forma

$$B_i(x) = \alpha_1(t_1 - x)^{k+1-s_1} + \cdots + \alpha_{i+k+1}(t_{i+k+1} - x)^{k+1-s_{i+k+1}}, \quad (2.36)$$

donde $\alpha_{i+k+1} \neq 0$. Por lo tanto podemos afirmar que

$$\text{Gen} \{B_{-k}(x), \dots, B_0(x)\} = \text{Gen} \{(t_1 - x)^{k+1-s_1}, \dots, (t_{k+1} - x)^{k+1-s_{k+1}}\},$$

así, basta con establecer la independencia lineal de las funciones

$$(t_1 - x)^{k-(s_1-1)}, \dots, (t_{k+1} - x)^{k-(s_{k+1}-1)}$$

sobre $[x_0, x_1]$, o de forma equivalente, de las funciones normalizadas

$$p^{(s_j-1)}(t_j - x), \quad j = 1, 2, \dots, k+1,$$

definidas por

$$p(t) := \frac{t^k}{k!},$$

es decir, hay que mostrar la independencia lineal entre las derivadas de $p(t)$ evaluadas en los puntos $t_j - x$. Para mostrar dicha independencia lineal, supongamos que para ciertas constantes $c_1, c_2, \dots, c_{k+1}$ se cumple que

$$\sum_{j=1}^{k+1} c_j p^{(s_j-1)}(t_j - x) = 0 \quad \text{para todo } x \in [x_0, x_1],$$

entonces, por la expansión de Taylor alrededor del punto $x = t_j$, el polinomio

$$\sum_{j=1}^{k+1} c_j \sum_{l=0}^k p^{(l+s_j-1)}(t_j) \frac{(-x)^l}{l!} = 0$$

se anula sobre $[x_0, x_1]$ y por lo tanto debe corresponder al polinomio nulo, así que para todo $l = 0, 1, \dots, k$

$$\sum_{j=1}^{k+1} c_j p^{(l+s_j-1)}(t_j) = 0,$$

pero esto significa que

$$\sum_{j=1}^{k+1} c_j \Pi^{(s_j-1)}(t_j) = 0,$$

donde $\Pi(t)$ es el vector fila dado por

$$\Pi(t) := [p^{(k)}(t), p^{(k-1)}(t), \dots, p(t)] = \left[1, t, \dots, \frac{t^k}{k!} \right],$$

pero por el teorema (10) los vectores $\Pi^{(s_j-1)}(t_j), j = 1, 2, \dots, k+1$, son linealmente independientes, por tanto $c_1 = c_2 = \dots = c_{k+1} = 0$ lo que a su vez implica que $a_i = 0$ para $i = -k, k+1, \dots, 0$.

Repetiendo este mismo razonamiento sobre cada subintervalo $[x_j, x_{j+1}], j = 1, 2, \dots, n-1$, de la partición Δ de $[a, b]$ obtenemos que $a_i = 0$ para todo $i = -k, -k+1, \dots, n-1$, mostrando así que todas las $B_i, -k \leq i \leq n-1$, son linealmente independientes sobre $[a, b]$ y sus restricciones sobre el mismo forman una base para el espacio $S_{\Delta, k}$. $\square$

Como vimos anteriormente, las funciones B-spline renormalizadas $N_{i,r}(x)$ pueden ser calculadas utilizando la recursión dada por (2.35), pero esto a su vez hace posible representar $B_{i,r}(x)$ directamente como combinación lineal de $B_{i,r-1}$ y $B_{i+1,r-1}$, en efecto, por (2.34) y (2.35) se tiene que:

$$N_{i,r}(x) = \frac{B_{i,r}(x)}{t_{i+r} - t_i} = \frac{x - t_i}{t_{i+r} - t_i} \times \frac{B_{i,r-1}(x)}{t_{i+r-1} - t_i} + \frac{t_{i+r} - x}{t_{i+r} - t_i} \times \frac{B_{i+1,r-1}(x)}{t_{i+1+r-1} - t_{i+1}},$$

luego

$$B_{i,r}(x) = \frac{x - t_i}{t_{i+r-1} - t_i} B_{i,r-1}(x) + \frac{t_{i+r} - x}{t_{i+r} - t_{i+1}} B_{i+1,r-1}(x). \quad (2.37)$$

Así pues, la ecuación (2.37) es una recursión directa para calcular los valores de todas las B-spline para un x fijo.

Las propiedades de las funciones B-spline establecidas en el teorema (9) y la recursión dada por (2.37) son las que hacen de dichas funciones las candidatas perfectas para la construcción de las curvas y superficies que son utilizadas ampliamente por los sistemas de diseño geométrico asistido por computadora (CAGD) para generar todo tipo de formas que posteriormente se convertirán en productos. La anterior es una de las principales razones del por que las geometrías basadas en estas funciones se han convertido en el estándar de la representación e intercambio de datos geométricos de la gran mayoría de programas especializados. La fácil implementación de los objetos que describiremos a continuación dentro de los software más importantes de diseño, así como la fácil manipulación y edición de los modelos construidos con los mismos, encuentra su explicación principalmente en la teoría que hemos discutido en esta sección.

3 Modelado geométrico

Hasta ahora hemos discutido la idea de interpolar una muestra arbitraria de puntos en el plano bidimensional, al hacer esto encontramos que las funciones spline cumplen a cabalidad con esta tarea gracias a sus propiedades de convergencia, entre otras. En las funciones que forman la base del espacio compuesto por splines, B-splines, encontramos importantes propiedades y una forma recursiva de calcularse que hacen de las mismas candidatas ideales para definir curvas paramétricas óptimas para el modelado geométrico que se lleva a cabo con los sistemas CAGD. En contraste a la forma cartesiana de una curva, la forma paramétrica de la misma es una manera más natural para diseñar y representar figuras en un computador, también los coeficientes de las funciones que definen estas curvas a menudo poseen significado geométrico relevante, lo anterior se traduce entonces en métodos de diseño intuitivo y algoritmos numéricamente estables de carácter esencialmente geométrico [17].

Las funciones B-spline están definidas por polinomios, y como tal hacen que las curvas construidas con las mismas puedan representar con precisión una amplia gama de formas, sean procesadas por un computador de forma precisa, eficiente y fácil. Sin embargo, hay un número importante de curvas (y superficies) que no puedan ser representadas con precisión utilizando polinomios, estas solo se pueden aproximar en el mejor de los casos. En particular los círculos, elipses, hipérbolas, cilindros, conos, esferas, etc., hacen parte de los objetos que son difíciles, y a veces imposibles, de representar solamente utilizando polinomios.

Por ejemplo, supongamos que el círculo de radio unitario pudiera ser representado por una curva paramétrica compuesta solamente de polinomios en sus dos componentes, esto es:

$$\begin{aligned}x(t) &= a_0 + a_1t + \cdots + a_nt^n \\ y(t) &= b_0 + b_1t + \cdots + b_nt^n.\end{aligned}$$

Entonces, $x^2 + y^2 - 1 = 0$ implica que

$$\begin{aligned}0 &= (a_0 + a_1t + \cdots + a_nt^n)^2 + (b_0 + b_1t + \cdots + b_nt^n)^2 - 1 \\ &= (a_0^2 + b_0^2 - 1) + 2(a_0a_1 + b_0b_1)t + (a_1^2 + 2a_0a_2 + b_1^2 + 2b_0b_2)t^2 \\ &\quad + \cdots + (a_{n-1}^2 + 2a_{n-2}a_n + b_{n-1}^2 + 2b_{n-2}b_n)t^{2n-2} \\ &\quad + 2(a_na_{n-1} + b_nb_{n-1})t^{2n-1} + (a_n^2 + b_n^2)t^{2n}.\end{aligned}$$

Pero esta ecuación debe cumplirse para todo t , es decir que todos los coeficientes deben ser iguales a cero. Comenzando desde la potencia más grande y yendo hacia atrás, se muestra

que $a_i = 0$ y $b_i = 0$ para todo $1 \leq i \leq n$. Lo cual conlleva a que $x(t) = a_0$ y $y(t) = b_0$, lo que es una contradicción. Afortunadamente, es sabido que todas las secciones cónicas, incluyendo el círculo, pueden ser representadas usando funciones racionales [13], las cuales están definidas como el cociente entre dos polinomios, esto es,

$$x(t) = \frac{X(t)}{W(t)}, \quad y(t) = \frac{Y(t)}{W(t)}, \quad (3.1)$$

donde $X(t)$, $Y(t)$ y $W(t)$ son polinomios, es decir, cada una de las componentes de la curva paramétrica tiene el mismo denominador. De esta forma, el círculo unitario puede representarse mediante la curva paramétrica dada por

$$x(t) = \frac{1 - t^2}{1 + t^2} \quad y(t) = \frac{2t}{1 + t^2},$$

pues claramente en este caso se tiene que

$$\begin{aligned} (x(t))^2 + (y(t))^2 &= \left(\frac{1 - t^2}{1 + t^2} \right)^2 + \left(\frac{2t}{1 + t^2} \right)^2 \\ &= \frac{1 - 2t^2 + t^4 + 4t^2}{(1 + t^2)^2} \\ &= \frac{(1 + t^2)^2}{(1 + t^2)^2} \\ &= 1. \end{aligned}$$

Curvas racionales con componentes de la forma (3.1) cuentan con una interpretación geométrica que conlleva a un procesamiento eficiente de las mismas. La idea es utilizar *coordenadas homogéneas* para representar curvas racionales en un espacio n -dimensional como una curva polinómica en el espacio $(n + 1)$ -dimensional. Considere por ejemplo un punto arbitrario de $\mathbb{R}^3$ de coordenadas arbitrarias $\mathbf{P} = (x, y, z)$, entonces $\mathbf{P}$ es de la forma $\mathbf{P}^w = (x, y, z, w) = (X, Y, Z, W)$ como punto de $\mathbb{R}^4$, con $w \neq 0$. Así, $\mathbf{P}$ es obtenido a partir de $\mathbf{P}^w$ al dividir cada componente entre la cuarta componente W , esto es, mapeando $\mathbf{P}^w$ desde el origen sobre el hiperplano $W = 1$. Este mapeo, denotado como H , es una aplicación de perspectiva con centro en el origen, $\mathbf{P} = H(\mathbf{P}^w) = H(X, Y, Z, W) = (X/W, Y/W, Z/W)$ [17]. De esta forma, mapear bajo H una curva en $\mathbb{R}^4$ resultará en curva en $\mathbb{R}^3$, análogamente, mapear bajo H una curva en $\mathbb{R}^3$ tendrá como resultado una curva de $\mathbb{R}^2$, la figura (3.1) ilustra este último caso. Esta idea será aplicada en líneas posteriores cuando definamos las curvas B-splines racionales.

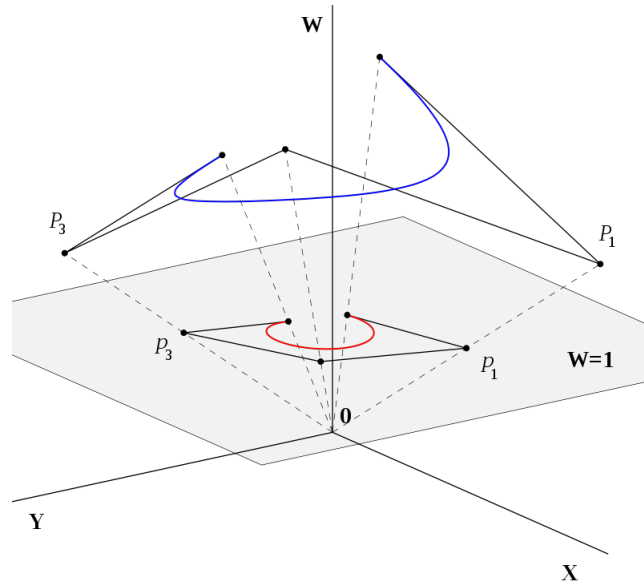


Figura 3.1: La proyección de una curva de $\mathbb{R}^3$ (azul) sobre el plano $W = 1$.

Los sistemas CAGD tienen como filosofía común la construcción de modelos a partir de *puntos de control*, los cuales controlan la forma local o general de las curvas y polígonos que componen el modelo. Estos puntos, bidimensionales o tridimensionales, no necesariamente hacen parte del mismo, sin embargo sirven para definir toda su estructura. Contrario a lo que discutimos en el capítulo anterior, las curvas y superficies que definiremos a lo largo de este capítulo no contemplan una interpolación completa de los puntos de control que las definen, más bien se conciben como aproximaciones a los mismos; esta manera de modelar conlleva a una forma más intuitiva y práctica para la implementación, edición, y manipulación de las construcciones realizadas por dichos sistemas.

Las curvas que consisten de solo un polinomio o un segmento racional a menudo son inadecuadas para el modelamiento geométrico, algunas de sus desventajas son:

- Requieren un alto grado para satisfacer un gran número de condiciones, por ejemplo, se requiere una curva polinomial de grado $(n - 1)$ para trazar una curva polinomial de Bézier a través de n puntos. Sin embargo, estas curvas son ineficientes de procesar y son numéricamente inestables.
- Se requieren curvas polinomiales de un alto grado para ajustar con precisión algunas formas complejas.

La solución es usar curvas (superficies) que sean *polinomiales a trozos*, o *racionales a trozos*. La figura (3.2) muestra una curva, $C(t)$, que consiste de tres *segmentos* polinomiales cúbicos. $C(t)$ está definida sobre $t \in [0, 1]$. Los valores de los parámetro $t_0 = 0 < t_1 < t_2 < t_3 = 1$ se

denominan *nodos*. Denotamos los segmentos por $C_i(t)$, $1 \leq i \leq 3$. Los segmentos están contruidos de tal forma que estén unidos con algún grado de continuidad (no necesariamente el mismo en cada nodo). Sea $C_i^{(j)}$ la j -ésima derivada de C_i . Decimos que $C(t)$ es C^k continua en el nodo t_i si $C_i^{(j)}(t_i) = C_{i+1}^{(j)}(t_{i+1})$ para todo $0 \leq j \leq k$ [17].

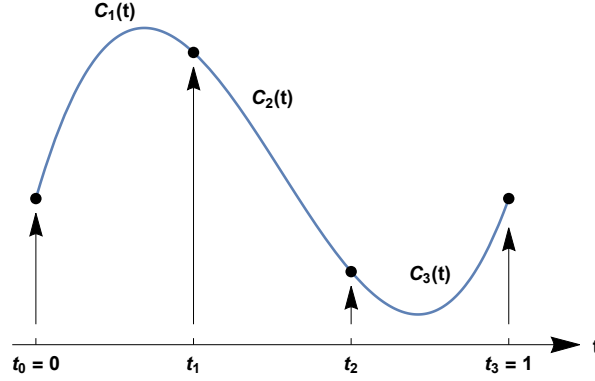


Figura 3.2: $C(t)$ representada por funciones polinomiales a trozos

Así, estos sistemas requieren entonces una representación de curvas con la forma

$$C(t) = \sum_{i=0}^n f_i(t) \mathbf{P}_i, \quad (3.2)$$

donde $\mathbf{P}_i$ denotan los puntos de control de la curva, mientras que las funciones $f_i(t)$, $i = 0, 1, \dots, n$, son funciones polinomiales a trozos que forman una base para el espacio de todas las funciones polinomiales a trozos del grado y continuidad deseada (para una sucesión fija de nodos $T = \{t_i, 0 \leq i \leq m\}$). Observe que la continuidad está determinada por las funciones base f_i y por tanto los puntos de control $\mathbf{P}_i$ pueden modificarse sin alterar la continuidad de la curva que controlan, más aún, las propiedades analíticas usuales de estas funciones deben hacer que las curvas definidas por (3.2) tengan importantes propiedades geométricas como por ejemplo:

- Control convexo: Las curvas están contenidas en la parte convexa del polígono que definen sus puntos de control.
- Variación decreciente: Ninguna línea recta intersecta la curva más veces de las que intersecta el polígono de control que la contiene.
- Invariancia bajo transformaciones: Rotaciones, traslaciones y escalamientos se aplican a la curva a través de sus puntos de control.

Además de las anteriores, una de las propiedades de mayor importancia que deben poseer estas funciones es la de soporte compacto, esto es, cada $f_i(t)$ es no nula solo en un número

limitado de subintervalos, más no en todo el dominio, $[t_0, t_m]$. Dado que $\mathbf{P}_i$ es multiplicado por $f_i(t)$, alterar la posición de $\mathbf{P}_i$ afecta la forma de la curva solo en los subintervalos en los cuales $f_i(t)$ no se anula [17].

Teniendo en cuenta todo lo anterior, en especial lo concerniente al soporte compacto de funciones y el uso de funciones racionales, es claro que al utilizar las funciones B-spline como bases en (3.2) se obtienen objetos matemáticos que permiten llevar a cabo un proceso de modelado geométrico de gran precisión, que hacen posible a su vez la consecución de una gama amplia de formas complejas bien elaboradas.

3.1. Curvas B-Spline

Se define la curva B-spline de orden r como

$$C(x) = \sum_{i=0}^n B_{i,r}(x) \mathbf{P}_i, \quad t_{min} \leq x \leq t_{max}, \quad (3.3)$$

donde los $\mathbf{P}_i$ son los puntos de control de la curva, que forman el polígono de control de la misma, y las $B_{i,r}$ son las funciones base B-spline definidas por (2.32), y la recursión (2.37), asociadas a la sucesión $\mathbf{t} = \{t_i\}$. En adelante nos referiremos a la sucesión $\mathbf{t}$ como *vector nodal* y lo denotaremos como T , es decir, las funciones $B_{i,r}$ estarán asociadas al vector nodal $T = \{t_{min}, \dots, t_{max}\}$, cuyos nodos satisfacen $t_i \leq t_{i+1}$.

En la figura (3.3) se muestran las funciones base B-spline definidas sobre un vector nodal T y una curva, junto a su polígono de control, construida con dichas bases.

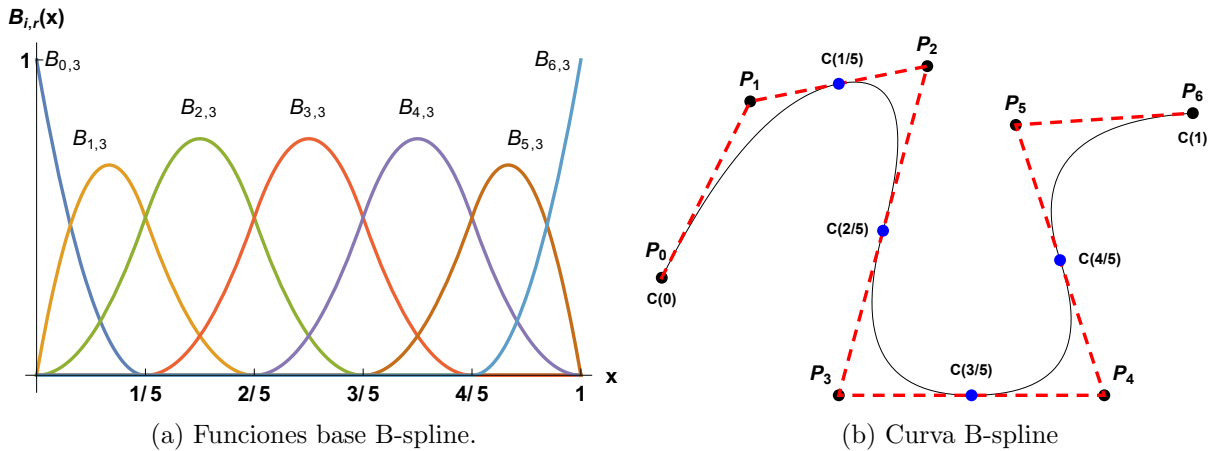


Figura 3.3: (a) Funciones base definidas sobre $T = \{0, 0, 0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1, 1, 1\}$. (b) Una curva cuadrática (orden 3) a trozos utilizando las funciones base B-spline de (a) y su polígono de control.

Recordemos que por la definición (2.32) y de la recursión dada por (2.37) tenemos entonces

que la i -ésima B-spline de orden $r \geq 1$ está dada por la fórmula recursiva:

$$B_{i,1}(x) = \begin{cases} 1; & t_i \leq x < t_{i+1} \\ 0; & \text{en otro caso.} \end{cases}$$

$$B_{i,r}(x) = \frac{x - t_i}{t_{i+r-1} - t_i} B_{i,r-1}(x) + \frac{t_{i+r} - x}{t_{i+r} - t_{i+1}} B_{i+1,r-1}(x).$$

Las propiedades de las funciones B-spline establecidas por el teorema (9) serán entonces heredadas de forma natural por la curvas definidas por (3.3), por ejemplo, el punto de control $\mathbf{P}_i$ solo afectará a la curva $C(x)$ localmente, justo sobre el soporte compacto que la función $B_{i,r}(x)$ tenga en el vector nodal T . Es precisamente el soporte de las bases las que permiten ver que el orden de la curva y el número de puntos de control son los que determinan el número de nodos de T sobre el cual se definen las B-spline que permitirán la construcción de la misma, esto es, el último punto de control $\mathbf{P}_n$ tendrá asociada la última base que contribuye a la construcción de la curva, $B_{n,r}$, cuyo soporte está sobre el intervalo $[t_n, t_{n+r}]$, y dado que el primer punto de control, $\mathbf{P}_0$ tiene asociada la primera base, $B_{0,r}$, cuyo soporte es $[t_0, t_r]$, por lo tanto se obtiene que el vector nodal es de la forma

$$T = \{t_0, t_1, \dots, t_{r-1}, t_r, t_{r+1}, \dots, t_{n-1}, t_n, t_{n+1}, \dots, t_{n+r}\},$$

es decir, el número m de nodos de T está determinado por el número de puntos de control y el orden de la curva, $m = n + 1 + r$. En efecto, observe que la curva presentada en la figura (3.3) tiene siete puntos de control y es de orden tres, lo que hace que su vector nodal este formado por diez nodos, $m = 6 + 1 + 3 = 10$. Cada lapso nodal, $t_i \leq x \leq t_{i+1}$, es mapeado sobre una curva polinomial comprendida entre dos puntos consecutivos de la curva, $C(t_i)$ y $C(t_{i+1})$.

Una curva B-spline en general no contiene sus puntos de control extremos, $\mathbf{P}_0$ y $\mathbf{P}_n$. Incrementar la multiplicidad de un nodo reduce la continuidad de la curva en la imagen de dicho nodo, más específicamente, la curva $C(x)$ es $r - p - 1$ veces continuamente diferenciable en un nodo de multiplicidad p ($\leq r$), es decir, tiene continuidad $C^{(r-p-1)}$. Por lo tanto, el polígono de control coincidirá con la curva en la imagen de un nodo de multiplicidad $p = r - 1$, así, un nodo de multiplicidad $p = r$ indica continuidad C^{-1} , lo cual significa una curva discontinua [19]. Entonces, repetir los nodos extremos del vector nodal r veces hará que la curva coincida con los puntos extremos, así, el primer y el último punto de control de una curva $C(x)$ construida con un vector nodal T de la forma

$$T = \underbrace{\{t_0, t_1, \dots, t_{r-1}\}}_{r \text{ nodos iguales}}, \underbrace{\{t_r, t_{r+1}, \dots, t_{n-1}, t_n\}}_{n-r+1 \text{ nodos internos}}, \underbrace{\{t_{n+1}, \dots, t_{n+r}\}}_{r \text{ nodos iguales}},$$

coincidirán con los puntos finales de la misma, esto es, $\mathbf{P}_0 = C(t_0) = C(t_{r-1})$ y $\mathbf{P}_n = C(t_{n+r}) = C(t_{n+1})$ [16]. En los sistemas CAGD se utilizan en general dos tipos de vectores nodales, *periódicos* y *no periódicos*, los cuales pueden ser a su vez *uniformes* o *no uniformes*.

Un vector nodal no periódico, uniforme, tiene multiplicidad r en sus extremos y sus nodos internos se encuentran equidistantes, es decir, existe un real d tal que $d = t_{i+1} - t_i$ para $i = r, \dots, n$. Los vectores periódicos no tienen nodos múltiples en sus extremos y pueden ser también uniformes o no uniformes, por ejemplo, un nodo interior múltiple automáticamente hace que el vector sea no uniforme. Si tal d no existe se dice que el vector es no uniforme [19]. Así, el vector nodal asociado a la curva de la figura (3.3) es uniforme y no periódico. La presencia de nodos múltiples en T también implica que al calcular las funciones $B_{i,r}$ utilizando la recursión (2.37) se presenten formas indeterminadas $0/0$, este cociente se define como 0.

Lo anterior muestra entonces que una de las formas de manipular la curva $C(x)$ es a través de los $n - r + 1$ nodos internos del vector nodal sobre el que estén definidas sus bases, incrementar la multiplicidad de los mismos resultará en una curva con cúspides. Este hecho también implicará que T pase a ser no uniforme, en caso de que inicialmente no lo fuera.

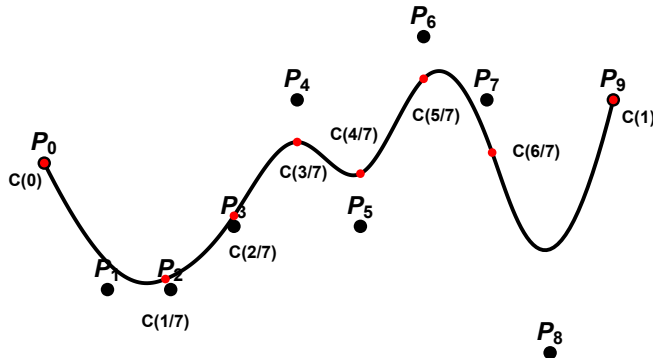


Figura 3.4: Curva B-spline de orden $r = 4$ (cúbica).

La figura (3.4) muestra una curva B-spline de orden cuatro controlada por diez puntos, $P_i, 0 \leq i \leq 9$, cuyo vector nodal está dado por $T = \{0, 0, 0, 0, \frac{1}{7}, \frac{2}{7}, \frac{3}{7}, \frac{4}{7}, \frac{5}{7}, \frac{6}{7}, 1, 1, 1, 1\}$, el cual es un vector nodal uniforme y no periódico. Dado que sus nodos internos no tienen multiplicidad, la curva es tres veces continuamente diferenciable sobre la imagen de cada uno de ellos. La figura (3.5) muestra la curva cuando el nodo $t_6 = 3/7$ presenta multiplicidad $p = 3$, podemos observar claramente como la diferenciabilidad en la curva disminuye en el punto $C(3/7)$, este punto de la curva pasa a coincidir con uno de los puntos de control. Por otro lado, la figura (3.6) muestra el efecto que causa la misma multiplicidad sobre las bases que definen la curva.

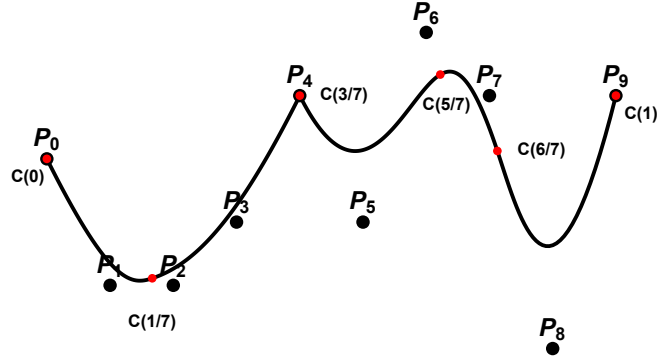


Figura 3.5: Curva B-spline de orden $r = 4$ con multiplicidad en uno de los nodos internos.

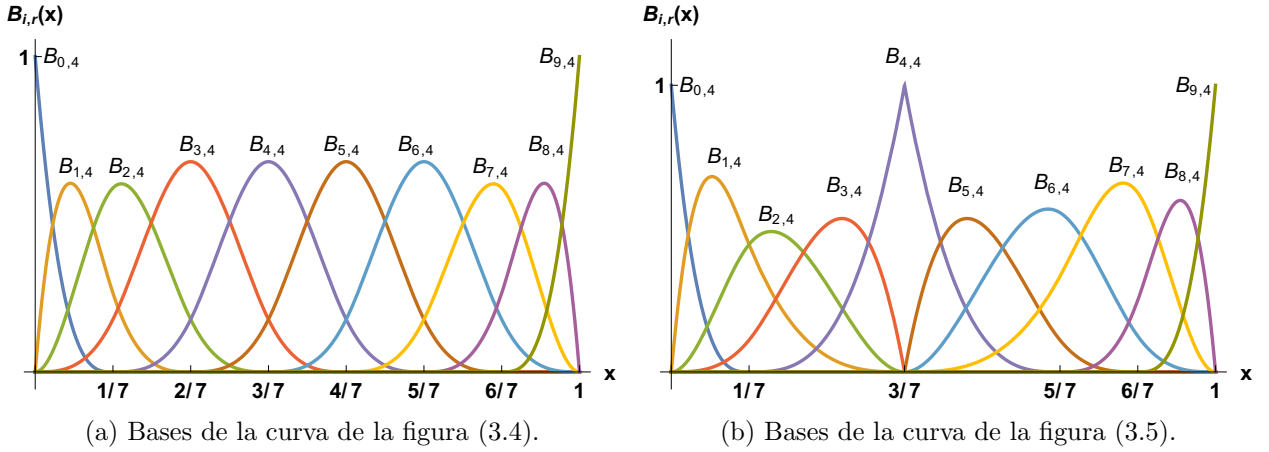


Figura 3.6: Efecto de la multiplicidad de un nodo interno sobre las bases de la curva.

La curva $C(x)$ está definida por varios trozos de curvas polinomiales, dichos trozos están controlados solo por r puntos de control, y cualquier punto de control afecta solamente r segmentos de la curva. En efecto, cambiar $\mathbf{P}_i$ afecta la curva en el rango del parámetro $t_i < x < t_{i+r}$, además, un punto x sobre la curva tal que $t_j < x < t_{j+1}$, está determinado completamente por los puntos de control $\mathbf{P}_{j-(r-1)}, \dots, \mathbf{P}_j$. Manipular los puntos de control es entonces otra de las formas efectivas de modificar la forma de la curva, esta propiedad en particular es utilizada ampliamente por los sistemas CAGD, pues la misma permite que quien modele tenga un control más intuitivo de sus creaciones. La figura (3.7) ilustra este hecho mostrando como cambia la curva (punteada) representada en (3.4) al mover el punto $\mathbf{P}_5$ dos unidades hacia abajo.

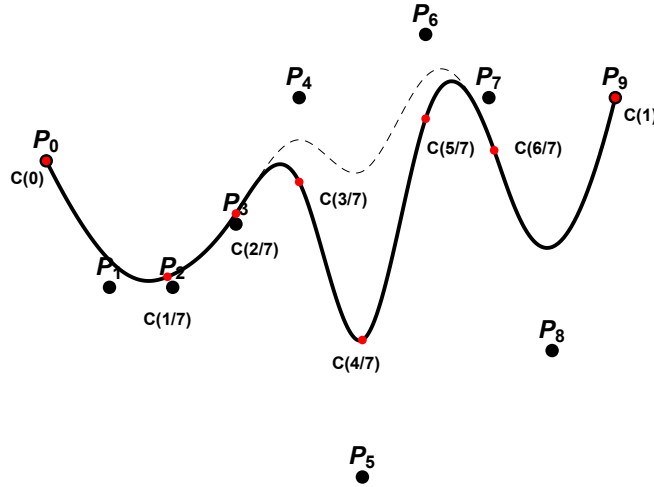


Figura 3.7: Efecto de la manipulación de los puntos de control sobre una curva B-spline.

Las formas de manipular las curvas discutidas hasta aquí son solo un par de ejemplos de como los sistemas de modelado geométrico llevan a cabo la representación de las muchas distintas formas que un usuario desea construir, además de estas por ejemplo está la repetición de los puntos de control P_n y P_0 al inicio y al final del polígono de control, esta acción permitirá construir curvas cerradas. Una discusión más amplia y detallada sobre esta forma de alteración, además de la manipulación de vectores nodales periódicos puede consultarse en [16, 17, 19].

Como lo discutimos al inicio de este capítulo, para poder representar una gama de formas más amplia aún de las que son posibles con las curvas B-spline es necesario generalizarlas a su forma racional, así, las curvas definidas por (3.3) son un caso particular de las curvas que discutiremos a continuación, las mismas generalmente están definidas en la práctica sobre vectores nodales no uniformes; es precisamente de allí que se deriva su nombre, *Non Uniform Rational B-Splines*, por sus siglas en inglés.

3.2. NURBS

En líneas anteriores discutimos brevemente la idea de la representación de curvas racionales en el espacio n -dimensional como la proyección de curvas polinómicas en el espacio $(n + 1)$ -dimensional, utilizaremos ahora dicha idea para la construcción de las curvas B-spline racionales. Consideraremos el caso en el cual los puntos de control pertenecen el espacio tridimensional, la construcción es análoga para puntos bidimensionales.

Dado el conjunto de puntos de control P_i de la curva B-spline definida por (3.3), asociaremos a cada uno de ellos un *peso* (o influencia) w_i ; con ello podemos entonces definir los puntos de control ponderados $P_i^{w_i} = (w_i x_i, w_i y_i, w_i z_i, w_i)$, los cuales sirven para construir la curva

B-spline de $\mathbb{R}^4$ definida por

$$C^w(x) = \sum_{i=0}^n B_{i,r}(x) \mathbf{P}_i^w, \quad (3.4)$$

así, aplicando el mapeo H a la curva $C^w(x)$ obtenemos la correspondiente curva B-spline racional, es decir, mapeando las componentes de (3.4),

$$\begin{aligned} X(t) &= \sum_{i=0}^n B_{i,r}(t) w_i x_i, & Y(t) &= \sum_{i=0}^n B_{i,r}(t) w_i y_i \\ Z(t) &= \sum_{i=0}^n B_{i,r}(t) w_i z_i, & W(t) &= \sum_{i=0}^n B_{i,r}(t) w_i, \end{aligned}$$

sobre el hiperplano $W = 1$ obtenemos la curva en el espacio tridimensional cuyas componentes estarán dadas por

$$\begin{aligned} x(t) &= \frac{X(t)}{W(t)} = \frac{\sum_{i=0}^n B_{i,r}(t) w_i x_i}{\sum_{i=0}^n B_{i,r}(t) w_i}, \\ y(t) &= \frac{Y(t)}{W(t)} = \frac{\sum_{i=0}^n B_{i,r}(t) w_i y_i}{\sum_{i=0}^n B_{i,r}(t) w_i}, \\ z(t) &= \frac{Z(t)}{W(t)} = \frac{\sum_{i=0}^n B_{i,r}(t) w_i z_i}{\sum_{i=0}^n B_{i,r}(t) w_i}, \end{aligned}$$

esto es,

$$C(t) = (x(t), y(t), z(t)) = \frac{\sum_{i=0}^n B_{i,r}(t) w_i (x_i, y_i, z_i)}{\sum_{i=0}^n B_{i,r}(t) w_i} = \frac{\sum_{i=0}^n B_{i,r}(t) w_i \mathbf{P}_i}{\sum_{i=0}^n B_{i,r}(t) w_i}.$$

3.2.1. Curvas NURBS

Una curva NURBS de orden r está definida por

$$C(x) = \frac{\sum_{i=0}^n B_{i,r}(x) w_i \mathbf{P}_i}{\sum_{i=0}^n B_{i,r}(x) w_i}, \quad t_{min} \leq x \leq t_{max} \quad (3.5)$$

de nuevo los $\mathbf{P}_i$ denotan los puntos de control de la curva, las funciones base $B_{i,r}$ están definidas sobre el vector nodal, no periódico y no uniforme,

$$T = \{t_0, t_1, \dots, t_{r-1}, t_r, t_{r+1}, \dots, t_{n-1}, t_n, t_{n+1}, \dots, t_{n+r}\},$$

y los w_i , escalares no negativos, son los pesos o factores de influencia de los puntos de control. Observe que si $w_i = 1$ para todo $i = 0, \dots, n$, entonces la curva (3.5) se reduce a la curva no racional (3.3), así, las NURBS se definen como una generalización de las curvas B-spline. Observe que si definimos

$$R_{i,r}(x) = \frac{B_{i,r}(x)w_i}{\sum_{i=0}^n B_{i,r}(x)w_i},$$

la curva (3.5) puede reescribirse como

$$C(x) = \sum_{i=0}^n R_{i,r}(x)\mathbf{P}_i,$$

de esta forma, las funciones $R_{i,r}$ se entienden como las bases racionales de la curva NURBS. Observe que las propiedades de estas nuevas curvas, así como la forma de manipularla a través de los nodos t_i del vector T y los puntos de control $\mathbf{P}_i$ son prácticamente análogas a las discutidas anteriormente. Sin embargo la influencia, o peso, w_i , de los puntos de control $\mathbf{P}_i$ de la curva si implica una forma particular de alterar la misma. Aumentar o disminuir el valor de este escalar geoméricamente significa mover la posición del punto de control a lo largo de la recta $\alpha(s) = \mathbf{P}_i + s\mathbf{P}_i$, y no de forma arbitraria sobre el espacio como podría darse al simplemente manipular $\mathbf{P}_i$.

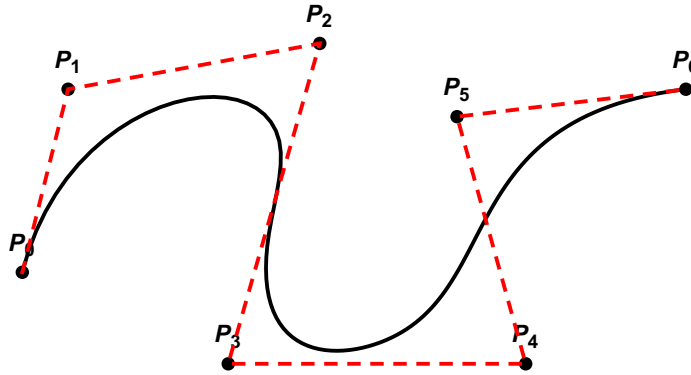


Figura 3.8: Curva NURBS cúbica controlada por 7 puntos.

La figura (3.8) muestra una curva NURBS de orden cuatro, en ella todos los pesos w_i de los puntos de control son iguales a uno. Al manipular el valor de w_3 el punto $\mathbf{P}_3$ tendrá mayor o menor incidencia sobre la curva, la misma se acercará o alejará cada vez más a dicho punto a medida que w_3 cambie, entre más grande sea su valor, más se aproximará y viceversa. La figura (3.9) ilustra este hecho, en ella las líneas punteadas representan las curvas que se obtienen al asignar a w_3 los valores de 0 , $\frac{1}{4}$, $\frac{1}{2}$ y 2 .

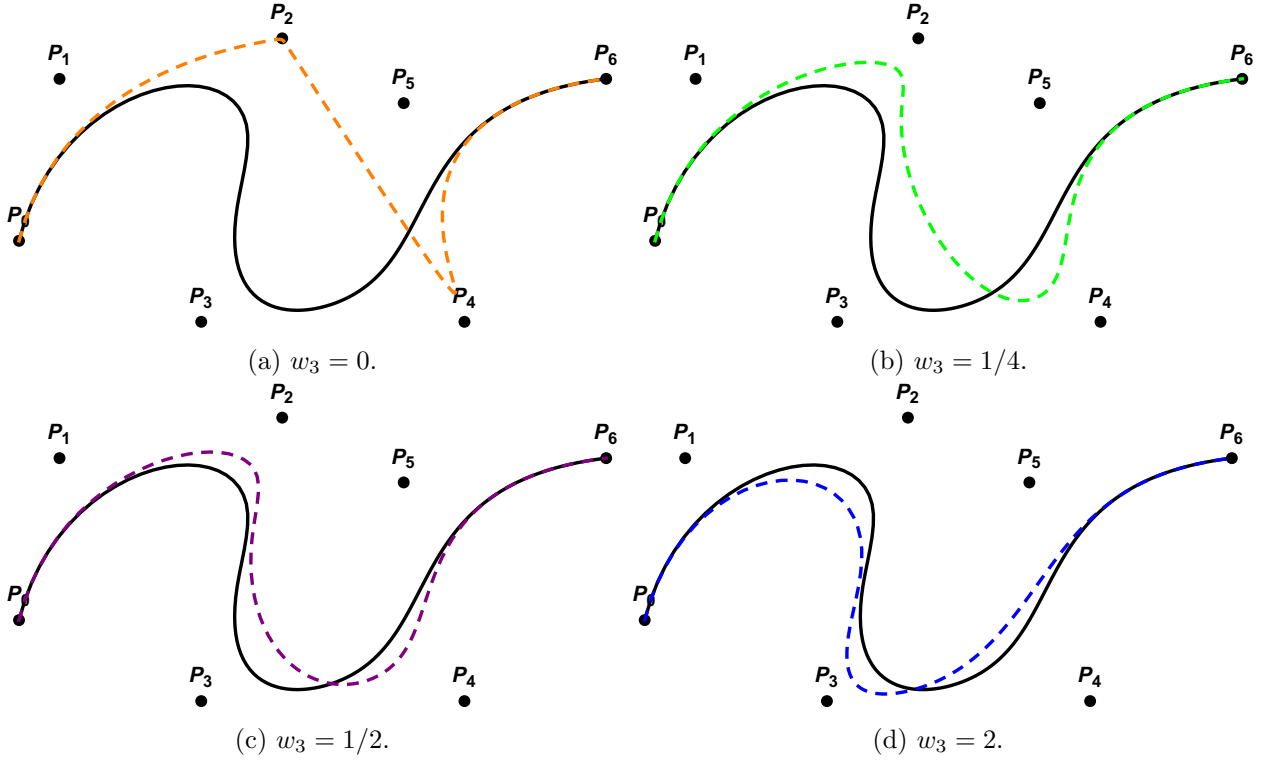


Figura 3.9: Efecto del cambio de w_3 sobre la curva NURBS representada en la figura (3.8).

Lo anterior entonces muestra la facilidad con la cual puede manipularse las curvas NURBS para generar todo tipo de formas, entre ellas las secciones cónicas. Esta, junto al hecho de que es posible generar a partir de las mismas otro tipo de curvas de amplio uso, como por ejemplo las curvas de Bezier, es una de las principales razones del por que es uno de los estándares de representación geométrica implementados en los sistemas de diseño asistido por computador y los distintos paquetes de software asociados a los mismos. A continuación mostramos dos ejemplos sencillos que ilustran la versatilidad de las formas que pueden construirse con estos objetos.

Ejemplo 1: Círculo unitario.

Para la construcción del círculo unitario se utilizaron puntos de control $\mathbf{P}_i$ de tal forma que el polígono de control coincidiera con un cuadrado de dos unidades de longitud, además, tomamos $\mathbf{P}_0 = \mathbf{P}_n$ para garantizar que la curva resultante fuera cerrada. La figura (3.10) muestra el resultado obtenido al definir las bases B-spline cuadráticas, $r = 3$, sobre el vector nodal

$$T = \left\{ 0, 0, 0, \frac{\pi}{2}, \frac{\pi}{2}, \pi, \pi, \frac{3\pi}{2}, \frac{3\pi}{2}, 2\pi, 2\pi, 2\pi \right\},$$

y asociar a los puntos de control los escalares de influencia

$$\begin{aligned} w_0 = 1, \quad w_1 = \sqrt{2}/2, \quad w_2 = 1, \quad w_3 = \sqrt{2}/2, \quad w_4 = 1, \\ w_5 = \sqrt{2}/2, \quad w_6 = 1, \quad w_7 = \sqrt{2}/2, \quad w_8 = 1. \end{aligned}$$

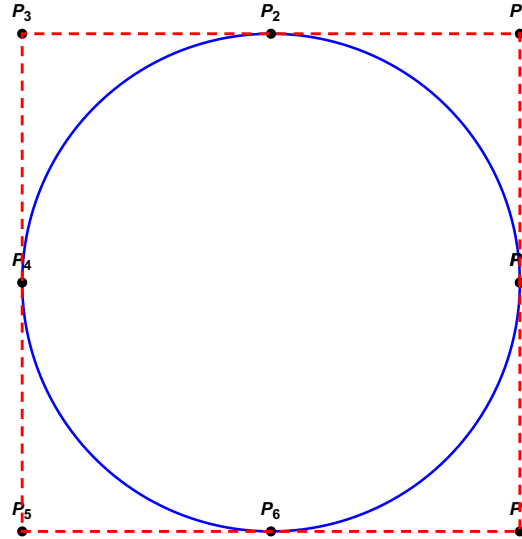


Figura 3.10: Círculo unitario NURBS controlado por 9 puntos.

Observe que al ser construida de esta forma, la curva circular está constituida por cuatro curvas que corresponden a cuartos de círculo unidos en sus extremos por nodos dobles. Si bien la multiplicidad de nodos en el vector nodal resulta en pérdida de la continuidad en la primera derivada, los puntos de control están posicionados de tal forma que la primera derivada sea continua.

Así, queda demostrada la ventaja de utilizar curvas polinomiales racionales frente a curvas polinomiales no racionales; la posibilidad de poder representar con precisión las secciones cónicas es de vital importancia en muchos procesos de diseño.

Ejemplo 2: Silueta Anger (completa)

En el ejemplo ilustrativo de los métodos de interpolación del capítulo anterior, mostramos la aproximación lograda a través de splines de la silueta superior de **Anger**, figura (2.3), en dicho ejemplo no era posible aproximar su contorno completo ya que el mismo correspondía a una curva cerrada, y de intentar hacerlo los nodos de interpolación no formarían una sucesión estrictamente creciente como es requerido. A continuación mostramos que esta tarea si se puede llevar a cabo con curvas NURBS.

Se situaron alrededor del contorno del personaje un total de 86 puntos de control, inicialmente el peso de cada uno de ellos fue de $w_i = 1$, la figura (3.11) muestra la curva NURBS cúbica resultante al emplear este factor de influencia.



Figura 3.11: Silueta aproximada de Anger, $w_i = 1$.

Si bien la curva resultante es una buena aproximación a la silueta del personaje, a través de la manipulación de los escalares w_i pueden pulirse algunas de las secciones para tener una forma más fiel a la original. Observe que debajo de los brazos del personaje la densidad de los puntos de control (en color verde) es mayor al resto de la silueta, esto se debe a que dicha sección es la más compleja de aproximar, de haber situado sobre dicha región una cantidad menor de puntos hubiera sido más difícil representarla. Esta técnica de repoblar con mayor puntos de control las partes más intrincadas de un modelo, es ampliamente utilizada dentro de los sistemas de modelado. La figura (3.12) muestra la curva NURBS modificar apropiadamente la influencia de los puntos de control.

La combinación eficaz de todas las formas de modificar las curvas NURBS hacen de esta una herramienta poderosa para la representación de todo tipo de formas, gran parte del éxito de los software más importantes de diseño se debe a que los mismos han podido implementar de forma efectiva estos objetos, lo que ha su vez ha hecho posible un entorno de trabajo más eficiente e intuitivo para sus usuarios. Si bien aquí nos hemos enfocado principalmente en discutir, aunque brevemente, las formas más importantes de manipular las curvas definidas con base en funciones B-spline, hay otras más que se pueden aplicar para construir una variedad importante de figuras, en la literatura especializada del campo de la computación gráfica se pueden encontrar discusiones más profundas al respecto de este tema, entre ellas se destaca la inserción y remoción de nodos del vector nodal asociado a las bases, así como también el incremento en el orden de las mismas. Lo anterior, y más, puede consultarse en [17, 20, 22].



Figura 3.12: Silueta aproximada de Anger, w_i modificados.

Habiendo definido ya estas curvas, y presentado sus propiedades más inmediatas, el paso siguiente será construir las superficies asociadas a las mismas, superficies que heredarán naturalmente las propiedades de control que hemos discutido hasta aquí.

3.2.2. Superficies NURBS

Al ser definidas como una función vectorial de un solo parámetro, u , las curvas NURBS mapean (deforman) los valores que toma dicho parámetro de un subconjunto de $\mathbb{R}$ y lo envía en el espacio tridimensional, $\mathbb{R}^3$, de otro lado, una superficie es una función vectorial de dos parámetros, u y v , que toman los valores de dichos parámetros en una región $\Omega \subset \mathbb{R}^2$ y la envía al espacio tridimensional. La forma más sencilla y adecuada para producir superficies en los sistemas de modelado geométrico es el esquema de *producto tensorial*, el cual es básicamente un esquema de curvas bidireccionales que utiliza funciones base y coeficientes geométricos. Las funciones base son funciones bivariantes construidas a partir del producto de funciones base univariantes, mientras que los coeficientes geométricos están organizados en una malla bidireccional de dimensión $n \times m$. Así, una superficie construida con este esquema es de la forma

$$S(u, v) = (x(u, v), y(u, v), z(u, v)) = \sum_{i=0}^n \sum_{j=0}^m f_i(u)g_j(v)\mathbf{P}_{i,j},$$

luego, el dominio de esta función es un rectángulo $\Omega \subset \mathbb{R}^2$. Teniendo en cuenta este esquema, las superficies NURBS surgen de forma natural como el producto de curvas NURBS.

Una superficie NURBS de orden r en la dirección x y de orden s en la dirección y está definida por

$$S(x, y) = \frac{\sum_{i=0}^n \sum_{j=0}^m B_{i,r}(x) B_{j,s}(y) w_{i,j} \mathbf{P}_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m B_{i,r}(x) B_{j,s}(y) w_{i,j}}, \quad t_{\min} \leq x \leq t_{\max}, \quad u_{\min} \leq y \leq u_{\max}. \quad (3.6)$$

Los $\mathbf{P}_{i,j}$ forman una malla de control bidireccional, los $w_{i,j}$ son los pesos asociados a cada uno de ellos, y las $B_{i,r}(x)$ y $B_{j,s}(y)$ son las funciones base B-spline definidas sobre los vectores nodales

$$T = \{t_0, t_1, \dots, t_{r-1}, t_r, t_{r+1}, \dots, t_{n-1}, t_n, t_{n+1}, \dots, t_{n+r}\},$$

$$U = \{u_0, u_1, \dots, u_{s-1}, u_s, u_{s+1}, \dots, u_{m-1}, u_m, u_{m+1}, \dots, u_{m+r}\},$$

respectivamente. La figura (3.13) muestra una superficie NURBS de orden tres en cada dirección.

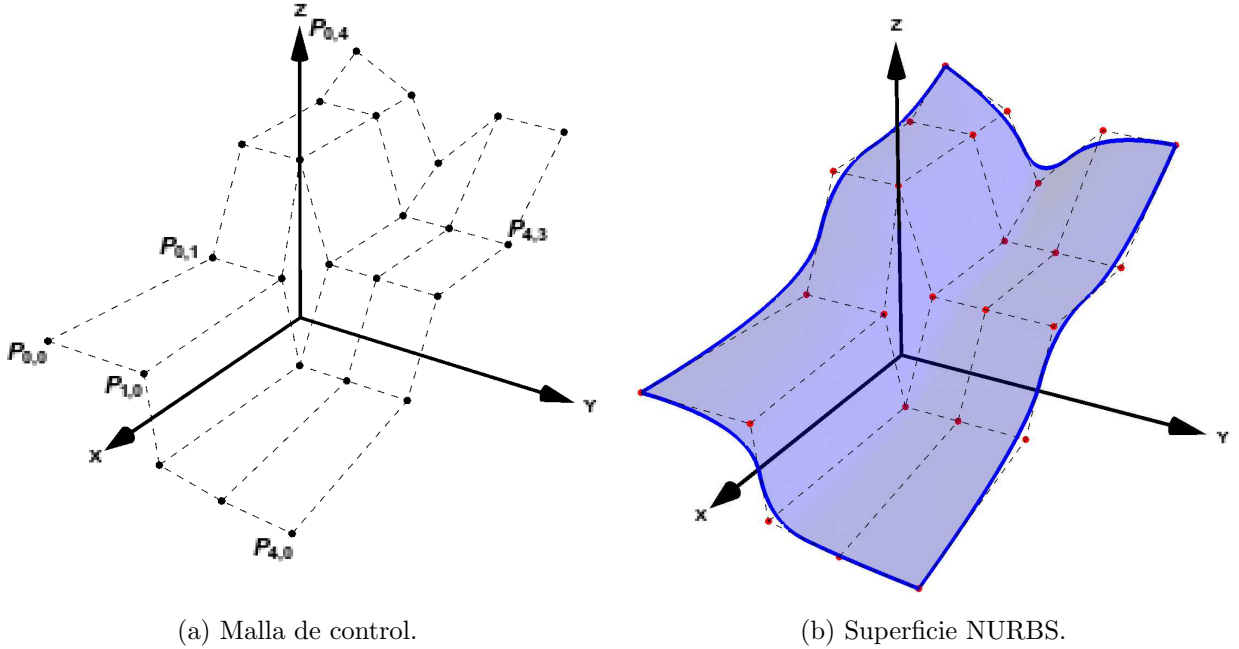


Figura 3.13: Las figuras (a) y (b) muestran la malla generada por los puntos de control $\mathbf{P}_{i,j}$ y la superficie NURBS generada a partir de los mismos utilizando funciones base B-spline de orden $r = 3$ sobre los vectores nodales dados por $T = \{0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1\} = U$.

Observe que definiendo las funciones base racionales

$$R_{i,j}(x, y) = \frac{B_{i,r}(x) B_{j,s}(y) w_{i,j}}{\sum_{k=0}^n \sum_{l=0}^m B_{k,r}(x) B_{l,s}(y) w_{k,l}}$$

la superficie de la ecuación (3.6) puede reescribirse como

$$S(x, y) = \sum_{i=0}^n \sum_{j=0}^m R_{i,j}(x, y) \mathbf{P}_{i,j},$$

así, las funciones $R_{i,j}(x, y)$ constituyen las funciones base bidireccionales de la superficie. Estas funciones heredarán entonces de forma natural todas las propiedades de las bases racionales que la definen.

En la práctica, los sistemas CAGD definen de forma automática el orden de las funciones base, $B_{i,r}(x)$ y $B_{j,s}(y)$, usualmente orden 4 en cada dirección, y los vectores nodales T y U sobre el que están soportadas las mismas, esto una vez se han especificado por parte del usuario los puntos que formarán la malla de control de la superficie. En particular, los vectores nodales son generados de tal forma que todos los valores de los parámetros x y y estén confinados al rectángulo $[0, 1] \times [0, 1]$; este hecho se da en parte debido a que al estar en este dominio se mejora la precisión numérica en el cálculo aritmético de valores en punto flotante, esto gracias a la alta densidad que presentan dichos valores en este conjunto [16]. Una forma popular de modelado geométrico lleva por nombre *NURBS modelling*, este consiste en modelar las piezas de interés a partir de superficies NURBS primitivas que se encuentran implementadas de forma genérica en la interfaz de la mayoría de programas de diseño tridimensional. Por ejemplo, si se pretende modelar la cabeza de un personaje lo ideal sería construirla a partir de la deformación adecuada de una esfera NURBS.

Observe que por la definición de la superficie $S(x, y)$ dada por (3.6), cada punto $\mathbf{P}_{i,j}$ de la malla de control rectangular tendrá incidencia solo sobre un subconjunto de Ω ; esto es, donde las funciones base $R_{i,j}(x)$ no se anulan. Sabemos que $B_{i,r}(x)$ está definida sobre $[t_i, t_{i+r}]$, mientras que $B_{j,s}(y)$ lo está sobre $[u_j, u_{j+s}]$, luego el producto $B_{i,r}(x)B_{j,s}(y)$ estará definido sobre el rectángulo $[t_i, t_{i+r}] \times [u_j, u_{j+s}]$, así, el punto de control $\mathbf{P}_{i,j}$ controla localmente solo la sección de la superficie que se genera cuando los parámetros viven en dicho rectángulo. Claramente, manipular el factor de incidencia $w_{i,j}$ de los puntos que forman el arreglo rectangular que controlan la superficie, tendrá como resultado que la misma se acerque o se aleja de dichos puntos. Al igual que las curvas NURBS, los vectores nodales T y U son en general no periódicos y no uniformes, el fijar los nodos extremos con multiplicidad igual al orden de cada dirección, r en la dirección x y s en la dirección y , hace que la superficie pase por los puntos extremos de su malla de control, esto es, $\mathbf{P}_{0,0}$, $\mathbf{P}_{0,m}$, $\mathbf{P}_{n,0}$, y $\mathbf{P}_{n,m}$, están sobre la superficie $S(x, y)$.

Análogo al caso de las curvas, manipular los vectores nodales, T y U , los puntos de la malla de control, y los pesos asociados a cada uno de ellos, constituyen las formas más efectivas de manipular la superficie de forma que puedan generar todo tipo de formas en el espacio tridimensional. La figura (3.14) muestra una superficie NURBS de orden cuatro en cada dirección controlada por una malla compuesta de 36 puntos, en ella cada uno de los factores de incidencia $w_{i,j}$ tiene valor igual a uno.

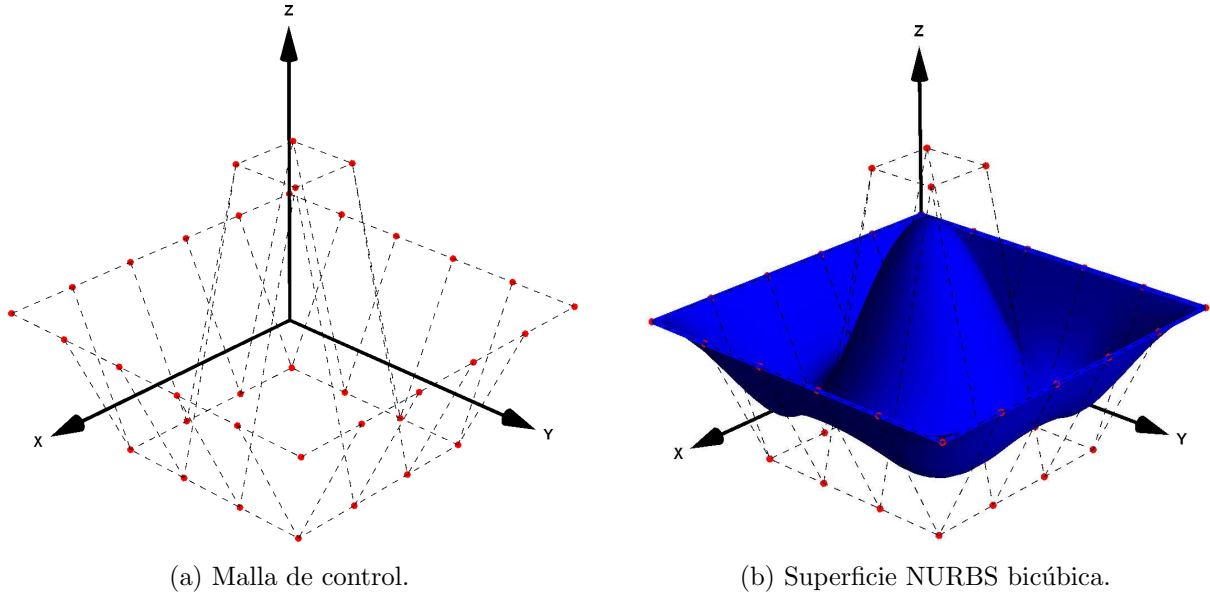


Figura 3.14: Las figuras (a) y (b) muestran la malla de control y la superficie NURBS generada a partir de la misma utilizando funciones base B-spline cúbicas sobre los vectores nodales dados por $T = \{0, 0, 0, 0, \frac{1}{4}, \frac{3}{4}, 1, 1, 1, 1\} = U$.

La cúspide que está presente en la región central de la superficie está siendo controlada por los cuatro puntos $\mathbf{P}_{i,j}$ de la malla de control que están situados justo encima de ella, al cambiar los factores de incidencia $w_{i,j}$ de dichos puntos podremos hacer que la superficie se acerque o se aleje de los mismos, lo cual se traducirá en tener una cúspide más o menos pronunciada. La figura (3.15) muestra las superficies resultantes al cambiar homogéneamente los pesos asociados a cada uno de estos puntos.

Al igual que en el caso de las curvas NURBS, cuando se requiere construir una superficie cerrada la estrategia será repetir puntos extremos de la malla de control, esto es, se deben repetir las filas o columnas externas del arreglo de control. Cabe mencionar que en los procesos de animación las superficies NURBS cerradas en las que coinciden solo la primera y la última fila (o columna) de la malla de control pueden presentar resultados no deseados, la superficie podría presentar discontinuidades en los bordes de coincidencia durante dicho proceso. Para contrarrestar estos inconvenientes muchos sistemas emplean rutinas que permiten enrollar la malla [19]; por ejemplo, en un arreglo de control de siete filas, el sistema podría cerrar la superficie haciendo coincidir la última fila con la segunda, y la primera con la quinta.

Por otro lado, las superficies que no son cerradas como la de la figura (3.13b) les permiten a los modeladores representar con facilidad una amplia gama de objetos inorgánicos, por ejemplo, les hacen posible modelar con mucha fidelidad los pliegues presentes en la ropa del personaje de la figura (2.3). Superficies que deban representar las formas de este tipo de objetos a menudo son manipuladas en mayor medida a través de los vectores nodales asociados a la misma, pues reducir la diferenciabilidad en ciertos puntos de la superficie puede permitir

construir figuras con secciones o bordes semi punteagudos. Considere por ejemplo la forma en la que un modelador de personajes debe construir los dedos de la mano de una de sus creaciones, para reproducirlos de forma creíble debe considerar las uniones entre los falanges cuando su personaje esté sosteniendo algo, tal vez un celular, a lo largo de dichas uniones entonces la superficie que modela su pieza debe perder suavidad para lograr un resultado satisfactorio.

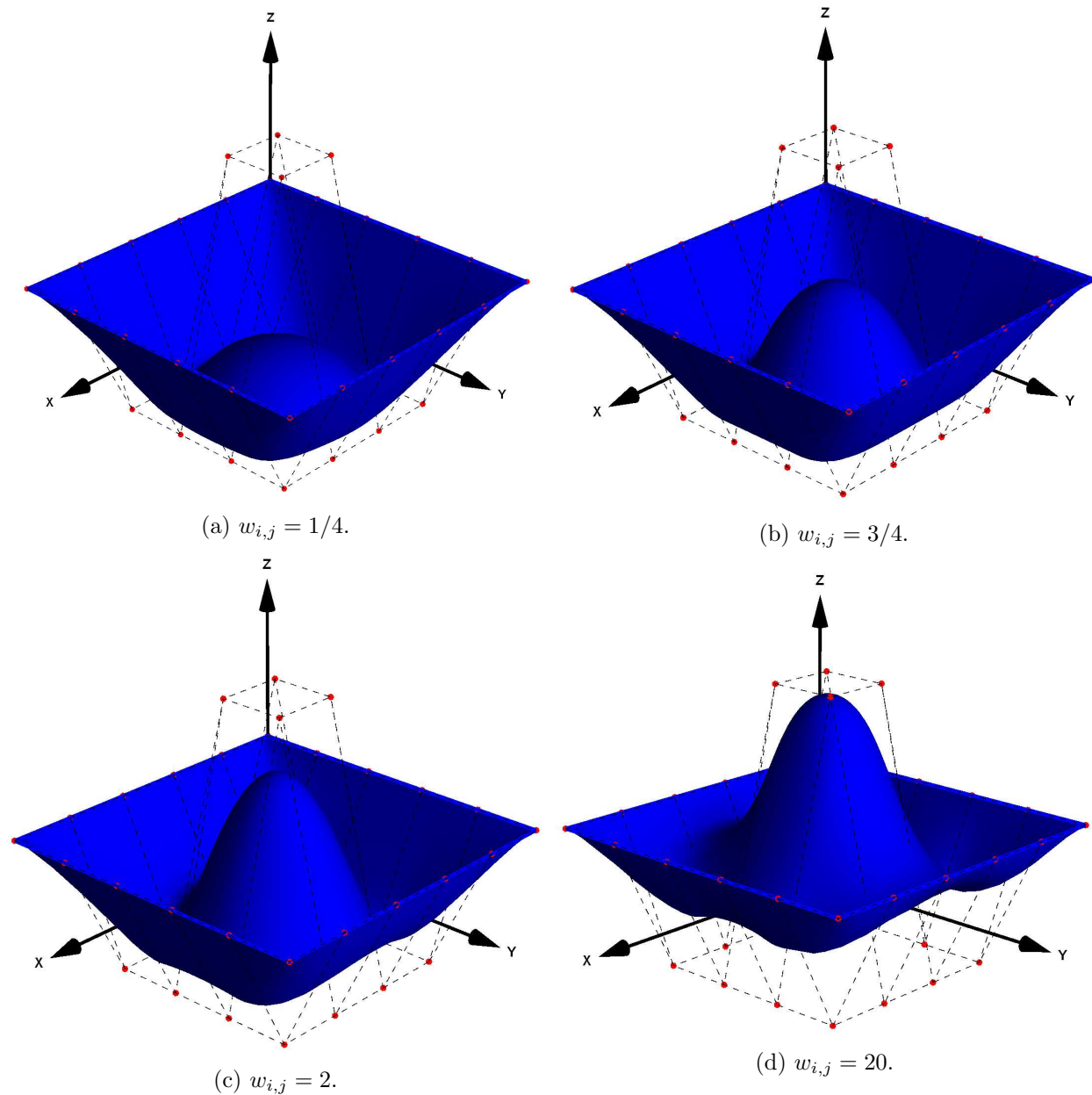


Figura 3.15: Efecto del cambio de los factores de incidencia $w_{i,j}$ en la región central de la superficie NURBS.

Ilustremos lo anterior con una aproximación simple a tal situación. La figura (3.16) presenta

una superficie NURBS¹ cilíndrica, dicha superficie fue construida con funciones B-spline de orden tres y tomando como vectores nodales

$$T = \left\{ 0, 0, 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1, 1, 1 \right\}, \quad \text{y} \quad U = \left\{ 0, 0, 0, \frac{1}{4}, \frac{1}{2}, \frac{1}{2}, \frac{3}{4}, 1, 1, 1 \right\}.$$

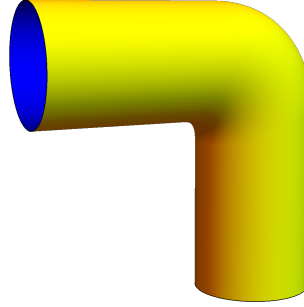


Figura 3.16: Superficie NURBS cilíndrica.

Observe la suavidad que presenta la superficie en la parte en la cual se dobla el cilindro, para lograr un doblez más marcado en dicha sección de la misma se puede aumentar la multiplicidad de los elementos del vector nodal. La figura (3.17) muestra el resultado que se obtiene al triplicar la multiplicidad inicial de t_3 en T .

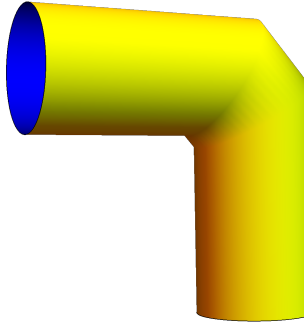


Figura 3.17: Superficie NURBS cilíndrica manipulada.

Note como la superficie perdió suavidad en esa región, permitiendo así un modelo de representación un poco más fiel al deseado. Otra estrategia para llevar a cabo esta tarea podría ser manipular los puntos de la malla de control que rodeen esa sección, o disminuir el factor de incidencia de los mismos. Como lo mencionamos antes, una mezcla eficaz de las formas de manipular una superficie NURBS es la que permitirá al final construir el mejor modelo de

¹Las curvas y superficies de este capítulo fueron generadas utilizando las funciones *BSplineBasis*, *BSplineCurve*, y *BSplineSurface* que vienen implementadas en el software *Wolfram Mathematica 12*.

representación posible que satisfaga los requerimientos exigidos en un proyecto. La articulación eficiente de estas herramientas de edición por parte de los profesionales del área, es una de las razones que han hecho posible la masificación de productos que tienen su génesis en los sistemas CAGD. Uno de los hechos más relevantes respecto a estas superficies es el de que fueron empleadas a fondo para poder producir la primera película completamente animada por computadora de la historia, *Toy Story* [7]. La versatilidad de estas superficies era ideal para modelar objetos inorgánicos, objetos que eran el común denominador al tratarse de una película sobre juguetes.

Sin embargo, ellas también presentan algunas desventajas, la más importante está ligada a la restricción topológica² inherente que presentan: la superficie de control de una superficie NURBS es una malla de control rectangular, esto significa que al crear una forma orgánica compleja como un personaje humano, es tarea del modelador dividir esa forma compleja en un colección de piezas rectangulares, cada una de las cuales constituirá una NURBS por separado. Además, si se requiere más detalle en alguna sección de la malla, se requiere entonces la adición de puntos de control adicionales, es decir, se deben agregar filas y columnas adicionales en el arreglo para satisfacer la restricción de poseer una malla de control rectangular, lo cual puede conducir rápidamente a modelos difíciles de manejar.

Al tratarse de curvas y superficies paramétricas, las NURBS presentan una lista de importantes propiedades que se han estudiado en mayor y menor medida por diferentes autores, como por ejemplo el significado de sus derivadas, unicidad de la representación, transformaciones, continuidad, etc.; sin embargo, la discusión de estas escapa del objetivo principal de este documento y por tanto no fueron discutidas. Una discusión formal de estos aspectos puede encontrarse en [3, 14, 17, 19], [14] y [19] en particular presentan un poco de la historia detrás de los hechos y protagonistas que influyeron directamente en el desarrollo de la representación de curvas y superficies en los entornos CAGD.

²En este contexto, la topología se refiere al conjunto de componentes geométricas de la malla de control tridimensional; vértices, bordes, y caras.

4 Superficies NURBS en sistemas CAGD

Ahora mostraremos la forma en la cual se implementan las superficies NURBS en un sistema CAGD, para ello utilizaremos el software de diseño *Autodesk Maya* (Versión estudiantil 2018), este programa es uno de los más utilizados en los estudios de animación y el mundo del diseño en general. Con él se diseñan los personajes, piezas, estructuras y escenarios presentes en distintas producciones audiovisuales.

Mostraremos inicialmente un ejemplo de las superficies NURBS primitivas que vienen implementadas en el programa, en este caso será la esfera unitaria el objeto de muestra. Este ejemplo nos permitirá familiarizarnos con los parámetros que caracterizan las superficies que son generadas dentro del programa y nos permitirá compararlos con los que hemos discutido en los capítulos anteriores.

La figura (4.1) muestra la interfaz principal del programa, en el es posible elegir la opción de crear superficies NURBS que vienen creadas de forma genérica dentro del mismo. En ella se muestra el menú de formas a partir de las cuales puede comenzar un diseñador a modelar alguna pieza de su interés.

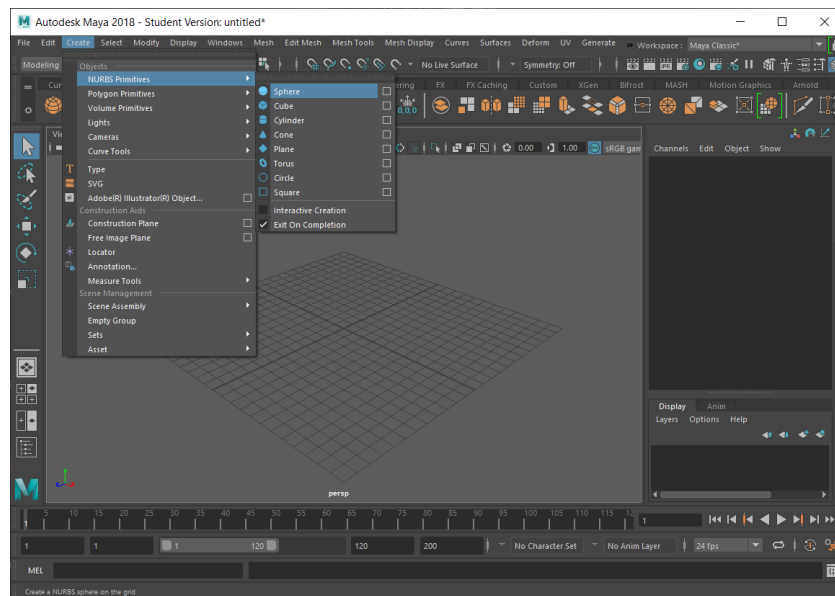
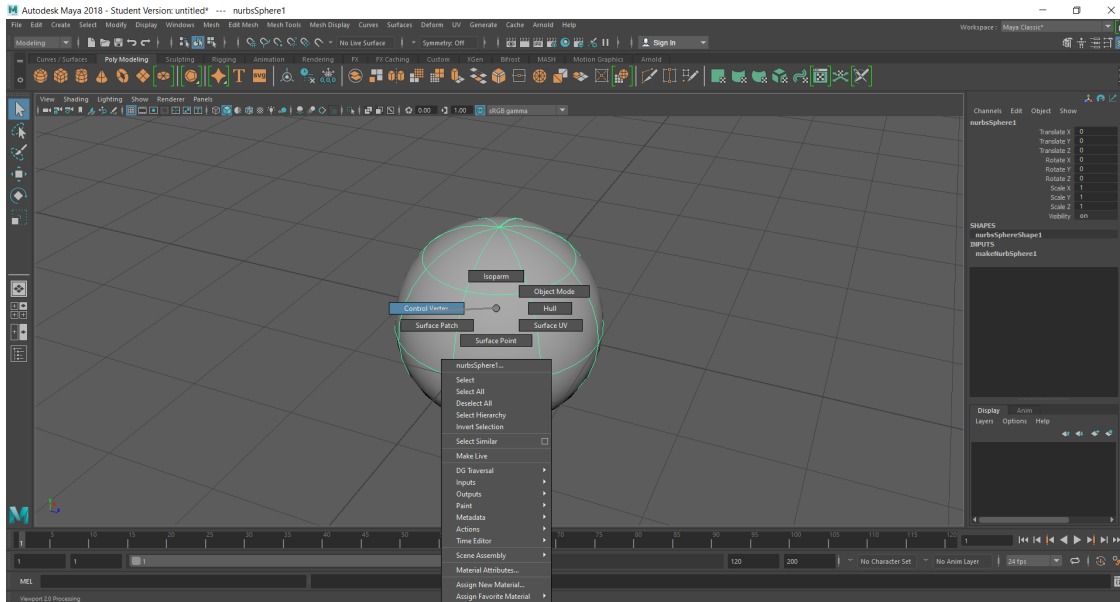


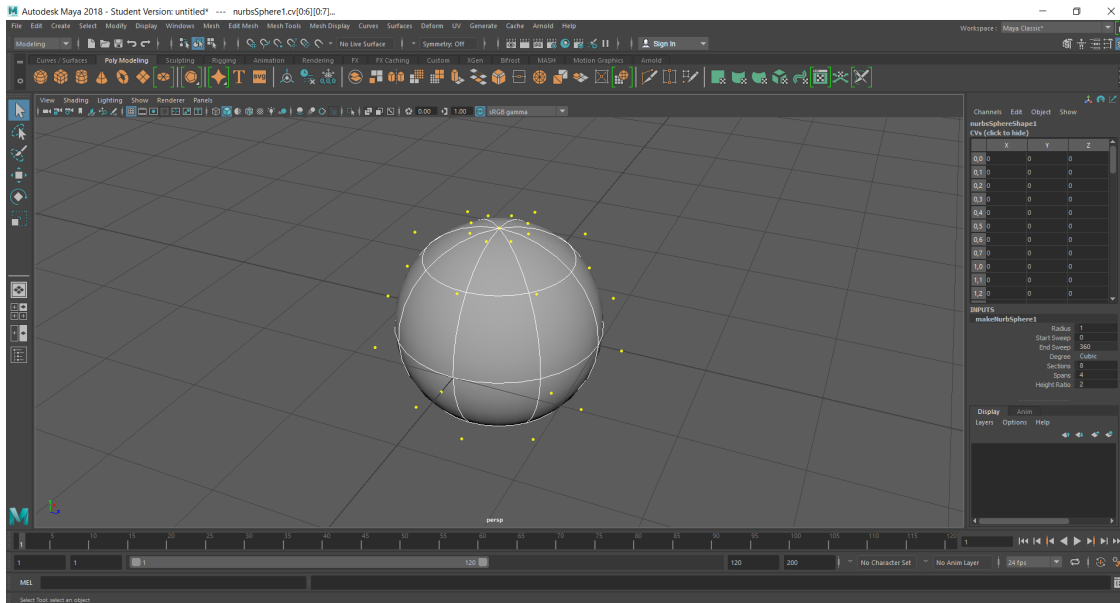
Figura 4.1: Interfaz del programa. Creación de Superficies NURBS primitivas.

Una vez creada la superficie el programa permite manipularla inmediatamente, para ello

permite tratar al objeto como un todo o seleccionar atributos de ella para comenzar a llevar a cabo el proceso de modelado. En particular se encuentra la opción de manipular los puntos de la malla de control denominada *Control Vertex* (ver figura (4.2)), en este caso el sistema genera un arreglo rectangular de $7 \times 8 = 56$ puntos de control.



(a) Esfera NURBS generada y menú de edición.



(b) Esfera NURBS y los puntos de su malla de control.

Figura 4.2: Esfera NURBS primitiva.

La pestaña de la parte derecha de la interfaz, editor de atributos, permite ver la forma como están referenciados cada uno de estos puntos. El arreglo varía desde el punto $P_{0,0}$

hasta $\mathbf{P}_{6,7}$. Por efectos de cálculo interno las coordenadas de estos puntos en la escena tridimensional en la que se sitúan aparecen puestas en el origen, las mismas solo se alterarán mostrando su posición real una vez se comiencen a mover durante el proceso de modelado. Dicha ventana muestra además información acerca del grado de los polinomios que generan la superficie, así como información sobre los vectores nodales. La figura (4.3) muestra parte de la información contenida en este editor, allí U y V son los vectores nodales, que en este caso son no periódico (open) y periódico (periodic), respectivamente. La subpestaña *Components* muestra los índices y las coordenadas de cada uno de los puntos que componen la malla de control de la esfera. Note además como este editor indica que el grado de los polinomios empleados para la construcción es tres en ambas direcciones; $Degree\ U\ V = 3\ 3$.

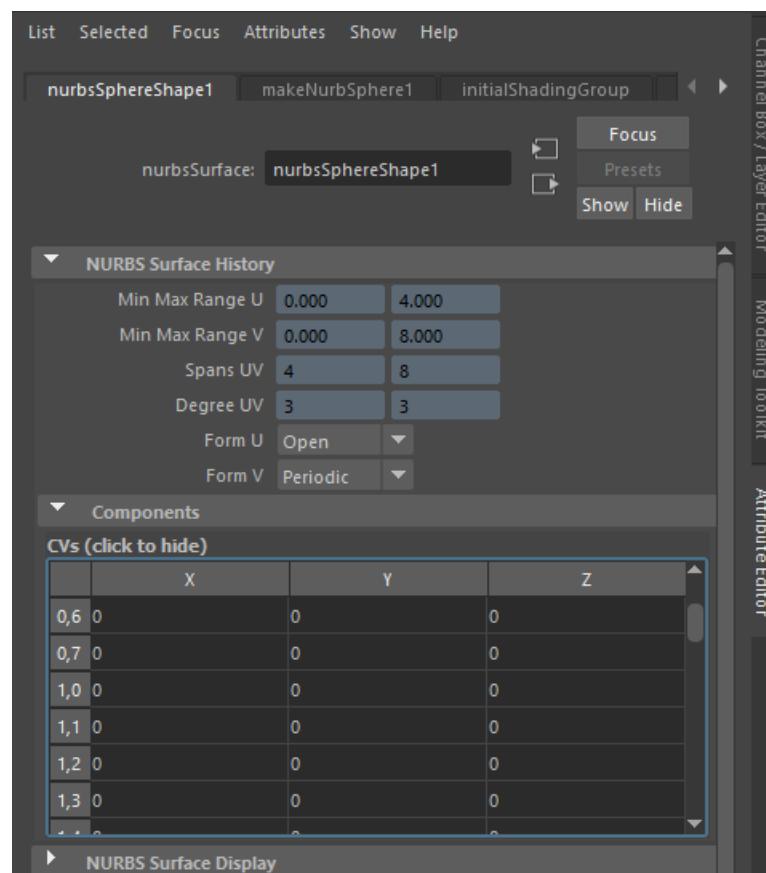


Figura 4.3: Editor de atributos.

Observe que en este caso estamos tratando con una superficie cerrada, es decir los puntos de control de sus filas o columnas deben estar repetidas al inicio o al final del arreglo para poder producir tal forma, más aún, el arreglo rectangular que controla la esfera más que una malla es un polígono de control. El programa permite ver este polígono de forma individual como lo muestra la figura (4.4).

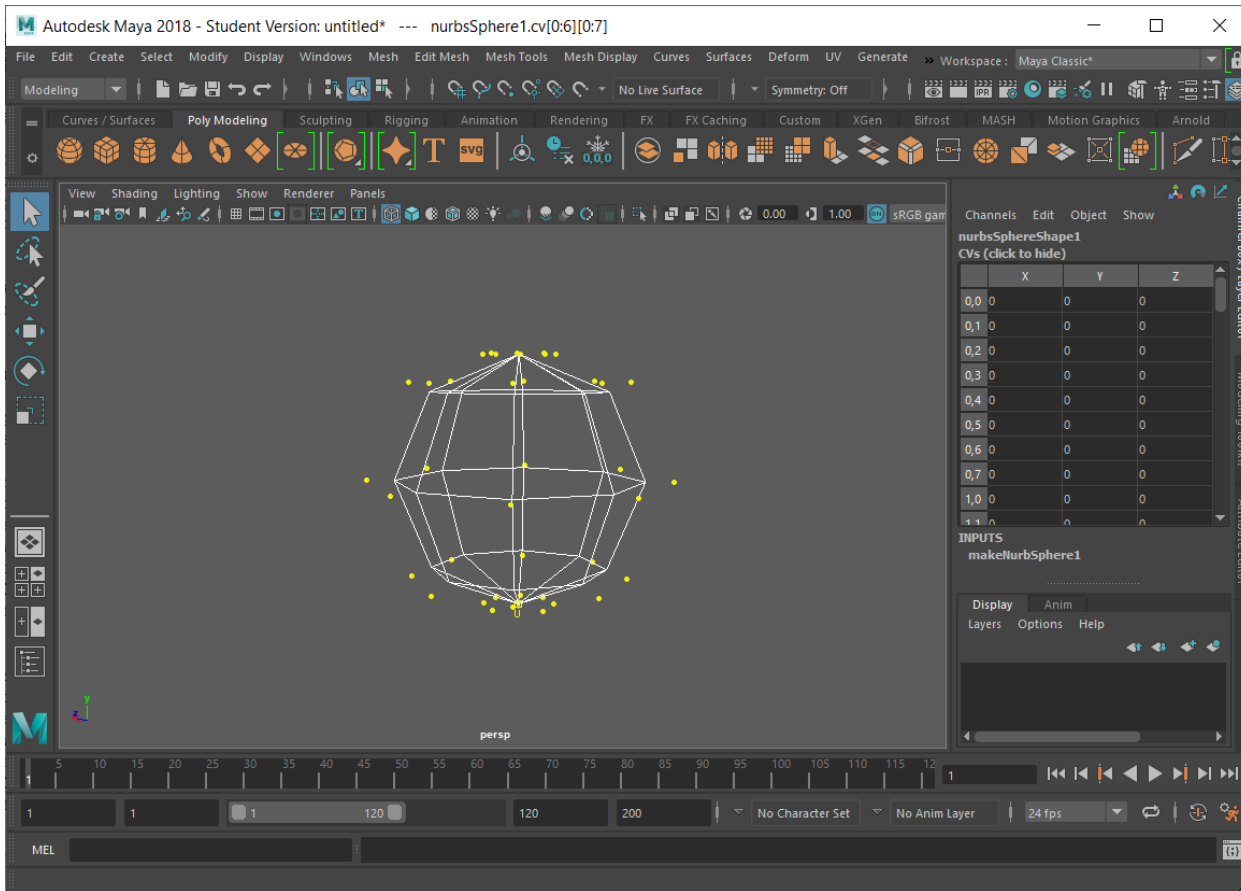
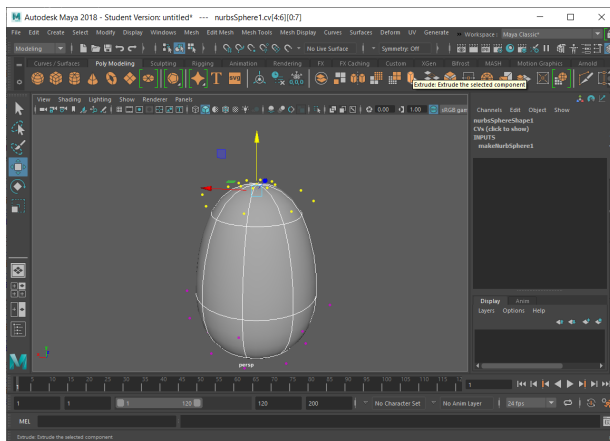
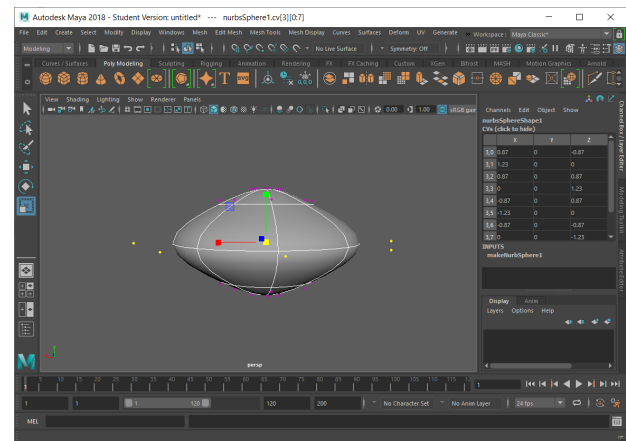


Figura 4.4: Polígono de control de la esfera.

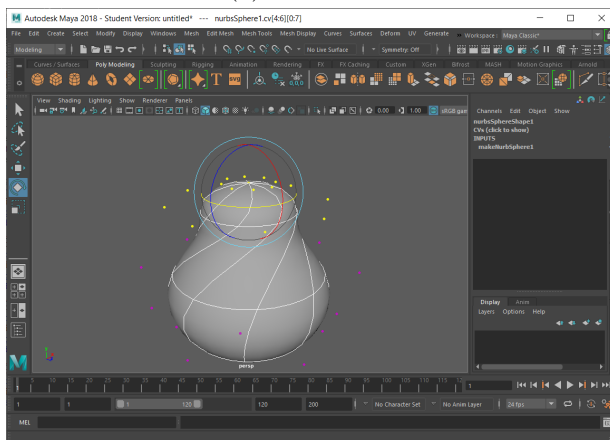
Como lo resaltamos en líneas anteriores, las propiedades de las funciones B-spline que definen estas superficies hacen que las mismas sean muy versátiles, la posibilidad de poder controlar localmente la superficie hace que se puedan generar todo tipo de formas, en especial inorgánicas, a partir de deformaciones de las superficies primitivas. Como lo mencionamos anteriormente, a este tipo de modelado geométrico se le conoce con el nombre de *NURBS Modelling*. Este programa en específico cuenta con herramientas bastante interactivas e intuitivas que permiten llevar a cabo de forma rápida este proceso, en particular las de mover, escalar, y rotar puntos, bordes, o caras seleccionadas de una superficie son bastante útiles para lograr las formas deseadas. La figura (4.5) muestra la transformación de la esfera inicial cuando se aplican traslaciones, escalamientos, y rotaciones de los puntos de control, la misma es evidencia de la propiedad de invariancia bajo transformaciones que poseen tales superficies; para transformarla basta transformar los puntos de control de su malla nodal. Observe la marcada diferencia que hay entre la esfera inicial y las superficies obtenidas al aplicar las transformaciones sobre sus puntos de control en cada caso.



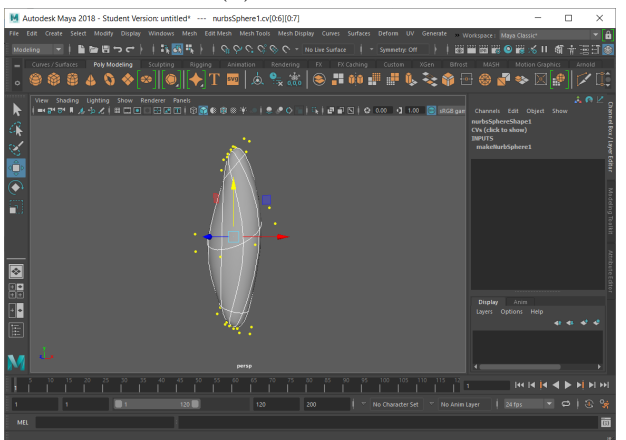
(a) Traslación.



(b) Escalamiento.



(c) Rotación.



(d) Combinación de transformaciones.

Figura 4.5: Traslación, escalamiento, y rotación de los puntos de control.

Modelar superficies utilizando esta técnica es una de las formas más sencillas en las que un diseñador puede trabajar, dependiendo de los resultados que quiera lograr puede optar por utilizar solo esta técnica o combinarla con otras, como por ejemplo el modelado a partir de poligonales.

Combinar apropiadamente estas herramientas y agregar atributos a la superficie primitiva a partir de la cual se comienza a modelar, como por ejemplo aumentar la densidad de los puntos en el polígono de control o manipular las propiedades de los vectores nodales, es la que permite entonces generar superficies de objetos de gran complejidad. Para ilustrar este hecho observe las figuras (4.6) y (4.7), en ellas se muestra el resultado de combinar distintas técnicas de modelado a partir de superficies primitivas para modelar la cabeza de un personaje. Note además como la interfaz del programa le permite al usuario navegar entre las vistas superior, frontal, lateral y de perspectiva del modelo; esta función le facilita en particular la selección de puntos de control específicos que controlan determinadas secciones de la superficie.

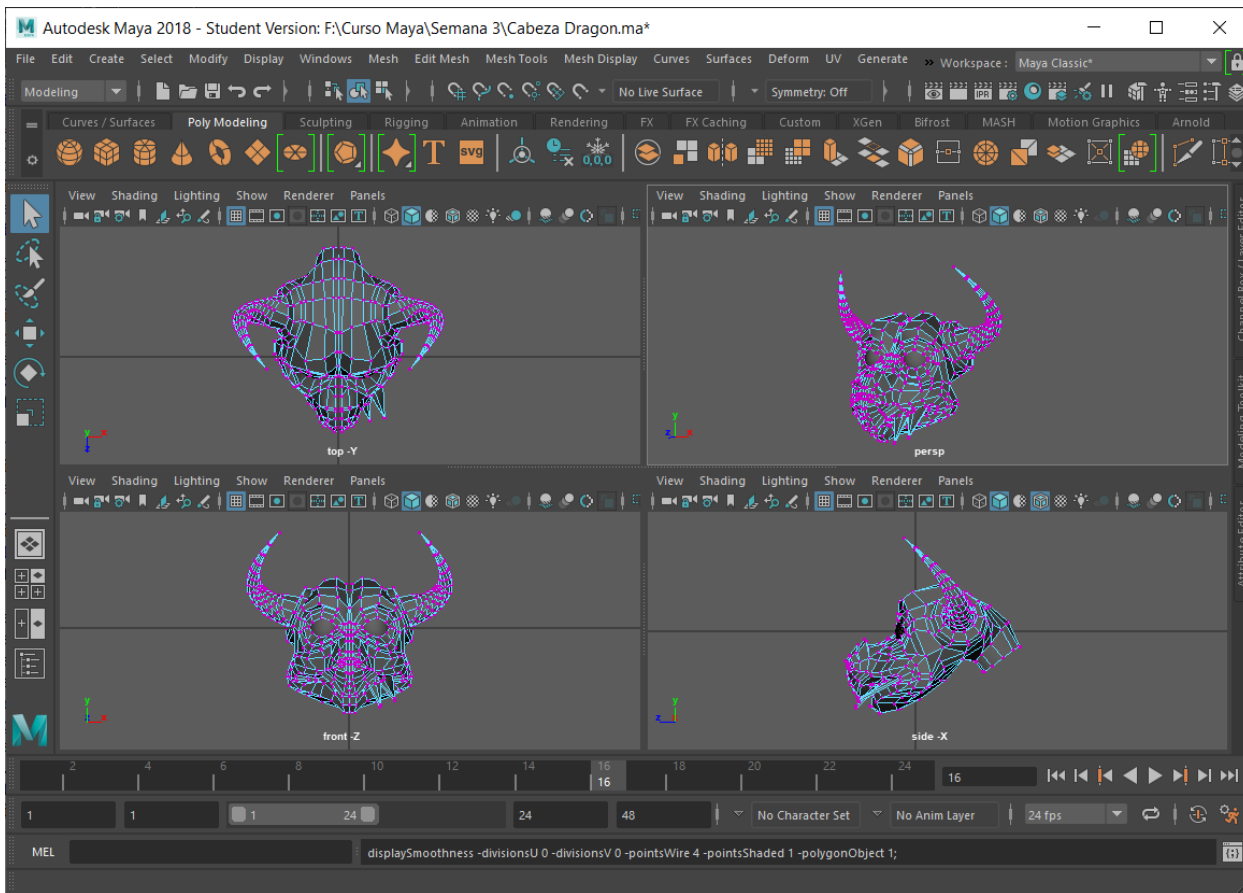


Figura 4.6: Polígono de control de la cabeza del personaje.

Si bien estas estrategias son efectivas y producen buenos resultados finales, no siempre es práctico modelar a partir de una única primitiva toda una figura, si esta es muy compleja al final se tendrá que lidiar con una malla de control igual de compleja, lo cual puede llegar a ser bastante tedioso para el modelador en caso de que quiera editar su producto final. Una manera eficiente de modelar geoméricamente superficies de este tipo es hacerlo por secciones, dividir la pieza para modelar primero objetos menos complejos que puedan manipularse con mayor facilidad y que puedan articularse luego para formar el modelo final.

El personaje¹ mostrado en las figuras (4.8), (4.9) y (4.10) fue construido en varias secciones, las cuales posteriormente se combinaron en una sola para obtener el resultado final, modelarlo a partir de una única primitiva hubiera resultado bastante dispendioso debido a la complejidad que el mismo posee en varias partes de su cuerpo. Si bien este personaje se clasifica dentro de las formas orgánicas, el mismo no precisa un nivel de detalle muy elevado en la superficie que lo modela, es por eso que las superficies NURBS fueron suficientes en este caso para representar el modelo.

¹Diseño realizado por el autor como proyecto final de un curso de modelado geométrico.

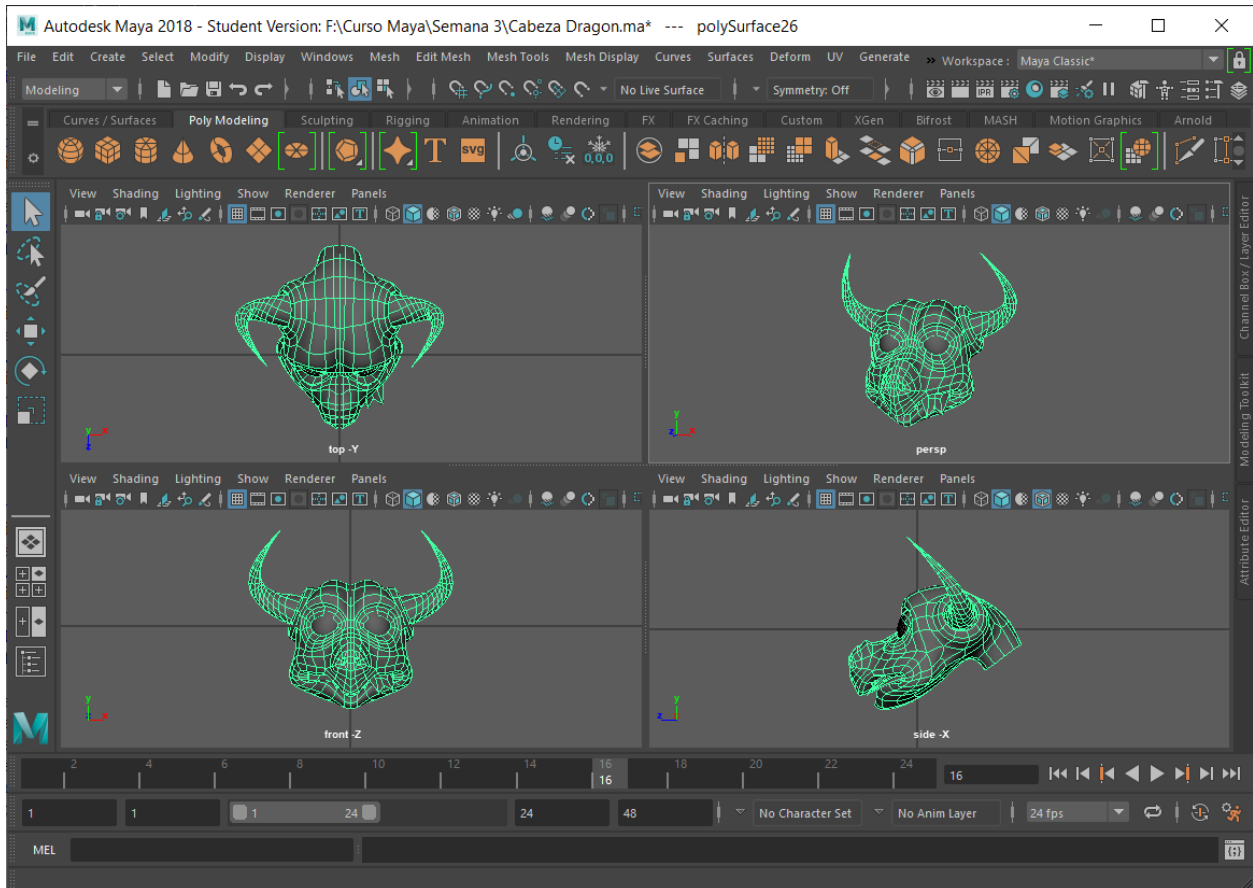


Figura 4.7: Superficie de la cabeza del personaje.

Este par de ejemplos ilustran de forma clara entonces la versatilidad que las superficies NURBS poseen como herramienta de modelado geométrico, el hecho de que los puntos de la malla de control tengan control local sobre la superficie que se modela, debido a las B-splines que las definen, es una de las características más notables que hizo de estas el estándar de representación de superficies en los sistemas CAGD, al menos en cuanto a objetos inorgánicos se refiere. Los objetos orgánicos que quieran ser representados con estas superficies no pueden contar con un nivel de detalle demasiado alto, un ejemplo de ello son los personajes humanos de la película animada *Toy Story*. Como lo mencionamos al final del capítulo anterior, la restricción topológica de estar amarrada a un arreglo rectangular hace que estas no sean ideales para representar superficies que requieran mucho nivel detalle como el que se presenta en la mayoría de objetos orgánicos, cuando se requiere representar formas de este tipo el protagonismo lo toman las superficies de subdivisión, las cuales también están construidas con base en funciones B-spline pero sin restricción topológica de algún tipo [6, 11, 12].

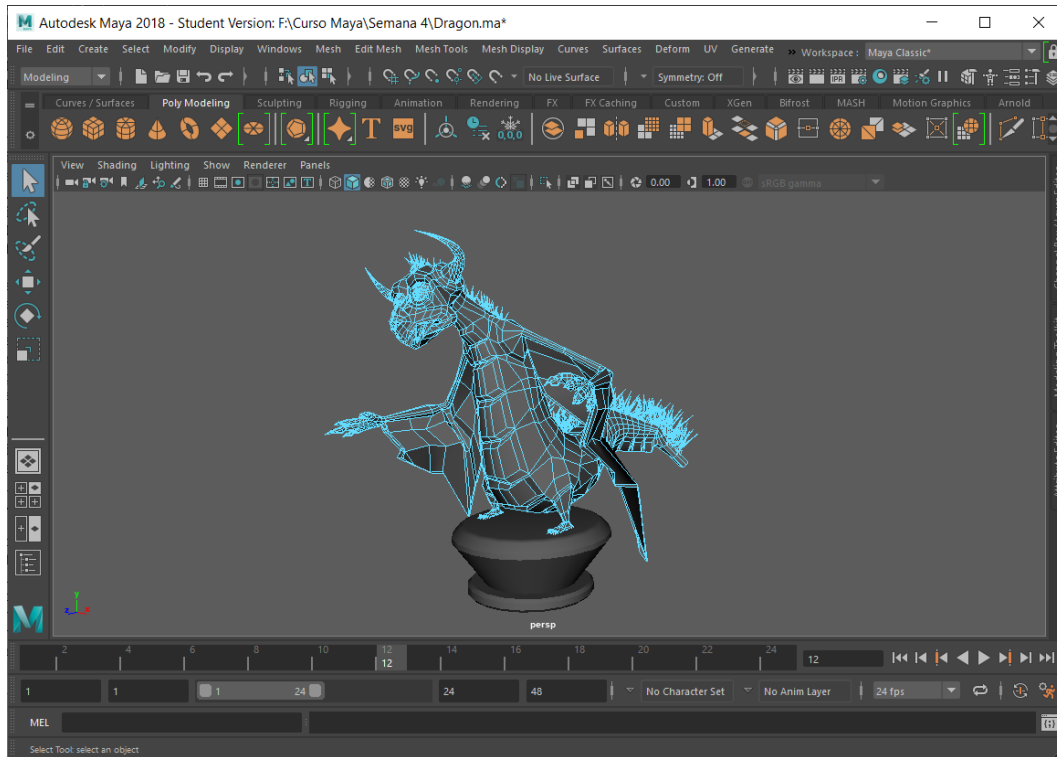


Figura 4.8: Polígono de control del personaje.

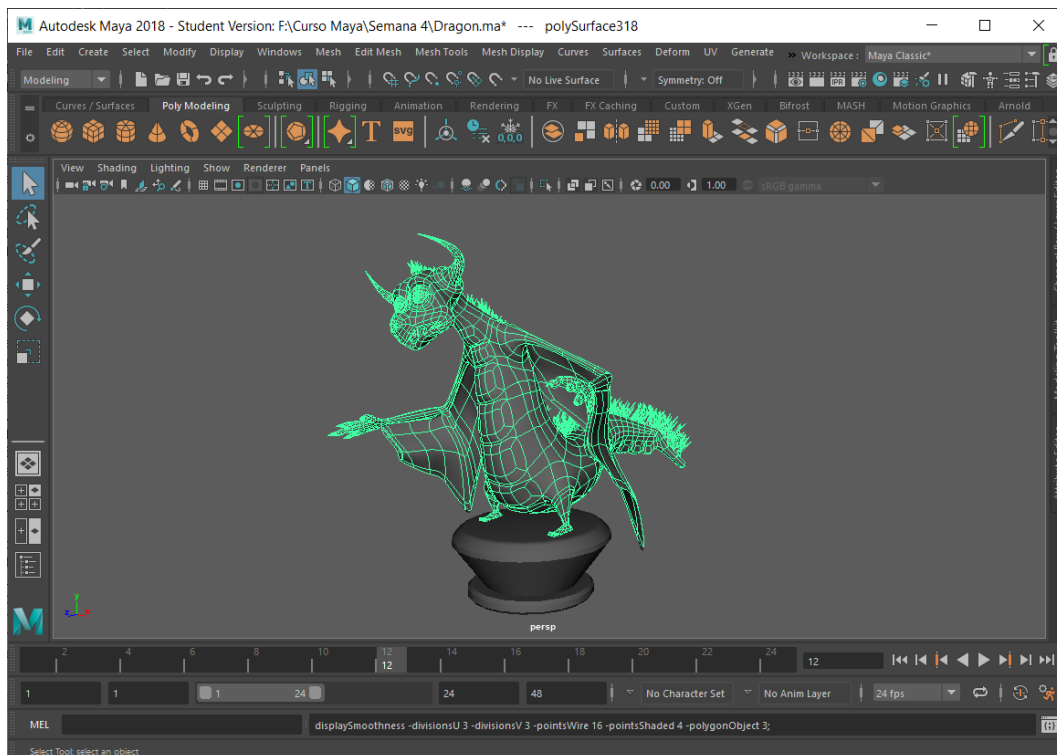


Figura 4.9: Superficie del personaje.



Figura 4.10: Render final del personaje.

5 Conclusiones

En este trabajo se ha presentado un compendio de los resultados de convergencia de las técnicas de interpolación más relevantes, interpolación de Lagrange, de Newton, de Hermite y tipo spline. Principal atención hemos prestado a la interpolación tipo spline por su papel predominante en el diseño y construcción de objetos animados en el área de la computación gráfica.

Desde el punto teórico hemos presentado y discutido los estimativos del error en cada uno de los tipos de interpolación; en detalle hemos discutido las cotas del error para los splines naturales y periódicos, y su relación con la curvatura del objeto que se desea interpolar. Para explicar la conexión entre la interpolación basada en funciones B-splines y el modelamiento de figuras en el computador se introdujeron las superficies NURBS, que están incorporadas en cualquiera sistema estándar CAGD. Resaltamos que la demostración del teorema de convergencia de la interpolación spline se fundamenta en resultados básicos clásicos del cálculo diferencial, y que su aplicación se esconde detrás de las animaciones que observamos en un cine. En este sentido el objetivo de la tesis se cumplió.

Como una debilidad del trabajo podemos mencionar que hace falta una descripción más detallada de las propiedades geométricas de las superficies NURBS, del manejo apropiado de las mallas de control, y del proceso de modelaje de la figura. Además, no se incluyó en este trabajo la técnica de subdivisión de superficies, que ha ganado especial relevancia últimamente con la disponibilidad de sistema de cómputo de alto rendimiento y servicios de renderizado en la nube, y que permite la creación de figuras mucho más realistas.

Bibliografía

- [1] ANDERSSON, Lars-Erik ; F. STEWART, Neil: *Introduction to the Mathematics of Subdivision Surfaces*. Philadelphia, PA, USA : Society for Industrial and Applied Mathematics (SIAM), 2010. – ISBN 978-0-89871-697-9
- [2] AUTODESK, INC. *Subdivision Surface Modeling*. <http://download.autodesk.com/us/support/files/subds.pdf>. 2011
- [3] BARTELS, Richard H. ; BEATTY, John C. ; BARSKY, Brian A.: *An Introduction to Splines for Use in Computer Graphics & Geometric Modeling*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1987. – ISBN 0-934613-27-3
- [4] BINEGAR, Birne. *Numerical Analysis - Lecture Notes: Errors in Polynomial Interpolation*. 1998
- [5] DE BOOR, Carl: *A practical guide to splines; rev. ed.* Berlin : Springer, 2001 (Applied mathematical sciences)
- [6] CATMULL, E ; CLARK, J: Recursively generated B-spline surfaces on arbitrary topological meshes. En: *Computer-Aided Design* 10 (1978), Nr. 6, p. 350 – 355. – ISSN 0010-4485
- [7] CHOPINE, Ami: *3D Art Essentials: The Fundamentals of 3D Modeling, Texturing, and Animation*. 1st. Focal Press, 2011. – ISBN 0240814711, 9780240814711
- [8] COONS, Steven A.: *Surfaces for Computer-Aided Design of Space Forms*. Cambridge, MA, USA : Massachusetts Institute of Technology, 1967. – Informe de Investigación
- [9] COX, M. G.: The Numerical Evaluation of B-Splines*. En: *IMA Journal of Applied Mathematics* 10 (1972), Nr. 2, p. 134-149
- [10] DEMANET, Laurent. *Introduction to Numerical Analysis - Lecture Notes: Chapter 3 - Interpolation*. 2012
- [11] DEROSE, Tony ; KASS, Michael ; TRUONG, Tien: Subdivision Surfaces in Character Animation. En: *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*. New York, NY, USA : ACM, 1998 (SIGGRAPH '98). – ISBN 0-89791-999-8, p. 85-94

- [12] DOO, D. ; SABIN, M.: Behaviour of recursive division surfaces near extraordinary points. En: *Computer-Aided Design* 10 (1978), Nr. 6, p. 356 – 360. – ISSN 0010-4485
- [13] FANG, Lian: A rational quartic Bézier representation for conics. En: *Computer Aided Geometric Design* 19 (2002), Nr. 5, p. 297 – 312. – ISSN 0167-8396
- [14] FARIN, G. ; HOSCHEK, J. ; KIM, M.S.: *Handbook of Computer Aided Geometric Design*. Amsterdam, The Netherlands : Elsevier Science B.V., 2002. – ISBN 9780080533407
- [15] HALL, Charles A. ; MEYER, W.Weston: Optimal error bounds for cubic spline interpolation. En: *Journal of Approximation Theory* 16 (1976), Nr. 2, p. 105 – 122. – ISSN 0021-9045
- [16] PATRIKALAKIS, Nicholas M. ; MAEKAWA, Takashi: *Shape Interrogation for Computer Aided Design and Manufacturing*. 1. Springer-Verlag Berlin Heidelberg, 2010. – ISBN 364204073X,9783642040733
- [17] PIEGL, Les ; TILLER, Wayne: *The NURBS Book (2Nd Ed.)*. New York, NY, USA : Springer-Verlag New York, Inc., 1997. – ISBN 3-540-61545-8
- [18] PIXAR ANIMATION STUDIOS. *OpenSubdiv*. <http://graphics.pixar.com/opensubdiv/docs/intro.html>. 2012
- [19] ROGERS, David F.: *An Introduction to NURBS: With Historical Perspective*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 2001. – ISBN 1-55860-669-6
- [20] SALOMON, David: *Curves and Surfaces for Computer Graphics*. Secaucus, NJ, USA : Springer-Verlag New York, Inc., 2005. – ISBN 0387241965
- [21] SCHMIDT, Jochen W.: Zur Konvergenz von kubischen Interpolationssplines. En: *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik* 58 (1978), Nr. 2, p. 109-110
- [22] SEDERBERG, Thomas W.: *Computer Aided Geometric Design*. Brigham Young University - BYU ScholarsArchive, 2012
- [23] STOER, J. ; BULIRSCH, R.: *Introduction to Numerical Analysis*. New York, NY, USA : Springer-Verlag New York, Inc., 2002. – ISBN 978-0-387-21738-3
- [24] VELÁSQUEZ ZAPATEIRO, Jorge ; OBESO FERNÁNDEZ, Virgilio: *Análisis Numérico: Notas de clase*. Ediciones Uninorte, 2007. – ISBN 9789588252582